A Dual-Stream Recurrent Neural Network for Student Feedback Prediction using Kinect

Shanfeng Hu[1,2], Hindol Bhattacharya[3], Matangini Chattopadhyay[3], Nauman Aslam[1], and Hubert P. H. Shum[1,*]

[1]Northumbria University, UK, Email: {shanfeng.hu, nauman.aslam, hubert.shum}@northumbria.ac.uk
[2]Beihang University, China, Email: hu_shan_feng@buaa.edu.cn
[3]Jadavpur University, India, Email: {hindolbhattacharjee12, matanginic}@gmail.com
[*]Corresponding author

***Abstract*—Convenience internet access and ubiquitous computing have opened up new avenues for learning and teaching. They are now no longer confined to the classroom walls, but are available to anyone connected to the internet. E-learning has opened massive opportunities for learners who otherwise would have been constrained due to geographical distances, time and/or cost factors. It has revolutionized the learning methods and represents a paradigm shift from traditional learning methods. However, despite all its advantages, e-learning is not without its own shortcomings. Understanding the effectiveness of a teaching strategy through learner feedback has been a key performance measure and decision making criteria to fine tune the teaching strategy. However, traditional methods of collecting learner feedback are inadequate in a geographically distributed, virtual setup of the e-learning environment. Innovative and novel learner feedback collection mechanism is hence the need of the hour. In this work, we design and develop a deep learning based student feedback prediction system by recognizing the subtle facial motions during a student's learning activity. This allows the system to infer the needs of the learners as if it is a real-human teacher in order to provide the appropriate feedback. We propose a recurrent convolutional neural network structure to understand the color and depth streams of video taken by an RGB-D camera. Experimental results have shown that our system achieve high accuracy in estimating the feedback labels. While we demonstrate the proposed framework in an e-learning setup, it can be adapted to other applications such as in-house patient monitoring and rehabilitation training.**

Keywords: feedback prediction, Kinect, convolutional neural networks (CNNs), recurrent neural networks (RNNs), deep learning, e-Learning

I. Introduction

Learning is an integral part of the intellectual development of a human being. Indeed learning forms the foundation on which we strive to progress towards a better future. Knowledge transfer in the traditional classroom-based environment has its limitations. To democratize learning, e-learning offers a cheap solution to educate a maximum number of learners with relatively less effort and constraints. However, e-learning has its own set of problems, especially when it comes to collecting user feedback. A key teaching strategy involves getting real-time feedback from learners regarding their attention level, interest and overall effectiveness of the teaching strategy. This helps educators to address the inadequacies of the lesson delivery system for an improved learning experience. In a classroom environment, this is relatively easy. A teacher can follow the visual cues of facial expressions of the learner to determine whether a learner has a favourable or unfavourable reaction to the lesson being taught.

While such visual cues could be obtained at ease in a classroom environment where both teacher and learner are interacting face-to-face in real time, things are different in an e-learning environment. In e-learning, study materials, such as lecture videos, interactive simulations, etc. are pre-recorded and delivered to users distributed across wide geographic locations. Such interaction of learner and teacher makes real-time feedback collection and evaluation of the effectiveness of the teaching strategy difficult. While static feedback collection mechanism is frequently employed by many e-learning providers such as questionnaire-based feedback, such a strategy has limited usefulness. The necessary questions may not be provided or the learner may also not be able to express their difficulties through such questions and answers. Frequently, the learners avoid providing feedbacks due to sheer apathy. Moreover, such feedbacks are frequently collected after the conclusion of the course, which leaves no room for improvement to the present course. These make such methods ineffective and encourages a research into new solutions.

Picking up visual cues and processing them for understanding the emotion of the subject is a core matter for image processing. A subject's image of the facial expression during learning can be analysed and a deep learning neural network can be trained to categorize the facial expression into one of the multiple emotion classes. Such classifications can give a better, dynamic and real-time feedback of the actual emotional reaction of the learner towards the learning module.

In this work, we propose a feedback system which captures multiple facial expressions of the learner while interacting with an e-learning system. The assumption is that a learner's mental state (like or dislike of a module, finding the module easy or hard etc.) is revealed through these facial expressions. A trained deep learning system is expected to give a more insightful and dynamic feedback from the learner. We have also performed experimental studies to measure the effectiveness of our system, by comparing the predicted feedback to the actual feedback provided by the test subjects. The use of the dual stream convolution neural network, each for colour and depth image has been validated through the results of the experiments

978-1-5386-9141-0/18/$31.00 ©2018 IEEE

performed. We have also validated the effectiveness of the use of multi-step long short-term memory (LSTM) network.

The structure of this paper is as follows. We first review literatures related to student feedback prediction in Section II. We then elaborate the construction of our feedback prediction database using Kinect in Section III. We propose our dual-stream RNN approach for feedback prediction in Section IV. We show experimental results in Section V. We finally conclude this paper and discuss future research directions in Section VI.

II. Related Work

Many studies utilize RGB-D cameras such as Kinect for virtual training and smart environments. For example, an innovate idea of interactive Chinese character learning environment with Kinect was proposed in [1]. It is shown that with a posture reconstruction algorithm, Kinect can be used for monitoring postures healthiness in a workspace environment [2]. The filtered pose graph can be used to achieve posture reconstruction in real-time [3]. In [4], an improved posture classification method was proposed by using Kinect and a max-margin classifier. In this research, we focus on the application of e-learning feedback based on the facial information captured by Kinect.

E-learning through virtual laboratory has been proposed as a cost-effective and scalable alternative to the physical laboratory. It has been shown in the literature that virtual environment based practical demonstration is an effective alternative to physical laboratory demonstration among the young learners [5] - something very suitable for engineering education. Virtual Laboratory can emulate the entire experience using virtual environment only without requiring any physical experiments to be carried out. The TRAILS laboratory that we will be using for our work [6] is an example of such an environment. While in other virtual laboratory implementations, part of the experiment is carried out in physical equipment and the rest uses computer simulation. An example in the literature is the engine calibration lab at the University of Bradford [7].

E-learning based hands-on training and experimental demonstration have been used for numerous purposes. Railway simulation-based training is particularly popular. One such example is SIGMA_RAIL of the Euro-Mediterranean project [8]. In the domain of mechanical engineering, work by Burk et al. [9] is quite promising. This work virtualizes an engine laboratory to demonstrate the powertrain calibration of an engine. The web is the most important and ubiquitous ICT medium today, because of its ease-of-use, wide reach and not requiring any specialized devices. TRAILS - the laboratory environment used for our work is one such web-based solution used for performing experiments in electrical, electronics and mechanical engineering. The system uses MVC architecture of Java Enterprise Edition for deployment [10] and [11] are examples of other similar works in the literature which uses a web-based framework for simulating experiments concerning electrical circuit and electromagnetics. Apart from the web, mobile-based access is becoming popular today. Web-based contents meant for wide-screen display in a computer monitor is not suitable for mobile display. In such cases, mobile content adaptation is required. [12] describes a potential solution for mobile content adaptation of e-learning websites.

Our work is based on the technique of facial expression recognition using a dual stream recurrent neural network based on deep learning. Works in the facial expression recognition literature [13] [14] uses only convolution neural network. Such a network cannot recognize the temporal motion of the faces captured across different frames of the Kinect recording. In our approach, convolution neural network is used to extract features in each individual frames, which are feed into a recurrent neural network. The use of recurrent neural network allows our system to detect temporal motion as recorded in the series of frames, thereby providing better results. We have also used more fine-grained labels in our test dataset for better training performance.

III. Construction of a Student Feedback Prediction Database

In this section, we describe the process of building a student feedback prediction database in an online laboratory training environment. We build this database because it allows us to exploit a data-driven approach for automatically investigating whether a student finds the module he/she being taught is useful or not. Compared to letting a student fill in a questionnaire form after the module is finished, such automatic feedback prediction mechanism is much more efficient and can produce much larger amounts of dynamic feedbacks for subsequent module assessment and improvement.

A. Participating Subjects

To build our database, we recruited 22 voluntary undergraduate and master students from Jadavpur University as our subjects. These subjects consist of 15 males and 7 females, age between 20 and 25 years old, and come from various study backgrounds (such as education, computer science, and electronic engineering). They were unknown about the background or purpose of our study prior to the experiment. As our experiment involves collecting visual data, we allowed them to wear glasses if needed and ensured that they all have normal visions. During the whole experiment, we kept them anonymous to avoid inducing any identity biases.

B. Learning Modules

We used the online laboratory training website developed in Jadavpur University as our experiment platform [6]. This website hosts a range of modules designed for instructing students to perform electronic engineering experiments online, so that they can improve learning efficiency offline when they only have limited access to physical equipments that are normally expensive and in short supply. We investigated 4 different kinds of learning modules in this study:

- **Theory**. This module introduces the prerequisite knowledge of electronic engineering experiments using texts, equations, and illustrative figures.
- **Video**. This module displays lecture videos recorded when professors and lecturers taught the theories and instructions of experiments in classes.
- **Animation**. This module shows the step-by-step manipulations of equipments using 3D animations, with narrative to facilitate understanding.

- **Simulation**. This module allows students to use a mouse to interact with experiment devices (e.g., wiring and switching on/off) according to taught instructions, and to read physically simulated metrics (e.g., currents and voltages) from virtual dash boards.

In this work, we built our database from the example of the 22 participating subjects learning the theory of a ceiling fan via the 4 learning modules. In the future, we will incorporate more examples to expand our database.

C. Experimental Setup and Protocol

We used a Microsoft Kinect (V1) [15] to capture the subjects' facial appearance and geometry while they were learning the theory of a ceiling fan via the Theory, Video, Animation, and Simulation modules. Our experimental setup consisted of a PC, a Kinect placed right below the PC monitor, and a subject sitting in front of the PC monitor. The Kinect was connected to and controlled by a Laptop, capturing a subject's upper body color and depth images in a fixed 30FPS rate, with a fixed resolution of 640×480. For each subject and each module in turn, we instructed him/her to do the following:

1) Sitting down and spending 1 minute to find out where the module is in the website and how to open it to start learning.
2) Learning the designated knowledge via the module for about 2 minutes, while the Kinect was capturing his/her upper body color and depth images.
3) When finished, he/she was asked to rate the quality of the module from 0-9, with 0 being the poorest and 9 being the best.

We captured 2 minutes because the Kinect produced 30 color and depth images per second and this already gave us a large amount of data (over 300GB in size). In the future, we will consider longer learning durations and finer-grained student feedbacks (e.g., on a subsequence level) such that module analysis and feedback prediction can be improved further.

D. Data Post-Processing

As the captured raw data is of a large size and thus difficult to directly use for machine learning algorithms, we resized each raw color and depth image from the original resolution of 640×480 to a lower resolution of 160×120. We sampled a shorter color and depth stream of length 12 every second from the original 2 minutes streams, with the sampling intervals of the shorter streams being 0.5 seconds (i.e., 15 consecutive images from the original streams). Please see Fig. 1 for an example. This way, we created a database of 22 subjects, 4 learning modules, and over 10,000 short image and depth streams of resolution 160×120.

IV. A Dual-Stream RNN Approach for Student Feedback Prediction

In this section, we propose a dual-stream RNN approach that simultaneously learns features from a color and a depth stream for student feedback prediction. Because a single frame lacks facial motion information, we use a RNN to adaptively combine features from all frames of a color and a depth stream, such that more discriminative features can be learnt for our sequence classification task [16].

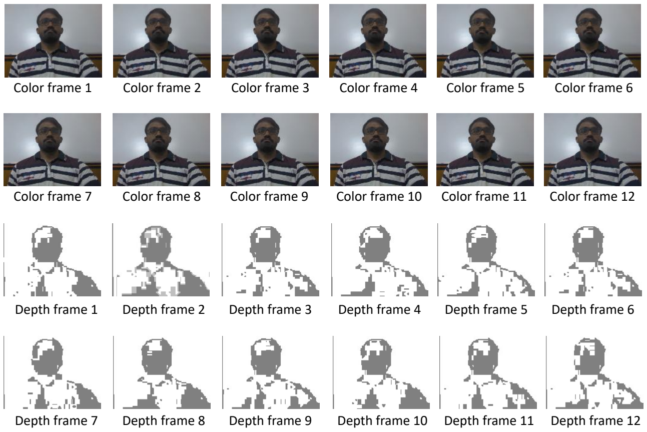

Fig. 1. **An example color and depth stream in our database**. For each stream there are 12 frames, two consecutive of which span 0.5 seconds in the original streams as captured by Kinect. The color frames capture the facial appearance of the shown subject and the depth frames capture his facial geometry information. The resolution of each frame is 160×120 as generated from the original resolution 640×480. The shown subject was using the Theory module while being captured by Kinect, and he gave a score of 7 to this module when finished.

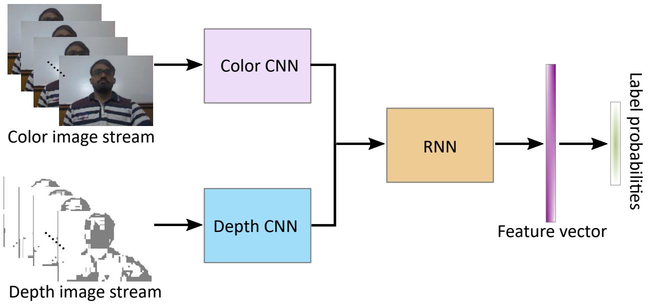

Fig. 2. **The overview of our approach**. The input to our approach is a color image stream and a depth image stream, and the output is a vector of 10 predicted probabilities corresponding to 10 student feedbacks. The feedback label with the highest probability is the predicted result. The color and depth CNNs share the same architecture but have separate parameters, which extract a feature vector from each input color and depth image separately and independently. The color and depth feature vectors of the same time step are concatenated and fed into a RNN in a step-by-step way. After all time steps have been processed, the internal cell state of the RNN that captures stream-level information is used for student feedback classification.

A. Method Overview

As shown in Fig. 2, the input to our approach is a color and a depth stream of length 12 with resolution 160×120, and the output is a label from 0 to 9 representing the predicted student feedback. Since our approach works on short color and depth streams, it can be efficiently applied to much longer learning durations via frame sampling, without increasing the storage or computation burdens. Therefore, it can be deployed to real laboratory environments with Kinect [15] for automatically providing interactive student feedbacks.

For each color and depth image in the input streams, we extract a color and a depth feature vector using a color and a depth CNN separately [17]. We use two separate CNNs because color and depth images have different modalities and thus require learning two different sets of feature extraction parameters. However, we do share the two CNNs among all frames in the input streams, as we want to extract motionindependent features at this stage and leave the task of extracting motion-dependent features to the next stage.

After CNN feature extraction, we concatenate the color and depth feature vectors of each frame and feed the resulting feature vector into a recurrent neural network. As the network maintains an internal cell state that is updated by each frame input [18], its state vector naturally summarizes the motion features of the input color and depth streams. We use this vector as our final learnt features, from which we predict a vector of 10 student feedback probabilities using a fully-connected classification layer. Given the ground-truth feedback labels, we can learn the color and depth CNNs, the RNN, and the classification layer all together [19].

B. CNN Architecture Design

Although the color and depth CNNs have separate parameters, their architectures are the same as illustrated in Fig. 3. In practice, we treat a depth image as a special type of color image whose red, green, and blue channels have the same grey (depth) values of corresponding pixels. Given a color/depth image of size $160 \times 120 \times 3$, after a series of convolutions, activations, and poolings, we obtain 64 feature maps of resolution 4×3 and finally reshape them into a single feature vector of size 768. We describe each building block in the following.

Convolution Block. For C input feature maps of resolution $W \times H$, we create C convolution kernels of small size 3×3 and slide each kernel across its corresponding feature map to compute a new feature map [20]. Each value on the new map is computed as the cross-correlation of the kernel parameters and the corresponding 3×3 pixel values from the input feature map. To preserve the spatial resolution, we pad the four boundaries of each feature map with one strip of zeros. From the C intermediate maps, we generate C' feature maps of the same spatial resolution using a single kernel of size 1×1. This kernel works on each pixel location, convolving C channel values into C' ones via a dense linear transformation. Essentially, we perform a 3×3 convolution on each input feature map separately and then use a 1×1 convolution to fuse different channels. This significantly reduces parameter sizes and accelerates training [21].

ReLU Function. This function applies element-wise to C feature maps of resolution $W \times H$, computing $ReLU(x) = x^+ = \max\{x, 0\}$ [17]. That is, a negative input element x is truncated to zero with a gradient zero, and a positive element passes freely with a constant gradient one. Compared to that of the traditional sigmoid and hyperbolic tangent (tanh) activation functions, this non-vanishing gradient property has been widely shown to be crucial for training deep CNNs and other neural networks [22], [23].

2×2 **Max-Pooling**. We use a max-pooling operator that takes the maximum element within each 2×2 small window (of stride 2) on the input feature maps after activation [20]. On one hand, the spatial resolution of feature maps can be halved from the original 160×120 to 80×60, 40×30, and 20×15. This greatly reduces the size of the final features learnt by a CNN and thus alleviates overfitting to a certain extent. On the other hand, because we only take the maximum element within a pooling window and do not care about where the element is, the pooled features have certain translational invariance that is beneficial for visual recognition tasks [24].

5×5 **Max-Pooling**. This block has the same function as that of the 2×2 max-pooling, except that its pooling window is 5×5 and thus larger and has a corresponding larger stride of 5. We use it to further down-size the feature maps from the resolution of 20×15 to that of 4×3. Because there are 64 feature maps of resolution 4×3, we finally obtain a feature vector of length $4 \times 3 \times 64 = 768$. As we have two CNNs corresponding to the color and depth streams separately, each frame in the streams is transformed to a feature vector of this length, resulting in $12 \times 2 = 24$ feature vectors.

C. RNN Architecture Design

We collectively denote the 12 feature vectors learnt from a color stream as $\boldsymbol{x}^c = \{\boldsymbol{x}_i^c \in \boldsymbol{R}^{768}\}_{i=1}^{12}$ and the 12 feature vectors learnt from a depth stream as $\boldsymbol{x}^d = \{\boldsymbol{x}_i^d \in \boldsymbol{R}^{768}\}_{i=1}^{12}$. As color and depth features reveal facial appearance and geometry respectively [15], we concatenate them together and obtain $\boldsymbol{x} = \{[\boldsymbol{x}_i^c, \boldsymbol{x}_i^d] \in \boldsymbol{R}^{1536}\}_{i=1}^{12}$. Our student feedback prediction task can be formulated as training a classifier that takes a sequence $\boldsymbol{x}$ as input and predicts a discrete label from 0 to 9 as output [16].

We may consider the multi-class logistic regression which linearly combines all features and generates a vector of 10 probabilities using the softmax function [25]. However, this approach would be unable to exploit the temporal features of facial motions, that can reveal subtle changes of facial expressions. We believe that such changes rather than absolute facial appearance or geometry are essential for telling whether a student likes or dislikes a learning module. For example, the change from a neutral expression to a happy one may indicate that he/she likes the module, while the reverse change may suggest that the module is becoming uninteresting to him/her. The logistic regression cannot differentiate the two opposite situations. Therefore, we consider the recurrent neural network approach to this problem [16].

Specifically, we learn a single feature vector of length 256 from $\boldsymbol{x}$ using the popular long short-term memory (LSTM) recurrent neural network [18], as shown in Fig. 4. The LSTM and its variants have been found to be very effective for sequence modelling and classification [26]. We describe its computation steps as follows:

$$\boldsymbol{f}_t = \underbrace{\sigma_s(W_{\boldsymbol{f}}\boldsymbol{x}_t + U_{\boldsymbol{f}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\boldsymbol{f}})}_{\text{forget gate activations}} \tag{1a}$$

$$\boldsymbol{i}_t = \underbrace{\sigma_s(W_{\boldsymbol{i}}\boldsymbol{x}_t + U_{\boldsymbol{i}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\boldsymbol{i}})}_{\text{input gate activations}} \tag{1b}$$

$$\boldsymbol{o}_t = \underbrace{\sigma_s(W_{\boldsymbol{o}}\boldsymbol{x}_t + U_{\boldsymbol{o}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\boldsymbol{o}})}_{\text{output gate activations}} \tag{1c}$$

$$\boldsymbol{c}_t = \underbrace{\boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \sigma_h(W_{\boldsymbol{c}}\boldsymbol{x}_t + U_{\boldsymbol{c}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\boldsymbol{c}})}_{\text{updated cell states}} \tag{1d}$$

$$\boldsymbol{h}_t = \underbrace{\boldsymbol{o}_t \circ \sigma_h(\boldsymbol{c}_t)}_{\text{output hidden states}} \tag{1e}$$

where σ_s and σ_h are the sigmoid and tanh activation functions respectively, the initial cell and hidden states $\boldsymbol{c}_0$

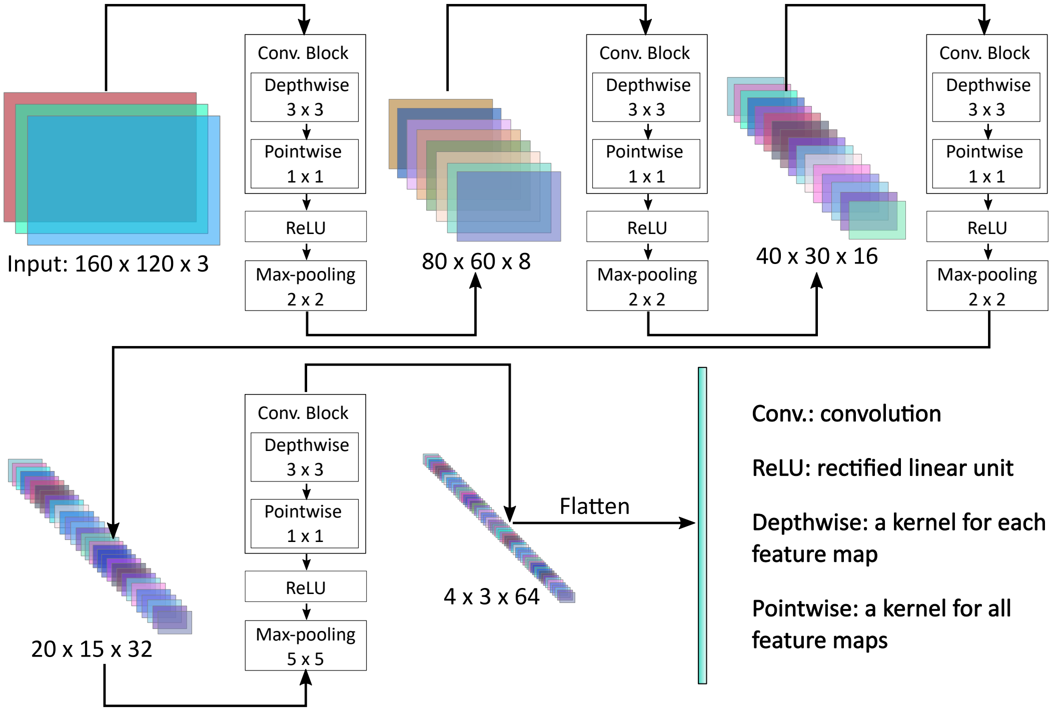

Fig. 3. **The architecture of the color/depth CNN**. The input is an image of resolution 160 × 120 with 3 channels (i.e., red, green, and blue), and the output is a feature vector of length 768. A depth image is treated as a special type of color image with its 3 channels the same as the depth values. For a depthwise convolution, we create a convolution kernel of small size 3 × 3 for each feature map separately, and pad the four boundaries of each feature map with a strip of zeros for maintaining the spatial resolution after convolution. For a pointwise convolution, we create a single kernel of size 1 × 1 that transforms the feature values of each pixel into a new set of values independently. We use three 2 × 2 max-pooling operators of stride 2 and a 5 × 5 max-pooling operator of stride 5 to downsize feature maps by 2 and 5 times respectively. We use the ReLU function for activation.

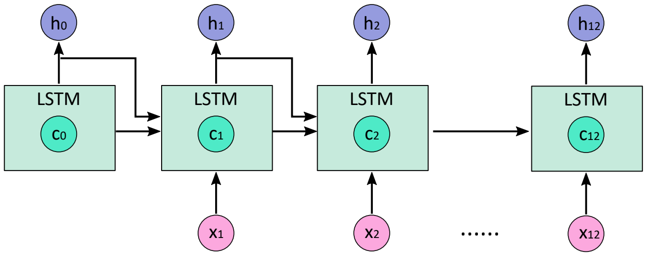

Fig. 4. **The unrolled architecture of the LSTM network**. The input is a sequence of 12 feature vectors of length $2 \times 768 = 1536$, as computed by concatenating the color and depth feature vectors of the same time step in a stream. The output is a sequence of hidden states that are fed back into the LSTM network in every time step. Inside the network there is an internal cell state that is updated according to the current input feature vector and the previous hidden and cell states. As the time unrolls, the cell state accumulates the facial motion information until the current time step. After the input sequence has been fully processed, the final cell state captures the motion features of the whole sequence. We use the final cell state for student feedback prediction by adding a fully-connected classification layer. Please refer to Section C for the internal mathematical mechanism of the LSTM network.

and $\boldsymbol{h}_0$ are all zeros, and the set of matrices and biases $\{W_{\boldsymbol{f}}, U_{\boldsymbol{f}}, \boldsymbol{b_f}, W_{\boldsymbol{i}}, U_{\boldsymbol{i}}, \boldsymbol{b_i}, W_{\boldsymbol{o}}, U_{\boldsymbol{o}}, \boldsymbol{b_o}, W_{\boldsymbol{c}}, U_{\boldsymbol{c}}, \boldsymbol{b_c}\}$ are the learnable parameters of the LSTM network.

It can be seen that the LSTM network's cell state $\boldsymbol{c}_t$ is updated by the current step input $\boldsymbol{x}_t$ and the previous step hidden state $\boldsymbol{h}_{t-1}$. Because $\boldsymbol{h}_{t-1}$ itself is dependent on all previous steps input $\{\boldsymbol{x}_i\}_{i=1}^{t-1}$, $\boldsymbol{c}_t$ naturally accumulates the motion features of the input stream until the time step t. We set the cell and hidden dimensionality to 256 and use $\boldsymbol{c}_{12}$, i.e., the cell state after the input stream has been fully processed, as the final learnt feature vector. We add a fully-connected classification layer to map $\boldsymbol{c}_{12}$ to a vector of 10 probabilities using the softmax function. The label that achieves the highest probability is the predicted student feedback.

D. *Optimization*

We use the stochastic gradient descent method to optimize the parameters of the color CNN, the depth CNN, and the LSTM network [19]. The optimization is done by minimizing the cross-entropy loss function [17]. We set the initial learning rate to be 0.1 and decay it by 10% every 2 epochs for a total number of 10 epochs. For each epoch, we sample a random training color and depth stream along with the ground-truth feedback label, until all streams in the training dataset have been accessed. To stabilize training, we use a momentum of 0.9 and set the weight decay value to be 0.0001.

V. Experimental Results

In this section, we present our experimental results to validate the effectiveness of our approach for student feedback prediction. For each experiment, we treat the short color and depth streams (with the ground-truth feedback scores) from a random subset of the 22 subjects as the training set and the remaining as the testing set. We train our approach end-to-end using the hyper-parameters in Section IV(D) and report the percentage of streams in the testing set that have the correct predicted student feedbacks.

TABLE I. TESTING ACCURACIES FOR DIFFERENT PARTITIONS OF THE TRAINING AND TESTING SUBJECTS IN OUR STUDENT FEEDBACK PREDICTION DATABASE.

Training/Testing subjects	9/13	11/11	13/9	15/7	17/5
Accuracy	58.34%	72.55%	79.28%	86.45%	87.21%

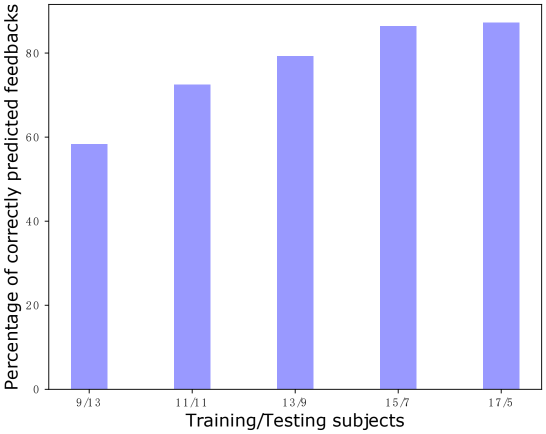

Fig. 5. The percentage of correctly predicted student feedbacks given different partitions of training and testing subjects in our database.

A. Classification Accuracy

To thoroughly evaluate the generalization performance of our approach, we randomly split the 22 subjects into a number of training/testing subjects partitions: 9/13 subjects, 11/11 subjects, 13/9 subjects, 15/7 subjects, and 17/5 subjects. We report the testing accuracy for each partition in Table I and Fig. 5. It can be seen that even with fewer training subjects than testing ones (i.e., 9/13), our approach achieves a feedback prediction accuracy significantly higher than that of random guessing, which is 10% since we have 10 labels to predict. It can also be seen that with more training subjects the prediction accuracy can be improved further. Under an 80% training/testing subjects partition, the accuracy can be as high as 87.21%, which may be usable in a realistic online laboratory training environment.

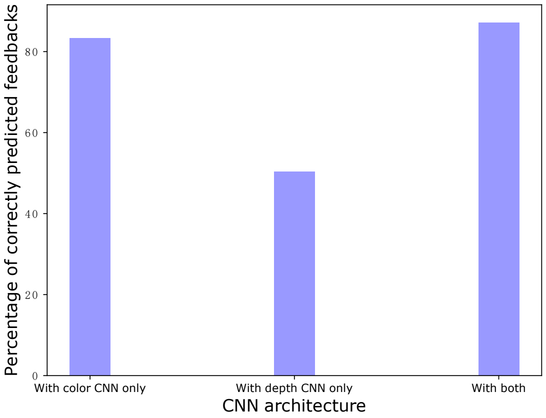

Fig. 6. The percentage of correctly predicted student feedbacks with the color CNN only, the depth CNN only, and both color and depth CNNs, respectively. The partition of training and testing subjects is 17/5.

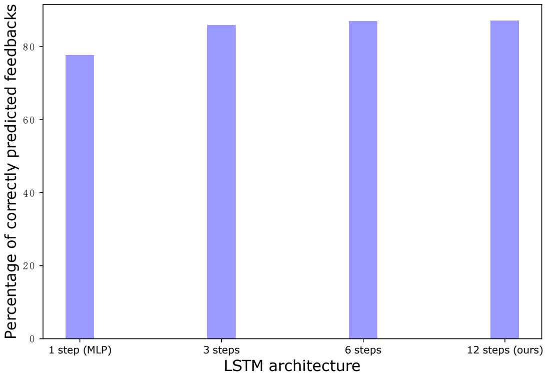

Fig. 7. The percentage of correctly predicted student feedbacks with the LSTM network of step size 1, 3, 6, and 12 respectively. The partition of training and testing subjects is 17/5.

B. Evaluation of the Color and Depth CNNs

We evaluate the effectiveness of using both color and depth information for student feedback prediction and show the result in Fig. 6. For this experiment we use the 17/5 training and testing subjects partition, as it gives us the highest prediction accuracy (Fig. 5). It can be seen that using the color CNN alone already gives us an 83.35% accuracy, as compared to the 50.32% accuracy achieved by using the depth CNN alone. This shows that facial appearance information is more predictive of student true feedbacks than facial geometry. However, adding the depth CNN can improve the accuracy further, which indicates that facial geometry complements facial appearance to a certain extent.

C. Evaluation of the LSTM Network

We also evaluate the effectiveness of learning facial motion features via the LSTM network for student feedback prediction. To do the evaluation, we consider shortening an original input stream of length 12 to the length of 6, 3, and 1 respectively, by concatenating the adjacent 2, 4, and all frames in the stream. This way, we maintain all input features while restricting the capacity (i.e., the total step size) of the LSTM network for learning temporal features. If the input stream is of length 1, the network is essentially reduced to a normal multi-layer perceptron (MLP). As shown in Fig. 7, the MLP architecture achieves the lowest feedback prediction accuracy, while adding more steps gradually improves the accuracy of the LSTM architecture. This is because more steps act as a temporal regularizer that forces the LSTM to exploit the hidden motion information within frames.

VI. CONCLUSION AND FUTURE WORK

We studied the problem of predicting whether a student likes a particular learning module or not in an online laboratory training environment. We used the Microsoft Kinect (V1) for capturing a student's facial appearance and geometry using color and depth images respectively, which are able to reveal subtle facial motions that are crucial for student feedback prediction. We recruited 22 undergraduate and master students of diverse backgrounds, incorporated 4 learning modules (including Theory, Video, Animation, and Simulation), and collected each student's responses to each of these modules for 2 minutes using the Kinect. After post-processing, we obtained over 10,000 short color and depth streams with the corresponding student ratings of modules from 0 to 9. Our database lays a foundation for data-driven student feedback prediction, module analysis and improvement.

For student feedback prediction, we built a dual-stream approach that learns features from a color and a depth stream using a color and a depth CNN respectively. To identify relative motion features rather than absolute ones, we combined the color and depth features of each time step and fed them into a LSTM network for recurrent feature embedding. We used the LSTM's final cell state for sequence-level feedback classification. Our experimental results showed that combining color and depth features are better than single ones, and that exploiting temporal features lead to better feedback classification performance.

Limitations and Future Work. This work has used only one particular experiment from electrical engineering category of TRAILS catalogue. The TRAILS e-learning module contains numerous experiments from electrical, electronics and mechanical engineering. In the future, an enlarged and diverse dataset could be created by incorporating numerous other experiments that TRAILS has to offer.

Due to the high frame rate of Kinect (30 FPS) and resulting large dataset size, we have been able to record each learning activity of 2 minutes duration. In the future, we would incorporate more realistic learning duration, roughly spanning 2 to 15 minutes. We would keep the dataset size manageable by exploiting online frame sampling method.

In our current work, we have used feedback scores from students for an entire learning duration. However, for our envisaged longer duration recordings, such scores would not be very accurate. A fine-grained score reflecting the feedback for each part of the learning module would be better suitable.

We explored facial appearance and facial geometry information for student feedback prediction. Another important cue of student learning activities is their eye movements and attention patterns on a screen [27]. Such extra information would help us identify whether a student is concentrated or not and which parts of the screen interest him/her more. We are interested in exploring the use of an eye-tracking device and the Kinect together in the future.

Acknowledgment

This work was supported in part by the Royal Society (Ref: IE160609), the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1), the Erasmus Mundus Action 2 Programme (Ref: 2014-0861/001-001), and the University Grants commission UPE Phase II Scheme from the Government of India.

References

[1] Y. Yang, H. Leung, H. P. H. Shum, J. Li, L. Zeng, N. Aslam, and Z. Pan, "Ccesk: A chinese character educational system based on kinect," *IEEE Transactions on Learning Technologies*, 2018.

[2] P. Plantard, H. P. H. Shum, A.-S. L. Pierres, and F. Multon, "Validation of an ergonomic assessment method using kinect data in real workplace conditions," *Applied Ergonomics*, vol. 65, no. Supplement C, pp. 562–569, 2017.

[3] P. Plantard, H. P. H. Shum, and F. Multon, "Filtered pose graph for efficient kinect pose reconstruction," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4291–4312, 2017.

[4] E. S. L. Ho, J. C. P. Chan, D. C. K. Chan, H. P. H. Shum, Y.-m. Cheung, and P. C. Yuen, "Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments," *Computer Vision and Image Understanding*, vol. 148, pp. 97–110, 2016.

[5] M. Budhu, "Virtual laboratories for engineering education," in *International Conference on Engineering Education*, 2002, pp. 12–18.

[6] "Trails teaching resources and interactive laboratory simulations," http://trails.jdvu.ac.in/290617, accessed: 2018-08-16.

[7] S. Moraitis, B. Mason, A. Pezouvanis, and M. Ebrahimi, "A practical, simulation based approach to the teaching of engine mapping and calibration fundamentals," SAE Technical Paper, Tech. Rep., 2011.

[8] B. Rajaonah, J. Sarraipa, M. Carnevale, M. Lebbar, M. Mestiri, C. Faure, and M. Abed, "E-learning training in railway engineering," in *IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2018, pp. 000 067–000 072.

[9] R. D. Burke, N. De Jonge, C. Avola, and B. Forte, "A virtual engine laboratory for teaching powertrain engineering," *Computer Applications in Engineering Education*, vol. 25, no. 6, pp. 948–960, 2017.

[10] I. A. Diaz-Diaz and I. Cervantes, "Development and implementation of an e-learning system for electric circuits laboratory," in *IEEE 7th International Conference on e-Learning in Industrial Electronics (ICELIE)*, 2013, pp. 28–32.

[11] V. Pulijala, A. R. Akula, and A. Syed, "A web-based virtual laboratory for electromagnetic theory," in *IEEE 15th International Conference on Technology for Education (T4E)*, 2013, pp. 13–18.

[12] S. Coondu, S. Chattopadhyay, M. Chattopadhyay, and S. R. Chowdhury, "Mobile-enabled content adaptation system for e-learning websites using segmentation algorithm," in *IEEE 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2014, pp. 1–8.

[13] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 435–442.

[14] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2983–2991.

[15] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[16] A. Graves, "Supervised sequence labelling," in *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, 2010, pp. 177–186.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[24] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 111–118.

[25] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[26] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[27] M.-L. Lai, M.-J. Tsai, F.-Y. Yang, C.-Y. Hsu, T.-C. Liu, S. W.-Y. Lee, M.-H. Lee, G.-L. Chiou, J.-C. Liang, and C.-C. Tsai, "A review of using eye-tracking technology in exploring learning from 2000 to 2012," *Educational research review*, vol. 10, pp. 90–115, 2013.