Building Interpretable Descriptors for Student Posture Analysis in a Physical Classroom

Lujie Karen Chen
University of Maryland Baltimore County
Baltimore, MD
lujiec@umbc.edu

David Gerritsen
Carnegie Mellon University
Pittsburgh, PA
dgerrits@andrew.cmu.edu

ABSTRACT
This research presents a process for simplifying video labeling and feature generation when building classification systems from real classrooms. Using video from a single, wide-angle recording of a live classroom, we create a low-level feature set of posture primitives built on keypoints from OpenPose. We use that feature set to build a posture recognition model of "natural labels" built from a scripted posture video using the same classroom. This model provides automatic labels for the real classroom data. We then derive a set of interpretable descriptors to characterize student-specific posture pattern dynamics. We show that those descriptors are able to discriminate between subtle differences in learning activities in a real college classroom.

Keywords
classroom analytics, posture analysis, student activity recognition

1. INTRODUCTION
The field of research into classroom sensing technologies and data mining is growing. One goal of this work is to provide automated feedback to instructors about anything from latent states of the students to overt actions by the teacher [15]. The motivation for this work is usually to empower teachers and scaffold instructional development without always relying on human consultants [13].

The promise of this field is high, but so are the costs. Technical staff and software development are all expensive. Additionally, labeling video data in order to derive insights about student interactions is particularly time-consuming and difficult. This study describes an attempt to reduce that cost. We used a freely available posture analysis tool (OpenPose) to produce keypoint data for human postures which we then used to build a generic set of labels for a class of students. Our goal was to simplify both the application and interpretability of data labels.

2. RELATED WORK
Emerging technologies for sensing pedagogical events in live classrooms include the detection of overt student behaviors (e.g., hand raising and gaze direction [1]), latent states (attention and engagement [16, 12]), and instructor actions (e.g., questions, activity sequences, gestures, and physical location in the room [3, 7, 11, 14]). Each approach has its own trade-offs in terms of reliability and effort required, but the models all require a dictionary of human-labeled body postures. Data annotation is time-consuming work requiring special expertise. For example, one must choose between coding in real-time [9, 10] or post-hoc [12, 16], and whether or not to use assisted label production [17].

Feature generation is a related but different concern. Education researchers may want to build models on comprehensible features, such as the words used during teachers' questions [3, 14], or the gestures students and teachers exhibit during interactions [5, 4, 7, 12, 16]. While it is possible to use a "kitchen sink" approach to quickly assess the success of an algorithm and its inputs, education researchers may prefer to use features that can be observed and understood by the end-user. This way the instructors using their systems might be able to make changes based on the model output, e.g., [2].

3. MOTIVATION
High-quality video cameras are ubiquitous. Researchers in education and machine learning can quickly generate large volumes of dynamic, rich data from the classroom. When turning video into data, education researchers traditionally code classroom videos using any number of methodologies [8]. Each approach for annotating and interpreting video data takes a significant amount of training and time.

To this end we present the following case study in which we demonstrate a pipeline that requires only minimal resource investment on the part of the experimenter, including the time it would normally take to define, identify, and verify student gestures. We propose that this savings is possible without sacrificing the interpretability of a human-coded feature set. To test the pipeline, we designed an easy and accessible feature generation strategy which we then tested against the most difficult in-class dataset we could imagine.

Figure 1 illustrates our workflow, broken into the following stages: (1) We collect a video recording (scripted posture data, section 4.2) with synchronized, scripted posture pat-

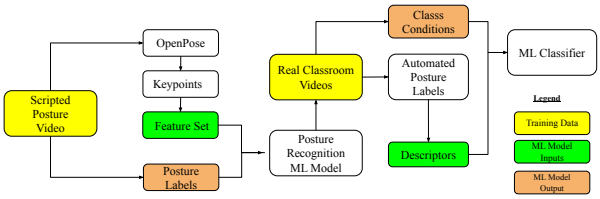

Figure 1: Workflow

terns commonly observed in classrooms; (2) We fit supervised machine learning models to automatically recognize the scripted posture patterns (section 5.1); (3) Using video from a real classroom (section 4.1), we demonstrate the utility of those auto-estimated posture patterns by applying the posture detection models to the real classroom dataset to discriminate between class conditions (section 5.2).

4. DATA SOURCES & AUTO-LABELING

Here we describe our data collection and analysis. We conducted the study at Carnegie Mellon University in the Spring semester of 2019. We generated model data from a group of volunteers, and target data from a class of real students. All students signed IRB-approved consent forms.

4.1 Real Classroom Data

In selecting a use-case for our approach we chose a class that embodied a traditional lecture-based class. We worked with a semester-long graduate level course on "Applied Data Science" (ADS) in Spring 2019. There were 22 enrolled students, all of whom could fit in a single frame of a wide-angle camera (Marshall CV505). The camera faced the rows of students, and the instructor was not in frame. We recorded throughout the entire semester of bi-weekly, 75-minute session. We collected 22 sessions for a total of about 30 hours of class time.

The format of the class was almost completely dominated by professor lecture. Halfway through the semester the students were put into groups for their final projects. During that second half of the semester, student groups took turns giving short presentations throughout the second session of each week. We used this naturally occurring difference in class format to inspire our classification problem. We thus generated two main class conditions: those led by the professor (i.e. *Professor-Lead*), and those led by groups of students giving project presentations (i.e. *Peer-Lead*). In the *Professor-Lead* condition (16 sessions), students listened to the professor lecture and were permitted but never required to ask questions. In the *Peer-Lead* condition (6 sessions), students listened to groups of peers take turns giving a short presentation describing their progress on an ongoing class project. After each presentation, all students were allowed to ask questions, and a random selection of students were required to ask questions for participation points.

Our goal was to model generic student posture patterns as descriptors to discriminate between *Professor-Lead* and *Peer-Lead*. From a naive perspective, the postures of the students in each condition were virtually indistinguishable. We chose this objectively difficult classification problem in order

Figure 2: A snapshot of scripted posture video in which volunteers were performing one of the scripted action of *Checking Phone*

to stress-test our approach. Our proposition was that students in either condition might have different internal states related to their expectation of learning useful information (*Professor-Lead*) vs. their potential requirement to ask a question (*Peer-Lead*), but that we would not have predictions about which gestures might reveal those latent states. This is a "good enough" test of our goal of building a practical process that could eventually be of some potential use to researchers who are likely to test less fuzzy classification problems.

4.2 Generic Student Posture Descriptors

To address our goal of helping researchers create descriptors without deep, costly annotation, we designed an approach that would create a catalogue of possible postures students exhibit in a typical class. We began by creating a 7-minute video of 11 volunteers arranged in the same seats as the students from the ADS class. Using the same equipment as would be used in the real class, we led the volunteers through a series of scripted movements. The "Scripted Posture Video" section of our workflow (Figures 1 and 2) comprises these data. The volunteers did not know what the prompts would be in advance, and their behaviors appeared natural. We guided them through 13 generic posture patterns: *Checking Phone*, *Looking at Computer*, *Looking Down*, *Looking Up*, *Looking at front-left*, *Looking at front-right*, *Looking Left*, *Looking Right*, *Performing Q&A*, *Talking to neighbor*, *Raising hand* (left and right) and *Writing*. We chose this list as a comprehensive representation of observable posture patterns in real classrooms when students listen to lectures. There are additional gestures we could have included, such as sleeping, eating, or drinking. However, this seemed outside of the scope of training on only the most frequent and probable behaviors rather than trying to include every conceivable movement that might exist.

Our next step was to produce an underlying set of low-level features for defining these generic posture patterns. Table 1 is a partial list of the 24 frame-by-frame features we created from OpenPose keypoints [6]. OpenPose[1] is a freely available toolkit for identifying physical landmarks, or "keypoints," on human figures in a picture, as shown in Figure 3 (left). Each keypoint is part of a 2-D array of real-valued numbers (Figure 3, right plot). In this analysis we only use upper body keypoints, including head, neck, and arms.

[1]https://github.com/CMU-Perceptual-Computing-Lab/openpose

Feature Name	Description
neck_nose	neck nose distance
Lshoulder_nose	left shoulder nose distance
Rshoulder_nose	right shoulder nose distance
rHand_nose	right hand nose distance
lHand_nose	left hand nose distance
nose_neck_h	nose neck horizontal displacement
nose_neck_v	nose neck vertical displacement
nose_shoulder angle	nose neck angle w.r.t shoulder
rElbow_angle	right elbow angle
lElbow_angle	left elbow angle

Table 1: A partial list of low-level features used in the posture recognition machine learning model

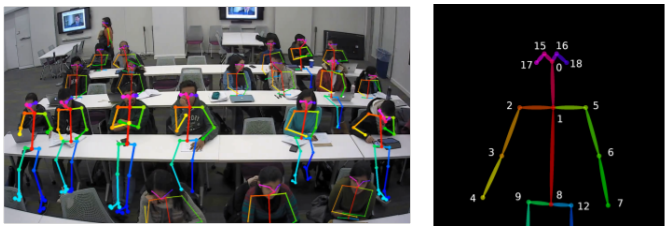

Figure 3: An example of OpenPose toolkit keypoints for a given frame of real classroom data (left); and the upper body keypoints used in our analysis (right)

We designed the features from Table 1 based on our professional experience performing student observation and an analysis of how groups of keypoints move together to produce gross postures. For example, the features *rHand_nose* and *lHand_nose* each measure the vertical distances between the nose and the hand. These distances can indicate vertical hand movements, e.g., as seen in hand-raising. "Nose-neck" related features (e.g. *nose_neck_h* or *nose_neck_v*) can indicate left/right head movements. With these low-level features in hand we then applied them to the scripted posture data and train a random forest classifier for recognizing posture patterns. We compiled a training set with each data point representing a person-frame pair and labeled each data point with labels naturally available from scripted posture data. We then fit several independent binary classifiers, each predicting the binary label of whether a given posture pattern occurred.

Our hope was that by having 11 different people perform the scripted movements in their own unique fashion, the model would be exposed to a sufficient amount of variability—such as one would expect to see in a the real world. We worked from the assumption that this would at least reduce the need for building and applying a precise annotation manual. This allowed us to quickly compile a posture-recognition model.

5. RESULTS

In this section, we present the frame-by-frame posture recognition model (section 5.1) using our generic behavior labels from the scripted posture dataset (4.2). We then applied that model to the ADS dataset (4.1), automatically labeling the posture patterns frame-by-frame. We derive descriptors from each 5-min segment of classroom video based on those machine labels, and built a classifier to discriminate between the two class conditions. The *Peer-Lead* segments of video did *not* include periods of question-asking after student presentations. This was meant to maximize surface similarity between the two conditions and provide a challenging test.

5.1 Posture Patterns Recognition

Table 2 summarizes the Area Under Curves scores (AUCs) for binary classifiers predicting whether a given posture pattern occurred in the appropriate position. An AUC rating of 0.50 is equivalent to chance, which means that *check-phone*, for example, does not have a reliable posture pattern as a composition of its keypoint structures. However, the model is able to identify head movement in left, right and up directions somewhat more reliably than other types of subtle head movements, such as *look-down*, *look-front-left* or *look-front-right*. Hand-raising postures and *writing* are also found to be relatively easier to identify. Similar to *check-phone*, actions without clear movement patterns exhibit low performance, i.e., *look-at-computer*, *Q&A*, and *talking-to-partner*.

Posture Patterns	AUC Scores	Posture Patterns	AUC Scores
check-phone	0.51	look-up	0.80
look-at-computer	0.63	look-left	0.84
look-down	0.53	look-right	0.88
look-front-left	0.67	writing	0.81
look-front-right	0.75	raise-left-hand	0.81
Q and A	0.63	raise-right-hand	0.84
talk-to-partner	0.60		

Table 2: Area Under Curve (AUC) scores from binary classifiers each predicting whether or not a given posture pattern has occurred, from leave-one-person out cross-validation experiment.

5.2 Discriminating Class Conditions

In this section we report the results from testing the hypothesis that there are discernible differences in students' posture patterns between the *Professor-Lead* and *Peer-Lead* class conditions. To answer this question, we formulated a machine learning classification task in which we used the descriptors derived from videos of the ADS class as input (right part of Figure 1). For output labels we used the class conditions. In creating the training dataset, we extracted a series of non-overlapping 5-minute segments from the real classroom videos and computed a list of statistics based on predicted student-by-student, frame-by-frame probabilities of posture patterns from the generic behavior model described in section 4.2. For each 5-minute video segment we derived five statistical values (*mean*, *standard deviation*, *min*,*max*, and *median*) summarizing the predicted probabilities of each of the 13 posture patterns. As a result, we have a training dataset with 65 features (13 posture patterns by 5 statistics) with each row representing a student-segment pair. We use random forest to fit the model. For comparison, we also derived an independent set of low-level features using only the keypoint structures described in Table 1.

We conducted two types of cross-validation experiments: *random split* and *leave-one-session-out*. In *random split* mode, the training and test datasets were constructed by random selection from the pool of 5-minute video segments,

irrespective of the class sessions to which they belonged. This design can yield relatively optimistic performance because of the likelihood that the segments from the same session can appear both in training and testing. In the second experiment, the split is based on class session, which results in a more conservative measurement of discrimination.

Table 3 shows the AUCs for model discrimination between *Professor Lead* vs *Peer Lead* using two different sets of input variables and under two different experimental conditions. The AUCs for each experimental design is beyond a random chance of 0.5. Specifically, we note that AUC decreases when using model-based descriptors compared to using a low-level, less interpretable feature set derived directly from the key points. The drop of AUC from *random split* to *leave-one-session-out* cross-validation suggests that certain predictive features are session specific and therefore make it difficult to predict labels for an unseen session. Future work will be of interest to identify those session specific features and investigate their roles in predicting class conditions.

Input Variables	Random Split	Leave-One Session Out
Posture-Model-based descriptors	0.72	0.64
Keypoint-based features	0.82	0.68

Table 3: Area Under Curve (AUC) scores for experiments to discriminate between *Professor-Lead* and *Peer-Lead* class conditions, comparing *random split* and *leave-one-session-out* cross-validation designs.

In order to understand the features that contribute to discriminating between the class conditions, we reviewed the feature importance from the random forest model that used interpretable posture-model-based descriptors as well as the low-level keypoint features. Figure 4 shows a selection of important and unimportant features from each approach. As noted in the upper portion of Figure 4, the most important input variables in the posture-based model are those describing the variation of left and right head movements. Other posture patterns, such as looking to the front, raising hands, and looking down did not play an important role in the model. The bottom portion of the figure shows that some of the important features were the distances between students' eyes and the angles between their nose and shoulders. Some of the less important keypoint structures included the relative angles of the left shoulder and elbow, as well as the distance between the left shoulder and the nose.

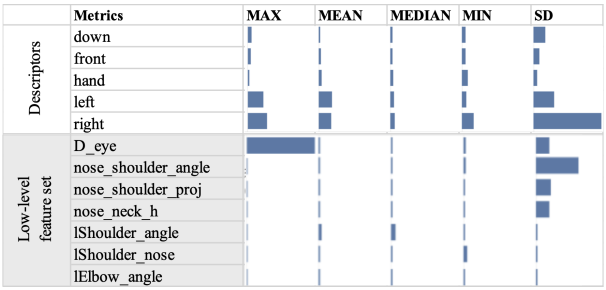

Figure 4: A selection of posture-based descriptors (white) and low-level features (gray) and their importance in a Random Forest model for discriminating between class conditions.

6. DISCUSSION

In this project we explored methods for extracting posture-related descriptors from videos of students in a real classroom. We derived the posture labeling model from a video of volunteers following scripted prompts. We extracted keypoint data from the videos using OpenPose, a freely available general purpose posture keypoints detection tool. We then showed that this method of automatic labeling could distinguish between two highly similar class conditions.

Large body movements such as hand raising and left/right head shifting were the easiest for the model to detect, and the most important descriptors in the posture model. In terms of using labels that are easy to interpret, these types of movements seem like a promising start. Without trying to interpret those movements at this time, they were at least important to the posture model. It maybe the case that simply informing an instructor about these movements could be a productive starting point for reflection.

Given that there were a number of features that did not contribute to the models, and that the raw keypoint model performed better than the derived model, we note that there is a trade off between the accuracy of this approach on the one hand, and its interpretability and transferability on the other. When we look at the variance in Figure 4, we see indications that the importance of some features (and the *lack of importance* of others) is more interpretable than the power of different keypoint angles and vectors. These higher level features say something about what students do differently in different scenarios. Our point here is not to deduce what those meanings are, but to show some of the student behaviors that are worth noticing. In terms of transferability, the fact that we built these labels from a 7-minute session of non-student volunteers shows that this approach may have some potential as a one-to-many label generation method, at least when the volunteers use the same classroom as the target students.

Finally, we propose that the pipeline we explored in this project, from feature generation to auto-labeling and from data prepossessing to feature extraction, can all be generalized to other teaching and learning scenarios in physical classrooms. For researchers in this space, i.e., developing classroom-based technologies for sensing behavior and providing automated feedback, our study may help simplify and accelerate their work by simplifying annotation and anticipating features that the end-user can understand.

7. ACKNOWLEDGMENTS

The research reported here was supported, in whole or in part, by the Institute of Education Sciences, U.S. Department of Education, through grant R305B150008 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

8. REFERENCES

[1] K. Ahuja, Y. Agarwal, D. Kim, F. Xhakaj, V. Varga, A. Xie, S. Zhang, J. E. Townsend, C. Harrison, and A. Ogan. EduSense: Practical Classroom Sensing at Scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–26, 2019.

[2] R. S. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. *JEDM| Journal of Educational Data Mining*, 1(1):3–17, 2009.

[3] N. Blanchard, P. Donnelly, A. M. Olney, S. Borhan, B. Ward, X. Sun, S. Kelly, M. Nystrand, and S. K. D'Mello. Identifying Teacher Questions Using Automatic Speech Recognition in Classrooms. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 191–201, 2016.

[4] P. Blikstein and M. Worsley. Multimodal Learning Analytics and Education Data Mining: Using Computational Technologies to Measure Complex Learning Tasks. *Journal of Learning Analytics*, 3(2):220–238, 2016.

[5] N. Bosch, S. K. D'Mello, J. Ocumpaugh, R. S. Baker, and V. Shute. Using Video to Automatically Detect Learner Affect in Computer-Enabled Classrooms. *ACM Transactions on Interactive Intelligent Systems*, 6(2):17, 2016.

[6] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[7] J. H. Correa, D. Farsani, and R. Araya. An application of machine learning and image processing to automatically detect teachers' gestures. In *International Conference on Computational Collective Intelligence*, pages 516–528. Springer, 2020.

[8] J. T. DeCuir-Gunby, P. L. Marshall, and A. W. McCulloch. Using mixed methods to analyze video data: A mathematics teacher professional development example. *Journal of mixed methods research*, 6(3):199–216, 2012.

[9] J. M. Girard. Carma: Software for continuous affect rating and media annotation. *Journal of Open Research Software*, 2(1), 2014.

[10] P. Goldberg, Ö. Sümer, K. Stürmer, W. Wagner, R. Göllner, P. Gerjets, E. Kasneci, and U. Trautwein. Attentive or not? toward a machine learning approach to assessing students' visible engagement in classroom instruction. *Educational Psychology Review*, pages 1–23, 2019.

[11] R. Martinez-Maldonado. "I Spent More Time with that Team". In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 21–25, Tempe, AZ, mar 2019. ACM.

[12] B. Ngoc Anh, N. Tung Son, P. Truong Lam, P. Le Chi, N. Huu Tuan, N. Cong Dat, N. Huu Trung, M. Umar Aftab, T. Van Dinh, et al. A computer-vision based application for student behavior monitoring in classroom. *Applied Sciences*, 9(22):4729, 2019.

[13] A. Ogan. Reframing classroom sensing: Promise and peril. *Interactions*, 26(6):26–32, 2019.

[14] L. P. Prieto, K. Sharma, Kidzinski, M. J. Rodríguez-Triana, and P. Dillenbourg. Multimodal teaching analytics: Automated extraction of orchestration graphs from wearable sensor data. *Journal of Computer Assisted Learning*, 34(2):193–203, 2018.

[15] M. K. Saini and N. Goel. How smart are smart classrooms? A review of smart classroom technologies. *ACM Computing Surveys*, 52(6), 2019.

[16] J. Zaletelj. Estimation of students' attention in the classroom from kinect features. In *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, pages 220–224. IEEE, 2017.

[17] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu. A generic framework for video annotation via semi-supervised learning. *IEEE Transactions on Multimedia*, 14(4 PART 2):1206–1219, 2012.