Classify Classroom Activities using Ambient Sound

Junze Li
jl11390@nyu.edu

Baosen Luo
bl3243@nyu.edu

Yuhe Tan
yt2336@nyu.edu

Abstract

This report documents the approach we used to build Classroom Activity Sound Event Detection, a supervised model capable of automatically detecting the type and duration of different pedagogical activities that occur in the classroom based on audio signals, which aims to provide post-hoc feedback to the instructors. Along the way, we developed an expandable pipeline which enables users to preprocess audios, extract features, tune parameters and predict class room activities using any set of annotated audios. Currently, our model can detect lecturing, Q/A, and collaborative student work reasonably well.

1 Introduction

Current college STEM(science, technology, engineering, and mathematics) teaching in the United States is mainly lecture-based and is relatively ineffective in promoting learning. Previous research has shown that replacing the standard lecture format with more active teaching strategies will increase the retention rate of students in STEM and produce superior learning outcomes. So far, hundreds of millions of dollars have been invested by national and federal agencies to shift large number of STEM faculty to include active learning in their teaching [Owens et al., 2017].

However, the extent to which large numbers of STEM faculty are changing their teaching methods to include active learning is unclear. To provide feedback for the faculty and collect statistics for future research, we need a measurement tool that could systematically inventory the presence and frequency of active learning. Currently available classroom observation tools, like Teaching Dimensions Observation Protocol (TDOP) and Reformed Teaching Observation Protocol (RTOP), all require trained human observers and do not scale.

To meet this need, we developed CASED: Classroom Activity Sound Event Detection. CASED is a machine-learning-based algorithm that can quickly analyze a large number of audio-recorded class sessions, with minimal costs and without need for human observers, to measure the use of teaching strategies beyond traditional lecture. In this report, we first introduce our dataset COAS, then document our methodology of building and evaluating CASED, and finally discuss the results and conclusions. The details of our implementation is stored in https://github.com/jl11390/classroom_audio_classification.

2 Related Work

The field of Learning Analytics focuses on the collection and analysis of educational data to provide actionable feedback for instructors and/or students. There is a branch of Learning Analytics, called Multimodal Learning Analytics(MmLA), that has specialized in collecting and analyzing data from physical learning settings, mainly classrooms. One of the main characteristics of the MmLA is the use of different audio, video, and physiological signals captured through various sensorsOchoa and Worsley [2016].

The solution proposed by this project will be based on an automated ambient sound classifier to distinguish between different types of pedagogical activities such as lecturing, Q/A, group work, etc.

Preprint. Under review.

Table 1: COAS dataset labels distribution

Label	Description	Dist.
Lecturing	teacher talking alone	0.407
Q/A	teacher and one student going in turns	0.383
Teacher-led Conversation	teacher and more than one student going in turns	0.046
Student Presentation	student presenting to the rest of the class	0.055
Individual Student Work	students work in silence	0.029
Collaborative Student Work	students talking among themselves to solve a question or task	0.078

Previous studies on the use of audio to study classroom activities show that this approach is feasible. For example, James et al. [2019] used speaker diarization and non-verbal voice cues to successfully classify the affective climate of the kindergarten classrooms, Owens et al. [2017] and Donnelly et al. [2016] used their respective ambient sound classification method to detect pedagogical strategies of instructors.

3 Problem Definition

The main objective of this project is to build and evaluate a prototype system to automatically detect and register the type and duration of different pedagogical activities that occur in the classroom, based on course audio signal, and then provide post-hoc feedback to the instructors.

However, the categories of pedagogical activities and their corresponding characteristics may vary in different teaching scenarios. For example, a K12 classroom could be dramatically different from a college lecture. Hence, instead of only finding the best model for a given fixed data set, this project aims to deliver a training and evaluation pipeline that automatically searches for the optimal model given any training data set that follows a certain annotation criteria.

4 Experimental Evaluation

4.1 Data

To develop CASED, human annotations were used to train the machine-learning-based algorithm that reports what types of activities are going on in a classroom, based on sound waveforms. The COAS dataset is collected by Prof. Xavier Ochoa and contains 133 course videos with their corresponding human annotations. The vast majority of these course videos are public available classroom recordings. Total length of these course videos is 22 hours.

In annotations, there are six activity prediction categories: lecturing, Question and Answer(Q/A), teacher-led conversation, student presentation, individual student work, and collaborative student work. One of the main characteristics of COAS dataset is its class imbalance, a modeling challenge which we will address in the later sections. The labels distribution is shown in Table 1.

An example of the annotation for a specific video file X.mp4 is: *{"video_url": "/data/X.mp4", "id": 1, "tricks": [{"start": 0, "end": 12.6, "labels": ["Q&A"]}, {"start": 13.5, "end": 173.2, "labels": ["Lecturing"]}, {"start": 173.9, "end": 203.9, "labels": ["Collaborative Student Work"]}]}*.

4.2 Methodology

In this section, we outline the machine learning pipeline, from data preprocessing and feature extraction to model training and inference. Figure 1 visually illustrates the workflow.

4.2.1 Data Preprocessing

The first step is to extract audios from videos. We used the Moviepy editor package in Python to extract the audio signal, stored in waveform file, from the video. Then we randomly assigned 20% of audios into the test set and left them unseen during the entire training stage.

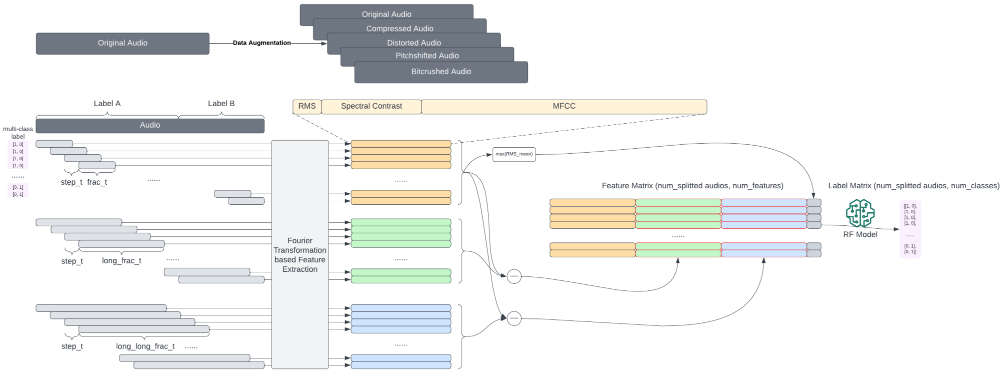


Figure 1: Training Pipeline

The annotated dataset for our project is relatively small and highly imbalanced. Although we tackled the data imbalance problem by assigning higher weights to the minority classes during training, the minority classes constantly under-performed compared to the majority classes. To deal with the data scarcity and improve model's robustness towards variations in the input audios, we augmented the training data by five times through audio modifications by using compressor, distortion, pitchshift and bitcrush effects from Pedalboard. Compressor reduces the volume of sounds over a loudness threshold, by 4 times if over 20 Db in our project and distortion applies harmonically pleasing distortion to the audio signals. We shifted the pitch of audios by 3 semitones and reduced the audio signals by 8 bit depth, giving the audio a low-fidelity, digitized sound. All effects do not change the length of audios so that we are allowed to refer to the original annotations.

Audio signal and human annotations live in continuous time domain with various time length, while general classifiers take in fixed input size and produce discrete output. To transcribe the annotated audios into training set, we segmented audios into fractions of equal size. By choosing a rolling window size w, i.e. the number of consecutive time periods per rolling window, and a step size s, i.e. the number of increments between successive rolling windows, we could segment an audio of length t into (t-w)//s + 1 fractions, with each fraction having a fixed size of w. To determine the labels for a fraction, we first took all the annotations for that fraction, then discarded any annotated activity that takes up less than 30% of the rolling window, and finally converted the remaining annotations to labels using the Multi-Label-Binary technique.

4.2.2 Feature Extraction

In the next step, we extracted features from each fraction using Fourier-Transform based technique McFee et al. [2015]. More specifically, we calculated the mean and standard deviation of the Root-Mean-Square(RMS) values, Mel Frequency Cepstral Coefficients(MFCC), and Spectral Contrast values. RMS represents the average, also called "integrated", loudness of a signal, while MFCC and Spectral Contrast yield representative figures for signals in different spectral bands. Mean values are good approximations of audio features and standard deviations unveil the degree of fluctuations. Since course audios have discrepant loudness levels, possibly due to different locations of the microphone in different classrooms, we added a global maximum RMS, which is derived from the complete audio rather than a single fraction, to 'normalize' the loudness levels.

To have a more intuitive understanding of these Fourier-Transform based features and how they combine to represent an audio signal, we visualize these features with heatmaps and line plots. Figure 2 shows a course audio and its corresponding feature visualizations.

As one of the main goals of our model is to distinguish single-participant activities such as lecturing from multiple-participants activities such as Q/A, multi-scale features are handy to get better audio representation and help detect switches in the source of sound. To obtain the multi-scale features for a fraction x, we first took a longer fraction y, which shares the same starting point as x, then extracted

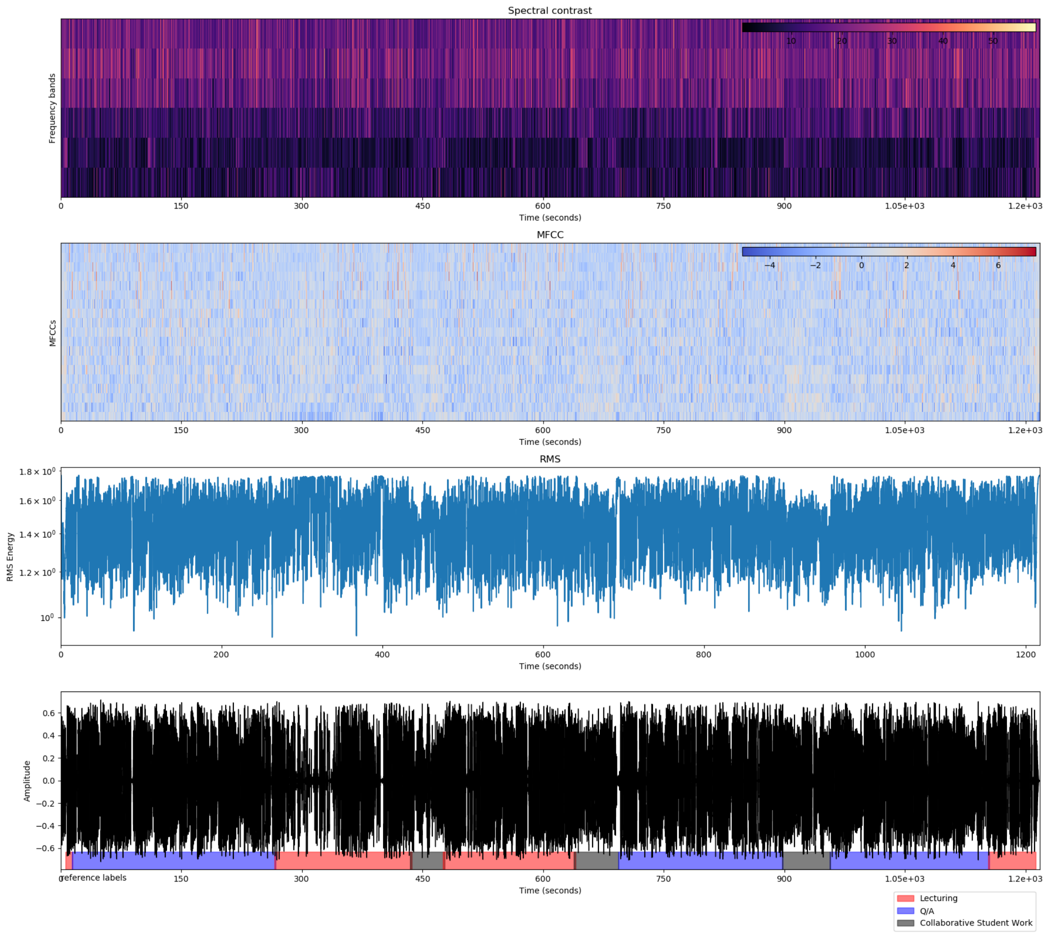


Figure 2: Raw Features

the features from y following the procedure described in the last paragraph, and finally calculated the difference between the two sets of features and appended the results into the feature space.

To examine the quality of the feature representation of the audio signal, we visualized these high dimensional features in two-dimensional space by using UMAPMcInnes et al. [2018] and a python package mlp-scatter-density. Figure 3 suggests that the feature representation captures certain characteristics of different classroom activity and is able to differentiate most of activities in two-dimensional space. Additional UMAP plots, which visualize MFCC, RMS, and Spectral Contrast features independently, are recorded in Appendix.

4.2.3 Model Training

We chose to train a random forest model due to its robustness towards collinearity among features, ability to assign sampling weights to imbalanced classes and versatility to output probabilities to multi-label output. All training audios are randomly separated into 5 folds. Then a grid search, combined with leave-one-group-out cross validation, is performed to obtain the best performing hyper parameters. More specifically, we targeted macro f1 score as our optimization metric, and searched for important hyper parameters related to the random forests, including number of estimators, max number of features used for splitting, max depth of the trees, and weights assigned to each class. After grid search, the final model was re-trained on all audios.

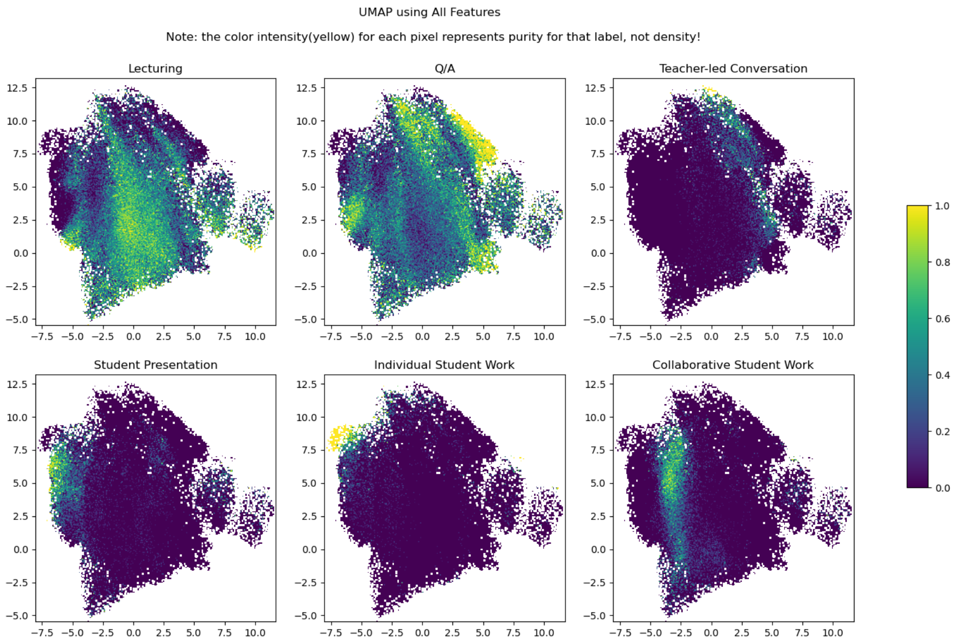


Figure 3: UMAP for Feature Representation

4.2.4 Inference

To make inference, segmentation with the same window length and step length would be performed on the observed audio. The random forest model will output class-wise probabilities for each segmented fraction. As we need deterministic predictions on each fraction, we applied the Viterbi algorithm to smooth the predictions for each class. More specifically, Viterbi algorithm transforms class-wise probabilities into binary predictions by computing the most likely sequence of states for each class with a two-by-two transition probability matrix. Each audio fraction in the sequence can only take 0 or 1 for each label class so that the probability of a state stays the same in the following fraction and the probability of current state changing to the other add up to 1. Hence, there are two transition probabilities we need to determine, namely from 0 to 1 and from 1 to 0. We also leveraged the marginal probability in Viterbi transformation to give minor classes higher probabilities of reaching positive predictions in sequences. Finally, we performed grid search with respect to the macro f1 score across all classes to find the best combination of transition probabilities and marginal probability for each class.

After having a sequence of binary predictions for each class in the audio fraction space, we converted the predictions back to the continuous time domain. The final annotation-like predictions for the observed audio are generated by concatenating continuous positive sub-sequences. The start time of the first positive fraction and the end time of the last positive fraction in a continuous positive sub-sequence determine a positive prediction for a class. Figure 4 is an example of our final prediction for a particular course audio.

As we are doing class-wise prediction, there exist overlapped predictions where our model prediction would annotate multiple labels over the same fractions of an audio. We found the prediction results reasonable since activities such as Q&A and lecturing are often times hard to distinguishable even for humans. Although predictions can be fragmented and overlapped, we observed several examples where model predictions are more sensitive than human annotations in detecting activities that only take place in a short amount of time. More details on inference results will be discussed in the later part.

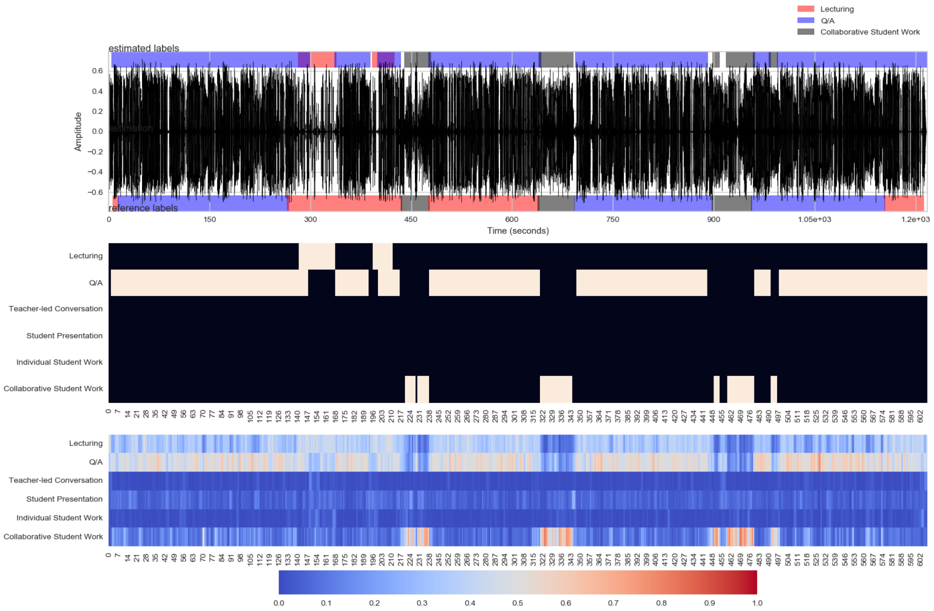


Figure 4: Inference Example

4.3 Results

The model performance was evaluated by using sed_eval, a Python toolbox Mesaros et al. [2016] designed specifically for sound event detection system. The task of sound event detection involves locating and classifying sounds in audio recordings, estimating onset and offset for distinct sound event instances, and providing a textual descriptor for each. Since our pipeline outputs prediction on each class for the segmented audios, for each audio file, we aggregated all the consecutive fractions of the same class, recorded the onset and offset, and marked it as a detected sound event.

Then, the ground truth annotation and the predicted annotation were compared in a fixed time grid segment, which we set as 1 second. Sound events are marked as active or inactive in each segment. We marked each segment as true positive, false positive, false negative, and true negative, and finally aggregated all the segment results for all files. After that, class based metrics were calculated, and we focused on precision, recall, F measure, and error rate.

4.3.1 Model Iteration

In order to acquire better prediction results, our pipeline went through several iterations as our project progressed, and the evaluation results for each iteration are shown in Table 2. The evaluation results are averaged across all classes.

4.3.2 Final Evaluation

The class based metrics are shown in Table 3. To get better insights of our prediction result, we also calculated class-wise multi label confusion matrices, as shown in Figure 5. Since we targeted on identifying more Q/A and preventing predicting everything as lecturing, we indeed got high precision for lecturing and high recall for Q/A. The model also does a great job in identifying collaborative student work, as precision and recall are relatively high and error rate is low. All the metrics for teacher-led conversation is 0, meaning that there is no positive example in the test data, and the model doesn't classify any example as teacher-led conversation. For student presentation, there are very few

Table 2: Stage-wise evaluation result

Stage	Methods	F Measure	Precision	Recall	Error Rate
0	original waves, original features, Random Forest Classifier (RFC), RFC predicted probabilities	0.40	0.44	0.38	0.75
1	original waves, multi-scale features, RFC, RFC predicted probabilities	0.39	0.50	0.34	0.58
2	original waves, multi-scale features, RFC, Viterbi smoothed prediction	0.42	0.47	0.40	0.72
3	original waves, multi-scale features, RFC, Viterbi smoothed prediction with marginal probabilities	0.43	0.43	0.45	0.56
4	data augmented waves, multi-scale features, RFC, Viterbi smoothed prediction with marginal probabilities	0.45	0.50	0.43	0.54

Table 3: Class based metrics

	F Measure	Precision	Recall	Error Rate
Lecturing	0.66	0.94	0.51	0.52
Q/A	0.82	0.72	0.94	0.42
Teacher-led Conversation	0	0	0	0
Student Presentation	0	0	0	1
Individual Student Work	0.37	0.52	0.29	0.98
Collaborative Student Work	0.86	0.84	0.88	0.28

positive examples in the test data, and the model fails to detect them. For individual student work, there are very few positive examples in the test data, and the model only detects some of them.

4.4 Discussion

Although the model performs well in classifying lecturing, Q/A, and collaborative student work, it does not have satisfying performance in classifying minority classes such as student presentation and individual student work. The possible cause could be the low quality of training data.

There are very few training data for student presentation and individual student work, so the model does not learn much information about these two classes during training. Secondly, after carefully examining the human annotations in training data, we found that there are some human errors that the true underlying classroom activity class is mistakenly annotated as other classes. This could make our training data not representative enough.

Another issues is that the run time of our model is not short enough for future usage in real-time. Currently, on average the model needs 11.36 seconds to process and classify a one minute audio. Although this project only serves as a baseline and technology demonstrator, it is worth considering run time efficiency, as for following stages of the broader project an enhanced classifier will be built and implemented on real-time pedagogical classroom activities.

5 Conclusions

The end product is a set of Python programs that enable users to train the model with new classroom audios, tune parameters, predict class room activities based on input audios, and evaluate prediction results. It is sufficient to serve as a prototype and baseline demonstrator of the classroom pedagogical activity classifier.

To further improve the classification result, in the future it could be helpful to collect data from simulated classes to enrich the model with pedagogical activities not available in public datasets. To improve the run time of the classifier, more work could be done on code optimization and using caching in librosa to store and re-use intermediate computation results.

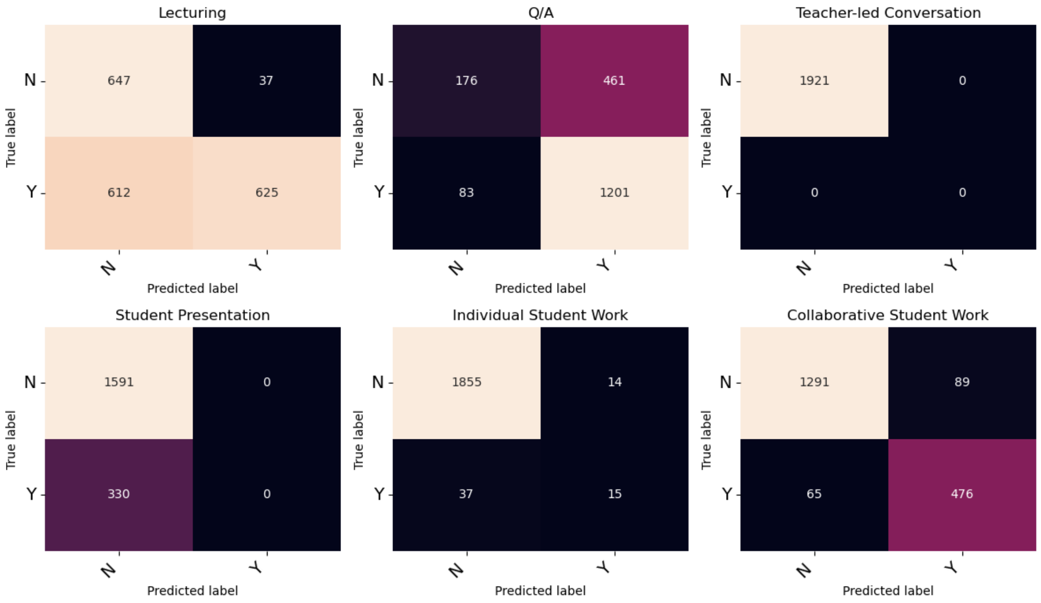


Figure 5: Confusion matrices

The potential impact of this project is improving teaching practices by supporting these practices with evidence-based reflection and decision-making with little additional effort for instructors in collecting and analyzing the data. This awareness and reflection will lead to better planning and execution of class activities, and consequently to more efficient and effective use of class time and the inclusion of more active learning strategies. Additionally, this project could serve as an example of how machine learning can be seamlessly integrated into pedagogical practice to augment the capabilities of both instructors and students.

6 Lessons learned

We iteratively developed CASED on a weekly basis. However, we did not calculate the evaluation metrics until our model has been in the fourth iteration, which forced us to roll back several versions to retrieve the model performances. Hence, it is super important to finalize how results shall be evaluated and keep track of the results of evaluation throughout the whole process of model development. Consistent evaluation methods help us understand how each component we added improve the performance and identify possible iteration directions along the way.

Code structure is crucial for the success of any sophisticated project. We have tried to make each functionality of our model as separable as possible so that we can expand and debug efficiently. However, when our model became more complex along the way, we gradually feel a lack of control over the code structure we previously designed. It would be hard to oversee all the possible iteration at the beginning of the project, but it would be helpful to always implement a code structure that takes highest freedom to expand.

Finally, It is always crucial to understand theories before implementation. As our project falls into the category of audio signal processing, there are tremendous amount of domain specific knowledge. Although most of signal extraction methods have been implemented by various packages online, we found it super helpful to learn how each method works in theory and in depth. We added local maximum RMS to our feature space because we understood that the RMS features we extracted are biased due to the discrepant locations of microphones in classroom.

References

Melinda T Owens, Shannon B Seidel, Mike Wong, Travis E Bejines, Susanne Lietz, Joseph R Perez, Shangheng Sit, Zahur-Saleh Subedar, Gigi N Acker, Susan F Akana, et al. Classroom sound can be used to classify teaching practices in college science courses. *Proceedings of the National Academy of Sciences*, 114(12):3085–3090, 2017.

Xavier Ochoa and Marcelo Worsley. Augmenting learning analytics with multimodal sensory data. *Journal of Learning Analytics*, 3(2):213–219, 2016.

Anusha James, Yi Han Victoria Chua, Tomasz Maszczyk, Ana Moreno Núñez, Rebecca Bull, Kerry Lee, and Justin Dauwels. Automated classification of classroom climate by audio analysis. In *9th International Workshop on Spoken Dialogue System Technology*, pages 41–49. Springer, 2019.

Patrick J Donnelly, Nathan Blanchard, Borhan Samei, Andrew M Olney, Xiaoyi Sun, Brooke Ward, Sean Kelly, Martin Nystran, and Sidney K D'Mello. Automatic teacher modeling from live classroom audio. In *Proceedings of the 2016 conference on user modeling adaptation and personalization*, pages 45–53, 2016.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

Student contributions

Our team worked closely on the project and we believe all members dedicated equal contribution to the outcome of this project.

A Appendix

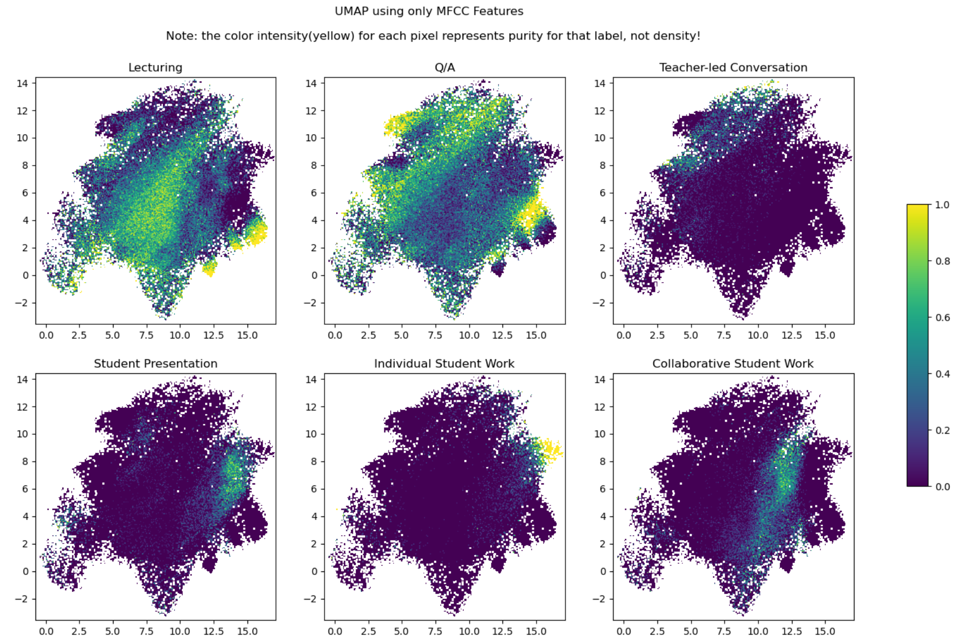


Figure 6: UMAP for Feature Representation

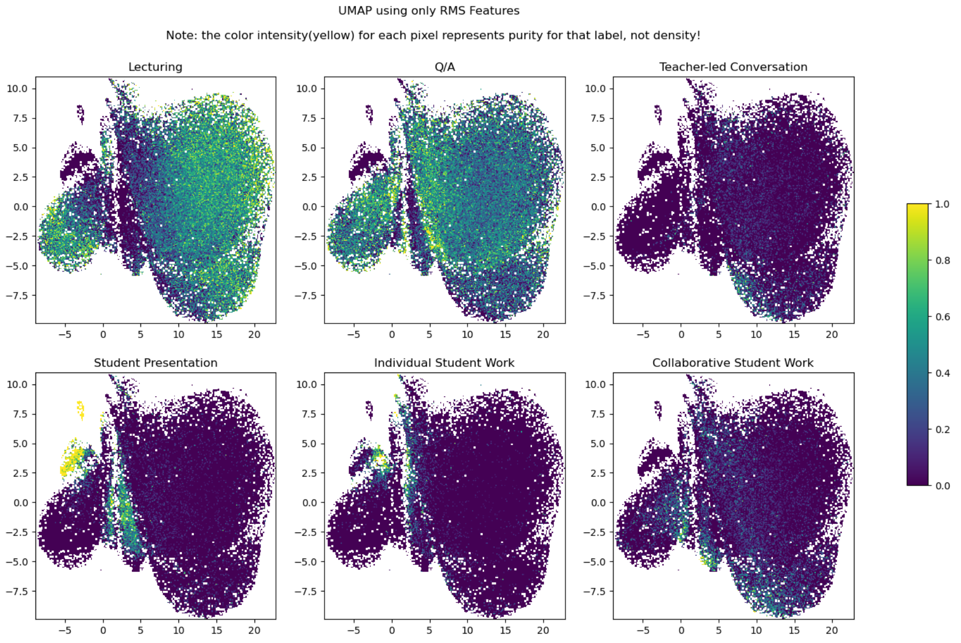
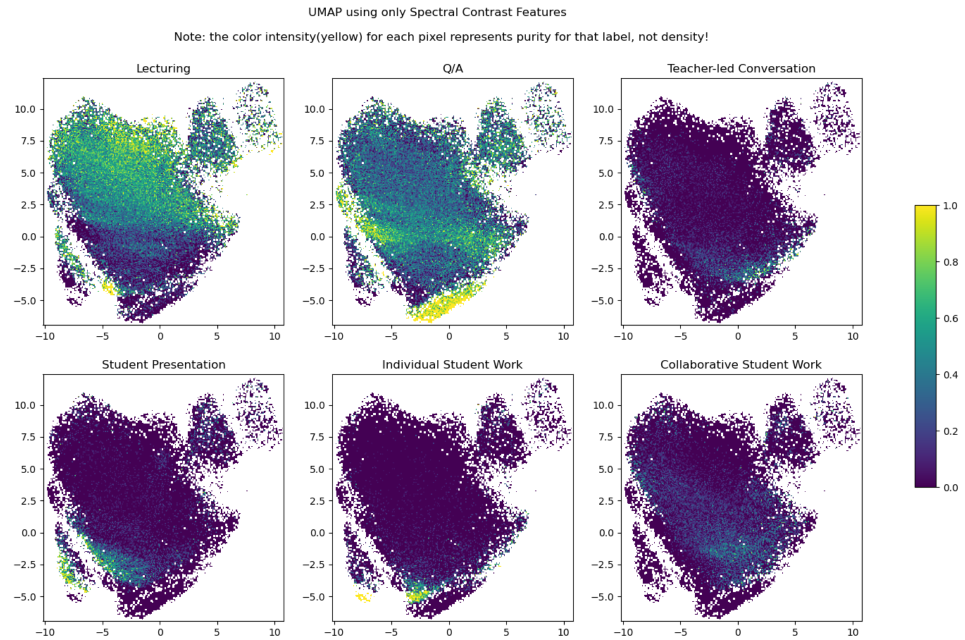


Figure 7: UMAP for Feature Representation



UMAP using only Spectral Contrast Features

Note: the color intensity(yellow) for each pixel represents purity for that label, not density!

Lecturing

Q/A

Teacher-led Conversation

Student Presentation

Individual Student Work

Collaborative Student Work

Figure 8: UMAP for Feature Representation