In the report, first you are required to describe the detailed steps and parameters in your MED pipeline with two types of features: MFCC and SoundNet. Secondly, please specify the size of your training and validation set and the evaluation metric you use (e.g., top-1 accuracy, top-5 accuracy, or average precision) for validation. Please report your model performance on the validation set. We ask you to also report the confusion matrix. Which class(es) is harder and with which it is confused? Lastly, we expect you to report the time your MED system takes (CPU time) for feature extraction and classification on the testing set. Please also tell us the amount of credits left on your AWS account after you finish homework 1.

My MED pipeline implementation is as follows:

1) Extract the audio part of the videos using ffmpeg
2) Extract the Mel-frequency cepstral coefficients (MFCCs) using openSMILE
3) To encode the Bag-of-Words representation, I trained the k-means clustering algorithm with the entire MFCCs feature. I tried with k = 200 to train 200 cluster centers, but with any k larger than 60, the sklearn package implementation exhausts the 8 GB memory that the t2.large EC2 instance which I was using. Hence, The highest number of clusters that I tried is 60, which also results in an improvement over the baseline MFCC models.
4) To extract the SoundNet features, I modified the provided extract_feat.py script to save the conv6, conv7, conv8, and the 18th layer of the SoundNet architecture as the feature. Because of the memory limitation of the EC2 instance, I broke the 7k+ audio files into batches of 1000 when running the script.
5) Then I trained the classification model based on SVM and MLP. I modified the training script for the MLP classifier to take an additional argument for the SoundNet feature vectors generated from the previous step so that I can concatenate the BoW MFCCs and SoundNet features during training and testing.
6) I trained a few different models with different inputs, the details can be seen in the following chart.

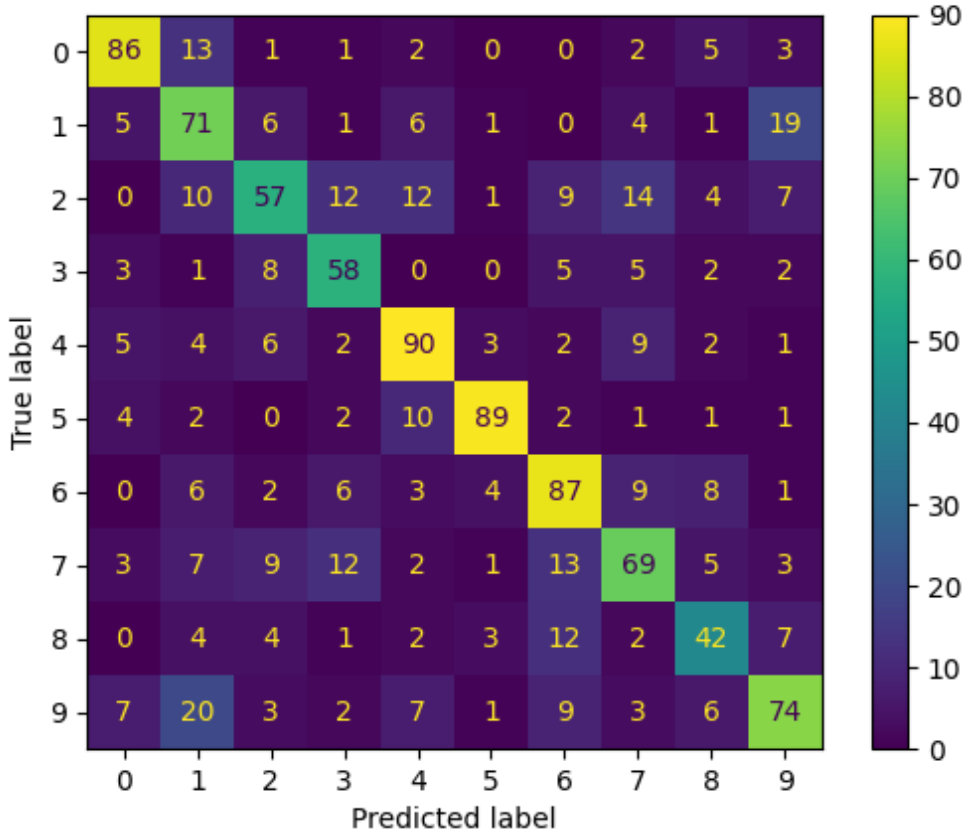
Model	Input	Training Accuracy	Validation Accuracy	Test Accuracy Private/Public
SVM (rbf kernel, one-versus-rest)	MFCC-50 ((50,) vectors)	81.5469%	31.5692%	0.38070/0.36228
MLP (hidden_layer = (100,), relu activation, adam solver, 5000 max iteration)	MFCC-50 ((50,) vectors)	83.8794%	42.9781%	0.44385/0.42719
Random Forest (max_depth=2)	MFCC-50 ((50,) vectors)	86.5846%	36.5498%	
Random Forest (max_depth=3)	MFCC-50 ((50,) vectors)	87.5846%	32.6897%	

SVM (rbf kernel, one-versus-rest)	SoundNet-Conv 8 (sum pooling (32,) vectors)	83.7846%	31.1692%	
MLP (hidden_layer = (100,100,100,), relu activation, adam solver, 5000 max iteration)	SoundNet-Conv 8 (sum pooling (32,) vectors)	82.1080%	11.8791%	
MLP (hidden_layer = (100,50,25,50,100,), relu activation, adam solver, 5000 max iteration	SoundNet-Conv 8 (sum pooling (32,) vectors)	81.5978%	36.4897%	
MLP (hidden_layer = (100,100,100,), relu activation, adam solver, 5000 max iteration)	SoundNet-Conv 18 (sum pooling (256,) vectors)	87.4872%	57.8564%	0.57368/0.57894
MLP (hidden_layer = (100,), relu activation, adam solver, 5000 max iteration)	SoundNet-Conv 18 (sum pooling (256,) vectors)	88.12938%	58.5120%	0.59912/0.58245
MLP (hidden_layer = (100,50,25,50,100,), relu activation, adam solver, 5000 max iteration	SoundNet-Conv 18 (sum pooling (256,) vectors)	87.9879%	54.1474%	0.57192/0.57807
MLP (hidden_layer = (100,50,25,50,100,), relu activation, adam solver, 5000	SoundNet-Conv 18 (sum pooling (256,) vectors) Concatenated with MFCC-60	89.5941%	59.7956%	0.55087/0.55964

max iteration	((316,) vectors)			
MLP (hidden_layer = (100,) relu activation, adam solver, 5000 max iteration	SoundNet-Conv 18 (avg pooling (256,) vectors)	92.4911%	64.7527%	0.58859/0.60614

7) In order to evaluate the performance of the various models, I experimented with two approaches. Since the Kaggle competition uses top-1 accuracy to evaluate the performance, I decided to use that to evaluate my models as well. First, for faster experimentation, I split the trainval set into an 80% training set and 20% validation set with shuffling. Then, I used 5-fold cross-validation to evaluate the mean top-1 accuracy and its standard deviation.

Results and Confusion Matrix:
The model with the highest top-1 accuracy that I trained is the MLP model with one hidden layer size of 100, ReLU activation, Adam solver, and 5000 max iterations. SVMs and Random forests generally do not perform as well as the MLP classifiers. The SoundNet feature extracted from average pooling the 18th layer outperforms both BoW and BoW+SoundNet features.



The confusion map shows that class 1 (mowing lawn) and class 9 (shoveling snow) are the most difficult classes to distinguish between. And class 2 (playing guitar), class 3 (playing piano), class 7 (singing), and class 8 (tickling) are also hard to distinguish from other classes.

The entire MED pipeline is composed of 4 parts: 1) extracting audio part 2) extracting MFCCs 3) Use k-means clustering to extract Bag-of-Words features 4) extracting SoundNet features. 5) training the model 6) evaluation the model on the validation set.

1) extracting audios takes around 20 minutes. 2) extracting MFCCs takes around 1 hour and 40 minutes. 3) extracting BoW features takes around 15 minutes 4) extracting SoundNet features takes around 1 hour and 35 minutes 5) reading the input data and training the model takes around 7-9 minutes 6) evaluating the performance takes around 15-30 seconds.

AWS Credits Left: I have used around 53 instance hours of t2.large ($0.09/hours), which sums up to be $4.8 in total. I have $50-$4.8 = $45.2 left for the rest of the semester.