Yun Jia Zhuang & Janelle Tam & Sina Salmannia Mar 22nd, 2024

MAIS 202 - Final Results

1. Final Training Results

We have tried a modified version of the model from our preliminary results.
Firstly, since we do not have sufficiently many audio files labelled "calm" to be able to accurately classify them, we have removed "calm"-labelled audio from our dataset.

We have also included a new dataset: EmoDB. We have only kept the emotions that were used for our previous model.

So the modified model is as follows:

- Datasets: keep only emotions among neutral, happy, sad, angry, fearful, disgust, surprise, from the RAVDESS, TESS, CREMA, SAVEE, and EmoDB datasets.
- Data preprocessing:
    - Remove silence from audio files (via pydub's split_on_silence)
    - Resample to sample rate = 24000
    - Normalize audio
    - Either pad audio to 4 seconds, or truncate to 4 seconds
    - Extract MFCC, delta, delta-delta features from processed audio

- CNN model:

```
(0): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(1): ReLU()
(2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
    ceil_mode=False)
(3): Dropout(p=0.2, inplace=False)
(4): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(5): ReLU()
(6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
    ceil_mode=False)
(7): Dropout(p=0.2, inplace=False)
(8): Flatten(start_dim=1, end_dim=-1)
(9): Linear(in_features=27072, out_features=256, bias=True)
(10): ReLU()
(11): Linear(in_features=256, out_features=128, bias=True)
(12): ReLU()
(13): Linear(in_features=128, out_features=7, bias=True)
(14): Softmax(dim=1)
```

The only change here is the number of in_features to the first Linear layer, and the number of out_features in the final linear layer, since we are now working with 7 classes instead of 8.

These changes have not significantly improved our results.
Out of the two learning rates we have tested (lr=0.0005, 0.001), lr=0.001 performs better, yet we still see a validation accuracy of approximately 56-58% accuracy across all folds.

This is as opposed to a 61% accuracy for our best fold in the preliminary results.

We can still observe overfitting across all folds and all learning rates.

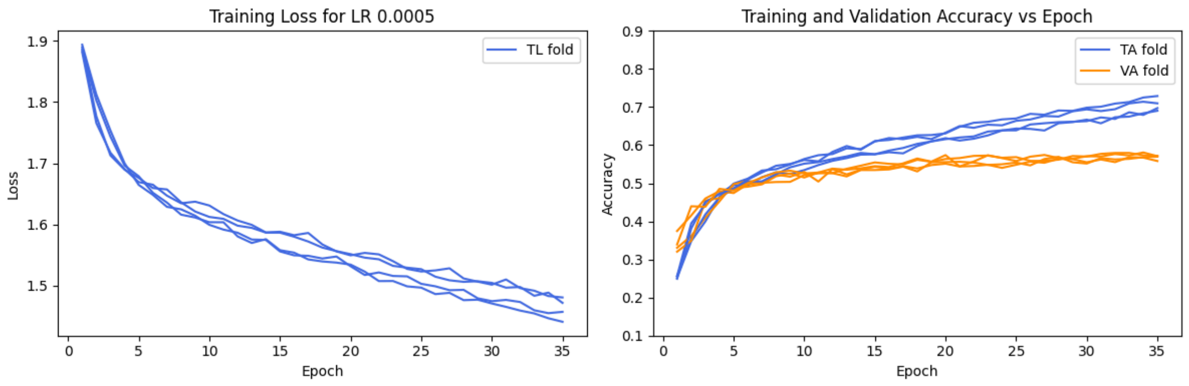


Fig. 1: Training loss and training/validation accuracies over all folds (k=4), lr=0.0005. Training accuracy steadily increases and reaches 70%, while validation accuracy remains stagnant at around 55%. This indicates overfitting.

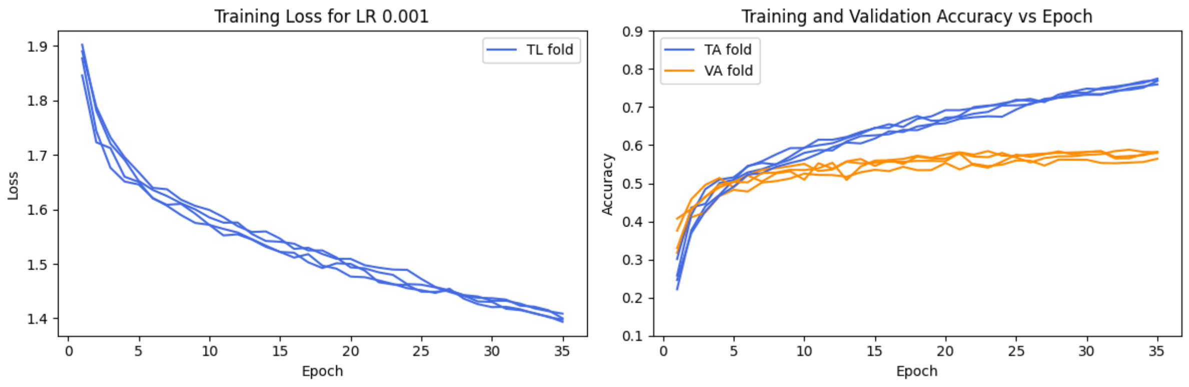
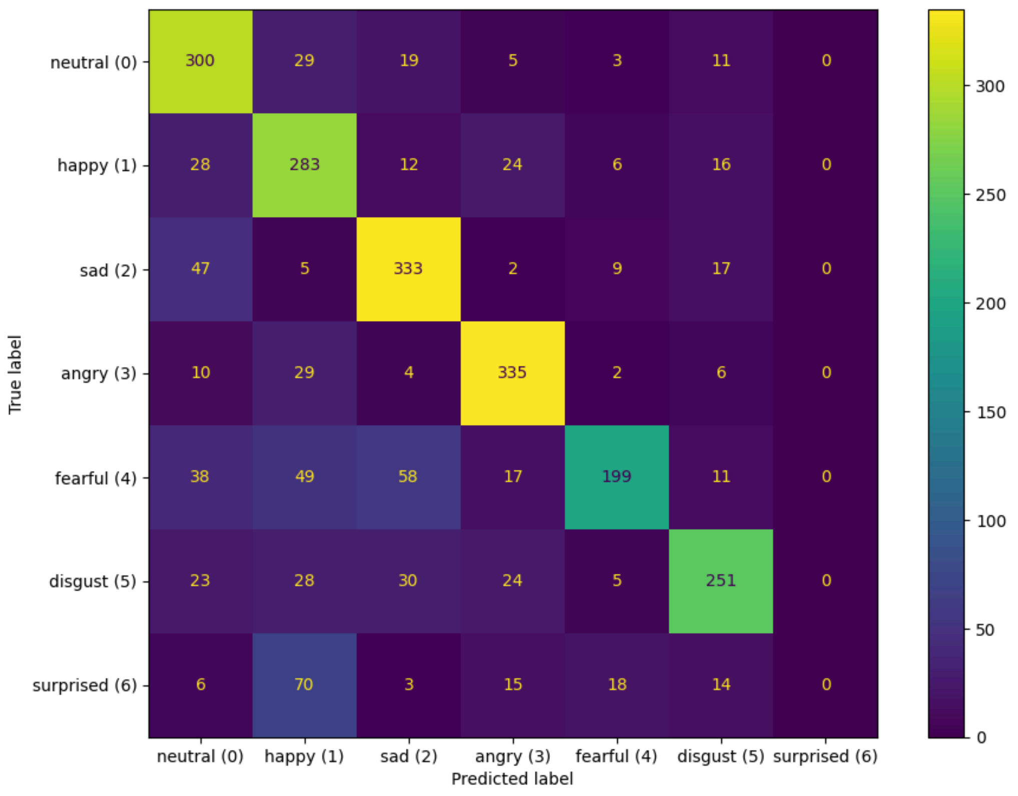
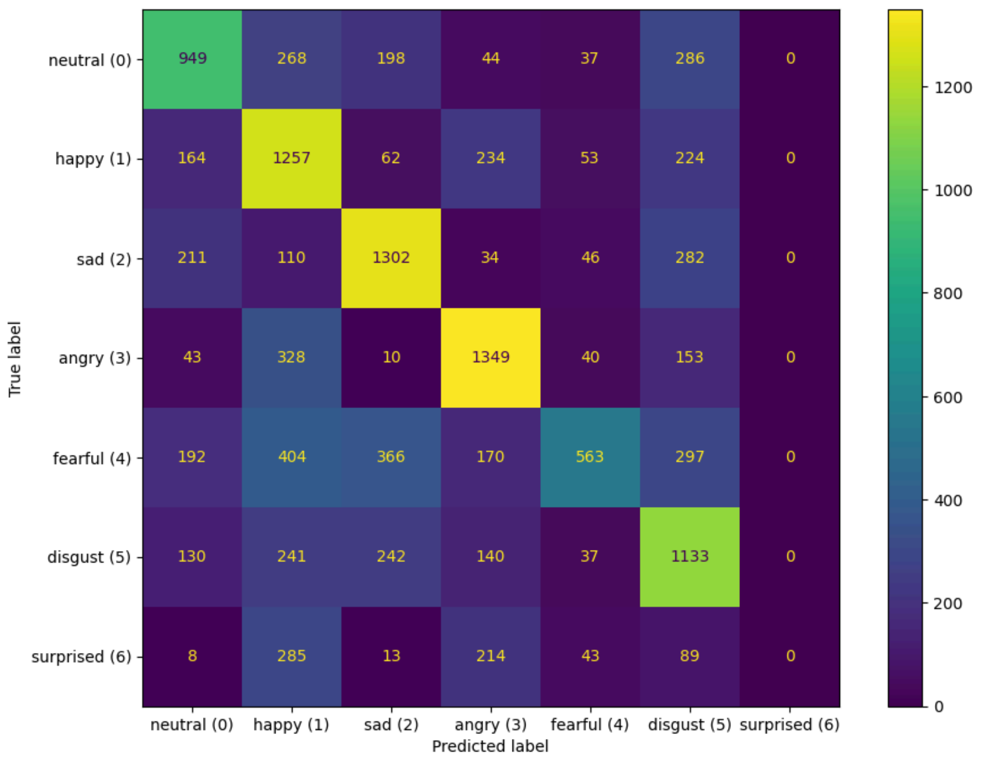


Fig. 2: Training loss and training/validation accuracies over all folds (k=4), lr=0.001. Training accuracy steadily increases and reaches 77%, while validation accuracy remains stagnant at around 55-58%. This indicates overfitting.



True label \ Predicted label	neutral (0)	happy (1)	sad (2)	angry (3)	fearful (4)	disgust (5)	surprised (6)
neutral (0)	300	29	19	5	3	11	0
happy (1)	28	283	12	24	6	16	0
sad (2)	47	5	333	2	9	17	0
angry (3)	10	29	4	335	2	6	0
fearful (4)	38	49	58	17	199	11	0
disgust (5)	23	28	30	24	5	251	0
surprised (6)	6	70	3	15	18	14	0

Fig. 3: Confusion matrix on test dataset. We notice that our model never predicted the ‘surprise’ label.



True label \ Predicted label	neutral (0)	happy (1)	sad (2)	angry (3)	fearful (4)	disgust (5)	surprised (6)
neutral (0)	949	268	198	44	37	286	0
happy (1)	164	1257	62	234	53	224	0
sad (2)	211	110	1302	34	46	282	0
angry (3)	43	328	10	1349	40	153	0
fearful (4)	192	404	366	170	563	297	0
disgust (5)	130	241	242	140	37	1133	0
surprised (6)	8	285	13	214	43	89	0

Fig. 4: Confusion matrix on train dataset. We notice that our model never predicted the surprise label, even on the train dataset. This seems like an anomaly.

Yun Jia Zhuang & Janelle Tam & Sina Salmannia Mar 22nd, 2024

Observations:
Our model NEVER predicts ‘surprise’ as the label for any input data.
We have tried looking into what caused this issue (as it was not present in our previous model), but we have been unable to find the root cause.

Therefore, we will not be using this modified model, and will have the model from the preliminary results as our backup model. We will keep working to try to fix this while we work on the webapp.

2. Final Demonstration Proposal

We are planning on creating a webapp with a React frontend and Flask backend.
Some of us have experience in working with Flask in other projects, but none of us have used React before.

The idea is to have the landing page allow for audio recording, which will be sent to the backend for prediction. The prediction will then be sent back to the landing page and displayed, with the probability/confidence for the predicted class displayed.

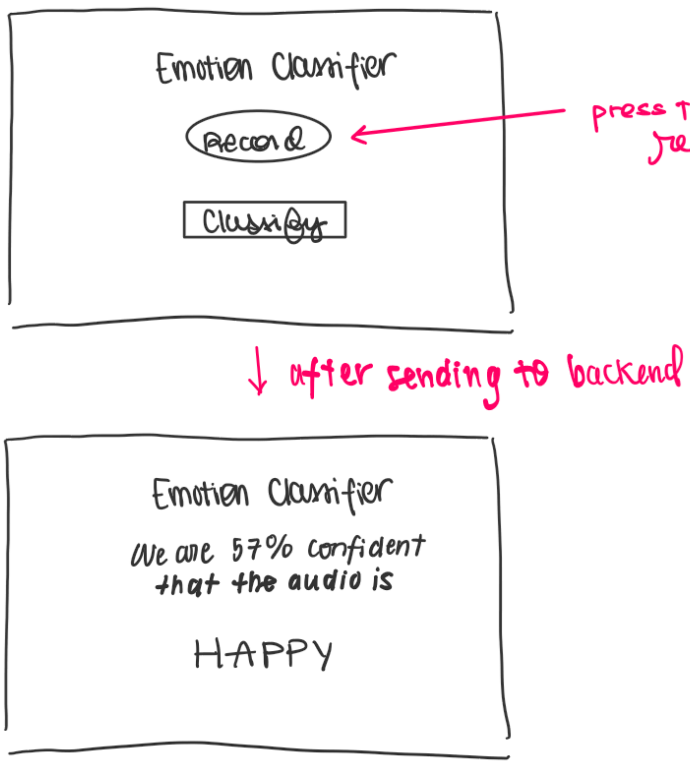


Fig. 5: Sketch of implementation of webapp, and user interface.

We will be looking at online tutorials (YouTube, websites) to learn how to use React and integrate it with a Flask backend.