Classifying Cat Meows by Emotion using Neural Networks

CS 109B Final Project - Spring 2024

Mads Groeholdt, Sean McCabe, Josie Mobley, Bridget Sands

Problem Statement!

Given an audio file of a cat's meow, is it:

1. Angry
2. Defence
3. Fighting
4. Happy
5. HuntingMind
6. Mating
7. MotherCall
8. Painting
9. Resting
10. Warning

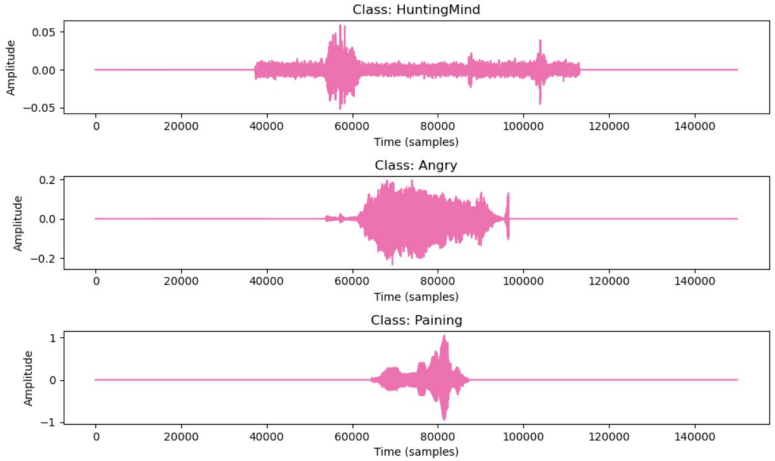
To answer this question, we attempt to create a model that reads in a preprocessed meow file and can tell us!

We are both inspired by and working off of Yagya Raj Pandeya, Dongwhoon Kim, and Joonwhoan Lee's paper, *Domestic Cat Sound Classification Using Learned Features from Deep Neural Nets*, originally published in October 2018. The group is the source of our data as well as well as the baseline ideas of this project.

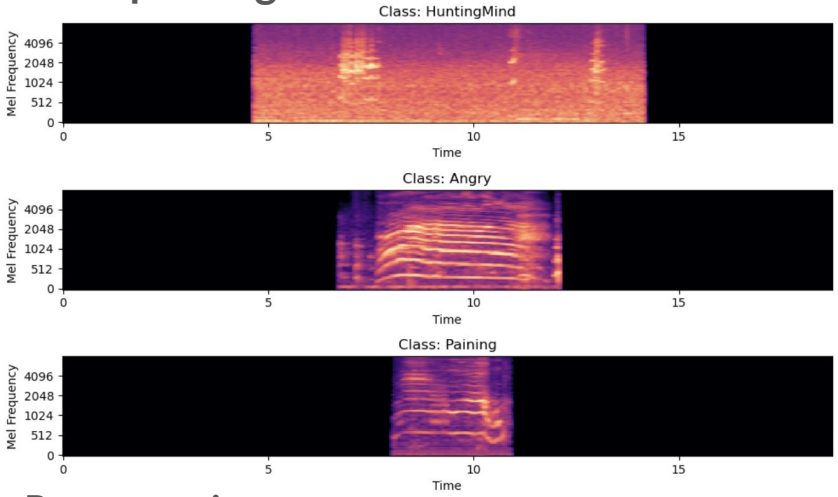
Data, EDA, & Preprocessing:

*Note: Because of space it didn't make sense to show visualizations for each of the 10 emotions

Raw Data:



Mel Spectrogram:



The Data:

- Acquired from group
- About ~3000 Mp3 Audio samples
- 10 classification labels, relatively balanced
- Different durations, different amplitudes→ no obvious consistencies
- Max length 150,000 → truncated, padded
- .2 Train Test Split
  - ~2400 train, ~600 test samples
  - Classes represented evenly

Preprocessing :

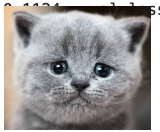
- Visualize example of each category, inconsistent
- Identify file length cut off→ max length
- Pad all samples to the max length
- Convert to Mel Spectrogram
- Augmentation
  - Randomly selected speed change, pitch shifting, dynamic range compression, insertion of noise, time shifting on previous files

Basic CNN (baseline model):

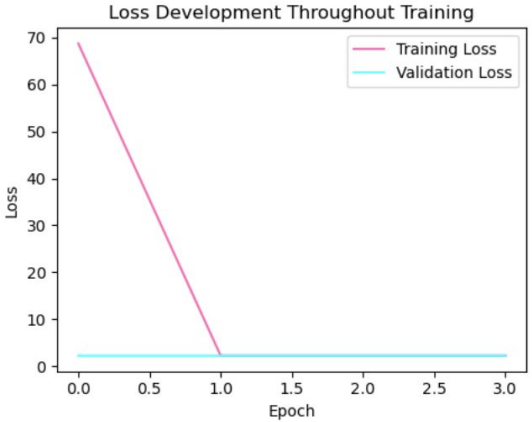
Model Architecture:

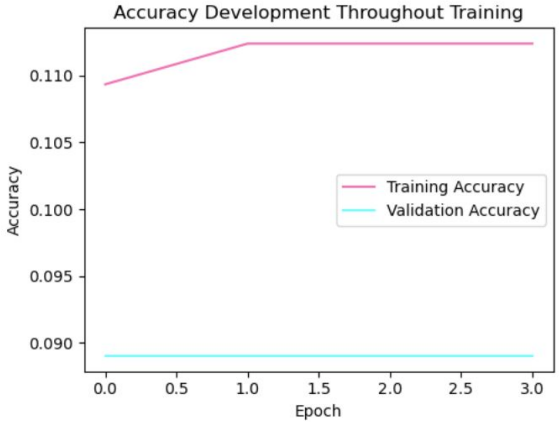
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 96, 586, 1)]	0
conv2d (Conv2D)	(None, 96, 586, 32)	320
max_pooling2d (MaxPooling2D)	(None, 48, 293, 32)	0
flatten (Flatten)	(None, 450048)	0
dense (Dense)	(None, 32)	14401568
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 10)	330

```
Epoch 1/4
118/118 [==============================] - 23s 190ms/
step - loss: 68.6376 - accuracy: 0.1093 - val_loss:
2.3030 - val_accuracy: 0.0890
Epoch 2/4
118/118 [==============================] - 22s 189ms/
step - loss: 2.3020 - accuracy: 0.1124 - val_loss: 2.
3036 - val_accuracy: 0.0890
Epoch 3/4
118/118 [==============================] - 22s 189ms/
step - loss: 2.3017 - accuracy: 0.1124 - val_loss: 2.
3042 - val_accuracy: 0.0890
Epoch 4/4
118/118 [==============================] - 22s 189ms/
step - loss: 2.3015 - accuracy: 0.1124 - val_loss: 2.
3047 - val_accuracy: 0.0890
```



Training and Evaluation Metrics:

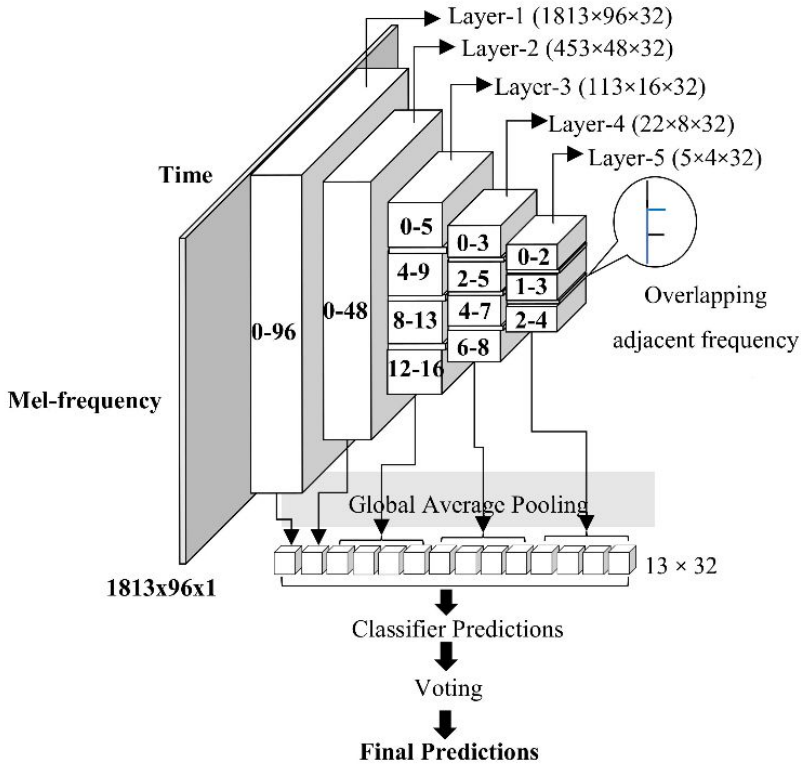




Notes and Takeaways:

- Basic CNN: 1 Convolution layer, 1 Max Pool layer
- Takes in pre-processed Mel Spectrogram data,
  - 150,000 max length→ padded/truncated
  - Created using librosa mel spectrogram
- Adam, sparse categorical cross entropy, accuracy
- 64 batch size, 4 epochs
- Significant training time for small number of epochs
- Horrible results, basically random
- Better fitting model will be more complex

CNN 2 (paper remake): pt1

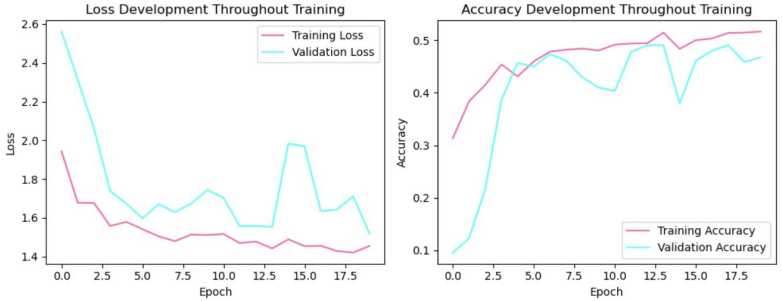


Model Architecture:

- Recreated architecture of paper used pre-trained model on million song dataset
- Same amount of filters, layers, relative dimensions
- Added batch normalization, dropouts, max pooling to each convolution layer
- Has classification in the pipeline rather than paper's method of exporting to additional classification algorithms
- Flattened layer of 13x32 GAP 2D array as final input
- Sparse categorical cross entropy, accuracy minimized
- Adam, .01 learning rate, 20 epochs
- Trained on non-augmented and augmented data

CNN 2 (paper remake): pt2

Non-augmented Data:

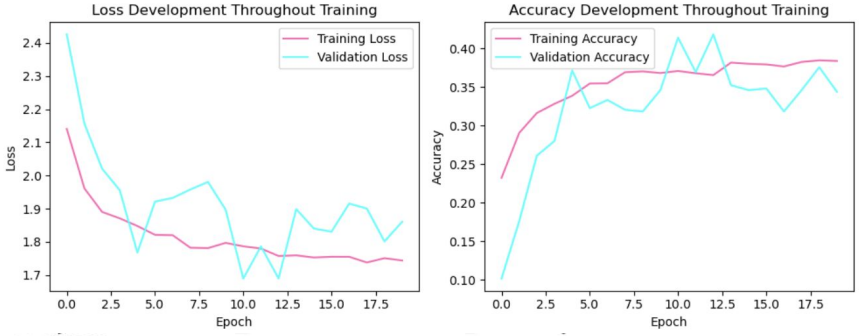



```
Epoch 20/20
118/118 [==============================] - 8s 68ms/step - loss: 1.4546 -
accuracy: 0.5165 - val_loss: 1.5207 - val_accuracy: 0.4671
```

Training accuracies varied between ~.45 - ~.55
Validation accuracies varied between ~.44 - ~.51

	precision	recall	f1-score	support
Angry	0.70	0.27	0.39	60
Defence	0.44	0.79	0.57	58
Fighting	0.74	0.52	0.61	60
Happy	0.33	0.33	0.33	58
HuntingMind	0.46	0.41	0.44	58
Mating	0.41	0.55	0.47	60
MotherCall	0.38	0.71	0.49	59
Paining	0.74	0.25	0.37	57
Resting	0.83	0.32	0.46	59
Warning	0.34	0.43	0.38	60
accuracy			0.46	589
macro avg	0.54	0.46	0.45	589
weighted avg	0.54	0.46	0.45	589

Augmented Data Included:

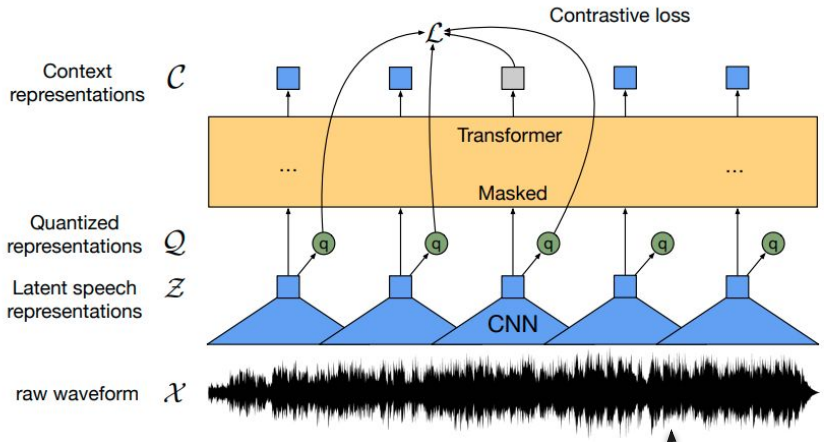



```
Epoch 20/20
236/236 [==============================] - 15s 65ms/step - loss: 1.7439 -
accuracy: 0.3839 - val_loss: 1.8606 - val_accuracy: 0.3439
```

Training accuracies varied between ~.35 - ~.45
Validation accuracies varied between ~.33 - ~.41

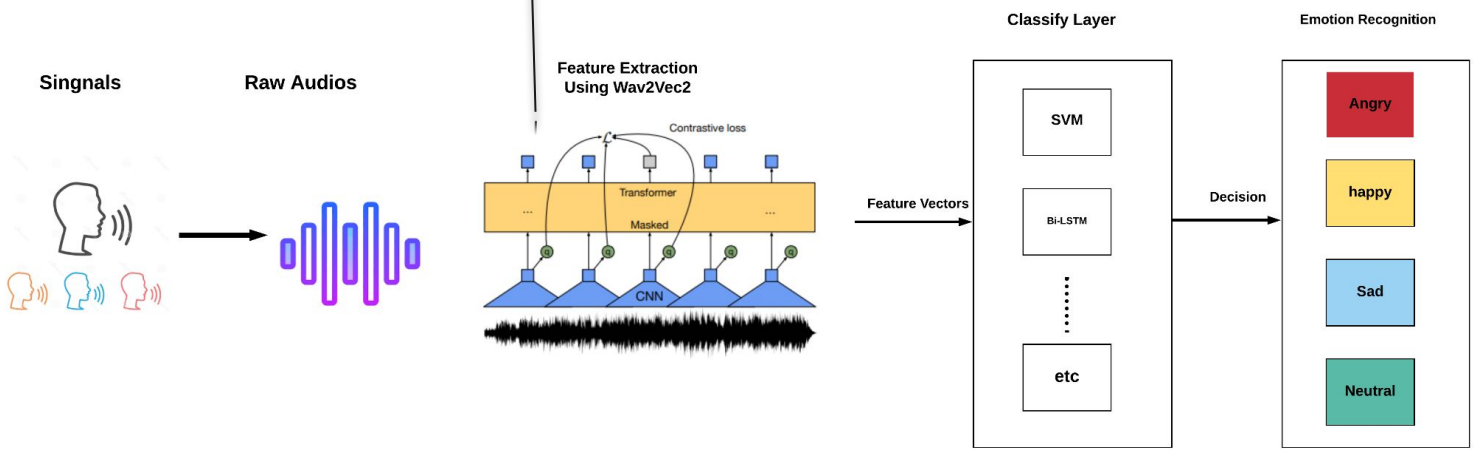
	precision	recall	f1-score	support
Angry	0.82	0.15	0.25	60
Defence	0.24	0.97	0.38	58
Fighting	0.77	0.40	0.53	60
Happy	0.20	0.17	0.18	58
HuntingMind	1.00	0.03	0.07	58
Mating	0.32	0.60	0.41	60
MotherCall	0.46	0.19	0.27	59
Paining	0.50	0.16	0.24	57
Resting	0.74	0.34	0.47	59
Warning	0.28	0.35	0.31	60
accuracy			0.34	589
macro avg	0.53	0.34	0.31	589
weighted avg	0.53	0.34	0.31	589

Transfer Learning: pt1



Model Architecture:

- Facebook AI Wave2Vec2 base model: Trained on 100,000+ hours of cross-lingual speech data
- Precedent for animal sound classification
- Fine-tuning attempts with frozen and unfrozen convolutional layers and varying hyperparams
- Fine-tuned for 3 epochs with lr=2e-5 as best
- Time-intensive training (30 min/epoch) on GPUs



Transfer Learning: pt2

Non-augmented Data:

Training History

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.808734	0.309979
2	2.109200	1.436614	0.562633
3	1.696600	1.274571	0.728238

Softmax Certainty MotherCall 35.8%

Accuracy by Category

	precision	recall	f1-score	support
Angry	0.68	0.75	0.71	48
Defence	0.86	0.93	0.90	46
Fighting	0.59	0.54	0.57	48
Happy	0.53	0.83	0.64	47
HuntingMind	0.90	0.78	0.84	46
Mating	0.82	0.58	0.68	48
MotherCall	0.83	0.83	0.83	47
Paining	0.63	0.37	0.47	46
Resting	0.85	0.98	0.91	47
Warning	0.69	0.69	0.69	48
accuracy			0.73	471
macro avg	0.74	0.73	0.72	471
weighted avg	0.74	0.73	0.72	471

Augmented Data Included:

Training History

Epoch	Training Loss	Validation Loss	Accuracy
1	1.694400	0.931758	0.730361
2	1.219900	0.844043	0.768578
3	1.007200	0.650730	0.842887

Softmax Certainty MotherCall 88.7%

Accuracy by Category

	precision	recall	f1-score	support
Angry	0.83	0.90	0.86	48
Defence	0.98	0.91	0.94	46
Fighting	0.89	0.81	0.85	48
Happy	0.72	0.89	0.80	47
HuntingMind	0.87	0.85	0.86	46
Mating	0.84	0.77	0.80	48
MotherCall	0.91	0.89	0.90	47
Paining	0.76	0.61	0.67	46
Resting	0.85	0.98	0.91	47
Warning	0.81	0.81	0.81	48
accuracy			0.84	471
macro avg	0.85	0.84	0.84	471
weighted avg	0.85	0.84	0.84	471

Conclusions, Improvements, Future Work:

Overall: **Transfer Learning Model Most Successful** → We could benefit from more data, relationship is complicated

Strengths:

- Immense improvement in later models from baseline
- Successful implementation of advanced techniques

Weaknesses:

- Little data & data augmentation computationally expensive
- Resource limitations → failing kernels

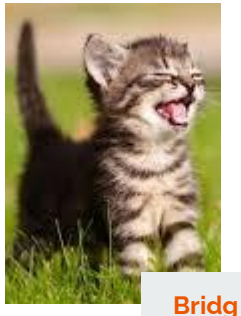
Future Work:

- More GPUs! We don't want to kill penguins with more training, but…
- More data! Better augmentation→ takes significant time
- For CNN remake models adjust training specifications
  - Decrease learning rate
  - Consequently adjust epochs
- Transfer learning model: More epochs (and "better" base models?)

Us, happy (classified by our models as such) with our results and knowing we applied what we learned in class to our final project:


Mads


Josie


Bridg


Sean