Deep Learning for Alcohol Recognition

Joseph Bernstein & Brandon Mendez

Table of Contents

Table of Contents

Motivation

After drinking, physical changes occur in the body to be able to handle the digestion of alcohol

Can this information be used to detect whether or not someone has consumed alcohol?

Why does this matter?

The Solution

Create a configurable pipeline

Take in data, output whether or not the user has consumed alcohol

By being configurable, different parameters can be tested to find optimal settings.

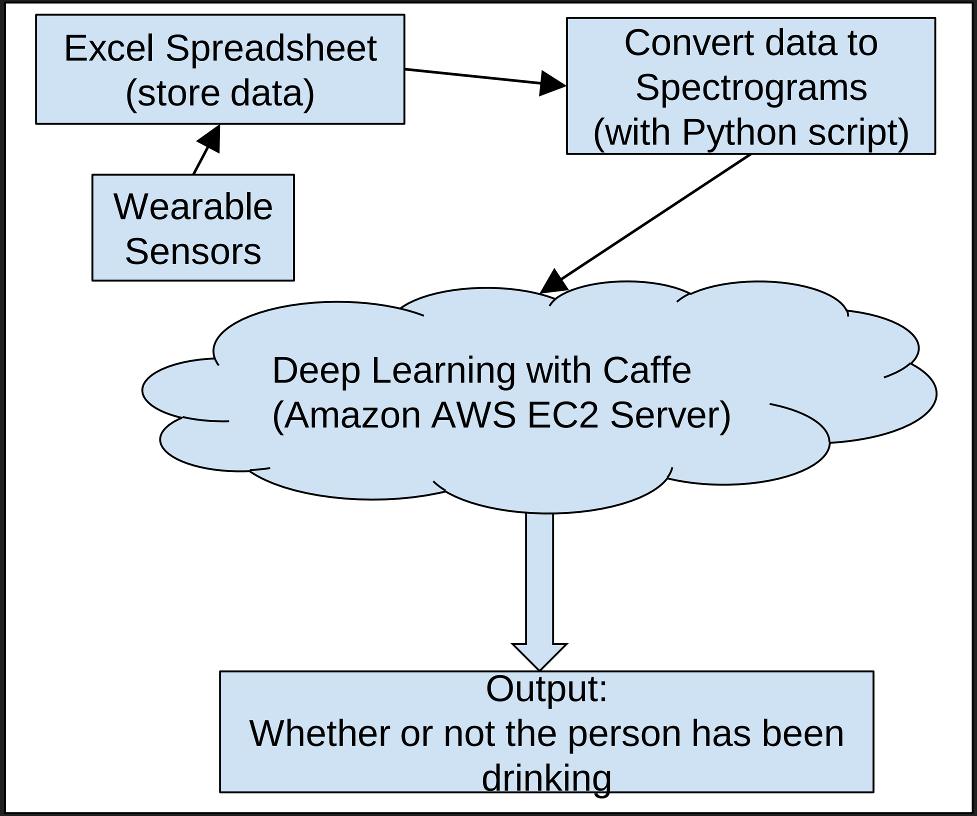
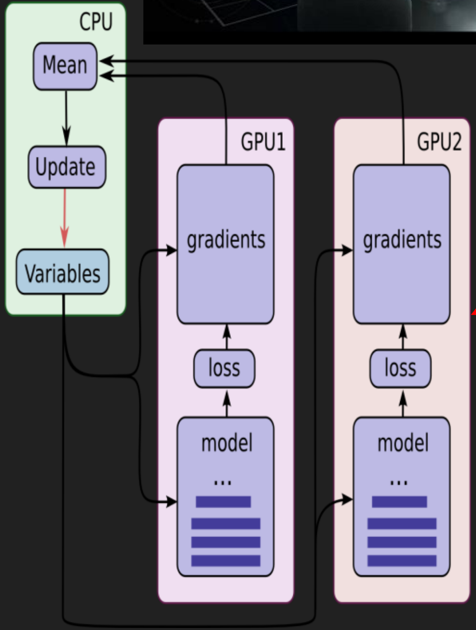
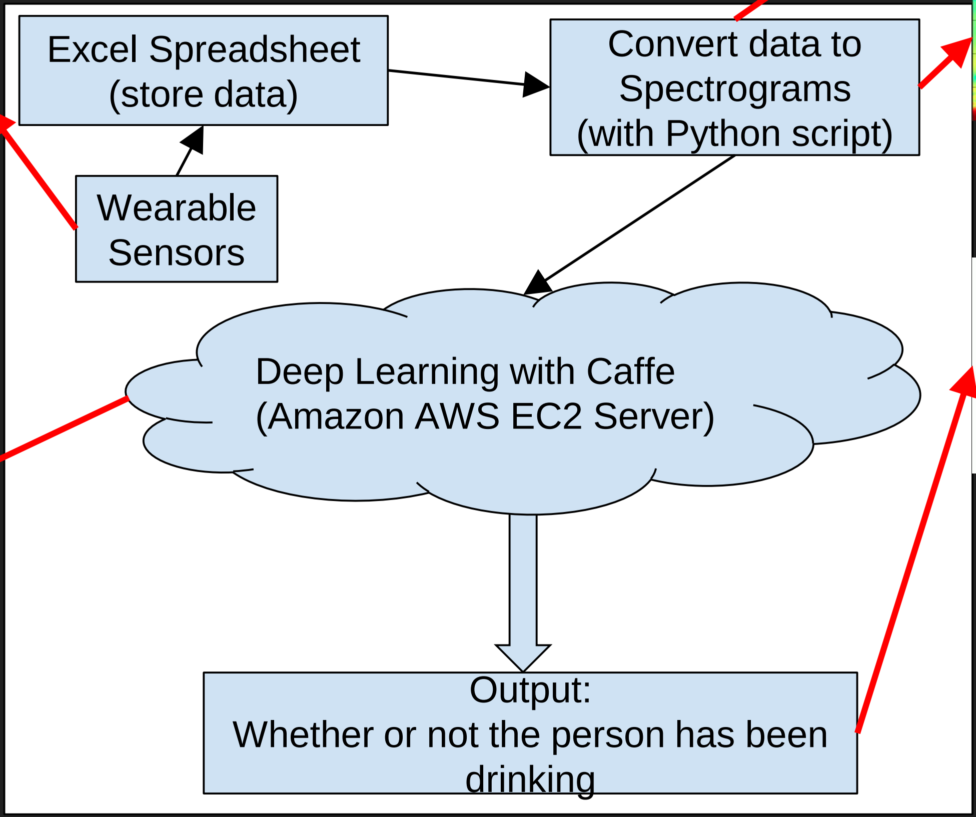
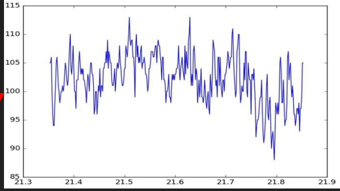


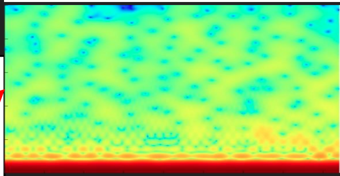
Table of Contents

The Pipeline

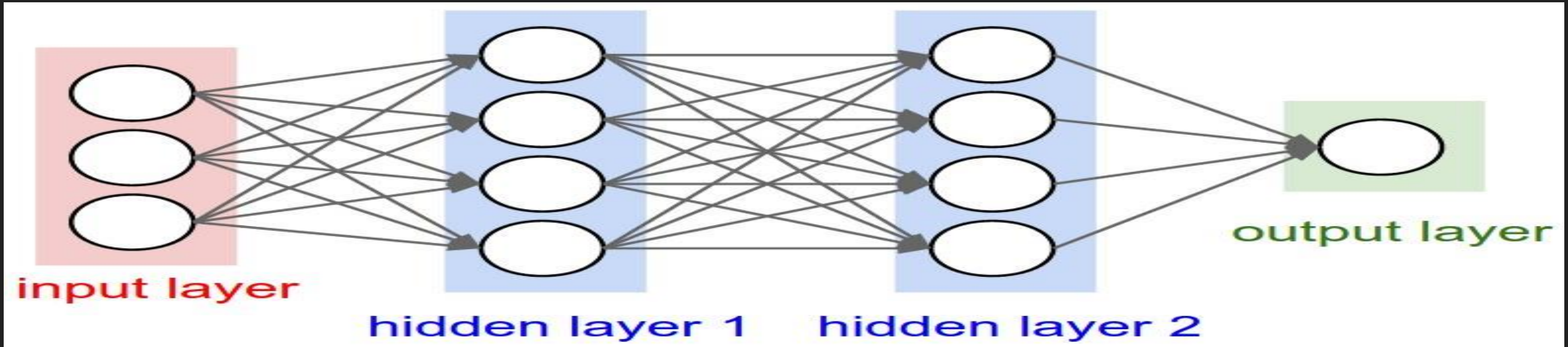


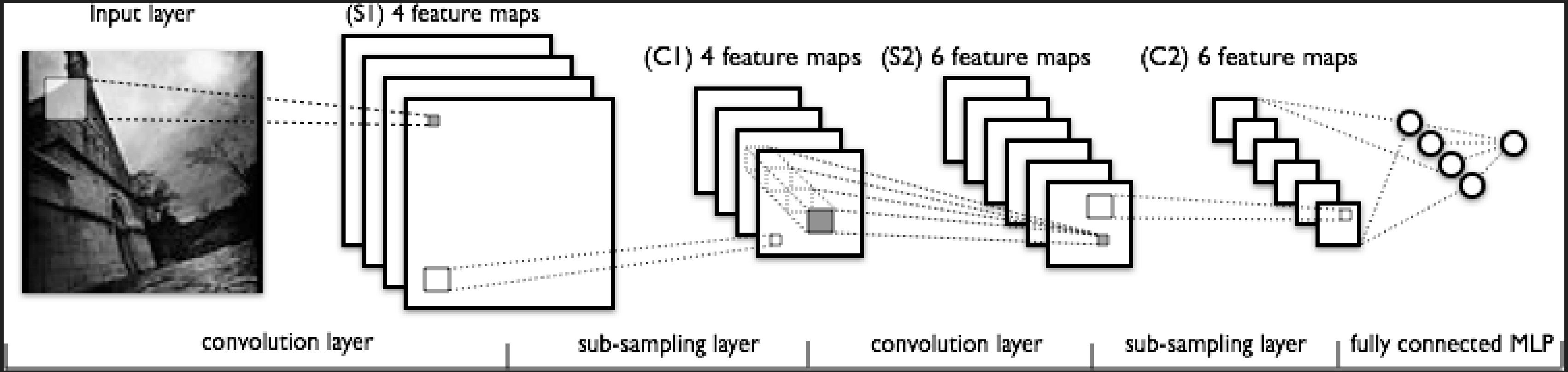






Neural Network



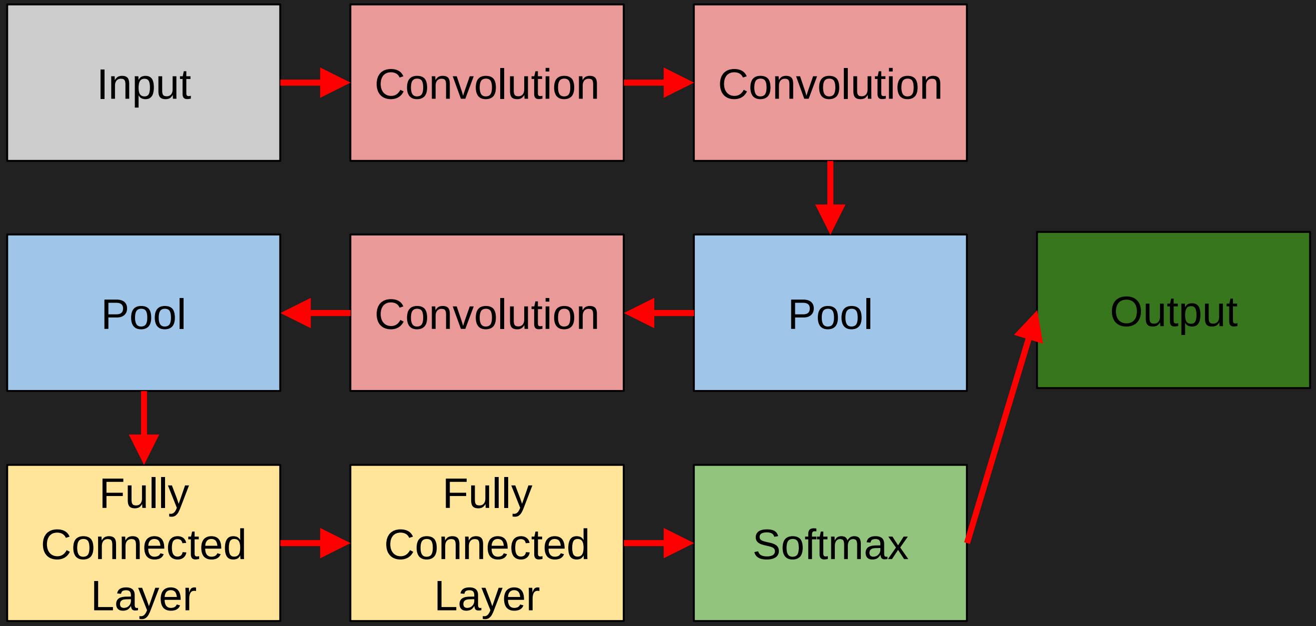


CaffeNet and AlexNet

Scales any image to 224x224 as input

Take various subsections of image to find features

Functions as “black box”



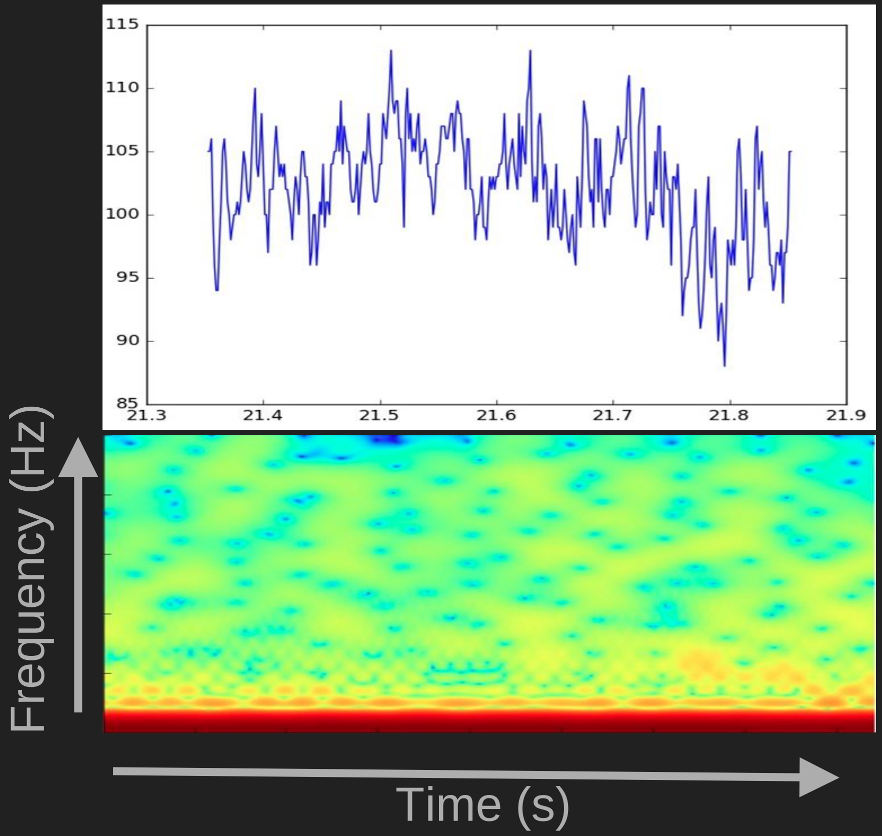
The Spectrogram

NFFT: 50, Number of Datapoints in each calculation

Fs: 0.2, Sampling Frequency, based off of time segments in the original data

Noverlap: 49, Overlap of each calculation

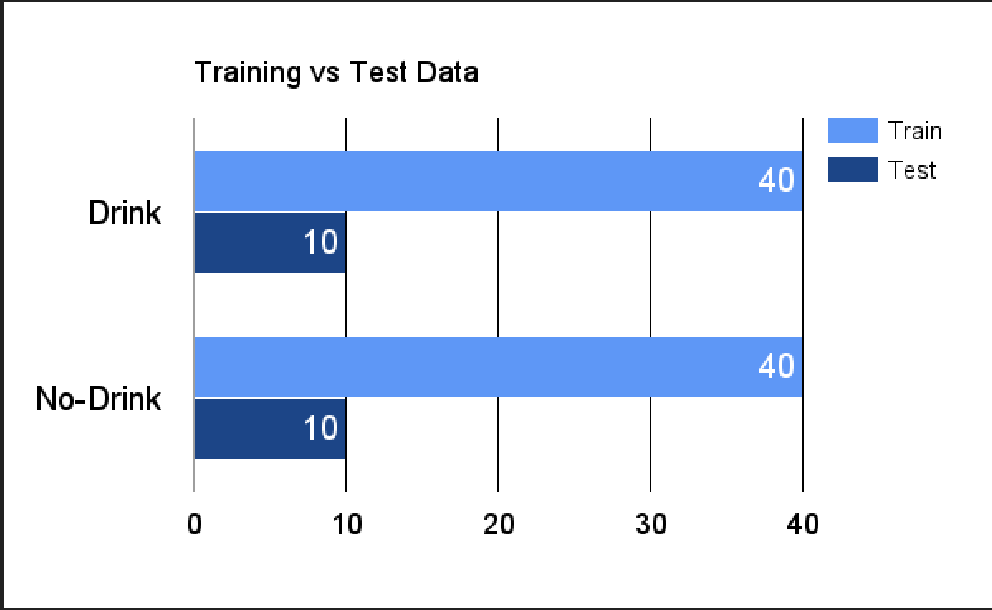
Pad To: ####, used for increasing number of calculations (more data points).



plot.Spectrogram(Datapoint=Data, NFFT=50, Fs=0.2, Noverlap=49, Pad_To=256)

The Data Used

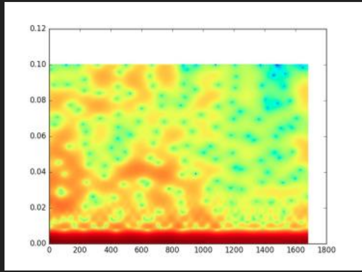
Drink	50 Spectrograms
No-Drink	50 Spectrograms
Training Data	40 Drink 40 No-Drink
Testing Data	10 Drink 10 No-Drink
Data Collected	ADA
Data Points	1,077,967
Spectrograms Generated In	Python's MatPlotLib

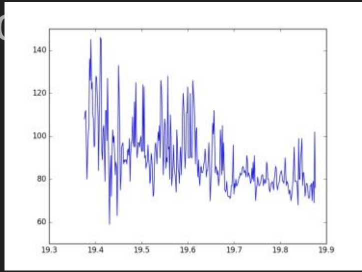


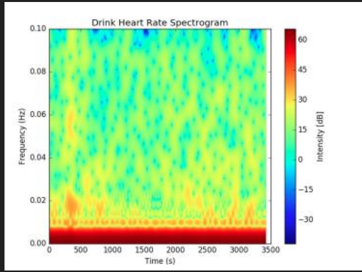
Example Spectrograms

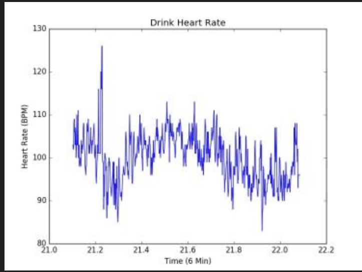
Drink Spectrograms

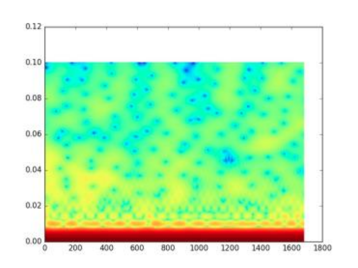
No … rams

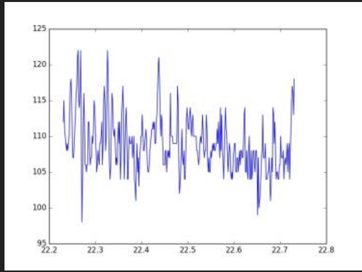


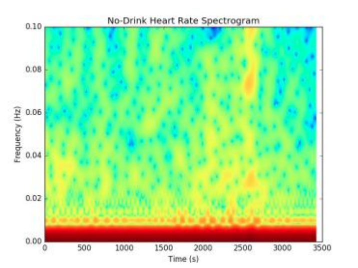


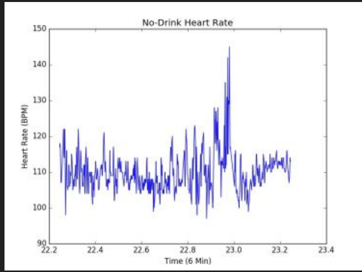












The ‘Drinking’ Spectrogram Algorithm

```
GET data FROM excelSheet

PUT data INTO table

DROP EACH row IN table WHERE there is no instance of drinking

FOR EACH row IN table of drinking instances

        GET all points IN original dataset

        REMOVE all in dataset EXCEPT those within 60 minutes of the

drink point.

    PLOT points
```

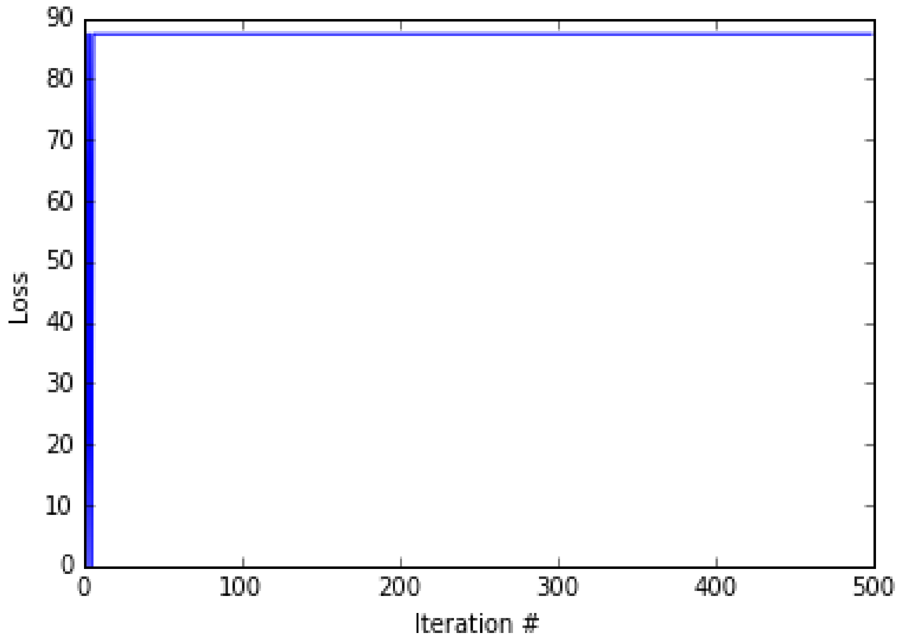
The ‘No Drinking’ Spectrograms

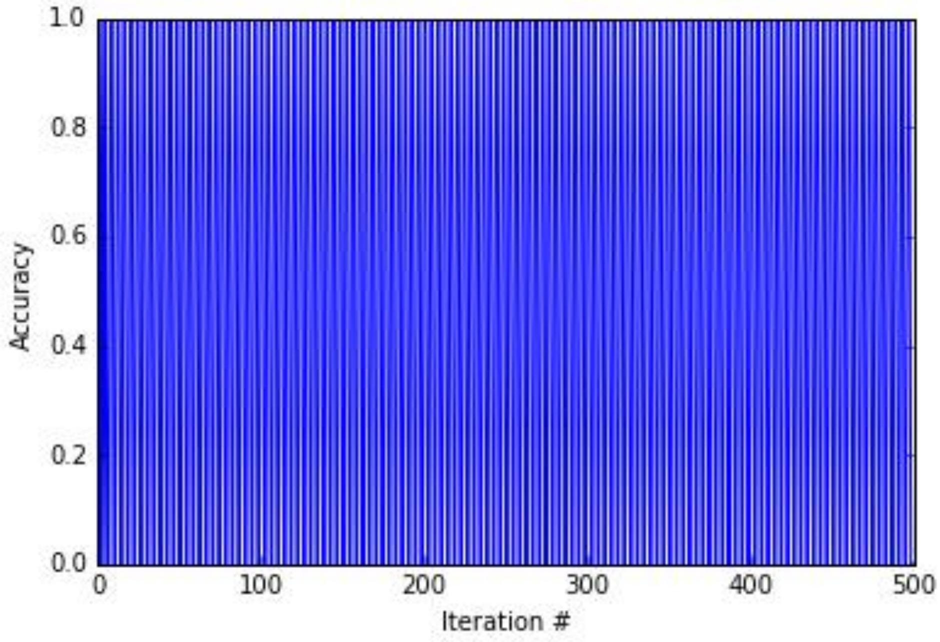
```
GET data FROM excelSheet

PUT data INTO table

DROP FROM table WHERE row was used for Drinking Spectrograms

WHILE(number of ‘No Drink’ Spectrograms ≠ number of ‘Drink’

Spectrograms)

    GET random date/time, select random 60 minute time window

    IF (random date/time NOT missing data)

        PLOT points
```

Table of Contents

Initial results

Overall Accuracy: 45%





Initial Results

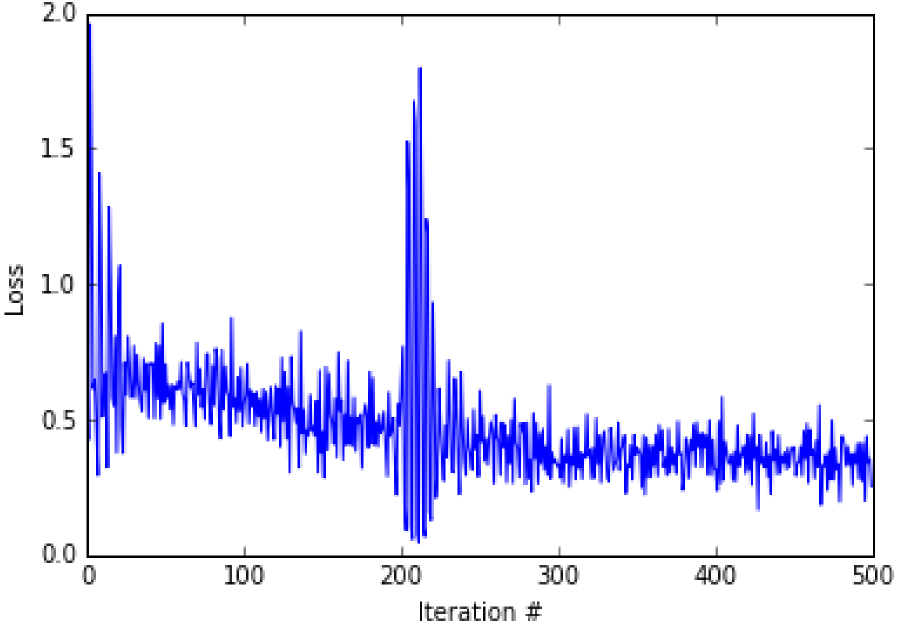
- Learning rate was too high

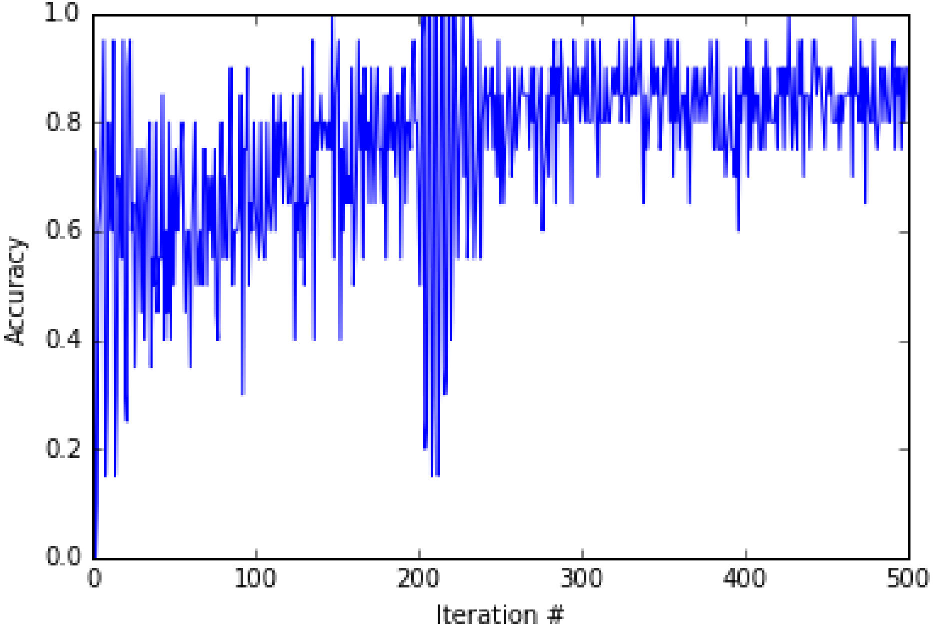
- Caffenet was being shown spectrograms but was not given enough time to learn the spectrograms

- Key to more accurate results is lower learning rate so Caffenet has more time learn spectrograms

Learning rate adjustment

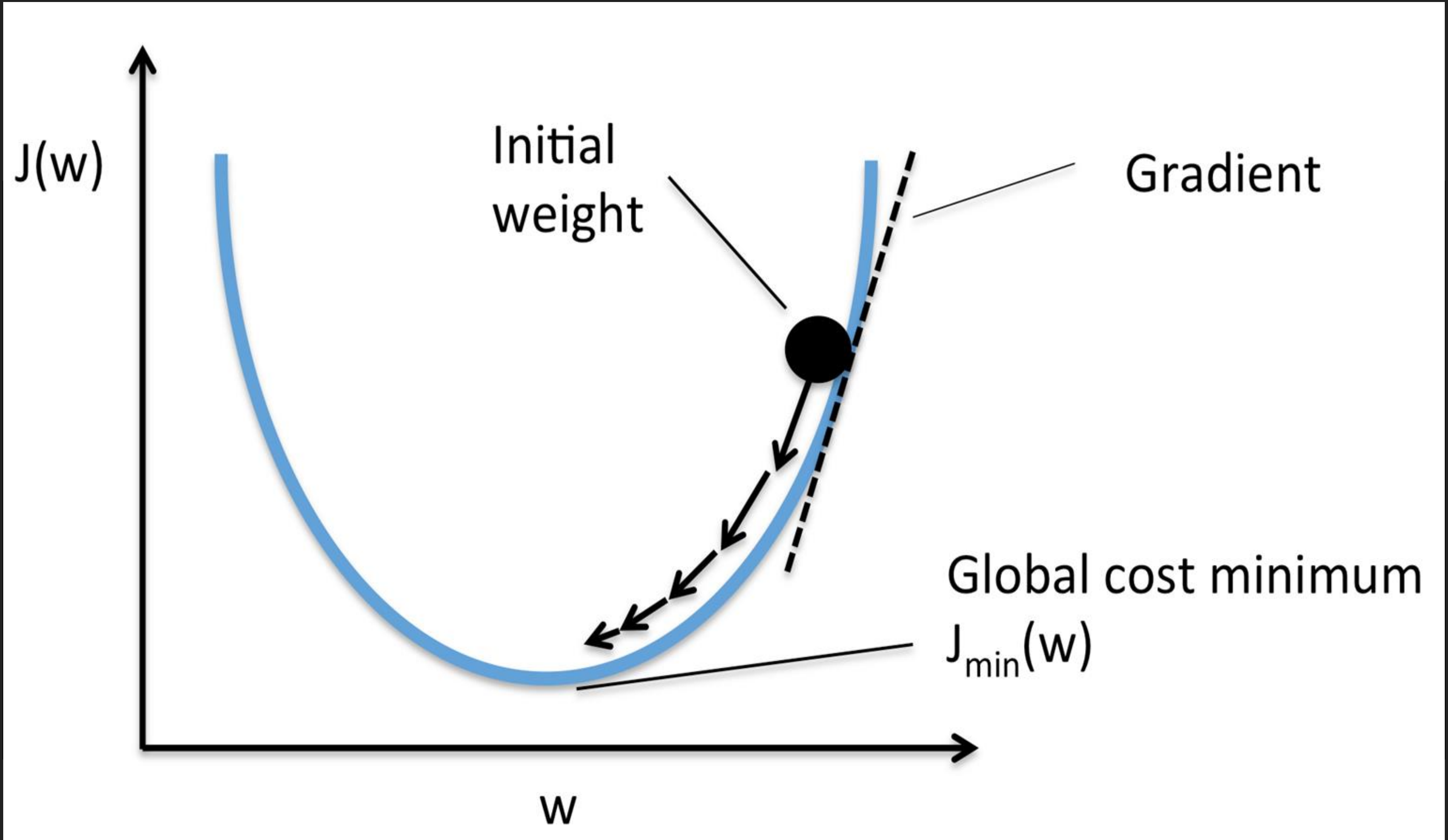
Overall Accuracy: 55%





Learning Rate Adjustment

- Results look better but could still go up so I decide to change gradient descent
- Gradient Descent is a first-order optimization algorithm to find a local minimum of a function
- Initially I used the default gradient descent which is SGD but decided to try Adam

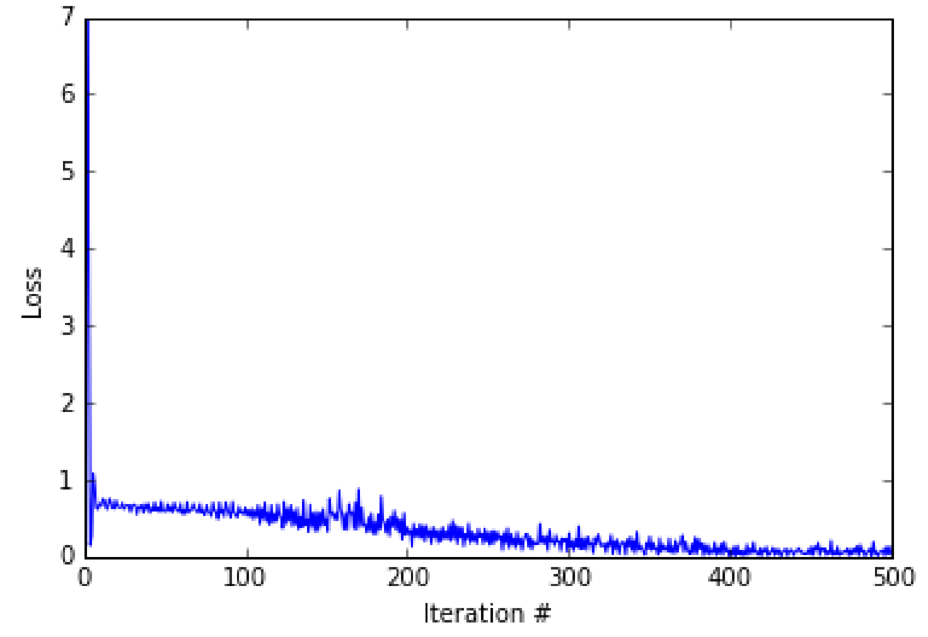

J(w)
Initial
weight
Gradient
Global cost minimum
$J_{min}(w)$
w

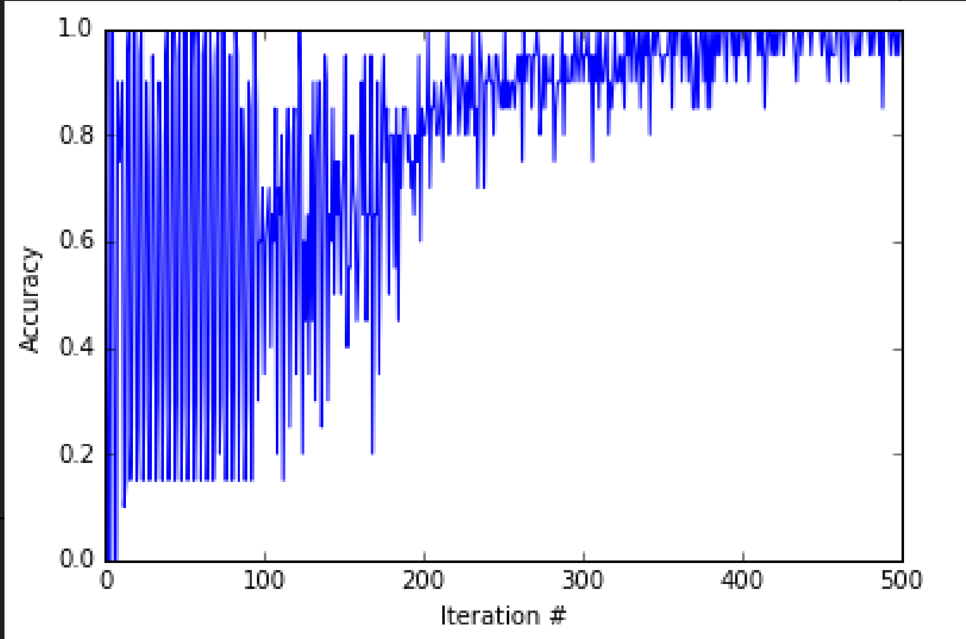
Gradient Descent

- Relates to caffenet because gradient descent will make many small iterations towards the local minimum
- For deep learning this means iterations towards editing the weights getting closer to the highest accuracy

Adam Algorithm

Overall Accuracy: 65%





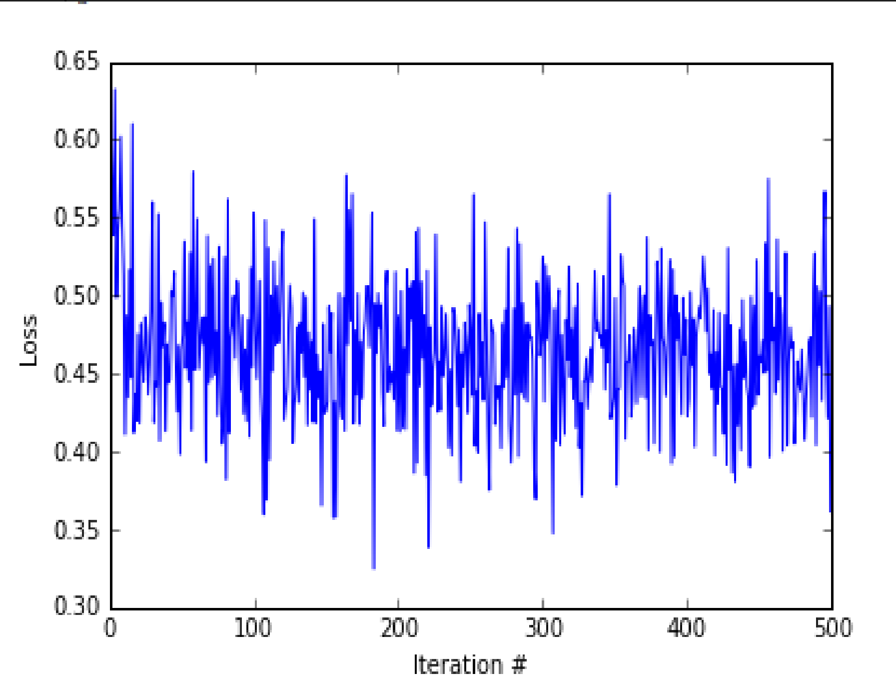
Pipeline

- After finding the right parameters using spectrograms produced in matlab I move on to the spectrograms produced by Joe in python

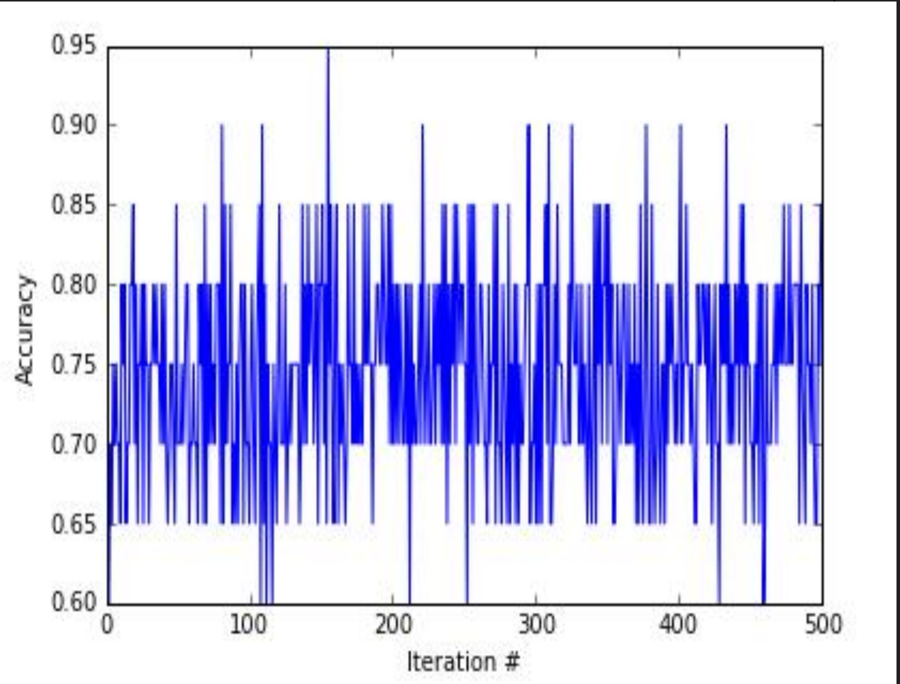
- I have tested these parameters on colored spectrograms and grayscale spectrograms along with different window sizes

- Parameters will all be the same with the only difference being I will try caffenet and alexnet

Caffenet



0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
Loss
0
100
200
300
400
500
Iteration #

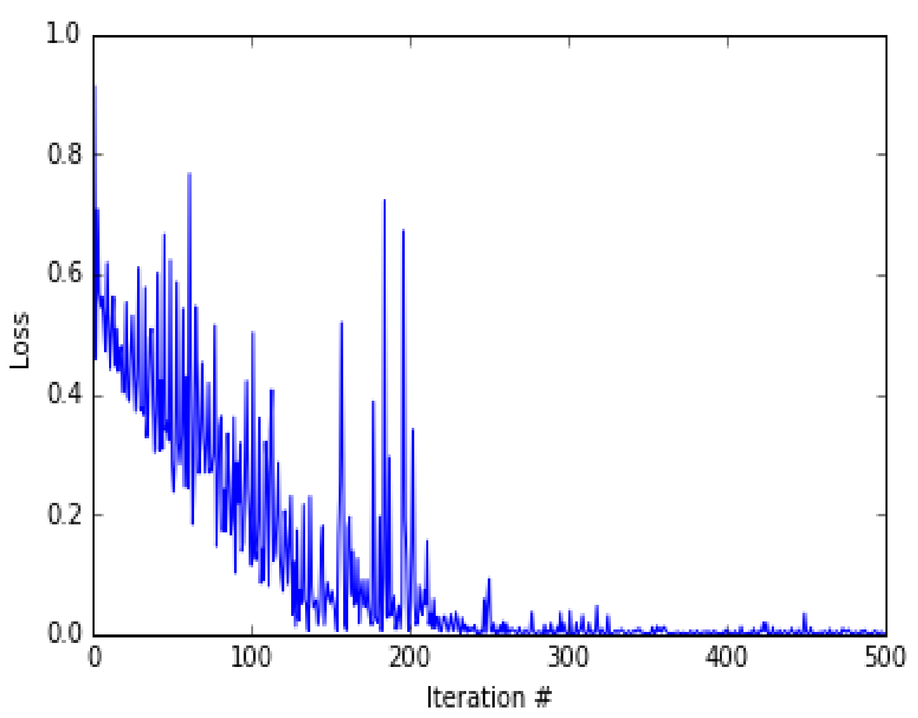


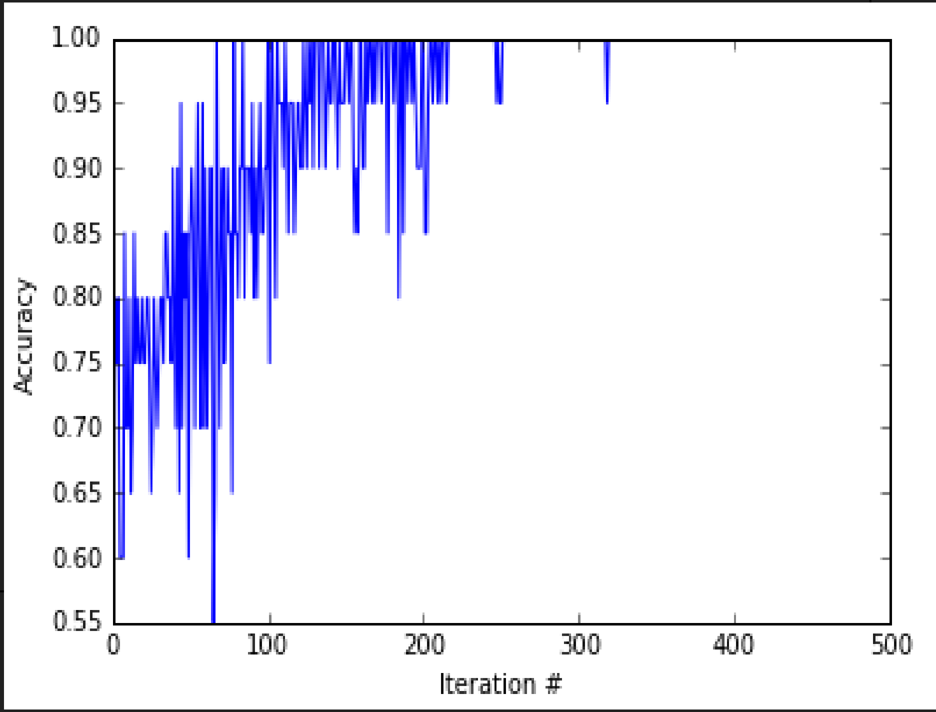
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Accuracy
0
100
200
300
400
500
Iteration #

Caffenet

- Results are not good very sporadic learning
- No real learning curve and accuracy is all over the place
- Overall all results from caffenet were less than stellar
- Decide to move onto alexnet which is just a modification of caffenet

Alexnet


1.0
0.8
0.6
0.4
0.2
0.0
Loss
0
100
200
300
400
500
Iteration #


1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
Accuracy
0
100
200
300
400
500
Iteration #

Alexnet

- Results are much better than caffenet
- Still some random spikes but the learning curve is starting to trend downward and the accuracy trend upward

Overall results

CNN	Window (Minutes)	Color	Gray Scale
AlexNet	30	61%	63%
	60	72%	59%
CaffeNet	30	56%	57%
	60	62%	60%

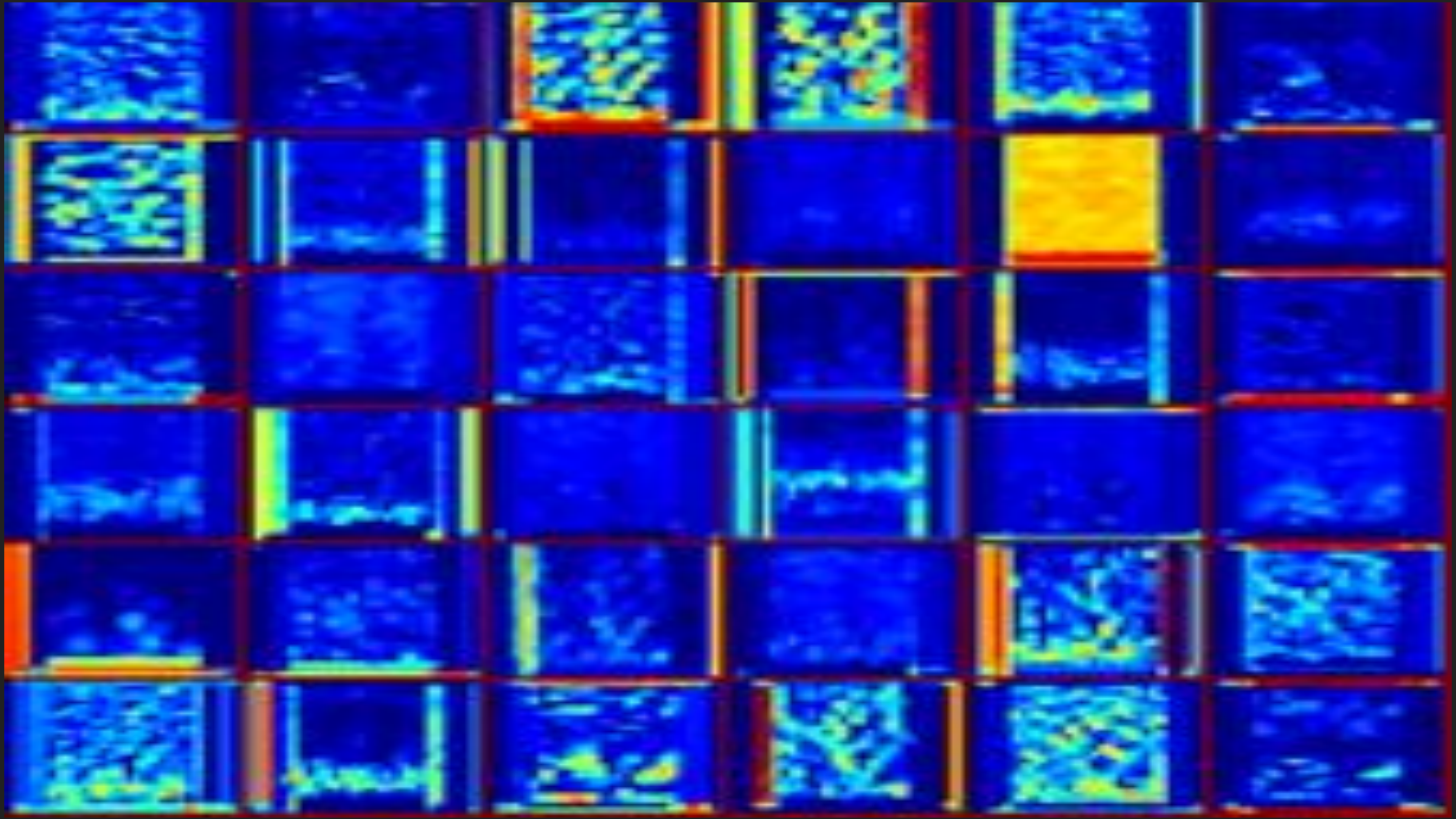
What is the correct frequency for heart rate

- We were inspired by waveforms of speech
- We know what range the frequency of waveforms of speech usually occur at however, we have no idea how what that range is for heart rate
- How can we find that range?
- The solution is using feature vectors

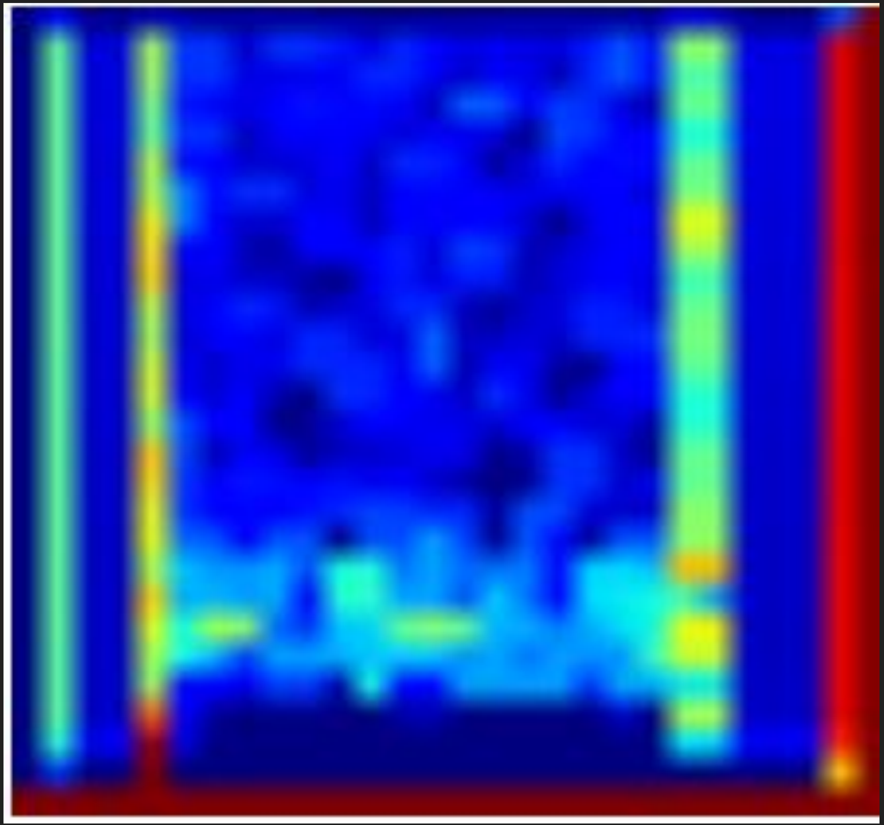
Feature Vectors

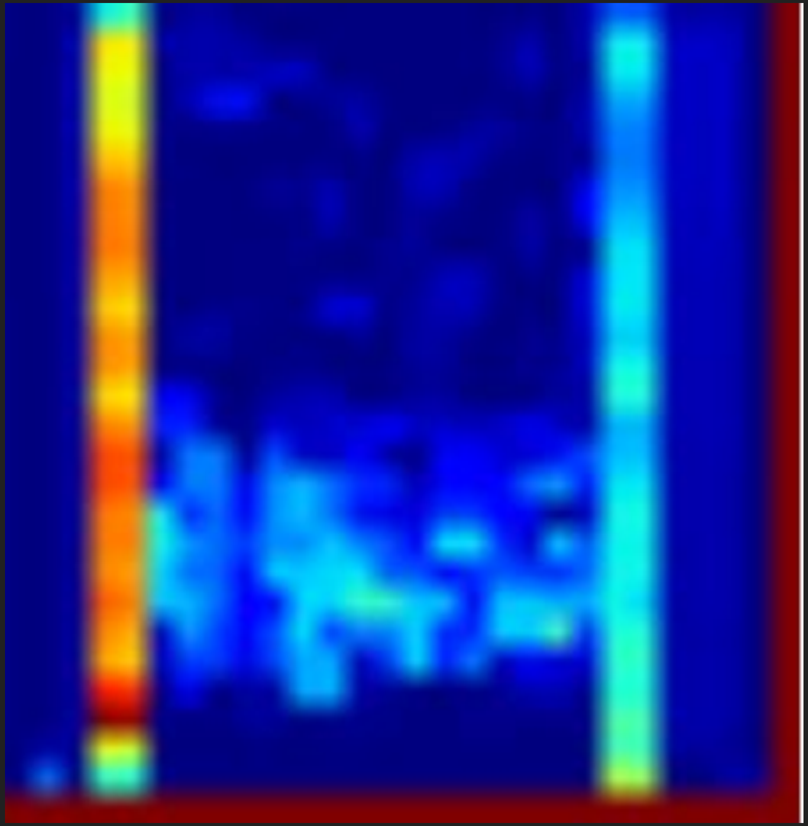
- Feature vectors show what caffenet is seeing when it is shown an image
- It can show where exactly it is extracting the most features from an image which in this case would mean show where the frequency range is

Feature Vectors

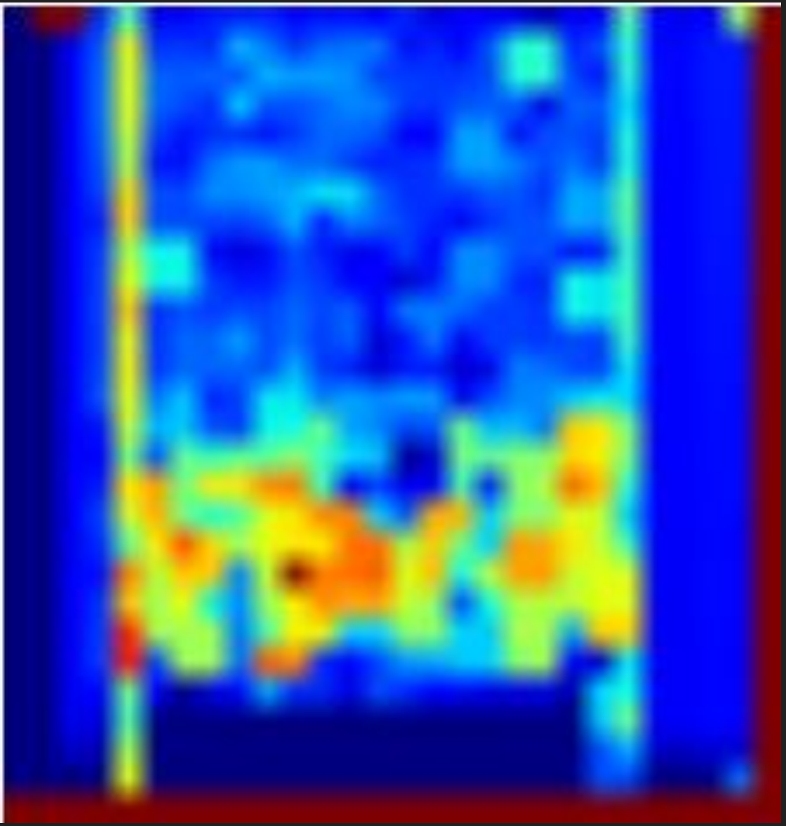


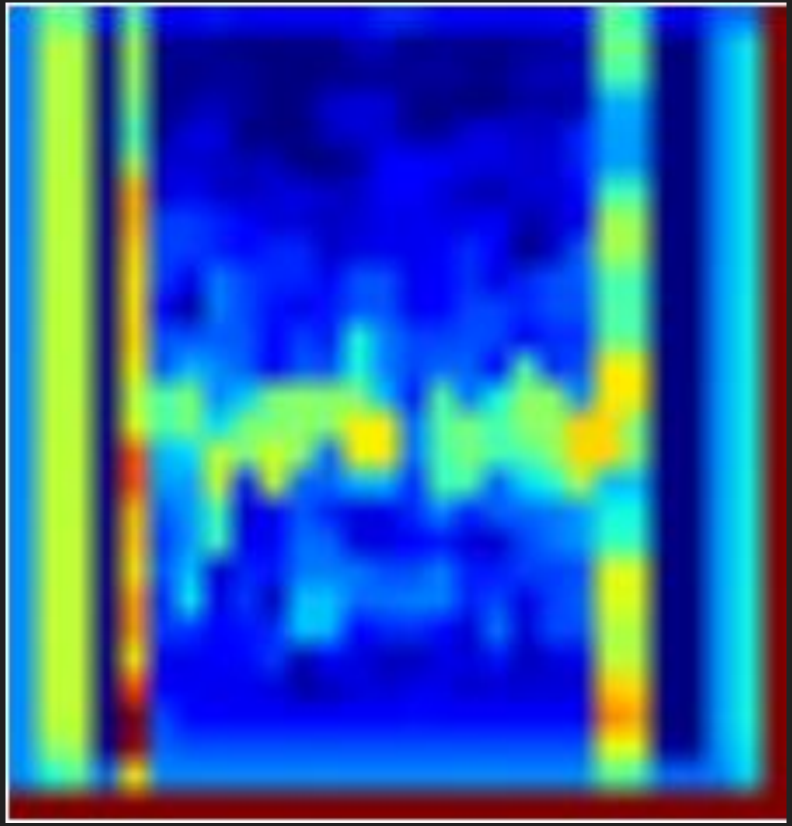
Good



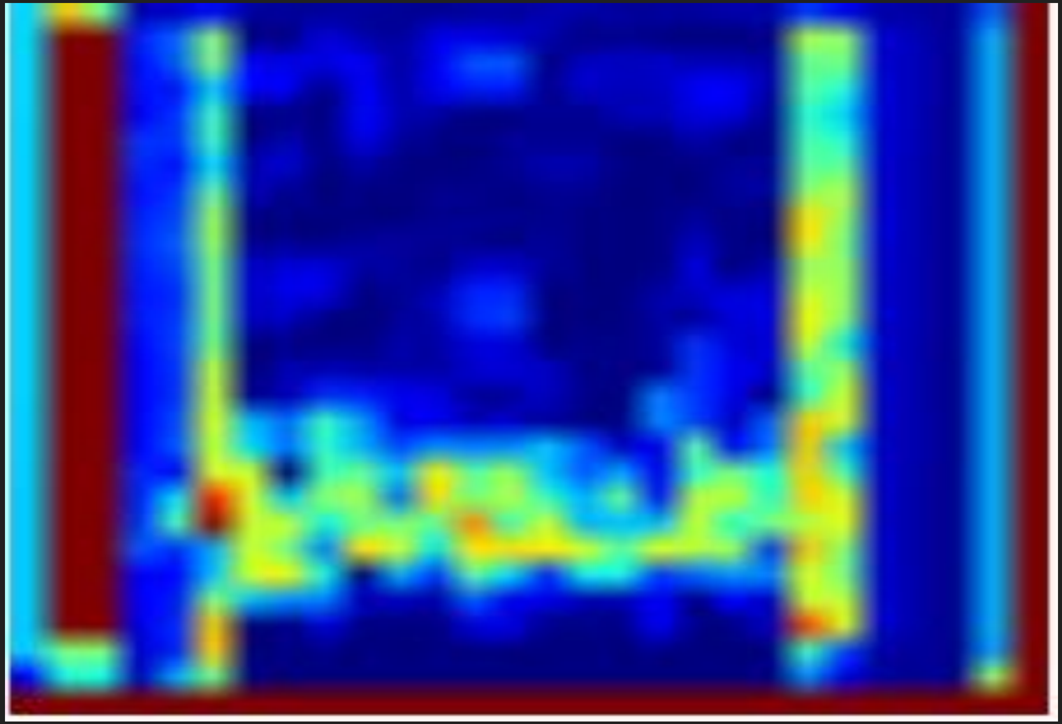


Good

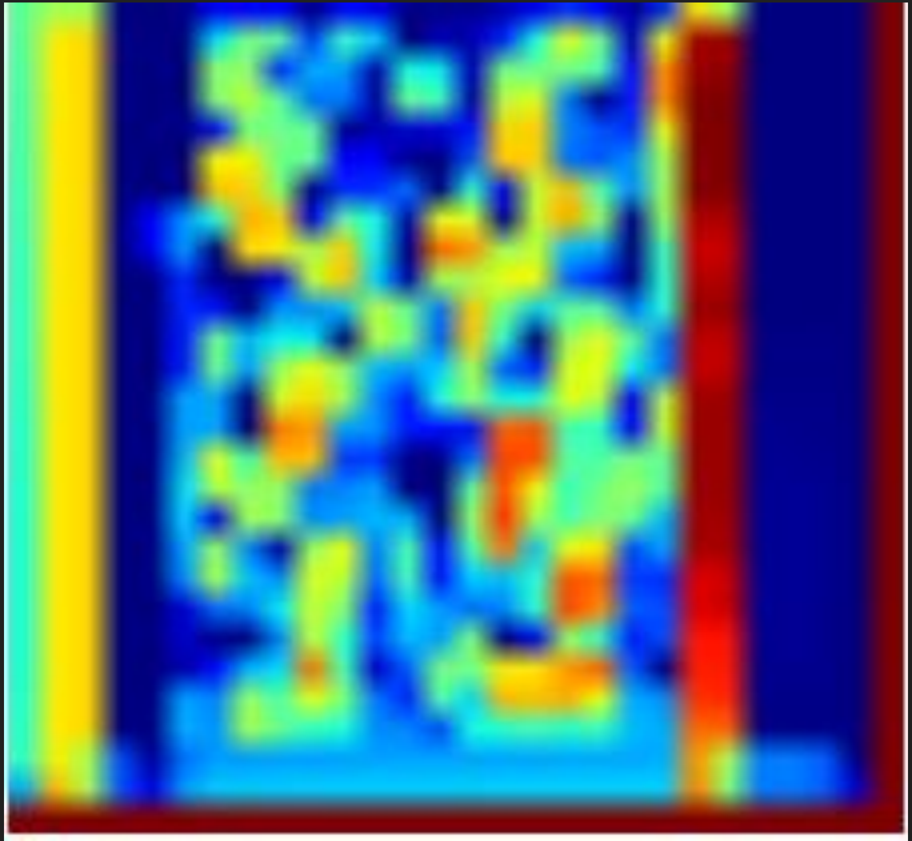




Good



Bad



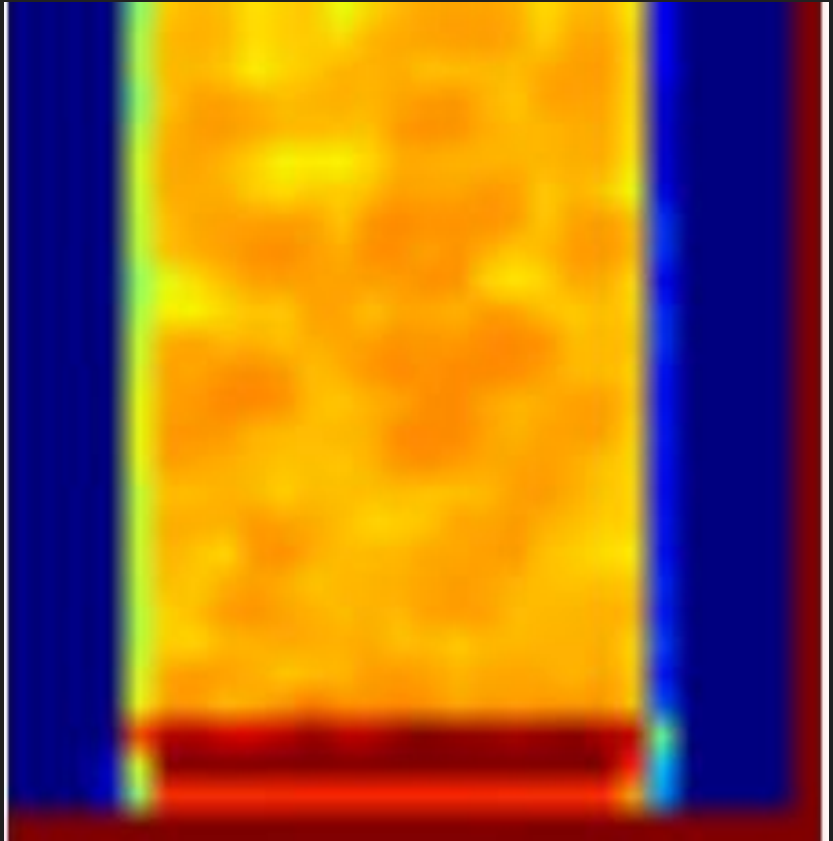


Table of Contents

Conclusion

-Successfully created Deep Learning Pipeline to predict alcohol consumption with promising results

-Configurable System with user-entered parameters

-Proof of concept for CNN and body sensor data

-Just the beginning of this area, more experiments will take place to find out more

Future Work

- Use knowledge of feature vectors to zoom into spectrograms and extract the most features possible while avoiding noise
- Apply more caffenet structures such as Googlenet
- Expand spectrogram configuration to zoom in on frequencies and change scaling

Questions?