CPE 440 Project
Classification of QAM Signals

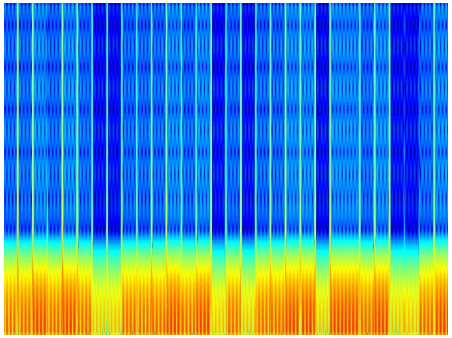
Justin Thompson
I pledge my honor that I have abided by the Stevens Honor System

https://github.com/justinbthompson/myCPE440project

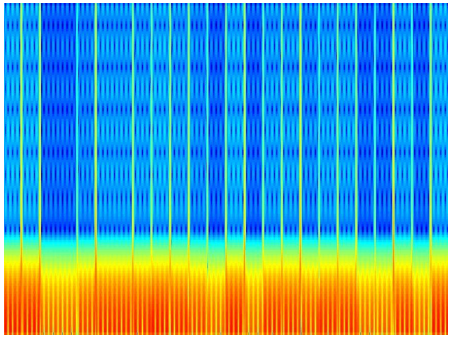
3 AI Models, One Goal

- Machine Learning SVM
- Deep Learning CNN
- Combination of CNN and SVM

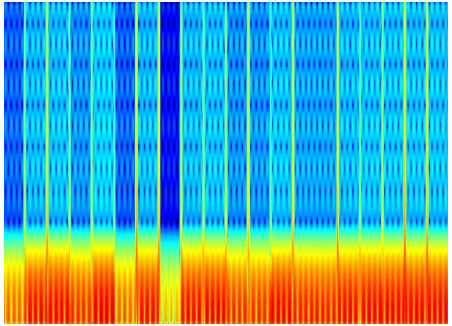
16QAM



32QAM



64QAM



- Goal is to classify 16QAM, 32QAM, or 64QAM with AI run on MATLAB.
- 900 sample images are supplied by Prof. Yao

Support Vector Machine (SVM)

	PREDICTED		
KNOWN	16QAM	32QAM	64QAM
16QAM	1.00	0.00	0.00
32QAM	0.00	1.00	0.00
64QAM	0.00	0.00	1.00

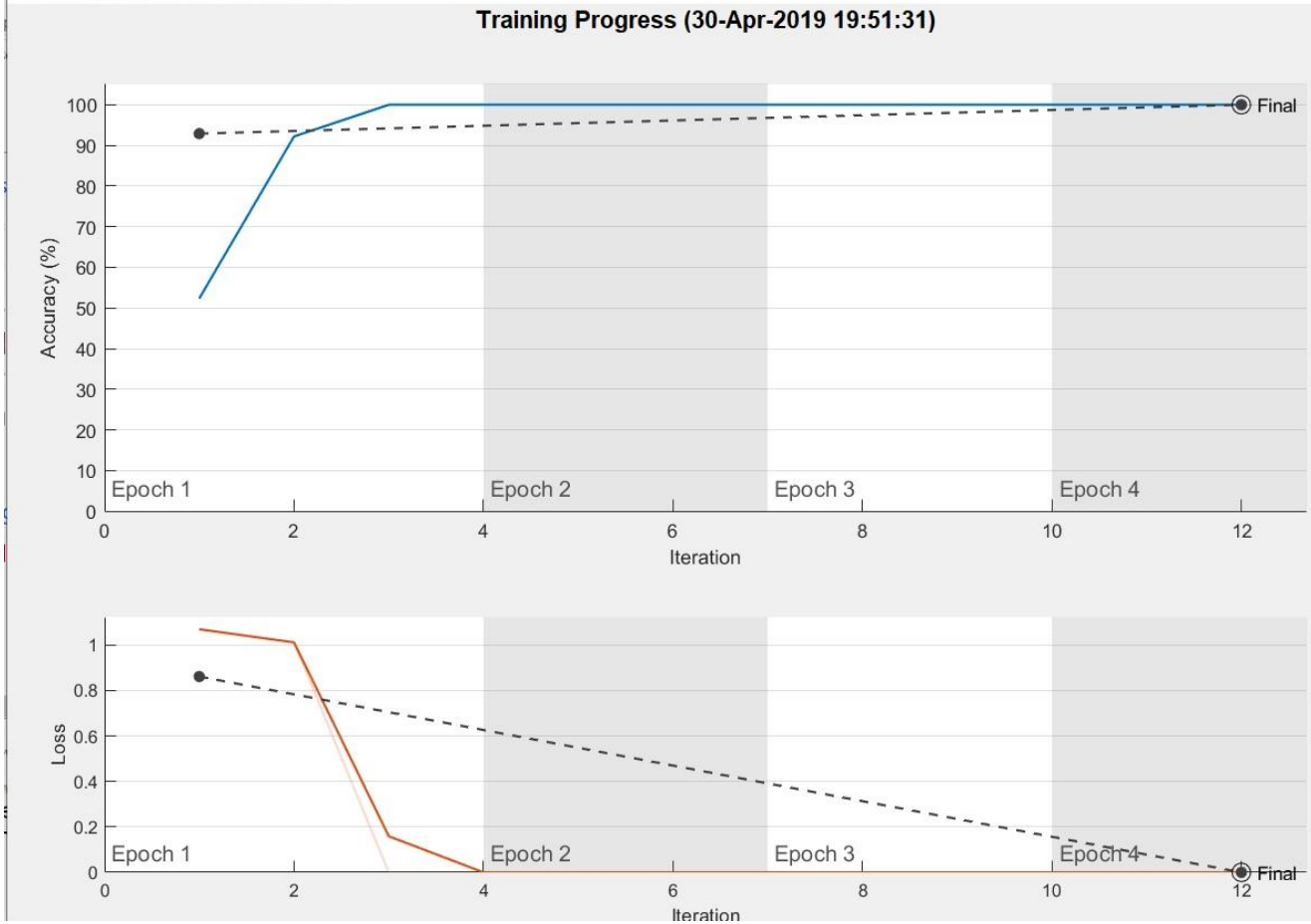
* Average Accuracy is 1.00.

- Supervised machine learning.
- Accurate, but slow and computationally expensive.
- My application based on MathWorks tutorial.
- QAMSVM_Trainer.m script uses MATLAB's bagOfFeatures to make criteria for the images.
- Took over 26 minutes to run.
- Is 100% accurate.
- View "QAMSVM_Predictor_Results.pdf" to see accurate classification of six new image files.

Convolutional Neural Network (CNN)

- Deep learning method, tries to simulate neural network of human mind.
- My application based on MathWorks tutorial.
- Training progress plot showed the mini-batch loss and accuracy and the validation loss and accuracy.
- QAMCNN_Trainer.m script defines the CNN architecture, sets options, and trains the AI.
- Took approximately 9 minutes to run.
- Is 100% accurate.
- View “QAMCNN_Predictor_Results.pdf” to see accurate classification of six new image files.

Training Progress (30-Apr-2019 19:51:31)



Results

Validation accuracy:	100.00%
Training finished:	Reached final iteration

Training Time

Start time:	30-Apr-2019 19:51:31
Elapsed time:	7 min 19 sec

Training Cycle

Epoch:	4 of 4
Iteration:	12 of 12
Iterations per epoch:	3
Maximum iterations:	12

Validation

Frequency:	30 iterations
Patience:	Inf

Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.01

Accuracy

Training (smoothed)
Training
Validation

Loss

Training (smoothed)
Training
Validation

```
confMat =

    1.0000         0         0
    0.0067    0.9733    0.0200
         0         0    1.0000


ans =

    0.9911
```

CNN and SVM Combination Model

- In this model, the CNN extracts the features of the pictures and the SVM classifies them.
- It allows the a less computationally expensive (and therefore quicker) SVM, but it reduces accuracy.
- Idea and application based on MathWorks tutorial.
- QAMCNNSVMCombo_Trainer.m script trains the AI.
- Took a mere 76 seconds to run.
- However, only 99.11% accurate.
- Still correctly classified six new image files, which can be seen in “QAMCNNSVMCombo_Predictor_Results.pdf” .

Results summary

- The CNN, SVM, and CNNSVMCombo models all accurately classified the six new signals after they were trained.
- For total accuracy using evaluation tools, the CNN model and the SVM model both reported accuracies of 100%. The CNNSVMCombo, however, only reported an accuracy of 99.11%.
- As a trade off for the reduced accuracy, the CNNSVMCombo was the quickest AI to train, taking only 76.61 seconds. The strictly CNN model was the next quickest, taking 9 minutes and 3 seconds. The SVM model took a large amount of time to train, taking 26.4 minutes while running on a powerful laptop.
- Given the large training time of the SVM model and the imperfect accuracy of the CNNSVMCombo model, I would recommend the CNN model as the best solution to classify QAM signals.