# Spectrum Sensing using Deep Learning

## Methodology & Results

### 1. Introduction

Objective:  
– Classify radio signal modulations (BPSK, QPSK, QAM16, AM-DSB, etc.) under various noise conditions using deep learning.  
Domain:  
– Automatic Modulation Classification (AMC) for cognitive radio systems in wireless communications.

### 2. Data Acquisition & Preprocessing

Dataset:  
– RadioML 2016.04C (~2GB).  
– Contains I/Q samples (size: 2×128 per sample), labels (modulation type, SNR).  
  
Preprocessing Steps:  
1. Unpack and load .pkl data.  
2. One-hot encode modulation labels.  
3. Shuffle and split into training (70%) and testing (30%).  
4. Save processed arrays in NumPy format.

### 3. CNN Model Architecture

Input Layer:  
- Reshapes 2×128 signals to 2×128×1.  
  
Feature Extraction:  
- Multiple Conv2D layers (ReLU) to capture temporal and amplitude patterns.  
- SeparableConv2D layers to reduce parameters.  
  
Regularization:  
- Dropout layers throughout.  
- L2 weight decay.  
  
Classification Block:  
- Flatten → Dense (ReLU) → Softmax output across modulation classes.  
  
Training Configuration:  
- Optimizer: Adam  
- Loss: Categorical cross-entropy  
- Validation split + Early stopping

### 4. Evaluation & Results

Accuracy vs. SNR

|  |  |
| --- | --- |
| SNR (dB) | Accuracy (%) |
| -20 | 22.3% |
| -10 | 46.5% |
| 0 | 64.9% |
| 10 | 88.7% |
| 20 | 96.3% |

Key Insight:  
Low SNRs show moderate classification; performance improves sharply at mid‐to‐high SNR.

#### Confusion Matrix Observations

At –20 dB:  
– Frequent confusion between QAM16/QAM64 and AM/FM.  
  
At ≥10 dB:  
– Clear class separation; strong diagonal entries and above 90% classification accuracy.

### 5. Conclusions

– Deep CNN demonstrates robust AMC performance across severe noise conditions.  
– High accuracy (up to 96%) at high SNR, with graceful degradation in noisy channels.  
– Valuable applications: cognitive radios, SDR platforms, wireless IoT deployments.  
  
Future Enhancements:  
– Transfer learning to adapt across SDR environments  
– Test robustness via adversarial/noisy attacks  
– Real-time implementation with GNU Radio or HackRF

### Usage & Collaboration

– Dataset access: RadioML 2016.04C (request via email: blb8.dev@gmail.com)  
– Models included: model\_d1.wts.h5 – model\_d4.wts.h5  
– Next Steps:  
 • Evaluate using your own test signals  
 • Integrate into SDR hardware  
 • Expand to additional modulation types