

# DL Project Assignment

Submitted by: Lekhraj Kumar (15MI449)

Divyansh Dhiman (15MI451)

**Title:** Modulation Recognition Using Deep Learning

**Objective:** The objective of this project is to classify various modulation techniques using Deep Learning algorithms.

**Methodology:**

- Dataset is generated using the below matlab code which gives a .mat file of available SNR values of different modulation techniques. (.mat) file is then converted to .csv file to see the generated values and then the dataset is uploaded to Google drive.
- Importing the required libraries as well as dataset from google drive.
- Reshaping and Labelling of different datasets according to respective modulation techniques.
- Splitting dataset into training set and test set in the ratio of 60:40.
- Apply Neural Networks to train the model and get the result.

**Dataset Generation:** The dataset is generated using MATLAB in which various modulation techniques is divided into 11 classes according to their amplitude and SNR values. The 11 classes are 2ASK, 4ASK, 8ASK, 2FSK, 4FSK, 8FSK, BPSK, QPSK, 8PSK, 16QAM and 64QAM.

**Data Generation Matlab Code:**

```
close all;
nsamples = 10000;
lensample = 1024;

fb=1e6;
fs=64e6;
fc=2e6;
Ns=lensample/(fs/fb);

mods = {'2ASK', '2FSK', 'BPSK', '4ASK', '4FSK', 'QPSK', '8ASK', '8FSK',
'8PSK', '16QAM', '64QAM' };
snrs = [ -20, -18, -16, -14, -12, -10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10, 12,
14, 16, 18];

chan = rayleighchan(1/fs,100,[0,1e-8],[0,-10]);
```

```

modData = zeros([(Ns+1)*fs/fb,2]);
wData = cell(1, nsamples);
for i=1:length(mods)
    mod=mods(i);
    mod=mod{1};
    disp(mod);
    for times = 1:nsamples/length(snr)
        for j=1:length(snr)
            snr=snr(j);
            start=randi([1,fs/fb]);
            start=1;
            modData(:, 1) = generate_signal(mod, snr, Ns+1, fb, fc, fs);
            modData(:, 2) = snr;
            wData{1, (times-1)*length(snr)+j} =
modData(start:(fs/fb*Ns+start-1),:);
        end
    end
    end
    fileName = ['C:\Users\LEKH RAJ\Desktop' , mod];
    save(fileName, 'wData');end

```

### Network Layers:

```

model.add(Reshape((1,2,128), input_shape=(2,128)))
model.add(Conv2D(filters=64, kernel_size=(1,3), data_format='channels_firs
t', activation='relu', padding='valid'))
model.add(Conv2D(filters=16, kernel_size=(2,3), data_format='channels_firs
t', activation='relu', padding='valid'))
#zero padding to get 128?!
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))

```

### Accuracy:

Accuracy vs SNR plot:

