

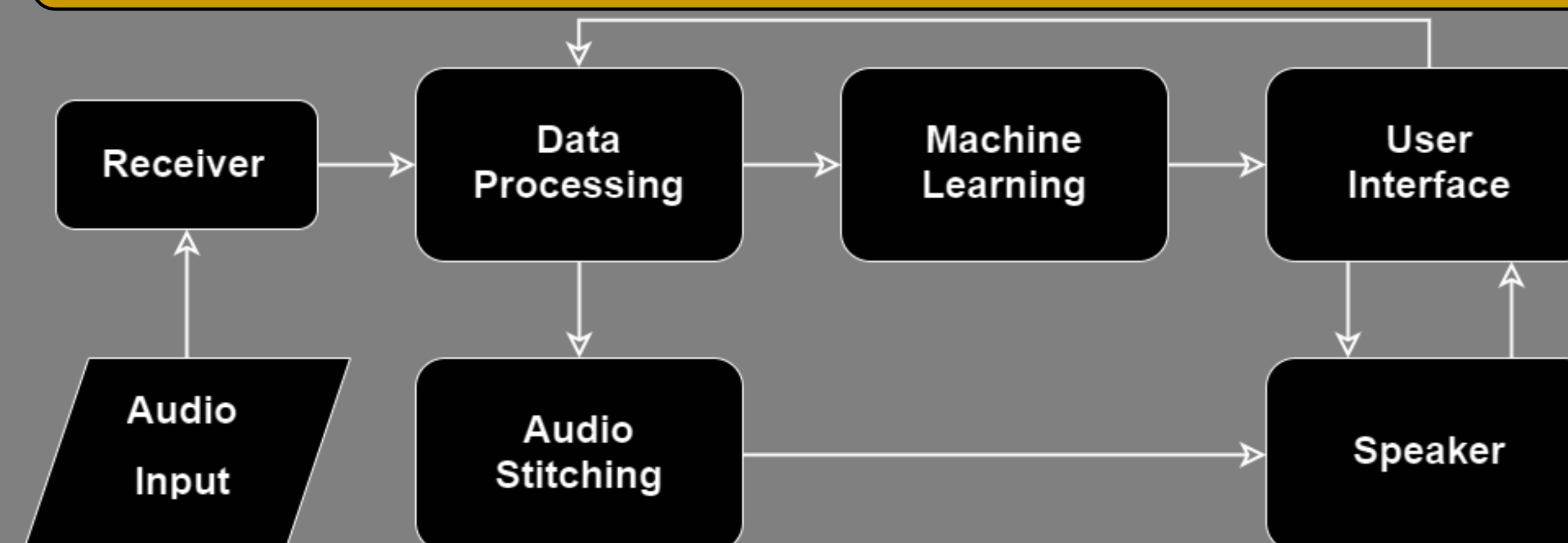
Motivation & Background

With the rise of machine learning there has been a push to develop new ways to utilize and implement it. The goal of this project was to build a system for classifying FM radio music signals into their respective musical genre through the use of Machine Learning. The best outcome would have been a real time radio interface that allowed the selection of genre and directed the appropriate station to play.

Requirements

- The system **shall** have simultaneous search and playback
- The system **shall** operate under the publicly defined frequency range for FM radio
- The system **shall** be capable of running the neural network
- The system **shall** have the capability to read input from defined FM channels

System Block Diagram

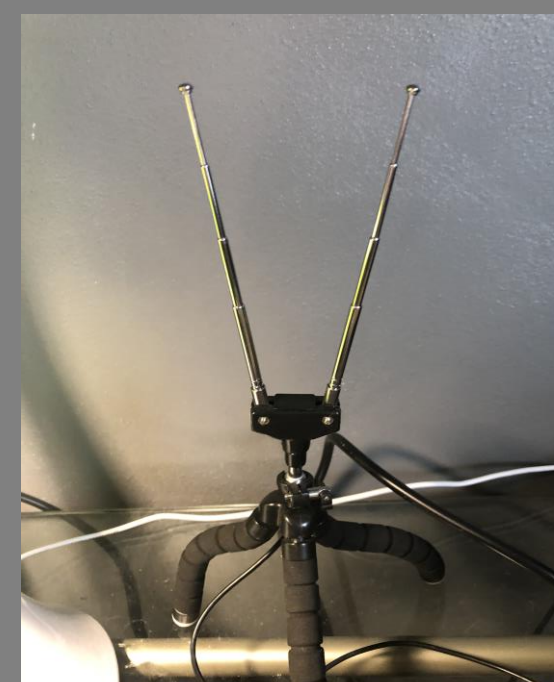


System Main Components

RTL-SRD USB Radio Receiver



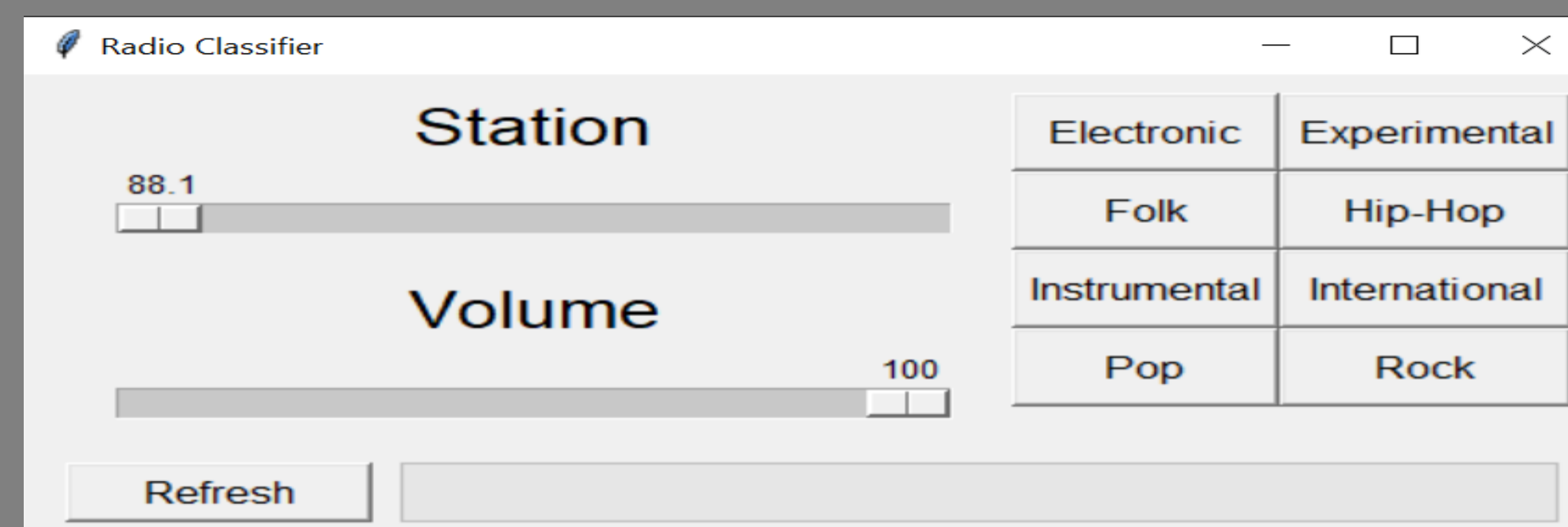
Radio Antenna



Computer with Python 3.8 installed



Project GUI



Results

- Trained with 158 batch files (3878 records each)
- Tested with 1 file (7554 records) - every epoch
- 10 epochs per batch
- 3 models

```
137 # -----
138 # Function for building a CNN model:
139
140
141 def create_model(trainX, trainY, testX, testY):
142     numTimesteps = trainX.shape[1]
143     numFeatures = trainX.shape[2]
144     numOutputs = trainY.shape[1]
145
146     inputShape = (numTimesteps, numFeatures)
147     activFunc = 'relu'
148
149     model = tf.keras.models.Sequential([
150         tf.keras.layers.Conv1D(numFilters, kernel_size, strides=strideLen,
151                                activation=activFunc, input_shape=inputShape),
152         tf.keras.layers.MaxPool1D(pool_size=poolSize),
153         tf.keras.layers.Conv1D(2*numFilters, kernel_size, strides=strideLen,
154                                activation=activFunc),
155         tf.keras.layers.MaxPool1D(pool_size=poolSize),
156         tf.keras.layers.Flatten(),
157         tf.keras.layers.Dense(100, activation=activFunc),
158         tf.keras.layers.Dense(numOutputs, activation='softmax')
159     ])
160     model.summary()
161     model.compile(loss='categorical_crossentropy',
162                  optimizer='adam', metrics=['accuracy'])
163     return model
```

- Accuracy: ~15%
- Validation Accuracy: ~12%

Future Work

- Train on actual radio data
- Introduce static to FMA
 - Mimic radio static
- Use different CNN architectures
- Longer audio samples