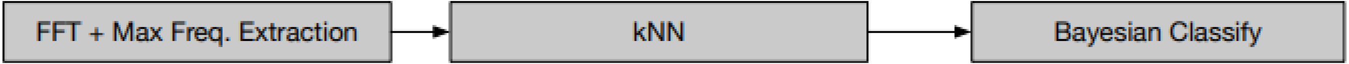

FFT + Max Freq. Extraction

kNN

Bayesian Classify

```
// Declare FFTW variables
fftw_complex *in, *out;
fftw_plan p;

// Compute FFT
size = sizeof(fftw_complex) * FFT_SIZE;
in = (fftw_complex*) fftw_malloc(size);
out = (fftw_complex*) fftw_malloc(size);
p = fftw_plan_dft_1d(FFT_SIZE, in, out,
    FFTW_FORWARD, FFTW_ESTIMATE);
fftw_execute(p);

// Compute max frequency
max_freq = max((sqrt(out[i][0]*out[i][0] +
      out[i][1]*out[i][1]));
```

```
// Compute Euclidean 1D distances
for(count = 0; count < NUM_SAMPLES; count++ ) {
  diff[count] = fabsf(diff[count] - maxfreq);
}

// Pick k Nearest Neighbours (k=8)
for(i = 0; i < K; i++) {
  for(j = 1; j < NUM_SAMPLES; j++) {
    if( diff[j] < min) {
      min = diff[j];
      min_vals[i] = diff[j];
      idx = j;
    }
  }
}
```

```
// Determine class count
if(idx <1000) {
  ++class_count[0];
} else if(idx >= 1000 && idx < 2000 ) {
  ++class_count[1];
} else {
  ++class_count[2];
}
diff[idx] = FLT_MAX;

// Determine conditional probability
for(i = 0; i < NUM_CLASS; i++) {
  prob[i] = (float)class_count[i]/K;
  if( prob[i] > max_prob )
    max_prob = prob[i];
}
```