Feature Learning from Spectrograms for Assessment of Personality Traits

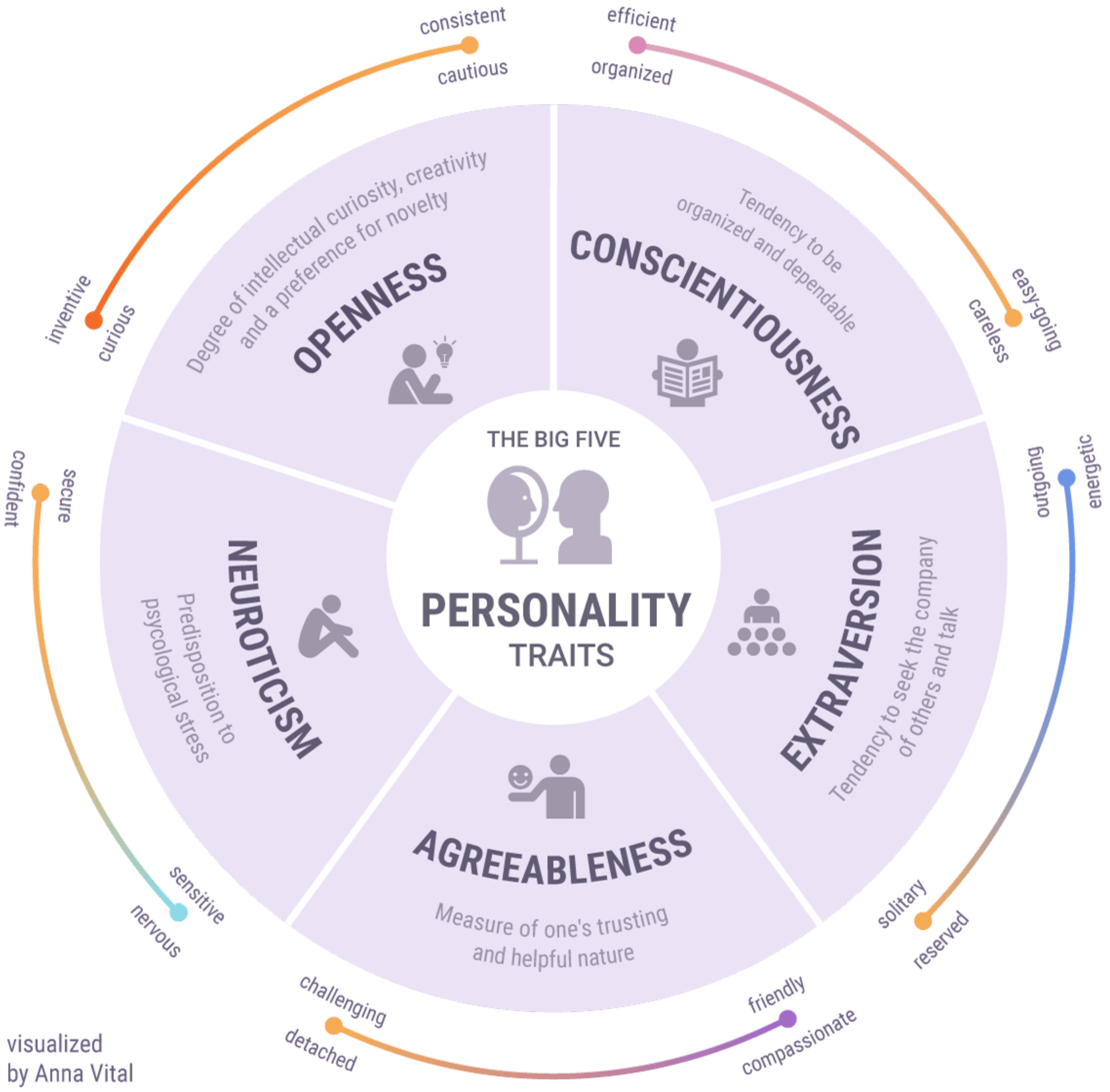
Published in: IEEE Transactions on Affective Computing (Volume: 11 , Issue: 1 , Jan.-March 1 2020)

Marc-Andre´ Carbonneau, *Member, IEEE,* Eric Granger, *Member, IEEE,* Yazid Attabi, *Member, IEEE,* and Ghyslain Gagnon, *Member, IEEE*

What is this research paper about?

To classify five personality traits (the *Big-Five*) using Speech Analysis :

1. openness
2. conscientiousness
3. extroversion
4. agreeableness
5. neuroticism



visualized
by Anna Vital

Source: J. M. Digman
Personality Structure: Emergence of the Five-Factor Model

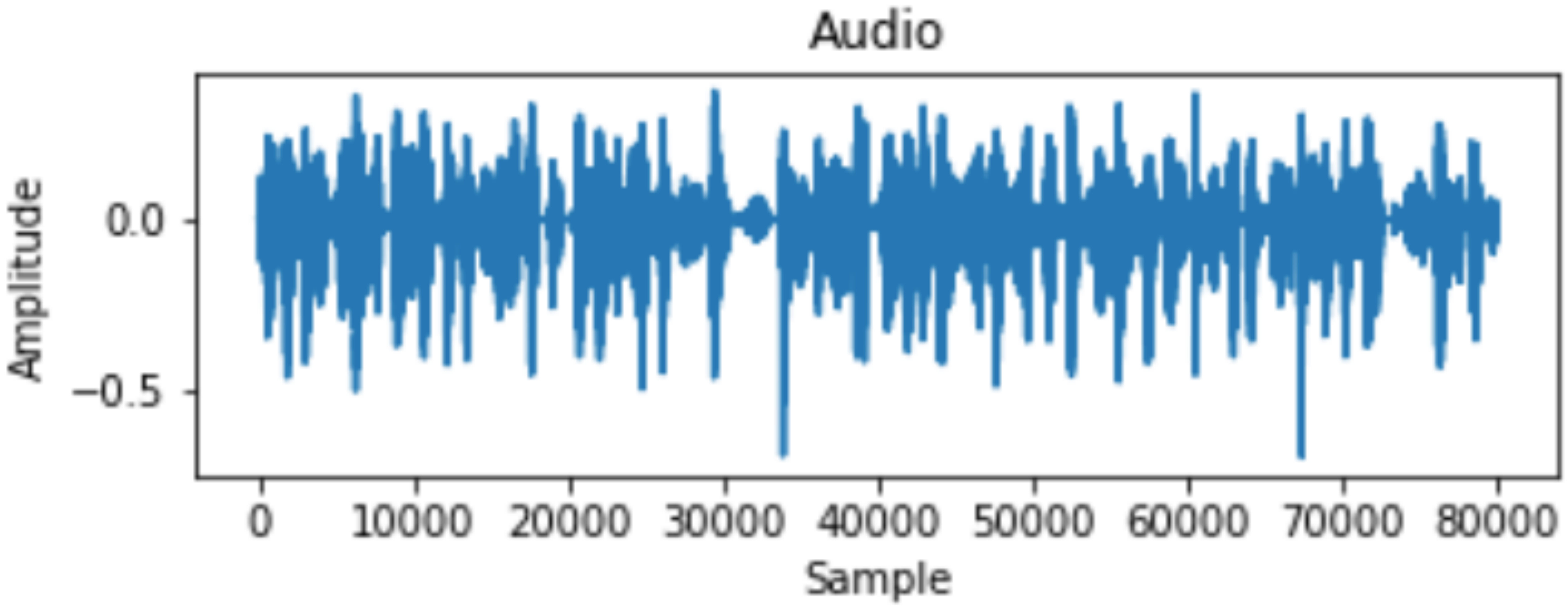
adioma

Block diagram for the prediction of a Personality Trait.

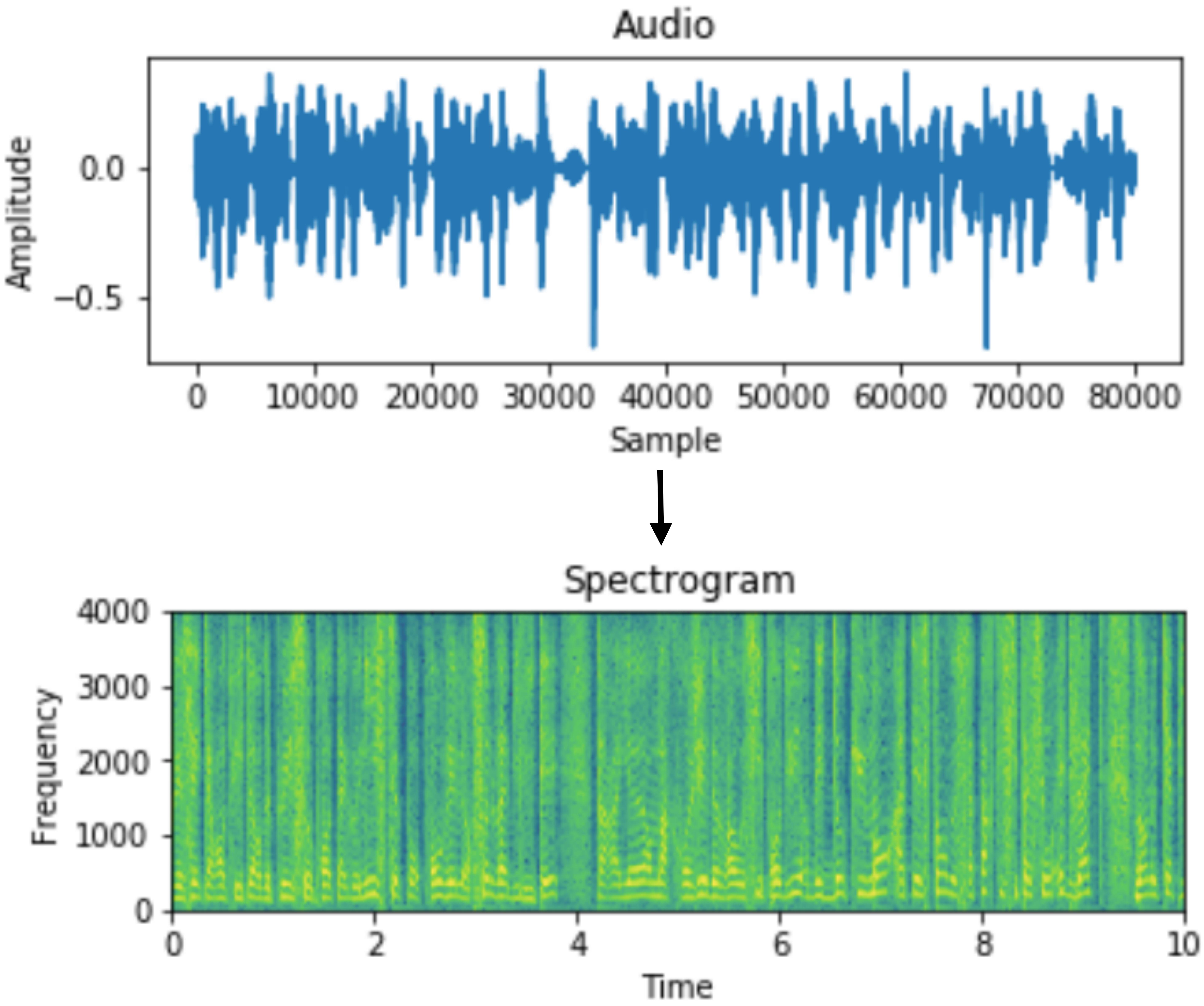
1.
Speech segments
{F1, F2, ...}
Spectrograms
{S1, S2, ... }
Patch set
{p1, ..., pk}
4.
Code-word set
{c1, ..., ck}
5
Histogram feature vectors
{h1, h2, ... }
Training Speech Segments
Spectrogram Extraction
Patch Extraction
Dictionary Learning
Pooling
Encoded Speech Segments
2.
3.
Dictionary atoms
{d1, ..., dm}
Training
Patch Dictionary
Testing
Speech segment
Fi
Spectrogram Extraction
Patch Extraction
Patch Encoding
Pooling
Classification
Label
Yi
Spectrogram
S
Patch set
{p1, ..., pN}
Code-word set
{c1, ..., cN}
Histogram feature vector
h

1. Training Dataset

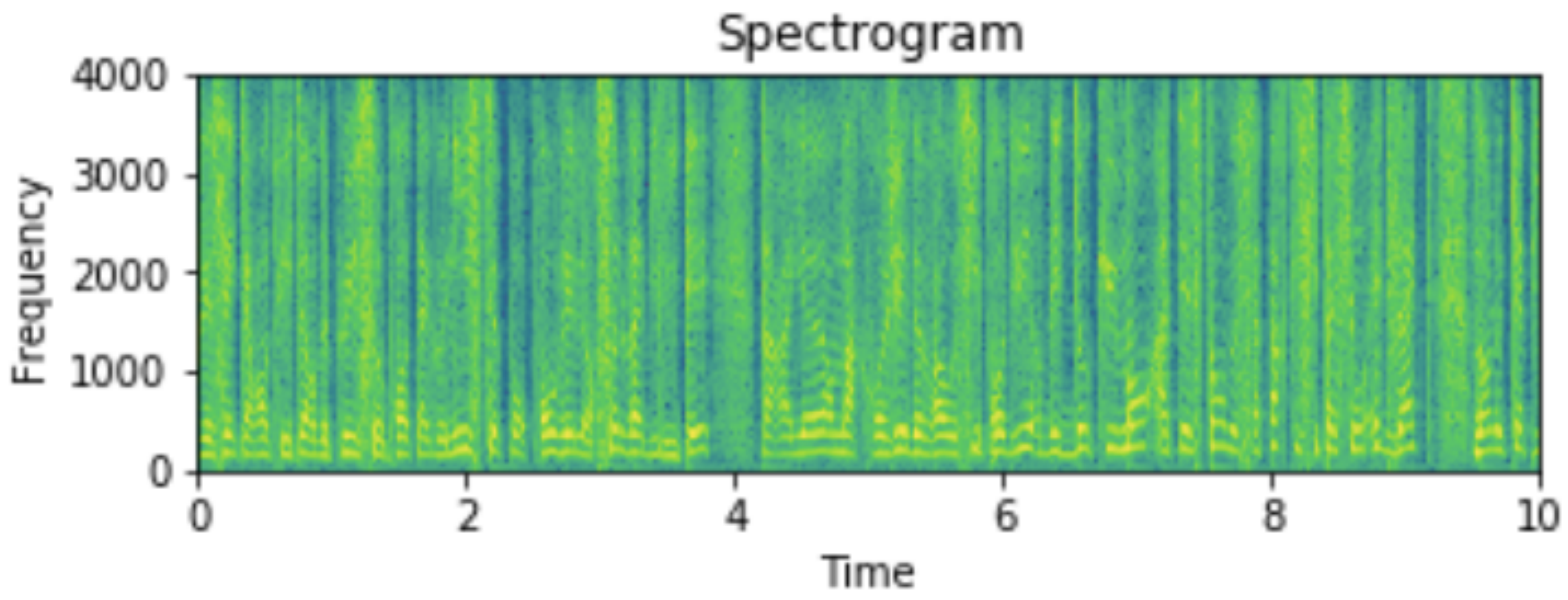
- 640 audio clips randomly extracted from French News
- Frequency of audio = 8 kHz
- Time = 10 sec
- 322 different speakers

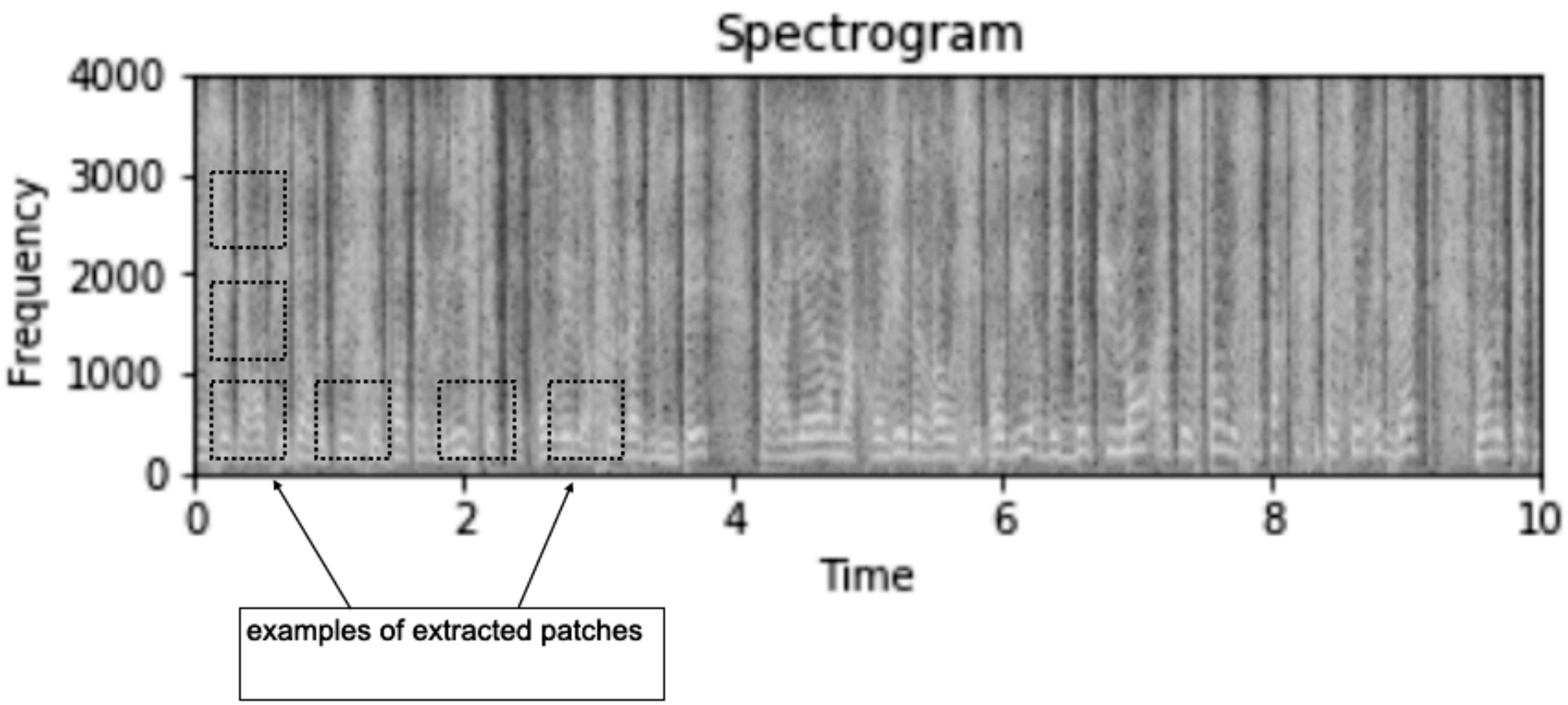


2. Spectrogram Extraction

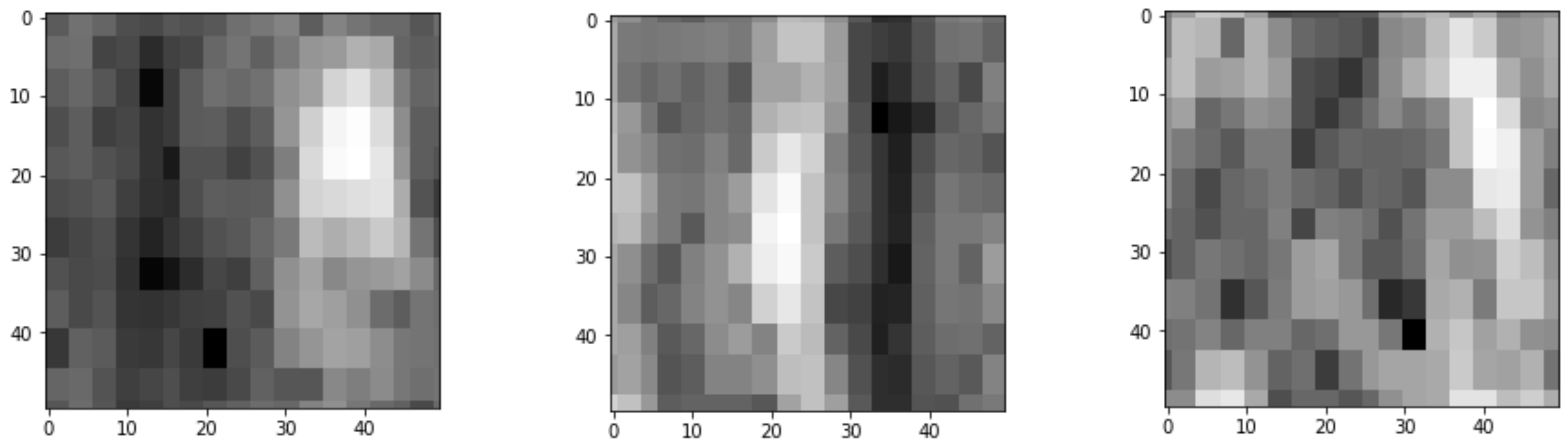


3. Patch Extraction



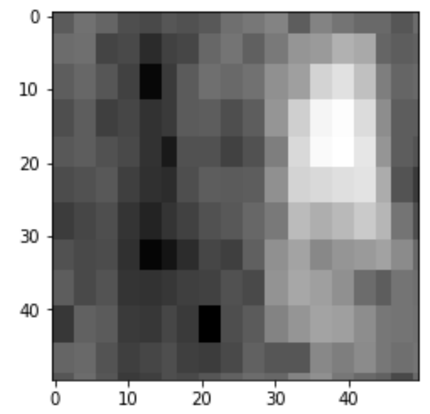


examples of extracted patches



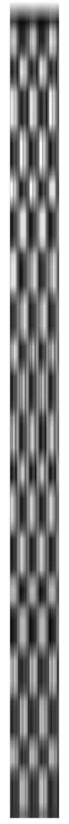
Patches are extracted at regular Interval

4. Dictionary Learning



Flatten

p x p



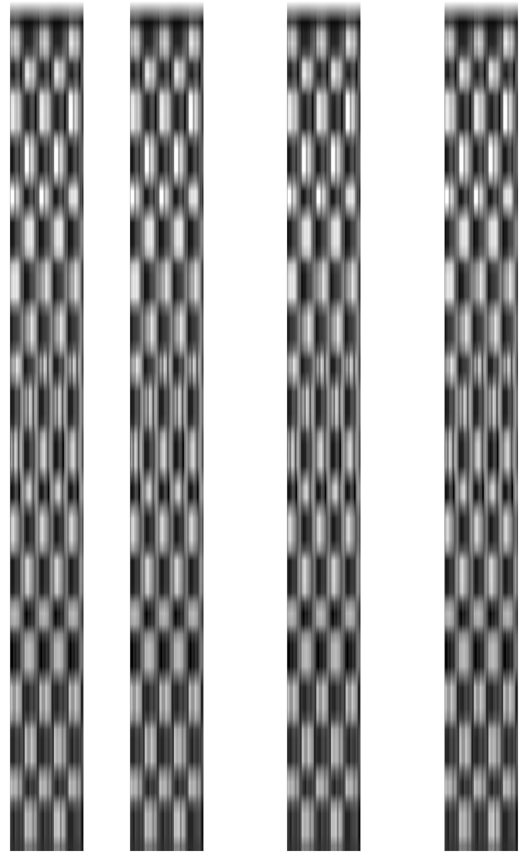
p

Dictionary Learning

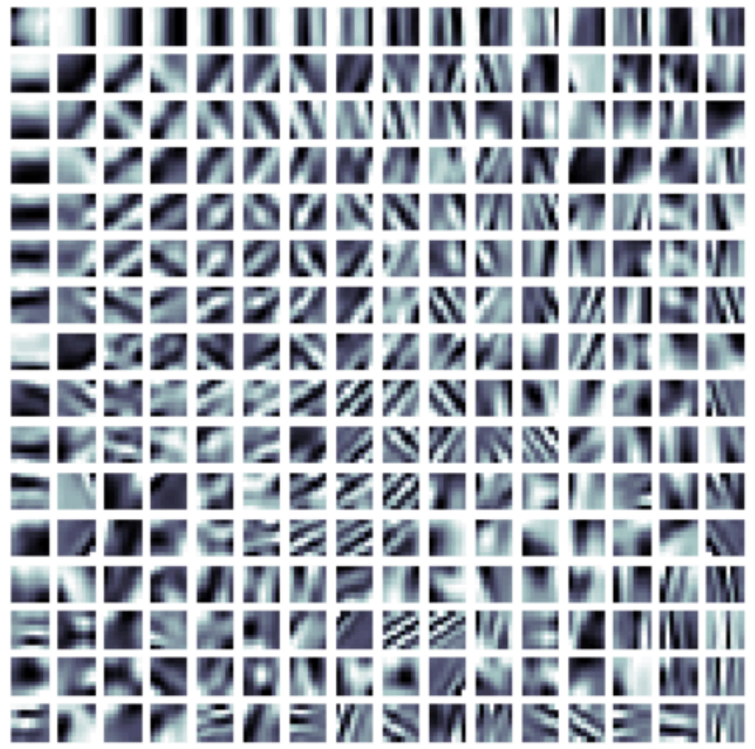
Sparse Coding

$$l(\mathbf{c}_i) \triangleq \min_{\mathbf{c}_i \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{p_i} - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_1$$

LARS-Lasso algorithm



Patch set $\{\mathbf{p}_1, \ldots, \mathbf{p}_k\}$



$[\mathbf{d}_1, \ldots, \mathbf{d}_m]$

+ Code Words $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$

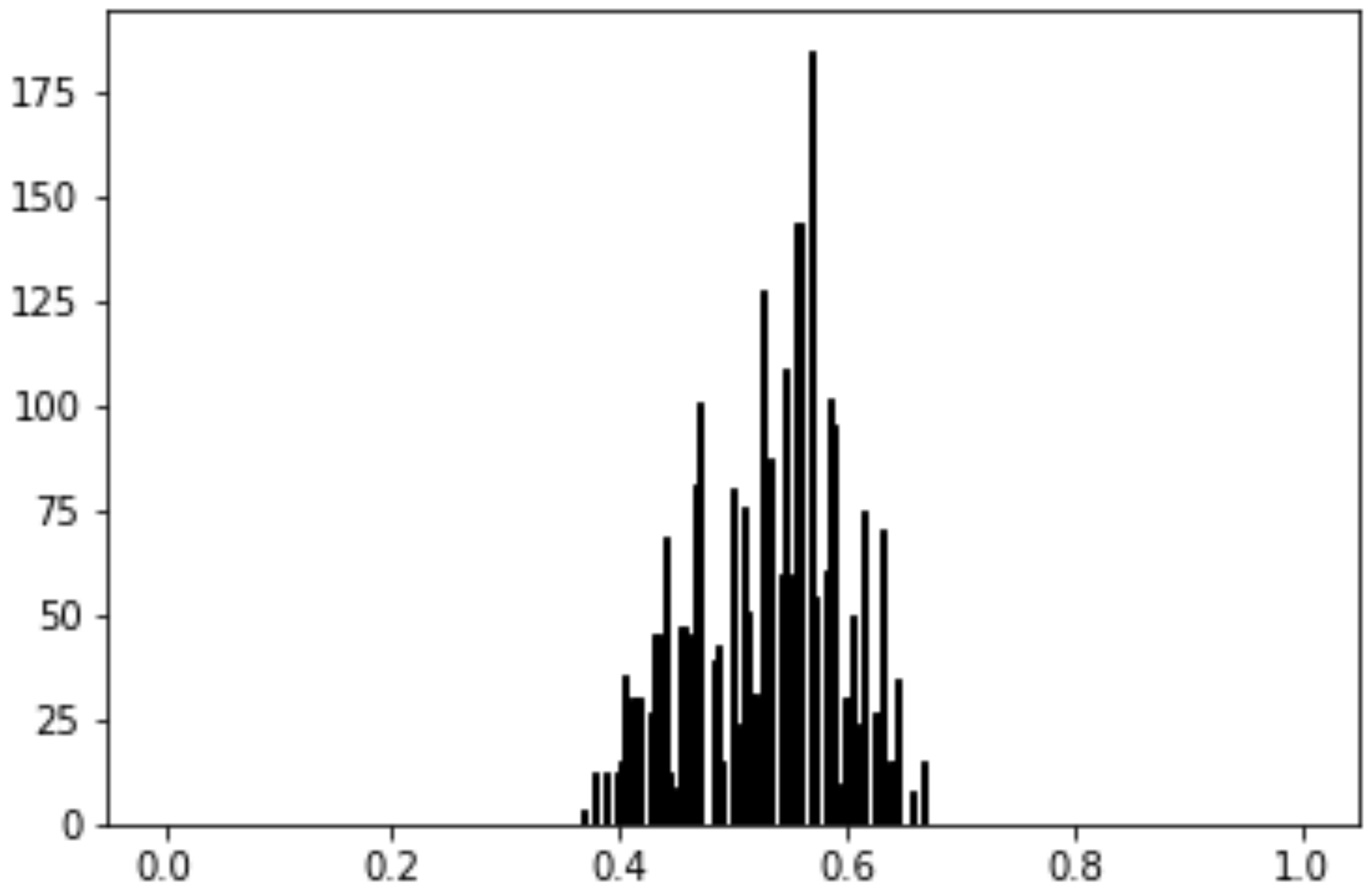
5.

Histogram feature vectors

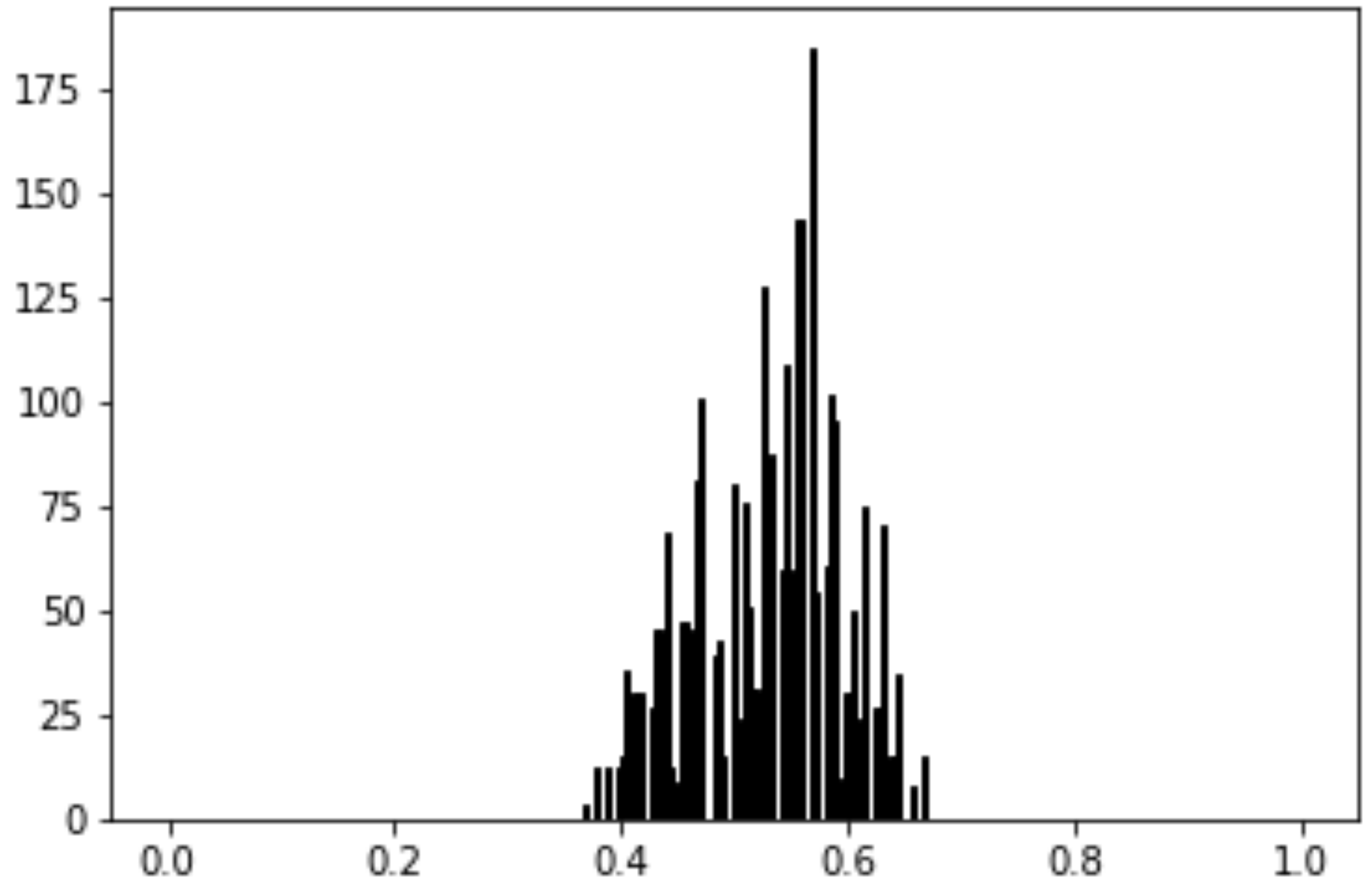
$$\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$$

Code Words

$$\mathbf{h} = \sum_i |\mathbf{c}_i|$$



6. Encoding Personality Traits and SVM

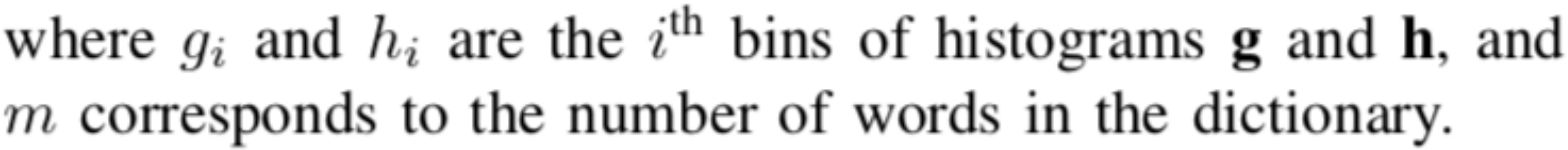


$$d(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^{m} \frac{(g_i - h_i)^2}{g_i + h_i}$$

SVM

OUTPUT

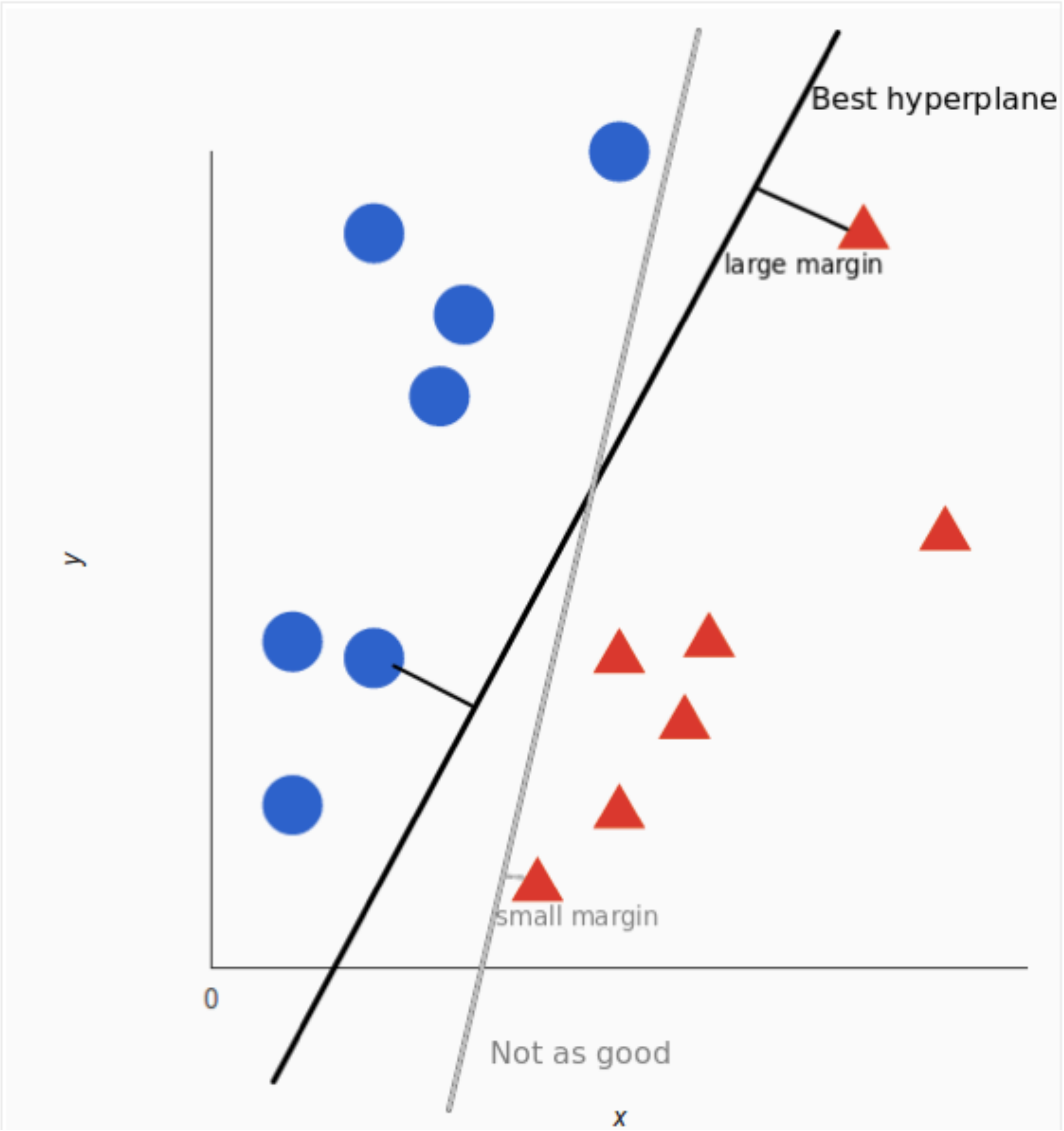
Classification



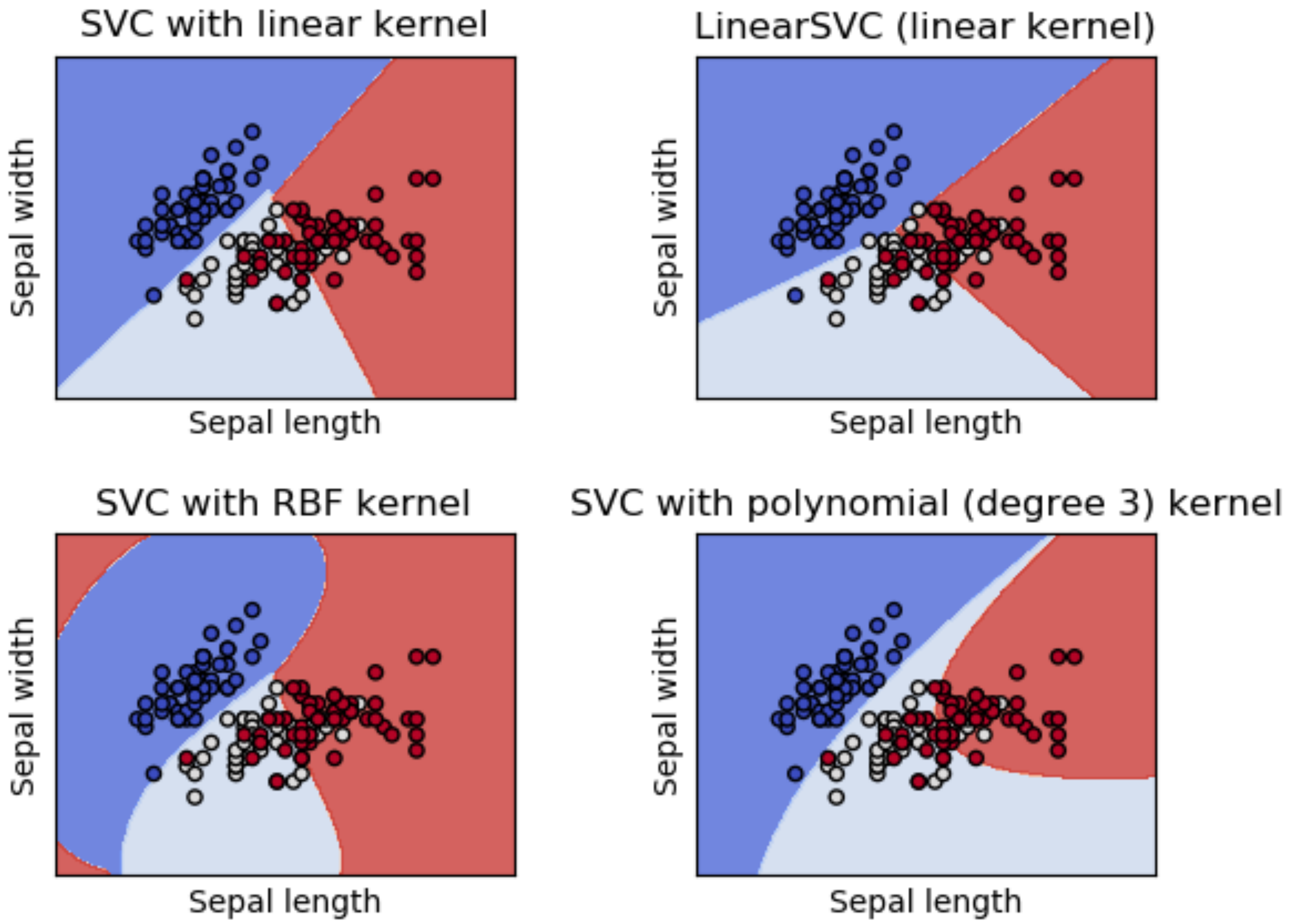
where g_i and h_i are the i^{th} bins of histograms $\mathbf{g}$ and $\mathbf{h}$, and m corresponds to the number of words in the dictionary.

How Does SVM Work?

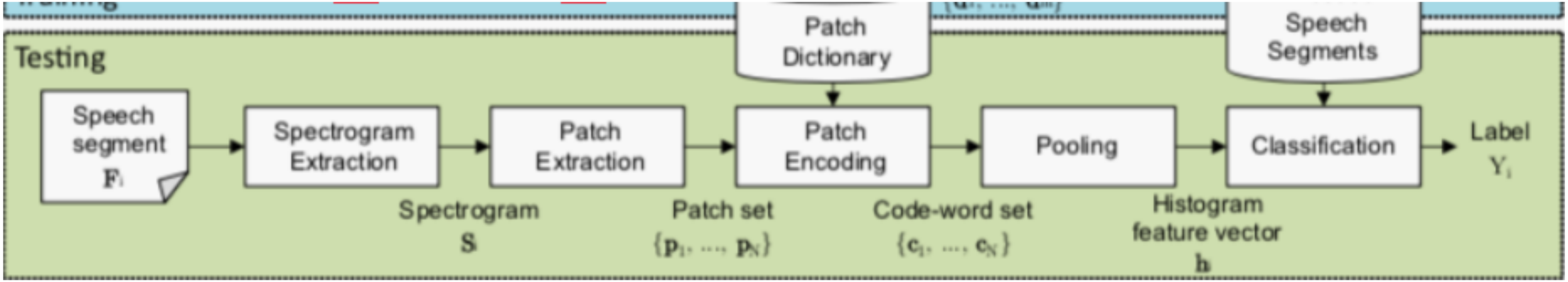
The basics of Support Vector Machines and how it works are best understood with a simple example. Letâ€™s imagine we have two tags: *red* and *blue*, and our data has two features: *x* and *y*. We want a classifier that, given a pair of *(x,y)* coordinates, outputs if is either *red* or *blue*. We plot our already labeled training data on a plane:



SVM working with different parameters



Testing the model


Testing
Patch Dictionary
Speech Segments
Speech segment $\mathbf{F}_i$
Spectrogram Extraction
Patch Extraction
Patch Encoding
Pooling
Classification
Label Y_i
Spectrogram $\mathbf{S}$
Patch set $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$
Code-word set $\{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$
Histogram feature vector $\mathbf{h}$

Thank You!

Arindam Jain

2K16/EE/27