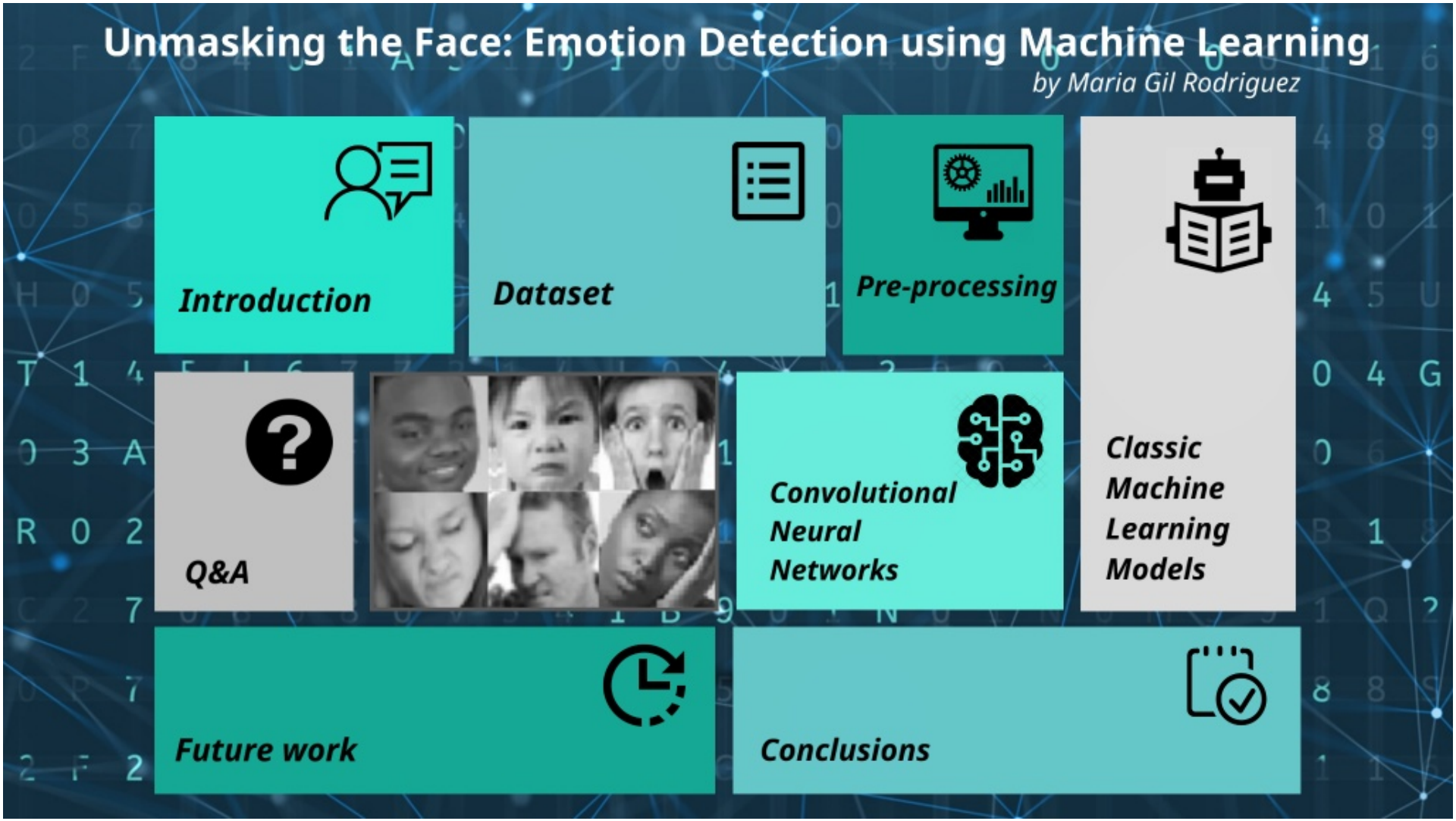
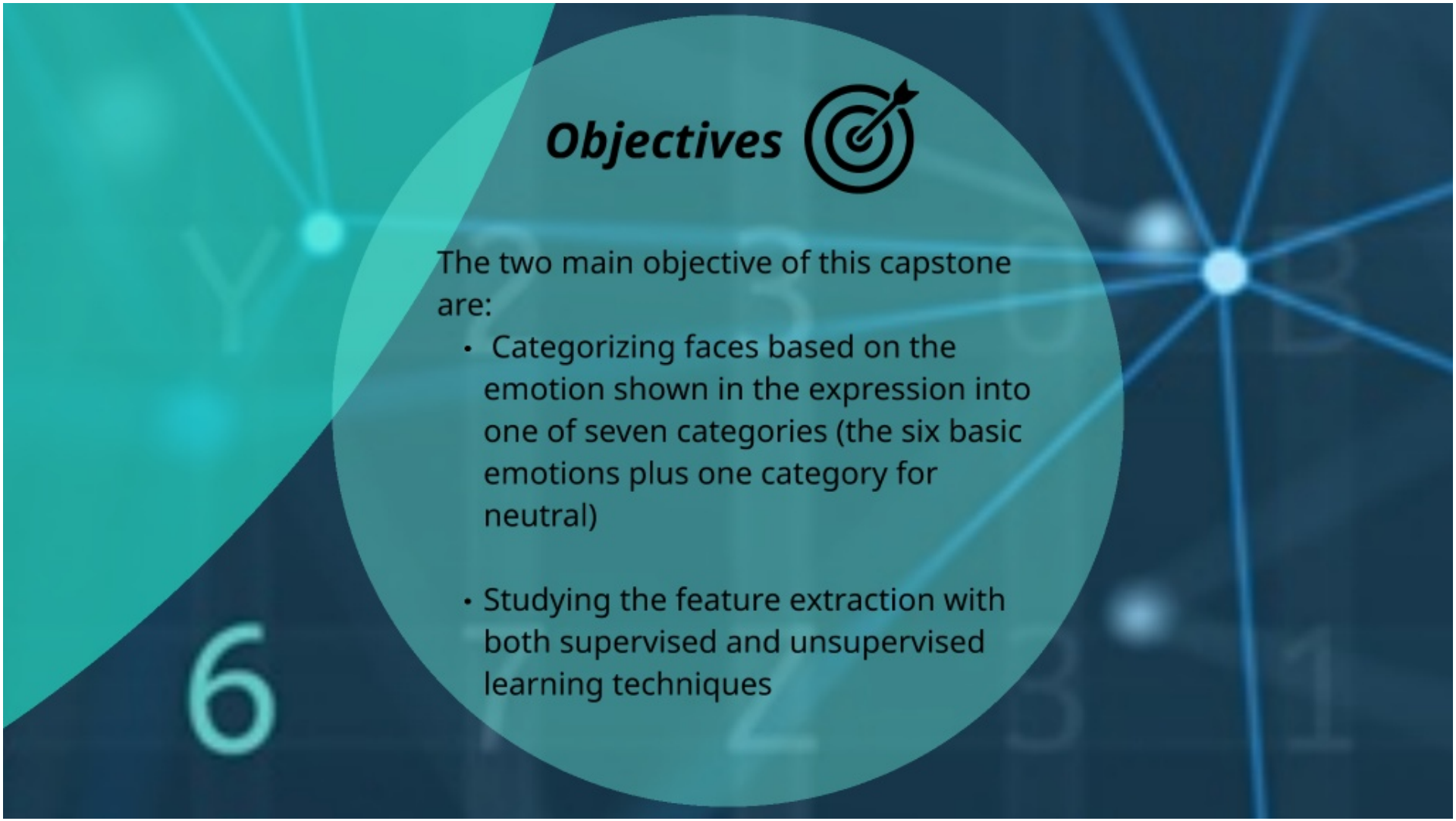

Unmasking the Face: Emotion Detection using Machine Learning
by Maria Gil Rodriguez
Introduction
Dataset
Pre-processing
Classic Machine Learning Models
Q&A
Convolutional Neural Networks
Future work
Conclusions

Introduction

- Human beings are responsible for the depiction of six basic emotions: happiness, anger, surprise, sadness, fear, and disgust.

- Creating a model that can accurately classify these emotions can be extremely useful in a variety of areas such as image processing, cybersecurity, robotics, psychological studies, virtual reality applications, etc.

Objectives

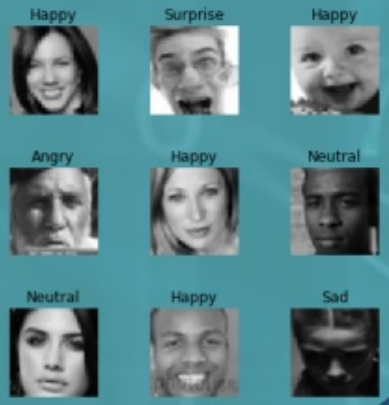

Objectives
The two main objective of this capstone are:
• Categorizing faces based on the emotion shown in the expression into one of seven categories (the six basic emotions plus one category for neutral)
• Studying the feature extraction with both supervised and unsupervised learning techniques

Dataset

- FER2013 (Kaggle Competition)
- 48x48 pixel grayscale images of faces
- 3 sets:

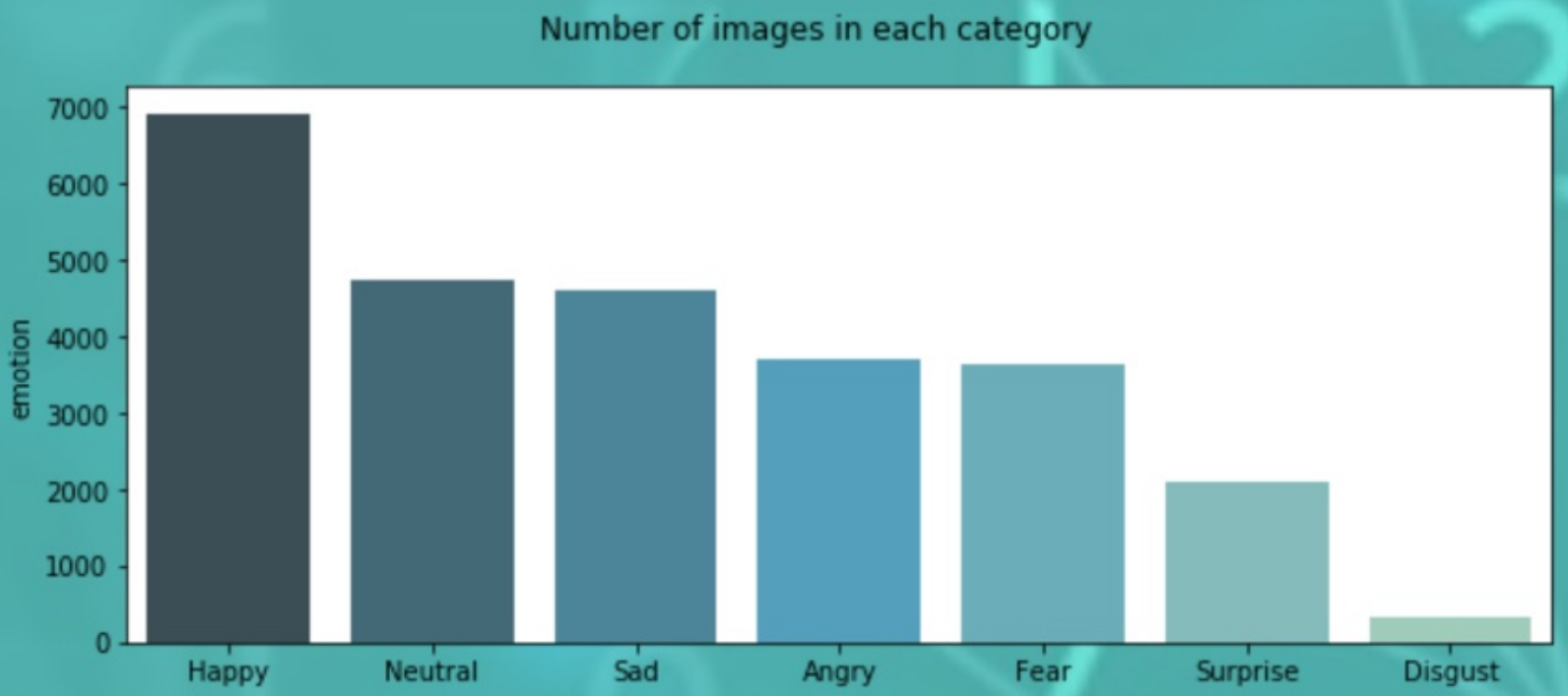
Training	22968 images
Validation	5741 images
Test (no labels)	3589 images

- Classes:
  - 0=Angry
  - 1=Disgust
  - 2=Fear
  - 3=Happy
  - 4=Sad
  - 5=Surprise
  - 6=Neutral



Class imbalance

Class imbalance



Pre-processing

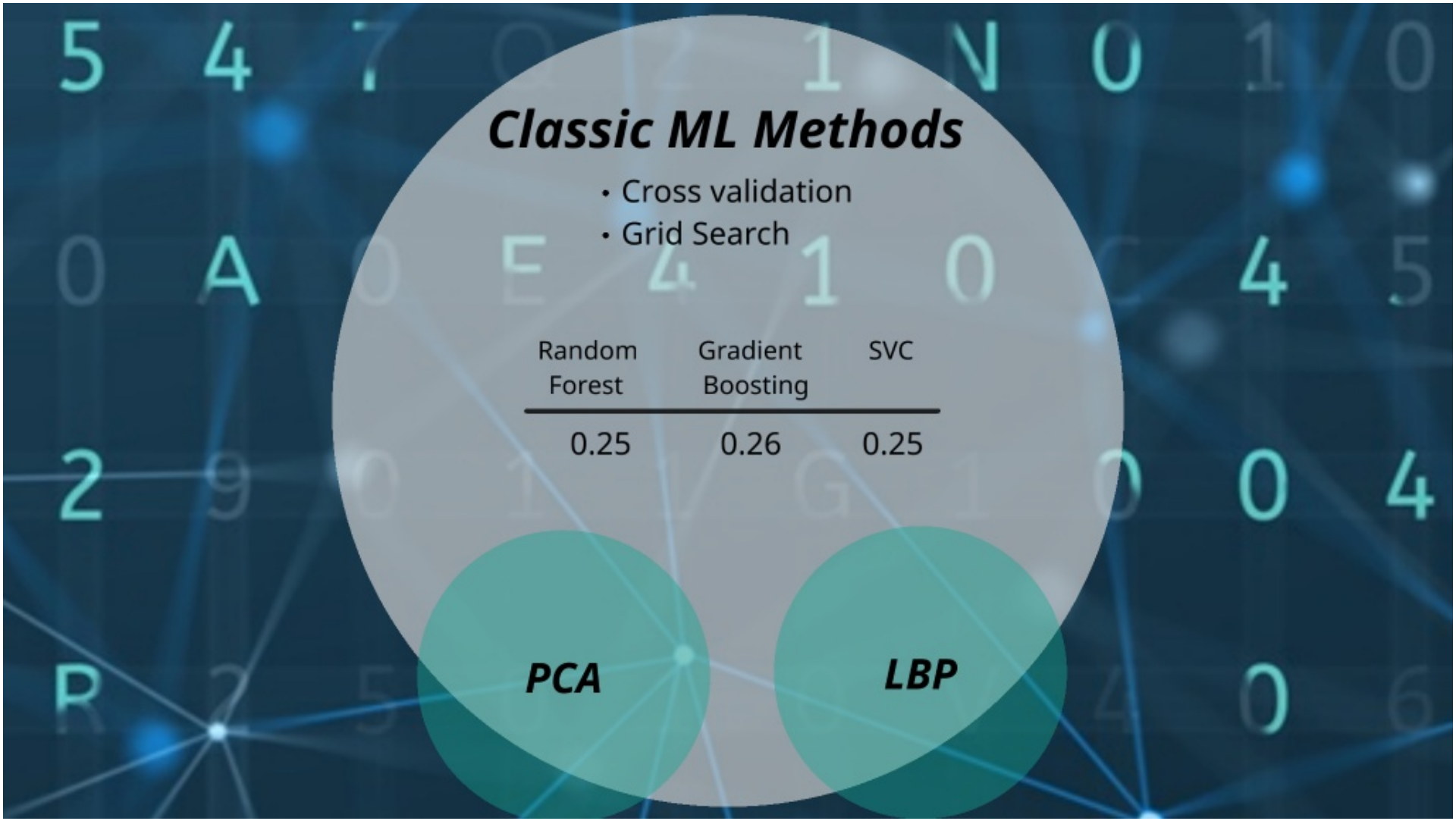
	emotion	pixels
0	0	[70.0, 80.0, 82.0, 72.0, 58.0, 58.0, 60.0, 63....
1	0	[151.0, 150.0, 147.0, 155.0, 148.0, 133.0, 111...
2	2	[231.0, 212.0, 156.0, 164.0, 174.0, 138.0, 161...
3	4	[24.0, 32.0, 36.0, 30.0, 32.0, 23.0, 19.0, 20....
4	6	[4.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...

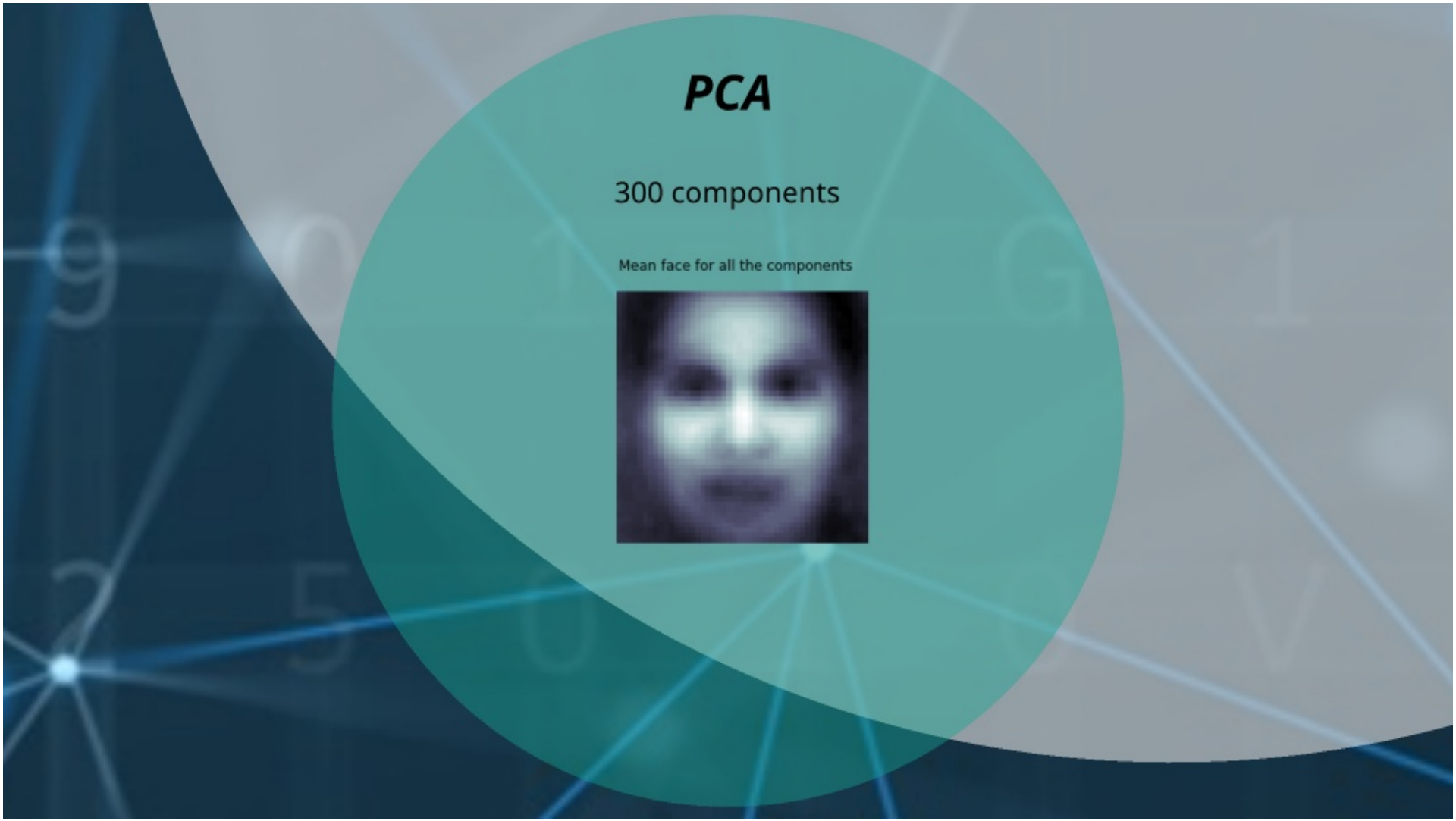
For the classic ML methods:

- Shuffle
- 100 images of each class for the training set
- 50 in the validation set

This:

- Avoids the class imbalance
- Reduces run time


Classic ML Methods
• Cross validation
• Grid Search
Random Forest
Gradient Boosting
SVC
0.25
0.26
0.25
PCA
LBP


PCA
300 components
Mean face for all the components

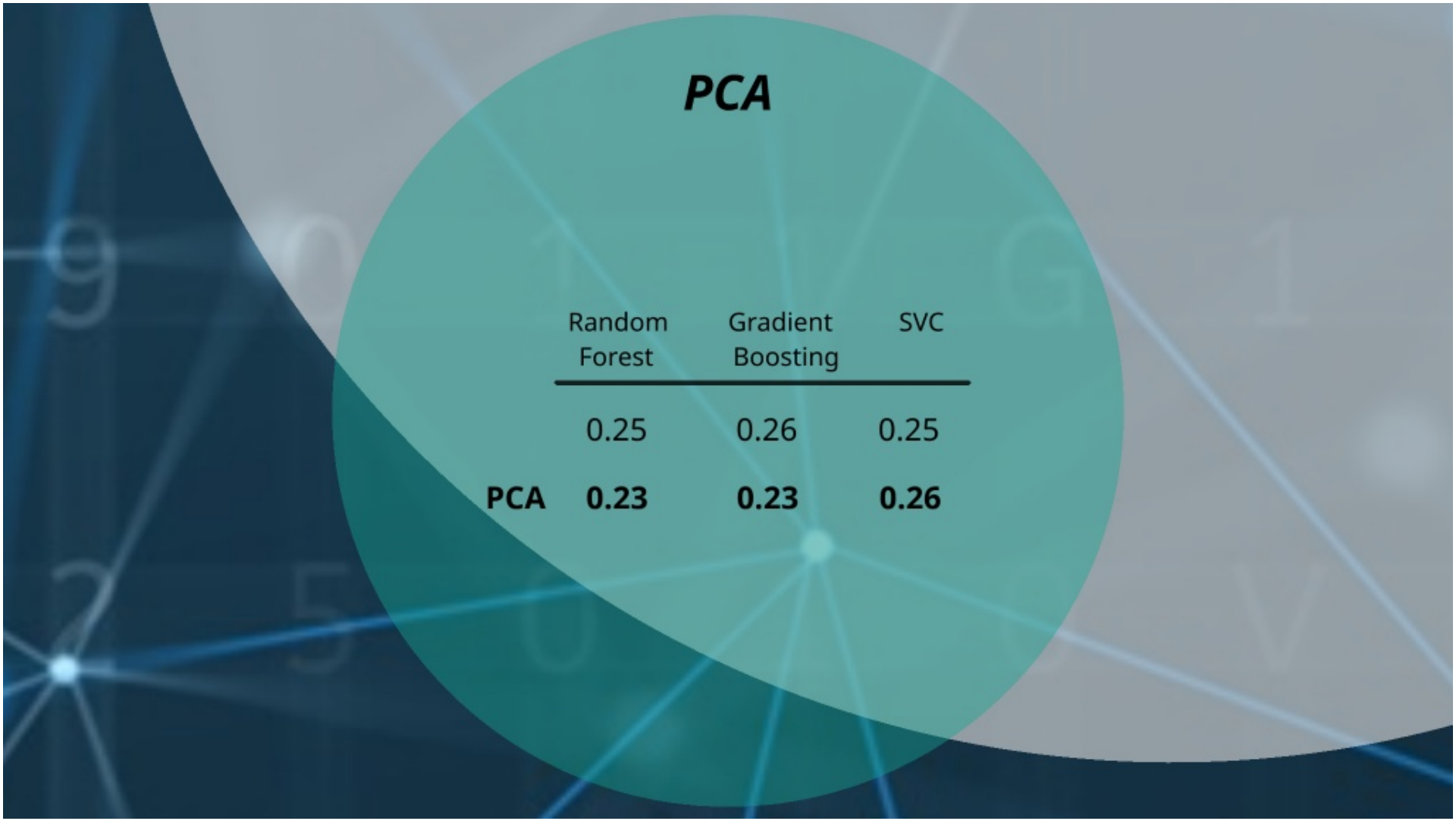
PCA

Eigen Faces

First 24 Eigen Faces

PCA

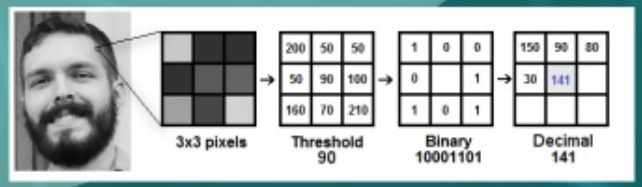

Weight = -100
Weight = -5
Weight = 5
Weight = 100
Component 1
Component 2
Component 3
Component 4
Component 5
Component 6


PCA
Random Forest
Gradient Boosting
SVC
0.25
0.26
0.25
PCA
0.23
0.23
0.26

Local Binary Patterns

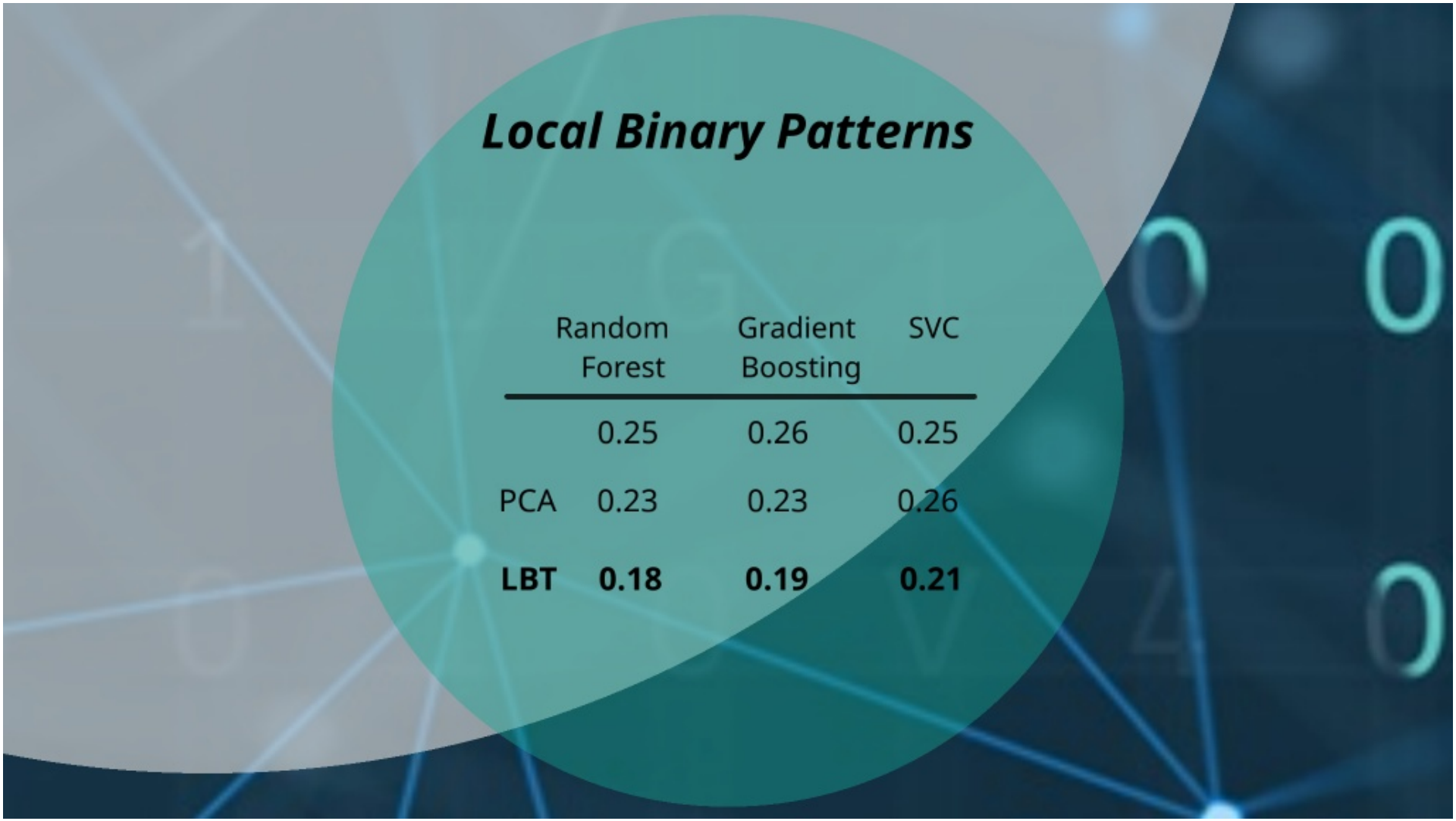
We consider the values of a 3x3 pixel neighborhood

- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.).
- Follow the pixels along a circle. Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1".
- This gives an 8-digit binary number (which is usually converted to decimal for convenience).



Local Binary Patterns

Some examples of images before and after LBP


Local Binary Patterns
Random Forest
Gradient Boosting
SVC
0.25
0.26
0.25
PCA
0.23
0.23
0.26
LBT
0.18
0.19
0.21

CNNs

fast.ai

fast.ai library
PyTorch

PyTorch

Data Augmentation:

- Rotation
- Zoom
- Lighting
- Warp
- Padding
- Horizontal flip

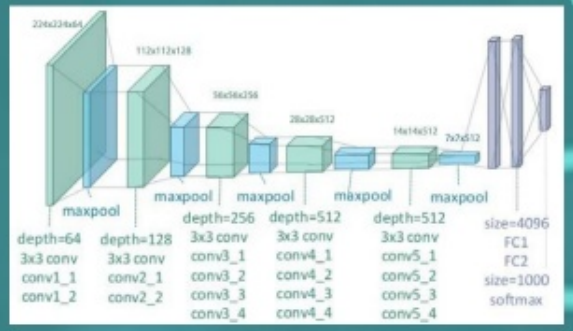
Pre- trained models

My model

Saving the model

Pre- trained Models

VGG-19: Transfer learning

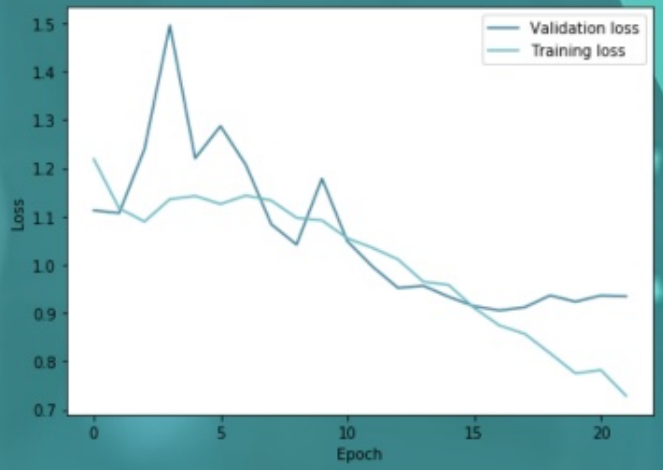


Tuning:

- # Epochs
- Learning rate

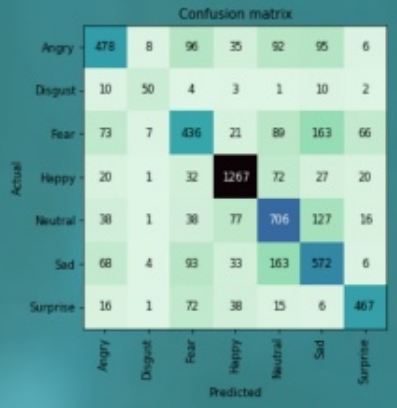
Pre- trained Models

VGG-19: Transfer learning



Pre- trained Models

VGG-19: Transfer learning



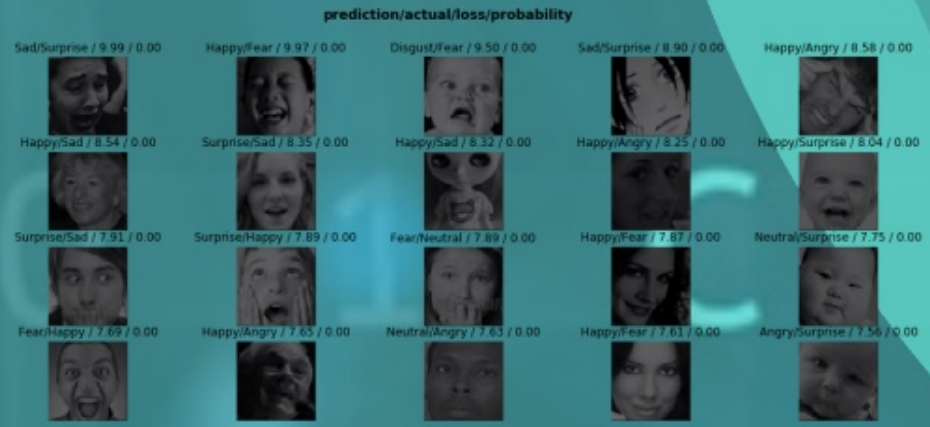
Accuracy: 69%

Most confused:

Fear / Sad:	163
Sad / Neutral:	163
Neutral / Sad:	127
Angry / Fear:	96
Angry / Sad:	95

Pre- trained Models

VGG-19: Transfer learning

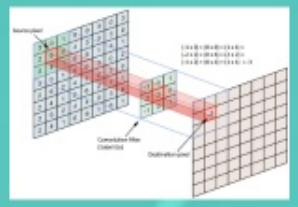


Cleaning?

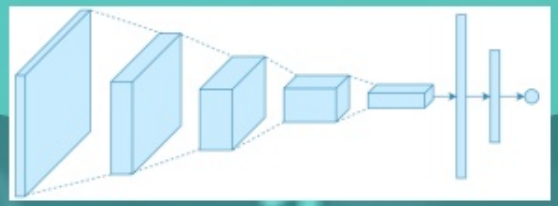
My model

Model from scratch

- Convolutions (stride and padding)

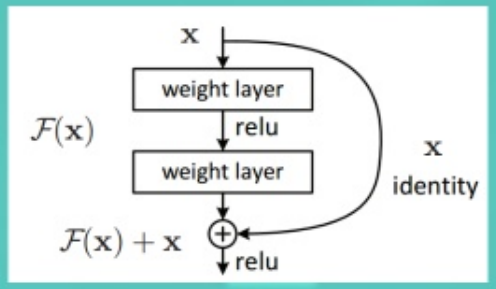


- RELU
- Pooling
- Batch Normalization



My model

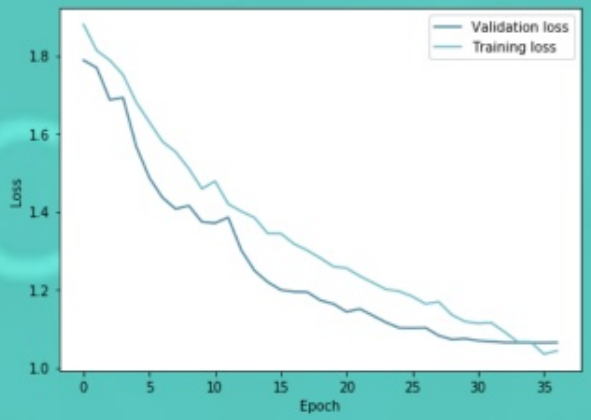
- Residual blocks




```
model_own = nn.Sequential(
    conv_and_res(3, 8), #24
    conv_and_res(8, 16),#12
    conv_and_res(16, 32),#6
    conv_and_res(32, 64),#3
    conv2(64, 128),#2
    nn.MaxPool2d(2,2),
    conv2(128, 256),#1
    Flatten(),
    nn.Linear(256,128),
    nn.Dropout(.5),
    nn.Linear(128,7))
```

- 14 convolutional layers
- 2 fully connected layers

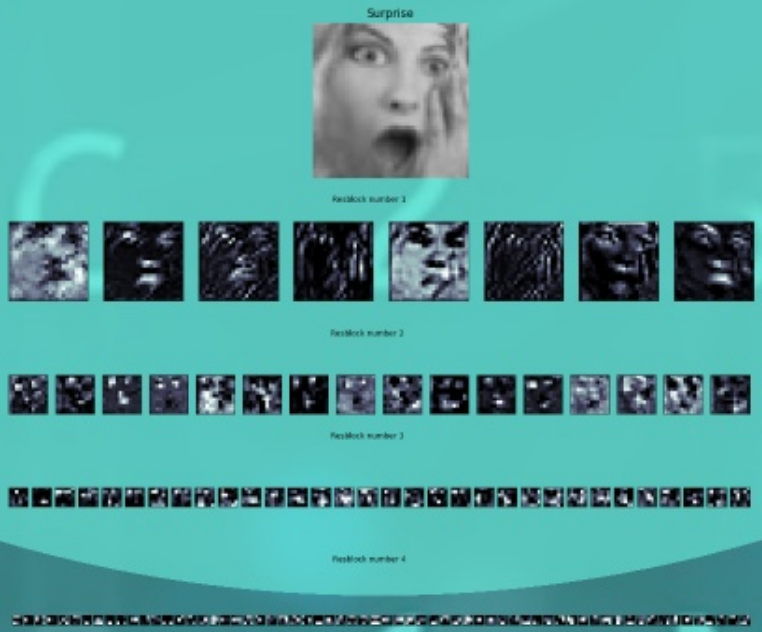
My model



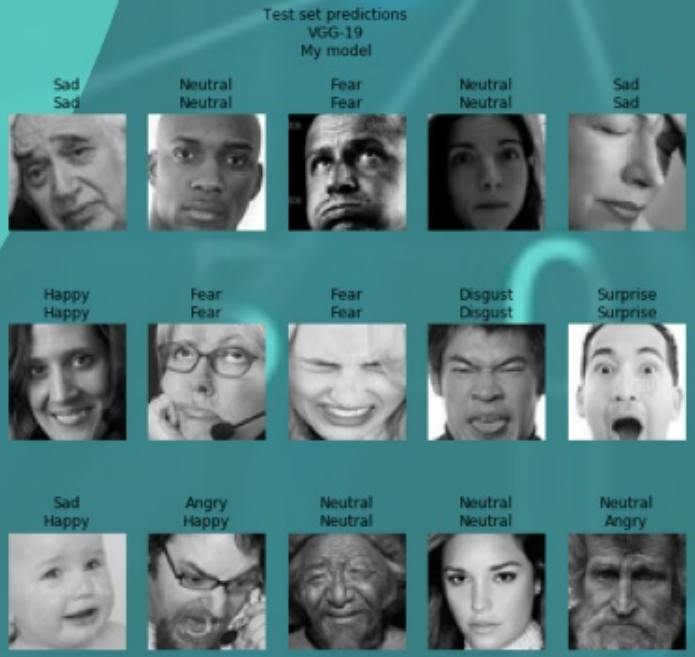
61% accuracy

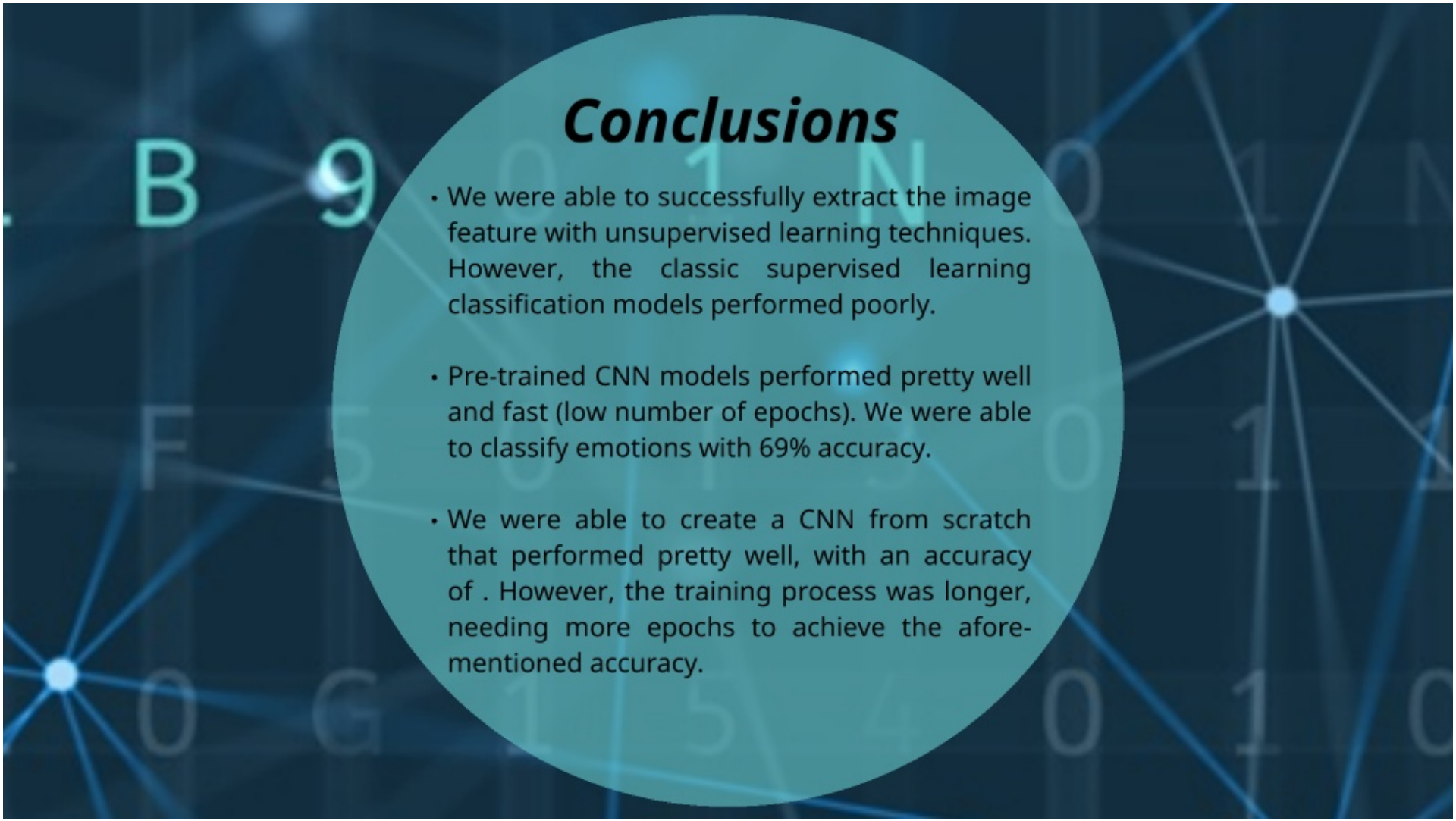
My model

- CNN Layers Visualization



Saving the model




Conclusions
• We were able to successfully extract the image feature with unsupervised learning techniques. However, the classic supervised learning classification models performed poorly.
• Pre-trained CNN models performed pretty well and fast (low number of epochs). We were able to classify emotions with 69% accuracy.
• We were able to create a CNN from scratch that performed pretty well, with an accuracy of . However, the training process was longer, needing more epochs to achieve the afore-mentioned accuracy.

Future Work

- Use other datasets with better labels and higher image resolution.
- In the pre-trained models, we can introduce weights obtained training facial features of other datasets (e.g. VGG-Face).
- Improve our own model: changing achitecture and tuning hyperparameters.
- Use the model to predict facial emotions in videos.

Q & A