

Pattern Recognition

Lecture 01: Introduction

Dr. Shanshan ZHAO

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

Academic Year 2023-2024

Table of Contents

- ① Module Introduction
- ② Course Overview
- ③ PR Introduction
- ④ Exercise

1 Module Introduction

2 Course Overview

3 PR Introduction

4 Exercise

Introduction

- Welcome to DTS201TC
 - Today:
 - Assessments Information
 - Module Information
 - Outline
 - warm up Exercises

Assessment

- 1. Group Course work : 40%
 - Teaming up : 2-4 students per group
 - **DDL: 23:59, Oct. 29th (Sunday, week 6)**
 - 2. Formal Exam : 60%
 - **Date: (could be) Nov. 3rd (Friday, week 7)**, it may be changed

Submission

- **Codes & report**
 - the report should contain necessary descriptions required in the marking scheme, but *not tedious*.
 - **latex** is preferred
 - Please submit in time, late submissions will be penalized.

Information

- Class website:
 - **Learning Mall** DTS201TC module : *announcement, materials,...*
 - **Forum** : Q&A; discussions.
 - Slides and additional materials will be uploaded before class
 - slides could be updated after class.
 - different versions are annotated with vx.y , e.g., v0.0 - before class.
 - Tutorials each week will be basically Q&A about exercise, coursework, etc.

Staff

- **Module Leader:**

Dr. Shanshan ZHAO : Shanshan.Zhao@xjtlu.edu.cn, TC D-5006
office hour for lecture Q&A : Wednesday/Thursday 13:00-15:00
(appointment by emails)

- **Co-teacher:**

Dr. Yuxuan ZHAO : Yuxuan.zhao02@xjtlu.edu.cn, TC-D5008
Dr. Hong Seng GAN : HongSeng.Gan@xjtlu.edu.cn, TC-D5029

- **TA:**

Yiyi Miao : *Yiyi.Miao21@student.xjtlu.edu.cn*
Tong Chen : *Tong.Chen19@student.xjtlu.edu.cn*
Huajun Yuan : *Huajun.Yuan21@student.xjtlu.edu.cn*

Communication

- LMO Announcement will automatically send you an email one hour after the post.
- LMO forum will NOT send notification emails unless you subscribe a specific topic.
- Suggestion: check the email and LMO daily.
- **Please post your questions on LMO forum.
(appreciated)**
- For individual questions, such as office hour reservation, please write emails,
 - emails: formal, in English.
- *No Wechats :(*

① Module Introduction

② Course Overview

③ PR Introduction

④ Exercise

Motivations

- Understand main algorithms in pattern recognition
 - *"We are majoring in engineering, not science. Why do we have to learn these complex mathematical algorithms?"*
- Practise programming skills

Motivations

Why do we need to learn this "old" knowledge while AI is so advanced already?

Does AI agree?



Motivations

Here are the responses from ChatGPT,

- **Fundamental Concepts:** Pattern Recognition provides the foundational understanding of how machines can learn from data and make decisions based on patterns. This knowledge is crucial for students who want to delve deeper into AI and machine learning. Understanding the underlying principles helps them grasp advanced techniques more effectively.
- **Interdisciplinary Understanding:** AI is a multidisciplinary field that draws from computer science, statistics, mathematics, and more. Pattern Recognition encompasses these disciplines, offering students a well-rounded education that enables them to explore various aspects of AI.

Motivations

- **Problem Solving:** Pattern Recognition equips students with problem-solving skills. While modern AI frameworks often automate much of the process, understanding the concepts of feature extraction, dimensionality reduction, and classification methods enables students to address unique challenges and devise creative solutions.
- **Model Interpretation:** Advanced AI models like deep neural networks can be complex and often operate as "black boxes." Knowledge of Pattern Recognition helps students understand model outputs, diagnose issues, and interpret results, enhancing the trustworthiness of AI systems.

Motivations

- **Customization and Adaptation:** Not all AI tasks can be solved using off-the-shelf models. Students with a background in Pattern Recognition can adapt and customize algorithms to fit specific applications, making them more versatile in real-world scenarios.
- **Research and Innovation:** Innovation often builds upon existing knowledge. While AI is rapidly advancing, new breakthroughs can still be rooted in Pattern Recognition principles. Students who understand these foundations are better positioned to contribute to research and push the boundaries of AI.

Motivations

- **Ethical Considerations:** Pattern Recognition includes discussions on data bias, fairness, and privacy, which are critical aspects of responsible AI development. Ensuring that AI systems do not perpetuate biases and respect user privacy is essential, and a solid understanding of these concepts starts with Pattern Recognition.
- **Learning Transfer:** The skills learned in Pattern Recognition can be transferable to other domains, not just within AI. Concepts like feature selection, classification, and decision-making have applications in various fields, making students adaptable and sought after by employers.

Motivations



Very convincing !

Motivations



Let's learn Pattern Recognition !

Lecture topics

- Bayes Decision Theory
- Parametric Estimation methods
- Non-parametric Estimation methods
- Discriminant Methods
- Clustering
- Feature Selection
- Model Evaluation

Software

- Python will be chosen as the programming language



- Some examples using Scikit-Learn Toolkit
<http://scikit-learn.org>
- Some demonstrations on Github related with PR
- IDE :
 - VScode, jupyter notebook/jupyter lab (*suggested*)
 - Pycharm
 - virtual environment : conda/virtualenv/pipenv

Reference books

- ① Richard O Duda, Peter E Hart, et al. Pattern classification. Wiley New York, 2 edition, 2000.
- ② Sergios Theodoridis and Konstantinos Koutroumbas. Pattern Recognition. Elsevier, 2009.
- ③ Keinosuke Fukunaga. Introduction to statistical pattern recognition. Elsevier, 2013.
- ④ Christopher M Bishop. Pattern Recognition and Machine Learning. Springer, 2006.
- ⑤ Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.
- ⑥ Kevin P. Murphy. Probabilistic Machine Learning: An Introduction. The MIT Press, 2020.

1,2 are suggested. 3,4,5,6 are only used as supplementary materials.

① Module Introduction

② Course Overview

③ PR Introduction

④ Exercise

What is a Pattern?

Pattern is an abstraction, represented by a set of measurements describing an object.

Many types of pattern exist: visual, temporal, sonic,...



(a)

(b)

(c)



(d)

(e)

(f)

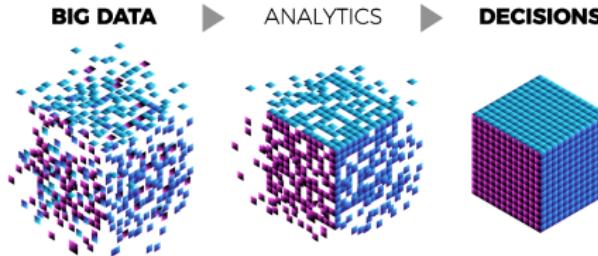
What is Pattern Recognition?

Basically, it refers to algorithms/theory/system to put patterns into categories.

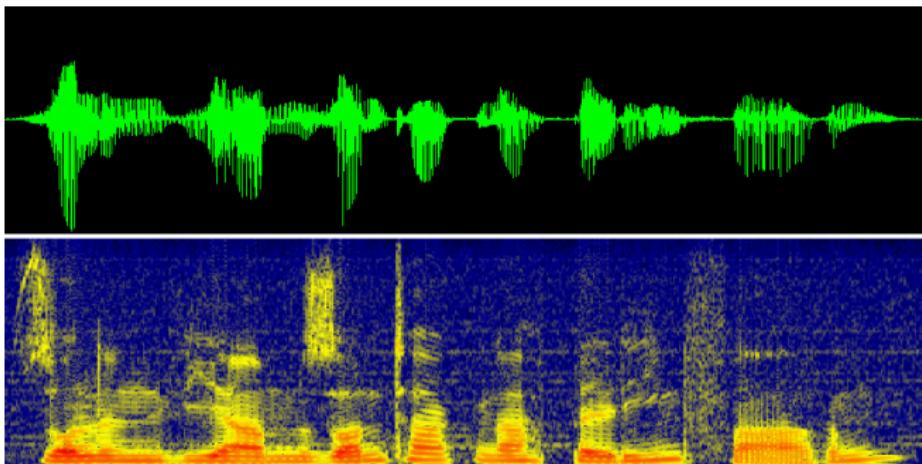
Pattern recognition is the process of recognizing regularities in data by a machine.

Pattern Recognition is the study of how machine can

- observe the environment
- learn to distinguish patterns
- make reasonable decisions about the categories of the patterns



Speech Recognition



Financial Trading

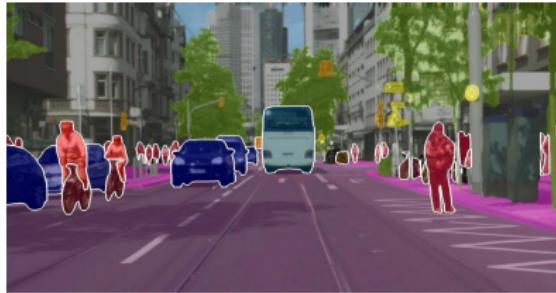


Object Detection/Classification



(g)

(h)

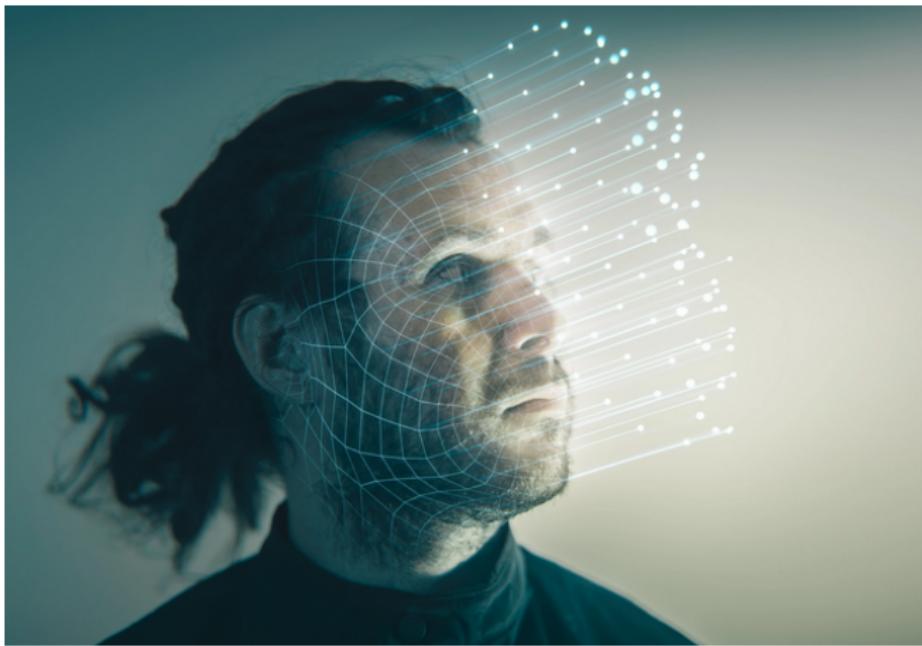


(i)

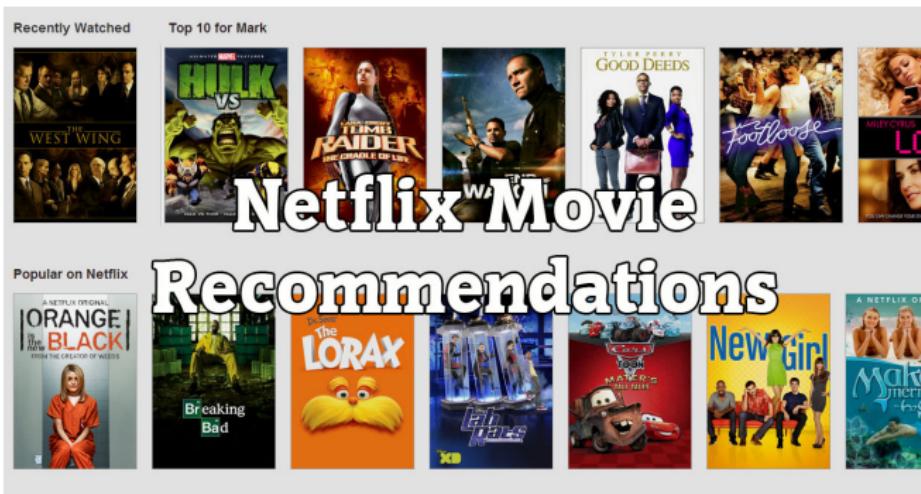
Fingerprint Identification



Face Recognition



Movie/Music Recommendation



Case study



(j) Salmon



(k) Sea Bass

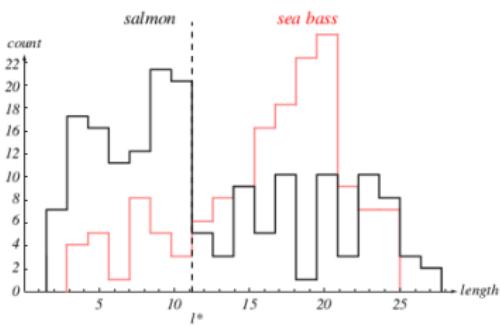


(l) NOT the
Salmon you are
thinking of

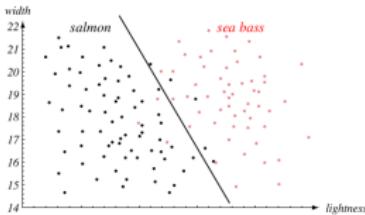
Case study



(m)

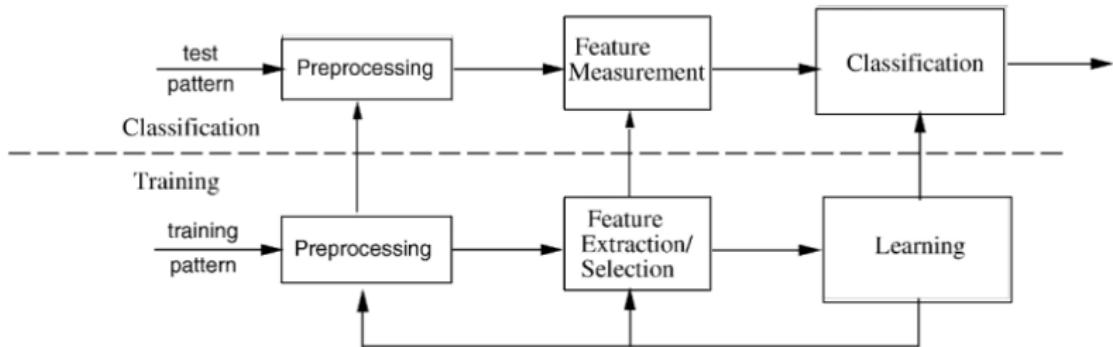


(n)



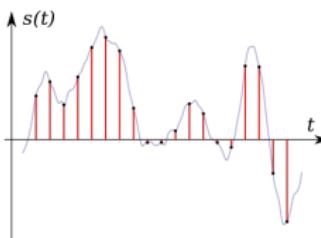
(o)

General PR workflow

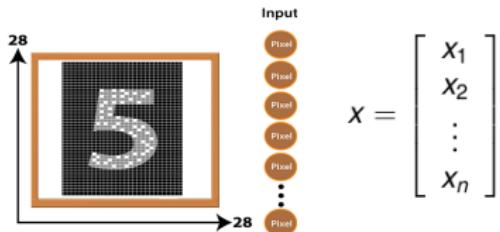


- Data collection: large and representative set of examples
- Feature choice: problem domain, invariant to irrelevant transformation, insensitive to noise
- Model choice: performance

Mathematical problem formulation



(p) waveform

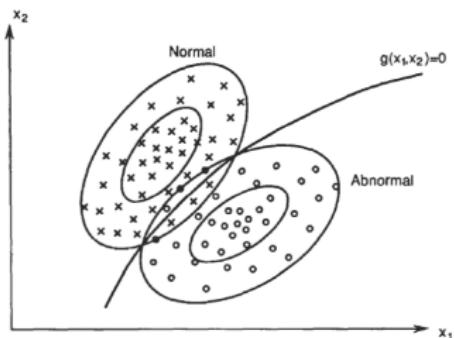


(q) character

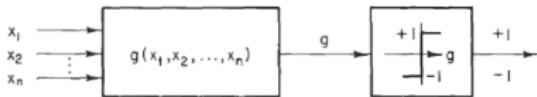
(r) a vector by sampling

Measurement of patterns.

Mathematical problem formulation



(s)



(t)

1 Module Introduction

2 Course Overview

3 PR Introduction

4 Exercise

YOU LOVE CODING ❤

Warm up Exercise

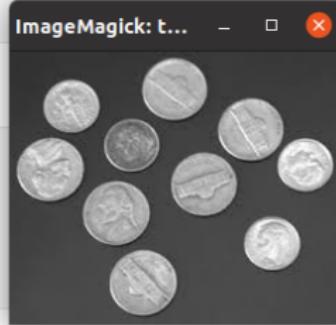
- load packages

```
[1]: import numpy as np
from scipy import signal
from PIL import Image
from matplotlib import pyplot as plt
```

- load image

```
[2]: img_filename = './imgs/coins.png'
with Image.open(img_filename) as img:
    img.show() # display the image, you may also use display(img)
    print(img.size, img.mode) # print the info of the image
    img_g = img.convert('F')
    print(img_g.size, img_g.mode)
    img_arr = np.array(img_g)

(240, 196) RGB
(240, 196) F
```



YOU LOVE CODING ❤

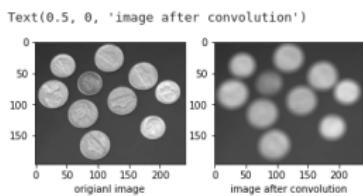
```
# Construct a filter kernel used for blurring
mean_kernel = np.ones((11, 11)) / (11*11)

# convolution
mean_arr = signal.convolve2d(img_arr, mean_kernel, mode='same', boundary='symm') # Q: how do the parameter 'mode' and 'boundary' make a difference?

print('Shape of original: ', img.size)
print('Shape of convolved: ', mean_arr.shape)

Shape of original: (240, 196)
Shape of convolved: (196, 240)

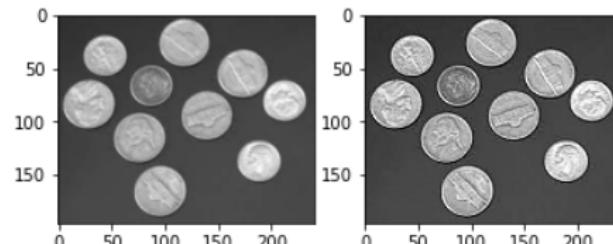
# Display the results
fig, axs = plt.subplots(1, 2)
axs[0].imshow(img_arr, cmap='gray', vmin=0, vmax=255)
axs[0].set_xlabel('original image')
axs[1].imshow(mean_arr, cmap='gray', vmin=0, vmax=255)
axs[1].set_xlabel('image after convolution')
```



YOU LOVE CODING ❤

2. Construct a filter kernel used for sharpening

```
sharpen_kernel = np.array([[0, -1, 0],  
                           [-1, 5, -1],  
                           [0, -1, 0]])  
  
# Convolution  
sharpen_arr = signal.convolve2d(img_arr, sharpen_kernel, mode='same', boundary='symm')  
  
# Display the results  
fig, axs = plt.subplots(1, 2)  
axs[0].imshow(img_arr, cmap='gray', vmin=0, vmax=255)  
axs[0].set_xlabel('original image')  
axs[1].imshow(sharpen_arr, cmap='gray', vmin=0, vmax=255)  
axs[1].set_xlabel('image after convolution')  
  
Text(0.5, 0, 'image after convolution')
```



YOU LOVE CODING ❤

Task 1

Thresholding

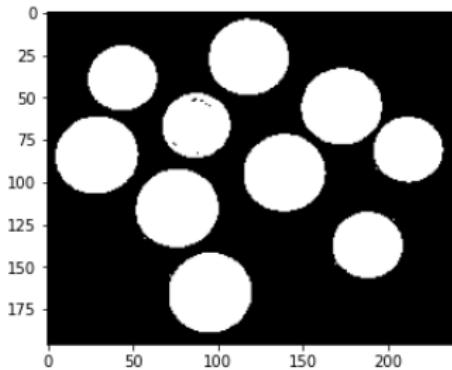
Use the operators $>$, $<$, \geq , \leq to threshold the image using an arbitrary threshold.

That is, divide the image pixels into two classes, the ones below and the ones above the threshold, and give them different values (e.g. 0 and 255)

YOU LOVE CODING ❤

```
THRE = 90
img_thre = img_arr > THRE
```

```
plt.imshow(img_thre, cmap='gray')
img_thre = img_thre.astype(int)
```



YOU LOVE CODING ❤

Task 2

Use the methods in Pillow to implement

<https://pillow.readthedocs.io/en/stable/handbook/index.html>

1. vertically flipped
2. horizontally flipped
3. rotated by 90 degrees

YOU LOVE CODING ❤

```
# load and display image
img_filename = './imgs/windows-xp.jpg'
with Image.open(img_filename) as img_obj:
    img_obj.show() # display the image, you may also use display(img)
    # if you fail to run the codes below,
    # try to put the codes here, under the 'with Image.open(...)'
```

```
# one solution
flip1 = img_obj.rotate(90)
flip1.show()
```

```
# another way
```

```
# 1. vertically flipped
verti_flipped = img_obj.transpose(Image.FLIP_TOP_BOTTOM)
verti_flipped.show()
```

```
# 2. horizontally flipped
hori_flipped = img_obj.transpose(Image.FLIP_LEFT_RIGHT)
hori_flipped.show()
```

```
# 3. rotated by 90 degrees
rotated90 = img_obj.transpose(Image.ROTATE_90)
rotated90.show()
# display(rotated90)
```

YOU LOVE CODING ❤

Task 3

Implement the **function swap_RB()** where:

- In the returned image, the red and blue channels of the input image have to be swapped.
- You can carry out the operation using arraying indexing or with loops

YOU LOVE CODING ❤

```
def swap_RB(filename):
    # load and display image
    with Image.open(filename) as img_obj:
        img_obj.show() # display the image, you may also use display(img)
        print(img_obj.mode)
    # convert the Image object to numpy array
    img_arr = np.array(img_obj)
    print(img_arr.shape)
    # create a new image with the same size as the original image
    swap_RB_img = np.zeros(img_arr.shape)
    swap_RB_img[:, :, 0] = img_arr[:, :, 2]
    swap_RB_img[:, :, 1] = img_arr[:, :, 1]
    swap_RB_img[:, :, 2] = img_arr[:, :, 0]
    img_swap = Image.fromarray(np.uint8(swap_RB_img))
    img_swap.show()
```

```
img_filename = './imgs/windows-xp.jpg'
swap_RB(img_filename)
```

RGB
(720, 1200, 3)

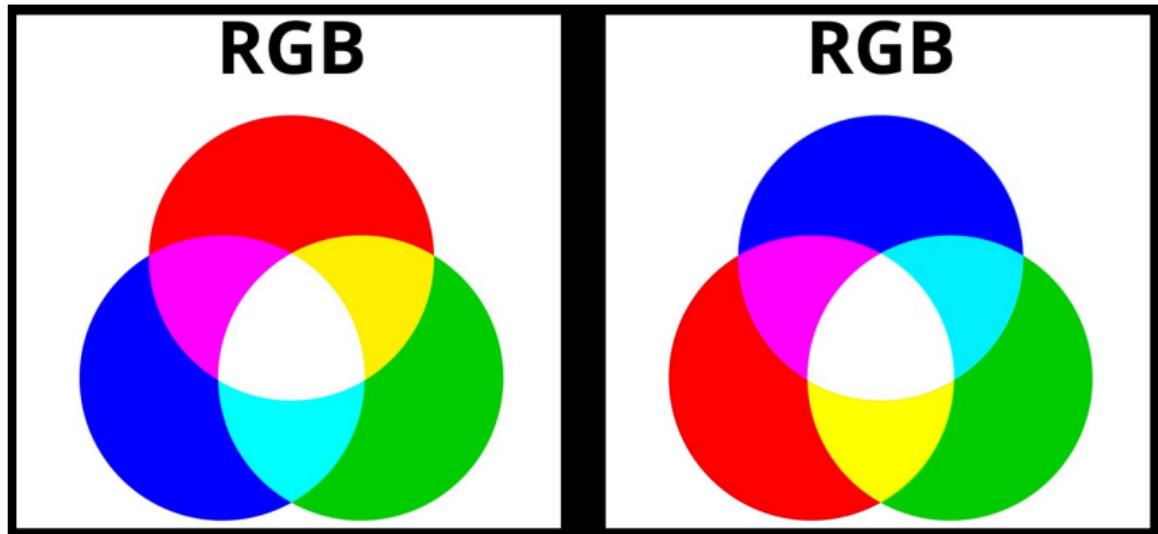
YOU LOVE CODING ❤

Results of task 3



YOU LOVE CODING ❤

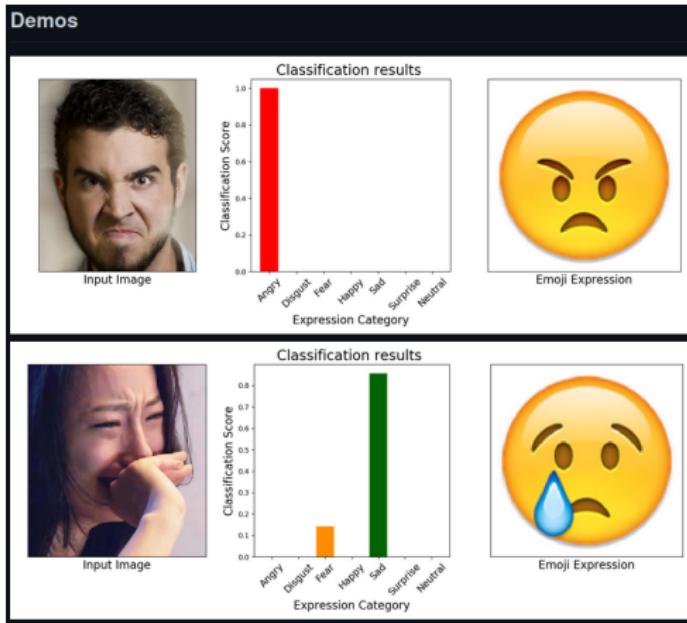
Results of task 3



YOU LOVE CODING ❤

Extras

<https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch>



Thank You !
Q & A