

经济管理学院

课 程 报 告

(复杂网络与社会计算)

题 目: week14 课程作业

课程教师: 赵吉昌

学院/专业: 信息管理与信息系统

学生姓名: 范春

学 号: 21377061

2024 年 5 月 28 日

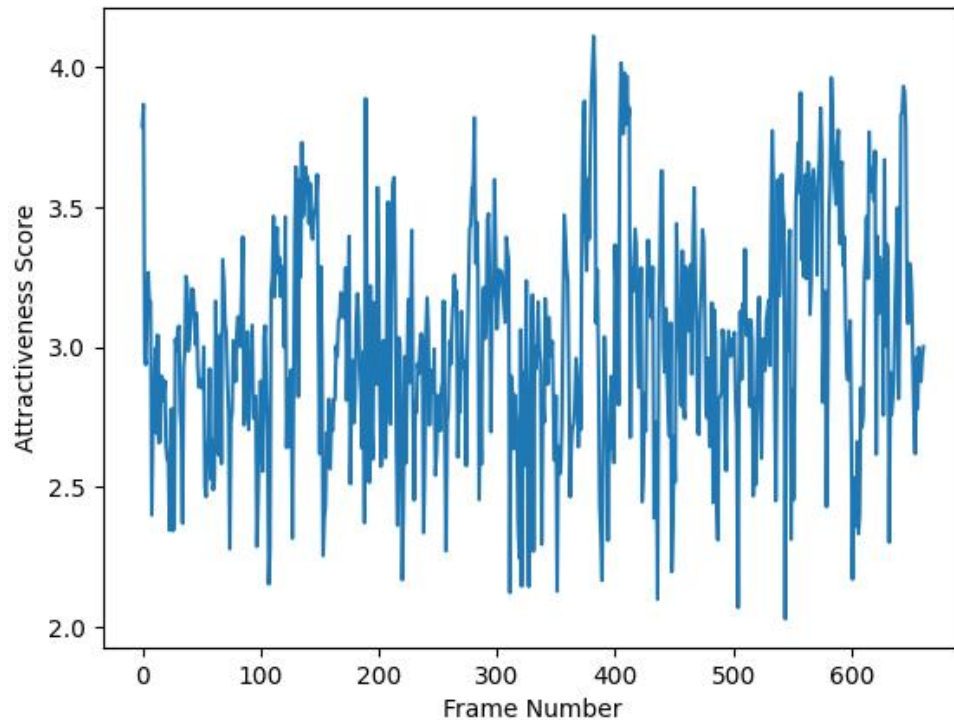


北京航空航天大学
BEIHANG UNIVERSITY

作业要求如下：

1. 阅读本周资料。
2. 根据1中的相关定义和理论，利用SCUT-FBP5500 (<https://github.com/HCIILAB/SCUT-FBP5500-Database-Release>) 公开的神经网络代码和模型权重，对提供的face_seq.zip的人脸进行面孔吸引力计算。假设图片的文件名为帧序号（虽不连续，但大小能够表征时间先后），请以得到的吸引力为y，连续化后的序号为x（从小到大，从0开始重新编号），观察直播过程中吸引力的波动情形。
3. 思考2中所得的曲线，对直播有何管理的实践意义？如有，请简单讨论并结合2中的例子给出具体计算与展示。

任务一：计算吸引力



代码如下：

```
import os
import torch
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from torchvision import transforms
from collections import OrderedDict
from Nets import AlexNet
import pandas as pd

# 1. 加载预训练模型
model_path = "C:\\Users\\范春\\Desktop\\week14\\pytorch-models\\alexnet.pth"
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = AlexNet(num_classes=1)
# 从 checkpoint 中提取 state_dict
checkpoint = torch.load(model_path, map_location=device, encoding='latin1')
state_dict = checkpoint['state_dict']
# 修改键名称以匹配模型
```

```

new_state_dict = OrderedDict()
for k, v in state_dict.items():
    name = k.replace('module.', '')
    new_state_dict[name] = v
model.load_state_dict(new_state_dict)
model.eval()
model.to(device)
# 2. 定义图像预处理
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
# 3. 计算面孔吸引力
image_dir = "C:\\Users\\范春\\Desktop\\week14\\face_seq\\face_seq"
image_files = sorted(os.listdir(image_dir), key=lambda x: int(os.path.splitext(x)[0])) # 按帧序号排序
scores = []
for img_file in image_files:
    img_path = os.path.join(image_dir, img_file)
    image = Image.open(img_path).convert('RGB')
    image = transform(image).unsqueeze(0).to(device)

    with torch.no_grad():
        score = model(image).item()
        scores.append(score)
df = pd.DataFrame({
    'Frame Number': list(range(len(scores))),
    'Attractiveness Score': scores
})
# 保存到 Excel 文件
output_path = "C:\\Users\\范春\\Desktop\\week14\\attractiveness_scores.xlsx"
df.to_excel(output_path, index=False)
# 4. 绘制时间序列图
plt.plot(scores)
plt.xlabel('Frame Number')
plt.ylabel('Attractiveness Score')
plt.show()

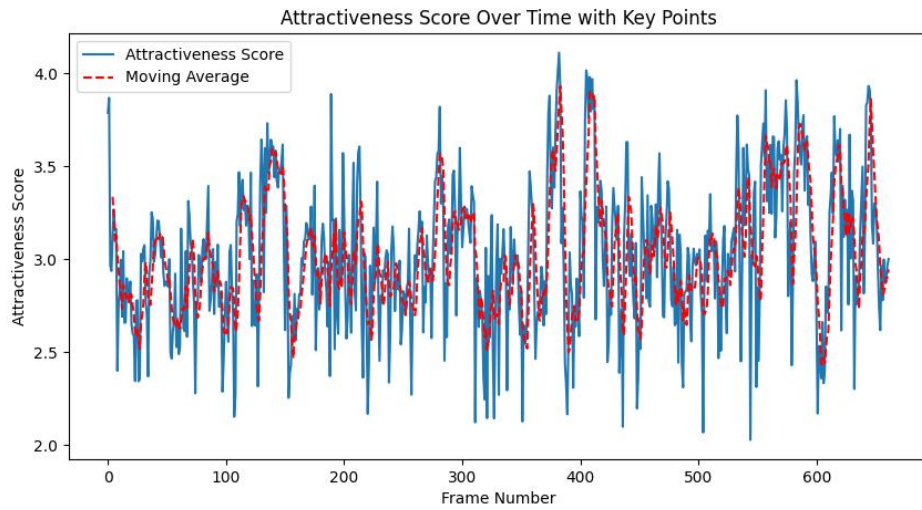
```

任务二：

任务一中，我们可以实时计算主播的面孔吸引力，该值一定程度上反映出观众的实时感受，主播可以据此对直播方式作出调整，采取观众最乐于接收的方式进行。此外，可以借助该模型对主播进行有针对性的培训，找到每个主播面孔吸

引力较大的表情管理，提升主播的表现和直播的吸引度。

设置移动窗口，计算移动均值。



代码如下：

```
import pandas as pd
import matplotlib.pyplot as plt

input_path = "C:\\Users\\范春\\Desktop\\week14\\attractiveness_scores.xlsx"
df = pd.read_excel(input_path)
# 计算移动平均
window_size = 5
df['Moving Average'] = df['Attractiveness Score'].rolling(window=window_size).mean()
plt.figure(figsize=(10, 5))
plt.plot(df['Frame Number'], df['Attractiveness Score'], label='Attractiveness Score')
plt.plot(df['Frame Number'], df['Moving Average'], label='Moving Average', linestyle='--',
color='red')
plt.xlabel('Frame Number')
plt.ylabel('Attractiveness Score')
plt.title('Attractiveness Score Over Time with Key Points')
plt.legend()
plt.show()
output_path = "C:\\Users\\范春\\Desktop\\week14\\attractiveness_scores_with_moving_average_and_top_points.xlsx"
df.to_excel(output_path, index=False)
```