

## HW2: EigenX

### 1 作业要求

# HW#2: EigenX

## 写代码实现某一种视觉相关数据（非人脸）的EigenX算法(包含训练与重构过程)

1. 自行选择一种视觉相关的数据对象（比如：某种形状或非人脸的某种物体等等），并自行构建一个该对象的数据集，样本数不少于100个；
2. 实现2个程序过程：训练、重构
3. **训练过程**，大致为：“**mytrain** (能量百分比 **model**文件名 其他参数...)”，用**能量百分比**决定取多少个特征量，将训练结果（即特征向量）输出保存到**model**文件中。演示程序同时将前10个特征量分别显示出来。

**Hint**: 对于维度较高的情况：可参考SVD（奇异值分解）定理，请自行学习。

4. **重构过程**：大致为：“**myreconstruct** (输入的待处理数据文件名 **model**文件名 其他参数)”，将**model**文件装载进来后，对输入的待处理数据进行变换到EigenX空间，然后再用变换后的结果重构回原来的数据。演示程序可以同时显示用10个PCs、25个PCs、50个PCs、以及75个PCs重构的结果。

**提交截至时间：2024年12月15日 8:00**

### 2 运行

#### 2.1 环境

```
python==3.9.18
opencv-python==4.10.0.84
numpy==1.26.4
torch==2.2.0
```

#### 2.2 运行

```
cd hw2
python download_data.py
python main.py
```

## 3 实现

### 3.1 数据集

- 使用了一个动漫人物全身像数据集[Fullbody Anime Girls Datasets for Gans](#)
- 随机抽取ultraclean\_256中的少量图像作为training set
- 抽取两张图展示



### 3.2 训练

#### 3.2.1 获取平均图

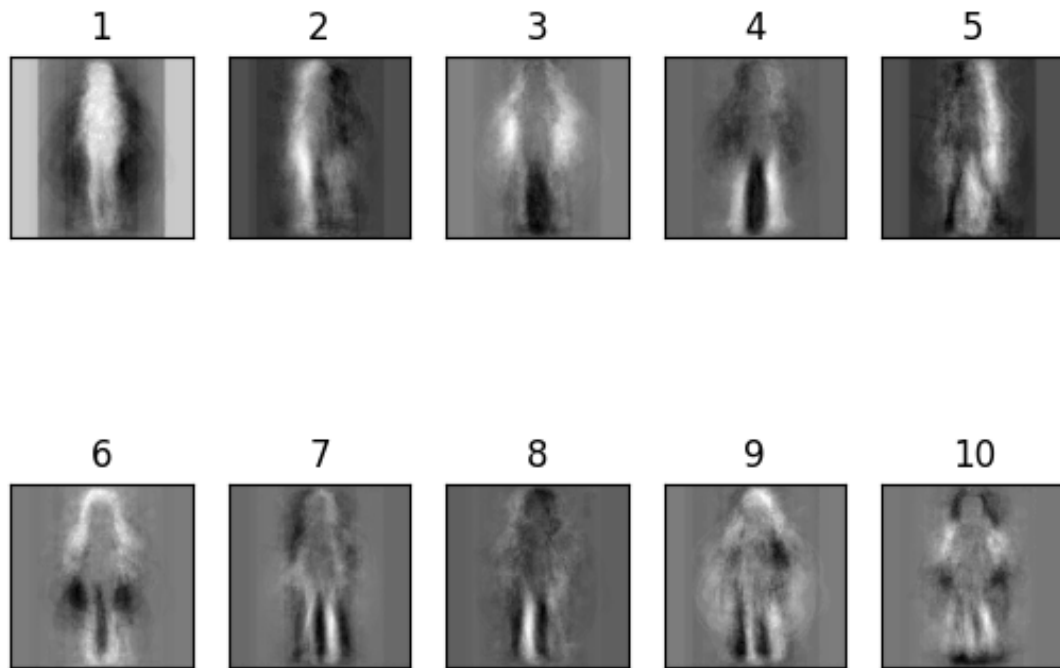


#### 3.2.2 Top10 特征向量

- 使用了SVD分解，放到GPU里算来加速

```
vec_norm = torch.tensor(vec_array - avg_vec).cuda()
u, s, v = torch.linalg.svd(vec_norm)
eigenvecs = v.cpu().numpy()[ :int(p * len(s))]
```

- 结果



### 3.3 重构

- 首先将原图project到特征向量张成的空间，再Top PCs个特征向量来重建图，

```
vec = (torch.tensor(self.img2vec(img)).cuda() -
        self.avg_vec).unsqueeze(0) # [1, h*w]

v_k = self.v[:, :k]
vec_encode = torch.einsum('mn,nk -> mk', vec, v_k)
vec_decode = torch.einsum('mk,kn -> mn', vec_encode, v_k.T)

img_recon = (vec_decode + self.avg_vec).cpu().numpy().reshape(
    self.img_shape)
```

- 结果

