

similarity search

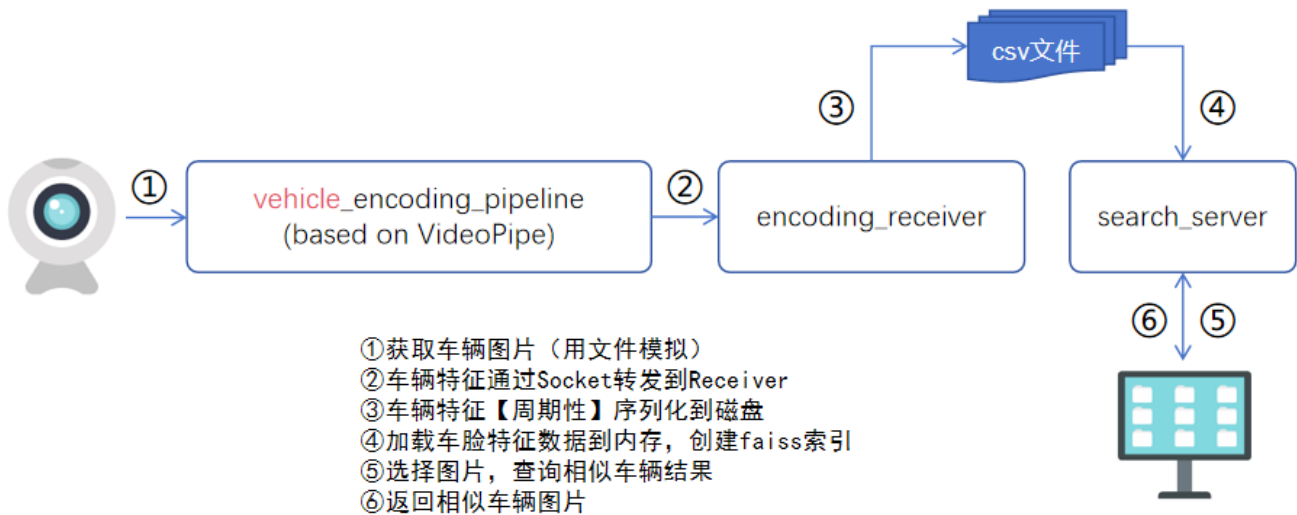
使用VideoPipe做以图搜图（车辆/人脸）。

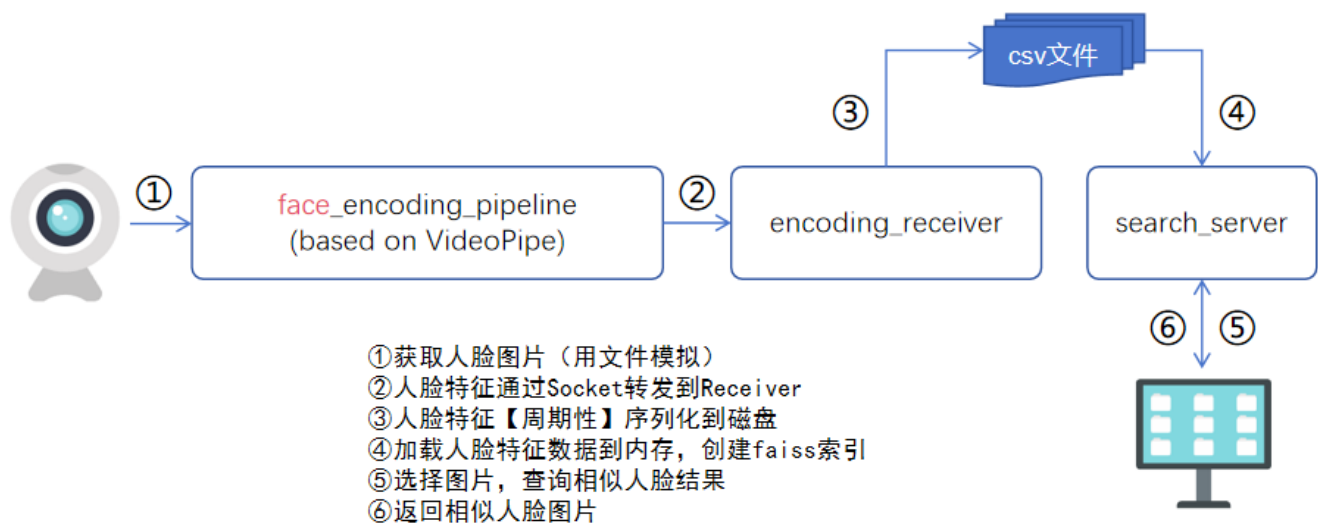
提前准备

- 1. 已准备好 VideoPipe 编译/运行环境（参见视频教程）
- 2. 已准备好本案例所需的测试模型和测试数据（即 `vp_data`，具体参见视频教程）
- 3. 已安装好本案例中 Python3 所需的依赖库（主要是 `numpy,pandas,opencv==4.6.0,flask,faiss` 等，其余按需安装即可）

主要文件

- 1. `vehicle_encoding_pipeline.cpp` 运行VideoPipe管道读取图片，检测车辆、提取车辆特征，并将特征通过Socket/UDP发送给后台服务。
- 2. `face_encoding_pipeline.cpp` 运行VideoPipe管道读取图片，检测人脸、提取人脸特征，并将特征通过Socket/UDP发送给后台服务。
- 3. `encoding_receiver.py` 用于接收车辆/人脸的特征数据脚本，将其写入 `pandas.dataframe` 结构，然后以 `.csv` 的文件格式序列化到磁盘（每接收100个特征，会写入一次磁盘）。
- 4. `search_server.py` 加载 `.csv` 文件中的特征数据，并创建 `faiss` 索引对象。该脚本同时会提供一个基于 `flask` 的Web UI界面，用户可以上传图片、检索与之相似的车辆/人脸图片。
- 5. `clean.sh` 帮助清空当前workspace下面的所有缓存数据。





如何使用

- 1. CMake编译 `vehicle_encoding_pipeline.cpp` 和 `face_encoding_pipeline.cpp` 生成二进制可执行文件并运行其中一个（车辆或人脸），它会实时检测目标、并提取特征数据。（编译之前请先修改代码中 `des_ip` 的值）
- 2. 运行 `python3 encoding_receiever.py` 脚本，它会接收特征数据，并周期性地存成 `.csv` 格式文件。（运行前，请先修改 `bind_ip` 的值）
- 3. 运行 `python3 search_server.py vehicle/face` 脚本（注意每次只能传入vehicle或face其中一个参数，表示当前检索车辆还是人脸），然后打开浏览器访问 `http://server-ip:9999`，即可开始测试以图搜图的功能。

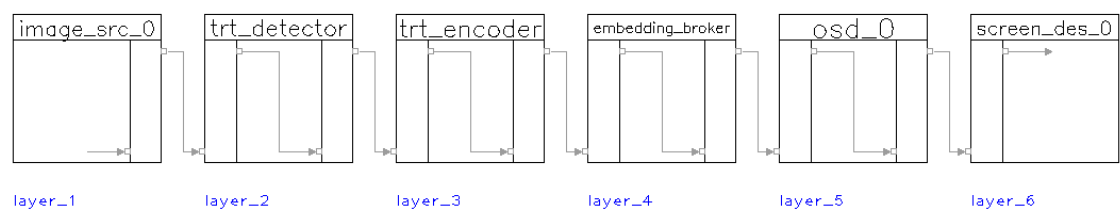
数据流向

- 1. [pipeline -> embedding data -> send via socket]
- 2. [receive embedding data -> serialize to disk]
- 3. [deserialize from disk -> create Index Object of faiss -> search images from web]

please run ``sh clean.sh`` if you want to regenerate embeddings, for example, switch vehicle similiarity search to face similiarity search.

VideoPipe管道介绍

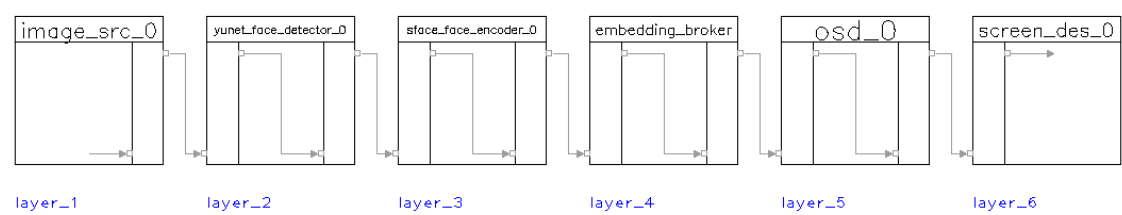
以图搜车↓



- 1. `layer_1` 模拟获取车辆图片（大图），实际可以换成file、application等来源。

- 2. layer_2 完成车辆检测，在大图中定位车辆位置，创建 vp_frame_target 对象，初始化它们的 primary_label/primary_class_id/primary_score。
- 3. layer_3 完成车辆特征提取，基于前面检测到的车辆目标，提取车辆特征，更新 vp_frame_target 的 embeddings。属于二级推理。
- 4. layer_4 完成数据转发，将管道中车辆特征 (embeddings)、车辆抠图路径按自定义格式，以 Socket/UDP的方式发送给Python后台。
- 5. layer_5 完成结果绘制，将前面检测到的车辆结果 (vp_frame_target) 绘制到帧上。
- 6. layer_6 负责GUI输出显示，显示基本无延时。注意管道没有rtmp输出。

以图搜人脸↓



- 1. layer_1 模拟获取人脸图片（大图），实际可以换成file、application等来源。
- 2. layer_2 完成人脸检测，在大图中定位人脸位置和5个关键点，创建 vp_frame_face_target 对象，初始化其 x/y/width/height/keypoints。
- 3. layer_3 完成人脸特征提取，基于前面检测到的人脸目标，提取特征，更新 vp_frame_face_target 的 embeddings。属于二级推理。
- 4. layer_4 完成数据转发，将管道中人脸特征 (embeddings)、人脸抠图路径按自定义格式，以 Socket/UDP的方式发送给Python后台。
- 5. layer_5 完成结果绘制，将前面检测到的人脸结果 (vp_frame_face_target) 绘制到帧上。
- 6. layer_6 负责GUI输出显示，显示基本无延时。注意管道没有rtmp输出。

重要知识点

- 1. vp_frame_face_target 类型。
- 2. 相似度检索(similiarity search)中topK的概念。
- 3. 数据代理节点工作原理。

什么是 faiss

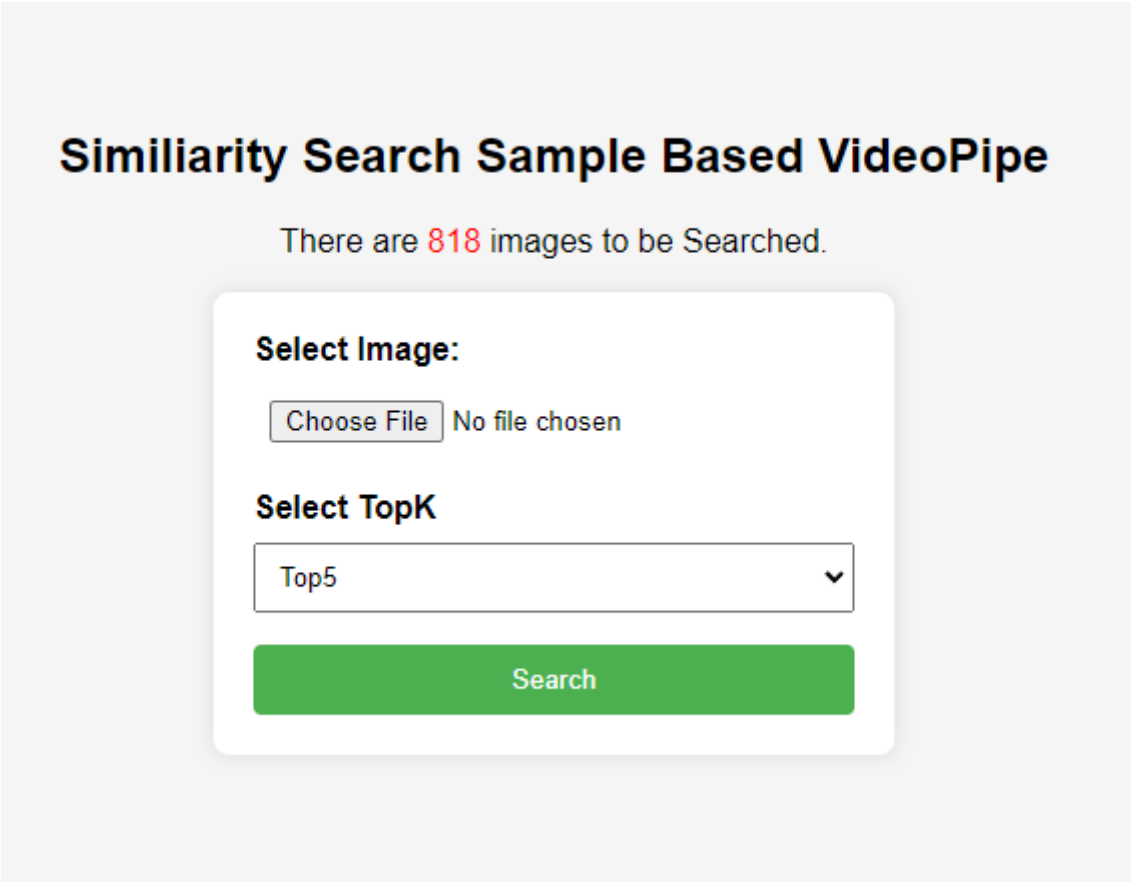
FaceBook开源的一款相似度检索工具，可用于图片、文本相似性检索场景, 全名是Facebook AI Similiarity Search.

如何安装 faiss

```
pip3 install faiss-cpu
# pip3 install faiss-gpu
```

Sample运行截图

选择查询图片



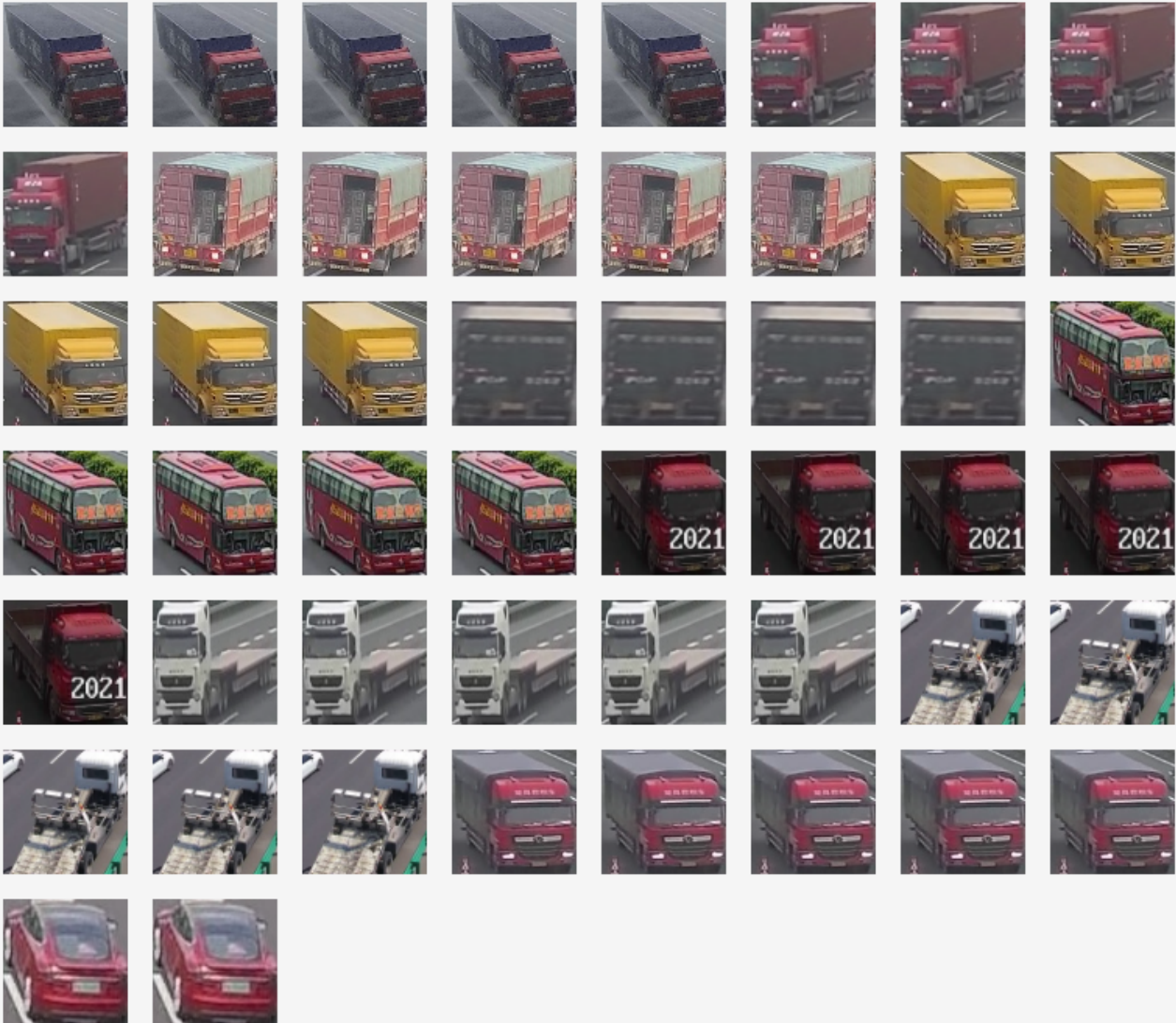
车辆搜索结果

[back to home](#)

Query Image



Search Results of top50



人脸搜索结果

Query Image

[back to home](#)



Search Results of top50

