Инструкция по настройке и использованию проекта с отслеживанием масок - Приложение для распознавания людей

Введение

Настоящее приложение служит составной частью инсталляции, наблюдающей за людьми в помещении.

Необходимые компоненты:

- ПК под управлением ОС Win10/11 с видеокартой, поддерживающей CUDA и cuDNN (начиная с GeForce GTX 750 / 900-серии и далее);
- Python 3.11.9;
- CUDA 12;
- cuDNN 9.5;
- IP-Камера с поддержкой RTSP потока.

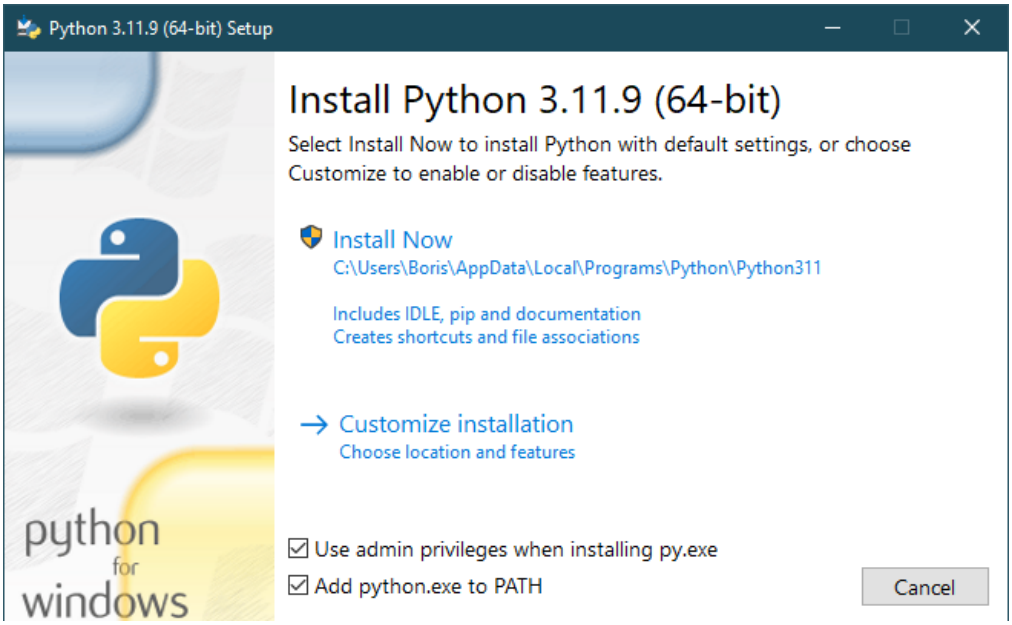
Отладка проекта (без дообучения модели) выполнялась в конфигурации:

- CPU – AMD Ryzen 5 2600 Six-Core Processor 3.40 GHz;
- GPU – Nvidia GeForse GTX 1060 6gb;
- RAM – 16Гб DDR4 3200МГц;
- Камера – Tiandy TC-C321N SPEC:I3/E/Y/2.8mm 1920x1080, 25 кадр./сек.

Для рассматриваемой задачи оптимальной GPU является RTX 3060 (12 ГБ) по соотношению цена-производительность

Инструкция по установке приложения

1. **Установите** Python 3.11.9, CUDA 12, cuDNN 9.5;



2. **Распакуйте проект MaskDetection в любой удобной директории;**

3. **Установите зависимости с помощью команды в командной строке предварительно перейдя в директорию с проектом:**

```
pip install -r requirements.txt
```

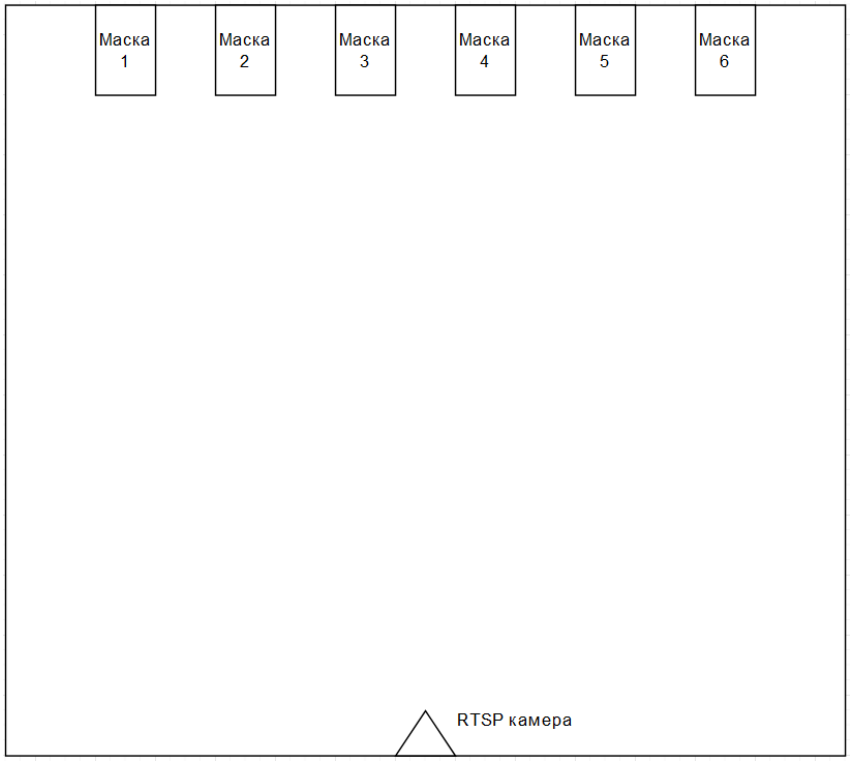
4. **Запустите проект с помощью команды:**

```
python app.py
```

В результате запустится веб-приложение для настройки стенда и управления процессом обнаружения людей в зале.

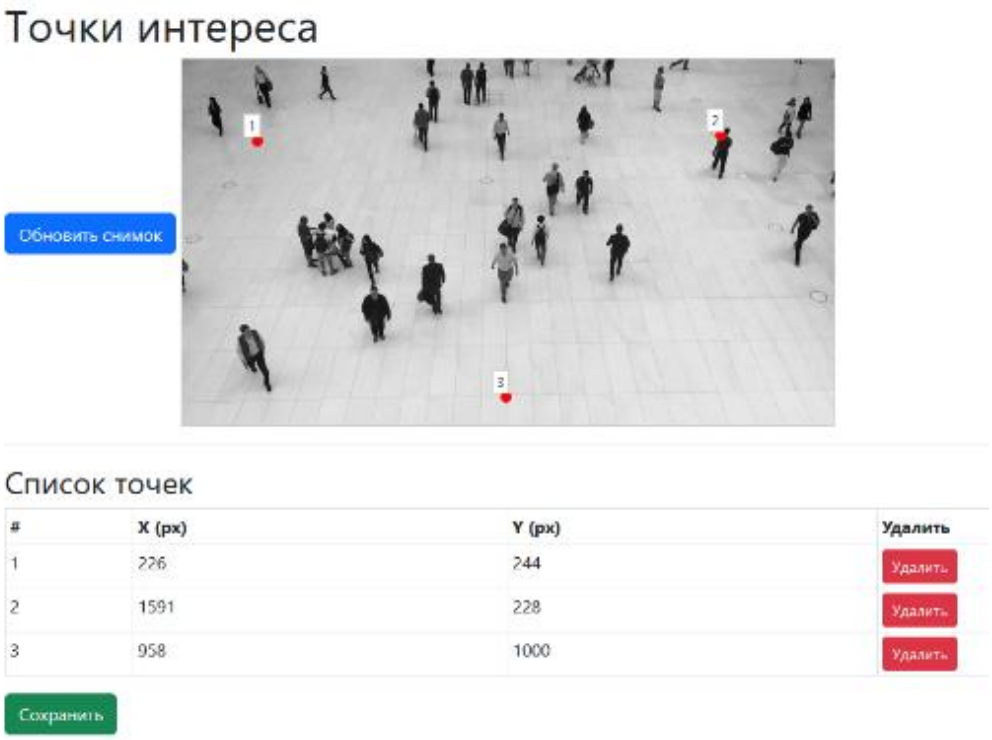
Адрес приложения по умолчанию: http://127.0.0.1:5000/

5. **Подготовьте камеру для съемки зала:**
   - Используйте камеру с поддержкой rtsp трансляции;
   - Разместите камеру как можно выше таким образом, чтобы она покрыла как можно большую площадь зала, так же очень важно, чтобы камера видела мониторы, на которых будут отображаться маски, хотя бы нижние грани;



6. **Настройте конфигурацию приложения:**
   - Откройте настройки приложения http://127.0.0.1:5000/config;
   - Укажите rtsp поток вашей камеры в поле «RTSP URL (камера)»;
   - Укажите адрес и порт стенда, на котором будет запущена сцена с масками;
   - Прочие настройки влияют на производительность и качество распознавания и будут описаны ниже.

7. **Настройка точек интереса (обозначение дисплеев с масками в зале)**
    - Откройте страницу с настройками точек интереса - http://127.0.0.1:5000/points;
    - Удалите имеющиеся точки и обновите снимок с камеры (обновление занимает несколько секунд);
    - С помощью мыши щелкните ЛКМ по каждому из дисплеев, на которых будут изображены маски;



    - Сохраните результат.

ВАЖНО: номер точки синхронизирован номеру маски на сцене (маска 1 работает с точкой интереса 1 и т.д.)

8. **Калибровка матрицы гомографии**
    - Откройте настройки гомографии http://127.0.0.1:5000/homography
    - Удалите существующие точки;
    - Укажите на снимке углы на полу помещения и назначьте для выбранных точек реальные координаты в см;
    - Сохраните результат.

- ***Минимальное количество точек***

➢ Для решения уравнения гомографии нужно указать минимум 4 точки.
➢ Однако, лучше использовать 5–6 и более точек, для устойчивого результата.

- ***Расположение точек***

➢ Не выбирайте точки, которые лежат (примерно) на одной прямой.
➢ Старайтесь распределить точки по всему полю кадра, охватывая углы, центр, разные части.
➢ Желательно выбирать углы или чётко определяемые ключевые места (пересечения линий, чёткие углы объектов и т. п.), чтобы уменьшить ошибку на этапе «глазомерного» попадания.

- ***Очерёдность выбора***
- Порядок (1-я, 2-я, 3-я, 4-я точка) не важен.
- Единственное требование: каждая пара должна соответствовать одной и той же «физической» точке.
- То есть (x1, y1) в пикселях должно совпадать по смыслу с (realX1, realY1) в реальных координатах.
- Если вы случайно «перепутаете» соответствие, гомография будет некорректной.

- ***Как лучше организовать сбор***
- Выберите не менее 4 чётко определённых точек на плоскости (например, углы ковра, пересечение разметки, углы какой-то прямоугольной зоны).
- При «клике» мышью на изображении программа запоминает (x, y) пиксельные. Вы вводите (realX, realY).
- Повторите минимум для 4–5 точек.
- При сохранении результата, приложение вычислит матрицу 3×3.

ВАЖНО: исходя из координат на сцене, координаты точек для гомографии могут быть отрицательными.

9. **Запустите процесс детекции**
    - Перейдите на главную страницу приложения - http://127.0.0.1:5000/
    - Запустите процесс детекции людей;
    - Через несколько секунд запустите расчёт координат и их отправку на стенд

Параметры скрипта обнаружения людей

`conf-thres` (порог доверия детектора)

- Это **порог уверенности** (confidence), с которым модель считает, что найденный объект действительно присутствует.
- **Если** вы **повышаете** `conf-thres` (например, `0.6` или `0.7`), детектор будет «строже», отсекает малейшие неуверенные предсказания.
- Оптимально будет указать значение близкое к результату обнаружения в конкретном случае:
  - Включите обнаружение с отображением потока и низкой уверенностью;
  - Определите среднее значение корректных срабатываний (рамки вокруг людей) и ложных (рамки вокруг других объектов)
  - Укажите доверие близкое к среднему корректному

`iou-thres` (NMS IoU threshold)

- Используется для **объединения** перекрывающихся детекций одного и того же объекта.
- Если `iou-thres=0.45`, значит если два бокса перекрываются больше, чем на 45%, NMS оставит один.
- **Если** вы **повысите** `iou-thres` (до `0.6`), NMS станет «мягче» и будет **оставлять** больше близких боксов.
  - Может увеличить кол-во дублей, если объекты реально перекрываются.
- **Если** вы **понизите** `iou-thres` (до `0.3`), станет «жёстче» и будет сильнее сливать боксы.
  - Может привести к пропаданию близко стоящих объектов, если их боксы пересекаются чуть сильнее.
- Значения `0.4–0.5` наиболее типичны.

`img-size`: размер входных картинок для сети (`640` по умолчанию).

- Вы можете указать `img-size 1280` для более точной детекции, но будет **медленнее**. Если нужно быстрее — ставьте `320` или `480` (но точность детектора снизится).

`max-det` (максимум объектов на кадр)

- Это лимит на кол-во детекций, которые возвращает модель.
- Если у вас никогда не будет более 50 или 100 объектов в кадре, можно снизить (например, `max-det 100`). Это чуть ускорит пост-обработку.

Важные замечания:

- Для минимизации задержки между съемкой камерой и реакцией масок, используйте IP-Камеру с ethernet разъемом;

- При использовании длинных проводов (10 метров и более) - используйте качественные ethernet кабели с экранированием;

- При запуске детекции обратите внимание на логи в cmd.exe, в случае некорректной установки драйверов и библиотек, распознавание может начаться с использованием CPU вместо GPU;

  Распознавание через CPU приведет к накоплению кадров в буфере и скорому прекращению процесса;

  Маркером начала работы через CPU этого будет сообщение в консоли:

```
2025-01-29 03:17:08.4063671 [E:onnxruntime:Default, provider_bridge_ort.cc:1862 onnxruntime::TryGetProviderInfo_CUDA] D:\a\_work\1\s\onnxruntime\core\session\provider_bridge_ort.cc:1539 onnxruntime::ProviderLibrary::Get [ONNXRuntimeError] : 1 : FAIL : LoadLibrary failed with error 126 "" when trying to load "C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\onnxruntime\capi\onnxruntime_providers_cuda.dll"
2025-01-29 03:17:08.4142990 [W:onnxruntime:Default, onnxruntime_pybind_state.cc:993 onnxruntime::python::CreateExecutionProviderInstance] Failed to create CUDAExecutionProvider. Require cuDNN 9.* and CUDA 12.*, and the latest MSVC runtime. Please install all dependencies as mentioned in the GPU requirements page (https://onnxruntime.ai/docs/execution-providers/CUDA-ExecutionProvider.html#requirements), make sure they're in the PATH, and that your GPU is supported.
```

- Проводите новый расчёт матрицы гомографии и указывайте новые точки интереса при подозрении на нарушения положения камеры;

- Проект настроен для адаптации под различные залы, но требует ручной доработки в каждом конкретном случае.

- Включайте процесс расчета координат и отправки на стенд после полного включения скрипта детекции. Признаком успешной синхронизации процессов – появление сообщений в консоли

```
Sending JSON to 127.0.0.1:7777 → {"1": "-1037.00,549.00", "2": "920.00,782.00", "3": "81.00,-651.00"}

POI: [958, 1000], Person: [1056, 1042], Transformed: (81, -651)
```

Сообщение указывает, что второй скрипт успешно определил тех людей, на которых маскам следует обращать внимание и отправил json с этими данными на стенд со сценой.

Следуя данной инструкции, вы сможете настроить сцену и достичь необходимого эффекта наблюдения масками за людьми в зале.