Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент Варламова Екатерина Алексеевна
фамилия, имя, отчество

Группа ИУ7-41Б

Тип практики Технологическая

Название предприятия МГТУ им. Н. Э. Баумана

Студент _____ _____
подпись, дата *фамилия, и.о.*

Руководитель практики _____ _____
подпись, дата *фамилия, и.о.*

Оценка _____

2021 г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Разработать программный модуль, отвечающий за обнаружение объектов на взлётно-посадочной полосе, а также предоставить программный интерфейс для взаимодействия с модулем. Данный модуль рассматривается в качестве составного элемента WEB-приложения, решающего следующую задачу: обнаружить и классифицировать мусор на взлётно-посадочной полосе по видеоматериалам, взятым с камер аэропорта.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Ежедневно в мире совершается около 100 тысяч гражданских авиарейсов, однако каждый полёт обязательно связан с рисками и опасностями. Одним из источников опасности является наличие мусора на взлётно-посадочной полосе. Кроме того, объекты на ВПП могут нанести существенный материальный ущерб из-за задержки рейсов и повреждения самолётов.

Многие аэропорты решают проблему обнаружения и классификации объектов на ВПП путём привлечения к этому работников аэропорта, однако в таком случае ошибка, связанная с человеческим фактором, может стоить жизней многих людей. С развитием современных технологий стали появляться системы автоматического обнаружения и классификации мусора, которые позволили снизить риск возникновения опасных ситуаций в аэропорту. В большинстве случаев подобные системы базируются на использовании камер.

Целью данной практики является разработка WEB-приложения, которое позволит по видеоинформации с камеры аэропорта обнаружить и классифицировать мусор на ВПП. Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих алгоритмов обнаружения и классификации объектов по видеоинформации;
- разработать архитектуру приложения и схему взаимодействия модулей;
- разработать приложение в соответствии с выбранными архитектурой и алгоритмами обнаружения и классификации объектов;

1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ

1.1. Обзор существующих решений

1.1.1. FODetect

Автоматизированная система FODetect обеспечивает непрерывный мониторинг взлетно-посадочных полос аэропорта для быстрого удаления посторонних предметов с полосы в любых погодных условиях. FODetect использует сочетание радаров миллиметрового диапазона с изображениями в высоком разрешении для большей эффективности обнаружения. Из недостатков данной системы стоит отметить использование радаров на взлетной полосе. Радары излучают электромагнитные волны, что может нарушить функционирование других технических средств аэропорта и самолётов.

1.1.2. FOD Finder XM

Система предназначена для обнаружения посторонних предметов, находящихся в любом месте в пределах обозначенных для наблюдения границ аэродрома. Недостатками этой системы является использование радаров на ВПП и отсутствие в системе камер, что делает невозможным классификацию обнаруженных объектов, а также оперативную проверку обнаружения человеком.

1.2. Анализ задачи обнаружения объектов

Задача обнаружения объектов на ВПП сводится к выделению на изображениях пикселей, соответствующих объектам. При наличии последовательности кадров, отражающих появление объекта в поле зрения камеры, целесообразно использовать модели фона для обнаружения объектов. Выходными данными алгоритмов, использующих модели фона, являются бинарные маски. Из-за особенностей работы алгоритмов бинарные изображения подвержены шуму «соль и перец», поэтому возникает необходимость дополнительной обработки изображений. После обработки по бинарной маске строятся охватывающие рамки обнаруженных объектов.

1.2.1. Анализ методов обнаружения объектов

1.2.1.1. Опорный кадр

Идея этого метода заключается в том, что сначала определяется опорный кадр, то есть кадр без объектов (фоновый). Затем для каждого кадра видеопотока вычисляется разностное изображение, которое строится следующим образом: если модуль разности фонового пикселя и пикселя проверяемого изображения больше порогового значения, то в результат на место этого пикселя помещается 1, иначе 0. Пороговое значение задаётся заранее.

Данный метод очень прост, однако подвержен влиянию качества съёмки, изменения уровня освещения, что обуславливает появление большого количества шумов.

1.2.1.2. Усреднённый фон

Данный метод является некоторой модификацией предыдущего: теперь при вычислении разностного изображения для k-ого кадра используется среднее значение пикселя предыдущих n кадров (n задаётся заранее). Использование подобных моделей фона несколько снижает степень влияния шумов и освещения на результат обнаружения, однако результат все равно сильно зависит от качества условий съемки и аппаратуры.

1.2.1.3. Модель фона по Гауссу

Данный метод заключается в том, что значение точек фона колеблется с некоторым отклонением от определённого среднего значения. Если это отклонение больше порога, то область на изображении содержит новый объект. Часто изменение значений пикселей описывают с помощью нормального распределения.

Рассмотрим некоторую точку f(i, j) на серии из n кадров. Пусть $X = \{x_1 \dots x_n\}$ – множество значений, которые принимает этот пиксель в серии кадров. Тогда математическое ожидание и дисперсия распределения для выбранной

точки будут определяться выражениями (1) и (2), а плотность распределения — выражением (3).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2 \tag{2}$$

$$\rho(x|\mu,\sigma) = \frac{1}{\sigma 2\pi} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}} \tag{3}$$

Таким образом, модель фона задается набором нормальных распределений для каждого пикселя, построенных по серии кадров. Проводить подгонку функции плотности можно по последним n кадрам, чтобы сократить количество вычислений. Новые значения параметров распределения рассчитываются на основе текущих параметров по формулам (4) – (6).

$$\mu_k = \rho x_k + (1 - \rho)\mu_{k-1} \tag{4}$$

$$d = |x_k - \mu_k| \tag{5}$$

$$\sigma_k^2 = d^2\rho + (1 - \rho)\sigma_{k-1}^2 \tag{6}$$

Распределение пикселя обновляется только в том случае, когда этот пиксель классифицируется как фон. Пиксель относится к фону, если его текущая интенсивность лежит в пределах некоторого интервала математического ожидания его распределения, задаваемого порогом T.

1.2.1.4. Смесь нормальных распределений

Еще более усложненная модель фона предполагает его представление в виде смеси нормальных распределений. Для построения такого фона для любого пикселя кадра I_k используются значения его интенсивности на всех предыдущих

кадрах X = {x_1 ... x_k}. Вероятность того, что наблюдается значение x_k, может быть представлена в виде (7).

$$P(x_k) = \sum_{j=1}^{s} w_j^k N(x_k|\mu_j^k \sigma_j^k) \quad (7)$$

где w_j^k - вес j-го распределения Гаусса в смеси для k-го кадра;

μ_j^k - математическое ожидание j-го распределения Гаусса для k-го кадра;

σ_j^k - – среднеквадратичное отклонение j-го распределения Гаусса для k-го кадра;

$N(x_k|\mu_j^k \sigma_j^k)$ - плотность распределения j-ой компоненты в смеси;

s – количество компонент в смеси.

Компоненты в смеси сортируют в порядке уменьшения величины r_j^k(8).

$$r_j^k = \frac{w_j^k}{\sigma_j^k} \quad (8)$$

Такая сортировка предполагает, что пиксель фона отвечает распределению с большим весом и малой дисперсией. Первые B^k распределений смеси, удовлетворяющих условию (9), определяют распределения фоновых пикселей.

$$B^k = argmin\left\{\sum_{j=1}^{b} w_j^k > T\right. \quad (9)$$

При обработке нового кадра I_{k+1} для каждого пикселя с использованием расстояния Махаланобиса определяется распределение, которому соответствует данный пиксель (10). Если распределение найдено, то пиксель относится к тому классу (объект или фон), который соответствует найденному распределению. Если не удалось обнаружить ни одного распределения, которому принадлежал бы пиксель, то этот пиксель считается принадлежащим объекту.

$$\sqrt{\left(x_{k+1} - \mu_j^k\right)^T (\sigma_j^k)^{-1} \left(x_{k+1} - \mu_j^k\right)} \leq 2.5\sigma_j^k \quad (10)$$

После обработки очередного кадра видеопотока производится обновление параметров распределений и весов компонент в зависимости от того, было ли найдено соответствующее распределение для пикселя.

Если распределение было обнаружено, тогда весовые коэффициенты параметры этого распределения пересчитываются согласно формулам (11) – (14).

$$w_j^{k+1} = (1 - \alpha)w_j^k + \alpha \tag{11}$$

$$\mu_j^{k+1} = (1 - \rho)\mu_j^k + \rho x_{k+1} \tag{12}$$

$$(\sigma_j^{k+1})^2 = (1 - \rho)(\sigma_j^k)^2 + \rho(x_{k+1} - \mu_j^{k+1})^{T+1} \tag{13}$$

$$\rho = \alpha N(x_k|\mu_j^k \sigma_j^k) \tag{14}$$

где α — заданная константа.

Параметры остальных распределений в том случае остаются без изменений, а веса пересчитываются по формуле (15).

$$w_j^{k+1} = (1 - \alpha)w_j^k \tag{15}$$

Если для пикселя x_k не было найдено соответствующее распределение из смеси, то последнее распределение в виду введенного выше упорядочивания распределение из смеси заменяется на новое с другими параметрами. Математическое ожидание выбирается равным значению пикселя, дисперсия — максимально возможной, а вес — минимально допустимым.

Количество распределений в смеси определяется сложностью фона и имеющимися вычислительными мощностями. Как правило, используется 2-5 компонент

1.2.1.5. Сравнение методов обнаружения, использующих модель фона

Таблица 1 Сравнение методов обнаружения

	Скорость	Влияние шума	Влияние освещения
Опорный кадр	очень высокая	очень высокое	очень высокое
Усреднённый фон	средняя	среднее	среднее
Модель фона по Гауссу	высокая	среднее	среднее
Смесь нормальных распределений	средняя	низкое	низкое

Для выделения объектов на видеопотоке была выбрана модель фона в виде смеси нормальных распределений. Из всех рассмотренных моделей фона она меньше всего подвержена влиянию шумов и неравномерностей освещения.

1.2.2. Анализ методов обработки изображений

1.2.2.1. Взвешенная ранговая фильтрация

При данном подходе выбирается некоторая область, которая задаётся матрицей весов. Она накладывается на каждый пиксель изображения. Вес каждого пикселя показывает, сколько раз значение этого пикселя будет повторено. Далее все значения пикселей и их повторов упорядочиваются по возрастанию. Результатом считается значение элемента, стоящего на позиции r в упорядоченном наборе. R задаётся извне. В частности, при срединной фильтрации $r = (m + 1) / 2$, m - количество пикселей в области.

1.2.2.2. Морфологический фильтр

Входными данными метода математической морфологии являются два объекта: обрабатываемое изображение и примитив, зависящий от решаемой задачи. Базовые операции морфологии — эрозия и дилатация.

При выполнении дилатации примитив проходит по всем пикселям обрабатываемого изображения. В тех положениях, в которых центр примитива

совмещается с единичным пикселем исходного изображения, ко всему примитиву применяется логическое сложение с соответствующими пикселями исходного изображения. Результат сложения попадает в результат дилатации. Таким образом, дилатация расширяет объекты на исходном изображении. Конкретный вид и степень расширения определяется формой примитива.

При выполнении эрозии примитив так же проходит по всем пикселям исходного изображения. Пиксель в исходном изображении будет считаться единичным, только если все пиксели под примитивом равны единице, в противном случае он обнуляется. Эрозия приводит к уменьшению объектов на исходном изображении и удалению деталей, размеры которых меньше примитива.

Размыкание строится как эрозия исходного изображения по некоторому примитиву, результат которой затем подвергается дилатации по этому же примитиву. Замыкание строится как дилатация обрабатываемого изображения по некоторому примитиву, к результату которого применяется эрозия по этому же примитиву.

Морфологический фильтр строится на последовательном применении к исходному изображению сначала операции размыкания, а затем к результату — операции замыкания.

1.2.2.3. Сравнение методов обработки изображений

Таблица 2 Сравнение методов обработки изображений

	Эффективность	Скорость
Взвешенная ранговая фильтрация	-	+
Морфологический фильтр	+	-

Несмотря на то, что взвешенная ранговая фильтрация требует меньше проходов по пикселям изображения, чем морфологический фильтр, для обработки изображений был выбран морфологический фильтр, поскольку он более эффективно удаляет шумы.

1.3. Анализ архитектур приложения

При проектировании системы необходимо выбрать между монолитной и микросервисной архитектурой приложения. Рассмотрим достоинства и недостатки каждой из них.

1.3.1. Монолитная архитектура

Монолитная архитектура предполагает наличие единой платформы, на которой сконцентрированы все компоненты приложения. Такой подход имеет ряд преимуществ:

- быстрая разработка за счёт более простого межмодульного взаимодействия;
- простота реализации;
- простой запуск программы.

Однако у монолитной архитектуры есть ряд существенных недостатков:

- при обновлении одного из модулей возникает необходимость разворачивать заново всё приложение;
- возникновение ошибки в одном из модулей может привести к остановке всего приложения, а не одного компонента;
- «перегруженность» кода за счёт нескольких взаимосвязанных компонентов, границы между которыми размыты;

Таким образом, монолитная архитектура подходит для быстрой разработки небольших продуктов, долгосрочная поддержка и модификация которых не предполагается.

1.3.2. Микросервисная архитектура

Микросервисная архитектура предполагает наличие нескольких платформ, на каждой из которых разворачивается небольшое приложение, соответствующее определённому компоненту программного продукта. Эти приложения работают на разных серверах и взаимодействуют друг с другом по сети.

К преимуществам данной архитектуры можно отнести:

- обновление одного сервиса не затрагивает другой;
- чёткое разграничение ответственности и сохранение модульности;
- возможна разработка сервисов на разных языках;
- ошибка в одном сервисе не приведёт к остановке всего приложения;
- возможность масштабирования модулей независимо от других;

К недостаткам можно отнести:

- медленная разработка за счёт усложнения межмодульного взаимодействия;
- сложности при запуске программы.

1.3.3. Сравнение монолитной и микросервисной архитектур

Таблица 3 Сравнение монолитной и микросервисной архитектур

	Монолитная	Микросервисная
Быстрая и простая разработка	+	-
Чёткое разграничение ответственности каждого модуля	-	+
Обновление одного модуля не затрагивает другой	-	+
Ошибка в одном модуле не приведёт к остановке приложения	-	+
Разработка модулей на разных языках	-	+
Простота запуска	+	-

В качестве архитектуры приложения была выбрана микросервисная архитектура, так как она является более гибкой по сравнению с монолитной и имеет более существенные преимущества.

Выводы

В результате анализа:

- для выделения объектов на видеопотоке была выбрана модель фона в виде смеси нормальных распределений, так как из всех рассмотренных моделей фона она меньше всего подвержена влиянию шумов и неравномерностей освещения;
- для обработки изображений был выбран морфологический фильтр, поскольку он наиболее эффективно удаляет шумы;
- в качестве архитектуры приложения была выбрана микросервисная архитектура, так как она является более гибкой по сравнению с монолитной и имеет более существенные преимущества.

2. КОНСТРУКТОРСКИЙ РАЗДЕЛ

2.1. Общая архитектура приложения

В результате анализа для решения задачи практики была выбрана микросервисная архитектура. Диаграмма компонентов приложения выглядит следующим образом:

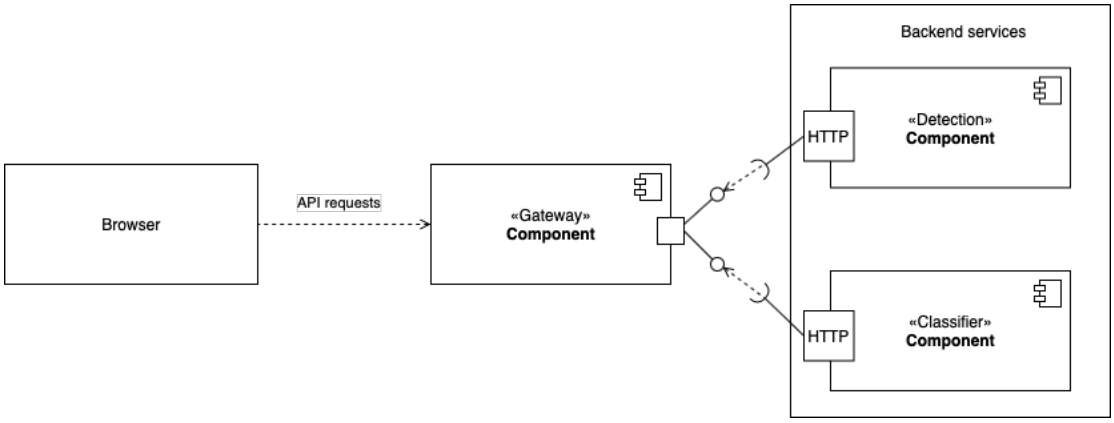

Рисунок 1 Диаграмма компонентов приложения

На диаграмме компонентов фигурируют браузер (интерфейсная часть приложения) и 3 сервиса:

1. browser (интерфейс) отображает данные в виде, понятном пользователю, а также отправляет видеоматериалы, полученные от пользователя, на сервис Gateway;
2. сервис Gateway отвечает за координацию запросов и подготовку данных для остальных сервисов и интерфейса;
3. сервис Detection получает изображения от сервиса Gateway и возвращает ему фрагменты кадра, на которых были обнаружены объекты;
4. сервис Classification получает фрагмент кадра с обнаруженным объектом от сервиса Gateway и возвращает ему класс этого объекта;

Сервисы общаются друг с другом посредством HTTP-запросов с поддержкой REST стиля.

2.2. Описание этапов работы сервиса обнаружения объектов

В аналитическом разделе были рассмотрены алгоритмы обнаружения объектов, основанные на использовании моделей фона, а также алгоритмы обработки бинарных изображений для удаления шумов. В результате анализа для обнаружения был выбран алгоритм, использующий модель фона на основе смеси нормальных распределений, а для обработки – морфологический фильтр.

Процесс обнаружения объектов на ВПП состоит из нескольких последовательных этапов:

1. Инициализация модели фона «чистыми» кадрами;
2. Работа алгоритма обнаружения, использующего модель фона на основе смеси нормальных распределений; в результате работы получают бинарную маску изображения, на которой единичные пиксели соответствуют предполагаемым объектам, нулевые – фону, однако бинарная маска может содержать шумы.
3. Работа алгоритма морфологического фильтра, результатом которого является изображение с уменьшенными шумами.
4. Определение контуров единичных областей с последующим построением минимальной выпуклой оболочки.
5. Выделение прямоугольных областей с объектами, реализованное на основе построенных оболочек.

Тогда обнаружение объектов на ВПП можно представить так:

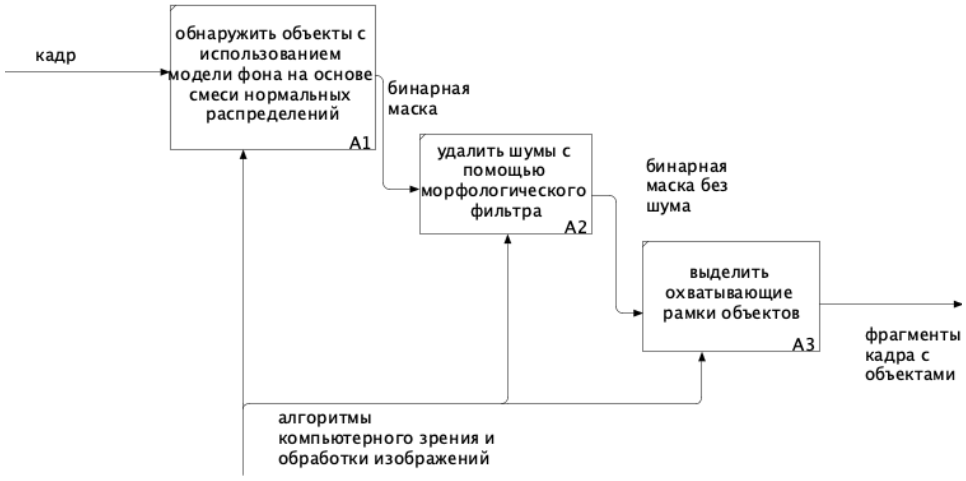

Рисунок 2 IDEF0-диаграмма декомпозиции обнаружения объектов

2.3. Структура модуля обнаружения объектов

Структура модуля выглядит следующим образом:

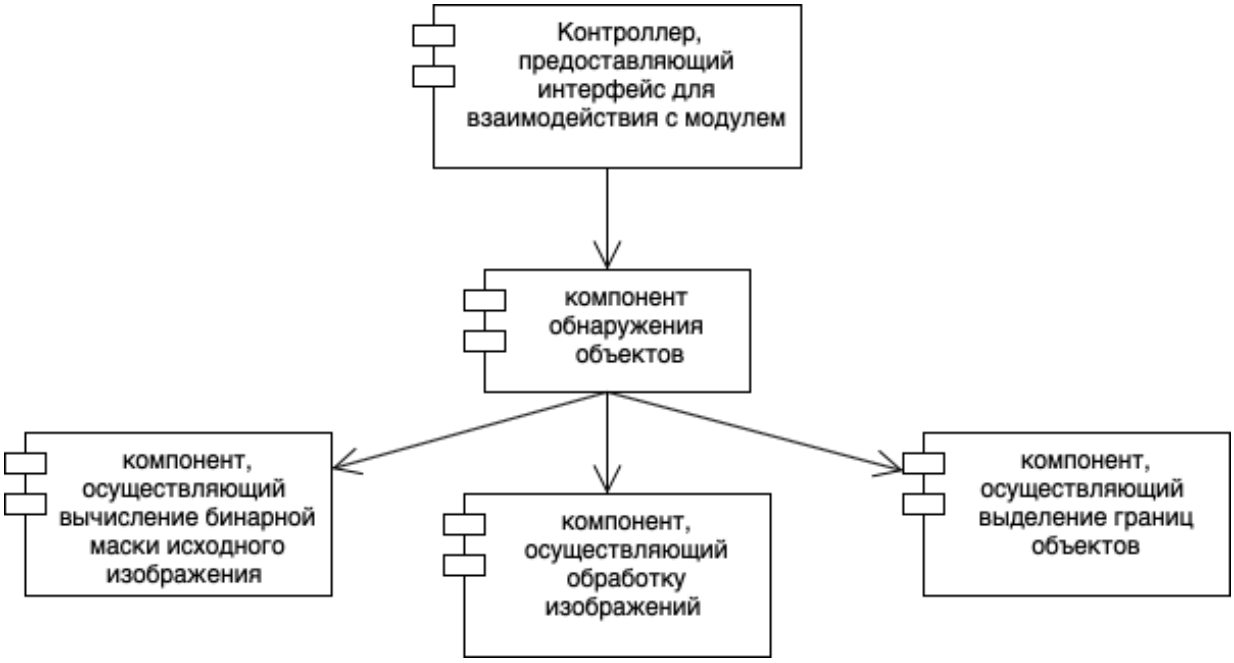

Рисунок 3 Структура модуля обнаружения объектов

Выводы

В данном разделе была приведена архитектура всего ПО и указано место модуля обнаружения в нём, кроме того, были описаны основные этапы работы данного модуля с учётом выбранных в аналитическом разделе алгоритмов.

3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

3.1. Выбор языка программирования

Для программной реализации сервиса обнаружения был выбран язык программирования Python как наиболее удобное средство для создания серверной части приложения. Основные его архитектурные черты – это динамическая типизация, автоматическое управление памятью, механизм обработки исключений, высокоуровневые структуры данных. Немаловажно, что Python поддерживает объектно-ориентированное программирование.

К плюсам языка Python относятся его простота, большое количество библиотек для использования в самых различных областях. Основным минусом языка является низкая скорость выполнения программ по сравнению с низкоуровневыми языками.

3.2. Использование сторонних библиотек и фреймворков

Для реализации межмодульного взаимодействия был использован фреймворк Flask как наиболее простое и удобное средство для создания веб-приложений. С его помощью сервисы общаются между собой посредством HTTP-запросов. Кроме того, для отправки HTTP-запросов используется библиотека requests.

Метод обнаружения объектов подразумевает использование нескольких методов компьютерного зрения. В целях ускорения разработки и использования наиболее эффективных реализаций методов было решено использовать библиотеку OpenCV. OpenCV предоставляет более 2500 оптимизированных алгоритмов, которые включают полный набор как классических, так и самых современных алгоритмов компьютерного зрения, машинного обучения и обработки изображений. Эти алгоритмы могут использоваться для различных целей, включая задачи обнаружения объектов.

3.3. Структура разработанного модуля

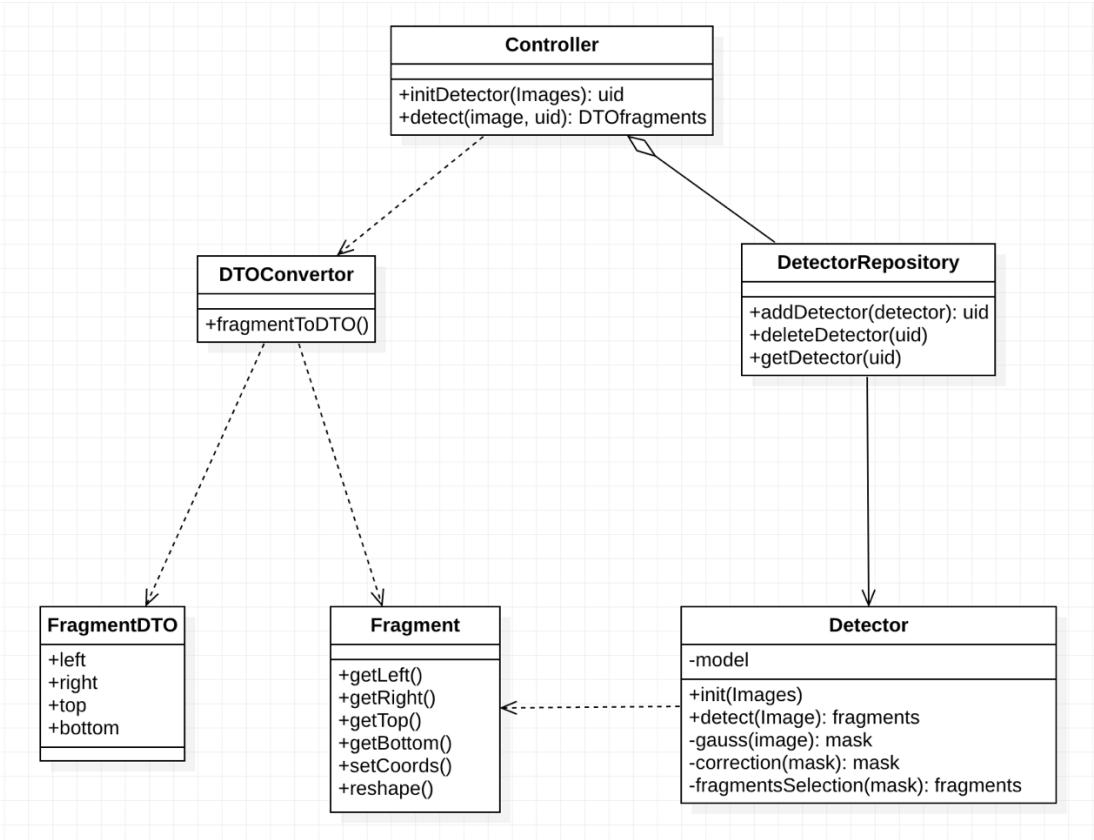

Рисунок 4 Структура разработанного сервиса детекции

На рисунке 5 представлена диаграмма классов программного модуля, реализующего обнаружение объектов.

Компонет Controller реализует программный интерфейс для работы с модулем. Здесь происходит извлечение данных из HTTP-запросов, вызов необходимых методов, реализующих логику, и подготовка данных для ответа на запрос.

Класс Detector реализует основную логику работы сервиса. Экземпляр этого класса требуется сначала инициализировать, передав в метод init коллекцию кадров для определения начальных параметров фона и созданию объекта model. Основную обработку кадров осуществляет метод detect, принимающий на вход изображение и возвращающий фрагменты кадра. Метод detect вызывает последовательно методы, реализующие обнаружение объектов на кадре. Сначала вызывается метод gauss, принимающий изображение, в котором происходит обнаружение объектов на кадре. Этот метод использует

инициализированную модель фона и возвращает бинарную маску изображения. Далее бинарная маска повергается обработке с помощью морфологического фильтра, реализованного в методе correction. После этого на бинарной маске выделяются контуры единичных областей в методе fragmentsSelection и на основе них выделяются прямоугольные области с объектами. Далее найденные прямоугольные области (или фрагменты) возвращаются из метода detect.

Класс DetectorRepository реализует хранилище используемых детекторов: при добавлении детектора этому детектору присваивается уникальный идентификатор, и пара id – detector помещается в хранилище, а идентификатор возвращается из метода добавления. Получение и удаление детектора используют этот идентификатор. Это хранилище необходимо по той причине, что одновременно могут использоваться несколько детекторов.

Класс Fragment хранит в себе координаты левого верхнего и правого нижнего углов фрагмента кадра. FragmentDTO используется для отправки результатов детекции. Разбиение на 2 класса одной сущности мотивировано тем, что в процессе эволюции сервиса внутренняя структура фрагмента может измениться, и тогда при наличии классов FragmentDTO и DTOConvertor интерфейс всего модуля не изменится.

3.4. Программный интерфейс для взаимодействия с модулем

POST /api/v1/detection/initmodel

Описание: в теле запроса передаётся несколько «чистых кадров», которыми инициализируется модель. Ответом на данный запрос является идентификатор инициализированного детектора.

POST /api/v1/detection/detect

Описание: в теле запроса передаётся идентификатор детектора и кадр, на котором необходимо обнаружить объекты. Ответом является массив координат охватывающих рамок.

Формат получаемых и отправляемых данных: json (изображения кодируются в base64 для удобства передачи).

Выводы

В соответствии с выбранной конструкторском разделе структурой ПО был реализован модуль обнаружения посторонних объектов на ВПП. Был описан интерфейс модуля и приведена его структура классов.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработан модуль обнаружения посторонних объектов на ВПП в рамках общей задачи разработки WEB-приложения, позволяющего обнаружить и классифицировать мусор на взлётно-посадочной полосе по видеоматериалам, взятым с камер аэропорта. В ходе работы были выполнены все поставленные задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, Pentland Alex Paul. Pfinder: Real-Time Tracking of the HumanBody // IEEE Transactions on pattern analysis and machine intelligence, Vol. 19(7). — 1997.
2. Pakorn KaewTraKulPong, Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection // Video-Based Surveillance Systems, 2002.
3. Обработка изображений. Часть 1 [Электронный ресурс] – Режим доступа: http://www.graph.unn.ru/rus/materials/CG/CG03_ImageProcessing.pdf (Дата обращения 10.07.2021).