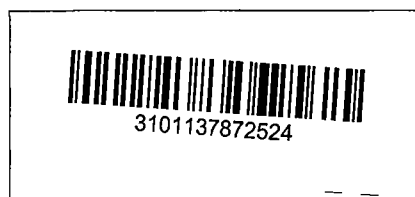




ИЗУМРУД СТУДЕНТ

ИАДА АЛ С ЕД АЛ УН С



Титульный лист

Направление Естественные науки Инженерные науки
 Математика и информатика Социальные и
 Экономика и управление гуманитарные науки

Вариативный блок 1 2 3 4 5

Курс 1 2 3 4 5 отсутствует

Фамилия А Р Г У Н О В

Имя Д М И Т Р И Й

Отчество А Н А Т О Л Ь Е В И Ч

Дата рождения 2 0 0 2 2 0 0 4

Город участия Е К А Т Е Р И Н Б У Р Г

Аудитория 2 2 8

Дата 0 2 0 2 2 0 2 6

Подпись

Пример заполнения
 А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф
 Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0



ИЗУМРУД СТУДЕНТ
И ШИДАУ АЛЬ ЕДЕ АЛЬН ОУНИ РС 1



3101137872524

Проверочный лист

Заполняется участниками

Направление Естественные науки Инженерные науки
 Математика и информатика Социальные и
 Экономика и управление гуманитарные науки

Вариативный блок 1 2 3 4 5

Курс 1 2 3 4 5 отсутствует

Город участия **Е К А Т Е Р И Н Ъ Б У Р Г**

Заполняется организаторами

Количество доп листов Количество черновиков к проверке

Время выхода с до

Протокол проверки

Заполняется жюри

Номер задания	1	2	3	4	5	6	7	8	9	10
Балл члена жюри №1	<input type="checkbox"/> 5	<input type="checkbox"/>	<input type="checkbox"/> 50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Балл члена жюри №2	<input type="checkbox"/> 10	<input type="checkbox"/>	<input type="checkbox"/> 50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Итоговый балл 50

Подпись члена жюри №1

Подпись члена жюри №2

Пример заполнения

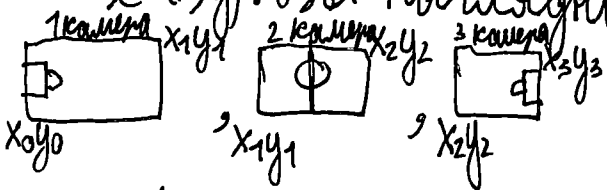
А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф
 Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0



Блок 3. Информационные системы и технологии Анализ активности футболиста

Пункт 1. Итак у нас 3 камеры с обзором поля с разных ракурсов. Пусть частота обновления кадров равняется 60 Гц, а с 1 секунды видео мы имеем 60 изображений. Теперь предлагаю рассмотреть следующий алгоритм распознавания игроков с 3х камер:

1. Для начала мы знаем, что камеры смотрят на 3 части поля, каждая на свою часть, судя по примерам ракурсов камер, также нам известно что с камеры видно боковую линию. Отсюда предлагаю оцифровать поле для его представления в координатах (x, y) . Вот нам нужен пример относительно 3х камер:



Т.е. у первой камеры ее координаты x_0y_0 (начало) и x_1y_1 (конец), а для 2ой камеры x_1y_1 это начало. Таким образом у нас итоговый размер поля от x_0y_0 до x_3y_3 . Но нам нужно рассчитывать изменение положения игрока и сообщ. он в определенный момент времени может быть как виден как на одной камере, так и переключаться на другую.

2. Чтобы распознать игрока на поле предлагаю использовать нейронную сеть а.к.а модель YOLOv8 Instance Segmentation. Именно Instance Segmentation нужен, чтобы найти и идентифицировать маски для каждого отдельного игрока на поле, т.к. по сути каждый игрок у нас уникальный объект. Чтобы улучшить качество выданной модели можно использовать модель Segment Anything для абстрактных кадров с видео и приложение Computer Vision Annotation Tool для редактирования разметки.

3. Таким образом алгоритм схематически выглядит так, отнеси
 только одной камере (но по сути для каждой камеры отдельно);

1. На вход в скрипт на Python с использованием библиотек
 OpenCV и Ultralytics (+ можно добавить эфалсхему для
 связи с БД, но про БД распишу позже).

2. Видеопоток нарезается на кадры.

3. Каждый кадр сохраняется и передается в модель
 иммеренс модели YOLOv8, на выходе получаем картинку
 с разметкой каждого игрока на ней и соответственно
 и координаты относительно размеров картинки. Вот тут
 то и нужны $x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3$, чтобы зная координаты
 относительно одного участка поля в заданный момент
 времени, можно было найти эти координаты относительно
 всего поля, т.к. игрок может переместиться с одной части поля
 на другую и для нас это важно для замеров характерис-
 тик игрока.

4. Данные в виде: путь к кадру картинки, распознанные
 игроки, их координаты относительно участка поля, их
 координаты относительно всего поля.



На рис 1 демонстрация того как у нас захва-
 тываются координаты игрока, т.е.
 (x'_0, y'_0) начало осями и (x_0, y_0, x_1, y_1) это
 его координаты и (x_0, y_0, x_1, y_1) это координаты
 участка поля и можно нормализовать коор-
 динаты игрока от участка поля
 и тогда относительные коор. всего поля для последующего
 удобства расчета его скорости.

Пункт 2. Систему координат я всё объяснил, теперь перейдем к
 физике, а именно к расчету скорости. Т.к. поле большое, то можно
 представить положение игрока в виде материальной точки, для этого

нужно провести

следующее преобразование координат шара: $(\frac{x'_1 - x'_0}{2}, \frac{y'_1 - y'_0}{2})$, так мы найдем центр ~~о~~ шара. Затем переходим в

двумерной системе координат и зная, что у нас 60 кадров в секунду, мы рассчитываем скорость шара в заданный момент в текущий момент времени. По координатам считаем ^{прямое} расстояние по модулю (т.к. шар летит как вектор от начала координат к концу, так и наоборот) и затем делим на кол-во кадров, т.е. 60 и получаем среднюю скорость в заданный ~~но~~ текущий момент времени. Т.к. для выносимости нам нужно посчитать

среднее минимальное распределение ~~скр~~ скорости в течение 30 мин, то для начала мы возьмем из БД данные по $v_{ср}$ (средней скорости за каждую секунду и рассчитаем $v_{ср}$ минимально. Т.е. берем запись из БД для $t=1с, t=2с, \dots, t=60с$, суммируем скорости и делим на 60 и получаем среднюю скорость в течение минуты. И так повторяем 30 раз, т.е. условно берем средние значения по времени с 1 по 60с это первая минута, далее с 61с по 120с и т.д.

Итого для средней скорости шара формула: $\frac{|S_{60}|}{60}$, т.е. модуль измеренная расстояния за 60 кадров, деленный на кол-во кадров. Для выносимости формула: $\frac{|S_{60_1}|}{60} + \frac{|S_{60_2}|}{60} + \dots + \frac{|S_{60_{60}}|}{60}$, это ср.

~~$\frac{|S_{60}|}{60}$~~ $\frac{|S_{60_1}|}{60} + \frac{|S_{60_2}|}{60} + \dots + \frac{|S_{60_{60}}|}{60}$, это ср.

~~скорость для одной минуты~~

это ср скорость для одной минуты, нам нужно посчитать их по отрезкам, как я написал выше.

2) 155

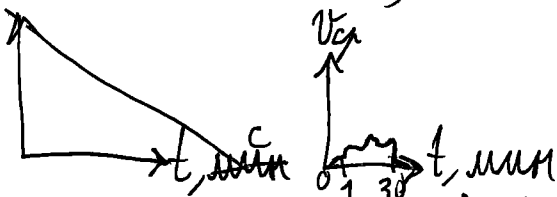


График на рис. 2 можно строить при помощи Vektor и scatterplotli.

рис 2 - пример графика выносимости,

Пункт 3. Что касается скелета данных архитектуры системы, то она мне видится следующим образом:

Входной видеопоток

Синхронизация кадра

Выходной видеопоток

Обработка кадров

Преобразование кадров в цифровой вид

Сохранение данных в БД

3) 20 б

Итого 50 б

Это пожалуй самый сложный этап, как мне кажется

P.S. Хотя YOLOv8 может и видео принимать на вход.

Тут по сути мы получаем сырые координаты местоположения

Это я описывал в п.1.

По сути программа может представлять собой либо CLI-приложение на Python, либо веб-приложение. На вход мы всегда получим видеопоток с 3х камер, на выходе БД, с которыми по какой-то схеме.

Про структуру БД. Предлагаю ~~лучше~~ рассмотреть упрощенную версию. Самая важная часть это таблица с данными по передвижению:

players_moving	timestamp: datetime
	user_id: unique PK
	current_coord: [x0, y0, x1, y1]
	coord_field: [x0, y0, x1, y1]
	field_curr: int

timestamp - это время для временная метка
 user_id - уникальный номер игрока
 current_coord - текущее местоположение игрока
 coord_field - координаты текущего участка на поле
 field_curr - номер участка на поле (1, 2 или 3)

users	user_id: PK
------------------	------------------------

users	user_id: PK unique
	user_name: str
	user_team: str
	user_team_id: PK
	user_speed: float
	user Stamina: float

Т.к. вычисление скорости и возможности то их можно получить через таблицу 'players_moving' и затем игроку в таблице 'users' вписать их.

