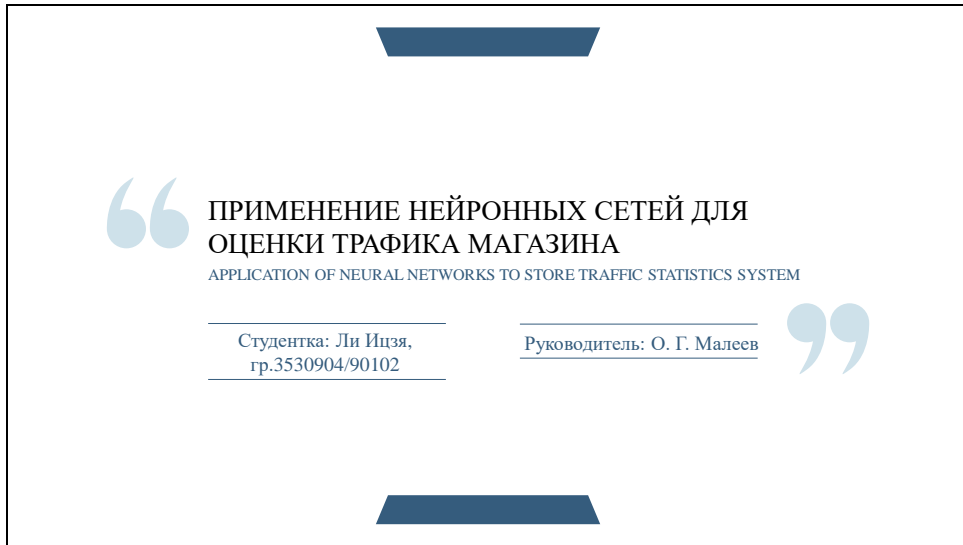


幻灯片 1

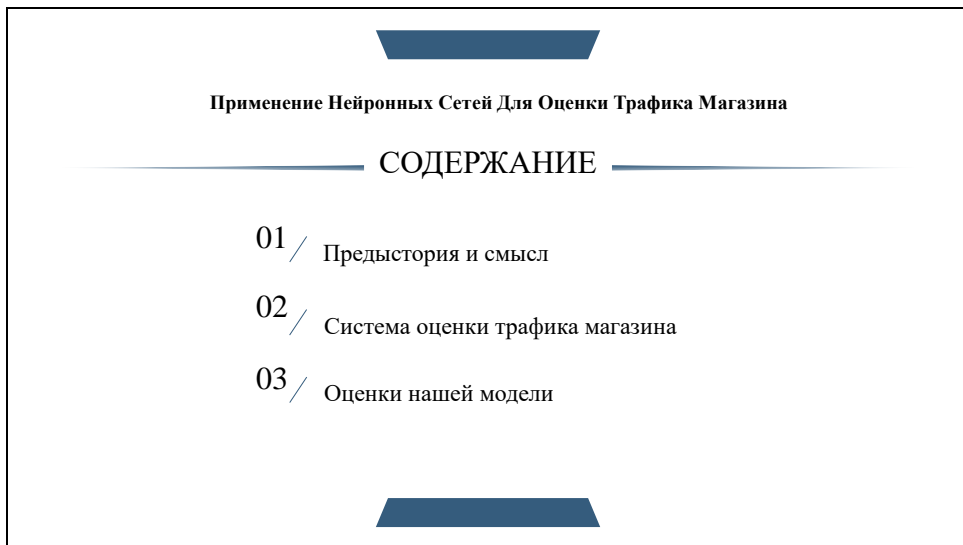


“ ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ
ОЦЕНКИ ТРАФИКА МАГАЗИНА
APPLICATION OF NEURAL NETWORKS TO STORE TRAFFIC STATISTICS SYSTEM ”

Студентка: Ли Ицзя,
гр.3530904/90102

Руководитель: О. Г. Малеев

幻灯片 2



Применение Нейронных Сетей Для Оценки Трафика Магази́на

СОДЕРЖАНИЕ

01 / Предыстория и смысл

02 / Система оценки трафика магазина

03 / Оценки нашей модели

Я расскажу об этом со трех сторон.

1. Нужно знать какой смысл есть для этого ислндования
2. Я представлю нашу систему
3. Буду показать оценки нашей модели

幻灯片 3

PART 01

Предыстория и смысл

Background and significance of topic selection

幻灯片 4

Предыстория и смысл

Background

Сегодня на рынке конкуренция очень ожесточена, и продавцам необходимо принимать ряд мер для улучшения продаж. Одной из ключевых проблем является то, как лучше понять поведение потребителей, чтобы оптимизировать количество клиентов?



Конфликт

Традиционные методы:

1. Неточны и ненадежны
2. Невозможно точно определить время посещения посетителя в магазине
3. Невысокая пропускная способность

Для магазина трафик покупателей в магазине считается одним из наиболее важных показателей, оказывающим воздействие на объем продаж. Но время посещения магазина покупателями распределяется неравномерно. И так появится вопрос : Как лучше понять поведение потребителей, чтобы оптимизировать количество покупателей в разное время? Традиционные методы подсчета количества посетителей часто неточны и ненадежны. 【счетчик, автоматический штатив, инфракрасные датчики】 . Но с помощью этих методов невозможно точно узнать время, когда каждый покупатель заходит в магазин.

Предыстория и смысл

Основная проблемы

Необходимо распознавать время входа и выхода каждого клиента - разработка **новой системы подсчета потока клиентов**

Наше решение

Система подсчета трафика клиентов:

1. Основана на технологии **нейронных сетей**
2. Вычислить **время пребывания** каждого клиента в магазине
3. Статистическое **распределение времени** пребывания всех клиентов



чтобы преодолеть эти недостатки, мы введем систему на основе нейронной сети – система оценки трафика магазина.

Наша система состоит из нескольких камер и сервера, на котором работает модель нейронных сетей.

我们的系统由多个摄像机和服务器组成，在服务器上运行神经网络模型。

Устанавливаем камеры в интересующих отделах магазина – например ввход, выход.

我 们把摄像机安装在感兴趣的商店的区域。

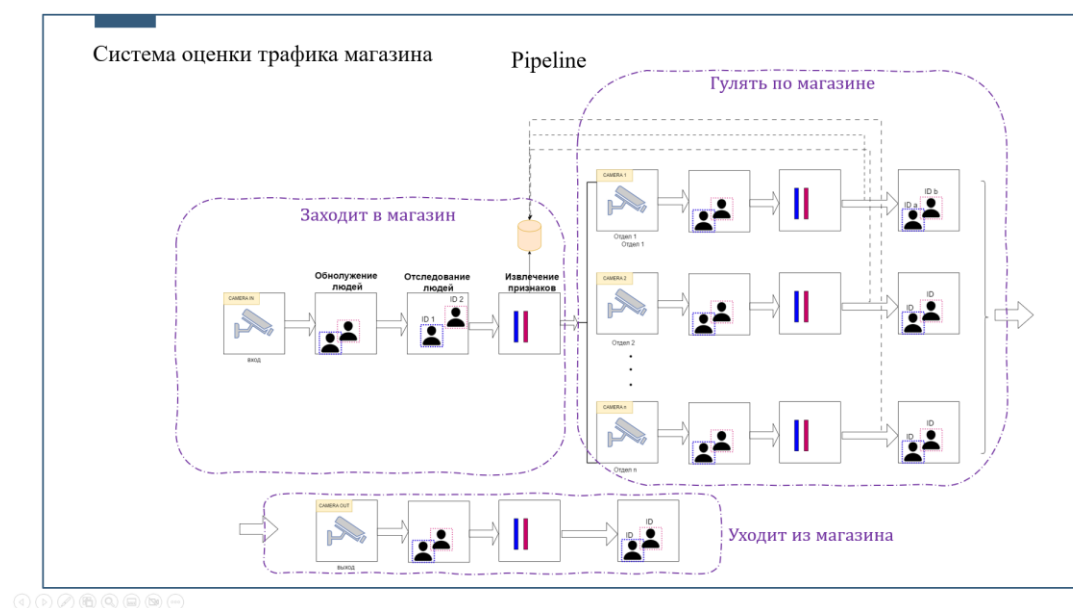
Запускаем все камеры одновременно, и эти камеры будут записывать видеопоследовательности. На нашей модели проходить обработки всех видеопоследовательностей. В результате будет создать отчет, где записано в какое время, какой покупатель заходит в магазина, проходит какие отделы и время ухода.

我们同时启动所有摄像头，这些摄像头将记录视频序列。在我们的模型上，通过所有视频序列的处理。结果，将创建一个报告，其中记录了哪个买家在什么时间经过了商店的哪个部门。

PART 02 Системы оценки трафика магазина Retail traffic statistics system

Теперь давайте посмотрим на общую архитектуру этой системы.

7



Далее мы опишем по временной порядку поведение системы. То есть покупатель входит в магазин, проходит через несколько отделов и, наконец, уходит.

接下来，我们按时间顺序描述当顾客进入商店、经过商店的几个区域并最终离开时系统的行为。

Преобразуем записанные камерами видео в кадры для обработки.

让我们将摄像机录制的视频转换为帧进行处理。

【При входе】

Когда в кадре появляется человек, модель YOLO обнаружит пешехода и вернет положение всех пешеходов в кадре, а также присвоит каждому пешеходу идентификатор.

当人出现在帧中的时候，YOLO 模型会检测行人并返回帧中所有行人的位置，同时为每个行人分配 ID。

Получили несколько изображений, где все пешехода с рамками.

我们得到了几张图像，其中所有行人都带有框架。

Затем модель DeepSORT with FastReID генерирует траектории для каждого пешехода на основе полученных изображений.

然后 deepSORT 模型基于这些结果图像生成每个行人的轨迹。

Используем структуру данных для записи идентификатора и траектории пешеходов.使用对应的数据结构记录行人的 id 和轨迹。

С этими траекториями будем определить, входил ли пешеход в магазин.

我们使用这些轨迹判定行人是否进入商店

[2]

Если пешеход входит в магазин, наша модель зафиксирует время входа, захватит изображение данного человека и через модель ReID сгенерирует 512-мерный вектор признаков на основе его характеристики. К характеристикам пешеходов относятся цвет одежды, цвет волос, поза тела и т. д., которые можно задавать и регулировать в модели ReID.

假如行人进入了商店，我们的模型就会记录入店时间，截取这个人物图像，并且根据他的特征通过 ReID 模型生成 512 维的特征向量。行人的特征包括衣服的颜色，头发的颜色，身体姿态，有无手提包等，这些都可以在 ReID 模型中进行设置和调整。

Сохраним векторы признаков всех покупателей, входящих в магазин, которые позже будут использоваться для сравнения и запроса.

保存所有进店顾客的特征向量，用于后面的对比查询。

[3]

Теперь покупатели вошли в магазин и прошлись по разным отделам.

现在顾客已经进入商店并在各个区域之间逛。

Когда они проходят под камерой система извлечет изображения человека, преобразует в вектор признаков и через модель ReID запросит ранее сохраненные вектора признаков, найдет все следы человека и добавит созданные новые следы под конкретной камерой, до тех пор, пока покупатели уходят из магазина.

当他们从摄像头下方经过时，系统会提取人的图像，通过 ReID 模型将其转换为特征向量，并查询之前保存的特征向量，找到人的所有足迹，并添加创建的新足迹在特定的摄像机下，直到顾客离开商店。



Давайте покажу 2 сцены системы. Это покупатели входят в магазин и уходят из него.

Сценарий 1 реализует обнаружения входа в магазин пешеходов.

В частности, программа использует модель обнаружения целей YOLOv5.

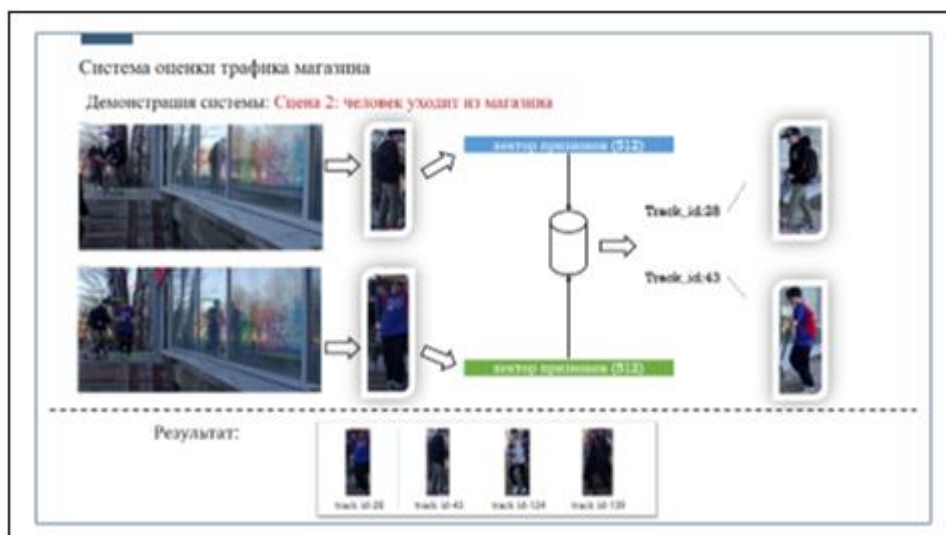
场景一实现了一个入店检测系统，其主要功能是检测摄像头拍摄的视频中是否有人进出商店，并对每个进店的人进行图像截取和保存。

具体地，该程序使用 YOLOv5 目标检测模型和 DeepSORT 目标跟踪模型，通过对视频帧进行处理得到每个人的位置信息。

При обнаружении того, что человек входит, программа вырежет изображение человека и сохранит его в папку, преобразует его в вектор признаков, и в то время присвоивает человеку индентификатор. Сохраним все вектора и индентификаторы.

Кроме этого, программа отобразит количество людей, входящих и выходящих из магазина, а также последнюю информацию о входе в магазин на видеокадре.

当检测到某人进入时，程序会抠出该人的图像，并将其保存到文件夹中以备后续使用。最后，程序会在视频帧上显示出入商店的人数和最新的进店信息，并输出相应的日志信息。



Сценарий 2 реализует обнаружения выхода из магазина. Симметрично предыдущей сцене, когда обнаруживается, что кто-то уходит, программа извлекает изображение человека и преобразует его в вектор признаков, а затем использует модель Рида для сопоставления всех покупателей, входящих в магазин, до тех пор, пока не будет найден тот же человек.

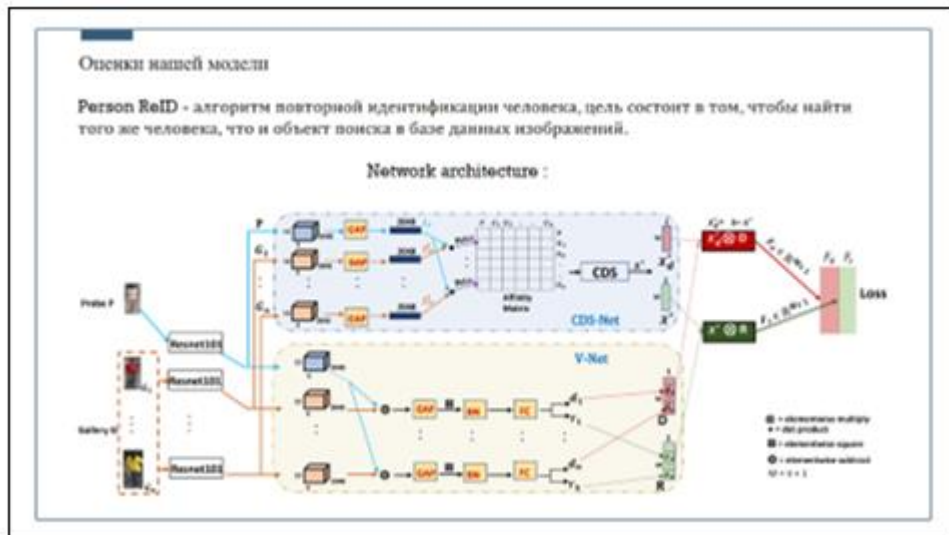
Наконец, программа отобразит количество людей, выходящих из магазина, и последнюю информацию о выходе из магазина на видеокадре.

场景 2 实现了一个出店检测系统，其主要功能是检测摄像头拍摄的视频中是否有人离开商店，并对每个离店的人进行图像截取和保存。与前面的场景是对称的，当检测到某人离开时，程序会抠出该人的图像并转化为特征向量，并运用 reid 模型匹配入店的所有顾客，直到找到同一个人。最后，程序会在视频帧上显示出离店的人数和最新的离店信息，并输出相应的日志信息。

10



11



Сначала я хотела бы показать важную используемую модель: ReID.

在介绍我们程序的性能指标之前, 我想先展示一个使用到的重要模型: ReID。

Person ReID - алгоритм повторной идентификации человека, цель состоит в том, чтобы найти того же человека в базе данных изображений, которые получили через несколько камер.

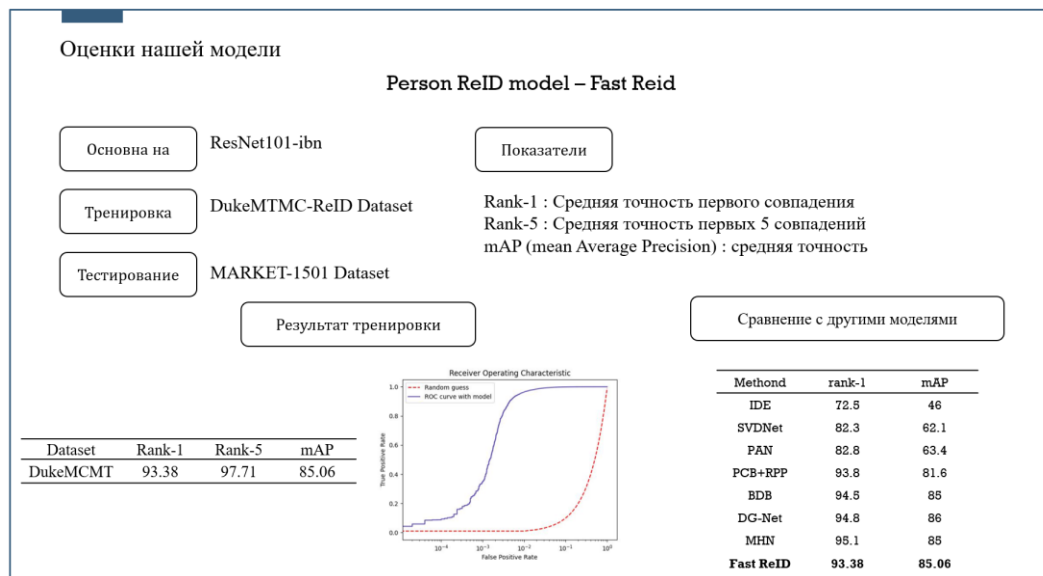
Person ReID 是一种人物重识别算法, 目标是在图像数据库中找到与搜索对象相同的人。

Эта модель играет большую роль в сопоставлении пешеходов между кадрами при отслеживании траектории людей.

这个模型在轨迹跟踪时对前后帧行人匹配起到很大的作用。

Чтобы сделать алгоритм отслеживания более разборчивым, мы используем более выразительную модель повторной идентификации человека FastReID.

为了使多目标跟踪算法具有更强的鉴别力, 我们采用了表达能力更强的行人重识别算法 FastReID。



Мы обучили собственную сеть Fast ReID.

我们训练了自己的阅读网络

Fast reid основа на сети ResNet, 101 слоев. Для тренировки выбрали базу данных DukeMTMC, которая состоит из 8 камер и 1500 пешеходов в университете Duke. А для тестирования выбрала MARKET-1501, где 6 камера и 750 человек.

После тренировки получили обученную модель. Видим что при тренировке rank-1 достичь 93.3, то есть точность первого раза совпадения – 0.93. mAP – 85.06, это средняя точность всей базы данных.

Также сделали сравнение с другими моделями. Видим что fastreid работает достаточно хорошо.

[Rank-1, Rank-5, mAP и mINP являются метриками, используемыми в оценке производительности систем распознавания объектов.

Rank-1 - это метрика, которая измеряет долю правильных ответов системы, когда самый вероятный предсказанный класс совпадает с настоящим классом тестируемого объекта. То есть, если система правильно распознала объект, этот объект будет иметь ранг 1.

Rank-5 - это метрика, которая измеряет долю правильных ответов системы, когда истинный класс тестируемого объекта находится среди пяти наиболее вероятных классов, предсказанных системой.

mAP (mean Average Precision) - это среднее значение точности для всех классов. Она определяется как среднее арифметическое точности (precision) при поиске объектов каждого класса. mAP широко используется в задачах поиска объектов.]

Оценки нашей модели							
model DeepSORT with Fast ReID							
Тренировка	MARKET-1501 Dataset			Показатели			
Тестирование	MOT16 Dataset			<p>MOTA : описывает качество построения траекторий людей MOTP : точность определения положения людей на кадрах видеопоследовательности. IDF1 : Возможность сохранения ID пешеходов IDs : Количество изменений ID пешеходов MT : наиболее отслеживаемые цели ML : наиболее пропущенные цели</p>			
Algorithm	MOTA ↑	MOTP ↑	IDF1 ↑	MT / % ↑	ML / % ↓	IDs / % ↓	FPS / Hz ↑
SORT	59.8	79.6	53.8	25.4	22.7	1423	8.6
DeepSORT	61.4	79.1	62.2	32.8	18.2	781	6.4
JDE	64.4	-	55.8	35.4	20	1544	18.5
Ours	66.2	80.8	65.8	35.3	17.6	760	5.8

[Мы извлекаем подмножество данных о пешеходах из крупномасштабного общедоступного набора данных обнаружения целей MARKET-1501 в качестве обучающего набора и используем обучающий набор MOT16 в качестве проверочного набора для проверки производительности алгоритма.

Экспериментальная среда основана на Windows 10, графической карте Nvidia GeForce RTX 1050Ti, оперативной памяти 4G, с использованием среды глубокого обучения Pytorch 1.13.0 и реализована на сервере Python 3.8. Выберите тестовый набор MOT16 для многоцелевого отслеживания общедоступных данных, чтобы протестировать этот алгоритм, и отправьте результаты теста на официальный веб-сайт MOT Challenge для оценки, сравнения с другими передовыми алгоритмами и анализа производительности модели.

Для оценки мы использовали общие показатели оценки в области многоцелевого отслеживания: точность многоцелевого отслеживания (MOTA), точность многоцелевого отслеживания (MOTP), возможность обслуживания идентификатора многоцелевого трекера (IDF1), время переключения идентификатора пешехода (IDs), наиболее отслеживаемые цели (MT), наиболее пропущенные цели (ML).

Результаты анализа: Мы улучшили алгоритм повторной идентификации пешеходов (fastREID) на основе DeepSORT, благодаря чему была повышена точность, точность и возможность сохранения идентификатора пешехода модели. Кроме того, замечено, что производительность в реальном времени немного хуже, чем у алгоритма JDE, потому что JDE является одноэтапным алгоритмом отслеживания, а отслеживание в реальном времени относительно высоко, но переключение идентификаторов относительно частые, что обусловлено взаимным перекрытием мишеней. В розничной среде часто бывают сцены с большим количеством плотных пешеходов, поэтому наш алгоритм является подходящим выбором в это время.]

我们提取大型目标检测公开数据集 MARKET-1501 中的行人类数据子集作为训练集，用

MOT16 的训练集作为验证集来验证算法的性能。

实验环境基于 Windows 10, Nvidia GeForce RTX 1050Ti 显卡, 运行内存为 4G, 采用 Pytorch 1.13.0 深度学习框架, 在 Python 3.8 的服务器上实现。选择多目标跟踪公开数据 MOT16 测试集测试本算法, 并将测试结果提交 MOT Challenge 官网进行评估, 与其他先进算法进行对比, 并分析模型性能。

我们采用了多目标跟踪领域通用的评估指标进行评估: 多目标跟踪准确度 (MOTA)、多目标跟踪精度(MOTP)、多目标跟踪器 ID 维持能力(IDF1)、行人 ID 切换次数(IDs)、大多数跟踪目标百分比 (MT)、大多数丢失目标百分比(ML)。

分析结果: 我们在 DeepSORT 的基础上改良了行人重识别的算法 (fastREID), 因此模型的准确度、精度和维持行人 ID 的能力都有提升。此外注意到与 JDE 算法相比实时性略差, 这是因为 JDE 属于一阶段的跟踪算法, 实时性跟踪相对较高, 但是 ID 切换相对频繁, 这是由于目标相互遮挡导致的。在零售的环境下经常出现大量密集行人的场景, 所以此时我们的算法是一个合适的选择。

14

