

Международная научная конференция школьников

«XXI Колмогоровские чтения»

КОНВЕЙЕРНАЯ ЛЕНТА ДЛЯ СОРТИРОВКИ ИЗДЕЛИЙ С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ

Муниципальное автономное учреждение дополнительного образования
«Дворец творчества детей и молодежи имени О.П.Табакова», Саратов

Выполнили: Чернякова Юлия (12 лет),
Черняков Александр (9 лет)

Преподаватель: Нагорнова Марина Сергеевна

г. Москва
4 мая 2021 года

Актуальность проекта:



Интерес к искусственному интеллекту растет с каждым днем. Многие компании работают над созданием роботов, которые могут «думать» и способны заменить человека. Роботы не устают и не делают перерывов, поэтому они работают эффективнее.

Мы тоже решили попробовать свои силы в этом направлении и создать робота, который самостоятельно осуществляет сортировку изделий на конвейере.

Конвейерная лента для сортировки изделий с искусственным интеллектом

Цель проекта - создать робота-помощника, который может сортировать изделия на конвейере без участия человека.

Задачи проекта:

- ▶ изучить возможности компьютера RaspberryPi;
- ▶ изучить возможности применения ИИ для создания роботизированного помощника.
- ▶ изучить различные технические решения, которые использовались ранее для решения данной проблемы;
- ▶ найти необходимое техническое решение для проекта;
- ▶ написать необходимые программы на языке Python.

Конвейерная лента для сортировки изделий с искусственным интеллектом

Техническое оснащение:

- компьютер RaspberryPi 3 Model B;
- камера ночного видео RaspberryPi 3 1/4 дюйма;
- сервомотор 360 гр. – 1 шт.;
- сервомоторы 9G Micro Servo– 3 шт.;
- инфракрасный датчик препятствия на LM393 – 3шт.;
- лопатки для сортировки – 2 шт.;
- элементы конструктора Lego Mindshtorm;
- сервис ИИ <https://www.clarifai.com/>



Результаты работы:

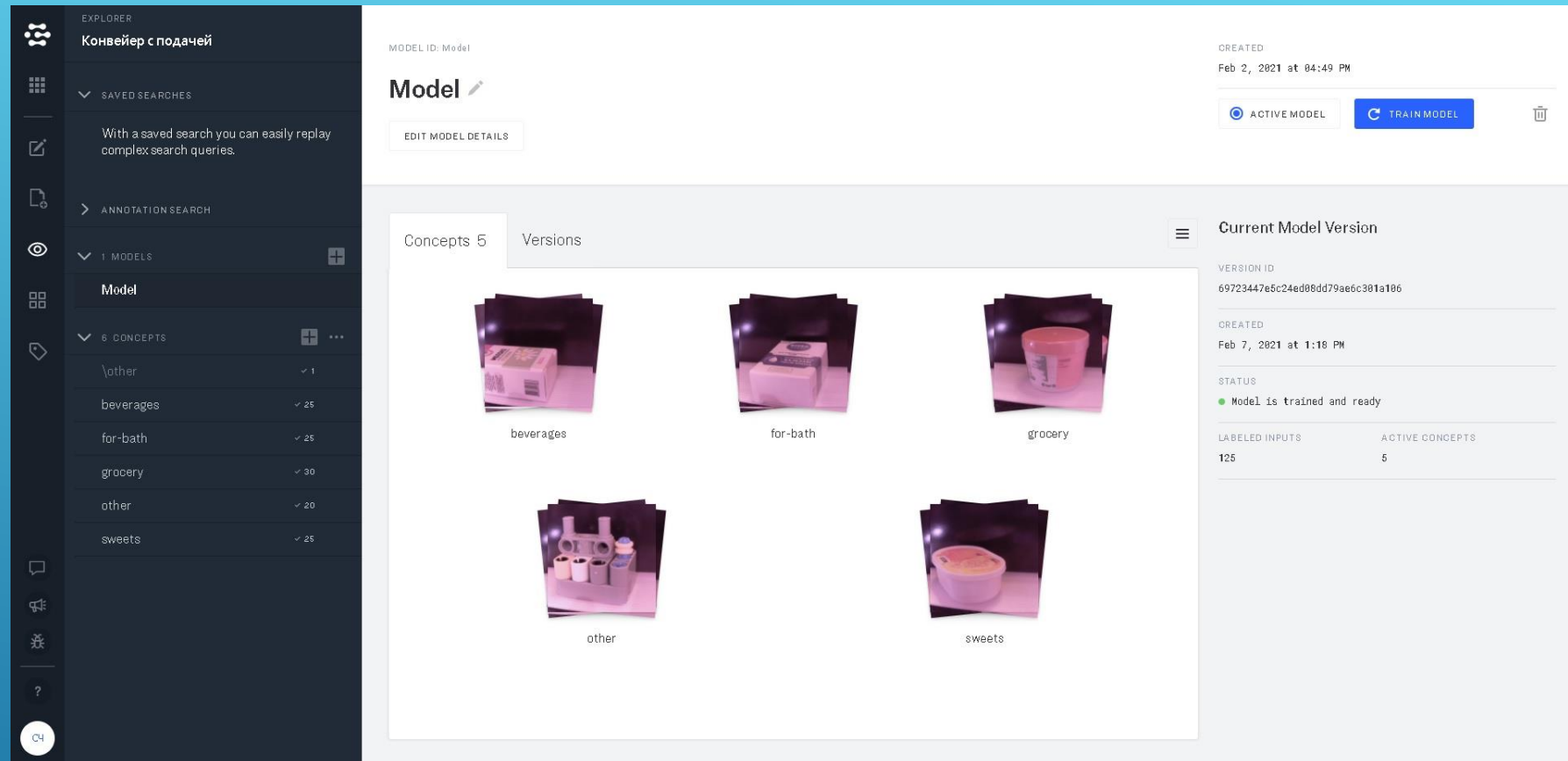
Робот сортирует изделия, помещая их в соответствующий контейнер. Наш конвейер сортирует продукты и товары для дома. Конвейер оснащен механизмом автоматической подачи объектов. Подъемная платформа с помощью сервопривода выталкивает объект на конвейерную ленту.

Объекты могут быть любыми, если обучить модель ИИ на эти категории. Поэтому робот может быть использован, например, для сортировки игрушек, продуктов, мусора и т.п.

Для распознавания объектов роботу помогает компьютерное зрение.



Результаты работы:



Была создана модель ИИ на сервисе <https://www.clarifai.com/>. Модель содержит пять категории объектов «Напитки (beverages)», «Бакалея (grocery)», «Сладости (sweets)», «Предметы для ванной комнаты (for-bath)», «Остальное (other)». Обучение модели проводилось на 20-25 изображениях по каждой категории.

Результаты работы:

Камера фиксирует изображение объекта, управляющая программа передает изображение на сервис. Модель ИИ распознает изображение и возвращает список вероятностей соответствия каждой категории. Управляющая программа принимает данные, переводит сервомоторы в нужное положение, объект подъезжает к нужному контейнеру и лопатка сбрасывает объект в контейнер. Управляющая программа написана на языке Python.



Код программы:

```
try:
    import clarifai_grpc
    import threading
    import picamera
    import gpiozero
    import time
    import io

    stub = clarifai_grpc.grpc.api.service_pb2_grpc.V2Stub(clarifai_grpc.channel.clarifai_channel.ClarifaiChannel.get_grpc_channel())
    metadata = (('authorization', 'Key 1d3bae95e78f46b2887674fd43ae2b6c'),)
    camera = picamera.PiCamera()

    mot = gpiozero.Motor(18, 15, pwm = True)
    sens1 = gpiozero.DigitalInputDevice(14, pull_up = True)
    sens2 = gpiozero.DigitalInputDevice(23, pull_up = True)
    sens3 = gpiozero.DigitalInputDevice(21, pull_up = True)
    ser1 = gpiozero.Servo(25, initial_value = None, min_pulse_width = 0.000554, max_pulse_width = 0.0024)
    ser2 = gpiozero.Servo(12, initial_value = None, min_pulse_width = 0.000554, max_pulse_width = 0.0024)
    feedser = gpiozero.Servo(24, initial_value = None, min_pulse_width = 0.000554, max_pulse_width = 0.0024)
    tofeed = 0
    feedrun = True
```


Код программы:

```
def autofeed():
    global tofeed
    global feedrun
    while feedrun:
        if tofeed > 0:
            feedser.value = -0.6
            while feedser.value < 0.3:
                feedser.value += 0.01
                time.sleep(0.01)
            time.sleep(1)
            feedser.value = -0.6
            time.sleep(1)
            feedser.detach()
            tofeed -= 1
feedthread = threading.Thread(target = autofeed)
feedthread.start()

input('Enter')
while True:
    mot.forward(speed = 0.2)
    tofeed += 1
    sens1.wait_for_active()
    sens1.wait_for_inactive()
    time.sleep(0.7)
    mot.stop()
    time.sleep(1)
```

Код программы:

```
image = io.BytesIO(b'')
camera.capture(image, 'jpeg')
request = clarifai_grpc.grpc.api.service_pb2.PostModelOutputsRe
quest(
    model_id='Model',
    inputs=[
        clarifai_grpc.grpc.api.resources_pb2.Input(
            data=clarifai_grpc.grpc.api.resources_pb2.Data(
                image=clarifai_grpc.grpc.api.resources_pb2.Image(
                    base64=image.getvalue()
                )))
    ])
response = stub.PostModelOutputs(request, metadata=metadata)
mot.forward(speed = 0.2)

name = response.outputs[0].data.concepts[0].name
value = response.outputs[0].data.concepts[0].value

print(f'Name: {name}\tValue: {value}')

if name == 'beverages':
    ser1.value = -0.5
    sens2.wait_for_active()
    time.sleep(0.75)
    ser1.value = 0.5
```

Код программы:

```
if name == 'for-bath':  
    ser1.value = 0.5  
    sens2.wait_for_active()  
    time.sleep(0.75)  
    ser1.value = -0.5  
if name == 'grocery':  
    ser1.value = -0.5  
    ser2.value = -0.15  
    sens3.wait_for_active()  
    time.sleep(0.9)  
    ser2.value = 0.4  
if name == 'sweets':  
    ser1.value = -0.5  
    ser2.value = 0.4  
    sens3.wait_for_active()  
    time.sleep(0.9)  
    ser2.value = -0.15  
if name == 'other':  
    sens3.wait_for_active()  
time.sleep(1)  
ser1.detach()  
ser2.detach()  
finally:  
    mot.stop()  
    feedrun = False
```

Результат работы программы:

```
root@raspberrypi:~/Desktop/Конвейер/Конвейер с подачей# python3 main1.py
Enter
Name: beverages Value: 0.8260912299156189
Name: grocery Value: 0.8119040727615356
Name: for-bath Value: 0.9938953518867493
Name: other Value: 0.9999927282333374
Name: sweets Value: 0.9070728421211243
Name: sweets Value: 0.9682923555374146
Name: for-bath Value: 0.7267459034919739
Name: other Value: 0.9999963045120239
Name: beverages Value: 0.759863555431366
```


СПАСИБО ЗА ВНИМАНИЕ!

