

Научно-исследовательская работа

Предмет: информатика

**РАЗРАБОТКА СИСТЕМЫ, РАСПОЗНАЮЩЕЙ ПО ВИДЕО НАЛИЧИЕ
НА РАБОТНИКЕ ХИМИЧЕСКОГО ПРЕДПРИЯТИЯ СРЕДСТВ
ИНДИВИДУАЛЬНОЙ ЗАЩИТЫ**

Выполнил:

Федченко Максим Владимирович,

учащийся 9 класса

МБОУ СОШ №15, Россия, г.Апатиты

Руководитель:

Коркачева Дина Александровна,

учитель информатики

МБОУ СОШ №15, Россия, г.Апатиты

Введение

В эру цифровых технологий, машинное обучение, в частности компьютерное зрение – важная и значимая отрасль. С помощью машинного обучения люди упрощают себе жизнь, делают мир цифровых технологий лучше и безопасней. Согласно результатам международного исследования Microsoft, 94% руководителей считают, что технологии искусственного интеллекта важны для решения стратегических задач их организаций. При этом 27% опрошенных уже внедрили соответствующие технологии в ключевые бизнес-процессы, еще 46% ведут пилотные проекты.

Россия, благодаря традиционно сильной математической школе не отстает от мировых тенденций. И если судить по докладам на конференции «Технологии машинного обучения. Искусственный интеллект и нейросети: инструменты и опыт реальных проектов», организованной издательством «Открытые системы», применение средств ИИ в нашей стране уже стало если не обыденностью, то достаточно распространенным способом не только оптимизировать, но и радикально поменять бизнес-процессы.

Поскольку разнообразие методов и способов обучения искусственного интеллекта не меньше, чем у интеллекта естественного, то, прежде чем начать проект в этой области, я ознакомился с методами и способами машинного обучения, их возможностями, сферами применения и ограничениями.

Компания «ФосАгро» в настоящий момент активно внедряет элементы цифровизации на производственных предприятиях. На крупном химическом предприятии безопасности работников и охране труда уделяется большое внимание. От применения средств индивидуальной защиты на производстве зависит жизнь и здоровье человека. Сегодня все большее внимание уделяется применению технологий нейросетей, автоматическому сбору, обработке информации и формированию баз данных.

Заказчиком проекта выступила Дирекция по информационным технологиям АО «Апатит» в лице заместителя директора департамента по цифровизации Виноградова С.Е.

Объект исследования: распознавание по видео наличие на работнике химического предприятия средств индивидуальной защиты.

Предмет исследования: разработка системы, распознающей по видео наличие на работнике химического предприятия средств индивидуальной защиты.

Изучение информационных источников и уточнение темы: в процессе работы над данной темой была проанализирована основная учебная и научно-популярная литература, которая позволила осмыслить и осуществить выполнение учебно-исследовательской работы. Много интересной информации узнал из статьи Григория Алексева, опубликованной на сайте Введение в машинное обучение <https://habr.com/ru/post/448892/> [1]

Гипотеза: если изучить основы компьютерного зрения, то на практике можно реализовать модель машинного обучения, распознающую систему индивидуальной защиты (далее СИЗ) на сотруднике предприятия. **Цель:** применение технологий компьютерного зрения при создании модели машинного обучения на Python. Для достижения цели были поставлены и решены основные задачи:

1. изучить технологии машинного обучения;
2. приобрести навыки работы в Python;
3. разработка системы на Python, распознающей по видео наличие на работнике химического предприятия средств индивидуальной защиты.

Методы исследования: теоретические (анализ и синтез), математические.

Основная часть

Машинное обучение

Машинное обучение - обозначает множество математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных.

Машинное обучение используется в структурах обеспечения безопасности. Например, система распознавания лиц в метро. Камеры сканируют лица людей,

входящих и выходящих из метро. Аналитические машины сравнивают снимки с лицами, которые находятся в розыске. Если сходство высоко, то система подает сигнал. Примеры применения в реальной жизни: диагностика заболеваний, поиск мест залегания полезных ископаемых, оценка надёжности и платёжеспособности кандидатов на получение кредитов и так далее [2].

Методы машинного обучения

Существует множество методов машинного обучения. Но самые распространённые из них – обучение с учителем и без.

Обучение с учителем — один из способов машинного обучения, в ходе которого испытуемая система принудительно обучается с помощью примеров «стимул-реакция». Метод применяется, когда необходимо найти функциональную зависимость результатов от входов и построить алгоритм, на входе принимающий описание объекта и на выходе выдающий ответ.

Обучение без учителя — один из способов машинного обучения, при котором испытуемая система спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора [3].



Кластер. Машинное обучение

Компьютерное зрение

Компьютерное зрение - это область искусственного интеллекта, связанная с анализом изображений и видео. Она включает в себя набор методов, которые наделяют компьютер способностью «видеть» и извлекать информацию из увиденного.

Выбор языка программирования

Для того чтобы реализовать алгоритм машинного обучения, необходимо выбрать язык программирования, на котором будет писаться код.

Сейчас на рынке в основном выделяют 5 языков программирования, для машинного обучения и анализа данных: Python, R, C++, JavaScript и C#.

Проведя анализ, я пришёл к выводу, что самыми подходящими языками являются C++ и Python. Свой выбор я остановил на Python.

Выбор библиотеки для проекта

Для упрощения реализации искусственного интеллекта, были придуманы специальные библиотеки, позволяющие без написания длинного кода использовать нужные функции.

Для моей задачи отлично подойдёт ImageAI - библиотека Python, созданная для того, чтобы дать разработчикам возможность создавать приложения и системы с автономными возможностями глубокого обучения и компьютерного зрения с использованием простого и небольшого количества строк кода.

Нахождение данных для анализа

Чтобы наша будущая модель смогла различать объекты, ее необходимо обучить на множестве различных данных. Для этого я использовал видео с предприятия Фосагро, а также набор данных, с большим числом фотографий защитных касок.

Обработка данных

Для этого с помощью программы LabelImg я создавал аннотации к фотографиям, проще говоря, размечал нужные объекты на фото. Суммарно я разметил 800 фотографий вручную. Остальные изображения были уже аннотированы.



Фото 1. Разметка объектов

Помимо этого, желательно иметь +- схожий размер фотографий. В моём случае это 416*416.

Для этого я написал вот такой код, который задаёт нужным размерам фотографиям:

```
from PIL import Image
import os, sys

path = "/root/Desktop/python/images/"
dirs = os.listdir( path )
def change():
    for item in dirs:
        if os.path.isfile(path+item):
            img = Image.open(path+item)
            f, e = os.path.splitext(path+item)
```

```

        imResize = img.resize((416,416), Image.ANTIALIAS)
        imResize.save(f + ' resized.jpg', 'JPEG',
quality=90)

change()

```

Листинг 1. Тестирование модели

Затем общее число фотографий нужно разбить в процентом соотношении. А именно 80 и 20 процентов соответственно. Где 80% - обучающие данные, а 20% - тестовые.

Создание и обучение модели

```

from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="helmet")
trainer.setTrainConfig(object_names_array=["helmet", "head", "pe
rson"], batch_size=4, num_experiments=30,
train_from_pretrained_model="detection_model-ex-020--loss-
0019.130.h5")
trainer.trainModel()

```

Листинг 2. Создание и обучение модели

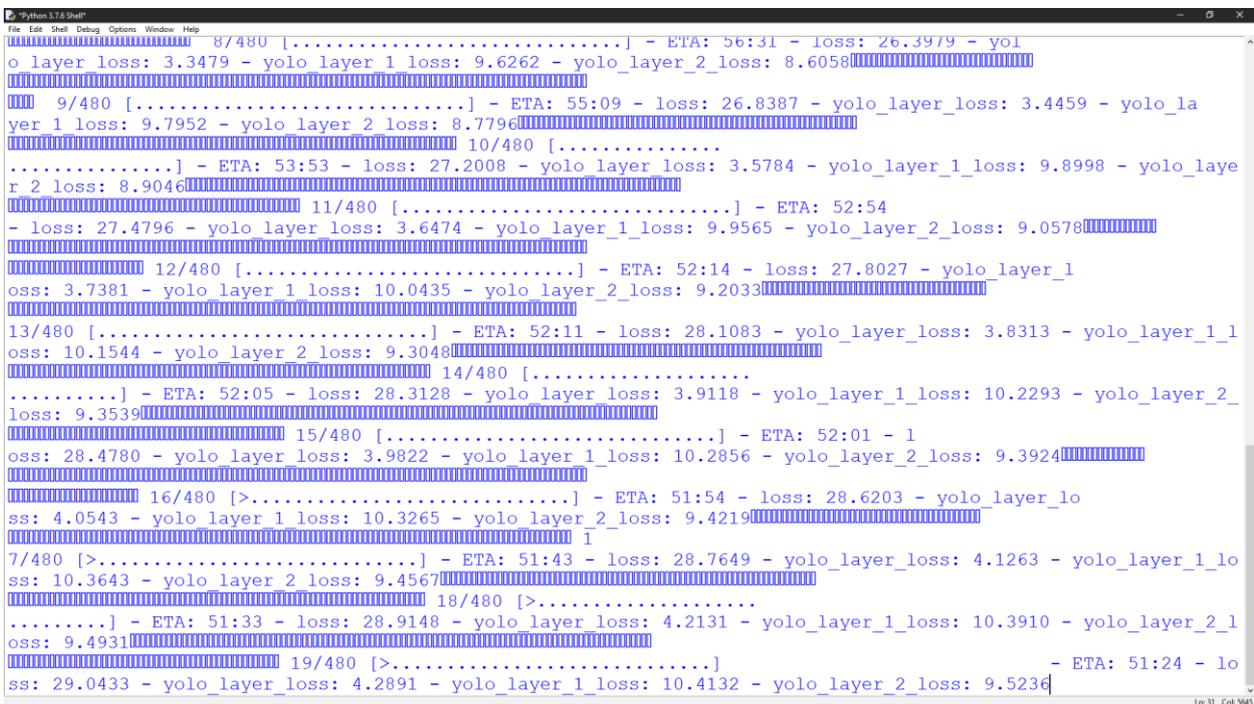
Для начала импортируем нужную библиотеку, затем создаём модель с нужными параметрами обучения.

Trainer – Наша модель

Object_names_array – Названия, тех объектов, которые мы будем находить. Это объекты находятся в файлах для аннотации фотографий.

num_experiments – Количество эпох. Эпоха - одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и, возможно, проверку качества обучения на контрольном множестве.

Запускаем код и ждём выполнения обучения



```

8/480 [.....] - ETA: 56:31 - loss: 26.3979 - yolo_layer_loss: 3.3479 - yolo_layer_1_loss: 9.6262 - yolo_layer_2_loss: 8.6058
9/480 [.....] - ETA: 55:09 - loss: 26.8387 - yolo_layer_loss: 3.4459 - yolo_layer_1_loss: 9.7952 - yolo_layer_2_loss: 8.7796
10/480 [.....] - ETA: 53:53 - loss: 27.2008 - yolo_layer_loss: 3.5784 - yolo_layer_1_loss: 9.8998 - yolo_layer_2_loss: 8.9046
11/480 [.....] - ETA: 52:54 - loss: 27.4796 - yolo_layer_loss: 3.6474 - yolo_layer_1_loss: 9.9565 - yolo_layer_2_loss: 9.0578
12/480 [.....] - ETA: 52:14 - loss: 27.8027 - yolo_layer_loss: 3.7381 - yolo_layer_1_loss: 10.0435 - yolo_layer_2_loss: 9.2033
13/480 [.....] - ETA: 52:11 - loss: 28.1083 - yolo_layer_loss: 3.8313 - yolo_layer_1_loss: 10.1544 - yolo_layer_2_loss: 9.3048
14/480 [.....] - ETA: 52:05 - loss: 28.3128 - yolo_layer_loss: 3.9118 - yolo_layer_1_loss: 10.2293 - yolo_layer_2_loss: 9.3539
15/480 [.....] - ETA: 52:01 - loss: 28.4780 - yolo_layer_loss: 3.9822 - yolo_layer_1_loss: 10.2856 - yolo_layer_2_loss: 9.3924
16/480 [>.....] - ETA: 51:54 - loss: 28.6203 - yolo_layer_loss: 4.0543 - yolo_layer_1_loss: 10.3265 - yolo_layer_2_loss: 9.4219
17/480 [>.....] - ETA: 51:43 - loss: 28.7649 - yolo_layer_loss: 4.1263 - yolo_layer_1_loss: 10.3643 - yolo_layer_2_loss: 9.4567
18/480 [>.....] - ETA: 51:33 - loss: 28.9148 - yolo_layer_loss: 4.2131 - yolo_layer_1_loss: 10.3910 - yolo_layer_2_loss: 9.4931
19/480 [>.....] - ETA: 51:24 - loss: 29.0433 - yolo_layer_loss: 4.2891 - yolo_layer_1_loss: 10.4132 - yolo_layer_2_loss: 9.5236
  
```

Скриншот 1. Ход выполнения

По мере обучения, смотрим параметры обученной модели. Для этого напишем ещё один файл с кодом

```

from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="helmet")
  
```

```

metrics =
trainer.evaluateModel(model_path="helmet/models/detection_model-ex-002--loss-0023.363.h5",
json_path="helmet/json/detection_config.json",
iou_threshold=0.5, object_threshold=0.3, nms_threshold=0.5)
print(metrics)

```

Листинг 3. Параметры обученной модели

```

Python 3.7.6 Shell
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Максим\Desktop\det2\metrics.py =====
Starting Model evaluation...
Evaluating over 61 samples taken from helmet/validation
Training over 237 samples given at helmet/train
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Model File: helmet/models/detection_model-ex-002--loss-0023.363.h5

Evaluation samples: 61
Using IoU: 0.5
Using Object Threshold: 0.3
Using Non-Maximum Suppression: 0.5
head: 0.5166
helmet: 0.3694
person: 0.0000
mAP: 0.2954

[{'model_file': 'helmet/models/detection_model-ex-002--loss-0023.363.h5', 'using_iou': 0.5, 'using_object_threshold': 0.3, 'using_non_maximum_suppression': 0.5, 'average_precision': {'head': 0.5166423442132411, 'helmet': 0.3694361574174364, 'person': 0.0}, 'evaluation_samples': 61, 'map': 0.29535950054355914}]
>>>

```

Чем ближе показатель к 1(в нашем случае – head и helmet), тем модель лучше.

Тестирование модели

Для тестирования понадобится наша модель, а также любая картинка, с объектами, которые нужно найти.

Напишем следующий код

```

from imageai.Detection.Custom import CustomObjectDetection

detector = CustomObjectDetection()
detector.setModelTypeAsYOLOv3()

```

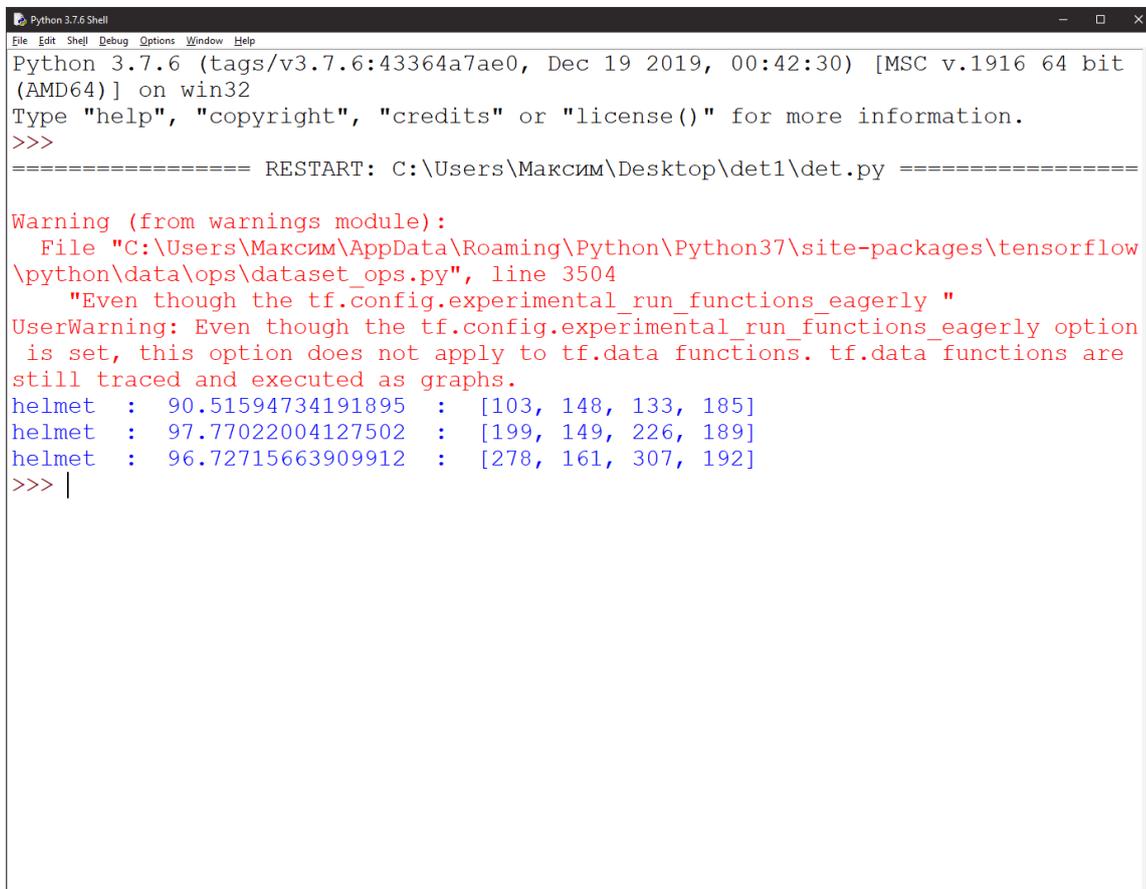
```

detector.setModelPath("helmet/models/detection_model-ex-00")
detector.setJsonPath("helmet/json/detection_config.json")
detector.loadModel()
detections =
detector.detectObjectsFromImage(input_image="t.jpg",
output_image_path="tr.jpg")
for detection in detections:
    print(detection["name"], " : ",
detection["percentage_probability"], " : ",
detection["box_points"])

```

Листинг 4. Тестирование модели

После его выполнения, программа выведет координаты, найденных объектов, а также нарисует их границы.



```

Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Максим\Desktop\det1\det.py =====
Warning (from warnings module):
  File "C:\Users\Максим\AppData\Roaming\Python\Python37\site-packages\tensorflow
\python\data\ops\dataset_ops.py", line 3504
    "Even though the tf.config.experimental_run_functions_eagerly "
UserWarning: Even though the tf.config.experimental_run_functions_eagerly option
is set, this option does not apply to tf.data functions. tf.data functions are
still traced and executed as graphs.
helmet : 90.51594734191895 : [103, 148, 133, 185]
helmet : 97.77022004127502 : [199, 149, 226, 189]
helmet : 96.72715663909912 : [278, 161, 307, 192]
>>> |

```



Фото 2. До обнаружения



Фото 3. После обнаружения

Заключение

Данное решение может стать частью большого проекта по внедрению цифровых решений в области промышленной безопасности и охраны труда на предприятиях компании «ФосАгро», а также использовано как основа для решения подобных задач в других направлениях производственной деятельности.

Список литературы

1. Введение в машинное обучение [Электронный ресурс]
<https://habr.com/ru/post/448892/>
2. Машинное обучение [Электронный ресурс]
<https://www.osp.ru/cio/2018/05/13054535>
3. Обзор алгоритмов машинного обучения [Электронный ресурс]
<https://tproger.ru/translations/top-machine-learning-algorithms/>
4. Гид: алгоритмы машинного обучения и их типы [Электронный ресурс]
https://www.sas.com/ru_ru/insights/articles/analytics/machine-learning-algorithms-guide.html
5. Бизли Д. Python. Подробный справочник. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.