**Проект на тему:**

**«Детекция медицинских работников на изображении»**

**Работу выполнил: Кулиговский Д.В.**

***Задача***

Сделать модель нейронной сети, принимающую на вход фотографию с медиками в халатах, а на выходе получаем изображение с отрисованными детекциями.

***Модель для реализации проекта***

В рамках задачи использовалась модель YOLOv3, подход включает в себя единую глубокую сверточную нейронную сеть.

Модель YOLO использует набор заранее определенных размеров и пропорций, которые используются для вычисления координат и размеров найденных объектов. YOLOv3 разбивает изображение на сетку и каждая ячейка сетки отвечает за предсказание объекта, находящегося в этой ячейке.

Ключевые преимущества, которые можно выделить в рамках реализации проекта с использованием модели YOLOv3:

- Высокая скорость работы;

- Настройка процесса обучения;

- Возможность для реализации идей, нацеленных на достижение результата проекта.

- Обнаружение нескольких объектов: YOLOv3 может одновременно обнаруживать несколько объектов на изображении.

Существует несколько других моделей YOLO, таких как YOLOv4 и YOLOv5, которые также имеют свои преимущества и недостатки.

***Датасет***

Датасет состоит из 750 изображений. В выборке содержатся изображения с 1, 2, 3 и 4 медработниками в халатах.

В основном, медики на фотографиях занимают большую часть изображения по пояс, меньшую часть выборки составляют изображения, где врачи в полный рост. Фон в большей части выборки светлый, реже темный.

Для сборки использовалась платформа freepick, также для лучшего обучения модели проводилась предобработка изображений с целью уменьшения засвета и выделения контура медработников на общем фоне. Такие шаги помогли сделать обучение модели более продуктивным и предсказуемым.

Датасет формировался плавно, параллельно велось обучение и анализ влияния новых картинок на предсказание модели. Поскольку в ходе работы было обнаружено, что резкое увеличение датасета не позволяло оценивать его влияния на предсказание модели YOLO. И было принято решение проводить обучение и параллельно формировать выборку.

Пример изображения для обучения:



***Разметка датасета***

Детекция датасета с медработниками проведена по классу „Medic“ и проводилась вручную, сейчас модель можно использовать для автоматической разметки изображений.

В начале работы над моделью, у меня не было такой хорошо обученной модели, для разметки датасета и в этом помог сервис Roboflow.

Также изображения, подобранные для датасета имеют разную размерность. Модель YOLOv3 работает только с квадратными изображениями, поэтому перед подачей изображений на вход модели, необходимо изменить их размер до 416 на 416 пикселей.

Хочу отметить, что ручная разметка позволяет детально изучить датасет и понять его особенности, при этом настраивать его на конечный результат.

Примеры размеченных изображений:

****

**Обучение модели**

Написание кода и обучение модели велось на платформе Google Colab с использованием их ускорителя для обучения модели - CPU, что очень помогло, так как было проведено множество экспериментов.

После каждого эксперимента и заметного улучшения в предсказании, сохранялись веса модели и датасет, показавший результат в работе.

Общий подход к обучению был таков: первым делом создали модель, проанализировали получившийся код, подготовили небольшой датасет для оценки работы модели и ее возможностей. После того, как модель YOLO была подключена и ее можно было использовать для достижения результата, приступили к более масштабному отбору картинок под обучение и валидацию.

В ходе работы были подключены 2 колбека, которые сыграли важную роль в обучении. Один из них сохранял лучшие веса, каждый раз, когда у модели снижалась валидационная ошибка. Другой уменьшал скорость обучения, если модель на протяжении трех эпох обучения не снижала ошибку на валидационной выборке или выходила на плато.

Приходилось также бороться с переобучением, начинал скакать вал лосс (ошибка на валидационном наборе), это происходит, когда модель слишком хорошо запоминает обучающие данные и не может обобщить на новые данные. Тут в ход идут подборка параметров сети.

Для данной модели важнейшими параметрами, которые влияли на качество обучения модели были процент выборки под валидацию и размер батча, то есть шага обучения в ходе одной эпохи.

Было проведено много экспериментов для подбора процента валидации значение колебались от 10 до 20 %, где в рамках нашей задаче сработал высокий процент — 18 % от общей выборки.

А размер батча равнялся 3. Тестировался от 2 до 10. Слишком маленький размер батча может привести к тому, что модель будет слишком сильно адаптироваться к отдельным примерам. При большом, модель может упустить некоторые тонкие детали в данных и будет склонна обобщать их.

Подбирались эти параметры опытным путем, до 25 эпох для каждого. А когда пришло понимание, что настройка оптимальная (батча = 3, процент вал лосса = 18 %), то среднее количество эпох до выхода модели на плато варьировалось от 150 до 180.

Как я уже писал ранее, обучение проводилось параллельно с формированием датасета и именно это помогло добиться лучшего результата.

Во время анализа велся блокнот экспериментов, который позволил возвращаться к результатам и проводить изменения в модели более продуктивно.

Грамотный анализ невозможен без ведения записей, когда ты вносишь изменение в параметры модели или выборку датасета, то при обучение получаешь результат и важно видеть, что приводит к конечной цели, а что отталкивает назад.

Немного о вал лоссе, в первые дни обучения модели лучший результат, который удавалось достичь был в районе 20-25, при таком вал лоссе модель дает заметную ошибку при тестировании на общей выборке, а на новых картинках совсем теряется. Самый лучший результат вал лосс, которого удалось достичь на конечном датасте получился 9.5, уменьшение в 2 раза дало заметное улучшение в работе, модель на общей выборке полностью отрабатывала себя. А на тестовых картинках уже демонстрировала результат намного лучше прежнего.

Конечно, есть что дорабатывать, но общий подход к улучшению понятен — увеличение датасета качественными изображениями с разметкой, детальный анализ работы модели при обучение и тенденции в предсказании на тестовом наборе данных.

Модель YOLOv3 полностью оправдывает себя и способна решать такие задачи.

Код с работой в Colab:

https://colab.research.google.com/drive/1I6vycik-HiCjxHzupAK1ByxZBA1XPXKe?usp=sharing

***Тестирование модели***

Тестирование модели проводилось в два этапа:

Первый этап включал в себя анализ предсказания на общей выборке из датасета, здесь у нас появлялась информация, чего не хватает в рамках уже имеющихся изображений. И что можно сделать здесь и сейчас.

И вторым этапом на предсказание подавались картинки, которые модель ранее не видела, здесь для себя отмечали общие черты - изменения в предсказании по сравнению с предыдущими экспериментами, слабые стороны в работе модели на данном этапе и где уже проявляет себя как требуется. Данный этап показывает в каком направлении продолжать работу.

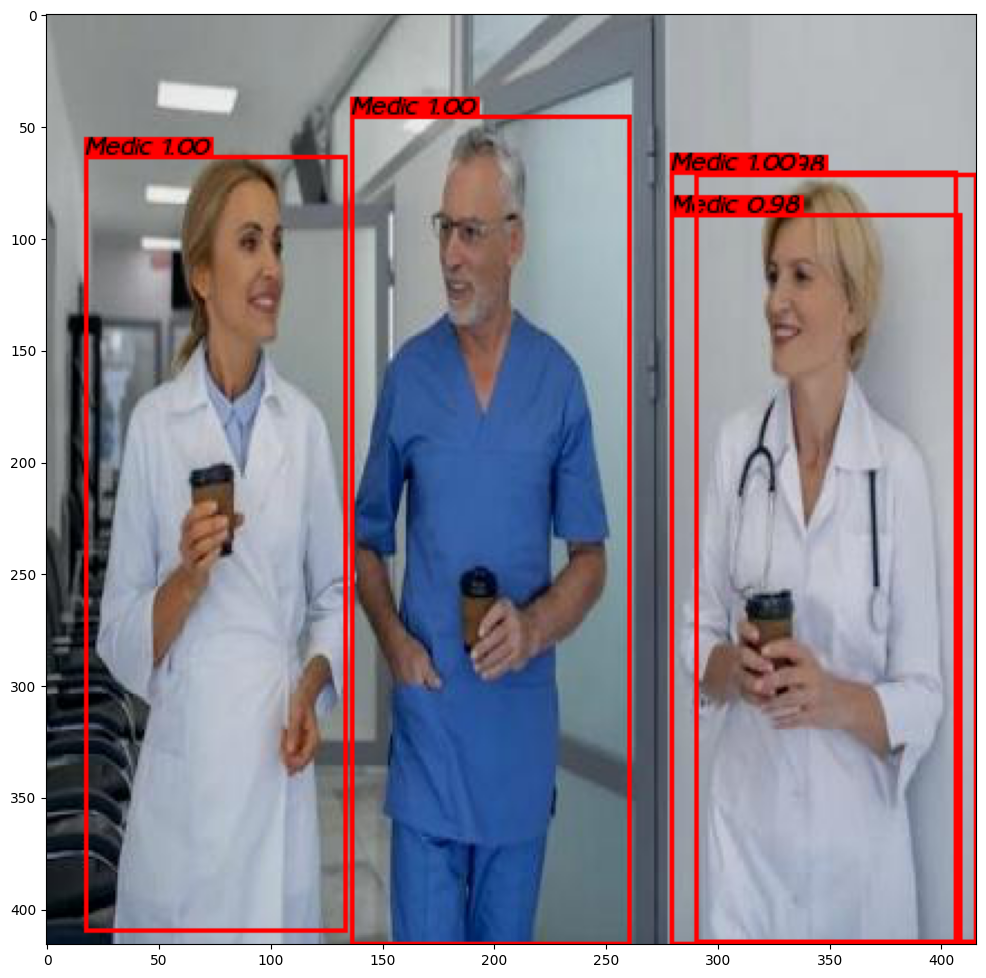
Эти два этапа также важны, как описанные выше — формирование датасета, разметка изображений и обучение модели.

***Итоговый результат работы модели***

Модель для детекции изображений медиков показывает достойный результат в предсказании.

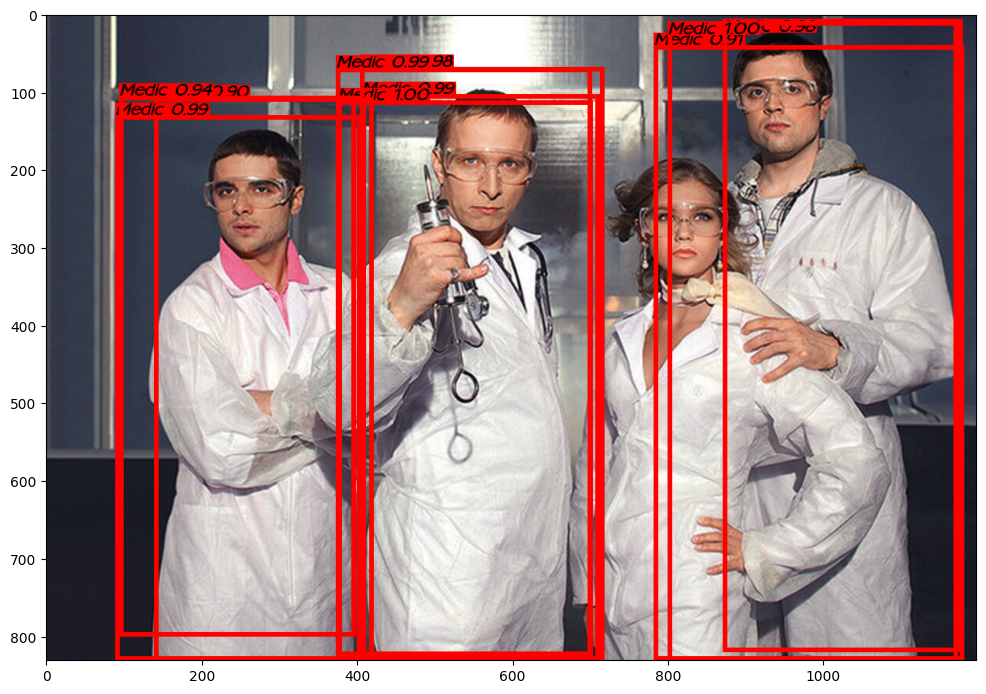
Легче даются изображения, где медики стоят на фоне потемнее на расстоянии друг от друга и крупные планы.

*Пример:*



Проблемы возникают на засвеченных изображениях и мутных кадрах, когда человек стоит один за другим, над синими халатами также нужно поработать детальнее.

*Пример:*

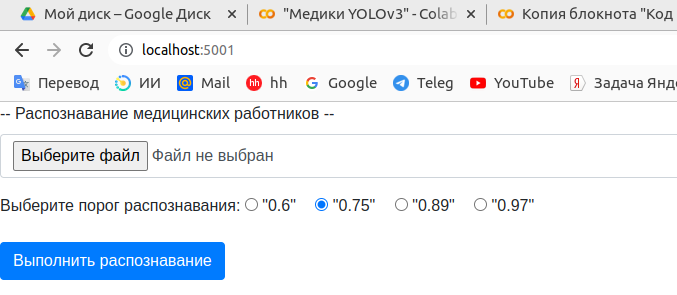


**Представление результата**

Для проверки работы модели реализовано веб-приложение на Flask. Для упрощения совместного запуска Flask и модели YOLOv3 был использован Docker.

Видео-инструкция по подключению находится в архиве, который был выслан вместе с данным описанием, в папке «Результат с Flask».

Так выглядит интерфейс:



Для распознавания загрузите изображений, нажав «Выберите файл», также можно установить порог для распознавания от 60 % до 97 %. Модель построит детекцию на изображении.

Результат предсказания:

