

## **ОБНАРУЖЕНИЕ ВОЗГОРАНИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ С ПОМОЩЬЮ YOLO**

**Коваленко Карина Александровна**

магистрант, Гомельский государственный технический университет, Республика Беларусь, г Гомель

YOLO – это алгоритм, который обнаруживает и распознает различные объекты на фотографии. Алгоритм YOLO использует сверточные нейронные сети (CNN) для обнаружения объектов в режиме реального времени. Как следует из названия, алгоритм требует только одного прямого прохождения через нейронную сеть для обнаружения объектов.

Это означает, что предсказание по всему изображению выполняется за один запуск алгоритма. CNN используется для одновременного прогнозирования вероятностей различных классов и ограничивающих границ.

Существует готовая реализация алгоритма YOLO, основанная на Darknet. Darknet – это нейронная сеть с открытым исходным кодом, написанная на C и CUDA. Она быстрая, простая в установке и поддерживает CPU и GPU вычисления. Darknet предоставляет набор предобученных моделей для классификации ImageNet.

Для запуска алгоритма YOLO и Tiny YOLO можно воспользоваться обученными для двадцати классов на различных наборах данных моделями и использовать их веса `yolo.weights` и `tiny-yolo-weights` соответственно. Так как модели имеют различные конфигурации и набор слоев, они не могут быть заменены друг на друга. Задача – изменить количество классов для обнаружения, и при этом внести новые классы, которых ранее не было в YOLO. Для этого нужно сделать свой файл с весами, так как `yolo.weights` обучен для конфигурации YOLO с двадцатью классами, он не подойдет для этой цели. Чтобы не обучать нейросеть с нуля, можно воспользоваться предобученными весами сверточных слоев `yolov3-tiny.conv.15`.

Теперь необходимо изменить конфигурационный файл для нейросети. Изменить файл `yolov3-obj.cfg`, который содержит в себе архитектуру улучшенной версии – YOLOv3. В этом файле нужно изменить несколько показателей: `classes=1`, `filters` с 255 на 18. Это число соответствует выражению `filters = (classes + 5)*3`, где `classes` количество классов в новой модели.

Установим название нового класса в файле `obj.names`: `fire`.

Для обучения необходимо создать набор изображений, для которых существует соответствующий набор разметки. В данной реализации для изображения с названием, например `fire.jpg`, должен существовать текстовый файл с таким же названием, но с расширением `.txt`, т.е. `fire.txt`. Этот файл должен быть заполнен в соответствии с форматом набора данных VOC2007, т.е. содержать в себе набор строк вида: «класс» «x» «y» «ширина» «высота».

– «класс» – натуральные числа от нуля до `(classes-1)`.

– «x» «y» «ширина» «высота» – дробные числа, соответствующие ширине и высоте изображения от 0.0 до 1.0 – т.е. соответствуют масштабу всего изображения.

– «x» = «x координата» / «ширина изображения»

- «высота» = «высота объекта» / «высота изображения»

- «x» «y» - центр квадрата.

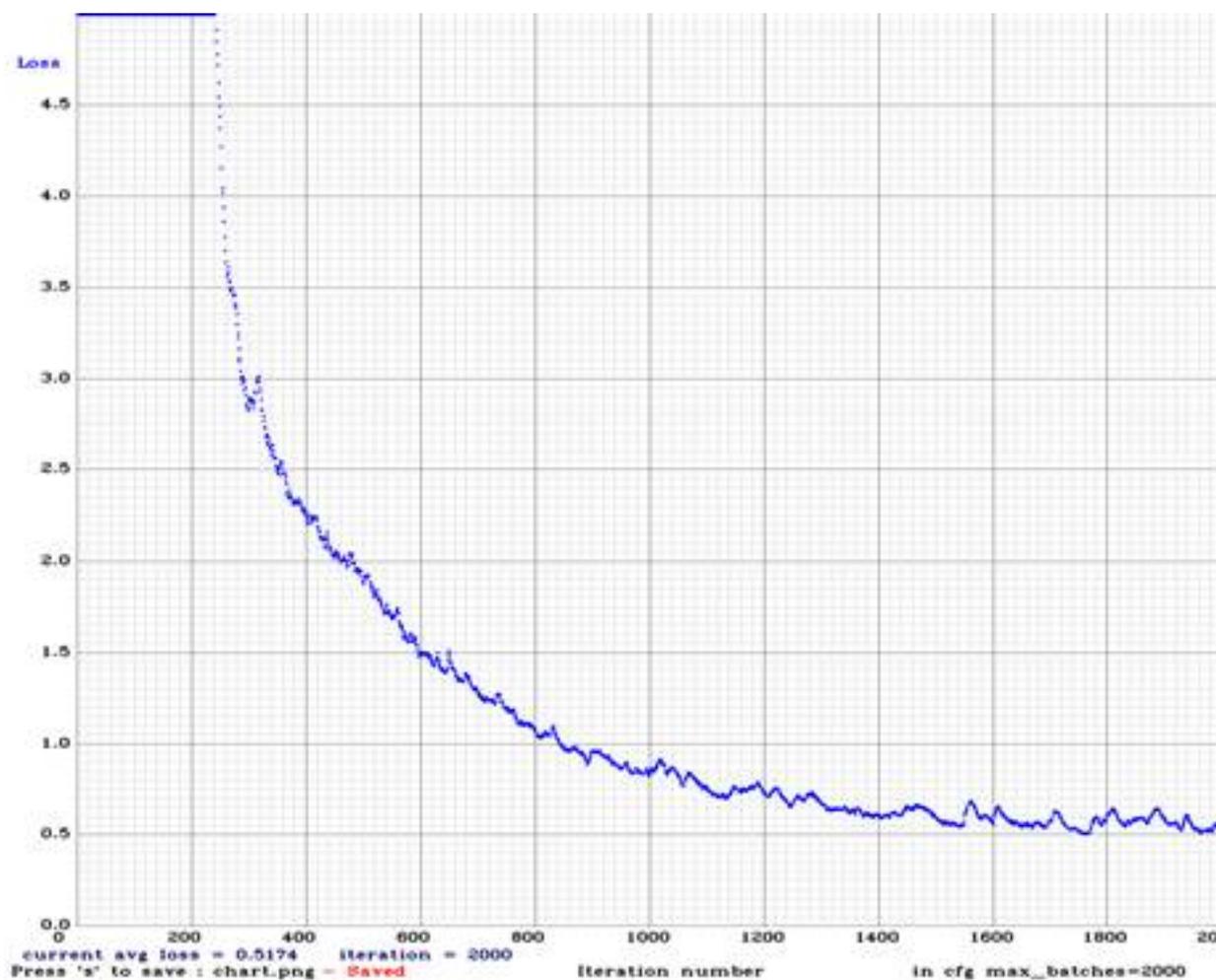
Для того чтобы запустить обучение исходного алгоритма с двадцатью классами на наборе данных VOC2007 и VOC2012 нужно выполнить скрипт, чтобы сформировать изображения .jpg и .txt. Запустим обучение выполнив команду (рисунок 2.1):

```
./darknet detector train obj.data yolov3-tiny-obj.cfg fire/model/yolov3-tiny.conv.15
```

**Рисунок 1. Команда обучения нейронной сети**

В процессе обучения будут сохраняться промежуточные веса - каждые сто итераций до 1000, далее каждые 1000 итераций.

Длительность обучения детектора измеряется в количествах итераций обучения - такова особенность фреймворка. YOLOv3 обучалась в течении двух тысяч итераций, в каждой из которых было по 64 изображения. Процесс уменьшения ошибки и роста метрики mAP показан на рисунке 2.



**Рисунок 2. График обучения YOLOv3**

Тестовая выборка должна содержать изображения, которые не участвовали в этапе обучения. Если производительность измеряется данными, которые модель уже видела во время обучения, результатам нельзя доверять из-за общей проблемы машинного обучения, известной как переобучение. Переобучение – это когда модель изучает то, что было специфично для данных, показанных ранее, но которые не обобщаются на другие данные.

### **Список литературы:**

1. Lee Y. H., Kim Y. Comparison of CNN and YOLO for Object Detection // Journal of the semiconductor & display technology. – 2020. – Т. 19. – №. 1. – С. 85-92.