

(студент, пенсионер, трудоспособный), имеющими схожие условия проживания и т.д.

Разработка программного средства велась на языке Java, с использованием фреймворка Spring. Также использовалась база данных PostgreSQL для хранения всех данных.

УДК 004.9

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОБНАРУЖЕНИЯ ДИОДА В ВИДЕОПОТОКЕ

Мисякова В.А.

Научный руководитель – Ковалева И. Л., к.т.н., доцент

В работах [1], [2] и [3] были рассмотрены ключевые преимущества и недостатки нейронной сети YOLO, а также ее архитектура. В рамках данной работы предлагается провести обучение нейронной сети YOLOv4 на собственном наборе данных для обнаружения диода, расположенного на рабочем инструменте 3D-принтера.

Чтобы обучить нейронную сеть для обнаружения диода, необходимо создать собственный набор данных, который должен содержать не менее 700 изображений для обеспечения качественного обучения. Для этого были записаны несколько видео, на которых рабочий инструмент с диодом размещался в различных положениях (рисунок 1). Затем видео были разбиты на кадры. Для ручной маркировки изображений использовалась программа для разметки OpenLabeling. В результате был сформирован набор из 1011 изображений, к каждому из которых были созданы файлы аннотаций. В этих файлах содержится информация о классе объекта и о координатах прямоугольника, ограничивающего этот объект.

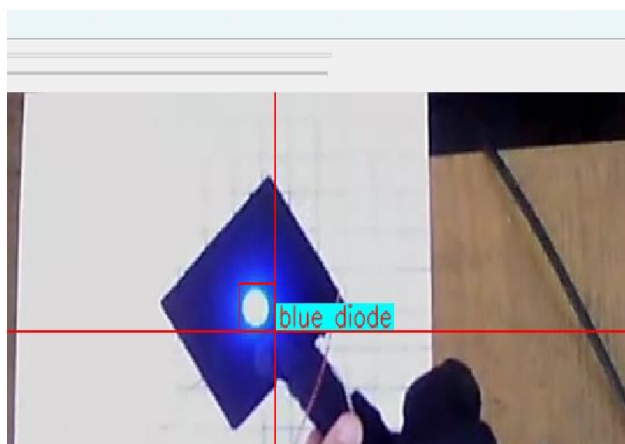


Рис.1. Маркировка изображений в OpenLabeling

YOLOv4 в основном реализуется на фреймворке Darknet [4]. Darknet - это фреймворк с открытым исходным кодом для обучения нейронных сетей, написанный на языке программирования C и с использованием технологии CUDA (Compute Unified Device Architecture) от NVIDIA. Этот фреймворк поддерживает вычисление как на центральном процессоре (CPU), так и на графическом процессоре (GPU).

Для работы с Darknet требуется установить несколько дополнительных компонентов. Библиотека OpenCV используется для работы с изображениями и видео и конфигурируется с использованием CMake. Для ускорения обучения нейронной сети и выполнения инференса на графическом процессоре используются CUDA и cuDNN.

Для обучения YOLO было применено "переносное обучение", которое предполагает использование весов предварительно обученной модели на наборе данных COCO в качестве отправной точки. Также были настроены параметры файла конфигурации, описанные ниже.

`batch = 64` - размер пакета или количество изображений, которые обрабатываются за одну итерацию обучения нейронной сети. Большие значения `batch size` могут ускорить обучение, однако при этом может возникнуть нехватка памяти на GPU. Кроме того, большие значения `batch` могут сделать процесс обучения менее стабильным.

`subdivisions = 32` - параметр отвечает за разделение пакета на подпакеты для снижения потребления памяти на графическом процессоре.

`max_batches = 6000` - общее количество итераций обучения. Установка слишком малого значения может не дать модели достаточно времени для сходимости, а слишком большое значение может привести к переобучению.

`steps = 4800, 5400` - параметр определяет на каких итерациях будет изменяться скорость обучения. Изменение скорости обучения в процессе обучения может помочь модели быстрее сойтись к оптимальным весам и улучшить качество обучения.

`width = 416, height = 416` - размеры входных изображений. Более высокие разрешения могут улучшить качество предсказаний за счет более детальной информации на изображении, но могут потребовать больше вычислительных ресурсов и времени обработки.

`filters = 18` - количество фильтров (и, следовательно, количество карт признаков) для каждого слоя свертки. Каждая карта признаков представляет собой 2D-матрицу, содержащую информацию о различных аспектах изображения, выявленных нейронной сетью в процессе обучения.

Алгоритм обучения принимает на вход обучающий набор, содержащий изображения и их аннотации. Затем начинается итерация обучения (эпоха), где каждое изображение последовательно проходит через все слои модели. Полученные на выходе результаты сравниваются с соответствующими аннотациями с использованием функции ошибки.

В процессе обучения с каждой последующей эпохой веса нейронной сети постепенно изменяются в направлении оптимальных значений, необходимых для достижения точного решения поставленной задачи. Средняя ошибка также уменьшается с каждой итерацией, что отражается на рисунке 2.

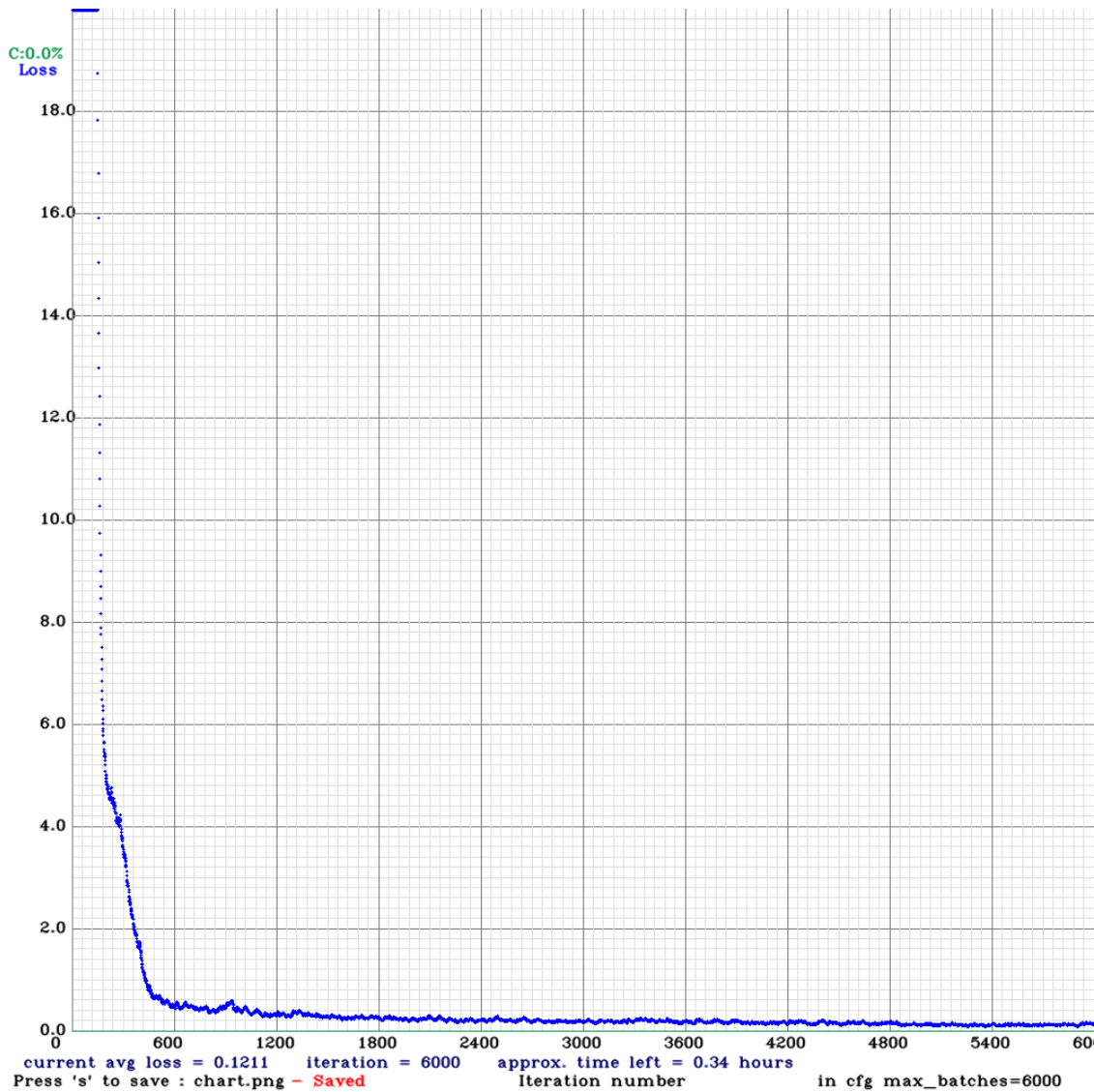


Рис.2. График средних ошибок

Обучение нейронной сети происходило на видеокарте NVIDIA GeForce GTX 1650. Было выполнено 6000 итераций на 1011 изображениях. В результате был получен файл весов diode_6000.weights, который в дальнейшем можно будет использовать для вычисления координат диода и передачи их с компьютера на Arduino.

Для визуализации результатов был написан скрипт на языке программирования Python, который рисует ограничивающую рамку вокруг

объекта, указывает класс, к которому принадлежит объект и процент уверенности.

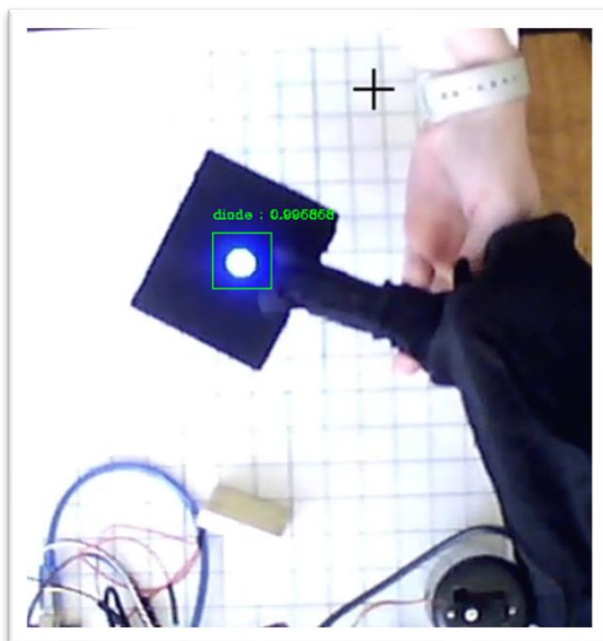


Рис.3. Результат работы скрипта с использованием полученного файла весов

Литература

1. Мисякова В.А., Ковалева И.Л. Обоснование использования нейронной сети уolo для определения положения рабочего инструмента 3d-принтера // Новые техно-логии и материалы, автоматизация производства : материалы международной научно-технической конференции (Брест, 16-17 ноября 2023 г.) / редкол. С.Р. Онысько [и др.] ; Министерство образования РБ, Брестский государственный технический университет. – Брест, 2023. – С. 45-49.
2. Мисякова, В. А. Особенности сети YOLOv4, используемой для определения положения инструмента / В. А. Мисякова, И. Л. Ковалева // Инновационные технологии, автоматизация и мехатроника в машино- и приборостроении [Электронный ресурс] : материалы XII международной научно-практической конференции (Минск, 13 марта 2024 года) : в рамках выставки «Автоматизация, электроника – 2024» / редкол.: А. Р. Околов (гл. ред.) [и др.] ; сост. А. Н. Дербан. – Минск : БНТУ, 2024. – С. 85-86.
3. Мисякова, В. А. Выбор модели нейронной сети для отслеживания движения рабочего инструмента 3D-принтера / В.А. Мисякова, И.Л.Ковалева // Машиностроение : республиканский межведомственный сборник научных трудов / Белорусский национальный технический

университет ; редкол.: В. К. Шелег (гл. ред.) [и др.]. – Минск : БНТУ, 2023. – Вып. 35. – С. 92-96

4. .Darknet [Электронный ресурс]. URL: <https://github.com/AlexeyAB/darknet> (Дата обращения: 05.05.2024)

УДК 004.9

ПРОЕКТИРОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ В ОБРАЗОВАНИИ

Ошуковская О.Э.

Научный руководитель - Ковалёва И.Л., к.т.н., доцент

Задачи проектирования системы управления проектами, применимой в области образования, включают анализ особенностей студенческой сферы коммуникаций и разработку концепции такой системы. Ключевыми факторами являются удобство использования и функциональность, предполагающие простоту в использовании, доступность для студентов и соответствие потребностям студентов в области планирования, взаимодействия между собой и преподавателями, отслеживания и контроля проектов. Требования к такой системе обширны, важно учесть каждый аспект. Поэтому для автоматизации работы с большим объемом факторов и реализации системы предлагается разработать концептуальную модель в результате применения структурного моделирования.

Для формирования концептуальной модели использовался метод моделирования структурных уравнений с частичным наименьшим числом квадратов (PLS-SEM). Метод PLS-SEM (Partial Least Squares Structural Equation Modeling), пример модели которого представлен на рисунке 1, является статистическим методом из области машинного обучения, который часто используется для моделирования отношений между конструктами (абстрактными переменными) в исследованиях [1].

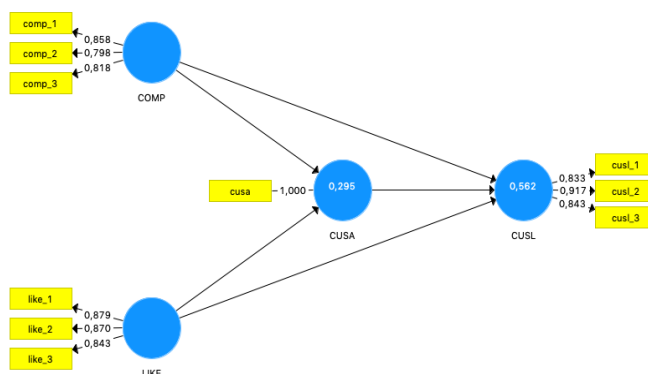


Рис.1. Модель метода PLS-SEM