

**РАСПОЗНАВАНИЕ И ОБНАРУЖЕНИЕ ПРЕПЯТСТВИЙ
ПЕРЕД БЕСПИЛОТНЫМ ЛЕТАТЕЛЬНЫМ АППАРАТОМ
ВО ВРЕМЯ ПИЛОТИРОВАНИЯ**

Аннотация. В данной статье представлено описание решения по распознаванию и обнаружению препятствий перед беспилотным летательным аппаратом (БПЛА) во время пилотирования с использованием встраиваемой системы компьютерного зрения – NVIDIA Jetson TX1.

Ключевые слова: NVIDIA Jetson TX1, компьютерное зрение, машинное обучение, распознавание, обнаружение, БПЛА.

Развитие машинного обучения и нейронных сетей способствовало распространению технологий компьютерного зрения. Под этими технологиями подразумевается получение данных со своеобразного органа зрения – оптической камеры, и их обработка при помощи использования математических методов и моделей. Применение технологий компьютерного зрения можно встретить во многих сферах: в системах видеонаблюдений, в медицине, в производственных процессах и т.д.

В последнее время особую актуальность приобрела задача по использованию технологий компьютерного зрения в беспилотных аппаратах (как летающих, так и наземных): инвентаризация на складах, отслеживание нарушений правил дорожного движения, аэрофотосъемка и многое другое.

Так, например, пилотирование большинством беспилотных летательных аппаратов (БПЛА) в таких задачах ведется вручную при помощи пультов дистанционного управления. Нехватка кадров, возможные потери связи с аппаратом могут привести к лишним затратам. Чтобы автоматизировать деятельность оператора, необходимо наличие системы автономной навигации.

Одним из главных этапов в решении этой проблемы является задача по распознаванию и обнаружению препятствий перед БПЛА во время пилотирования.

В рамках проекта по созданию автономной системы навигации БПЛА необходимо было разработать модель нейронной сети, использование которой позволит обнаруживать и распознавать возможные препятствия перед беспилотником. Использование суперкомпьютера Jetson TX1 обеспечит производительность и энергоэффективность визуальных вычислений прямо во время полета. Архитектура NVIDIA Maxwell™ с 256 ядрами CUDA, 64-битные процессоры, возможности кодирования и декодирования видео 4К, а также интерфейс камеры со скоростью 1400 мегапикселей в секунду делают эту систему лучшей для глубокого обучения, компьютерного зрения и графических вычислений [1].

Подготовка данных для обучения модели проводилась в несколько этапов:

1. Разбиение нескольких видеозаписей общей длиной около 15 минут на кадры при помощи библиотеки Python OpenCV [2].
2. Ручная фильтрация полученных кадров – удаление всех снимков, на которых отсутствуют какие-либо препятствия.
3. Графическая аннотация изображений в XML-файлы формата Pascal VOC при помощи инструмента labelImg [3].

Обучение нейронной сети выполнялось на основании базовой модели MobileNet SSD [4] (рис. 1), которая отличается своей мобильностью, скоростью и точностью. Трансферное обучение строится на том, что знания, накопленные в модели, подготовленной для выполнения одной задачи, могут быть перенесены на другую модель, чтобы помочь в построении прогнозов.

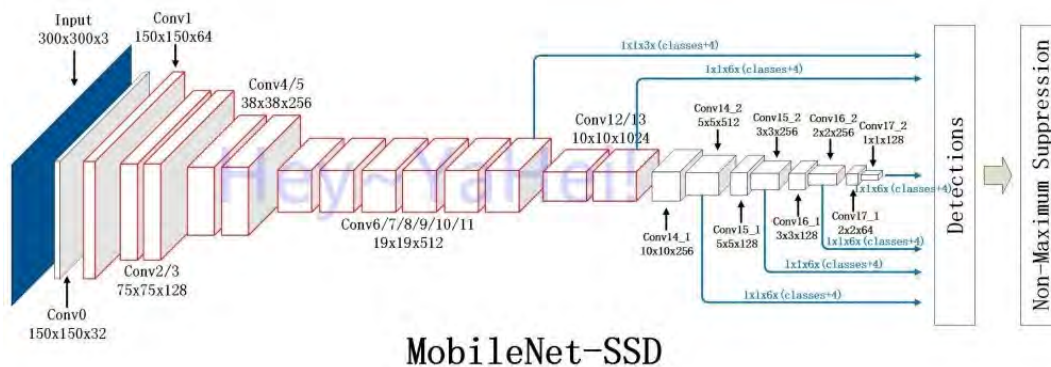


Рис. 1. Модель «MobileNet SSD»

В процессе итерирования обучения нейронной сети было выявлено, что оптимальное количество эпох равнялось 27. Именно при этом значении сумма регрессионной и классификационной потерь данных для валидации была наименьшей – 2,2716 (рис. 2).

Значения потерь данных для валидации на каждой эпохе

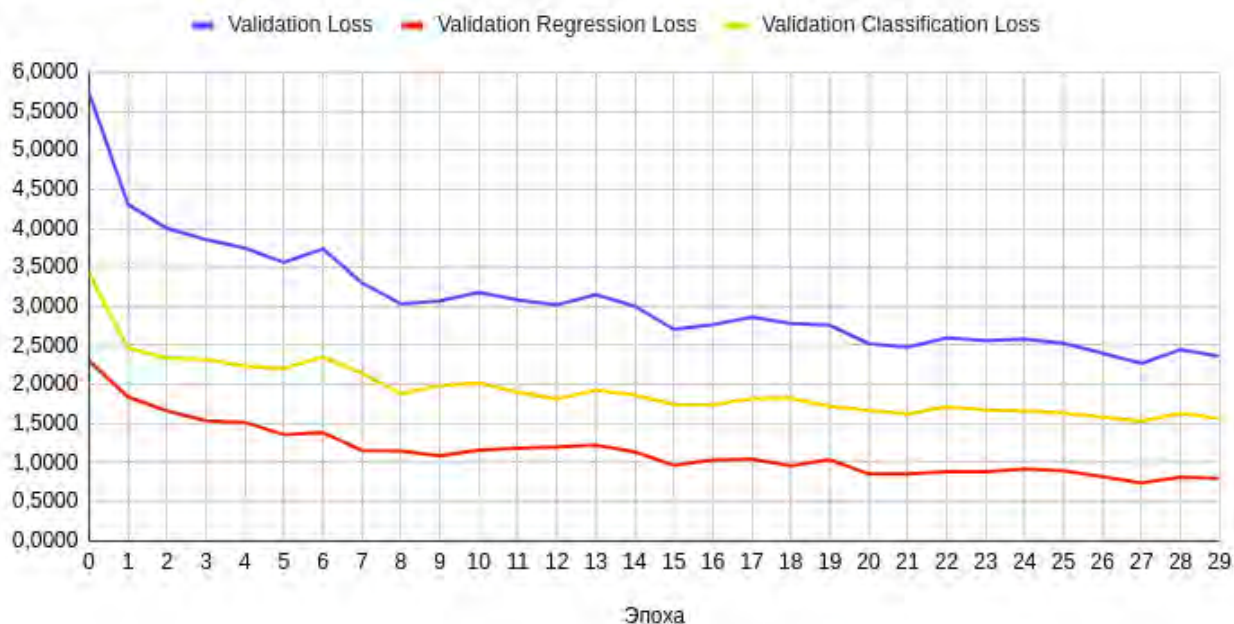
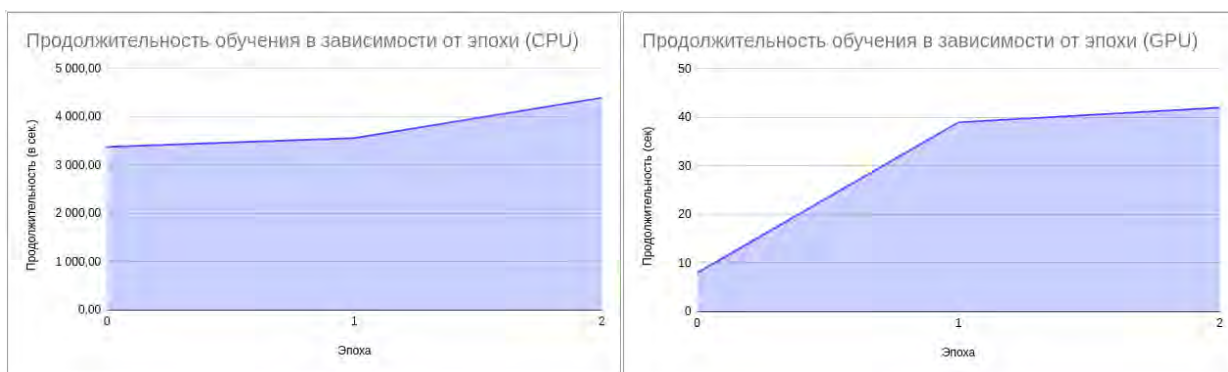


Рис. 2. Зависимость потери для данных валидации от эпох

В ходе выполнения сравнительного анализа выполнения обучения на CPU и GPU микрокомпьютера было выявлено, что время, затрачиваемое на вычисления, с использованием GPU более чем в 60 раз меньше, чем при использовании CPU. На рисунке 3 представлены графики затрат времени на обучение нейронной сети для трех эпох при использовании GPU или CPU.



а) б)

Рис. 3. Зависимость времени обучения от эпох при: а) использовании CPU; б) использовании GPU

Выполнение графических вычислений проходит в высокопроизводительной среде выполнения NVIDIA TensorRT [5], включающей в себя оптимизатор вывода глубокого обучения и среду выполнения, которая обеспечивает низкую задержку и высокую пропускную способность для приложений вывода глубокого обучения.

Для взаимодействия обученной модели с высокопроизводительной средой TensorRT необходима ее конвертация в открытый формат, созданный для представления моделей машинного обучения, – ONNX [6] (рис. 4).



Рис. 4. Описание процесса преобразования размеченных данных в модель нейронной сети

TensorRT предоставляет оптимизацию INT8 и FP16 для производственных развертываний приложений вывода глубокого обучения, таких как потоковое видео, распознавание речи, рекомендации, обнаружение мошенничества и обработка естественного языка. Снижение точности вывода значительно снижает задержку приложений, что является обязательным требованием для многих сервисов реального времени, а также автономных и встроенных приложений [5].

Для тестирования полученной модели использовалась видеозапись с дрона, которая передавалась на модуль обнаружения в виде потока данных (рис. 5).

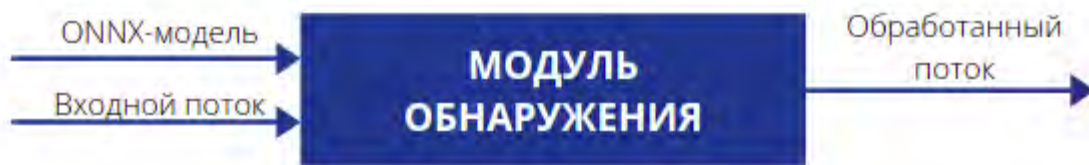


Рис. 5. Описание процесса использования модели нейронной сети

По результатам тестирования, представленным на рисунке 6, было выяснено, что среднее время вычислений нейронной сети равнялось 14,9 мс, а время визуализации – 4,5 мс.



Рис. 6. Производительность обработки входящего видеопотока

Таким образом, можно сделать вывод, что изложенный способ использования компьютерного зрения с применением графического процессора способствует разработке различных систем мониторинга и обработки графических данных в режиме реального времени. В свою очередь, развитие

вычислительных мощностей позволит автоматизировать множество важных процессов, требующих участия человека.

СПИСОК ЛИТЕРАТУРЫ

1. Jetson TX1 Module [Электронный ресурс]. URL: <https://developer.nvidia.com/embedded/jetson-tx1> (дата обращения: 25.05.2021, режим доступа: свободный).

2. Open Source Computer Vision Library [Электронный ресурс]. URL: <https://opencv.org> (дата обращения: 25.05.2021, режим доступа: свободный).

3. Инструмент графической аннотации “labelImg” [Электронный ресурс]. URL: <https://github.com/tzutalin/labelImg> (дата обращения: 25.05.2021, режим доступа: свободный).

4. Модель обнаружения объектов “MobileNet SSD” [Электронный ресурс] URL: https://docs.openvinotoolkit.org/latest/omz_models_model_mobilenet_ssd.html (дата обращения: 25.05.2021, режим доступа: свободный).

5. NVIDIA TensorRT [Электронный ресурс]. URL: <https://developer.nvidia.com/tensorrt> (дата обращения: 25.05.2021, режим доступа: свободный).

6. Open Neural Network Exchange [Электронный ресурс]. URL: <https://onnx.ai/> (дата обращения: 25.05.2021, режим доступа: свободный).