

УДК 004.932

**БЛОЧНЫЙ АЛГОРИТМ ОБНАРУЖЕНИЯ И КЛАССИФИКАЦИИ ОБЪЕКТОВ
С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ
ДЛЯ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ БОЛЬШОГО РАЗРЕШЕНИЯ**

И.Ю. ЗАХАРОВА

(Представлено: канд. техн. наук, доц. Р.П. БОГУШ)

Представлен алгоритм обнаружения объектов на видеоизображениях с разрешением 4К и выше, основанный на блочной обработке кадра с использованием сверточной нейронной сети Yolo v3. Отличительной особенностью при этом является построение и использование групп коррелированных объектов, что позволяет значительно уменьшить вероятность фрагментарной классификации одного объекта для граничных областей блоков. Для программной реализации использовался фреймворк Pytorch и база данных объектов СОСО, которая включает в себя 80 классов.

Для обнаружения объектов на видеопоследовательностях в настоящее время используются архитектуры сверточных нейронных сетей (СНС) такие как: ResNet[1], Inception v4[2], Inception v4b ResNet[2], R-FCN[3], Faster R-CNN[4], Yolo[5], Yolo v2[6], Yolo v3[7]. Однако, в данном случае для изображений и видеопоследовательностей с высоким разрешением актуальной задачей является обеспечение возможности обнаружения объектов небольших размеров, т.к. СНС предполагают значительное уменьшение размеров входного изображения. В работе [8] для решения указанной задачи предложен алгоритм на основе разбиения входного кадра на фрагменты, которые подаются на вход СНС. Однако, при таком подходе один объект может быть разделен на несколько частей, которые в дальнейшем классифицируются как независимые различные объекты. На рисунке 1 показан пример ложной классификации носа самолета (класс «car») в качестве автомобиля. Это связано с тем, что вычисленные СНС признаки выделенного фрагмента более схожи с признаками, соответствующими объекту «автомобиль», чем объекту «самолет».

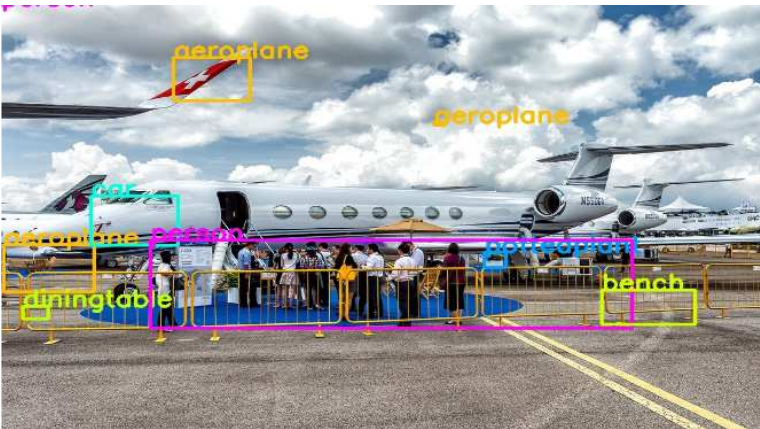


Рисунок 1. – Пример ложной классификации фрагмента объекта

Для решения указанной проблемы предлагается построение и использование групп коррелированных классов объектов (объекты с большим количеством схожих признаков), что позволит значительно уменьшить вероятность фрагментарной классификации одного объекта для граничных областей блоков. Например, для базы данных СОСО [9] сформированы следующие группы коррелированных классов: машина, самолет, автобус, поезд, грузовик; лодка, доска для серфинга; птица, самолет, воздушный змей и др. Для обеспечения возможности классификации большого количества объектов выбрана СНС YOLO v3.

Разработанный алгоритм обнаружения объектов на видеоизображениях с разрешением 4К и выше, основанный на блочной обработке кадра с использованием сверточной нейронной сети Yolo v3 и использовании коррелированных объектов требует выполнения следующих шагов:

1. Расчет размера блока. Размер блока выбирается исходя из минимизации ложноотрицательного обнаружения и определяется как

$$Bl = \frac{Ii}{\Pi} \tag{1}$$

где B_l – размер блока, l_i – размер исходного изображения в пикселях, для разрешения 4К максимальный размер длинны или ширины варьируется от 3600 до 4200 пикселей, l – размер входного слоя СНС, для СНС Yolo v3 он составляет [416×416] пикселей.

2. Обнаружение областей интереса на фрагментах кадра. Для этого каждый блок поступает на вход СНС Yolo v3. Выходным результатом для каждой области интереса, которая содержит объект или фрагмент объекта, является набор данных вектор $[X_1, Y_1, X_2, Y_2, \text{Confidence}, \text{Class_index}, B_n]$, где X_1, Y_1 – координаты левого верхнего угла области интереса, X_2, Y_2 – координаты правого нижнего угла области интереса, Confidence – степень уверенности системы, Class_index – индекс класса объекта, B_n – номер блока в котором осуществлялось обнаружение.

3. Нахождение координат обнаруженных областей на кадре. Обнаруженные координаты областей интереса на фрагментах кадра, записанные в текстовый документ, пересчитываются на координаты входного кадра:

$$\begin{aligned} \text{Add_X} &= B_n \bmod s; \\ \text{Add_Y} &= B_n \text{ div } s \end{aligned} \quad (2)$$

где Add_X и Add_Y – добавочные значения для X и Y координат соответственно, s – количество разбиений для одной стороны кадра

4. Объединение областей интереса. Для объединения соседних областей интереса рассчитывается коэффициент Жаккарда:

$$Iou = \frac{In}{B_1(x_2 - x_1) \cdot B_1(y_2 - y_1) + B_2(x_2 - x_1) \cdot B_2(y_2 - y_1) - In}, \quad (3)$$

где

$$\begin{aligned} In &= \left(\min(B_1(x_1) + B_1(x_2 - x_1), B_2(x_1) + B_2(x_2 - x_1)) - \max(B_1(x_1), B_2(x_1))) \right) \times \\ &\left(\min(B_1(y_1) + B_1(y_2 - y_1), B_2(y_1) + B_2(y_2 - y_1)) - \max(B_1(y_1), B_2(y_1)) \right), \end{aligned} \quad (4)$$

где B_1 – область интереса, $B_1(X_1, Y_1)$ – координаты левого верхнего угла области интереса, $B_1(X_2, Y_2)$ – координаты правого нижнего угла области интереса

Если рассчитанное значение коэффициента Жаккарда выше порогового и объекты относятся к одной группе, то данные области интереса объединяются в один объект. В данной работе пороговое значение выбиралось аналитически и составило 50%. Класс объекта определяется максимальной величиной Confidence для объединяемых областей.

Алгоритм реализован с использованием языка программирования Python, с фреймворка Pytorch [10] и библиотеки OpenCV [11]. В качестве СНС была использована СНС Yolo v3 предварительно обученная на базе данных СОСО содержащей 80 классов объектов.

Рисунок 2 свидетельствует о том, что предложенный алгоритм позволил исключить ложное обнаружение автомобиля, как показано на рисунке 1.

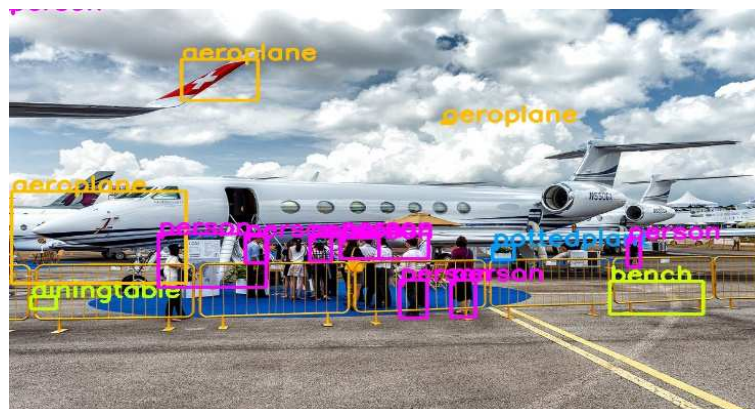


Рисунок 2. – Пример обнаружения и классификации объектов на основе предложенного алгоритма

Результаты экспериментов показали, что предложенный алгоритм позволяет уменьшить вероятность фрагментарной классификации одного объекта для граничных областей блоков, возникающих из-за разбиения входного кадра на фрагменты. Однако, наблюдается некоторое количество ложноположительных результатов обнаружения. Этот недостаток можно исключить, переобучив нейронную сеть на выборке, включающей фрагменты изображений.

ЛИТЕРАТУРА

1. Deep Residual Learning for Image Recognition / K. He [et al.] // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA. – 2016. – P. 770–778.
2. Szegedy, C. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning / C. Szegedy, S. Ioffe, V. Vanhoucke // Proc. of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017. – 4278–4284.
3. Object Detection via Region-based Fully Convolutional Networks [Electronic resource]. – Mode of access: <https://arxiv.org/pdf/1605.06409.pdf>. – Date of access: 20.12.17.
4. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Electronic resource]. – Mode of access: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>. – Date of access: 20.12.17.
5. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon [et al.] // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA. – 2016. – P. 779–788.
6. YOLO9000 Better, Faster, Stronger [Electronic resource]. – Mode of access: <https://arxiv.org/pdf/1612.08242.pdf>. – Date of access: 20.12.17.
7. YOLOv3: An Incremental Improvement [Electronic resource]. – Mode of access: <https://arxiv.org/pdf/1804.02767.pdf>. – Date of access: 15.06.18.
8. An Effective Object Detection Algorithm for High Resolution Video by Using Convolutional Neural Network / Denis Vorobjov [et al.] // Huang T., Lv J., Sun C., Tuzikov A. (eds) Advances in Neural Networks. – ISSN 2018. Lecture Notes in Computer Science, vol 10878, Cham, Switzerland. – 2018. – P. 503–510.
9. COCO – Common Objects in Context [Electronic resource]. – Mode of access: <http://cocodataset.org>. – Date of access: 15.06.18.
10. PyTorch [Electronic resource]. – Mode of access: <https://pytorch.org>. – Date of access: 15.06.18.
11. OpenCV library [Electronic resource]. – Mode of access: <https://opencv.org>. – Date of access: 15.06.18.