Детекция объектов малого размера на снимках высокого разрешения

Аннотация

В статье рассматривается упрощенное решение задачи детекции малых объектов, занимающих менее 1% площади изображения. В качестве алгоритма извлечения признаков (feature extractor) используются классификаторы EfficientNet и MobileNet из библиотеки Tensorflow, предобученные на данных ImageNet. Карта признаков берется с последних слоев нейросети, предсказание класса делается по вектору признаков в глубину. В связи с малым размером объектов, уточнение координат не производится. Таким образом, упрощается архитектура сети и снижается количество вычислений. Эта особенность является особенно важным для мобильных приложений, т.к. позволяет уменьшить время детекции и расход батареи.

Введение

Детекция объектов это довольно распространенная задача в области машинного зрения. Широкое применение в различных приложениях получили модели SSD [SSD] и YOLO [Yolo].

Ежегодно в лесах и болотах теряются тысячи людей. Их поиск осуществляется различными способами, в т.ч. с помощью БПЛА и нейросетей. Задача поиска пропавших людей на снимках БПЛА имеет особенность – размер объектов очень мал по отношению к размеру изображения и встречаются эти объекты очень редко.

Стандартные алгоритмы детекции объектов необходимо настраивать, т.к. из по умолчанию они ищут объекты разных размеров. Мы предполагаем, что при наличии априорной информации о малом размере объекта, нет нужды искать на снимке объекты с большой площадью. Кроме того, в известных алгоритмах много вычислений производится для уточнения координат объекта и постобработки non-maximum supression [NMS]. Это создает дополнительную нагрузку при работе моделей на мобильные устройства, что увеличивает время

детекции и расход батареи. В статье предлагается использовать более простой метод детекции малых объектов с грубой оценкой их координат, который требует меньше вычислений и более прост в обучении.

Метод исследования

БПЛА оснащаются камерами высокого разрешения: при 12 МП изображения имеют разрешение около 4000x3000 точек. Примерный размер человека при высоте съемки 50-100 метров составляет около 50 - 150 px. Предобученные классификаторы имеют входное разрешение от 224 до 600 px. Если сжимать размер исходного изображения до такого разрешения, то по каждой оси в 10 раз, площадь сократится в 100 раз.

С одной стороны, размер человека не должен быть очень малым, т.к. это очень снизит качество распознавания. С другой стороны, обработка несжатых изображений потребует очень много вычислений. Поэтому исходное изображение необходимо разбить на некоторое оптимальное количество окон (кропов). Для этого выбирается количество окон по каждой оси W, H и их размер CROP_SIZE. Исходное изображение сжимается до ширины W * CROP_SIZE и высоты H * CROP_SIZE, из которого берется W * H кропов заданного размера.

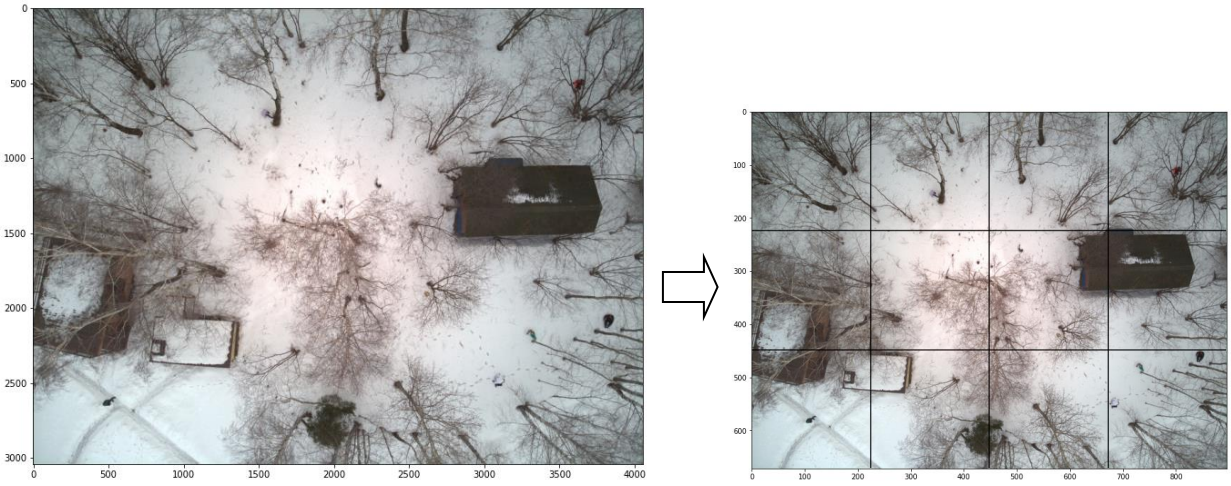


Рисунок 1. Схема разделения изображения на окна.

Каждый кроп подается на вход предобученного классификатора. Карта признаков берется с последних слоев и преставляет собой тензор размером (W_OUT, H_OUT, N_FEATURES). Например, для EfficientNetB0 эти значения составляют (7, 7, 1280).

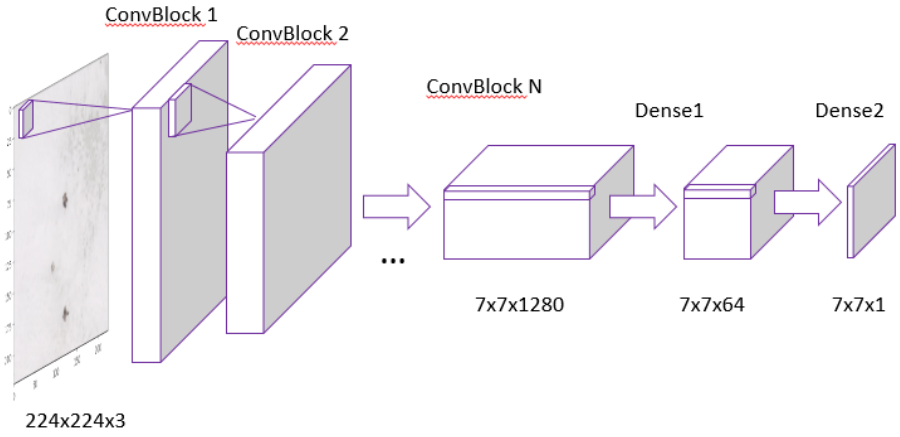
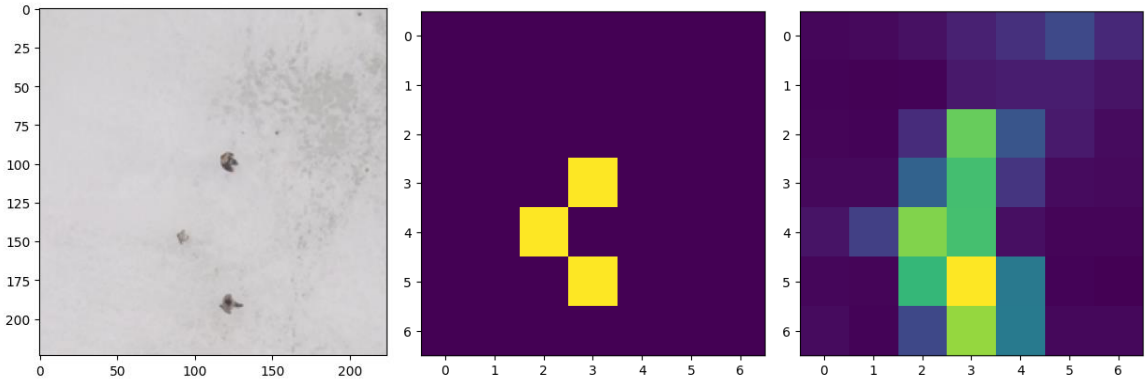


Рисунок 3. Архитектура детектора.

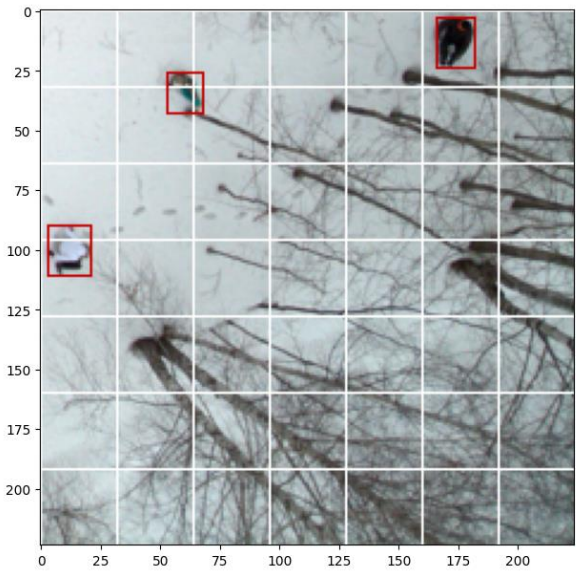
Мы считаем, что одному элементу карты признаков соответствует некоторая область изображения (рецептивное поле). В приведенном примере его размер составил 224 / 7 = 32x32 px.



а) б) в)

Рисунок 3. Выходная карта признаков (feature map).
а) Исходное изображение. б) Разметка для обучения. в) Результат предсказания.

Имеется стандартная разметка объектов в виде прямоугольников. Единица назначается тому квадрату, в котором находится наибольшая часть площади границ прямоугольной разметки (bounding box).



0	0	0	0	0	1	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Рисунок 4. Схема разметки карты признаков.

Для каждого элемента карты признаков мы делаем бинарную классификацию по N_FEATURES с помощью двухлойной сети с размером скрытого слоя 128, активация ReLU, на выходе sigmoid. В результате для каждой области исходного изображения мы получаем оценку вероятности того, содержит ли она искомый объект.

В качестве функции потерь использована Focal Cross Entropy [RetinaNet]. Метрики стандартные для бинарной классификации - precision, recall, RocAuc. [ML metrics]. В контексте задачи поиска людей метрика полноты важнее чем точность, поэтому при расчете метрик порог классификации (threshold) брался равным 0.3.

Результаты исследования

Исследования проводились на датасете Liza Alert Drone Dataset (LADD) v4: 1036 изображений, в обучающей выборке - 911, в тестовой - 125? Для извлечения признаков использовались предобученные на датасете ImageNet модели из пакета tf.keras.applications: EfficientNetB0, EfficientNetB7, MobileNetV3Small, MobileNetV3Large.

Аугментации для обучающей выборки реализованы средствами библиотеки albumentations [albumentations]:

1. Случайный кроп размером 224x224 из области 260x260 для того, чтобы человек попадал в разные области рецептивного поля;

2. Горизонтальное и вертикальное отражение;

3. Поворот на 90, 180, 270 градусов;

4. Случайное изменение контраста и яркости.

Использовано косинусное затухание скорости обучения, график приведен на рисунке 5.

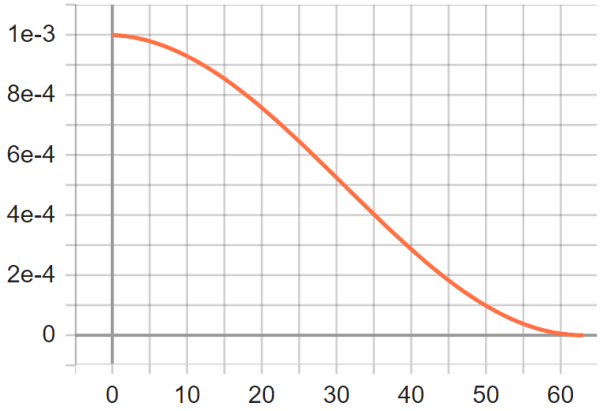
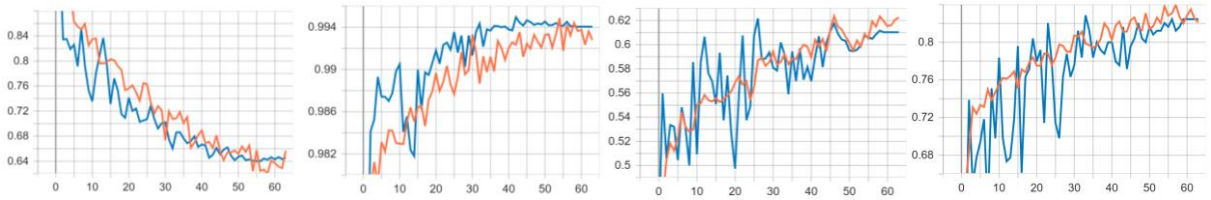


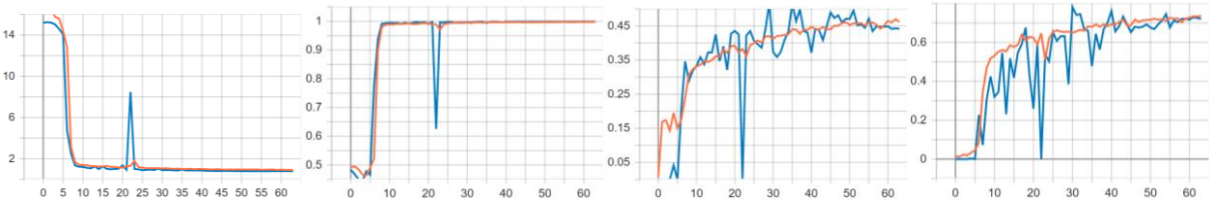
Рисунок 4. Затухание скорости обучения в процесс обучения сетей.

В качестве оптимизатора использовался Adam [Adam], обучение длилось 64 эпохи, размер батча равен 8. Процесс обучения различных моделей приведен на рисунках 5-8.



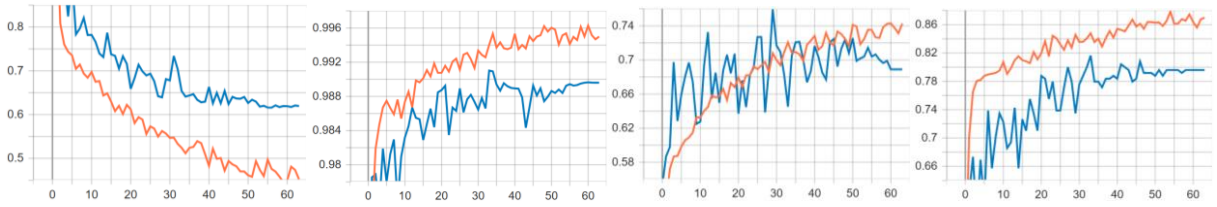
а) б) в) г)

Рисунок 5. Процесс обучения сети EfficientNetB0. а) Функция потерь (FocalLoss), б) Интегральная метрика (ROC AUC), в) Точность (Precision), г) Полнота (Recall)



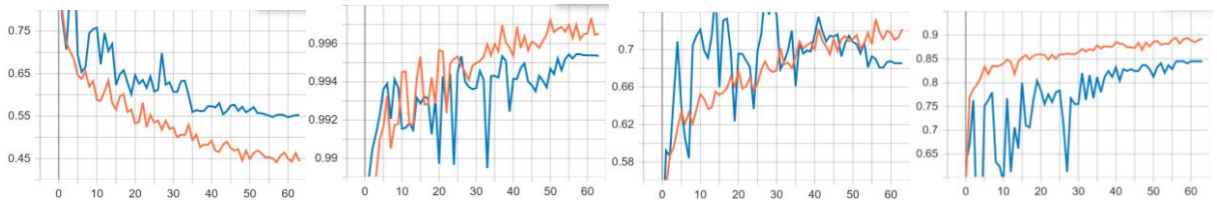
а) б) в) г)

Рисунок 6. Процесс обучения сети EfficientNetB7. а) Функция потерь (FocalLoss), б) Интегральная метрика (ROC AUC), в) Точность (Precision), г) Полнота (Recall)



а) б) в) г)

Рисунок 7. Обучение сети MobileNet3Small. а) Функция потерь (FocalLoss), б) Интегральная метрика (ROC AUC), в) Точность (Precision), г) Полнота (Recall)



а) б) в) г)

Рисунок 8. Процесс обучения сети MobileNet3Large. а) Функция потерь (FocalLoss), б) Интегральная метрика (ROC AUC), в) Точность (Precision), г) Полнота (Recall)

Таблица 1. Результаты обучения

Model	Loss	RocAuc	Precision	Recall
EfficientNetB0	0.64	99.3	59	82
EfficientNetB7	0.86	99.8	42	75
MobileNet3Small	0.62	98.6	69	78
MobileNet3Large	0.55	99.6	68	85

Выводы и дальнейшие исследования

В результате исследования подтверждена гипотеза о возможности применения последних слоев классификатора для детекции малых объектов без уточнения их границ. В данном исследовании не использовалось дообучение последних слоев предобученных классификаторов из-за относительно небольшого количества данных. Возможно, на более крупных датасетах transfer learning позволит извлекать более информативные признаки. Алгоритм перехода от координат bounding box к индексам карты признаков также можно улучшить за счет назначения величин вероятностей, пропорциональных площади объекта, попавшего в определенную область.

Список источников

[Adam] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 2014.

[Albumentations] Buslaev, Alexander, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. "Albumentations: fast and flexible image augmentations." Information 11, no. 2 2020

[ML metrics] https://keras.io/api/metrics/classification_metrics/

[RetinaNet] Tsung-Yi Lin, Priya Goyal, R. Girshick, Kaiming He, Piotr Dollár. Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018

[SSD] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector. 2016

[YOLO] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. 2016.