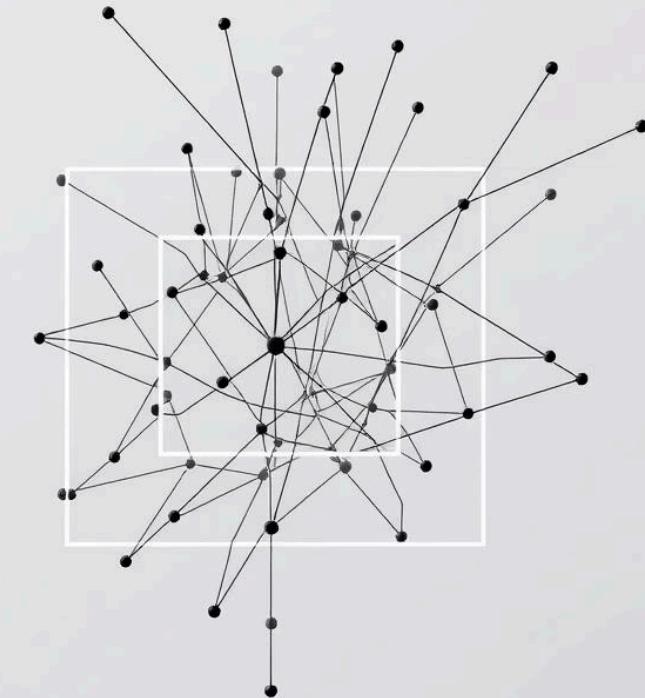


Повышение точности малых моделей детекции объектов

Плугин Олег 22503



Повышение точности малых моделей детекции объектов

Необходимо найти баланс между высокой точностью и эффективным использованием ресурсов

Постановка задачи: обнаружение объектов на ограниченных ресурсах

В задачах компьютерного зрения для мобильных и встроенных устройств, критически важно обеспечить высокую точность обнаружения объектов при сохранении скорости работы в реальном времени и минимальных вычислительных затратах.

Стандартные ограничения устройств



Процессор

4-ядерный, тактовая частота 2.2 ГГц.



Оперативная память

До 4 ГБ ОЗУ.



Графический процессор

Обработка изображений размером до 1920 × 1080.

Оптимизация под мобильные GPU-архитектуры.

Проблема и цель повышения точности

При приведенных аппаратных ограничениях использование крупных нейронных сетей (например, стандартных версий YOLO или Faster R-CNN) становится невозможным. Требуется упрощение архитектуры:

- Уменьшение числа слоев и их глубины.
- Сокращение количества фильтров (каналов).
- Снижение общего числа обучаемых параметров.

Основная проблема: Упрощение архитектуры нейронной сети приводит к значительному снижению точности (mAP) распознавания.

Ключевой целью является достижение точности, максимально приближенной к исходной, более крупной модели, при существенно меньшем размере панели и более высокой скорости работы.

Параметры задачи

Категория	Основные параметры	Влияет на
Архитектура сети	количество слоёв n , шаг карты признаков (stride feature map) $feat_{st}$, количество фильтров f	скорость работы (FPS) и точность (mAP)
Входные данные	размер входного изображения (W, H)	точность распознавания
Якори (anchors)	количество якорных рамок k и их размеры (w_a, h_a)	качество локализации объектов и IoU
Метрики оценки	mAP (mean Average Precision) $Precision$ $Recall$	среднее значение точности по всем классам и порогам IoU . Основная метрика для сравнения моделей (ед.: %) доля найденных моделью объектов действительно является корректной (ед.: %/0–1) доля реальных объектов, найденная моделью (ед.: %/0–1)
	IoU (Intersection over Union)	точность покрытия предсказанными областями исходных (ед.: 0–1)
	FPS (frames per second)	количество обрабатываемых кадров в секунду (ед.: $\frac{\text{кадры}}{\text{сек}}$)
	T_1	время обработки одного кадра (ед.: мс)

Формулировка задачи оптимизации

Данную задачу можно представить в виде задачи оптимизации.

Пусть существуют параметры:

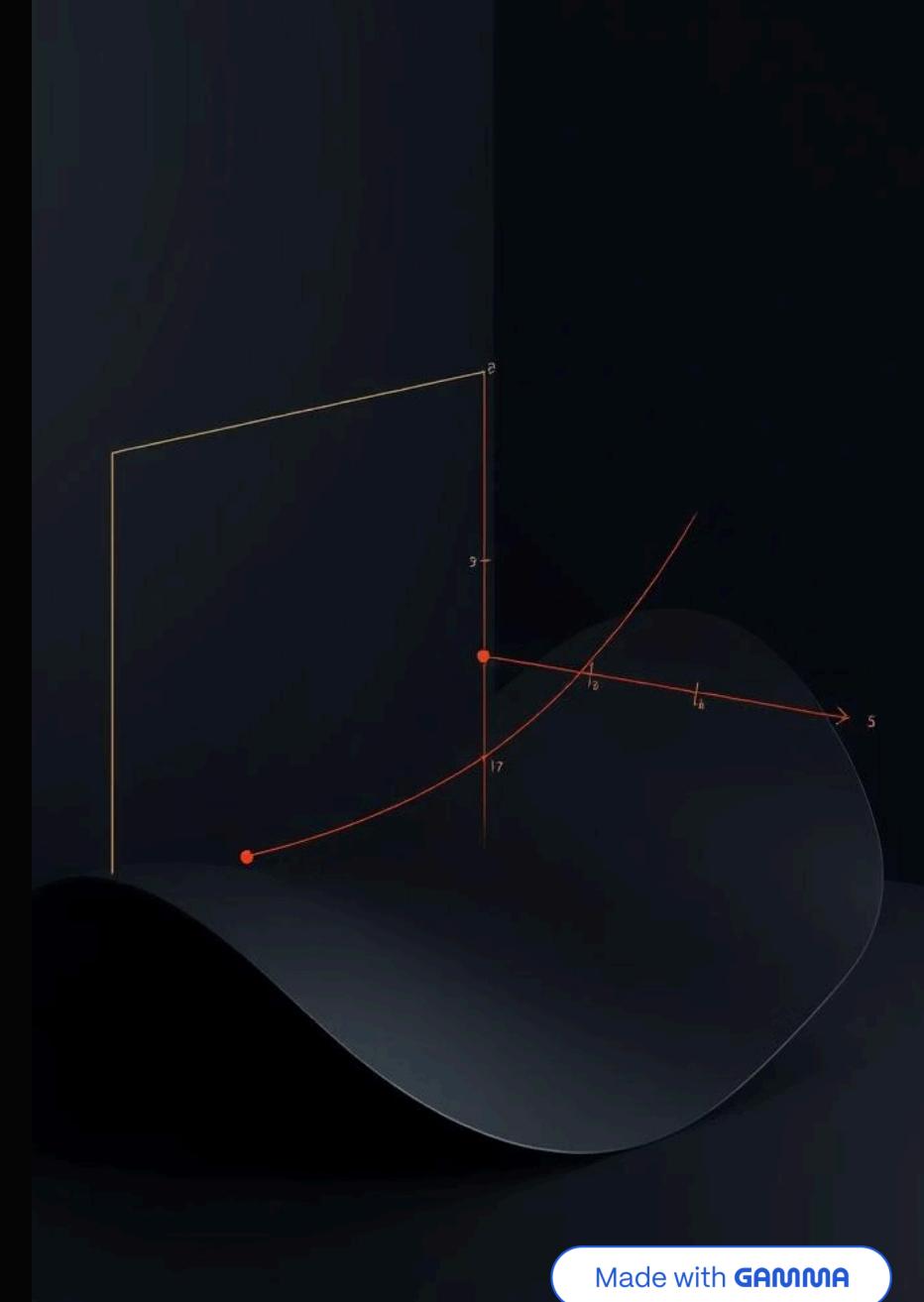
- W - ширина входного изображения (пиксели)
- H - высота входного изображения (пиксели)
- n - количество слоёв нейронной сети
- k - количество якорных рамок

Необходимо набрать такой набор параметров (W, H, n, k) при котором:

- точность $mAP \rightarrow \max$
- время обработки $T_1 \rightarrow \min$, т.е. частота кадров $FPS \rightarrow \max$ (т.к. $FPS = \frac{1}{T_1}$)

при выполнении следующих ограничений:

- $FPS(W, H, n, k) \geq FPS_{min}$ (FPS_{min} задается исходя из задачи)
- $n < n_{max}$
- $mAP(n) \geq mAP_{n_0} - \Delta$, где n_0 - начальное количество слоев, а Δ - допустимое снижение точности работы модели в процентах по сравнению с исходной (задается исходя из характеристик задачи)



Алгоритм повышения точности сети:

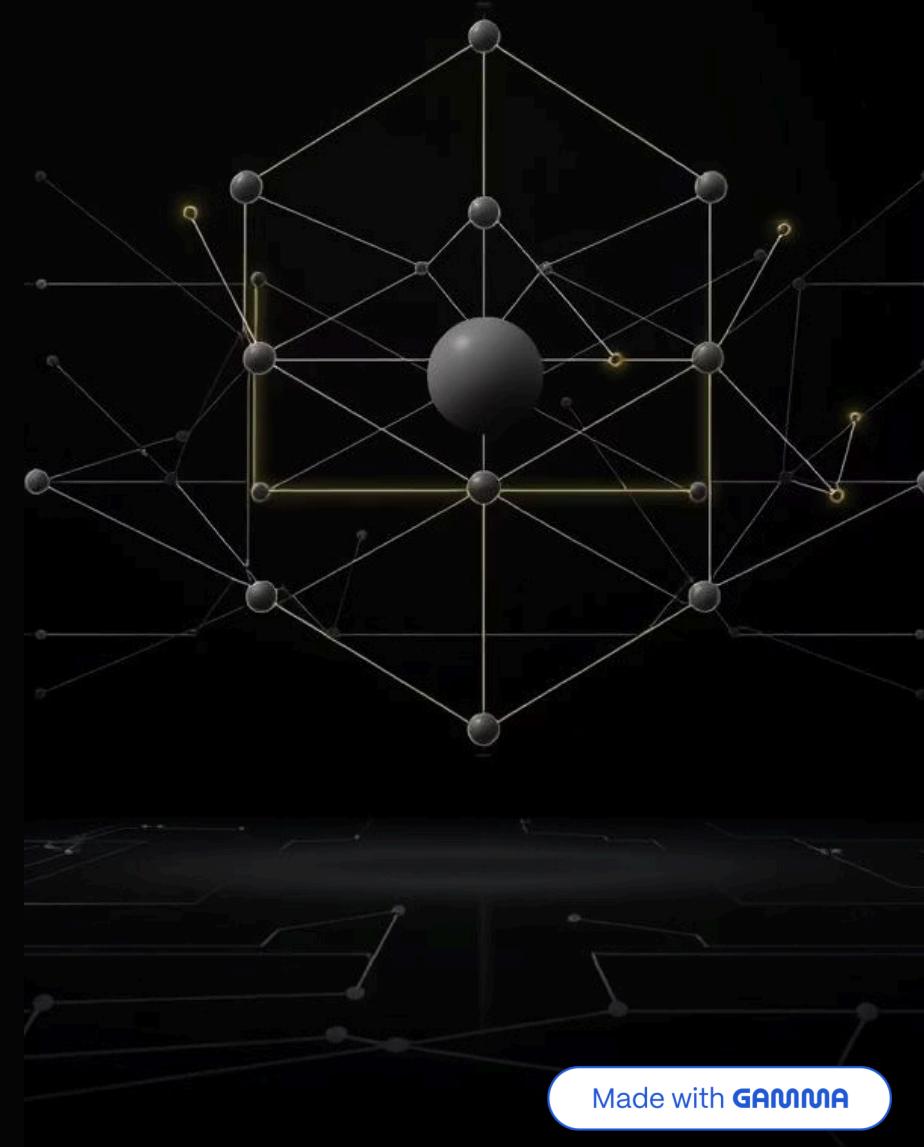
Архитектура и входные данные

Анализ архитектуры

1. Зафиксировать начальное количество слоев как n_0
2. Уменьшить количество слоев до $n < n_0$ следующим образом:
 - Установить диапазон поиска $n \in [n_{min}; n_{max}]$
(изначально $n_{min} = 5$, $n_{max} = n_0$)
 - Вычислить $n_{mid} = (n_{min} + n_{max})/2$
 - Измерить метрики: $mAP(n_{mid})$, $FPS(n_{mid})$
 - При $mAP(n_{mid}) \geq mAP_{min}$ и $FPS(n_{mid}) \geq FPS_{min}$ уменьшаем глубину модели
 $n_{max} = n_{mid}$ и запоминаем значение $n_{correct} = n_{mid}$
 - Иначе $n = n_{correct}$ (находим последнее удовлетворяющее значение)

Изменение входных данных

1. Изменить размер входного изображения $W_n \times H_n$ по числам, кратным 32
(примеры: уменьшение $640 \rightarrow 416 \rightarrow 320$, увеличение $416 \rightarrow 512 \rightarrow 640$)
2. Оценить изменения mAP и FPS
3. Отметить зависимость: $mAP \propto f_1(W_n, H_n)$, $T_1 \propto f_2(W_n, H_n)$, $FPS = 1/T_1$ (T_1 – время обработки обратно пропорционально частоте кадров). f_1 – функция, описывающая зависимость точности от разрешения входного изображения, f_2 – функция, описывающая зависимость времени от разрешения входного изображения.



Алгоритм повышения точности сети: Оптимизация якорных рамок

$$(w_r, h_r) = (W_n/W_i \times w_t, H_n/H_i \times h_t)$$

Формула обеспечивает пропорциональное масштабирование якорных рамок под новый размер.

$$(w_c, h_c) = k - means(w_r, h_r)$$

$$\bar{S}(k) = \frac{1}{n} \sum_{i=1}^n S(i)$$

$$S_k = (b - a) / max(a, b),$$

1. Масштабировать рамки по формуле:

- (w_t, h_t) - граничные рамки исходного значения (пиксели)
- (w_r, h_r) - ремасштабированные рамки (пиксели)
- (W_n, H_n) - исходное разрешение входа сети (пиксели)
- (W_i, H_i) - ширина и высота входного изображения (пиксели)

2. Провести кластеризацию k -средних по ширине и высоте рамок:

- (w_c, h_c) - k центров кластеров, соответствующие усредненным размерам объектов

3. Найти оптимальное количество кластеров k с помощью силуэта средних коэффициентов:

- Проэкспериментировать со значением k от 3 до 15
- При каждой итерации вычислять средний силуэт по всем точкам:
 - n - общее количество элементов, участвующих в кластеризации
 - a - среднее внутрикластерное расстояние,
 - b - ближайшее межкластерное расстояние
- Максимум среднего $\bar{S}(k)$ определит оптимальное значение k

Алгоритм повышения точности сети: Нормализация якорей и оценка результатов

Нормализация якорей

Нормализуем размеры якорных рамок

$$(w_a, h_a) = (w_c / \text{feat}_{st}, h_c / \text{feat}_{st})$$

- w_a, h_a - размеры рамок в масштабе feature map, то есть в тех единицах, с которыми работает последний слой модели
- $\text{feat}_{st} = 2^n$ - шаг карты признаков, где сеть содержит n слоев max pooling

Это необходимо, чтобы размер якорей соответствовал масштабу признаков на последнем уровне сети.

Оценка результатов

1. Измерить метрики *Precision*, *Recall*, *mAP*, *FPS*, *IoU*
2. Проанализировать зависимости параметров
 - увеличение $W_n, H_n \rightarrow$ рост *mAP*, падение *FPS*
 - увеличение $k \rightarrow$ рост *IoU*
 - уменьшение $n \rightarrow$ рост *FPS*, падение точности *mAP*
3. Сделать выводы, удовлетворяют ли текущая конфигурация и метрики целевым ограничениям

Заключение

Представленный подход позволяет подбирать архитектурные и входные параметры модели под доступные вычислительные ресурсы.

Сопоставимые показатели

Достижение показателей, сопоставимых с исходной моделью при оптимизации архитектуры.

Эффективность ресурсов

Значительно меньшая сложность модели, что важно для систем компьютерного зрения, работающих на мобильных и малопроизводительных устройствах.

Источники

1. Tarek Teamaa, Hongbin Ma, Ali Maher, and Mohamed A. Kassab. 2019. Real Time Object Detection Based on Deep Neural Network. In Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part IV. Springer-Verlag, Berlin, Heidelberg, 493–504.