Распознавание объектов на художественных изображениях*

Лемтюжникова Д. В., Апишев М. А., Козинов А. В.

kozinov.av@phystech.edu

Задача обнаружения объектов на художественных изображений имеет несколько особенностей по отношению к обнаружению объектов на фотографиях. ведь картины написаны с определённым стилем, который не всегда передаёт действительность. Объекты не всегда поддаются определённым правилам: например, они могут быть разной формы или с неточными границами

Ключевые слова: *Анализ изображения, CNN, boxing изображений.*

1 Введение

В данной работе рассматривается задача понимания художественных изображений алгоритмами машинного обучения. Основная цель - это распознать на изображении ключевые символы, а с помощью инфрмации о них и информации об изображении сформировать текстовое описание.

Стоит определиться с тем, что такое **символ**. Существует много определений понятия символ, одно из них следующее: "Символ имеет очень сложное значение, потому что не подчиняется причине; он всегда предполагает много значений, и эта многозначность не может быть сведена к единой логической системе"(В.И. Иванов). И принято решение определять символ с помощью экспертов. Таким образом сформирована база размеченных изображений с выделенными фрагментам, которым сопоставлено название символа и его значение при данном контексте.

На предложенной выборке производится обучение свёрточной нейронной сети. А далее процесс анализа изображения происходит следующим образом: выделяются основные объёкты и фон, производится классификация полученных объёктов, далее для каждого из объёкта выбирается описание на основе того, как элементы связаны друг с другом и с фоном.

Это решение может быть использовано для оценки стоимости картины перед аукционом. Но в отличии от подкхода [5], который анализирует картину целиком, представленный подход учитывает наличие специальных смысловых единиц — символов.

* Научный руководитель: Стрижов В. В. Задачу поставила: Лемтюжникова Д. В.

В данной статье рассматривается упрощённая формулировка задачи, в которой нужно определить, присутствует ли символ на изображении или нет.

2 Постановка задачи

2.1 Входные данные

На вход подаётся RGB изображение. Под изображением мы понимаем матрицу I размера $H \times W \times 3$, где $H = W = 224$, причём элементы матрицы определяются следующим образом $I_{i,j,k} \in \overline{0, 255}$. Такой размер изображения выбран эмпирическим путём([LINK]), так как достигается баланс скорости обработки и качества распознавания.

2.2 Выходные данные

Для каждого класса $m \in \overline{1..M}$ следует получить список ограничивающих прямоугольников $B_1^m, B_2^m ... B_{a_m}^m$ с соответсвующими рангами уверенности $t_l^m \in (0, 1)$, где 0 — наименее вероятное совпадение, 1 — наиболее вероятное, $l \in \overline{1..a_m}$.

2.3 Качество решения

Обозначение	Определение
B_p^m	Прямоугольник, выделяющий объект класса m. Получен в качестве предсказания алгоритма.
B_{gt}^m	Прямоугольник, точно выделяющий объект класса m. Поступает из датасета вместе с изображением.
$area(S)$	Площадь, количество пикселей, заданной области S
$IoU(A, B)$	Отношение $\frac{area(A \cap B)}{area(A \cup B)}$, где A и B — прямоугольники, выделяющие область на изображении
$T_P(m)$	Число *верно обнаруженных* объектов зафиксированного класа m.
$F_P(m)$	Количество ограничивающих прямоугольников, которые *неверно обнаруживают* объект заданного ласса m.
$F_N(m)$	Количество ограничивающих прямоугольников, которые *неверно обнаруживают* объект класса, отличного от m.
$Recall(m)$	Отношение $\frac{T_P(m)}{T_P(m)+F_N(m)}$.

$Precision(m)$	Отношение $\frac{T_P(m)}{T_P(m)+F_P(m)}$.

Для измерения качества решения была выбрана метрика Average Precision (AP) из соревнования VOC2007 challenge([1]), так как она не зависит от метода решения и является достаточно употребляемой для задач подобного типа ([4], [6], [2]). А для этого введём несколько дополнительных определений:

Для начала отметим, что B_p^m **верно обнаруживает объект**, если найдётся B_{gt}^m , такой что отношение $IoU(B_p^m, B_{gt}^m) \geqslant 0.5$ и максимально среди всех $B_i^m, i \in \overline{1..a_m}$.

В этом методе *precision/recall curve* для заданного класса m вычисляется на основе полученных рангов уверенности. Для каждого ранга уверенности t рассматриваются предсказанные прямоугольники с рангом большим либо равным t. И для зафиксированных прямоугольников вычисляется точка $(Precision(m), Recall(m))$.

Определим тогда $p(\tilde{r})$ как $Precision$ при заданном $Recall$ $\tilde{r}$. А $Precision$ при заданном $Recall$ уровне r — это $p_{interp}(r) = \max\limits_{\tilde{r} \geqslant r} p(\tilde{r})$

Тогда AP определяется как средняя $Precision$ на заданном множестве из одиннадцати $Recall$ уровней $[0, 0.1, ..., 1]$:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, ..., 1\}} p_{interp}(r)$$

3 Выбор решающей модели

Для решения задачи детекции объектов на изображении была выбрана архитектура нейронных сетей Faster R-CNN [6]. Данная модель состоит из трёх модулей: глубокая свёрточная нейросеть, которая извлекает карту признаков из изображения, полностью сверточная нейросеть, которая предсказывает возможные регионы с объектами, и детектор FAST R-CNN [2].

3.1 Глубокая свёрточная нейросеть

В качестве глубокой свёрточной нейросети была выбрана модель ResNet101 [3]. Но в качестве выхода выступет матрица признаков. Для этого из стандартной архитектуры удаляются последние два слоя: average pooling и полносвязный слой.

3.2 Полностью свёрточная нейросеть

В этой моделе использование полносвязных слоёв заменено на последовательность сверток/пулингов. Выходом данной нейронной сети является список предложенных обрамляющих прямоугольников, причём каждый с рангом уверенности.

Для получения предсказаний используется небольшая нейросеть, которая в качествы входа использует скользящее по карте признаков окно размера 3×3. Далее вход этой нейросети сжимается первым свёрточным слоем до размера 1×1. Затем параллельно следуют регрессионный слой, который предсказывает координаты обрамляющего прямоугольника, и слой классификации, который для каждого класса предсказывает ранг уверенности.

Для каждого предположения в соответствие ставиться *anchor* — прямоугольник на исходном изображении, с положением, связанным с текущей позицией скользящего окна. Например, 9 *anchor* прямоугольников можно задать с помощью размеров максимальной стороны 128, 256 или 512 и с помощью отношения сторон $1:1$, $1:2$ или $2:1$. Таким образом

С помощью anchor прямоугольников для каждой позиции скользящего окна предсказывается не один а несколько обрамляющих прямоугольников. Предположим, что максимальное число объектов, которые могут быть обнаружены — k. Тогда регрессионный слой задаёт координаты k обрамляющих прямоугольников и имеет размер $4k$, а выход слоя классификации имеет размер $2k$ — вероянтность того, что прямоугольник содержит объект и того, что не содержит.

** ИЗОБРАЖЕНИЕ **

3.3 Fast R-CNN детектор

На вход эта сеть получает список предположительных обрамляющих прямоугольников и матрицу признаков изображения из *глубокой свёрточной сети*. Далее для каждого прямоугольника из матрицы признаков извлекается фиксированного размера список признаков с помощью *region of interest pooling* слоя (RoI). Каждый список проходит через последовательность полносвязных слоёв. Затем результат вычисления проходит независимо через два полносвязных слоя: первый предсказывает оценочные *softmax* вероятности принадлежности одному из M классов и классу "фон", второй получает по четыре действительных числа для каждого из M классов.

RoI слой использует max pooling для преобразования признаков внутри предполагаемого прямоугольника в небольшой прямоугольник фиксированного размера $H_{RoI} \times W_{RoI}$. Размер этого прямоугольника — гиперпараметр данной сети. Каждый обрамляющий прямоугольник задаётся четвёркой чисел (r, c, h, w), которая задаёт верхний левый угол (r, c) и его высоту и ширину (h, w). Это окно делится на сетку, состоящую из ячеек размером $h/H_{RoI} \times w/W_{RoI}$. Тогда операция max pooling применяется для пропорционального прямоугольника признаков.

4 Процедура обучения

В данной архитектуре обучаются только два модуля: полностью свёрточная нейросеть и Fast R-CNN детекстор. Так как процедура обучения использует метод обратного распространения ошибки, то следует описать подсчёт функции ошибки для обучающихся модулей.

4.1 Ошибка полностью сверточной нейросети

Сперва присвоим бинарную метку каждому anchor — является ли объектом или нет. Положительную метку присвоим в двух случаях: anchor прямоугольник имеет наибольший IoU с истинным прямоугольником, или anchor прямоугольник имеет $IoU > 0.7$ с каким-либо истинным прямоугольником. Отрицательную метку присвоим в том случае, если anchor прямоугольник имеет $IoU < 0.3$ с каждым истинным прямоугольником. Anchor прямоугольники, которые не относятся ни к положительному, ни к отрицательному классы, не рассматриваются.

Обозначение Определение

Литература

[1] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010.

[2] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. Pvanet: Deep but lightweight neural networks for real-time object detection. 2016.

[5] Vidush Mukund Rafi Ayub, Cedric Orban. Art appraisal using convolutional neural networks. 2017.

[6] Ross Girshick Jian Sun Shaoqing Ren, Kaiming He. Faster r-cnn: Towards real-time object detection with region proposal networks. 2015.