

Урок 5

Практические задачи компьютерного зрения

5.1. Детекция объектов

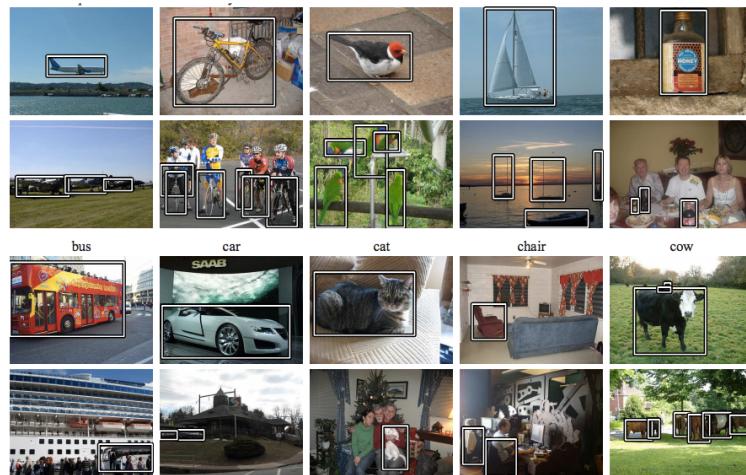


Рис. 5.1: Детекция объектов

Далее будет разобрано несколько практических задач. Первая из них — это детекция объектов. До этого речь шла о задаче классификации, когда по целой картинке требовалось определить, к какому классу она принадлежит. Задача детекции объектов подразумевает, что необходимо не только назвать класс изображения, но и найти на нём место, где расположен объект.

5.1.1. Скользящее окно

Один из традиционных подходов — это скользящее окно. Он уже обсуждался в контексте задачи фильтрации изображений и в данном случае не сильно отличается. Окно перемещается по картинке, к каждому такому окну можно применить классификационную нейронную сеть, обученную на тех классах, объекты которых требуется детектировать. В случае, если сеть обнаружила присутствие объекта в окне, оно помечается рамкой и классом.

R-CNN

У этого подхода есть недостаток. Чтобы пройти скользящим окном по изображению, ещё и на разных масштабах, нужно многократно применять классификационную нейронную сеть, а значит, это очень медленный процесс. Чтобы его ускорить, были придуманы разнообразные эвристики. Один из очевидных способов —

применять не тяжёлую классификационную сеть, а некий классификатор, отбрасывающий окна, в которых явно нет искомых объектов. Этот подход называется R-CNN (<https://github.com/rbgirshick/rcnn>), в нём с помощью метода Selective Search отбираются окна, а затем на них запускается классификационная нейронная сеть. R-CNN — это достаточно простой метод, он до сих пор используется как baseline при детекции объектов.

Faster R-CNN

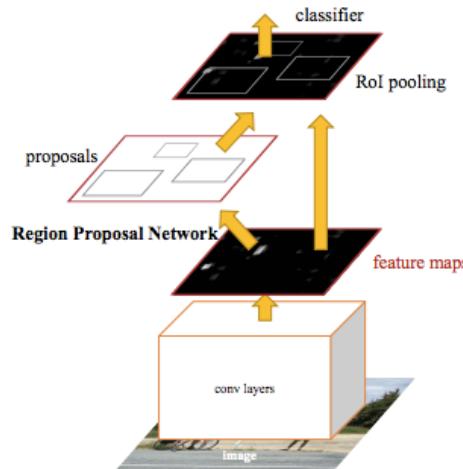


Рис. 5.2: Принцип работы метода Faster R-CNN

Несмотря на то что предыдущий метод достаточно хорошо себя показал, существует пространство для его улучшения. Та же группа авторов предложила метод R-CNN (рис. 5.2), где для предложения гипотез об окнах не используется внешний классификатор. В данном случае нейронная сеть выдвигает гипотезы об окнах и классифицирует. Вторая инновация заключается в том, что выдвигается не только гипотеза о наличии объекта в окне, но и уточняется его положение внутри. Идея оказалась успешной, и в данный момент этот метод является лучший по предсказанию объектов.

YOLO

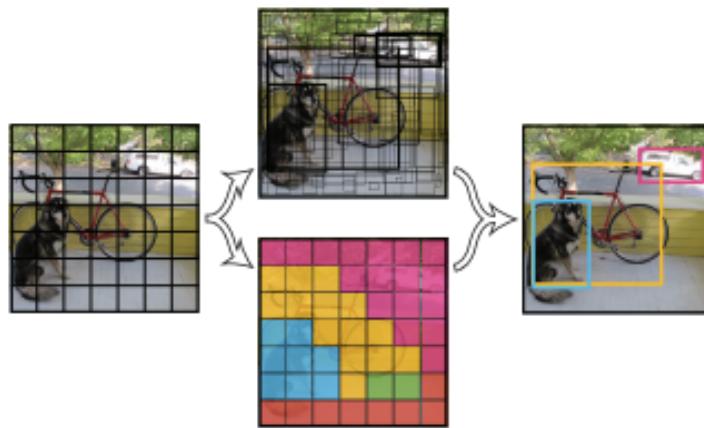


Рис. 5.3: Принцип работы метода YOLO

Аналогичная идея использования одной нейронной сети для предсказания класса объекта и положения ограничивающего прямоугольника использовалась при создании метода YOLO (рис. 5.3). Этот подход яв-

ляется ещё более простым. Картинка делится на несколько ячеек, в каждой из которых классификатор применяется отдельно. После этого строится предсказание о классе объекта и положении ограничивающего прямоугольника. Метод Yolo работает очень быстро, на рисунке 5.4 представлены результаты сравнения качества и скорости различных подходов. Faster R-CNN показывает лучший результат по качеству, скорость работы при этом около 7 кадров в секунду. В то же время, метод YOLO работает с качеством на 10% хуже, зато скорость работы достигает 155 кадров в секунду, что позволяет использовать его для детекции объектов в режиме реального времени.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
<hr/>			
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

Рис. 5.4: Результаты сравнения различных методов детекции объектов по скорости и качеству

5.1.2. Генеративные сети

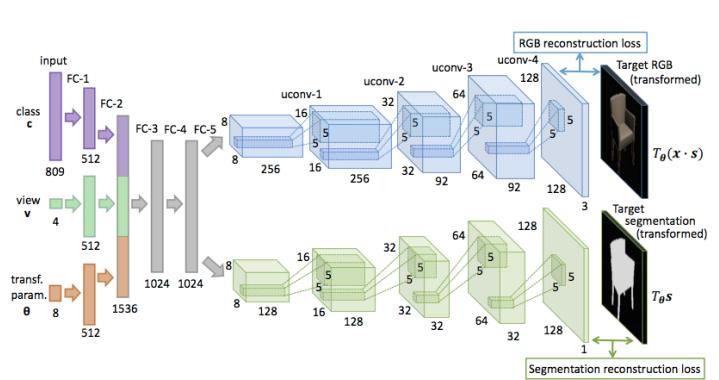


Рис. 5.5: Генеративная нейронная сеть

До этого обсуждались нейронные сети, которые по картинке выдают какие-то её параметры: либо класс объекта, либо координаты ограничивающего прямоугольника. Генеративные нейронные сети — это другой класс сетей, которые из векторов признаков или некоторых параметров умеют генерировать изображение. В одной работе демонстрировался интересный пример такой сети (рис. 5.5). На вход подаётся класс объекта, его положение в пространстве и то, с какой стороны на него смотрят, а нейронная сеть генерирует соответствующий объект. В качестве таких объектов в статье использовались стулья. На рисунке 5.6 в крайних столбцах находятся стулья из коллекций, а между ними — сгенерированные данной сетью. Видно, что промежуточные результаты похожи на стулья, и с помощью простого усреднения добиться такого результата не получилось бы. Это означает, что нейронная сеть в процессе обучения "осознала" форму стула, и использовала это для генерации новых.



Рис. 5.6: Крайние столбцы — реальные стулья, остальные — сгенерированные нейронной сетью

5.1.3. Семантическая сегментация

Далее будет под示范ировано, как можно использовать описанные нейронные сети для различных приложений. Изначально рассматривалась задача классификации, в которой необходимо что-то сказать о картинке целиком. В этой части была описана задача детекции объектов, в которой требуется найти на изображении объект, и что-то сказать о нём. Можно пойти дальше и для каждого пиксельного изображения на картинке определить, к какому классу оно принадлежит. На рис. 5.7 показан пример решения задачи сегментации, каждый класс на нём закодирован цветом. Это фотография улицы европейского города. На ней изображены объекты дорожной инфраструктуры: дорога, тротуар, велосипедисты, трамвай, люди, дорожные знаки. Задача состояла в том, чтобы найти все подобные классы на изображении.



Рис. 5.7: Пример решения задачи сегментации

Fully convolutional networks

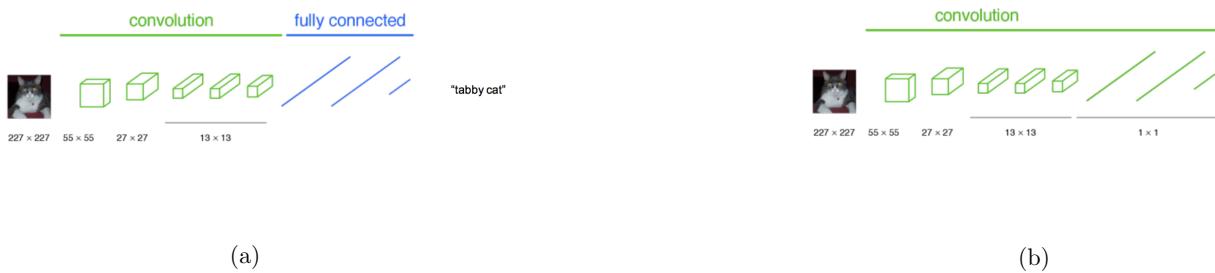


Рис. 5.8: (а) — свёрточная нейронная сеть, (б) — полностью свёрточная нейронная сеть

Существует несколько методов, которые решают описанную выше задачу. Один из них — это полностью свёрточные сети (fully convolutional networks). Ранее, когда речь шла о нейронных сетях, они состояли из нескольких свёрточных слоёв, а после них шло ещё несколько полносвязных слоёв (рис. 5.8а). На самом деле, нет никаких причин, по которым последние слои нельзя заменить на свёрточные (5.8б). Прелесть свёрточных слоёв заключается в том, что размер входной картинки неважен, а картинка на выходе будет пропорциональна исходной.



Рис. 5.9: Полностью свёрточная нейронная сеть

Итак, при использовании нейронной сети, состоящей только из свёрточных слоёв, результат на выходе пропорционален входному изображению. Например, если использовать архитектуру, похожую на AlexNet, но последние полносвязные слои заменить на свёрточные, то выход будет в 32 раза меньше, чем вход (рис. 5.9а). Если на вход нейронной сети подавать большую картинку, то выход будет содержать достаточно много информации. После этого к нему можно применить простые алгоритмы повышения размерности и получить изображение, разрешение которого равно разрешению входа (рис. 5.9б).

После того как появилась такая нейронная сеть, ничто не мешает в явном виде решать задачу сегментации: на вход подаётся изображение, на выходе получается изображение с таким же разрешением, но исходные пиксели заменены на номера классов, к которым они принадлежат. Если имеется обучающая выборка, то нейронную сеть можно обучать в явном виде. Проблема этого подхода заключается в том, что выход нейронной сети намного меньше по размеру, чем необходимое разрешение, а увеличение размерности в 32 раза может сильно портить качество.

В качестве решения авторы подхода предлагают использовать выходы не только последнего слоя, но и промежуточные: они меньше входа не в 32 раза, а в 16 и 8 раз. Агрегируя эти выходы в ответ, можно получить изображение хорошего качества. В некотором смысле такая нейронная сеть эмулирует применение сети на разных масштабах.

SEGNET — это другой метод решения задачи семантической сегментации. Авторы этого подхода использовали полносвязную архитектуру, поникающую размерность, и развернули её. Вопрос состоит в том, как повышать размерность. Для понижения используется операция тах-пулинг (из окрестности 2×2 выбирается максимальное значение, которое используется далее), каждое её применение понижает размерность в 2 раза. Для повышения размерности существует несколько методов. Один из них заключается в том, чтобы запоминать индексы максимумов, взятых при проведении тах-пулинга, хранить их до слоёв с обратным тах-пулингом, и на этом слое выбирать значения с теми же индексами. Такая конструкция хорошо работает и показывает результаты, превосходящие полностью свёрточные нейронные сети. Существуют эвристики

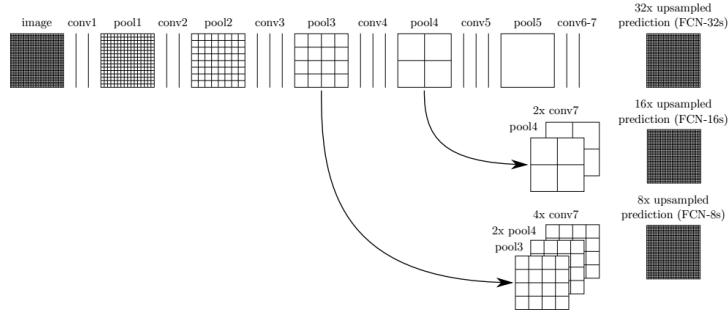


Рис. 5.10: Архитектура нейронной сети, решающей задачу семантической сегментации и использующей не только последний слой (https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf)

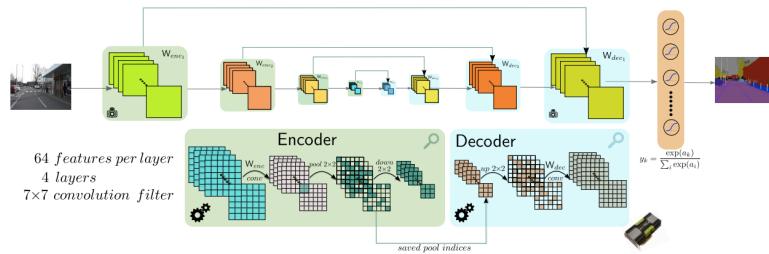


Рис. 5.11: Архитектура SEGNET

для улучшения этого подхода. В целом кажется, что такие нейронные сети представляют интерес не только применительно к задаче семантической сегментации.

5.2. Стилизация изображений

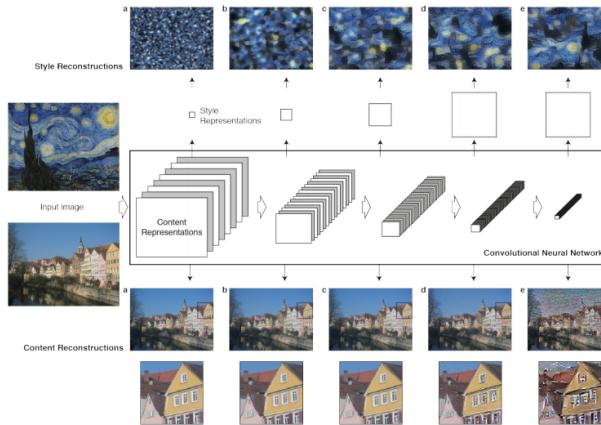


Рис. 5.12: Архитектура нейронной сети, выполняющей стилизацию изображений <http://arxiv.org/pdf/1508.06576.pdf>

Задача стилизации изображений заключается в том, что необходимо перенести стиль с картины художника на фотографию. Ученые из университета Тюбингена предложили алгоритм генерации таких картинок. Ключевую роль в нём играет предобученная нейронная сеть (использовалась архитектура VGG, как и во многих задачах компьютерного зрения). Фактически она оценивает, насколько сгенерированная картинка похожа на картину художника по стилю, а на фотографию — по содержанию. В данном случае веса нейронной сети оставались фиксированными, менялось изображение на входе.

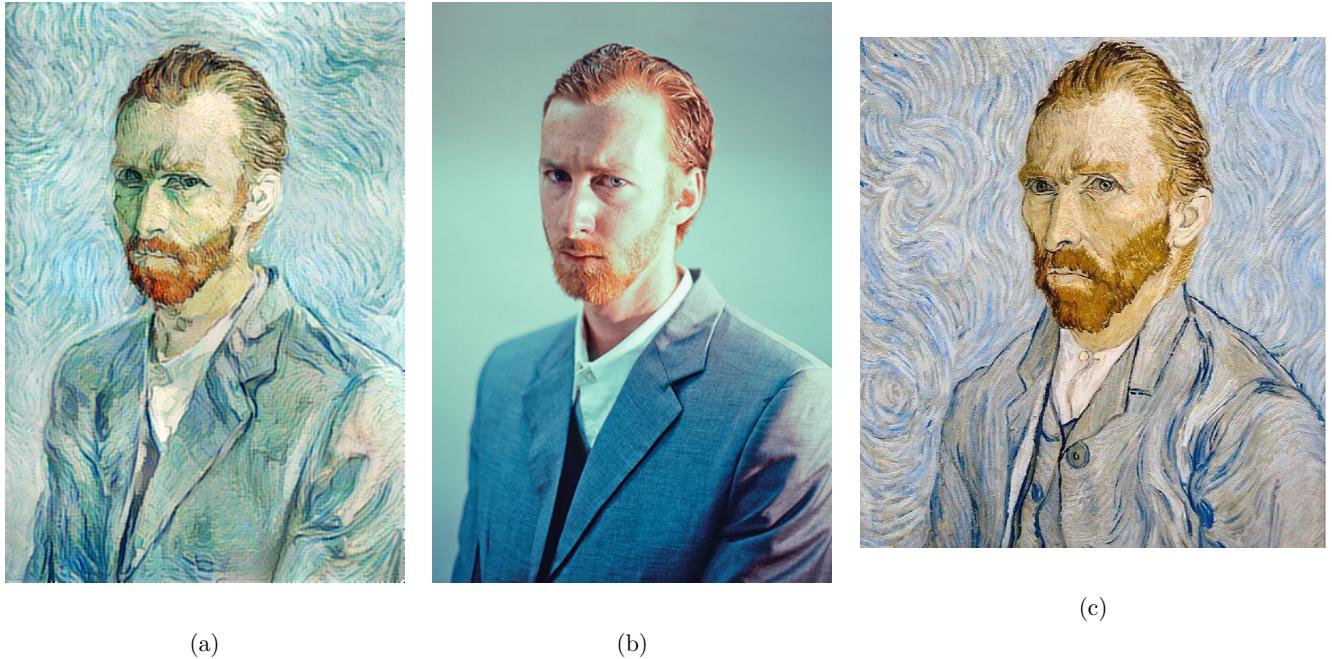


Рис. 5.13: (а) — фотография, стилизованная под картину Ван Гога, (б) — исходная фотография, (с) — картина Ван Гога

Проблема этого подхода заключалась в производительности. На генерацию одного изображения уходили десятки секунд, а иногда и минуты, в зависимости от разрешения исходной картинки.

Многим кажется, что картина на рисунке 5.13а написана Ван Гогом. На самом деле это фотография рыжего мужчины (рис. 5.13б), стилизованная под "Автопортрет" Ван Гога (рис. 5.13с).

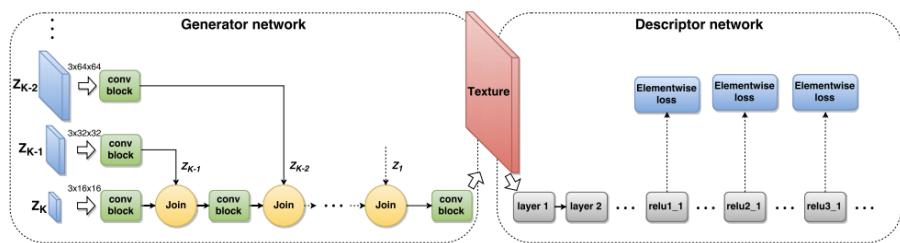


Рис. 5.14: Архитектура нейронной сети, ускоряющей стилизацию изображений (https://github.com/DmitryUlyanov/texture_nets)

Группа учёных из Сколтеха предложила способ ускорения данного алгоритма (рис. 5.14). Для генерации изображения они использовали полношёрточные нейронные сети, описанные в части про семантическую сегментацию. Для оценки, насколько изображение похоже по стилю на картину, а по смыслу на фотографию, используется предобученная сеть VGG. Данный метод работает быстрее, потому что в процессе обучения создаётся такая полношёрточная нейронная сеть, которая впоследствии применяется один раз для генерации картинки, в отличие от предыдущего метода, где для создания картинки использовались методы оптимизации.

Если для оценки похожести использовать только стиль, то описанный выше метод подходит для генерации текстур (рис. 5.16). Если же добавить оценку похожести содержания, то можно создавать стилизованные изображения (рис. 5.16).

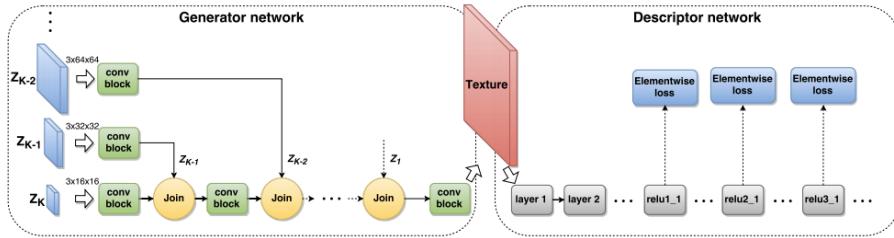


Рис. 5.15: Текстуры, сгенерированные нейронной сетью



Рис. 5.16: Стилизованные изображения, сгенерированные нейронной сетью

5.3. Распознавание китов

Задача распознавания китов была предложена в качестве соревнования на сайте kaggle. Требовалось реализовать распознавание лиц для китов.

В мире осталось очень мало гренландских китов (менее 500 особей). За ними внимательно следят: отслеживают состояние здоровья и пути миграции. Это делается с помощью фотографий с самолётов. Специалисты умеют отличать гренландских китов друг от друга по характерному рисунку из белых наростов на голове. Однако процедура узнавания китов является трудоёмкой, и это умеют делать лишь несколько специалистов. По этой причине создатели конкурса предложили участникам разработать автоматическую процедуру.

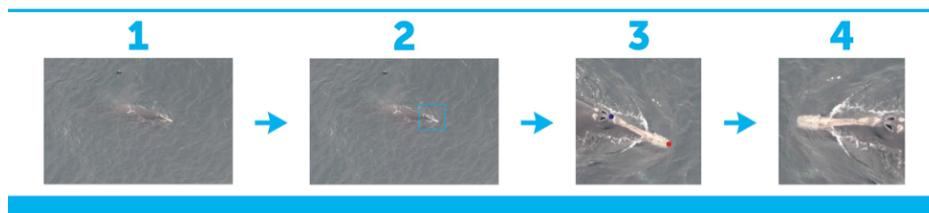


Рис. 5.17: Предобработка изображения в задаче распознавания китов

Данная задача очень сильно похожа на задачу распознавания лиц. Поэтому для её решения можно использовать ту же последовательность действий: детекция головы кита, выравнивание изображения и применение нейронных сетей. Примерно такой подход и был предложен победителями соревнования.

Процесс предобработки фотографии показан на рисунке 5.17. Сначала детектируется голова кита при помощи нейронной сети, которая натренирована отличать голову от фона. Затем другая нейронная сеть находит начало и конец головы. Зная эти две точки, можно развернуть изображение таким образом, чтобы на нём была голова, направленная в нужную сторону. Примеры того, как выглядят предобработанные фотографии с головами китов, показаны на рисунке 5.18. Видно, что по этим изображениям сравнивать китов друг с

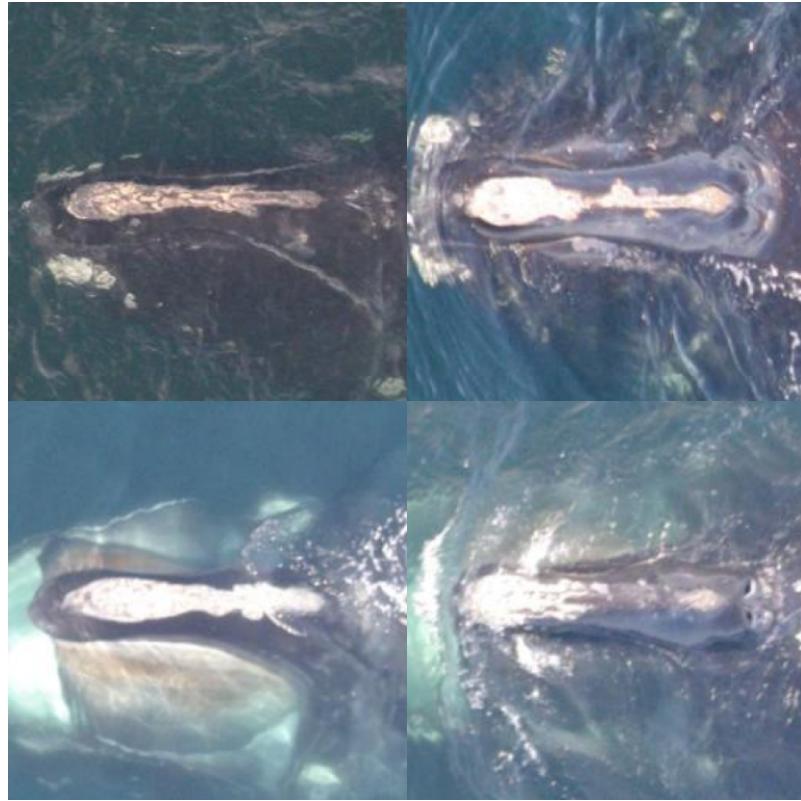


Рис. 5.18: Предобработанные фотографии с головами китов

другом становится гораздо проще.

Далее эти фотографии классифицировались с помощью нейронной сети, архитектура которой изображена на рисунке 5.19. Эта сеть похожа на VGG, но авторы использовали дополнительно несколько трюков, позволивших им улучшить результат.

Один из таких трюков — это увеличение скорости обучения. Обычно при обучении нейронных сетей используется метод уменьшения скорости обучения. Когда график ошибки выходит на плато, скорость обучения уменьшают, и иногда это приводит к тому, что обучение продолжается. Видимо, такой подход не принёс пользы, и авторы решили поступить иначе: увеличить скорость обучения. По графикам (рис. 5.20) видно, что иногда это позволяло продолжить обучение.

Часто соревнования на сайте kaggle сводятся к тому, чтобы попробовать множество различных методов, и найти среди них работающие. Интересно посмотреть, какие методы не сработали у авторов победного решения. Они пробовали вырезать голову кита без обучения, использовать какие-то другие методы обучения. В итоге для части голов пришлось вручную отметить точки начала и конца головы. Этих данных изначально не было.

Также у авторов не сработал метод Spatial transformer networks. Эта нейронная сеть пытается сама обучить преобразование, чтобы качество классификации улучшилось. Видимо, авторы хотели избежать процесса предобработки головы, но это не сработало.

Метод Deep residual networks также не сработал. Ранее упоминалось, что с его помощью хорошо решается задача классификации по базе ImageNet. Авторам это по какой-то причине пользу не принесло.

Наконец, авторы пробовали применять триплеты. В этом методе на вход нейронной сети подаются три картинки: две из них одного класса, третья — другого. Задача нейронной сети сделать так, чтобы изображения из одного класса были друг к другу ближе, чем к изображению из другого класса.

Описанные выше методы достаточно хорошо работают при решении задачи распознавания лиц. Интересно, что они не сработали в задаче распознавания голов китов.

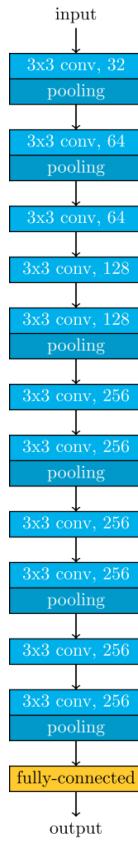


Рис. 5.19: Архитектура нейронной сети для классификации котов

5.4. Сбор больших коллекций изображений

При решении задач компьютерного зрения качество обучающей выборки зачастую оказывается важнее качества самих алгоритмов машинного обучения. Это утверждение можно применить и к анализу данных в целом. Далее речь пойдёт о том, как быстро и просто собрать коллекцию изображений.

5.4.1. Поиск по изображениям

В интернете хранится огромное количество различных изображений, многие из которых снабжены текстовым описанием. Это означает, что можно воспользоваться текстовым поиском по изображениям, чтобы найти картинки необходимого класса. Например, если ввести в поиск по изображениям запрос "машина", то выдача будет состоять из машин.

Группа из Оксфорда, для того чтобы успешно обучить свой классификатор распознавания лиц, собрала собственную базу лиц знаменитостей. Они создали большой список известных людей, затем по нему осуществлялся текстовый поиск по изображениям, и на каждый запрос было получено много фотографий знаменитостей.

Далеко не всегда поисковая система предоставляет качественную выдачу. Вместо того чтобы размечать каждую отдельную картинку, можно размечать запросы. Например, если среди знаменитостей есть однофамильцы, и по запросу, содержащему имя и фамилию, их фотографии выдаются вперемежку, то можно выбросить такой запрос. Кроме того, при поиске фотографий очень известных людей выдаётся много хороших изображений, а для чуть менее известных сначала идут очень хорошие фотографии, а затем начинаются ошибки. Такие запросы также достаточно легко отфильтровать: можно брать только первые несколько картинок, а остальные удалить.

Таким образом группе из Оксфорда удалось собрать базу, состоящую из двух миллионов лиц.

Аналогичным образом поступили исследователи при создании базы достопримечательностей. Идея заключалась в том, чтобы дообучить нейронную сеть на фотографиях достопримечательностей, и определить, насколько это улучшает качество поиска изображений по сравнению с предобученной сетью. Для создания

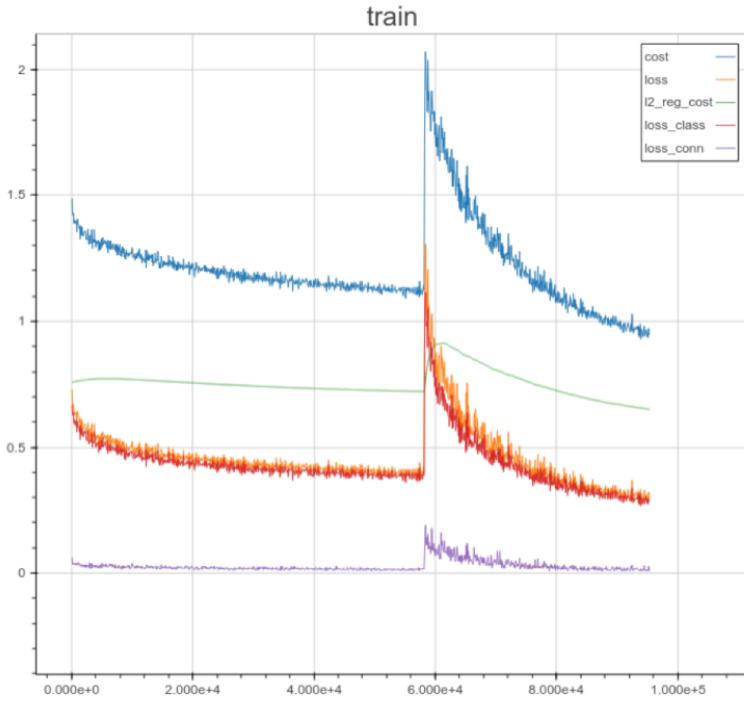


Рис. 5.20: Изменение ошибки нейронной сети при обучении

базы был собран список достопримечательностей, затем оттуда были убраны запросы, содержащие мусор. Например, существуют рестораны, названные в честь достопримечательностей, и иногда поисковые системы выдают изображения интерьеров этих ресторанов. Такие запросы можно выбросить, и достаточно быстро собрать необходимую базу.

5.4.2. Краудсорсинг

Другой способ сбора и разметки изображений — это система краудсорсинга. Существуют системы, позволяющие раздавать людям различные простые задания (например, Amazon Mechanical Turk и Яндекс Толока). С помощью них можно дёшево разметить большое количество изображений, и это удобно.

5.4.3. Итеративное построение базы

Можно использовать и комбинацию: часть изображений получить с помощью поиска по картинкам, а часть — с помощью краудсорсинга. Таким образом можно сформировать итеративную процедуру построения базы.

Для первых экспериментов нужна небольшая, шумная база. По ней обучается классификатор. Он выдаёт результаты для новых картинок, которых не было в базе. Их можно загрузить в систему краудсорсинга для разметки. Краудсорсеры находят ошибки, и размеченные данные доливаются в исходную базу. Процедуру можно повторять многократно, наращивая таким образом базу.

5.4.4. Человек помогает машине

В контексте взаимодействия человека и алгоритмов стоит упомянуть класс задач, в которых человек приходит на помощь алгоритму компьютерного зрения, чтобы исправить его ошибки.

Например, в задаче модерации алгоритм выдаёт в качестве результата степень уверенности. В случаях, когда алгоритм уверен, он сразу используется в системе, иначе данные подаются на ручную модерацию. В таких случаях компьютерное зрение используется для облегчения задачи модераторов, чтобы небольшое число людей могли размечать изображения.

Другой пример использования человеческого труда для улучшения качества системы — это мобильное приложение Camfind. Оно позволяет сфотографировать любой предмет и получить в ответ, что это такое.

Приложение работает в реальном времени, задержки составляют десятки секунд. Оно показывает высокое качество ответа на вопрос, что изображено на картинке. Если не знать, что приложение использует не только алгоритм, но и человеческий труд, это выглядит, как чудо. Создатели приложения долго работали над тем, чтобы люди могли быстро и качественно описывать изображение, в том числе, тщательно проработали интерфейс. С помощью приложения авторы смогли собрать большую базу картинок для своих нужд.

Подобные гибридные системы помогают улучшить качество исходного алгоритма, размечать базу, и снова улучшать качество автоматической части.

5.4.5. Синтетические изображения

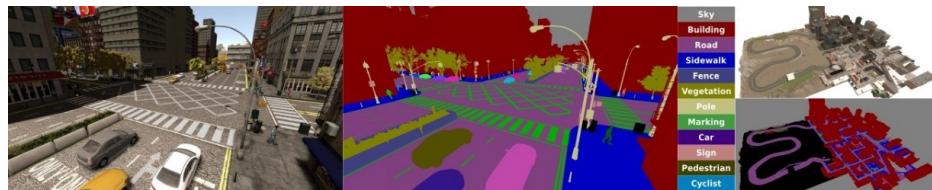


Рис. 5.21: Изображения из базы Synthia

Ещё один способ пополнения базы — это использование синтетических данных. Современная компьютерная графика позволяет генерировать реалистичные миры. Часто графические сцены выглядят реалистичнее, чем снятые в реальности. Возникает очевидна мысль использовать их для обучения классификаторов.

Примером использования такого подхода является база Synthia (рис. 5.21). Для её создания был взят искусственный город, и с помощью движка Unity были сгенерированы картинки с семантической разметкой. Эта база позволила существенно расширить существующие базы по семантической сегментации и заметно повысить качество обучения.

Использовать синтетические данные нужно аккуратно. Может сложиться так, что они отличаются от реальных, а нейронная сеть будет обращать внимание именно на эти отличия. Сгенерировать данные так, чтобы они не отличались от реальных, сложно, поэтому иногда синтетические данные никак не улучшали качество работы алгоритма. Их стоит использовать тогда, когда реальные данные действительно сложно разметить. Иначе может сложиться ситуация, когда создание системы, генерирующей синтетические данные, стоит дороже, чем ручная разметка данных.

5.4.6. Резюме

Часто наличие какой-либо базы являлось триггером для запуска исследований в той или иной области. Например, благодаря коллекции ImageNet началось современное развитие нейронных сетей. Также можно привести в качестве примера задачи распознавания лиц, детекции объектов и семантической сегментации. Во всех них начался прогресс с появления качественной коллекции.

Базы изображений могут являться конкурентным преимуществом. Корпорации делятся алгоритмами решения задач машинного обучения, выкладывают библиотеки в open source. С данными всё обстоит сложнее. Получить данные, принадлежащие какой-либо компании, практически невозможно.

Кроме того, увеличение размера и качества базы — это часто самый простой способ улучшить качество решения задачи. Вместо усложнения алгоритмов, использования ансамблей, усреднения результатов различных обучений можно просто увеличить выборку.