Московский государственный университет имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

«Визуализация нейросетей и генерация изображений»

Выполнил:

студент 3 курса 317 группы

Бикметов Данил Наильевич

Москва, 2021

Содержание

Аннотация

Настоящая работа является расширенным конспектом первой части лекции «Визуализация нейросетей и генерация изображений». Конспект основан на самой лекции и на материалах, предложенных в ней. В работе рассмотрены основные идеи визуализации работы нейросети: Deconvnet, CAM, Guided Backpropogation, Grad-CAM и Guided Grad-CAM, Occlusion Sensitivity, Saliency Maps и FullGrad. Разобран способ построения интерпретируемой свёрточной нейросети. Рассмотрены стандартные средства поиска похожих изображений.

1 Введение

Визуализация нейросети является важным этапом её отладки. Она позволяет следить за преобразованиями признаковых пространств, изобретать новые подходы, видеть проблемы в данных и моделях. Благодаря визуализации нейросети можно узнать, как именно происходит классификация изображений.

Визуализировать можно фильтры и внутренние активации, изображая их в виде картинок. Есть возможность наблюдать как за распределением активаций на отдельных объектах, так и за входами, максимизирующими тот или иной ответ. Поскольку сеть обучается методом обратного распространения ошибки, то существует возможность смотреть на производные по входу.

Проще всего дела обстоят со свёртками первого слоя. Поскольку они являются тензорами глубины 3, то есть возможность наблюдать за ними как за изображениями. Такая визуализация была продемонстрирована авторами сети AlexNet в [1]. В этой сети было использовано 96 свёрток на первом слое, каждая свёртка имела размер 11x11x3. Ниже представлены изображения этих свёрток:

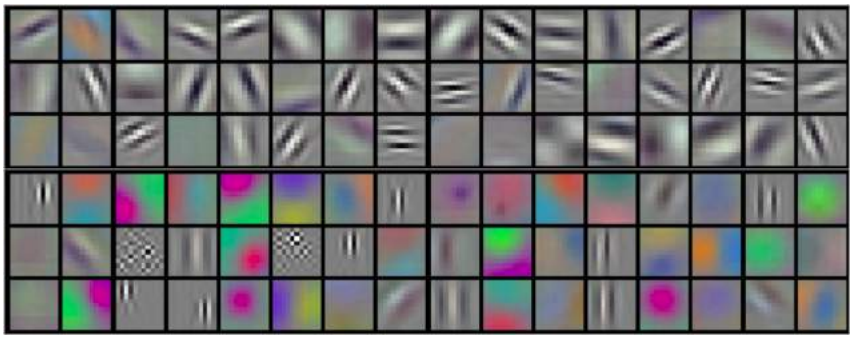

Рис. 1: Изображения 96 свёрток на первом слое AlexNet. Видны чёткие паттерны, переливы цветов и прочие простейшие геометрические узоры, на которые реагирует нейросеть. Изображение взято из [1].

Однако такая интерпретация невозможна для свёрток на последующих слоях. В самом деле, свёртки глубоких слоёв имеют большое число каналов, то есть они непредставимы в пространстве RGB. Тем не менее, существует множество различных подходов, позволяющих наблюдать за глубокими свёртками. В настоящей работе рассмотрены методы Deconvnet, CAM, Guided Backpropogation, Grad-CAM и Guided Grad-CAM. Внимание уделено интерпретируемой CNN, которую можно обратить без потерь информации. Рассмотрены методы анализа активации нейронов, а также такие методы, как Occlusion Sensitivity и Saliency Map.

2 Deconvnet

Одним из способов наблюдения за свёртками глубоких слоёв является deconvnet. Этот метод был впервые предложен Мэтью Зайлером в [2]. Идея метода заключается в следующем. Исходная свёрточная сеть является преобразованием изображения (тензора) в тензор. Значит, есть возможность как-то пропустить сигнал обратно и понять, на какие паттерны заточены свёртки. Deconvnet - это своего рода обратная сеть, получающая на вход тензор и возвращающая изображение. Её можно рассматривать как оригинальную CNN, которая использует те же компоненты (filtering, pooling), но в обратном порядке.

Общая схема метода следующая. Выбирается нейрон, затем находится top-k изображений, которые максимизируют значение этого нейрона. Сначала эти изображения совершают прямой проход по нейросети, после чего все нейроны, кроме рассматриваемого, зануляются и совершается обратный проход. Последовательно совершается

unpool, rectify и filter для восстановления активности в предыдущем слое, которая привела к выбранной активации. Разберём каждый из этапов подробнее.

Unpooling: в CNN операция Max-Pooling, вообще говоря, не обратима, однако мы можем получить примерное представление обратного тензора, запомнив расположения максимумов в массиве «switches». При обратном проходе операция unpooling использует информацию со «switches» для восстановления предыдущего слоя, подставляя текущие элементы в соответствующие позиции. Ниже приведён рисунок с применением операции unpooling.

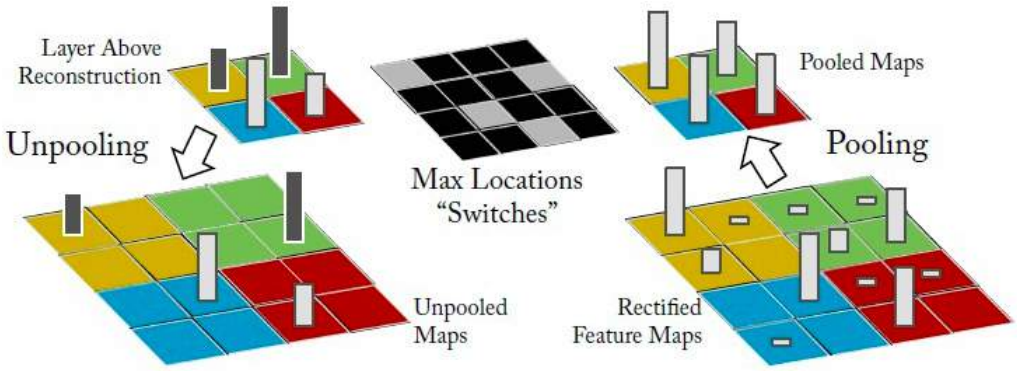

Рис. 2: Unpooling в Deconvnet. Изображение взято из [2].

Rectification: в CNN используются нелинейности ReLU, которые гарантируют, что выходные признаки всегда будут положительны. Чтобы получить достоверную реконструкцию на каждом слое, во время обратного прохода в Deconvnet восстановленный сигнал также проводится через ReLU.

Filtering: в CNN используются обученные фильтры для свёртки тензоров из предыдущего слоя. Чтобы инвертировать их, Deconvnet производит транспонирование этих фильтров и применяет их к восстановленным тензорам, полученным после предыдущего шага. На практике это означает переворачивание каждого фильтра по вертикали и горизонтали.

Ниже представлена архитектура свёрточной нейросети, используемой в экспериментах.

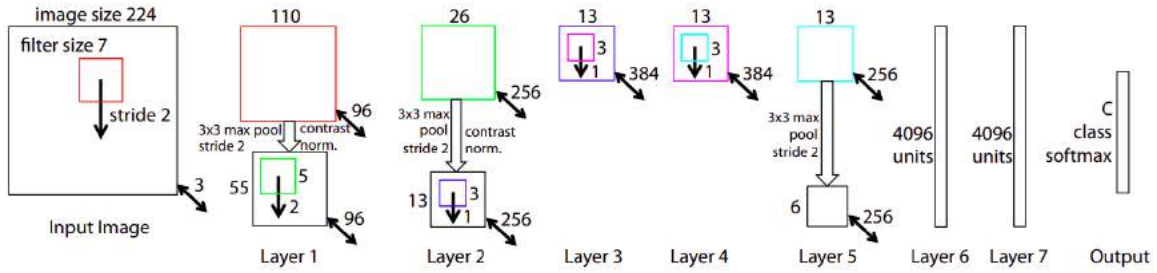

Рис. 3: Архитектура тестируемой нейросети (AlexNet), предложенной в [1]. Рисунок взят из [2].

На следующих рисунках представлены визуализации свёрток 2-5 слоёв, полученных с помощью Deconvnet. Справа от свёрток представлены восстановленные шаблоны, которые вызвали наибольшую активацию в выбранном нейроне рассматриваемого слоя. Изображения были взяты из датасета ImageNet 2012.

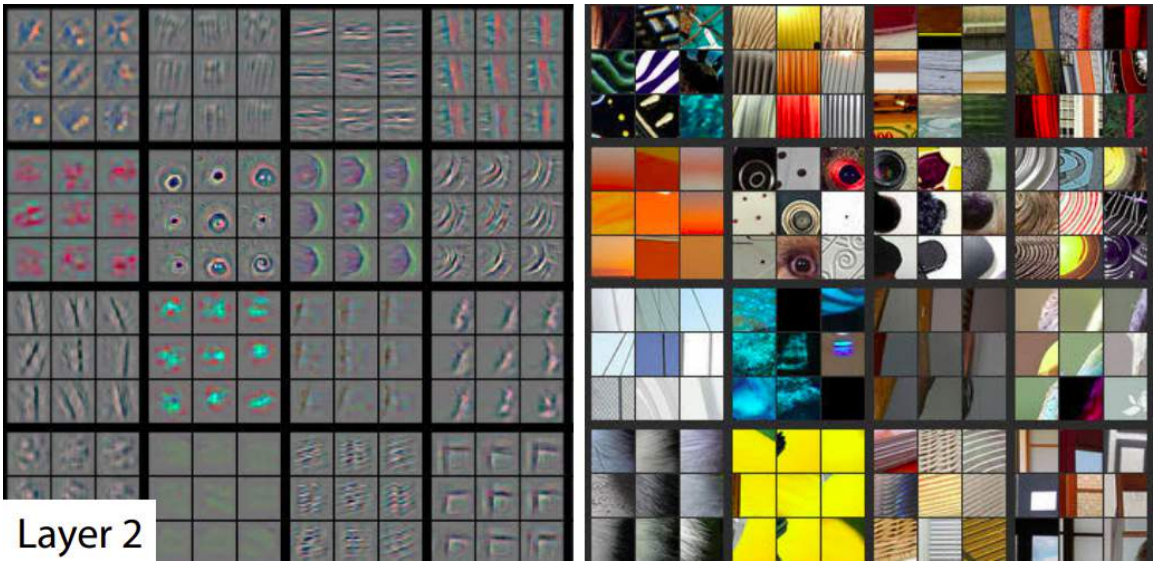

Рис. 4: Выход deconvnet со второго слоя, взято из [2].

На свёртках второго слоя отчётливо видны паттерны, на которые реагирует нейросеть. Можно понять, что признаковое пространство в этом слое состоит из различных геометрических узоров низкой абстракции, то есть нейроны активируются на углы, наклоны и окружности.

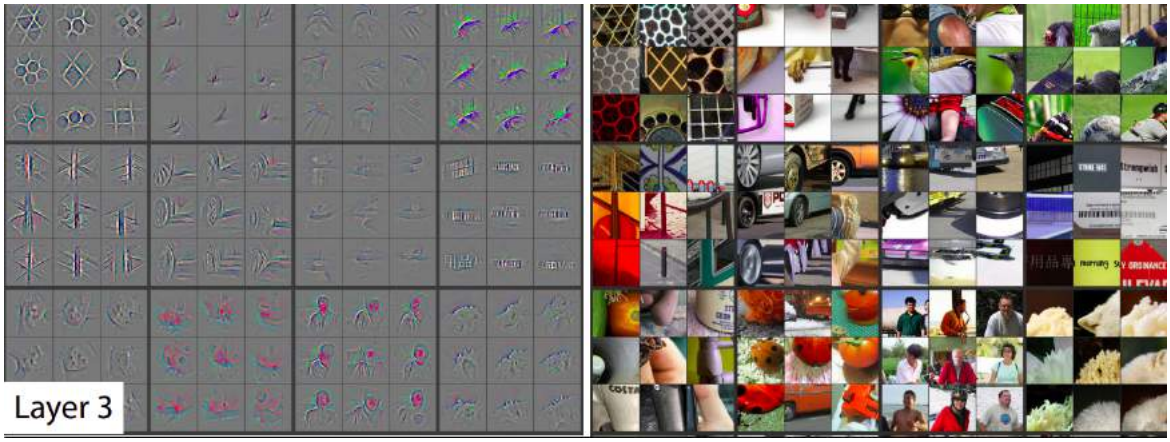

Рис. 5: Выход deconvnet с третьего слоя, взято из [2].

На третьем слое паттерны начинают создавать устойчивые узоры. Нейроны начинают активироваться на формы, контуры и регулярные структуры, причём активация происходит всей структурой целиком.

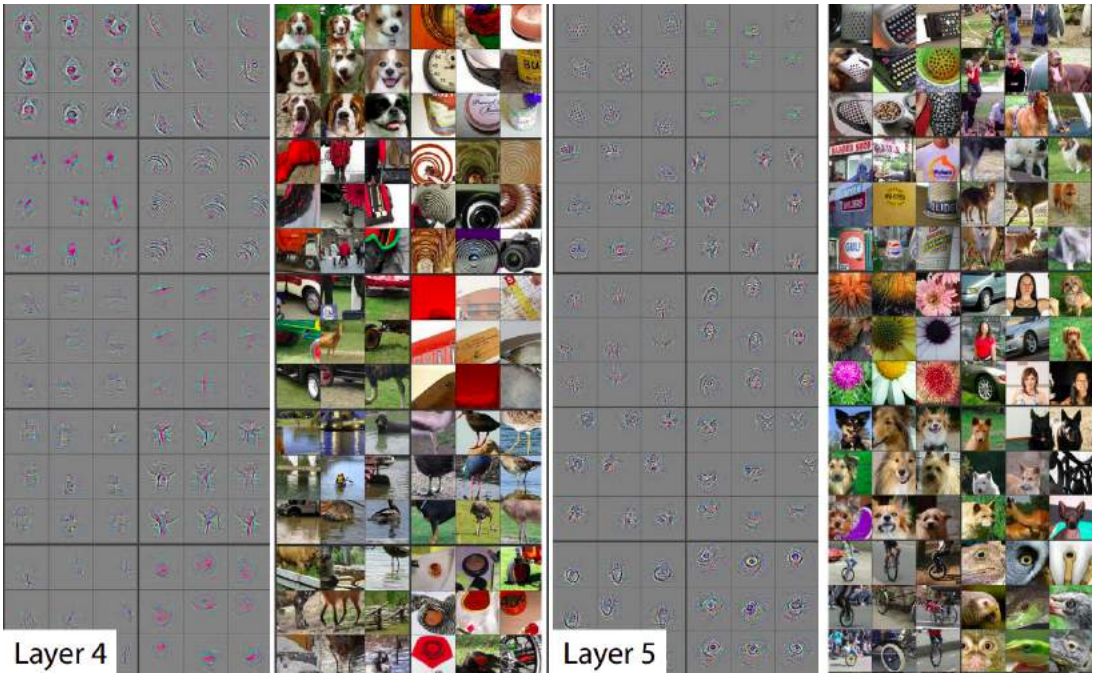

Рис. 6: Выход deconvnet с 4-5 слоёв, взято из [2].

На последних слоях повышается уровень абстракции. Паттерны, на которых активируется нейрон, имеют чёткую интерпретацию: можно разглядеть мордочку собаки, силуэты животных и различные уникальные узоры, отличающие изображение.

Общую динамику изменения признаковых пространств иллюстрирует следующий рисунок.

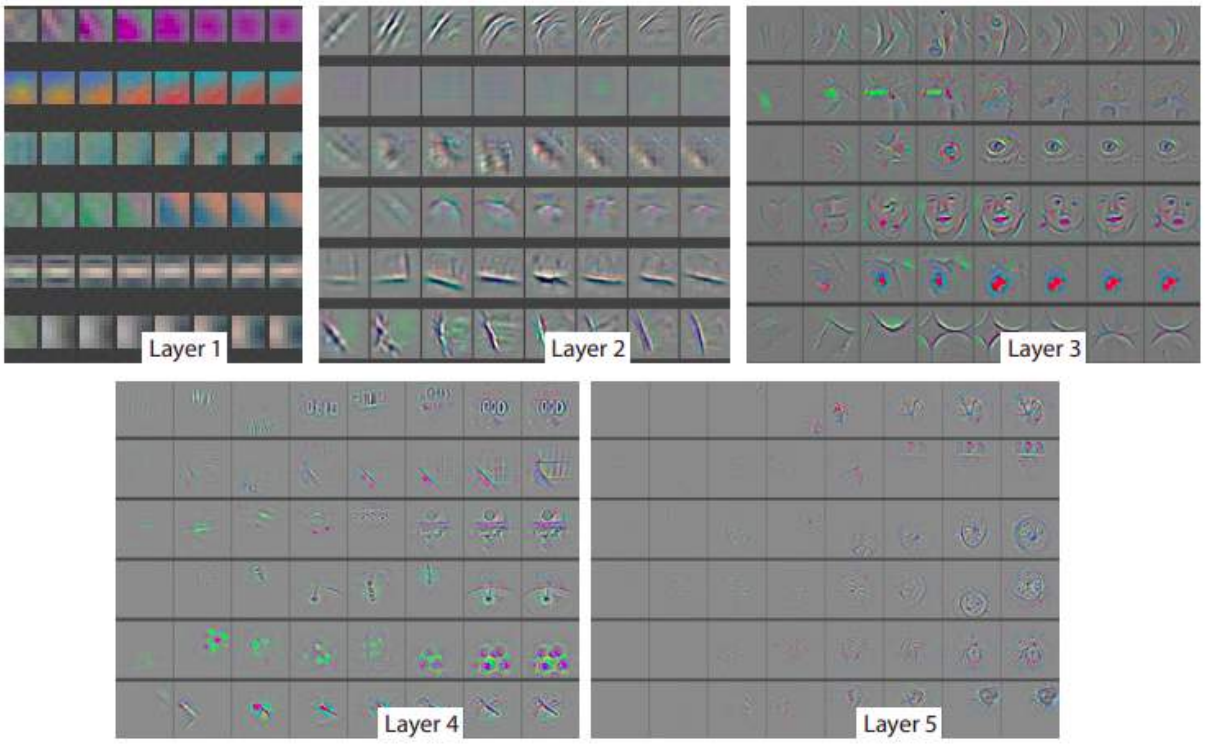

Рис. 7: Визуализация изменения признаковых пространств свёрточной сети. Объекты каждого слоя отображаются в отдельном блоке. Внутри каждого блока собрано случайное подмножество объектов в эпохах 1, 2, 5, 10, 20, 30, 40, 64. Изображение взято из [2].

С увеличением эпохи повышается чёткость визуализации. Особенно хорошо это видно на глубоких слоях. Это объясняется тем, что глубокому слою требуется больше времени на настройку весов. Тем не менее, Deconvnet недостаточно эффективен на последних слоях, поскольку выдаёт размытые, слабо интерпретируемые и разреженные контуры.

3 CAM

3.1 Сети с Global Average Pooling

Метод Class Activation Maps (карт активации классов) был предложен в [3] группой исследователей из MIT. Метод применим только к тем CNN, которые имеют слой Global Average Pooling (GAP) незадолго до выходного слоя. Этот слой усредняет тензор по каналам, переводя его в вектор, котрый затем поступает на вход

полносвязному слою. На самом деле такая архитектура является достаточно распространённой в свёрточных нейросетях, поэтому проблем с выбор CNN не возникает.

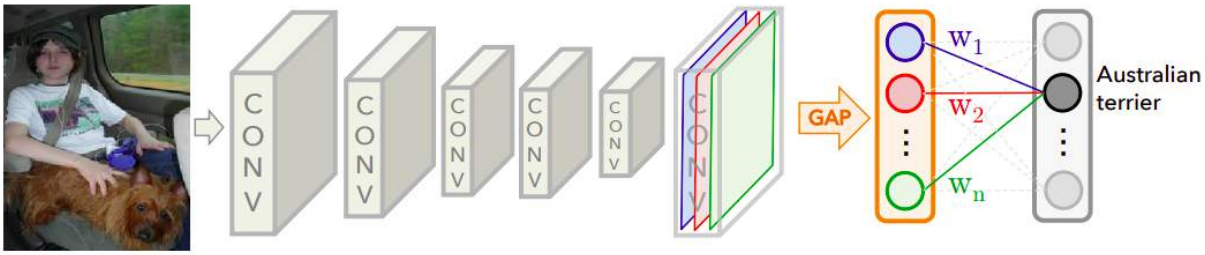

Рис. 8: Изображение сети с Global Average Pooling, взято из [3].

Процедура создания карт активации классов состоит в следующем. Мы можем посмотреть, с какими весами взялись элементы вектора, полученного после Global Average Pooling, в полносвязном слое. Каждый элемент есть усреднение определённого канала, следовательно, мы можем взять изображение, прошедшее через сеть, и посмотреть, как оно выглядело на этом канале. Затем мы можем сложить все каналы с соответствующими весами и получить карту активации классов. Таким образом, мы получим тепловую карту, показывающую, на что смотрела нейросеть во время классификации изображения.

Ниже приведён рисунок, иллюстрирующий работу метода CAM.

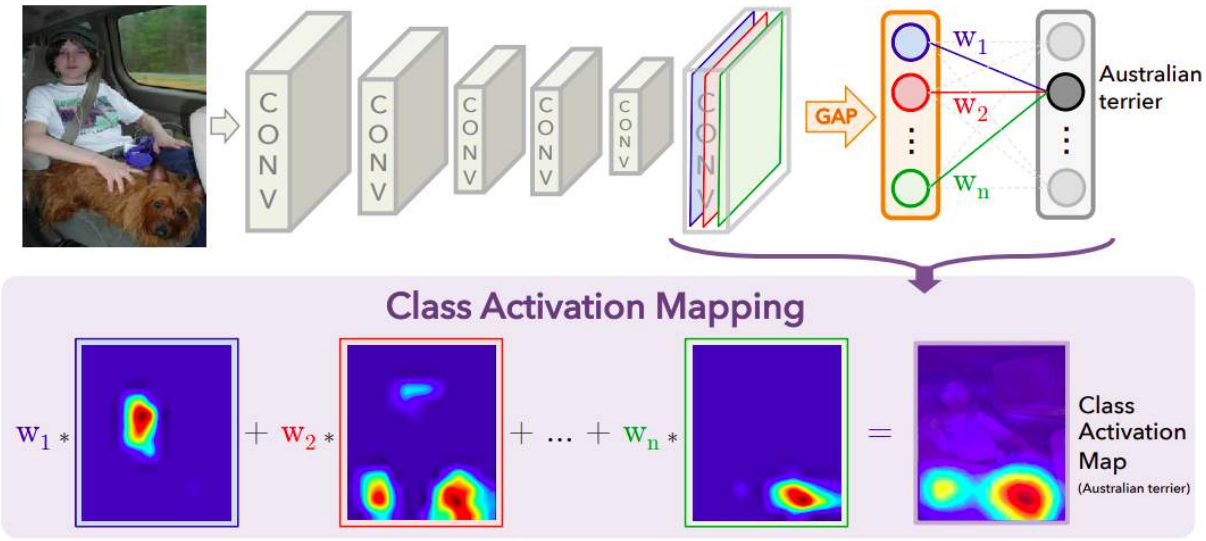

Рис. 9: Иллюстрация метода CAM, взято из [3].

Ниже изображён пример работы алгоритма.

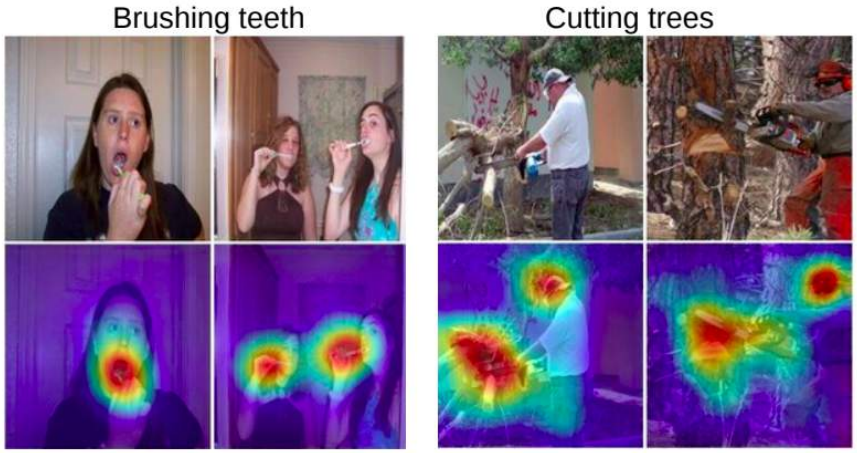

Рис. 10: Пример работы CAM, взято из [3].

Интерес вызывает тот факт, что мы можем построить карту активации не только для правильного класса, но и для произвольного. Таким образом, есть возможность искать на изображении различные объекты. Метод CAM фактически создаёт общее локазлизуемое представление об объектах на изображении. В [3] авторы деклаируют, что им удалось достичь хорошего качества на датасете ILSVRC 2014 при детектировании объектов. Процент ошибки составил 37.1%, в то время как спеиально обученная под детектирование объектов нейросеть имеет ошибку 34.2%. Таким образом, с помощью метода CAM можно заставить сеть заниматься детектированием объектов на изображении, несмотря на то, что она не была обучена этому.

3.2 Сети с Global Max Pooling

Вообще говоря, рассматриваемый подход был впервые предложен в [4] группой французских и американских исследователей. Авторы изначально ставили перед собой цель детектировать объекты с помощью нейросети, которая обучалась на классификации этих объектов. Отличие от рассмотреного CAM состояло в том, что вместо слоя Global Average Pooling использовался слой Global Max Pooling. Ниже приведена иллюстрация такой сети.

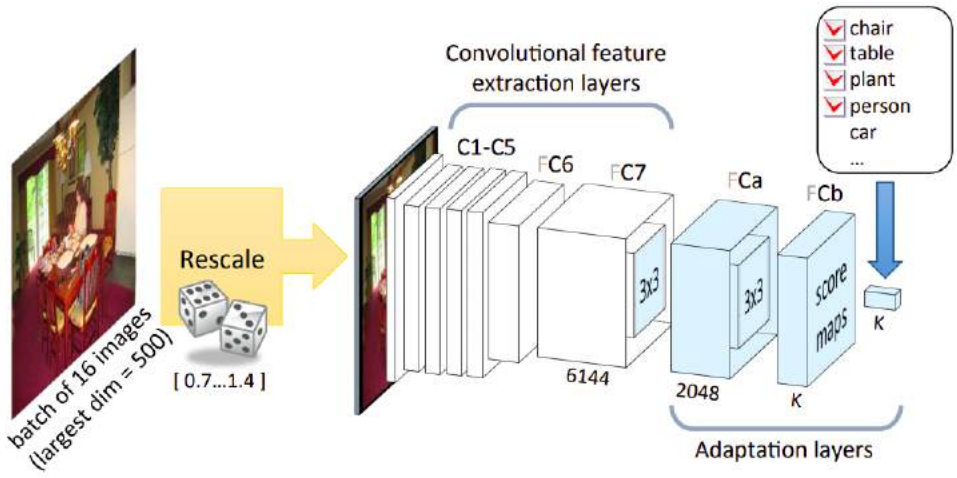

Рис. 11: Пример сети с Global Max Pooling, взято из [4].

Существенное отличие от рис. 8 состоит в последем слое, в остальном отличий нет. Ниже приведён пример работы метода с Global Max Pooling.

Рис. 12: Пример работы метода с Global Max Pooling, взято из [4].

На вход подано изображение самолёта, сеть детектирует область, на которой изображён самолёт, и область, на которой может быть изображён автомобиль.

Однако в [3] было показано, что такой метод уступает рассмотренному выше CAM с Global Average Pooling. Основная интуиция преимущества подхода со слоем Global Average Pooling заключается в том, что функция потерь для среднего позволяет лучше учитывать те объекты, которые не были классифицированы, но которые могли присутствовать на изображении или составляли часть классифицируемого объекта. Это заметно повышает качество детектирования, что экспериментально подтверждается в [3].

4 Guided Backpropogation

Guided Backpropogation - это дальнейшее развитие идей Deconvnet и CAM для определённого класса нейросетей. Идея, предложенная в [5], заключается в создании полностью свёрточной нейросети. Во-первых, авторы предлагают заменить pooling слои на свёрточные слои с параметром stride > 1. В работе приводится эвристика того, что такая замена допустима. Во-вторых, авторы используют небольшое количество слоёв (меньше 5), что значительно уменьшает количество параметров и служит своеобразной регуляризацией. Таким образом, вся архитектура нейросети сводится к тому, что она состоит только из свёрточных слоёв с ReLU, усреднениями и softmax слоем на выходе.

Итак, Deconvnet инвертировал поток данных в CNN, переходя от активации нейронов в рассматриваемом слое к изображению, после чего получал искусственное изображение, которое наиболее сильно активирует конкретный нейрон. Для выполнения обратного прохода через слои Max-Pooling, которые, вообще говоря, не являются обратимыми, требовалось выполнить сначала прямой проход для вычисления так называемых «switches» - положений максимумов в каждой области пулинга. Следовательно, обратный проход существенно зависел от входного изображения. Использование Guided Backpropogation в сетях без слоёв Max-Pooling не требует запоминания «swithes». Таким образом, имеется возможность получить более точное представление о том, чему учатся глубокие слои в нейросети.

Основное нововведение, предложенное в Guided Backpropogation, заключается в обратном проходе сигнала через ReLU. Напомним, что Deconvnet применял к обратному сигналу ReLU, то есть пропускал обратно только положительные значения. Существует другой подход (backward pass) - пропускать обратно только те значения, которые были пропущены через ReLU во время прямого прохода, и занулять остальные. Guided Backpropogation комбинирует эти методы, пропуская обратно только положительные значения, которые в свою очередь были положительными во время прямого прохода.

Формально это выглядит следующим образом. Пусть имеется активация $f_i^{l+1} = ReLU(f_i^l) = max(f_i^l, 0)$. Тогда для каждого метода обратный проход через ReLU будет выглядеть так:

$$Deconvnet : R_i^l = (f_i^l > 0) \cdot \frac{\partial f^{out}}{\partial f_i^{l+1}}$$

$$Backward\ pass : R_i^l = (R_i^{l+1} > 0) \cdot \frac{\partial f^{out}}{\partial f_i^{l+1}}$$

$$Guided\ Backpropogation : R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot \frac{\partial f^{out}}{\partial f_i^{l+1}}$$

Ниже приведена иллюстрация обратного прохода через ReLU всех трёх методов.

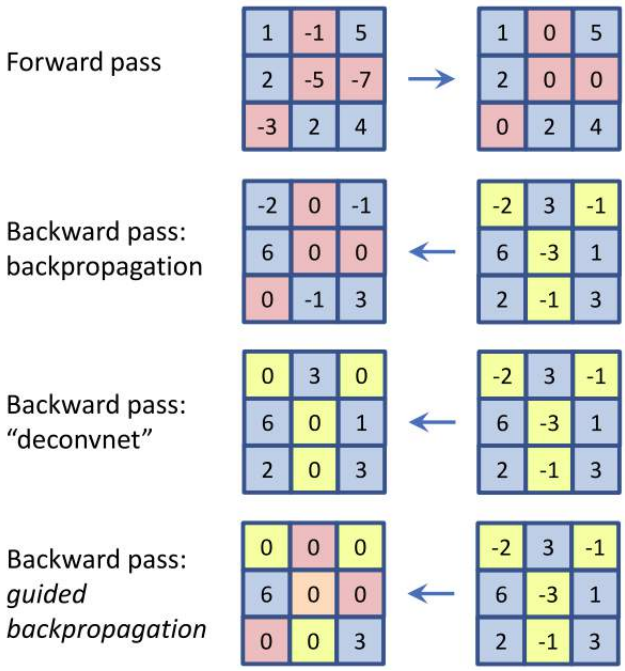

Рис. 13: Обратный проход через ReLU в методах Backpropogation, Deconvnet, Guided Backpropogation. Изображение взято из [5].

В [5] утверждается, что визуализация глубоких слоёв с помощью Guided Backpropogation даёт более чёткое изображение. В том, что Deconvnet недостаточно эффективен на глубоких слоях, мы убедились ранее. Это можно объяснить интуитивно, поскольку первые слои изучают объекты в общем смысле, стараясь реконструировать единый паттерн, в то время как глубокие слои изучают гораздо более инвариантные представления. Однако благодаря замене всех pooling слоёв на свёрточные слои нейросеть становится более обратимой, что положительно сказывается

на интерпретируемости обратного сигнала. Более того, модифицированный обратный проход через ReLU зануляет большее число значений, что в свою очередь позволяет бороться с шумами и размытиями, присущими обратному сигналу в Deconvnet.

Ниже на рисунке происходит сравнение результатов работы Deconvnet и Guided Backpropogation.

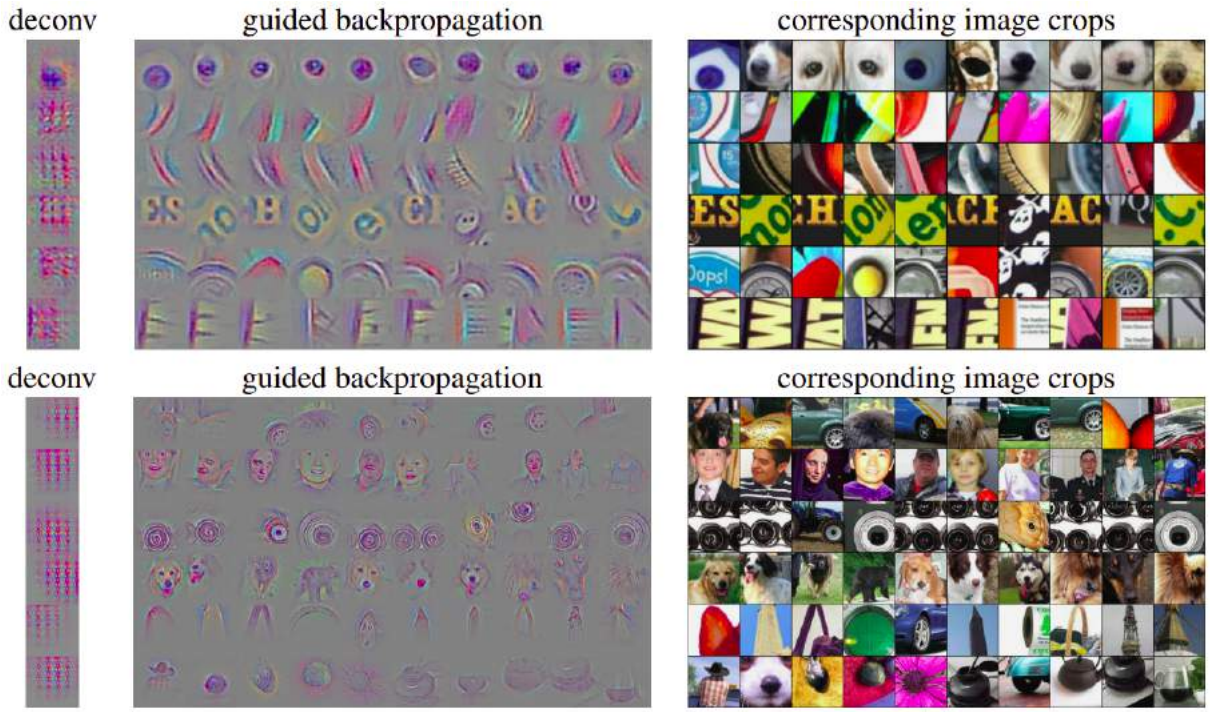

Рис. 14: Сравнение методов Deconvnet и Guided Backpropogation. Изображение взято из [5].

Картинки, которые получаются в результате Guided Backpropogation, содержат чёткие контуры, имеющиеся во входном изображении. Если изображение содержит округлый предмет, то видно, как определённый нейрон улавливает его. Если изображение содержит текст, то другой нейрон улавливает фрагменты букв.

На практике Guided Backpropogation выдаёт гораздо более чёткие картинки, чем Deconvnet. Напомним, что Deconvnet проводит сигнал через Max-Pooling обратно, запоминая позиции максимумов в массиве «switches». Авторы метода Guided Backpropogation в [5] сравнили работу двух методов, причём были использованы сети как с pooling слоями, так и без них. Результат представлен на рисунке ниже.

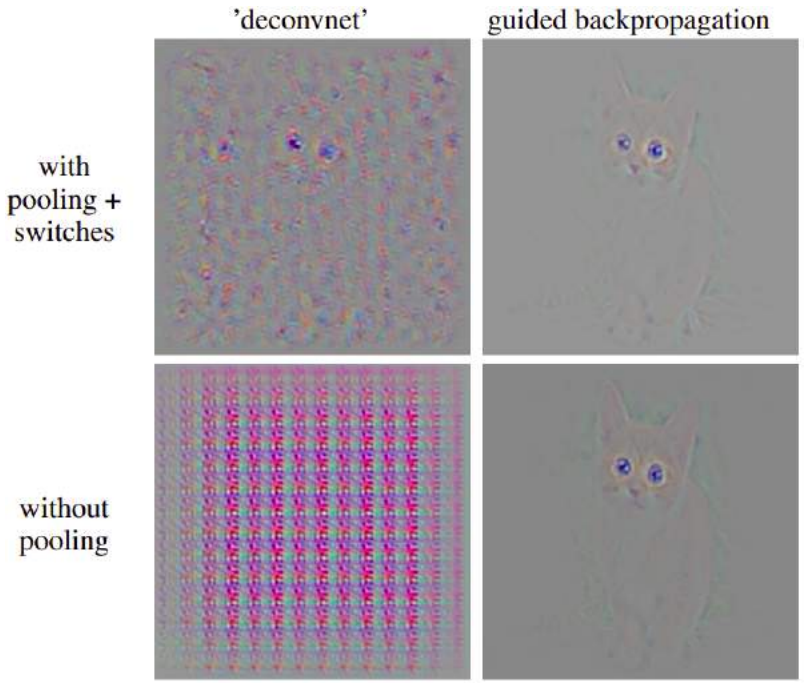

Рис. 15: Сравнение Deconvnet и Guided Backpropogation в сетях с pooling слоями и без них. Изображение взято из [5].

Котёнок лучше всего вырисовывается при использовании GuidedBackpropogation в сетях без pooling слоёв. Но даже если в сети есть pooling слои, результат метода GuidedBackpropogation всё равно превосходит Deconvnet, в то время как при отсутствии pooling слоёв Deconvnet выдаёт практически неинтерпретируемую картинку.

5 Интерпретируемые CNN

Следующий способ понять, на что реагирует CNN при классификации, заключается в создании интерспретируемой сети. Метод модификации традиционных CNN в интерпретируемые CNN был предложен в [6] в 2018 году. Идея в том, что в предложенных интерпретируемых CNN каждый канал в глубоких слоях ответственен за определённую часть объекта, причём сеть сама решает, какая именно часть объекта будет ему соответствовать. Авторы подчёркивают тот факт, что их нейросеть не требует дополнительной информации и способна обучаться на тех же данных, что и обычная CNN. Ниже приведён рисунок, показывающий разницу между каналами традиционной интерпретируемой свёрточной нейросети.

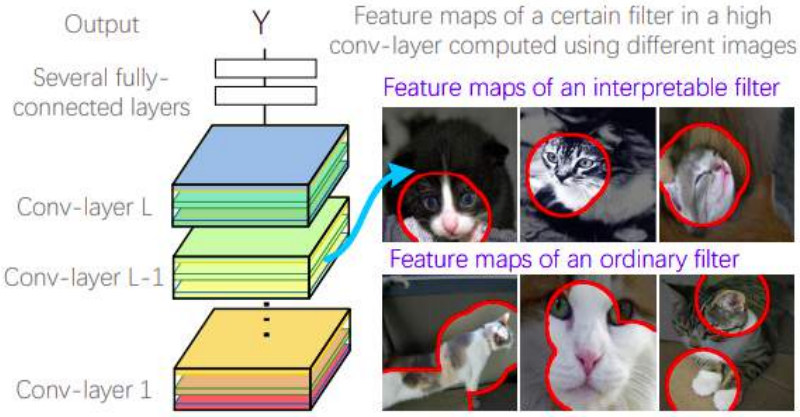

Рис. 16: Сравнение областей, за которые отвечает канал традиционной и интерпретируемой CNN. Изображение взято из [6].

В традиционной CNN свёртка глубокого слоя может описывать смесь паттернов, то есть одна и та же свёртка может быть активирована разными частями одного изображения. Такие сложные представления глубоких слоёв негативно сказываются на их визуализации. В интерпретируемой CNN каждая свёртка активируется определённой частью изображения. Таким образом, появляется возможность явно определить, какие части объекта запоминаются в CNN для классификации.

Достичь такой локализации помогает своеобразное маскирование тензора. Для канала размера $n \times n$ строится n^2 масок размера $n \times n$ следующего вида:

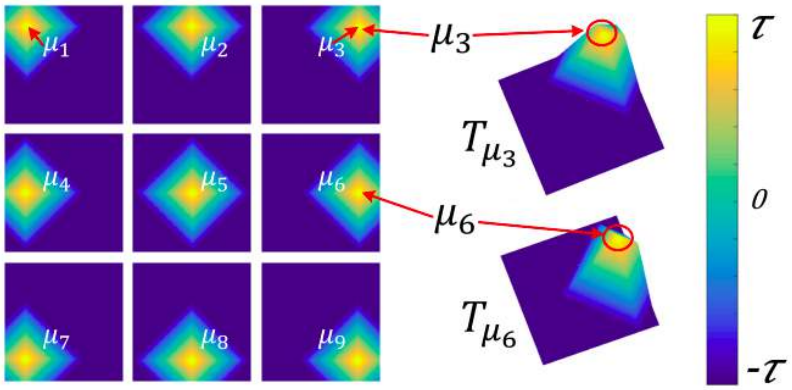

Рис. 17: Вид масок в интерпретируемой CNN. Изображение взято из [6].

Каждая маска имеет максимальный элемент в конкретной позиции и ромбовидные линии уровня, исходящие из этого элемента. Вообще говоря, линии уровня вполне могут иметь вид окружностей, то есть выступать в роли окрестностей по L2-метрике. Авторы используют L1-метрику лишь для упрощения вычислений. Итак,

на прямом проходе находится максимальный элемент в канале, после чего весь канал умножается на соответствующую маску. В результате сеть смотрит только на окрестность максимального элемента. Ниже приведена иллюстрация такого маскирования.

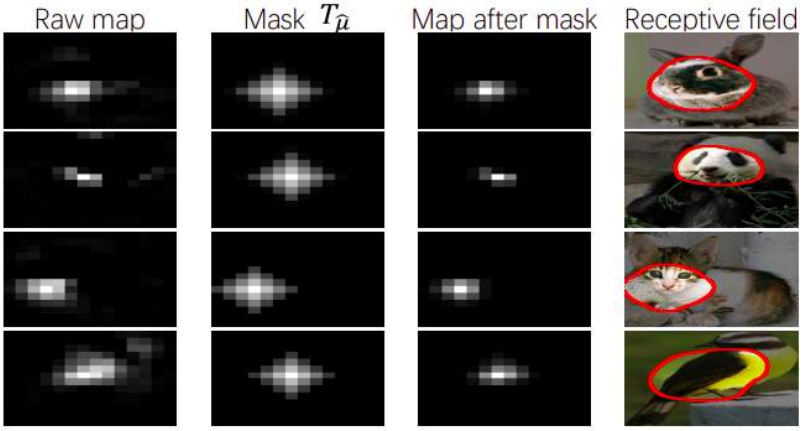

Рис. 18: Процесс маскирования в интерпретируемой CNN. Изображение взято из [6].

Более того, к каждому каналу тензора приписывается своя функция потерь. Её смысл заключается в том, что во время обратного распространения ошибки она сохраняет определённую часть объекта в заданной категории. Ниже на рисунке приведена схема интерпретируемой сети.

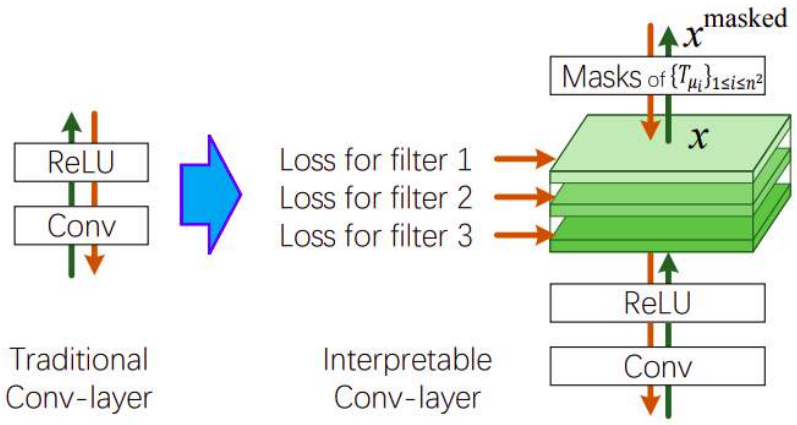

Рис. 19: Общая схема устройства интерпретируемой CNN. Изображение взято из [6].

Ниже на рисунке выделены регионы, с которыми взаимодействуют каналы глубоких слоёв интерпретируемой CNN, а также приведены соответствующие регионы в оригинальной CNN.

Рис. 20: Сравнение регионов изображения, с которыми взаимодействуют каналы интерпретируемой и оригинальной CNN. В первых четырёх строках отражены области, с которыми связан рассматриваемый канал свёрточной CNN. В последних двух строках изображены соответствующие области в оригинальной CNN. Изображение взято из [6].

Таким образом, интерпретируемая CNN обращает внимание только на локальные участки изображения, не смешивая на глубоких слоях информацию с разных его частей. В то же время оригинальная CNN не выделяет регионы, на которые ориентируется при классификации.

6 Grad-CAM и Guided Grad-CAM

Метод Grad-CAM, представленный в [7], является обобщением метода CAM для более широкого класса сетей. Напомним, что CAM создаёт тепловую карту изображений, поданных в CNN для классификации. Однако метод работает только с сетями, в которых последний свёрточный слой пропускается через Global Average Pooling, после чего поступает в полносвязный слой. В свою очередь, Grad-CAM использует информацию о градиенте, поступающего в последний свёрточный слой CNN. Это делается для того, чтобы присвоить значения важности каждому нейрону для конкретного интересующего изображения. Вообще говоря, такую методику можно использовать для интерпретации активаций на любом слое нейросети, но в [6] авторы сосредотачиваются только на интерпретации последнего свёрточного слоя, стоящего перед полносвязным. В работе утверждается, что таким образом сохраняется ком-

промисс между семантикой высокого уровня и подробной пространственной информацией.

Основная идея Grad-CAM заключается в следующем. Обозначим предпоследний слой, находящийся перед полносвязным, за A. Пусть A^k_{ij} - элемент k-го канала в позиции (i, j). Пусть y^c - оценка за класс c, полученная перед softmax. Сначала считается производная y^c по A^k_{ij}, после чего находятся коэффициенты α^c_k усреднением по k-му каналу:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

Затем все каналы суммируются с полученными коэффициентами и взвешенная сумма пропускается через ReLU:

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k)$$

В результате получаем тепловую карту, показывающую участки изображения, на которые смотрит нейросеть при классификации. Пример работы метода для разных классов представлен на рисунке ниже.

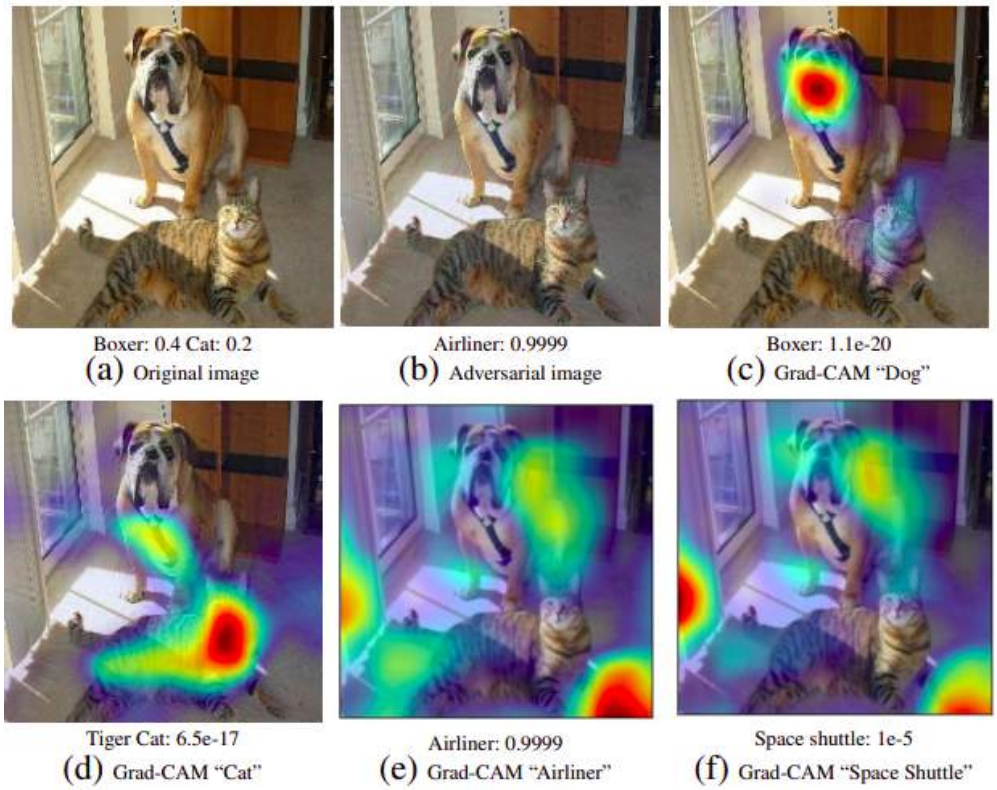

Рис. 21: Пример работы метода Grad-CAM. Изображение взято из [7].

Метод Grad-CAM отчётливо выделяет собаку и кота на изображениях (c) и (d), причём классификация собаки происходит по её морде, в то время как на кота нейросеть смотрит целиком. На изображении (b) представлено обманное изображение авиалайнера. На изображениях (e), (f) нейросеть действительно находит авиалайнер и космический корабль, но классификация проходит неравномерно и по разным участкам изображения. Такой разброс может свидетельствовать о том, что нейросеть пытаются обмануть.

Докажем теперь, что Grad-CAM действительно эквивалентен CAM. Пусть Global Average Pooling в CAM усредняет k каналов тензора A, и пусть A_{ij}^k - элемент k-го канала в позиции (i, j). Эти каналы усредняются и линейно преобразуются, после чего получаются оценки за каждый класс:

$$y^c = \sum_k w_k^c \cdot \frac{1}{Z} \sum_i \sum_j A_{ij}^k$$

Положим выход Global Average Pooling за F. Это вектор, элементы которого вычисляются по формуле $F^k = \frac{1}{Z} \sum_i \sum_j A_{ij}^k$. Пусть w^c - вектор весов полносвязного слоя при классе c. Тогда метод CAM получает окончательный результат по формуле:

$$y^c = \sum_k w_k^c \cdot F^k \tag{1}$$

Продифференцируем по F^k:

$$\frac{\partial y^c}{\partial F^k} = \frac{\frac{\partial y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} = \frac{\partial y^c}{\partial A_{ij}^k} \cdot Z$$

Из (1) следует, что $\frac{\partial y^c}{\partial F^k} = w_k^c$. Это значит, что $w_k^c = Z \cdot \frac{\partial y^c}{\partial A_{ij}^k}$. Просуммировав по i, j, получим:

$$\sum_i \sum_j w_k^c = \sum_i \sum_j Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k}$$

$$Z \cdot w_k^c = Z \cdot \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

В результате получено, что веса, с которыми суммируются каналы в CAM, совпадают с весами из метода Grad-CAM с точностью до константы $\frac{1}{Z}$. Таким образом, при наличии Global Average Pooling слоя методы действительно эквивалентны.

Метод Grad-CAM хорошо различает классы и локализует соответствующие области изображения, но ему недостаёт возможности выделять мелкие детали, как это было, например, в Guided Backpropogation. Напомним, что Guided Backpropogation фактически визуализирует градиенты по изображению, подавляя отрицательные значения после прохода через ReLU. Интуиция подхода заключается в захватывании пикселей, которые активизируют нейроны, а не подавляют их. В [7] был предложен подход, объединяющий методы Grad-CAM и Guided Backpropogation в Guided Grad-CAM. Для этого достаточно перемножить тепловую карту Grad-CAM и соответствующую карту Guided Backpropogation. Результат такого перемножения представлен на рисунке ниже.

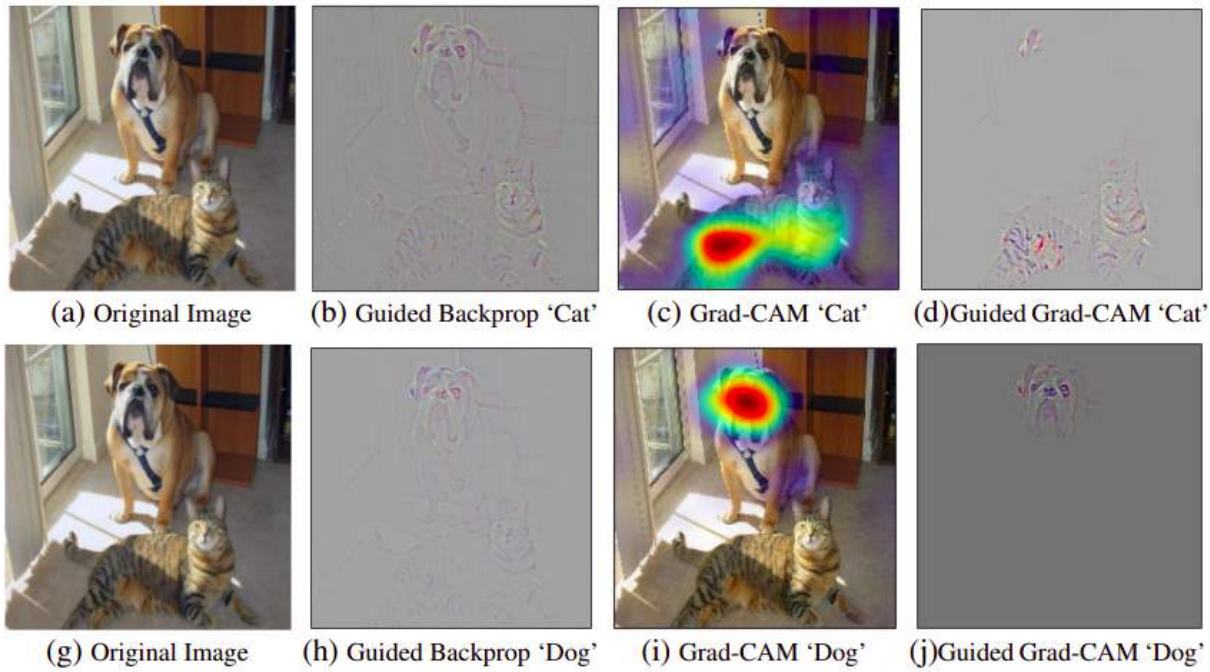

Рис. 22: Пример работы метода Guided Grad-CAM. Изображение взято из [7].

В результате получен более чёткий контур объекта, чем в Guided Backpropogation. Хорошо видны паттерны, такие как полосатая шерсть и круги под глазами. По изображениям Guided Grad-CAM можно сказать, какой объект отвечает за кота или собаку, в то время как по Guided Backpropogation такого сказать нельзя.

Авторы статьи [7] использовали метод Grad-CAM для задач сегментирования объектов. Они использовали сети VGG-16, AlexNet и GoogLeNet, отбирая из них лучшие по качеству. Детекцию проводили методом Grad-CAM по последнему свёрточному слою, данные брались с датасета PASCAL VOC 2012. Им удалось достичь наименьшей ошибки в 56.51%, в то время как CAM показал наименьшую ошибку в 57.2%. Результат сегментации представлен на рисунке ниже.

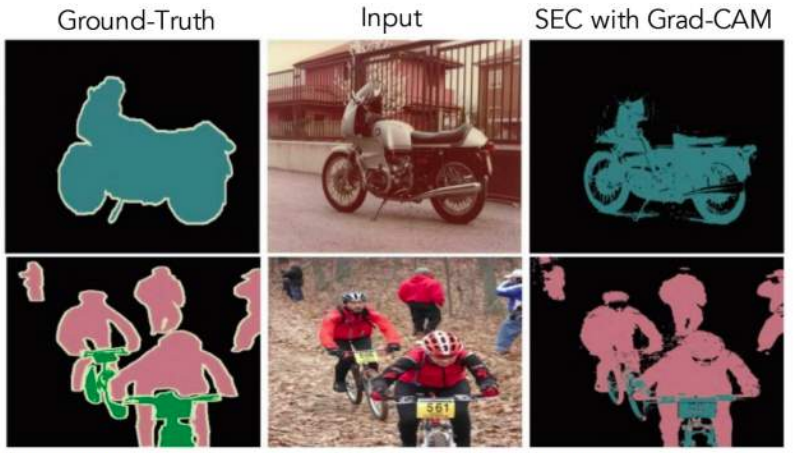

Fig. 4: PASCAL VOC 2012 Segmentation results with Grad-CAM as seed for SEC [32].

Рис. 23: Сегментация методом Grad-CAM. Изображение взято из [7].

7 Стандартные средства в признаковых пространствах

Важной с практической точки зрения задачей является поиск похожих изображений. Очевидно, проблема не может быть решена простым методом ближайших соседей, поскольку похожие в человеческом смысле изображения далеко не всегда обязаны быть близки по евклидовой метрике. Для решения такой задачи совсем не важно учитывать положение объекта или задний фон. Гораздо важнее учесть общие признаки и отличающие объект паттерны. На помощь приходят свёрточные нейросети.

7.1 Поиск соседей в последнем полносвязном слое

Создателями сети AlexNet в [1] был предложен метод поиска ближайших соседей изображения по последнему полносвязному слою. Его интуиция заключается в следующем: если два изображения создают в полносвязном слое близкие по евклидовой метрике признаки, то эти изображения можно считать похожими. Ниже представлено пять примеров из датасета ILSVRC-2010 и шесть их ближайших соседей, найденных таким способом.

Рис. 24: Результат поиска похожих изображений по последнему полносвязному слою. Изображение взято из [1].

Изображения сравнивались по 4096-мерному скрытому слою. Видно, что сами изображения далеко не всегда близки по евклидовой метрике. Например, найденные собаки и слоны появляются при разном освещении и в разных позах. Тем не менее, сеть на глубоком слое приняла их за близкие объекты. Во-первых, это свидетельствует о корректной работе сети. Во-вторых, это позволяет совершать поиск похожих с человеческой точки зрения картинок.

7.2 Уменьшение размерности

Найти похожие картинки и отобразить их на плоскости можно, уменьшив размерность последнего полносвязного слоя до двух. Такой метод был предожен Андреем Карпатым в [этой] статье. Было взято 50.000 изображений из датасета ILSVRC 2012 и из каждого извлечено по 4096-му вектору признаков с последнего полносвязного слоя AlexNet. Затем размерность этого признакового пространства была понижена до двух методом Barnes-Hut t-SNE. Отличие этого метода от стандартного t-SNE заключается в том, что он извлекает признаки за $O(N \cdot logN)$ операций, в то время как t-SNE извлекает их за $O(N^2)$ операций. Небольшие потери в качестве извлечения совсем несущественны, когда речь идёт о таком большом объёме данных.

После всех преобразований изображения можно поместить на плоскость. Результат приведён на рисунке ниже.

Рис. 25: Визуализация изображений на плоскости с помощью Barnes-Hut t-SNE преобразования. Изображения взяты [отсюда].

Невооружённым глазом видна схожесть изображений, лежащих недалеко друг от друга. Тем не менее, на плоскости остались пустые пространства. Их можно заменить ближайшими соседями. Ниже представлено множество изображений, полученное таким образом.

Рис. 26: Визуализация изображений на плоскости с помощью Barnes-Hut t-SNE преобразования и заполнение пустот ближайшими соседями. Изображение взято из [отсюда].

При визуализации расширенного множества изображений отчётливо видны кластеры с изображениями природы, животных, неба. Виден общий цветовой паттерн, объединяющий всё множество изображений в одно.

8 Анализ активации нейронов

Следующий метод визуализации работы нейросети, описанный [здесь], заключается в анализе активации нейронов. Предлагается простая идея - посмотреть на них во время прямого прохода. На рисунке ниже представлена типичная карта активаций.

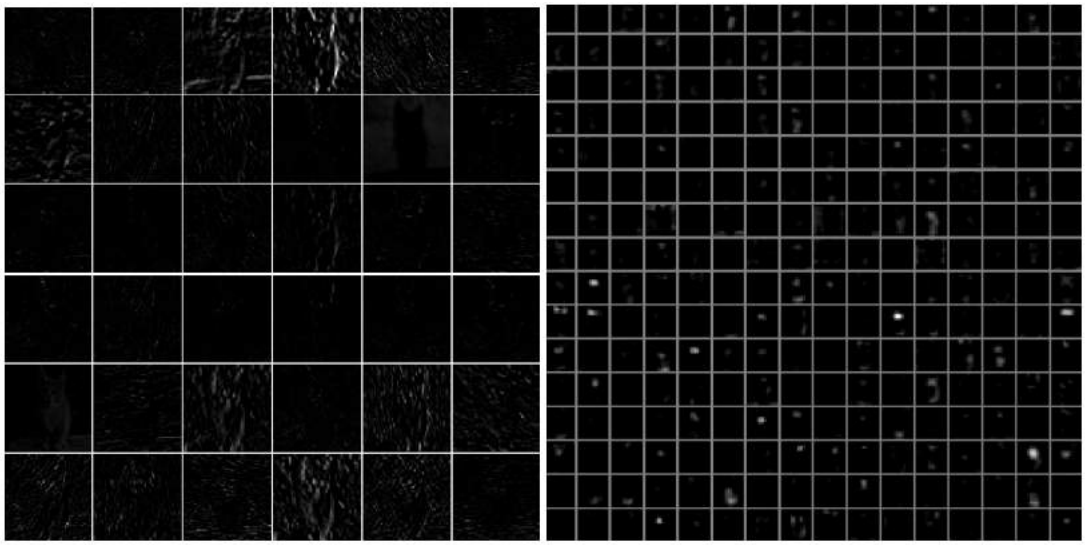

Рис. 27: Карта активации нейронов в CNN. Изображение взято [отсюда].

Каждый квадрат представляет собой карту активаций, соответствующую определённому фильтру. Для сетей с функцией активации ReLU плотность слоёв уменьшается с увеличением глубины. Это вполне естественно, но таким образом можно заметить опасный деффект в работе нейросети - некоторые области состоят целиком из нулей, значит, имеются "мёртвые"фильтры. В таком случае следует уменьшить learning rate при обучении.

Существует возможность посмотреть на изображения, вызывающие максимальную активацию нейрона. Такое уже было проделано в методе Guided Backpropogation: по таким изображениям можно проследить, на что именно активируется нейрон. Вообще говоря, этот приём был описан ещё раньше в [8] в 2013 году. Авторы ставили перед собой цель визуализировать изображения, ответственные за тот или иной нейрон в сети. Для этого находились изображения тестовой выборки, которые вызывали наиболее сильную активации рассматриваемого нейрона. Пример работы метода представлен на рисунке ниже.

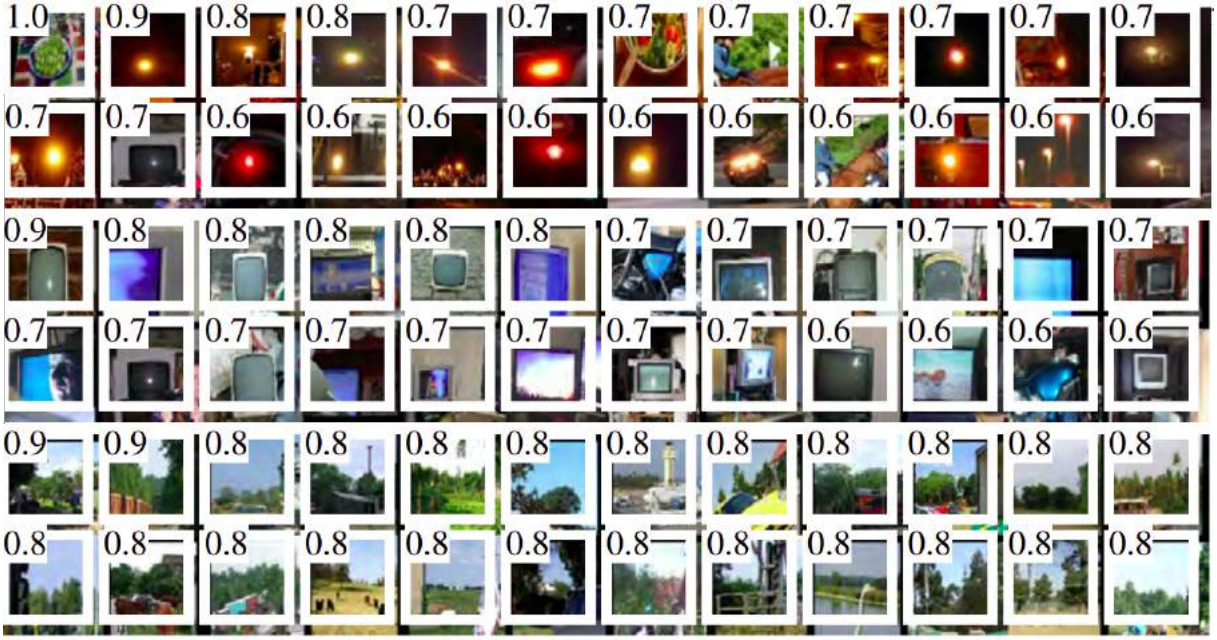

Рис. 28: Top-12 изображений, наиболее сильно активирующих нейрон в CNN. В большинстве случаев можно указать явный паттерн, который максимиирует нейрон, будь то оранжевая лампочка на темном фоне или синий квадрат на светлом. Изображение взято из [8].

9 Occlusion Sensitivity

Метод occlusion sensitivity, предложенный в в [2], является наиболее естественным способом визуализировать то, на что смотрит нейросеть при классификации. Идея подхода заключается в следующем. На изображение накладывается небольшая маска, после чего вычисляется уверенность классификации нейросети. Эта маска перемещается по всему изображению, в результате чего получается тепловая карта вероятностей. Если при наложении маски на некоторую область нейросеть начинает сомневаться в ответе, то эта область является важной. Пример работы метода представлен на рисунке ниже.

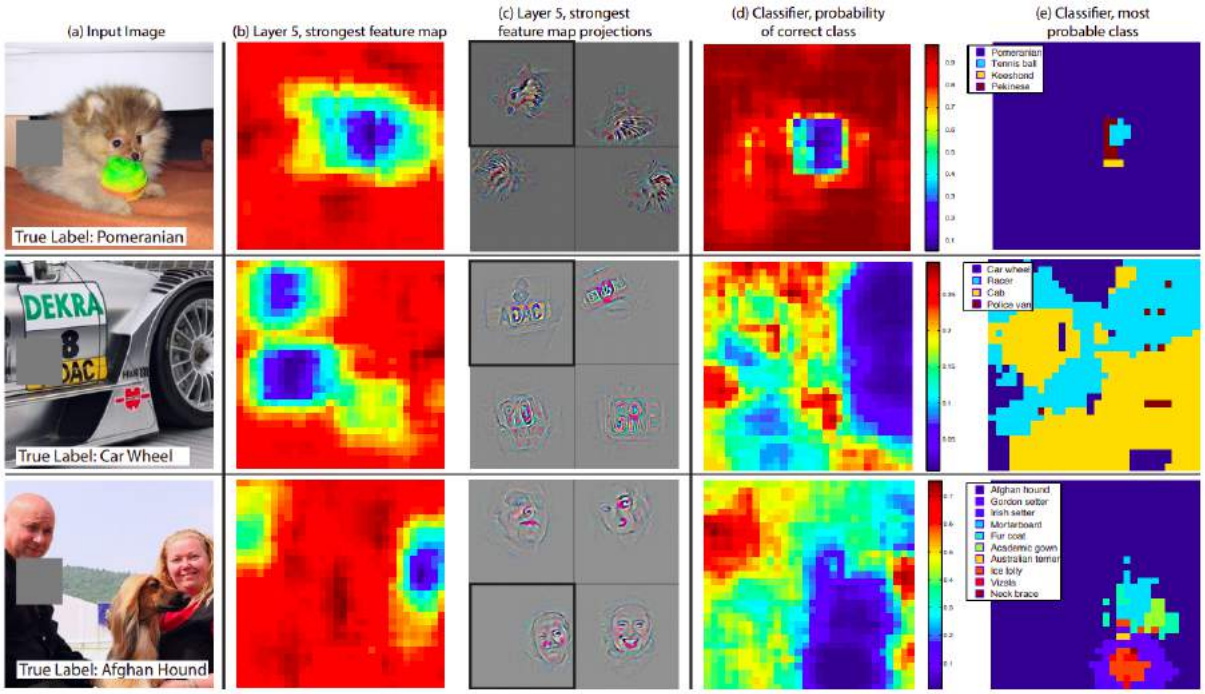

Рис. 29: Три тестовых изображения, к которым был применён метод Occlusion Sensitivity. На рисунке (b) представлено изменение последнего свёрточного слоя: для каждой позиции накладываемой маски подсчитывалась общая активация. На рисунке (c) представлены проекции наиболее значимых областей с последнего свёрточного слоя на входное изображение. На рисунке (d) представлены тепловые карты вероятностей, полученные после наложения закрывающей маски. На рисунке (e) представлена карта из наиболее вероятных классов. Изображение взято из [2].

Похожую тепловую карту рисуют методы CAM и Grad-CAM. Сравним метод Occlusion Sensitivity с методом Grad-CAM:

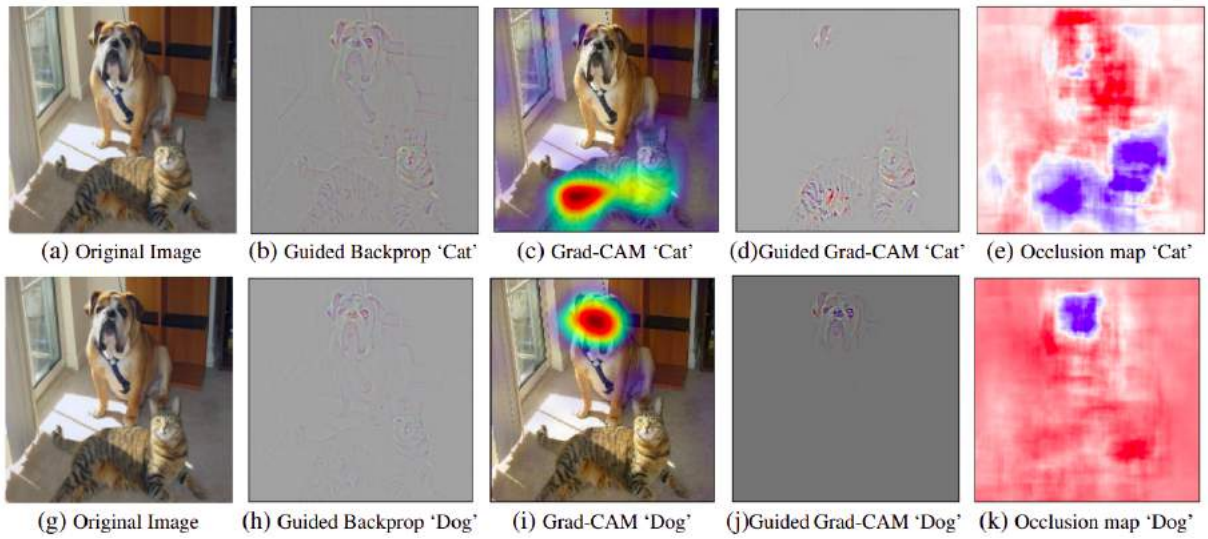

Рис. 30: Результат работы методов Guided Backpropogation, Grad-CAM, Guided Grad-CAM и Occlusion Sensitivity соответсвенно. Взято из [7].

Оба метода корректно выделяют классифицируемых животных, причём, как уже говорилось ранее, при классификации кота нейросеть смотрит на всего кота целиком, а при классификации собаки - только на её морду. Благодаря методу Occlusion Sensitivity можно понять, что если закрыть туловище кота, то уверенность классификации заметно снизится. Значит, нейросеть действительно активно смотрит на его шерсть, и эта шерсть вносит существенный вклад в распознавание.

10 Saliency Maps

Естественной является идея визуализации модулей градиентов по входу. Такой метод был предложен в [9] и получил название Saliency Maps. Интуиция метода очень похожа на интуицию Occlusion Sensitivity: если мы изменим некоторый пиксель входного изображения и получим большое изменение в уверенности в классификации, то этот пиксель является важным. Напротив, если при изменении пикселя уверенность классификации не изменилась, то пиксель не вносит вклад в распознавание. Ниже приведён результат работы Saliency Maps.

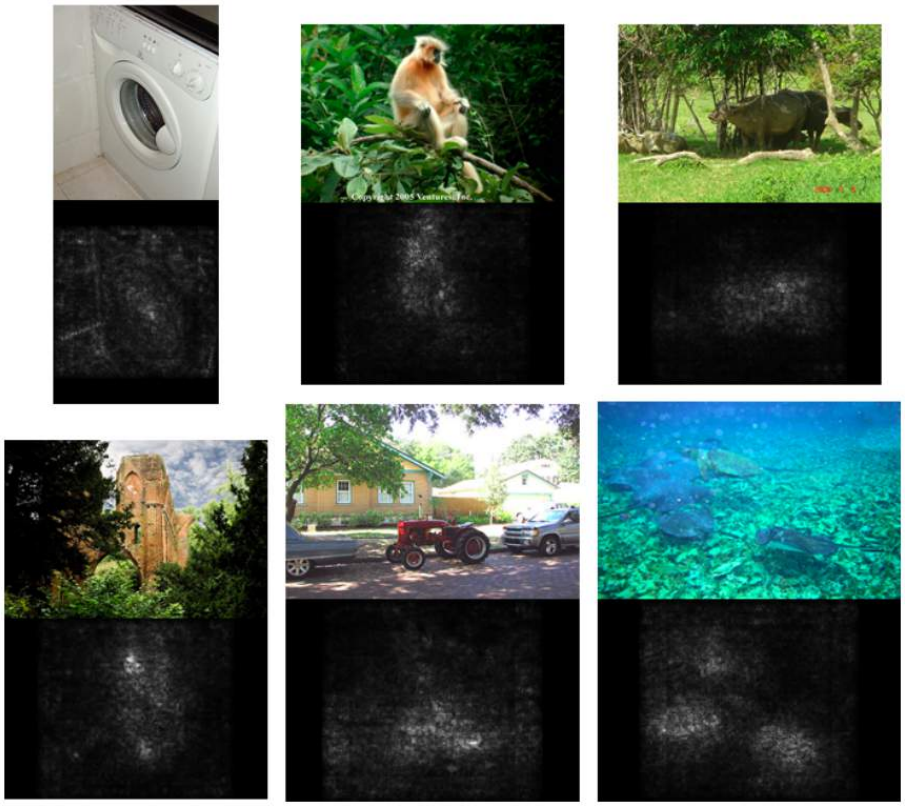

Рис. 31: Результат работы метода Saliency Maps. Изображения взяты из [9].

Метод действительно выделяет объекты: просвечиваются контуры стиральной машины, обезьяны, трактора. Но кроме этого мы видим много лишних объектов. Так, например, подав на вход изображение оленя, Saliency Maps выдаст следующий результат:

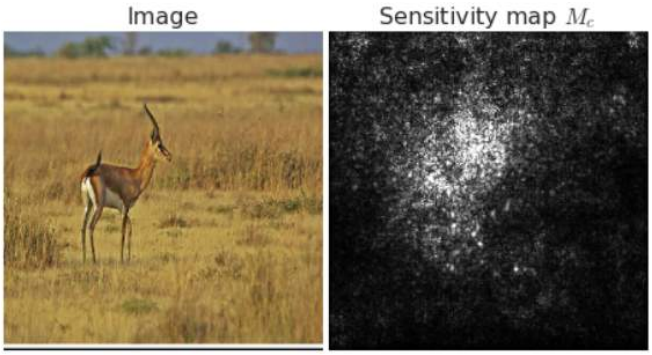

Рис. 32: Результат работы метода Saliency Maps. Рисунок взят из [10].

Решить эту проблему помогает метод SmoothGrad, предложенный в [10]. Его идея заключается в удалении шума с помощью шума. Метод имеет два гиперпараметра: σ - уровень шума, n - количество запусков, по которым усредняется ответ. В работоспособности такого подхода можно убедиться на рисунках ниже.

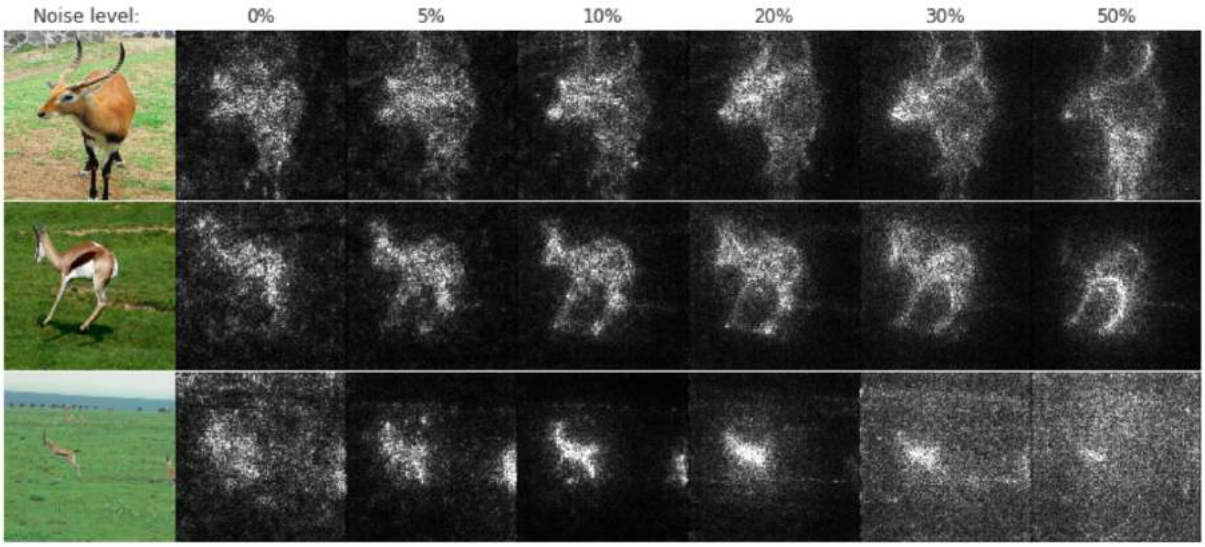

Рис. 33: Результат работы метода SmoothGrad. Представлено три тестовых изображения и результат работы SmoothGrad при разном значении параметра σ. Взято из [10].

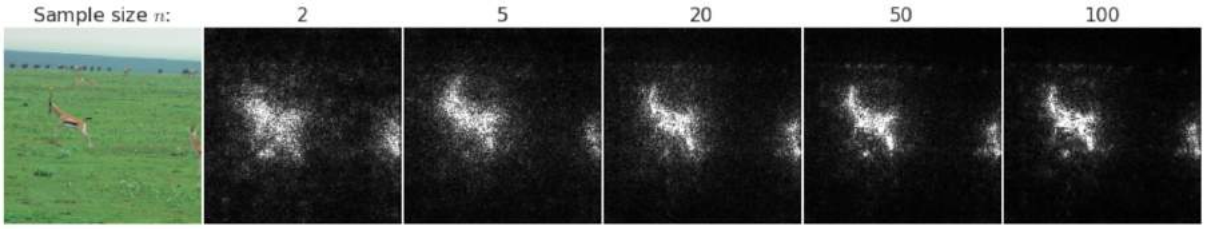

Рис. 34: Результат работы метода SmoothGrad. Представлено тестовое изображение и результат работы SmoothGrad при разном значении параметра n. Взято из [10].

Мы видим, что если зашумлять входное изображение, то выход Saliency Maps становится более чётким. Можно заметить, что оптимальное значение входного зашумления лежит в районе 15%. Если продолжать вносить шум, то результат работы Saliency Maps резко ухудшается. При этом ограничений на число запусков n нет, ведь с увеличением этого параметра результат становится только лучше.

11 FullGrad

Одним из наиболее современных подходов является предложенный в [11] метод FullGrad. Этот метод является модификацией Grad-CAM. Авторы утверждают, что им удалось добиться лучшей локализации объектов и более чёткого вырисовывания контуров.

Напомним, что метод Grad-CAM строил тепловые карты путём вычисления градиента выхода по изображению. Тепловые карты, используемые в методе FullGrad, получены путём агрегирования компонентов полного градиента. Понятие полного градиента было также введено в [11] - это разложение ответа нейронной сети на компоненты чувствительности ко входным данным и чуствительности к каждому нейрону. Говоря иначе, дифференцирование происходит не только по входному изображению, но и по промежуточным признаковым пространствам.

Ниже на рисунке приведён пример работы метода FullGrad.

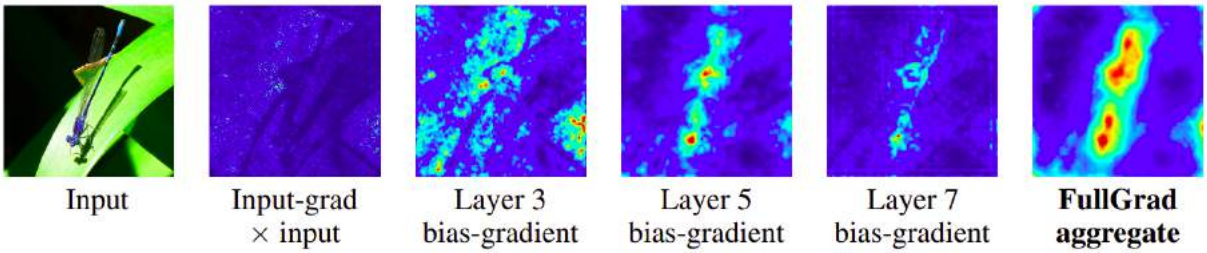

Рис. 35: Визуализация градиентов по различным слоям в обученной VGG-16. На последнем рисунке представлена агрегация всех градиентов методом FullGrad. Изображение взято из [11].

Вызывает интерес сравнение методов FullGrad и Grad-CAM.

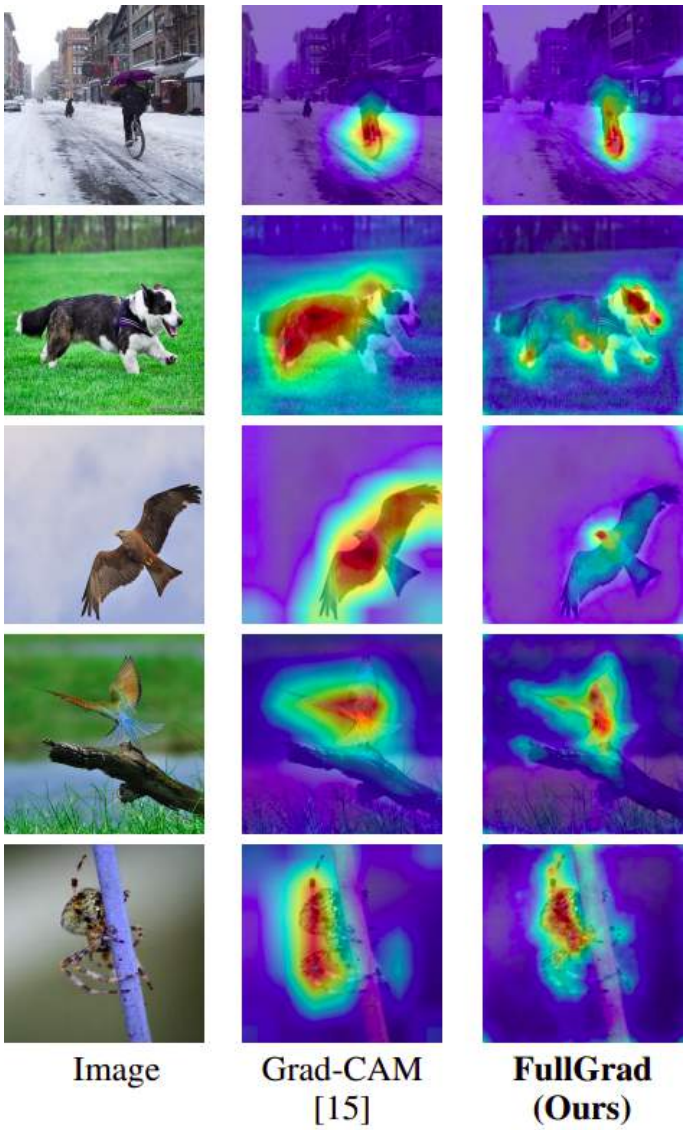

Рис. 36: Сравнение методов FullGrad и Grad-CAM на пяти тестовых изображениях. Рисунок взят из [11].

Наиболее заметное отличие двух методов заключается в том, что FullGrad выделяет гораздо более небольшие области изображения. Это значит, что FullGrad лучше локализует объект, на который смотрит нейросеть, выделяя самую важную его часть.

12 Заключение

Визуализация нейросети является важным этапом отладки. Она позволяет лучше понять принципы работы сети, найти её ошибки и возможности для улучшения. И если в CNN есть возможность следить за свёртками первого слоя как за изображениями, то в более глубоких признаковых пространствах возникают определённые проблемы. В настоящей работе был рассмотрен ряд методов визуализации и интерпретации глубоких слоёв свёрточной нейросети: Deconvnet, CAM, Guided Backpropogation, Grad-CAM и Guided Grad-CAM, Occlusion Sensitivity, Saliency Maps и FullGrad.

В работе рассмотрен и другой способ решения проблемы, заключающийся в построении так называемой интерпретируемой CNN. Были разобраны стандартные методы в признаковых пространствах, с помощью которых стало возможным искать похожие в человеческом смысле изображения.

Таким образом, появилось лучшее понимание того, как именно устроены свёрточные нейросети.

Список используемой литературы

[1] Krizhevsky A. et al. ImageNet Classification with Deep Convolutional Neural Networks. 2012. // https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[2] Matthew D. Zeiler, Rob Fergus. Visualizing and Understanding Convolutional Networks. November, 2013. // https://arxiv.org/pdf/1311.2901.pdf

[3] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba. Learning Deep Features for Discriminative Localization. December, 2015. // https://arxiv.org/pdf/1512.04150.pdf

[4] Maxime Oquab, Leon Bottou, Ivan Laptev, Josef Sivic. Weakly-supervised learning with convolutional neural networks. 2015. //
https://www.di.ens.fr/ josef/publications/Oquab15.pdf

[5] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller. Striving for simplicity: the all convolutional net. April, 2015. //
https://arxiv.org/pdf/1412.6806.pdf

[6] Quanshi Zhang, Ying Nian Wu, Song-Chun Zhu. Interpretable Convolutional Neural Networks. 2018. //
https://arxiv.org/pdf/1710.00935.pdf

[7] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. December, 2019. //
https://arxiv.org/pdf/1610.02391.pdf

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2013. //
https://arxiv.org/pdf/1311.2524.pdf

[9] Karen Simonyan, Andrea Vedaldi, Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. April, 2014. //
https://arxiv.org/pdf/1312.6034.pdf

[10] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, Martin Wattenberg. SmoothGrad: removing noise by adding noise. June, 2017. //
https://arxiv.org/pdf/1706.03825.pdf

[11] Suraj Srinivas, François Fleuret. Full-Gradient Representation for Neural Network Visualization. May, 2019. //
https://arxiv.org/pdf/1905.00780.pdf