Министерство науки и высшего образования Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Магистерская программа
«Методы и технологии искусственного интеллекта»

Раконяц Даниела

Интеграция графов знаний с компьютерным зрением: методы и подходы

Обзор исследований

Научный руководитель:
кандидат физ-мат наук
Куратов Юрий Михайлович

Москва

2024 г.

Содержание

1 Введение **3**

2 Aligning Knowledge Graph with Visual Perception for Object-goal Navigation **3**

2.1 Введение 3

2.2 Методология 6

2.3 Возможные улучшения 6

3 The More You Know: Using Knowledge Graphs for Image Classification **7**

3.1 Введение 7

3.2 Методология 8

3.2.1 Графовая нейронная сеть 8

3.2.2 Нейронная сеть для поиска по графам 10

3.3 Пайплайн изображений и базовые модели 13

3.4 Результаты 14

3.5 Наборы данных 14

3.6 Построение графа знаний 15

3.7 Детали обучения 16

3.8 Количественная оценка 16

3.9 Возможные улучшения 18

4 Context-driven Visual Object Recognition based on Knowledge Graphs **18**

4.1 Введение 18

4.2 Типы контекстов 21

4.3 Предсказания 25

4.4 Эксперименты 25

4.5 Детали реализации 26

4.6 Результаты 26

4.7 Возможные улучшения 29

1 Введение

Современные задачи компьютерного зрения, такие как навигация, классификация изображений и распознавание объектов, всё чаще требуют понимания контекста и более глубокой семантической обработки данных. Традиционные подходы, основанные только на визуальных признаках, сталкиваются с ограничениями при обработке сложных или зашумленных данных.

Графы знаний (Knowledge Graphs) обеспечивают возможность структурировать информацию о связях между объектами, добавляя семантические и контекстуальные слои. Это делает их полезным инструментом для интеграции с методами компьютерного зрения, улучшая способность моделей анализировать сцены, обобщать знания и принимать решения.

2 Aligning Knowledge Graph with Visual Perception for Object-goal Navigation

2.1 Введение

Навигация по объектам-целям – сложная задача,агент должен направляться к конкретным объектам на основе визуальных наблюдений от первого лица.Способность агента понимать окружающую обстановку играет решающую роль в успешном поиске объектов.Существующие навигационные системы, основанные на графах знаний, часто используют дискретные категориальные однократные векторы и стратегию подсчёта голосов для построения графового представления сцен, что приводит к несоответствию визуальным образам. Чтобы обеспечить более точное и последовательное описание сцены и решить проблему несоответствия, авторы статьи предлагают метод согласования графа знаний с визуальным восприятием (AKGVP) для навигации по объектам и целям. Технически такой подход предполагает непрерывное моделирование иерархической структуры сцены и использует предварительное обучение визуальному языку для согласования описания на естественном языке с визуальным восприятием. Интеграция непрерывной архитектуры графа знаний и многомодального согласования признаков позволяет навигатору выполнять навигацию без дополнительных действий. Метод оценен с помощью симулятора AI2-THOR и проведена серия экспериментов, чтобы продемонстрировать эффективность и про-

изводительность навигатора. Традиционные подходы сталкиваются с трудностями в условиях ограниченной видимости из-за отсутствия информации о пространственном расположении объектов, что препятствует эффективной навигации. Для успешной навигации по объектам-целям необходимо превзойти возможности восприятия и дать агентам полное представление об окружающей среде, чтобы они могли принимать разумные решения для достижения своих целей. Это требует бесшовной интеграции восприятия, языка и действий, когда агенты адаптируют свои навигационные стратегии на основе визуального ввода в реальном времени, языковых инструкций и сенсорной обратной связи. Успешная визуальная навигация зависит от сопоставления визуальных наблюдений с действиями и обучения машин воспринимать объекты, определять их свойства и рассуждать о взаимосвязях. Агенты должны обладать расширенными способностями к рассуждению, планированию и выполнению целенаправленных действий, которые остаются адаптивными к динамическим изменениям окружающей среды. Следовательно, изучение информативных визуальных представлений и моделирование семантических карт или графов знаний/сцен становятся незаменимыми для повышения производительности в незнакомых условиях.

Навигаторы, ориентированные на объекты и цели, делятся на три основные категории: сквозные методы, методы на основе графа знаний и методы на основе семантической карты. При сквозном подходе агенты собирают данные (в том числе RGB-изображения, глубину и позу) из окружающей среды на каждом временном шаге. Используя эти входные данные и категории объектов, агенты используют различные временные модели, такие как RNNS, LSTMs, модели на основе памяти, или даже без явного временного моделирования, чтобы определить последующие действия. Сквозной подход основан исключительно на входных данных, таких как RGB-изображения, для непосредственной генерации действий. Тем не менее, он часто испытывает трудности с обобщением в сложных сценариях. Чтобы решить эту проблему, исследователи внедрили дополнительные методы моделирования, такие как графы знаний и семантические карты . Графы знаний обеспечивают надёжные возможности представления, фиксируя взаимосвязи сущностей и контекстную информацию. Графические структуры объединяют огромное количество структурированных и полуструктурированных данных. В отличие от этого, семантические карты позволяют воспринимать пространство, описывая расположение объектов, направления и геометрию. Они обновляются в режиме реального времени, что способствует

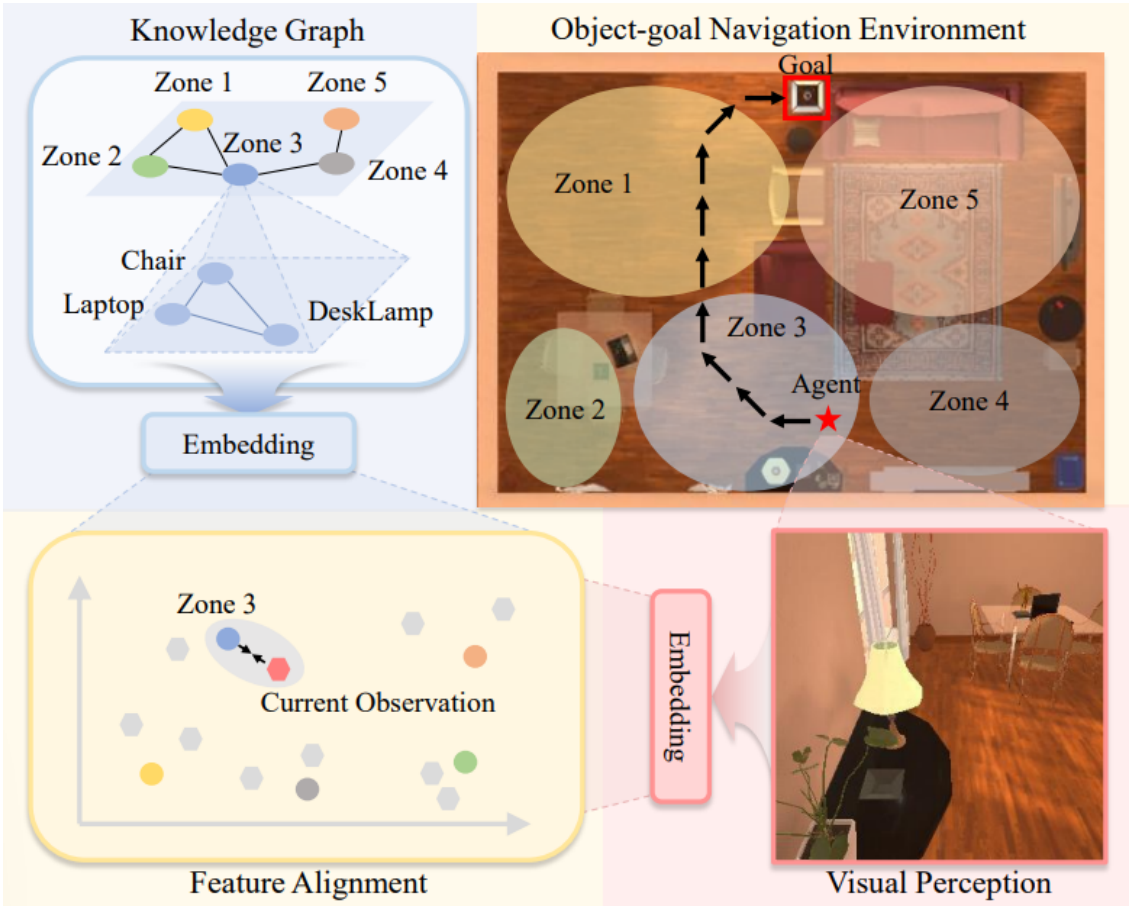

Рис. 1: Основная идея метода AKGVP. Чтобы эффективно описывать одну и ту же среду, можно использовать два типа данных: графы знаний, полученные из описаний на естественном языке, и изображения наблюдений, характеризующиеся визуальными описаниями. Основная цель — объединить эти два типа данных в общем пространстве признаков, чему способствует предварительное обучение визуальному языку. В конечном итоге эти типы данных объединяются для принятия решений.

быстрой адаптации в динамичных средах.

Тем не менее, существующие навигаторы на основе графов знаний имеют существенный недостаток, в первую очередь связанный с их сильной зависимостью от дискретных категориальных векторов и стратегии подсчета голосов для построения графического представления сцен. К сожалению, такой подход часто приводит к несоответствию между этими категориальными представлениями и реальными визуальными образами сцены. Полагаться исключительно на потери при принятии решений, такие как обучение с подкреплением, оказывается сложной задачей при эффективном согласовании двух методов. Устранение этого ограничения имеет первостепенное значение для повышения производительности и точности навигаторов "объект-цель"в реальных сценариях. Учитывая это, исследование направлено на

расширение возможностей навигаторов, основанных на графах знаний, путем сосредоточения внимания на улучшении их возможностей и общей производительности.

2.2 Методология

В задаче «Объектно-целевая навигация» (ObjNav) агент находится в трёхмерной внутренней среде Q и стремится эффективно приблизиться к заранее определённым целевым объектам G за наименьшее количество шагов. В каждом эпизоде агент случайным образом инициализируется в позиции $l = \{x, z, \theta_1, \theta_2\}$ в среде $q \in Q$. Здесь (x, z) — координаты в плоскости, а θ_1 и θ_2 — углы рыскания и тангажа соответственно. В каждый момент времени t агент изучает функцию политики, обозначаемую как $\pi(a_t|s_t, g)$, которая предсказывает подходящее действие $a_t \in \mathcal{A}$ на основе текущего состояния (наблюдения от первого лица) s_t и целевого объекта $g \in G$. Здесь $\mathcal{A} = \{MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, Done\}$. Примечательно, что агент самостоятельно определяет завершение действия, полагаясь на собственное суждение, а не на сигналы окружающей среды. Движение агента происходит в дискретизированном пространстве сцены с интервалами в 0,5 м. Угол поворота и наклона по вертикали остаются фиксированными на 45 и 30 градусов соответственно. Эпизод считается успешным, если агент выбирает действие «Готово», когда целевой объект виден в пределах порогового расстояния 1,5 м, что указывает на близость. Если агент не соответствует этому критерию, происходит сбой.

2.3 Возможные улучшения

Несмотря на успешные результаты, в методе остаются области для улучшения:
- Масштабируемость графа знаний: По мере увеличения количества объектов и связей в графе может возрасти сложность обработки. Использование более эффективных методов работы с большими графами могло бы улучшить производительность.
- Обработка динамических сред: Граф знаний считается статичным, что ограничивает его применение в средах с изменяющимися объектами. Добавление механизмов обновления графа в реальном времени могло бы решить эту проблему.
- Обучение на реальных данных: В статье рассматриваются симуляционные среды, однако эффективность модели на реальных данных остается не до конца изученной.
- Интеграция дополнительных сенсоров: Добавление данных из других источников, таких как карты глубины или информации от лидара, могло бы улучшить точность

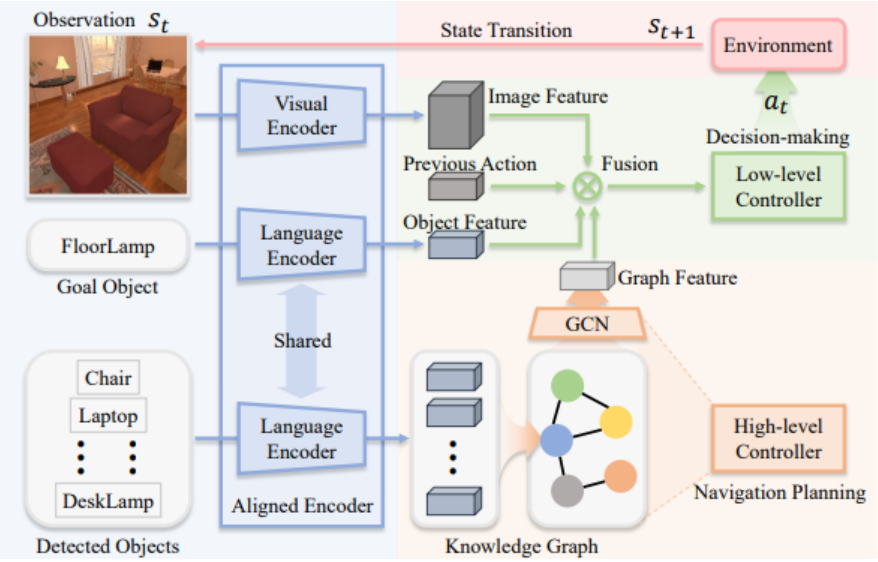

Рис. 2: Схема нашего метода AKGVP. AKGVP состоит из трёх основных компонентов: согласованного кодировщика, контроллера высокого уровня и контроллера низкого уровня. Кодировщик играет важнейшую роль, отдельно кодируя изображения и естественный язык, что способствует согласованию функций посредством мультимодального предварительного обучения. Контроллер высокого уровня использует моделирование графа знаний для эффективного планирования подцелей навигатора, направляя движение агента по разным зонам. С другой стороны, низкоуровневый контроллер использует объединённую мультимодальную информацию для принятия обоснованных решений, что позволяет агенту взаимодействовать с окружающей средой и умело контролировать свои движения.

восприятия.

3 The More You Know: Using Knowledge Graphs for Image Classification

3.1 Введение

Авторы данной статьи отмечают, что люди обладают замечательной способностью распознавать большое количество визуальных концепций,очень часто не очевидных опираясь не только на внешний вид,но и на знания о мире,полученные с помощью опыта и языка.Что же такого есть у людей чего нет у современных подходов?Структкрированные знания и рассуждения.Например, мы могли прочитать в книге о «слоновой землеройке» (возможно, даже видели её) и получить знания, которые пригодятся для распознавания этого животного. На рисунке 4 показано,

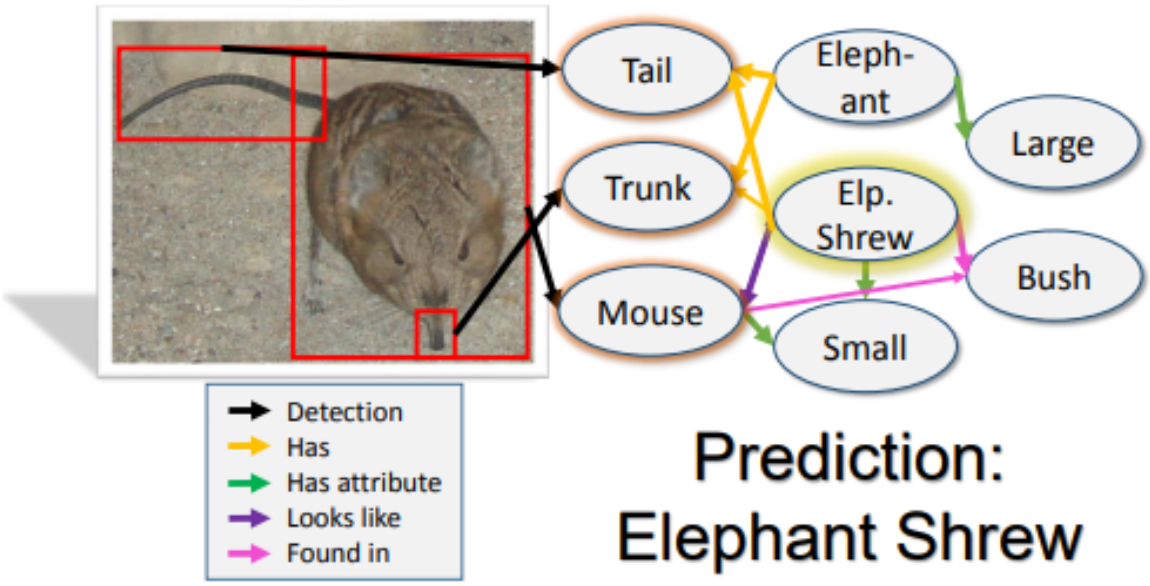

Рис. 3: Пример того, как семантические знания о мире помогают в классификации. Здесь мы видим слоновую землеройку. Люди способны правильно классифицировать её на основе того, что мы знаем о слоновой землеройке и других похожих животных.

как мы можем использовать свои знания о мире в этой задаче. Мы могли бы знать, что слоновая землеройка похожа на мышь, у неё есть хобот и хвост, она родом из Африки и часто встречается в кустах. Обладая этой информацией, мы, вероятно, смогли бы опознать слоновую землеройку, если бы увидели её в дикой природе. Мы делаем это, сначала распознавая (мы видим в кустах маленький похожий на мышь объект со хоботом), вспоминая (мы думаем о животных, о которых слышали, и об их частях тела, среде обитания и особенностях), а затем рассуждая (это слоновая землеройка, потому что у неё есть хобот и хвост, и она похожа на мышь, в то время как у мышей и слонов нет всех этих особенностей). Благодаря этой информации, даже если мы видели только одну или две фотографии этого животного, мы сможем его классифицировать.

3.2 Методология

3.2.1 Графовая нейронная сеть

Идея GGNN заключается в том, что, имея граф с N узлами, нужно получить некоторый результат, который может быть результатом для каждого узла графа $o_1, o_2, ...o_N$ или глобальным результатом o_G. Это достигается путём обучения модели

распространения, аналогичной LSTM. Для каждого узла в графе v у нас есть представление скрытого состояния $h_v^{(t)}$ на каждом временном шаге t.Начинаем с $t = 0$ с начальными скрытыми состояниями x_v, которые зависят от задачи. Например, для обучения достижимости графа это может быть двухбитовый вектор, указывающий, является ли узел исходным или конечным. В случае визуального анализа графа знаний x_v может быть однобитовой активацией, представляющей уверенность в наличии категории на основе детектора объектов или классификатора.

Затем используем структуру графа, закодированную в матрице A, которая служит для извлечения скрытых состояний соседних узлов на основе типов рёбер между ними. Затем скрытые состояния обновляются с помощью модуля обновления с воротами, аналогичного LSTM. Базовая рекуррентная функция для этой **сети распространения** выглядит так:

$$h_v^{(1)} = [x_v^T, 0]^T \tag{3.1}$$

$$a_v^{(t)} = A_v^T [h_1^{(t-1)} ... h_N^{(t-1)}]^T + b \tag{3.2}$$

$$z_v^t = \sigma(W^z a_v^{(t)} + U^z h_v^{(t-1)}) \tag{3.3}$$

$$r_v^t = \sigma(W^r a_v^{(t)} + U^r h_v^{(t-1)}) \tag{3.4}$$

$$\widetilde{h_v^t} = tanh(W a_v^{(t)} + U(r_v^t \odot h_v^{(t-1)})) \tag{3.5}$$

$$h_v^{(t)} = (1 - z_v^t) \odot h_v^{(t-1)} + z_v^t \odot \widetilde{h_v^t} \tag{3.6}$$

где $h_v^{(t)}$ — скрытое состояние узла v на временном шаге t, x_v — аннотации для конкретной задачи, A_v — матрица смежности графа для узла v, а W и U — обучаемые параметры. Уравнение 3.1 — это инициализация скрытого состояния с помощью x_v и пустых измерений. Уравнение 3.2 показывает распространение обновлений от соседних узлов. Уравнение (3-6) объединяет информацию от соседних узлов и текущее скрытое состояние узлов для вычисления следующего скрытого состояния.

После T временных шагов мы получаем конечные скрытые состояния. Выходные данные на уровне узлов можно вычислить следующим образом:

$$o_v = g(h_v^{(T)}, x_v) \tag{3.7}$$

где g — полносвязная сеть, **выходная сеть**, а x_v — исходная аннотация для узла.

3.2.2 Нейронная сеть для поиска по графам

Самая большая проблема при адаптации GGNN для задач, связанных с изображениями, — масштабируемость вычислений. Например, в NEIL [**Chen13**] более 2000 концепций, а в NELL [**NELL10**] более 2 млн достоверных убеждений. Даже после сокращения до размеров, необходимых для нашей задачи, эти графы всё равно будут огромными. Прямое распространение в стандартной GGNN занимает $O(N^2)$ времени, где N — количество узлов, а обратное распространение — $O(N^T)$, где T — количество шагов распространения.Авторы проводят простые эксперименты с GGNN на синтетических графах и обнаруживают, что при количестве узлов более 500 прямой и обратный проходы занимают более 1 секунды на одном экземпляре, даже при допущениях в отношении параметров. При 2000 узлах это занимает более минуты для одного изображения. Использование GGNN «из коробки» невозможно. В следствии чего,предлагают следующее решение,нейронная сеть для поиска по графам (GSNN). Как следует из названия, идея заключается в том, что вместо того, чтобы выполнять рекуррентное обновление для всех узлов графа одновременно,начинают с нескольких начальных узлов на основе входных данных и расширяют только те узлы, которые полезны для конечного результата. Таким образом, вычисляются шаги обновления только для подмножества узлов графа. Во время обучения и тестирования определяют начальные узлы графа на основе вероятности присутствия объекта, определяемой детектором объектов или классификатором. Для экспериментов авторы использовали более быстрый R-CNN [**renNIPS15fasterrcnn**] для каждой из 80 категорий COCO. Для получения результатов, превышающих некоторый выбранный порог, выбирались соответствующие узлы на графике в качестве начального набора активных узлов.

Как только есть начальные узлы, нужно добавить узлы, смежные с начальными узлами, в активный набор. Учитывая начальные узлы, хочется сначала распространить представления о начальных узлах на все соседние узлы. Однако после первого шага нужно решить, какие узлы расширять дальше. Поэтому обучают функцию оценки для каждого узла, которая определяет, насколько он «важен». После каждого шага распространения для каждого узла в текущем графе прогнозируется оценка важности

$$i_v^{(t)} = g_i(h_v, x_v) \tag{3.8}$$

где g_i — обученная сеть, **сеть важности**.

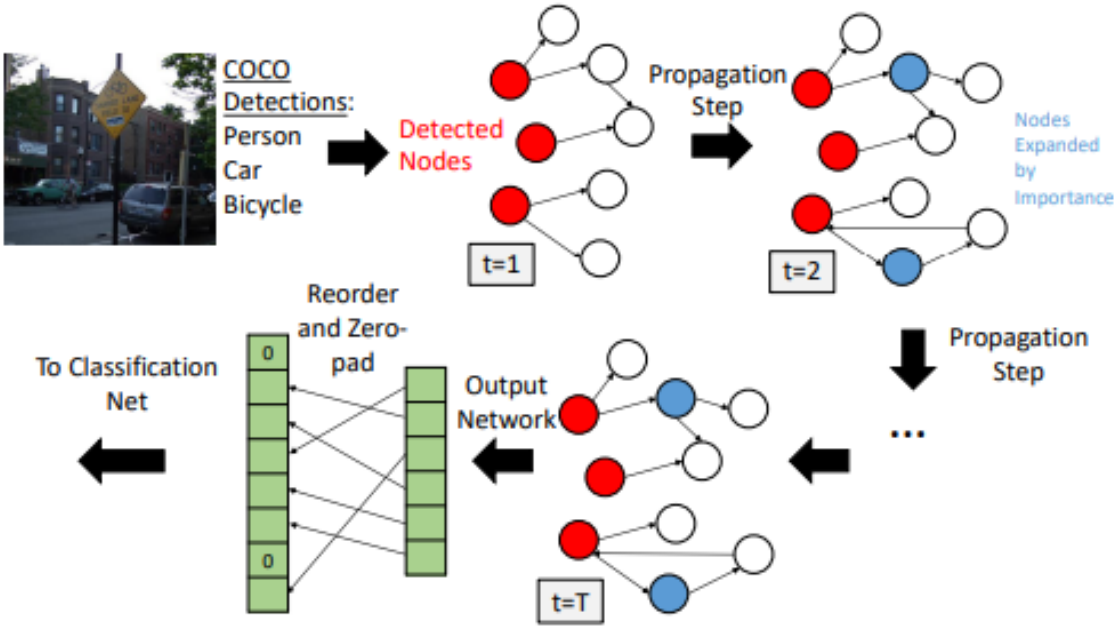

Рис. 4: Расширение нейронной сети поиска по графу. Начинается с обнаруженных узлов и расширяется за счет соседей. Добавляет узлы, примыкающие к узлам расширения, прогнозируемым по сети важности.

Как только получены значения i_v, берутся узлы с наивысшим рейтингом P, которые никогда не расширялись, и добавляются их в расширенный набор, а также добавляются все узлы, смежные с этими узлами, в активный набор. На рисунке 4 показано это расширение. При $t = 1$ расширяются только обнаруженные узлы. При $t = 2$ расширяются выбранные узлы на основе значений важности и добавляем их соседей к графу. На последнем временном шаге T вычисляются выходные данные для каждого узла, переупорядочиваются и обнуляются выходные данные в окончательной классификационной сети.

Чтобы настроить сеть важности, присваивается целевое значение важности каждому узлу на графике для данного изображения. Узлам, соответствующим основным понятиям на изображении, присваивается значение важности, равное 1. Соседям этих узлов присваивается значение γ. Узлы, находящиеся на расстоянии двух переходов, имеют значение γ^2 и так далее. Идея заключается в том, что узлы, наиболее близкие к конечному результату, являются наиболее важными для расширения.

Теперь есть сквозная сеть, которая принимает в качестве входных данных набор начальных узлов и аннотаций и выдает выходные данные для каждого узла для каждого из активных узлов в графе. Она состоит из трех наборов сетей: сети распро-

странения, сети важности и выходной сети. Окончательный результат решения задачи с изображениями может быть получен путём обратного распространения ошибки от конечного результата конвейера через выходную сеть, а потеря значимости распространяется обратно через каждый из важных результатов. См. рисунок 5, чтобы увидеть архитектуру GSNN. Сначала x_{init}, вероятности обнаружения инициализируют $h^{(1)}_{init}$, скрытые состояния изначально обнаруженных узлов. Затем инициализируются $h^{(1)}_{adj1}$, скрытые состояния соседних узлов, значением 0. Затем обновляются скрытые состояния с помощью сети распространения. Значения $h^{(2)}$ затем используются для прогнозирования показателей важности $i^{(1)}$, которые используются для выбора следующих узлов для добавления $adj2$. Затем эти узлы инициализируются с помощью $h^{(2)}_{adj2} = 0$, и скрытые состояния снова обновляются с помощью сети распространения. После T шагов используются все накопленные скрытые состояния h^T для прогнозирования выходов GSNN для всех активных узлов. Во время обратного распространения потери двоичной перекрестной энтропии (BCE) передаются обратно через выходной уровень, а потери важности передаются через сети важности для обновления параметров сети.

Последняя деталь — добавление «смещения узла» в GSNN. В GGNN функция вывода для каждого узла $g(h^{(T)}_v, x_v)$ принимает скрытое состояние и начальную аннотацию узла v для вычисления его вывода. В определённом смысле она не зависит от значения узла. То есть во время обучения или тестирования GSNN принимает граф, который, возможно, никогда раньше не видела, и некоторые начальные аннотации x_v для каждого узла. Затем он использует структуру графа, чтобы распространить эти аннотации по сети, а затем вычислить результат. Узлы графа могут представлять собой что угодно: от человеческих отношений до компьютерной программы. Однако в графовой сети тот факт, что конкретный узел представляет собой «лошадь» или «кошку», вероятно, будет иметь значение, и также можно ограничиться статическим графом, связанным с концепциями изображений. Поэтому вводятся смещения узлов, которые для каждого узла в графе имеют определённые значения. Теперь выходные уравнения выглядят так: $g(h^{(T)}_v, x_v, n_v)$, где n_v — коэффициент смещения, привязанный к конкретному узлу v в общем графе. Это значение хранится в таблице и обновляется с помощью обратного распространения ошибки.

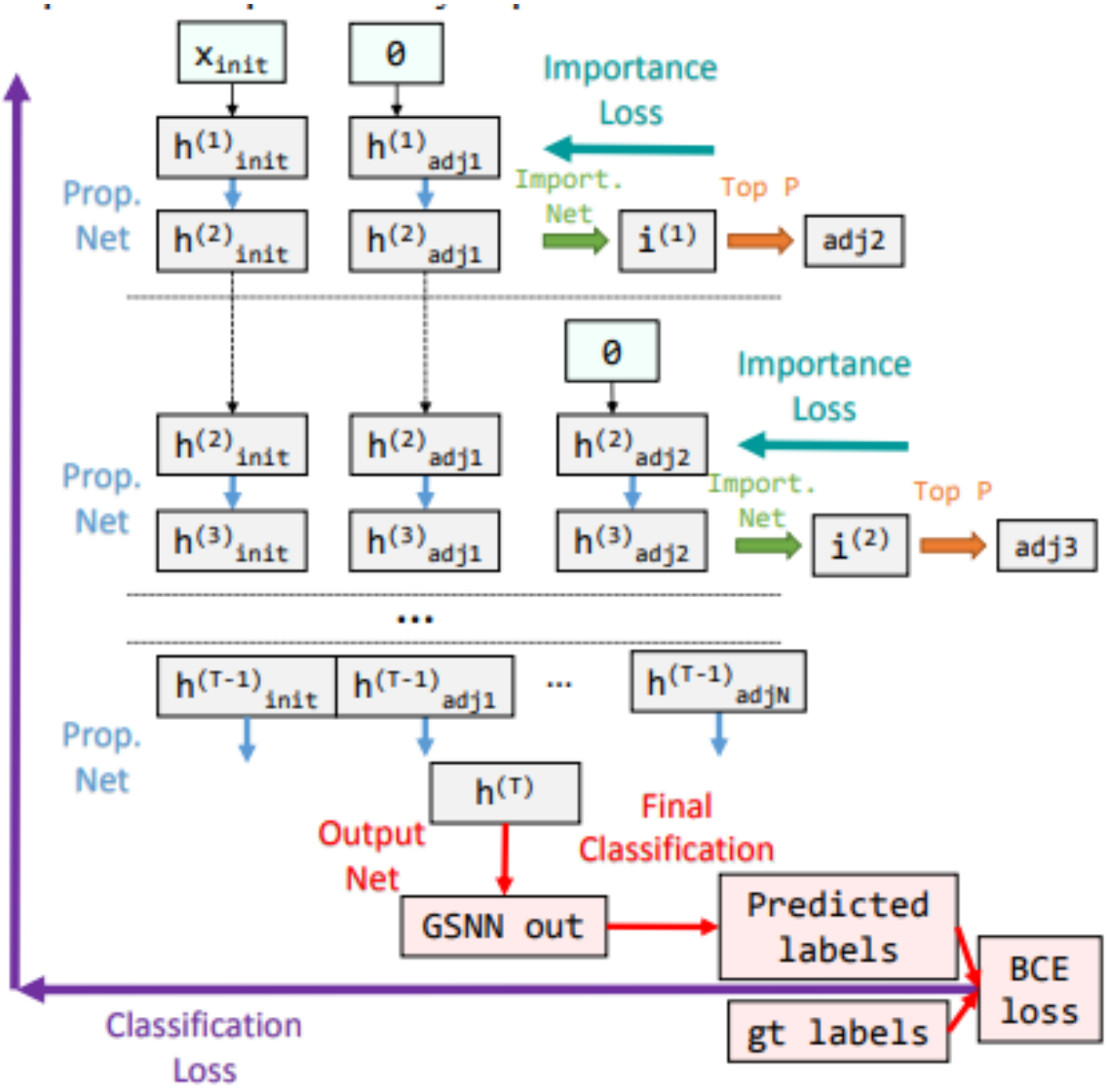

Рис. 5: Диаграмма нейронной сети для поиска по графику. Показывает инициализацию скрытых состояний, добавление новых узлов по мере расширения графа и поток потерь через выходные сети, сети распространения и важности.

3.3 Пайплайн изображений и базовые модели

Еще одна проблема, с которой можно столкнуться при адаптации графовых сетей для задач компьютерного зрения, заключается в том, как включить графовую сеть в конвейер изображений. Для классификации это довольно просто. Берутся выходные данные графовой сети, переупорядочиваем их так, чтобы узлы всегда появлялись в одном и том же порядке в итоговой сети, и заполняем нулями все узлы, которые не были расширены. Таким образом, если есть граф с 316 выходными узлами, и каждый узел предсказывает 5-мерную скрытую переменную,создаётся 1580-мерный вектор признаков на основе графа. Также объединяется этот вектор

признаков со слоем fc7 (4096-мерный) сети VGG-16 с тонкой настройкой [**VGG14**] и наивысшим баллом для каждой категории COCO, предсказанным Faster R-CNN (80-мерный). Этот вектор признаков стоимостью 5756 долларов затем передается в одноуровневую сеть окончательной классификации, обученную с помощью dropout.

Что касается исходных данных, то они сравниваются с: (1) Исходными данными VGG - ввод только fc7 в итоговую классификационную сеть; (2) Исходными данными обнаружения - ввод fc7 и лучшие результаты COCO в итоговую классификационную сеть.

3.4 Результаты

3.5 Наборы данных

Для экспериментов авторы статьи хотели протестировать набор данных, представляющий сложный, зашумлённый визуальный мир с множеством различных видов объектов, где метки потенциально неоднозначны и перекрываются, а категории распределены по длинному хвосту. Люди хорошо справляются с этой задачей, но алгоритмы компьютерного зрения всё ещё испытывают трудности. Поэтому авторы выбрали набор данных Visual Genome версии 1.0.

Visual Genome содержит более 100 000 естественных изображений из Интернета. Каждое изображение помечено объектами, атрибутами и связями между объектами, которые были добавлены аннотаторами. Аннотаторы могли добавлять любые объекты на изображении, а не только из заранее подготовленного списка, поэтому в результате появились тысячи меток объектов, некоторые из которых встречаются чаще, а большинство — гораздо реже. В среднем на изображении есть 21 размеченный объект, поэтому по сравнению с такими наборами данных, как ImageNet или PASCAL, рассматриваемые авторами сцены гораздо сложнее. Visual Genome также содержит информацию о взаимосвязях между объектами и атрибутами объектов, которую авторы используют для GSNN.

В экспериментах создается подмножество Visual Genome, которое называют набором данных Visual Genome с несколькими метками или VGML. В VGML берут 200 наиболее распространённых объектов из набора данных и 100 наиболее распространённых атрибутов, а также добавляют все категории COCO, которых нет в этих 300, в общей сложности 316 визуальных концепций. Задача — многоклассовая классификация: для каждого изображения предсказать, какое подмножество из 316 категорий

присутствует в сцене. Случайным образом делят изображения примерно на 80% для обучения и 20% для тестирования. Поскольку использовали предварительно обученные детекторы из COCO, можно убедиться, что ни одно из тестовых изображений не пересекается с обучающими изображениями детектора.

Также метод оценивался на более стандартном наборе данных COCO, чтобы показать, что подход полезен для нескольких наборов данных и что метод не зависит от графиков, построенных специально для наборов данных, используемых авторами статьи. Обучают и тестируют в режиме с несколькими метками и оценивают на мини-наборе.

3.6 Построение графа знаний

Visual Genome использовали в качестве источника для графа знаний. Используя только обучающую выборку, строили граф знаний, соединяя понятия с помощью наиболее распространённых связей между объектами и атрибутами в наборе данных. Было подсчитано, как часто в обучающем наборе встречалась связь между объектами или пара «объект-атрибут», и удалены все рёбра, у которых было менее 200 экземпляров. В результате был получен граф для всех изображений, в котором каждое ребро представляет собой распространённую связь. Идея состояла в том, что бы получить очень распространённые связи (например, трава зелёная или человек одет), но не редкие связи, которые встречаются только на отдельных изображениях (например, человек едет верхом на зебре).

Графы Visual Genome полезны для решения задачи, поскольку они содержат связи между объектами на уровне сцены, например, «человек одет в брюки» или «пожарный гидрант красный», и, таким образом, позволяют графической сети рассуждать о том, что находится на сцене. Однако они не содержат полезных семантических связей. Например, может быть полезно знать, что собака — это животное, если наша зрительная система видит собаку, а одна из наших меток — «животное». Чтобы решить эту проблему,создаётся версия графа, объединяя графы Visual Genome с WordNet. Используя подмножество WordNet ,сначала собирают новые узлы WordNet, не входящие в выходную метку, включая те, которые напрямую связаны с выходными метками и, следовательно, могут быть релевантными, и добавляем их в объединённый граф. Затем берутся все рёбра WordNet между этими узлами и добавляются их в объединённый граф.

3.7 Детали обучения

Обучаются все части конвейера (кроме детекторов). Все модели обучаются с помощью стохастического градиентного спуска, кроме GSNN, которая обучается с помощью ADAM. Используется начальная скорость обучения $0,05$, $0,005$ для сети VGG перед $fc7$, уменьшая её в 10 раз каждые 10 эпох, штраф L2 $1e^{-6}$ и импульс $0,5$. Авторы установили размер скрытого состояния GSNN равным 10, коэффициент дисконтирования важности γ равным $0,3$, количество временных шагов T равным 3, начальный порог достоверности равным $0,5$ и количество расширений P равным 5. Сети важности и выходные сети GSNN представляют собой однослойные сети с сигмоидальной активацией. Все сети обучались в течение 20 эпох с размером пакета 16.

3.8 Количественная оценка

В таблице 1 показаны результаты метода для многоклассовой классификации Visual Genome. В этом эксперименте объединённый граф Visual Genome и WordNet превосходит граф Visual Genome. Это говорит о том, что включение внешних семантических знаний из WordNet и выполнение явных рассуждений на графе знаний позволяет нашей модели изучать более эффективные представления по сравнению с другими моделями. В таблице 2 показаны результаты на многоклассовом наборе данных COCO. Gприрост производительности при использовании графа знаний более значителен, чем при использовании Visual Genome. Одно из возможных объяснений заключается в том, что граф знаний Visual Genome предоставляет важную информацию, которая помогает повысить производительность на самом наборе данных COCO. В предыдущем эксперименте с Visual Genome большая часть информации в графе содержится в самих метках и изображениях. Ещё один интересный результат заключается в том, что граф Visual Genome превосходит объединённый граф для COCO, хотя оба превосходят базовые модели. Одна из возможных причин заключается в том, что исходный граф VGML меньше, чище и содержит больше релевантной информации, чем объединённый граф. Кроме того, в эксперименте с VGML WordNet является новой внешней информацией для алгоритма, которая помогает повысить производительность.

Одной из возможных проблем является чрезмерная зависимость графа от набора из 80 детекторов COCO и начальных обнаружений. Поэтому авторы провели

Таблица 1: Средняя точность классификации по нескольким меткам в наборе данных Visual Genome с несколькими метками. Цифры для базового уровня VGG, базового уровня VGG с обнаружением, GSNN с использованием графа Visual Genome и GSNN с использованием комбинированного графа Visual Genome и WordNet.

Метод	mAP
VGG	30,57
VGG+Det	31,4
GSNN-VG	32,83
GSNN-VG+WN	**33**

Рис. 6: Средняя точность на Visual Genome при малом количестве данных. Показывает производительность всех методов для полного набора данных, 40 000, 20 000, 10 000, 5 000, 2 000, 1 000 и 500 обучающих примеров.

Таблица 2: Средняя точность для многоклассовой классификации на COCO. Цифры для базового уровня VGG, базового уровня VGG с обнаружением, GSNN с использованием графа Visual Genome и GSNN с использованием комбинированного графа Visual Genome и WordNet.

Метод	mAP
VGG	69,86
VGG+Det	73,93
GSNN-VG	**77,57**
GSNN-VG+WN	75,73

эксперимент по удалению, чтобы выяснить, насколько чувствителен наш метод к наличию всех начальных обнаружений. Повтор экспериментов с COCO на обоих графиках, используя два разных подмножества детекторов COCO. Первое подмножество - это только четные категории COCO, а второе - только нечетные категории. Из таблицы 3 видно, что методы GSNN снова превосходят базовые показатели.

Как можно было предположить, метод авторов не одинаково хорошо работает во всех категориях, а скорее лучше в одних категориях и хуже в других.

Таблица 3: Средняя точность для мультиклассовой классификации на COCO с использованием только четных и нечетных детекторов.

Метод	четная mAP	нечетная mAP
VGG+Det	71,87	71,73
GSNN-VG	73	73,43
GSNN-VG+WN	**73,59**	**73,97**

3.9 Возможные улучшения

Повысить качество работы метода на всех категориях,а не на каких-то отдельных.

4 Context-driven Visual Object Recognition based on Knowledge Graphs

4.1 Введение

То, как люди воспринимают реальный мир, сильно зависит от контекста. Особенно в ситуациях с низким качеством визуального ввода, например, вызванных большими расстояниями или коротким временем захвата, контекст, по-видимому, играет важную роль в повышении надежности распознавания. На восприятие влияют не только сопутствующие объекты или визуальные особенности на одном изображении, но также опыт и память. Есть свидетельства того, что люди воспринимают похожие изображения по-разному, учитывая данный контекст. Известным примером являются неоднозначные рисунки, как показано на рисунке 7. В зависимости от контекста, т.е. если это Пасха или Рождество, рисунок 1а может быть либо уткой, либо кроликом. Аналогичным образом, под влиянием социальных предубеждений собственного возраста, рисунок 1б может быть как молодой леди, так и пожилой женщиной. Люди классифицируют изображения на основе различных типов контекста. Известные категории основаны на визуальных признаках или семантических концепциях, но также могут быть основаны на другой информации, такой как атрибуты, описывающие их функцию. Соответственно, нейробиология показала, что человеческий мозг кодирует визуальный ввод в индивидуальные контекстуальные представления объектов, а именно визуальные, таксономические и функциональные. Конкретно, в



а) Утка или кролик? [25]

(б) Молодая леди или пожилая женщина? [1].

Рис. 7: Ментальное представление неоднозначных образов может меняться в зависимости от контекста, хотя воспринимаемый образ остается тем же самым

визуальном контексте изображения барабана и бочки имеют большое сходство, поскольку они имеют схожие визуальные характеристики. В таксономическом контексте барабан был бы похож на скрипку, поскольку они оба являются музыкальными инструментами. И в функциональном контексте барабан был бы похож на молоток, поскольку одно и то же действие удара может быть выполнено обоими объектами. В то время как существует много свидетельств того, что интеллектуальные машины также должны представлять отправленную информацию в контекстуализированных вложениях, глубокие нейронные сети (DNN) формируют свои объектные представления только на основе распределения признаков в наборе данных изображений. Следовательно, они не срабатывают, если объекты помещены в несоответствующий грубый контекст, которого не было на ранее просмотренных изображениях. В рамках данной статьи авторы исследуют следующие вопросы:

1. RQ1 — Может ли контекст, предоставляемый в виде графа знаний, влиять на обучение представлений изображений в глубокой нейронной сети, конечную точность и прогнозы изображений?

2. RQ2 — Может ли контекст помочь избежать критических ошибок в сценариях смены домена,когда глубокие нейронные сети дают сбой?

Чтобы стандартные глубокие нейронные сети могли создавать представления контекстуальных объектов авторы статьи представляют контекст,используя граф знаний (KG) и соответствующий ему эмбеддинг в графе знаний ($\vec{h}_{KG}$).Подобно тому, как это происходит в человеческом мозге, проводят эксперименты с тремя различными типами контекстов: визуальным контекстом, таксономическим контекстом и функциональным контекстом .Предлагается две версии внедрения знаний в DNN и сравнивается индукция различных контекстных моделей, количественно исследуя их контекстные пространства с помощью косинусного сходства, связанного с классами. Кроме того, подход оценивается количественно, сравнивая их конечную точность в задачах распознавания объектов в исходной и целевой областях, и предлагаем идеи и задачи. Структура данной статьи выглядит следующим образом:
- Рассматриваются три различных типа контекста и возможность моделирования этих представлений в графе контекстных знаний.
- Показаны два способа внедрения контекста в визуальную нейронную сеть.
- Эксперименты на семи наборах данных изображений в двух сценариях трансферного обучения.

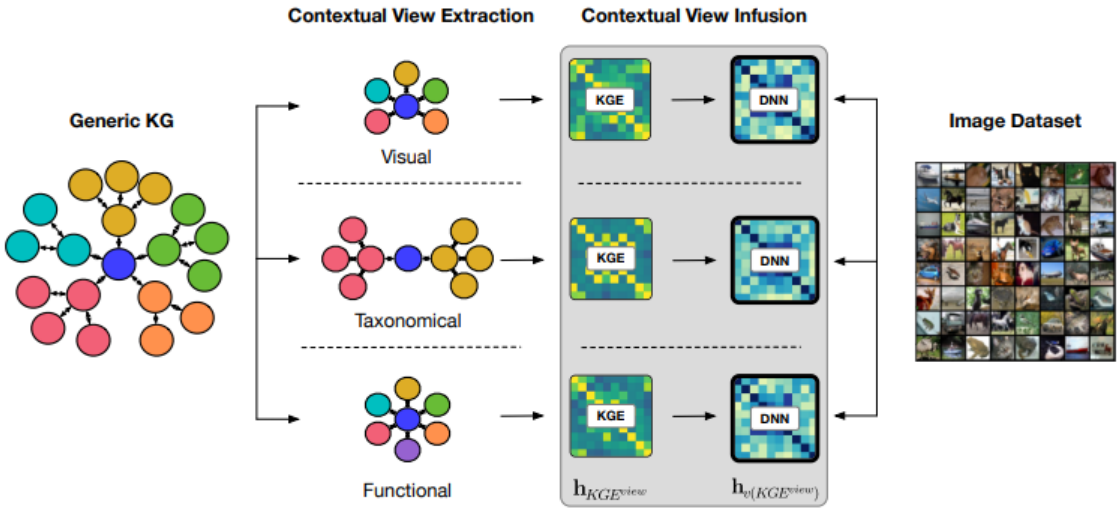

Рис. 8: Подход к изучению контекстных представлений изображений состоит из двух основных частей: 1) извлечение контекстного представления; 2) внедрение контекстного представления.

- Обобщение основных идей подхода авторов.

4.2 Типы контекстов

Структура, как показано на рисунке 8, состоит из двух основных частей: 1) извлечение контекстного представления, при котором из общего графа знаний извлекаются знания, относящиеся к задаче; и 2) внедрение контекстного представления, при котором контекстное представление внедряется в DNN.

Извлечение контекстных представлений Граф знаний может представлять собой базу знаний, закодированную с помощью богатой семантики в структуре графа. GKG, содержащий n контекстных представлений: представляет собой набор разнородных источников знаний, где каждое контекстное представление определяет конкретные взаимосвязи между закодированными классами.Однако для конкретной задачи может быть актуальна только определённая часть GKG. Таким образом, подграф, содержащий одно контекстное представление: или комбинация представлений извлекается из GKG. Поскольку модели распознавания объектов используются в реальном мире, который отличается от области их обучения, необходимо кодировать предварительные знания, которых нет в наборе данных. Основываясь на представлениях изображений в нашем мозге и на том, как люди склонны классифицировать объекты,авторы вводят три различных типа контекстных представлений, как по-

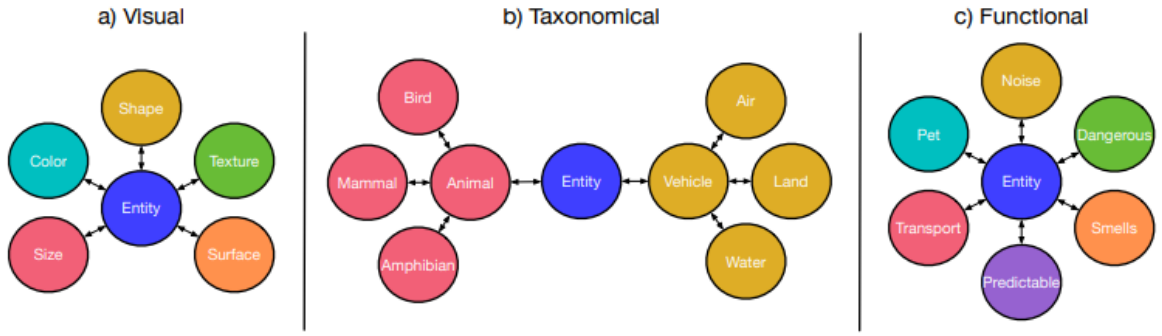

Рис. 9: Контекст может проявляться по-разному. Основываясь на представлениях о том, как люди воспринимают мир, мы представляем три контекстных представления общего графа знаний, а именно визуальное, таксономическое и функциональное представления.

казано на рисунке 9. Первое контекстное представление основано на визуальных, второе — на таксономических, а третье — на функциональных свойствах. Визуальный контекст. Визуальное представление (GKG^v) описывает визуальные свойства классов высокого уровня, например, свойства, описывающие цвет, форму или текстуру. Эти свойства могут присутствовать или не присутствовать в наборе данных изображений. Например, если все лошади в наборе данных белые, мы хотим указать, что лошади могут быть и других цветов. Таксономические контексты. Таксономическое представление (GKG^t) описывает взаимосвязи классов на основе иерархических схем. Таксономия создаётся экспертами и может содержать категории, основанные на концепциях из области биологии, места обитания, способа питания и т. д. Например, биологическая таксономия отделяет животных от транспортных средств и делит их на более мелкие подкатегории. Функциональный контекст. Функциональное представление (GKG^f) содержит свойства, описывающие функцию класса. Известно, что инструменты классифицируются в человеческом мозге по их функциям. В этом смысле такие свойства, как удар, трение или сверление, определяют категорию данного инструмента. Однако для расширения сферы применения можно ввести дополнительные функциональные свойства, такие как шум, транспортировка или запах.

Внедрение контекстуального представления При переносе знаний из GKG^{view} с помощью метода встраивания графа знаний (KGE) в эмбеддинг графа знаний:

$$\vec{h}_{KGE^{view}} = KGE(GKG^{view})$$

отношения, основанные на графах, преобразуются в пространственные отношения. Интуитивно понятно, что другой контекст приводит к другому представлению в векторном пространстве, где $\vec{h}_{KGE^{view}}$ отражает все отношения, которые моделируются в GKG^{view}. Как показано на рисунке 11, представлены два разных способа обучения визуальному контекстуальному встраиванию $\vec{h}_{v(GKG^{view})}$. Первый из них — $DNN_{KGE_u^{view}}$, который использует граф знаний в качестве обучающей выборки и, таким образом, обучает $\vec{h}_{KGE_u^{view}}$ без каких-либо дополнительных данных изображения. Второй способ — $DNN_{KGE_s^{view}}$, который использует граф знаний в качестве одноранговой сети и, таким образом, обучает $\vec{h}_{v(KGE_s^{view})}$ и $\vec{h}_{KGE_s^{view}}$ совместно с дополнительными данными изображения. В обоих способах используется потеря контрастности для выравнивания изображения, встраивающего $\vec{h}_{v(KGE^{view})}$ изображений $\vec{x}$ и DNN,

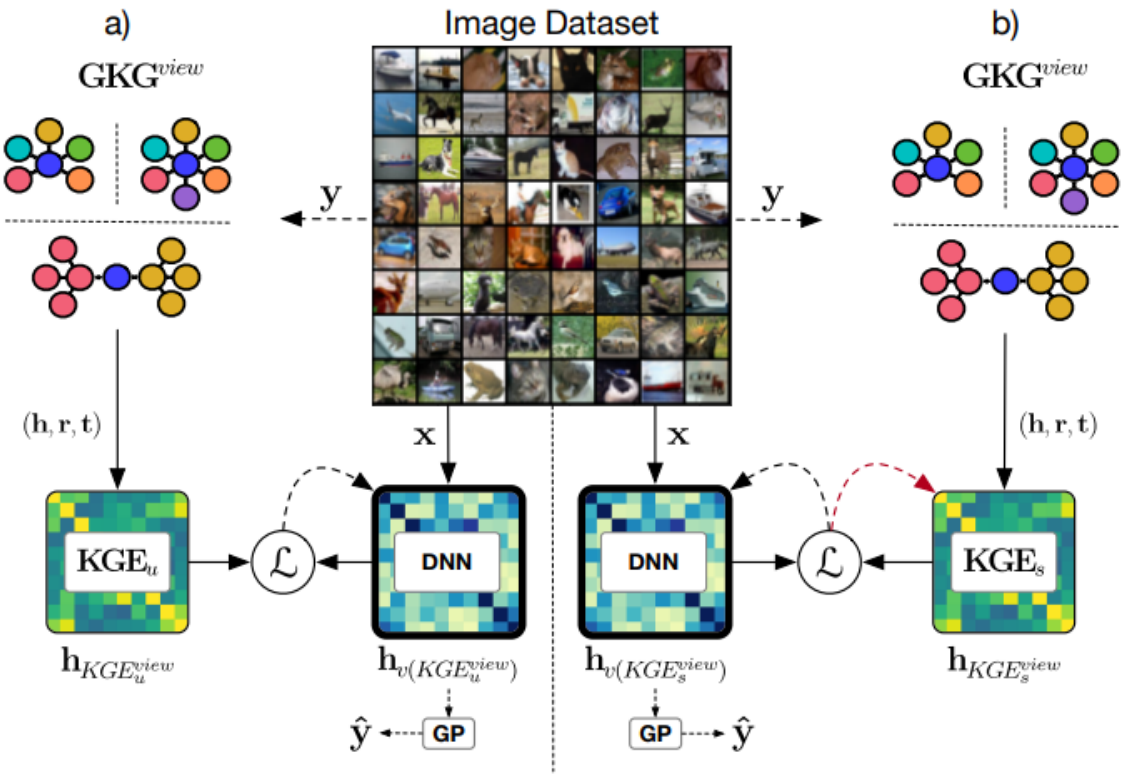

Рис. 10: Внедрение контекстуального представления

с графом знаний, встраивающим $\vec{h}_{KGE^{view}}$ информации о метке. Пакет состоит из N дополненных обучающих выборок

4.3 Предсказания

Чтобы предсказать метки классов неизвестных изображений, обычно обучают линейный слой (LL) или используют гауссовский процесс (GP) поверх $\vec{h}_{v(KGE^{view})}$. Для GP прогоняется весь обучающий набор данных через обученный DNN и вычисляем среднее значение и ковариационную матрицу для всех классов в $\vec{h}_{v(KGE^{view})}$. GP и LL вычисляются границы решений в $\vec{h}_{v(KGE^{view})}$ для всех классов набора данных. При выводе, когда цель состоит в том, чтобы предсказать метку класса для неизвестного изображения, GP или LL присваивают вероятность того, что изображение принадлежит к определённому классу. В качестве окончательного прогноза выбирается максимальная вероятность.

4.4 Эксперименты

Цель данного исследования — дать ответ на вопросы RQ1 и RQ2. Для этого проведены эксперименты с семью наборами данных в двух конкретных областях обобщения: Cifar10 и Mini-ImageNet. Для обоих экспериментов созданы отдельные GKG, включающие три различных контекстных представления: визуальное (GKG^v), таксономическое (GKG^t) и функциональное (GKG^f) представления, соответственно. Основываясь на структуре, описанной ранее, авторы используют GKG^{view} для обучения контекстной DNN в сочетании с данными изображений.Оцениваются и сравниваются обе версии подхода авторов:, $DNN_{KGE_u^{view}}$ и $DNN_{KGE_s^{view}}$.

Эксперименты проводились на следующих датасетах:

- Cifar10
- Mini-ImageNet и его производные:

Mini-ImageNet (Mini)

ImageNetV2 (V2)

ImageNet-Sketch (Sketch)

ImageNet-R (R)

ImageNet-A (A)

4.5 Детали реализации

Для обоих экспериментов используется реализация подхода автора. Из GKG извлекают различные представления GKG^{view} с помощью соответствующих запросов SPARQL. А Архитектура ResNet-18 используется в качестве бэкенда DNN с 128-мерным MLP в качестве головы. Обучают все конфигурации с помощью оптимизатора ADAM, с шагом обучения 0,001, без затухания весов и с помощью планировщика с косинусным отжигом с коэффициентом затухания обучения 0,1. Изображения дополняются с помощью случайного обрезания, случайного горизонтального переворота, дрожания цвета, случайного изменения яркости и уменьшения до 32x32 пикселей. Все модели обучаются в течение 500 эпох. Для а) $DNN_{KGE_u{}^{view}}$ авторы преобразуют GKG^{view} в векторное пространство с помощью графового автокодировщика (GAE), который обозначают как модель $DNN_{GAE^{view}}$. GAE авторов состоит из двух свёрточных слоёв с размером скрытого слоя 128. Авторы обучают GAE с помощью оптимизатора ADAM с шагом обучения 0,01 в течение 500 эпох. Для б) $DNN_{KGE_s^{view}}$ сеть внимания на графах (GAT) обучается в сочетании с данными изображений и называется моделью DNNGAT. GAT состоит из двух GAT-слоёв с 256 скрытыми измерениями, 8 головами и выходным измерением 128. Обучение выполняется с помощью той же контрастной потери на основе KG для изображений в дополнение к входным данным GKG^{view}.GAT оптимизируют с помощью оптимизатора ADAM с скоростью обучения 0,001 и без коррекции веса.

4.6 Результаты

В таблице 1 авторы сравнивают итоговую точность распознавания объектов контекстуальными DNN с базовым уровнем SupSSL. SupSSL — это та же модель, обученная с помощью контролируемой контрастной потери [**DBLP:conf/nips/KhoslaTWSTIMLK20**] и без вспомогательного контекста.

Можно заметить, что при различных контекстных вставках общая точность существенно не меняется. Для Cifar10 ΔDNN_{GAE^t} с $-7,0$ является наихудшей моделью, в то время как ΔDNN_{GAE^f} с $0,4$ является наилучшей моделью.

Можно заметить, что для каждого класса одна контекстная модель превосходит другие. Это также показывает, что контекст влияет на то, как DNN фокусируется на прогнозировании конкретного класса.

В таблице 1 показана относительная точность контекстных моделей на наборе

Results on Cifar10											
Cifar10	Airplane	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	All
SupSSL	95.1	97.0	91.8	83.9	92.9	85.7	96.0	93.5	96.8	95.9	92.9
ΔDNN_{GAE^v}	-1.2	**0.5**	-2.6	-0.2	**2.3**	-0.8	-0.2	-1.1	-0.5	-0.9	-0.5
ΔDNN_{GAE^t}	-0.9	-0.6	-1.2	-30.8	-29.8	-0.2	-2.2	-1.6	-1.5	-1.3	-7.0
ΔDNN_{GAE^f}	**1.0**	0.2	-1.1	**1.9**	0.1	0.6	**0.7**	1.2	-0.1	-0.4	**0.4**
ΔDNN_{GAE}	-0.7	0.0	-2.3	0.4	0.6	-0.6	-1.1	0.0	0.3	-1.8	-0.5
ΔDNN_{GAT^v}	-0.6	-0.3	**0.2**	0.3	0.1	-1.0	0.3	0.9	**0.7**	-0.8	-0.0
ΔDNN_{GAT^t}	-0.9	0.0	-1.7	1.8	0.1	1.0	0.4	0.5	-0.1	0.3	0.1
ΔDNN_{GAT^f}	-0.4	**0.5**	-3.0	1.7	1.5	-0.4	0.4	0.8	0.4	-0.1	0.1
ΔDNN_{GAT}	-1.0	0.3	-1.8	1.2	-0.3	**2.0**	-0.5	**1.7**	0.0	**0.7**	0.2
Results on Stl10											
Stl10	Airplane	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	All
SupSSL	85.4	86.9	82.4	56.6	91.5	60.5	-	76.5	84.5	74.1	77.6
ΔDNN_{GAE^v}	1.0	0.2	-2.6	**3.4**	**1.2**	-4.5	-	-4.8	-0.6	3.9	-0.3
ΔDNN_{GAE^t}	2.4	-1.0	-1.5	-10.1	-32.9	0.2	-	-0.5	-1.6	-1.6	-5.2
ΔDNN_{GAE^f}	1.9	-0.8	-1.3	1.4	-2.4	-0.5	-	**3.4**	0.9	3.3	**0.7**
ΔDNN_{GAE}	0.4	**0.5**	-1.9	1.8	-1.5	2.6	-	-1.4	-0.6	2.1	0.2
ΔDNN_{GAT^v}	0.5	-0.9	**2.6**	-0.6	-0.1	0.5	-	0.5	1.0	0.0	0.4
ΔDNN_{GAT^t}	1.0	-2.1	-0.5	1.9	-0.4	0.8	-	0.5	**1.6**	3.0	0.6
ΔDNN_{GAT^f}	**2.7**	-0.3	-1.5	-0.7	-1.0	-2.6	-	0.0	0.4	1.8	-0.1
ΔDNN_{GAT}	-1.6	-1.0	-2.6	-2.2	-1.2	**2.8**	-	3.1	1.2	**4.3**	0.3

Таблица 4: Сравнение точности отдельных классов для набора данных Cifar10 в качестве обучающей выборки и набора данных Stl10 в качестве тестовой выборки. Сравниваются контекстуальные представления обученных DNN с базовыми представлениями SupSSL .

данных Stl10. Обратите внимание, что модели обучаются только на данных Cifar10. Цель этого сценария обобщения в области применения — проверить устойчивость моделей. При оценке в целевой области можно заметить, что почти в каждой контекстуальной модели относительная точность повышается по сравнению с базовой моделью без знания контекста.

В сценариях, где меняется предметная область, можно наблюдать странные явления, происходящие таким образом, что модель со второй наихудшей производи-

ImageNet	Mini	V2	Sketch	R	A
SupSSL	58.6	43.0	20.3	4.3	1.2
ΔDNN_{GAE^v}	-0.3	0.0	-0.6	**0.2**	-0.2
ΔDNN_{GAE^t}	-19.6	-13.7	-8.8	-2.8	0.0
ΔDNN_{GAE^f}	-5.2	-3.3	-2.3	-0.7	**0.3**
ΔDNN_{GAE}	0.8	1.6	-0.6	-0.1	-0.1
black!30black	0.9	**2.3**	**0.2**	0.0	**0.3**
ΔDNN_{GAT^v}					
ΔDNN_{GAT^t}	**1.3**	0.6	0.1	0.1	0.0
ΔDNN_{GAT^f}	0.4	0.4	0.0	-0.1	-0.1
ΔDNN_{GAT}	0.5	0.6	0.1	0.0	0.0

Таблица 5: Сравнение моделей контекстного представления и их базовой версии SupSSL на Mini-ImageNet и его производных: Mini-ImageNet (Mini), ImageNetV2 (V2), ImageNet-Sketch (Sketch), ImageNet-R (R) и ImageNet-A (A).

тельностью DNN^t_{GAE} для класса воздушных судов в наборе данных Cifar10 является моделью со второй наилучшей производительностью для воздушных судов в Stl10. Однако для большинства классов видна тенденция к тому, что наиболее эффективная модель для класса в Cifar10, как правило, лучше работает и в целевой области. Как показано в таблице 5, DNN^t_{GAE} и DNN^f_{GAE} значительно уступают базовому $SupSSL$ и другим моделям с различными контекстными представлениями. В отличие от эксперимента с Cifar10, где наименее эффективная модель всего на 8% хуже базовой, в Mini-ImageNet наихудший результат составляет около 34%. Кроме того, видно, что DNN_{GAT^t} не страдает от ограничений, накладываемых GKG^t. Это открытие подтверждает наше предположение о том, что совместное обучение может ослабить ограничения GKG.

Результаты, представленные на рисунке 11, показывают, что для изображений, не получивших распространения, контекстуальные представления играют важную роль в получении обоснованных прогнозов. Идея в том, что некоторые различия в классах не так критичны, как другие. В этом смысле для некоторых задач некритично путать гуся с домовым вьюрком, поскольку они оба принадлежат к семейству птиц, однако спутать музыкальный инструмент (гобой) с собакой (маламутом) может привести к проблемам. Можно заметить, что DNN_{GAE} (GAE) и DNN_{GAT} (GAT) не обязательно предсказывают одно и то же изображение на основе данного контекста.



Рис. 11: Контекстные прогнозы DNN_{GAE} (GAE) и DNN_{GAT} (GAT) и их контекстный вид на Mini-ImageNet. Контекстный вид влияет на представление изображения и, следовательно, на окончательный прогноз для одного и того же входного изображения.

4.7 Возможные улучшения

Авторы указывают следующие области для будущей работы:

1. Динамические графы: Возможность адаптации графов знаний в реальном времени на основе контекста или новых данных.

2. Расширение данных: Использование дополнительных источников (включая пользовательские аннотации и текстовые данные) для создания более полных графов.

3. Оптимизация: Уменьшение вычислительных затрат, связанных с обработкой больших графов.

4. Мультимодальные данные: Включение дополнительных модальностей (например, видео или текст), чтобы дополнить визуальные и семантические данные.

Список литературы

0. *Kenneth Marino Ruslan Salakhutdinov A. G.* The More You Know: Using Knowledge Graphs for Image Classification // arXiv:1612.04844. — 2017.

0. *Sebastian Monka L. H.*, *Rettinger A.* Context-driven Visual Object Recognition based on Knowledge Graphs // arXiv:2210.11233. — 2022.

0. Nuo Xu, Wen Wang, Rong Yang, Mengjie Qin, Zheyuan Lin, Wei Song, Chunlong Zhang, Jason Gu, Chao Li // arXiv:2402.18892. — 2024.