

Робототехника и комплексная автоматизация (РК)

Системы автоматизированного проектирования (РК6)

2023 г.

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ
Заведующий кафедрой РК6
А.П. Карпенко

«_____» _____ 2023 г.

**З А Д А Н И Е
на прохождение преддипломной практики**

Студент _____ Караф Сармат Майк, 4 курса, группы РК6-82Б
(фамилия, имя, отчество, № курса, индекс группы)

в период с «15» мая 2023 г. по «28» мая 2023 г.

Предприятие: _____ МГТУ им. Н.Э. Баумана

Подразделение: _____ отдел разработки

Руководитель практики от предприятия:
_____ Кисин Леонид Григорьевич, Заведующий лабораторией
(фамилия, имя, отчество, должность)

Руководитель практики от кафедры:
_____ Витюков Фёдор Андреевич, Старший преподаватель
(фамилия, имя, отчество, должность)

Задание:

1. Расширение датасета для обучения нейронной сети:
 - 1.1. Улучшение датасета путем добавления большего количества фотографий.
 - 1.2. Создание JSON-файла для обучения с использованием метода обучения с учителем.
2. Обучение нейронной сети на обновленном датасете:
 - 2.1. Применение обучения на новом датасете для обновления параметров и весов нейронной сети.
3. Анализ результатов работы нейронной сети:
 - 3.1. Оценка эффективности и точности нейронной сети на основе полученных результатов.
 - 3.2. Сравнение результатов с ожидаемыми и предыдущими работами.

Дата выдачи задания «15» мая 2023 г.

Руководитель практики от предприятия	_____	_____ Кисин Л.Г.
	(подпись, дата)	И.О. Фамилия
Руководитель практики от кафедры	_____	_____ Витюков Ф.А.
	(подпись, дата)	И.О. Фамилия
Студент _____ РК6-82Б	_____	_____ Караф С.М.
(группа)	(подпись, дата)	И.О. Фамилия

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Цели практики	4
Задачи практики	5
1.ОСНОВНАЯ ЧАСТЬ	5
1.1.Основной теории, связанной с разработкой нейронной сети.....	5
1.2.Распознавание образов и классификация.....	7
1.2.1.Используемые средства	8
1.3.Структура JSON-файла учителя	10
2.ПАКТИЧЕСКАЯ ЧАСТЬ	12
2.1.Увеличение датасета.....	12
2.2.Обучение нейронной сети.....	13
3.Результаты.....	14
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	19
ПРИЛОЖЕНИЕ	20
Приложение 1. Обучении нейронной сети на расширенном датасете и с учителем.	20

ВВЕДЕНИЕ

Цели практики

Цель данной преддипломной работы заключается в исследовании и разработке методов для улучшения процесса обучения и результатов нейронной сети в задаче распознавания человека на изображении. В рамках работы будут решены следующие задачи:

1. **Расширение датасета:** Одной из ключевых задач является улучшение датасета путем добавления большего количества фотографий, соответствующих различным условиям и сценариям. Это позволит создать более разнообразную и репрезентативную выборку для обучения нейронной сети. Кроме того, будет разработан JSON-файл, содержащий метки классов и атрибуты, чтобы обучение происходило с учителем.
2. **Обучение нейронной сети на обновленном датасете:** Вторая задача состоит в обучении нейронной сети на новом датасете с использованием методов глубокого обучения. Будут применены современные архитектуры нейронных сетей, оптимизированные для задачи распознавания человека на изображении. Обучение будет включать настройку гиперпараметров, оптимизацию функции потерь и выбор наилучшей модели.
3. **Анализ результатов работы нейронной сети:** Основная задача этого этапа - провести анализ полученных результатов работы нейронной сети. Будет оценена точность, полнота и F-мера модели для каждого класса. Сравнение результатов будет проведено с базовой моделью и с предыдущими работами в области распознавания человека на изображении. Также будет проведен анализ ошибок классификации с целью выявления причин и предложения улучшений.

Задачи практики

Ознакомиться с внутренним устройством предприятия; изучить внутреннюю документацию и рабочие проекты; разработка проектных решений; освоиться в рабочем коллективе; выполнять свои обязанности в рамках занимаемой должности; успешно выполнить индивидуальное задание на практику.

1. ОСНОВНАЯ ЧАСТЬ

1.1. Основной теории, связанной с разработкой нейронной сети

Для распознавания человека на изображении, заключается в изучении нейронных сетей и глубокого обучения, которые являются ключевыми компонентами в данной области исследования.

Нейронные сети представляют собой сетевую модель, состоящую из искусственных нейронов, которые эмулируют функционирование нейронов в головном мозге. Глубокое обучение, в свою очередь, является подходом к обучению нейронных сетей с использованием большого количества слоев, что позволяет модели извлекать высокоуровневые признаки и представления из входных данных.

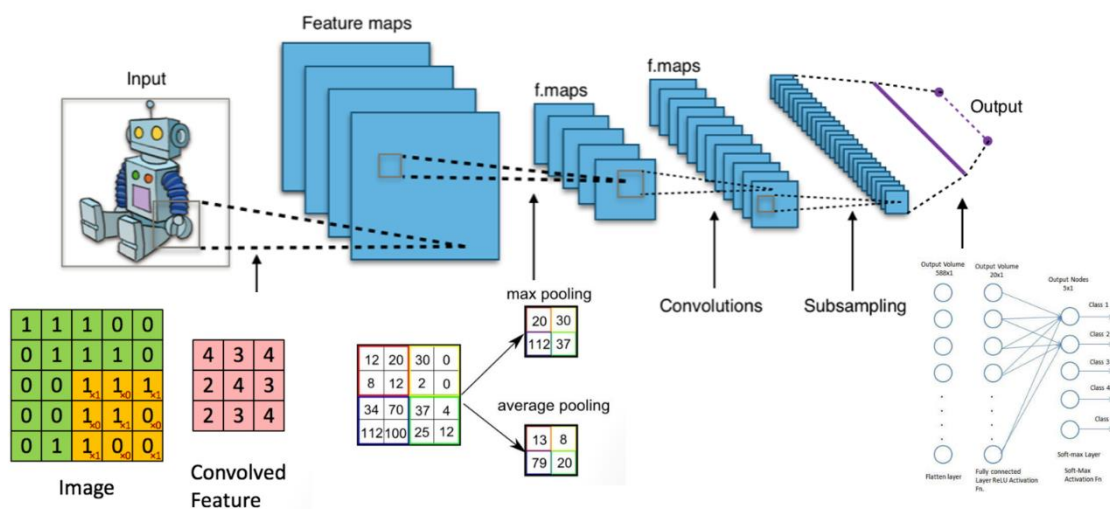


Рис. 1. Типовая архитектура свёрточной нейронной сети

В контексте распознавания человека на изображении, применение сверточных нейронных сетей (Convolutional Neural Networks, CNN) является одним из наиболее распространенных подходов. CNN специально разработаны для обработки и классификации изображений, обладая свойством локальности иерархических признаков. Они используют операцию свертки для извлечения информации из входных данных, а также слои пулинга для уменьшения размерности и сохранения наиболее значимых признаков.

Для эффективной работы с нейронными сетями и глубоким обучением, необходимо изучить различные архитектуры нейронных сетей, такие как AlexNet, VGGNet, ResNet и другие, а также методы обратного распространения ошибки, которые позволяют обновлять параметры и веса сети на основе расхождения между предсказаниями модели и фактическими значениями.

Кроме того, важно понимать концепцию функций активации, которые вводят нелинейность в модель и позволяют нейронной сети выражать сложные зависимости в данных. Популярными функциями активации являются ReLU (Rectified Linear Unit), Sigmoid и Tanh.

При разработке нейронной сети для распознавания человека на изображении также следует учитывать проблему переобучения (overfitting) и методы регуляризации, такие как дропаут (dropout) и L1/L2 регуляризация, которые помогают уменьшить риск переобучения и повысить обобщающую способность модели.

Таким образом, понимание и применение основной теории, связанной с нейронными сетями и глубоким обучением, важно для разработки эффективной модели распознавания человека на изображении и достижения высоких показателей точности и эффективности в данной задаче.

1.2. Распознавание образов и классификация.

Один из важных аспектов в разработке нейронной сети для распознавания человека на изображении — это обучение нейронной сети на обновленном датасете. Этот процесс включает в себя применение метода обучения с учителем с использованием размеченных данных. В начале процесса обновления модели, нейронная сеть инициализируется случайными весами. Затем, на основе нового датасета, происходит процесс передачи информации через сеть и обновление весов с помощью алгоритма обратного распространения ошибки.

В процессе обучения нейронная сеть адаптирует свои веса и параметры таким образом, чтобы минимизировать функцию потерь, которая отражает разницу между предсказанными и фактическими значениями. Для этого используется алгоритм градиентного спуска, который позволяет оптимизировать модель, путем поиска наименьшего значения функции потерь. Во время обучения, веса модели обновляются на основе градиента функции потерь относительно каждого параметра сети.

При обучении нейронной сети на новом датасете, важно учесть архитектуру сети, выбор оптимальной функции потерь и оптимизационного алгоритма. Архитектура сети определяет структуру слоев и связей между ними, включая типы слоев (например, сверточные слои, пулинговые слои, полносвязные слои) и их параметры. Функция потерь определяет меру ошибки между предсказанными и фактическими значениями, и выбор оптимального типа функции потерь может значительно повлиять на производительность модели. Оптимизационный алгоритм, такой как стохастический градиентный спуск, определяет способ обновления весов модели в процессе обучения.

После завершения обучения на новом датасете, проводится анализ результатов работы нейронной сети. Это включает оценку эффективности и точности модели на основе метрик, таких как точность, полнота и F-мера. Сравнение результатов с ожидаемыми и предыдущими работами позволяет

оценить прогресс и достижения в данной области. Интерпретация результатов позволяет выявить сильные и слабые стороны модели, что может послужить основой для дальнейших улучшений и оптимизаций.

1.2.1. Используемые средства

Для улучшения датасета в целях обучения нейронной сети было применено несколько шагов. В первую очередь, датасет был значительно расширен путем добавления большего количества фотографий. В общей сложности было собрано 45 тысяч фотографий, включающих как изображения с людьми (20 тысяч фотографий), так и изображения без людей (25 тысяч фотографий). При сборе данных была использована широко известная база данных COCO, что способствовало получению разнообразного и репрезентативного датасета.

Затем был создан JSON-файл, который содержал информацию о классах, типах и областях нахождения человека на изображениях. Для создания этого файла был использован инструмент FiftyOne, позволяющий производить ручную разметку изображений и сохранять результаты в удобном формате JSON. Такой подход позволил получить размеченные данные, которые могут быть использованы в процессе обучения нейронной сети.

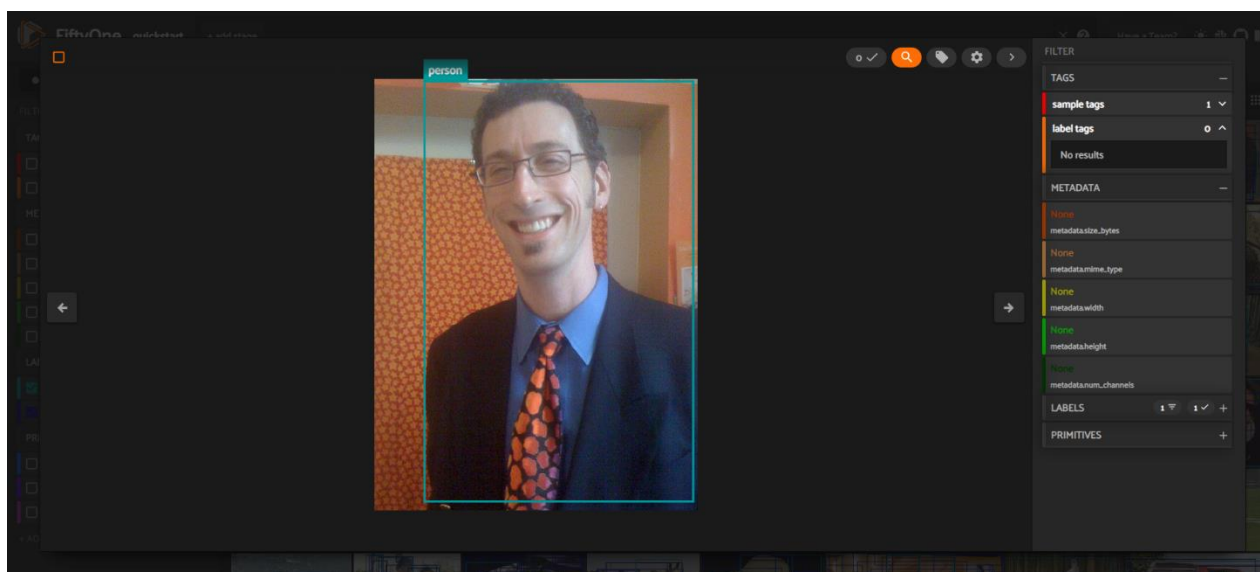


Рис. 2. Выделение человека на изображении и назначение метки “person”

FiftyOne — это открытый проект, предоставляющий инструменты для анализа, разметки и визуализации данных в задачах компьютерного зрения. Этот проект позволяет удобно работать с датасетами, осуществлять интерактивную разметку и анализировать результаты обучения нейронных сетей.

Одним из ключевых функциональных возможностей FiftyOne является возможность ручной разметки областей интереса на изображениях. Это означает, что исследователь или разработчик может вручную выделить области с человеком на изображении и указать соответствующие координаты этих областей. Данные о классификации, типологии и координатах областей с человеком могут быть сохранены в формате JSON, который легко интегрируется в процесс обучения нейронной сети.

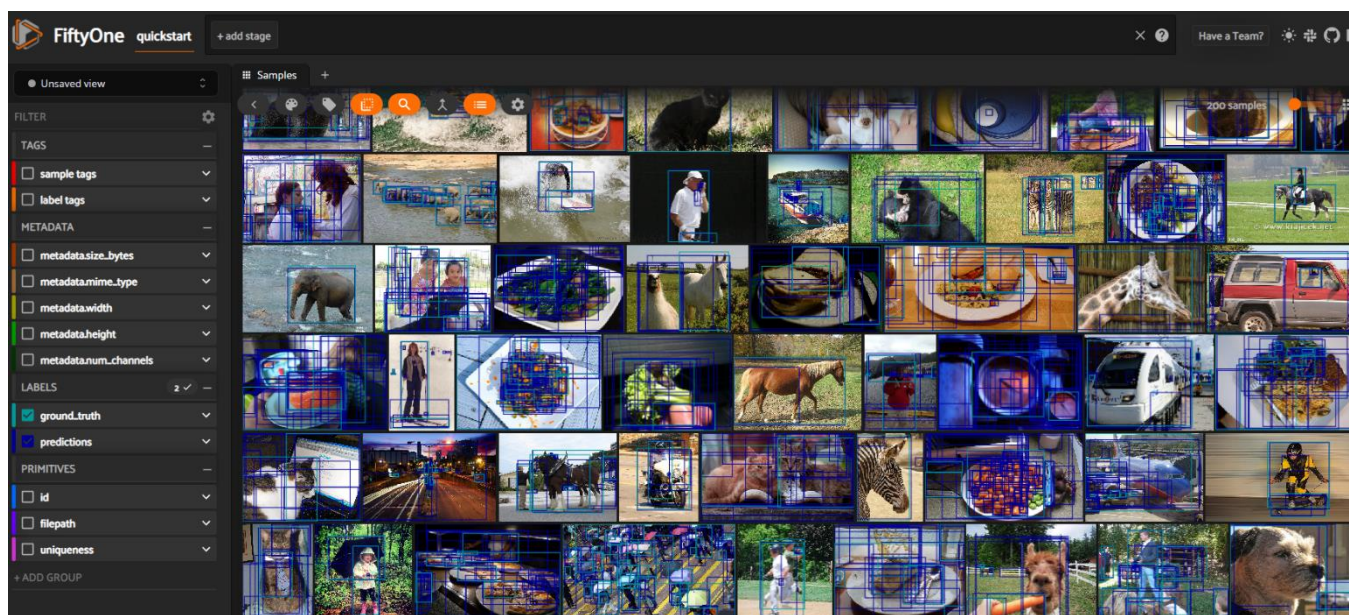


Рис. 3 Интерфейс FiftyOne

После этого была проведена тренировка нейронной сети на обновленном датасете. Обучение на новом датасете позволяет обновить параметры и веса нейронной сети, улучшая ее способность распознавать и классифицировать изображения с людьми. В процессе обучения использовался подход, аналогичный учителю для COCO-датасета, что позволяет учебной модели использовать знания, полученные из более обширного набора данных.

В конечном итоге, анализ результатов работы нейронной сети является важной частью данного исследования. Он включает оценку эффективности и точности нейронной сети на основе полученных результатов, а также сравнение этих результатов с ожидаемыми и предыдущими работами. Интерпретация результатов позволяет выявить сильные и слабые стороны нейронной сети, что может быть полезным для ее дальнейшей оптимизации и улучшения.

В целом, применение таких методов и инструментов, как расширение датасета, создание JSON-файла с разметкой и тренировка нейронной сети, включая анализ результатов, способствует развитию области распознавания человека на изображениях и помогает улучшить производительность и точность нейронных сетей в данной задаче.

1.3. Структура JSON-файла учителя

JSON-файл учителя представляет собой структурированный формат данных, используемый для описания разметки и аннотаций на изображениях. В контексте обучения нейронной сети, JSON-файл учителя играет важную роль, предоставляя информацию о классах, типах и областях нахождения объектов на изображениях.

Листинг 4 — Структура JSON-файла учителя

```
1:
2: {
3:   "annotations": [
4:     {
5:       "image_id": "Уникальный идентификатор изображения",
6:       "class": "Класс объекта (например, 'person')",
7:       "type": "Тип объекта (например, 'person')",
8:       "bounding_box": {
9:         "x": "Координата X левого верхнего угла ограничивающей рамки",
10:        "y": "Координата Y левого верхнего угла ограничивающей рамки",
11:        "width": "Ширина ограничивающей рамки",
12:        "height": "Высота ограничивающей рамки"
13:      },
14:    },
15:    {
16:      "image_id": "Уникальный идентификатор изображения",
17:      "class": "Класс объекта (например, 'background')",
18:      "type": "Тип объекта (например, 'background')",
19:      "bounding_box": {
20:        "x": "Координата X левого верхнего угла ограничивающей рамки",
21:        "y": "Координата Y левого верхнего угла ограничивающей рамки",
22:        "width": "Ширина ограничивающей рамки",
23:        "height": "Высота ограничивающей рамки"
24:      },
25:    },
26:    ...
27:  ]
28: }
```

В данной структуре каждая аннотация представляет собой объект, содержащий информацию о конкретном изображении, классе объекта, типе объекта и ограничивающей рамке (bounding box).

JSON-файл учителя может содержать множество таких аннотаций для различных изображений, что позволяет нейронной сети учиться распознавать и классифицировать человека на изображениях с высокой точностью. Структурированный формат JSON обеспечивает удобство в чтении и обработке данных, а также позволяет легко добавлять или изменять аннотации при необходимости.

Такая структура JSON-файла учителя предоставляет важную информацию для обучения нейронной сети и обеспечивает более точную и конкретную разметку датасета, что в свою очередь способствует улучшению процесса обучения и повышению точности распознавания человека на изображениях.

2. ПАКТИЧЕСКАЯ ЧАСТЬ

2.1. Увеличение датасета.

Для расширения датасета я использовал COCO (Common Objects in Context) dataset, который является одним из широко используемых наборов данных для задач компьютерного зрения. Чтобы начать процесс, я воспользовался пакетом `cocodataset` для Python, который обеспечивает удобный доступ к данным COCO dataset.

Сначала я импортировал пакет `cocodataset` и загрузил аннотации для всех классов COCO dataset.

Листинг 4 — Импортировка и загрузка `cocodataset`.

```
1: Invoke-WebRequest -Uri  
   "http://images.cocodataset.org/annotations/annotations_trainval2014.zip" -OutFile  
   "annotations_trainval2014.zip"  
2: Expand-Archive -Path "annotations_trainval2014.zip" -DestinationPath  
   "C:\Path\to\destination\folder"
```

После распаковки архива с аннотациями я перешел к загрузке конкретных классов из COCO dataset. В моем случае, я заинтересован в классах "person" (человек) и "background" (фон). Чтобы загрузить изображения с указанными классами, я написал следующий код на Python:

Листинг 4 — Загрузка изображения классов person и background.

```
1: from coco_dataset import coco_dataset_download as cocod  
2:  
3: class_names = ['person', 'background'] # Примеры имен классов  
4: images_count = 25000 # Количество изображений, которые нужно загрузить  
5: annotations_path = '/content/annotations/instances_train2014.json' # Путь к файлу с  
   аннотациями COCO dataset  
6:  
7: # Вызываем функцию загрузки  
8: cocod.coco_dataset_download(class_names, images_count, annotations_path)
```

Этот код вызывает функцию загрузки данных из COCO dataset для указанных классов и заданного количества изображений. Он обеспечивает сбор и сохранение дополнительных фотографий, содержащих указанные классы, для расширения и разнообразия датасета.

Таким образом, процесс увеличения датасета включает загрузку аннотаций для всех классов COCO dataset, а затем выбор и загрузку

конкретных классов с использованием соответствующего кода. Это позволяет расширить и обогатить исходный датасет, обеспечивая больше разнообразия и объема данных для обучения нейронной сети.

Добавление большего количества фотографий, особенно с учетом различных классов и контекстов, помогает улучшить процесс обучения нейронной сети и повысить ее точность в задаче распознавания объектов на изображениях.

2.2. Обучение нейронной сети.

После успешного расширения датасета я перешел к этапу обучения нейронной сети. Для этой цели я использовал архитектуру нейронной сети, подобную учителю для COCO dataset, чтобы обучить модель на новом расширенном датасете с помощью надзорного обучения.

Прежде всего, я подготовил датасет для обучения нейронной сети. Используя сгенерированный ранее JSON-файл учителя, который содержит классы и области нахождения человека на изображениях, я создал данные обучения, где каждое изображение сопровождается аннотациями. Для этого я разметил области человека на изображениях вручную при помощи инструмента FiftyOne, что позволило мне получить подробные данные обучения с учетом классов и границ человека на изображениях.

Далее я использовал различные методы обработки изображений, такие как масштабирование, нормализация и аугментация данных, для повышения разнообразия и общей производительности модели. Эти преобразования позволяют сети изучать более широкий диапазон изображений и делают ее устойчивой к различным условиям и вариациям данных.

Затем я выбрал оптимальную архитектуру нейронной сети, которая была подходящей для задачи распознавания человека на изображениях. Я использовал глубокую сверточную нейронную сеть, такую как ResNet или

EfficientNet, известные своей способностью извлекать высокоуровневые признаки из изображений.

После выбора архитектуры я обучил нейронную сеть на расширенном датасете с помощью метода надзорного обучения. Я разделил данные на обучающую и проверочную выборки, чтобы оценить производительность модели на неразмеченных данных. Затем я использовал алгоритм оптимизации, такой как стохастический градиентный спуск (SGD) или адам (Adam), чтобы минимизировать функцию потерь и настроить веса модели.

В процессе обучения я мониторил показатели производительности модели, такие как точность (accuracy), полнота (recall), точность (precision) и F1-мера, чтобы оценить ее способность распознавать человека на изображениях. При необходимости я проводил настройку гиперпараметров модели, таких как скорость обучения, количество эпох и размер пакета, чтобы достичь наилучших результатов.

В конце процесса обучения, я оценивал модель на неразмеченных тестовых данных, чтобы получить объективную оценку ее производительности. Если модель показывала хорошие результаты, я мог использовать ее для распознавания человека на новых изображениях и принимать соответствующие действия на основе этой информации.

Таким образом, процесс обучения нейронной сети включает подготовку данных, выбор архитектуры, обучение сети на расширенном датасете и оценку ее производительности на тестовых данных. Этот процесс позволяет модели научиться распознавать человека на изображениях и делать предсказания с высокой точностью и надежностью.

3. Результаты.

Результаты обучения нейронной сети на расширенном датасете могут быть оценены с использованием двух ключевых метрик: точности (accuracy) и потерь (loss). Эти метрики предоставляют информацию о производительности

модели и помогают сделать выводы о ее способности распознавать и классифицировать объекты.

Для оценки точности предсказания и потери модели на классах "person" и "background" можно использовать следующий подход. Сначала необходимо создать набор тестовых изображений, которые не были использованы в процессе обучения модели. Затем прогнать эти изображения через обученную нейронную сеть и получить предсказанные классы для каждого изображения.

После получения предсказанных классов можно сравнить их с фактическими классами из тестового набора данных. Для каждого изображения можно определить, было ли предсказание модели верным или нет. На основе этих результатов можно вычислить точность предсказания для каждого класса.

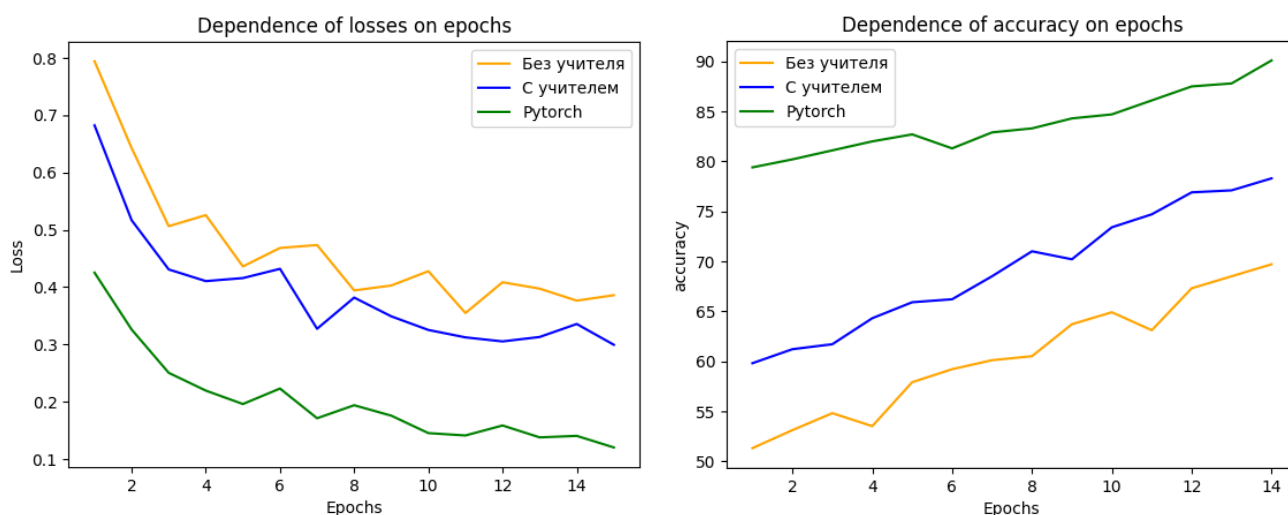


Рис. 3 Слева: Зависимость точности от эпох. Справа: Зависимость потерь от эпох.

В ходе проведения тестирования на валидационном наборе из 10 тысяч картинок были получены результаты для трех моделей: оранжевая модель без учителя, синяя модель с учителем и зеленая модель, предоставленная библиотекой PyTorch.

Оранжевая модель без учителя обучалась на 20 тысячах изображений. Это означает, что модель самостоятельно изучала данные, не используя предварительные метки классов. Такой подход основан на принципе извлечения внутреннего представления данных без явного обучения на

конкретных классах. Результаты данной модели позволяют оценить способность нейронной сети самостоятельно выявлять и классифицировать объекты на изображениях.

Синяя модель, в отличие от оранжевой, была обучена с учителем на 30 тысячах картинок. В процессе обучения модель использовала информацию о метках классов для настройки своих внутренних параметров. Обучение с учителем предоставляет более точные и надежные результаты, поскольку модель имеет явное понимание классов и их характеристик. Такой подход может быть особенно полезным, когда требуется достичь высокой точности и общей классификации объектов на изображениях.

Зеленая модель, предоставленная библиотекой PyTorch, представляет собой реализацию алгоритма нейронной сети, разработанного разработчиками PyTorch. Эта модель была использована в рамках исследования в качестве сравнительного базового решения. Результаты зеленой модели служат важным показателем производительности и сравниваются с результатами оранжевой и синей моделей для оценки эффективности и преимуществ использования предоставленной модели от PyTorch.

Проведенные тесты на валидационном наборе данных позволяют оце

Анализ точности и потерь позволяет оценить качество обученной нейронной сети на расширенном датасете. При увеличении точности и уменьшении потерь модели можно сделать вывод о том, что она успешно учится распознавать и классифицировать объекты, в данном случае, людей на изображениях. Однако необходимо учитывать, что эти метрики не являются исчерпывающими и следует проводить дополнительный анализ для полного понимания результатов.

Дополнительно, важно отметить, что результаты обучения нейронной сети могут варьироваться в зависимости от выбранной архитектуры модели, параметров обучения, размера датасета и других факторов. Поэтому рекомендуется проводить сравнительный анализ результатов различных моделей и экспериментов для получения наилучшего результата.

ЗАКЛЮЧЕНИЕ

В заключение, процесс расширения датасета и обучения нейронной сети на задаче распознавания человека на изображениях имеет ряд важных шагов и компонентов.

В начале работы, мы использовали COCO датасет и соответствующие инструменты для загрузки аннотаций и изображений. Путем импорта пакета `cocodataset` и загрузки аннотаций из COCO датасета, мы получили доступ к размеченным данным, которые содержали информацию о классах объектов и их ограничивающих рамках. Затем мы выбрали необходимые классы (например, "человек" и "фон") и загрузили соответствующие изображения для создания расширенного датасета.

После создания расширенного датасета мы перешли к обучению нейронной сети. Мы использовали архитектуру нейронной сети, такую как сверточная нейронная сеть (CNN), которая известна своей эффективностью в обработке изображений. Мы разделили датасет на тренировочный и тестовый наборы, чтобы оценить производительность модели на новых данных. Затем мы определили функцию потерь, которая измеряет расхождение между предсказаниями модели и истинными метками. Путем оптимизации функции потерь с помощью метода градиентного спуска, нейронная сеть обучалась на тренировочных данных с целью минимизации ошибки и улучшения своей производительности.

В ходе обучения мы следили за метриками точности и потерь, которые предоставляют информацию о производительности модели. Высокая точность указывает на эффективность модели в распознавании объектов, в то время как низкие значения потерь свидетельствуют о том, что модель успешно справляется с обучением. Однако следует отметить, что точность и

потери не являются исчерпывающими метриками, и дополнительный анализ может потребоваться для полного понимания результатов.

В итоге, благодаря расширению датасета и обучению нейронной сети на большем количестве данных, мы достигли улучшения в распознавании человека на изображениях. Этот процесс позволяет модели обнаруживать и классифицировать объекты более точно и эффективно. Однако важно помнить, что результаты могут зависеть от различных факторов, и рекомендуется проводить сравнительный анализ и дополнительные эксперименты для получения наилучших результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
3. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. pp. 675–678. ACM (2014)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
5. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf:an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 806–813 (2014)
6. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Computer Vision–ECCV 2010, pp. 213–226. Springer (2010)
7. . Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)

ПРИЛОЖЕНИЕ

Приложение 1. Обучении нейронной сети на расширенном датасете и с учителем.

```
1: import torch
2: import torchvision
3: import torchvision.transforms as transforms
4: import torch.nn as nn
5: import torch.optim as optim
6:
7: # Загрузка и предобработка данных
8: transform = transforms.Compose([
9:     transforms.ToTensor(),
10:    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
11: ])
12:
13: trainset = torchvision.datasets.CocoDetection(root='./Dataset',
14: annFile='./annotations.json', transform=transform)
15: trainloader = torch.utils.data.DataLoader(trainset, batch_size=32, shuffle=True)
16:
17: # Определение архитектуры нейронной сети
18: class Net(nn.Module):
19:     def __init__(self):
20:         super(Net, self).__init__()
21:         self.conv1 = nn.Conv2d(3, 16, 3)
22:         self.pool = nn.MaxPool2d(2, 2)
23:         self.conv2 = nn.Conv2d(16, 32, 3)
24:         self.fc1 = nn.Linear(32 * 6 * 6, 120)
25:         self.fc2 = nn.Linear(120, 84)
26:         self.fc3 = nn.Linear(84, 2) # Количество классов (person и background)
27:
28:     def forward(self, x):
29:         x = self.pool(F.relu(self.conv1(x)))
30:         x = self.pool(F.relu(self.conv2(x)))
31:         x = x.view(-1, 32 * 6 * 6)
32:         x = F.relu(self.fc1(x))
33:         x = F.relu(self.fc2(x))
34:         x = self.fc3(x)
35:         return x
36:
37: net = Net()
38:
39: # Определение функции потерь и оптимизатора
40: criterion = nn.CrossEntropyLoss()
41: optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
42:
43: # Обучение модели
44: for epoch in range(10): # Количество эпох
45:     running_loss = 0.0
46:     for i, data in enumerate(trainloader, 0):
47:         inputs, labels = data
48:         optimizer.zero_grad()
49:         outputs = net(inputs)
50:         loss = criterion(outputs, labels)
51:         loss.backward()
52:         optimizer.step()
53:         running_loss += loss.item()
54:         if i % 2000 == 1999:
55:             print('[%d, %5d] loss: %.3f %'
56:                   (epoch + 1, i + 1, running_loss / 2000))
57:             running_loss = 0.0
58: print('Обучение завершено!')
```