



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)

ФАКУЛЬТЕТ

Робототехника и комплексная автоматизация (РК)

КАФЕДРА

Системы автоматизированного проектирования (РК6)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ НА ТЕМУ:

*«Анализ методов для улучшения точности модели  
сверточной нейронной сети, выполняющей задачу  
классификации.»*

Студент РК6-72Б

\_\_\_\_\_  
(Подпись, дата)

**Караф С.М.**

И.О. Фамилия

Руководитель НИР

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**

И.О. Фамилия

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

«\_\_\_\_» \_\_\_\_\_ 2022 г.

**З А Д А Н И Е**  
**на выполнение научно-исследовательской работы**

по теме: Анализ методов для улучшения точности модели сверточной нейронной сети, выполняющей задачу классификации. \_\_\_\_\_

Студент группы РК6-72Б

Караф Сармат Майк  
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.) учебная  
Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения НИР: 25% к 5 нед., 50% к 11 нед., 75% к 14 нед., 100% к 16 нед.

**Техническое задание:** Анализ методов для улучшения точности модели сверточной нейронной сети, выполняющей задачу классификации.

**Оформление научно-исследовательской работы:**

Расчетно-пояснительная записка на 14 листах формата А4.  
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

Дата выдачи задания «28» марта 2023 г.

Руководитель НИР

\_\_\_\_\_  
(Подпись, дата)

**Витюков Ф.А.**  
И.О. Фамилия

Студент

\_\_\_\_\_  
(Подпись, дата)

**Караф С.М.**  
И.О. Фамилия

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. Влияние размера целевого набора данных на передачу обучения.....	5
2. Передача обучения: обобщаемость признаков и влияние размера набора данных .....	7
3. Исследования и результаты .....	9
3.1. Исследования и результаты .....	9
3.2. Предварительная обработка данных.....	10
3.3 . Исследования и результаты .....	12
ЗАКЛЮЧЕНИЕ .....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	20

## ВВЕДЕНИЕ

В данной работе исследуется важная проблема точности моделей сверточных нейронных сетей (Convolutional Neural Networks, CNN) в задачах классификации. В современном информационном обществе, где классификация данных играет значительную роль, обеспечение высокой точности моделей является неотъемлемым требованием. Однако, на практике модели CNN могут сталкиваться с различными ограничениями, такими как сложность данных, ограниченное количество обучающих примеров, проблемы с дисбалансом классов и переобучением, что может снижать их точность.

Цель данного исследования заключается в анализе и исследовании различных методов, которые могут быть использованы для улучшения точности моделей сверточных нейронных сетей в задачах классификации. В работе рассматриваются различные подходы, такие как изменение архитектуры сети, применение регуляризации, оптимизация гиперпараметров и аугментация данных. Цель состоит в определении эффективных методов, которые помогут повысить точность моделей CNN и преодолеть ограничения, с которыми они сталкиваются.

В рамках исследования особое внимание уделяется влиянию размера целевого набора данных на обучение сверточных нейронных сетей глубокого обучения. В условиях, когда маркировка данных является дорогостоящей задачей и доступно ограниченное количество маркированных экземпляров для новых или специфических классов, важно определить наилучший подход к обучению моделей CNN. В работе представлены результаты серии экспериментов, включающих обучение целевых наборов классов с нуля, переобучение всех слоев и добавление дополнительных слоев к сети для наборов данных Tiny-ImageNet и Mini Places 2. Полученные результаты показывают, что для небольших целевых наборов данных наиболее эффективным подходом является замораживание весов начальных уровней сети. Кроме того,

представлена простая и доступная для реализации обучающая эвристика, основанная на этих выводах.

Дополнительно, в работе рассматривается метод случайного "выключения" (dropout), который позволяет значительно улучшить результаты моделей CNN на многих задачах-бенчмарках и достичь новых рекордов в распознавании речи и объектов. Этот метод позволяет предотвратить переобучение и сложные коадаптации между признаковыми детекторами, обучая каждый нейрон на обнаружение признаков, которые в целом полезны для правильного ответа в разнообразных контекстах.

Анализ методов для улучшения точности модели сверточной нейронной сети в задаче классификации представляет собой важное и актуальное направление исследования. Результаты данной работы могут быть полезными для исследователей и практиков, занимающихся разработкой и применением моделей CNN в различных областях, где задача классификации данных является ключевой.

## **1. Влияние размера целевого набора данных на передачу обучения**

В текущих исследованиях глубокого обучения достигается передовой уровень результатов в задачах классификации изображений. Современные модели используют глубокие сверточные нейронные сети (CNN), такие как AlexNet. Однако тренировка этих моделей на больших наборах данных, таких как ImageNet, может занимать значительное количество времени, а количество размеченных примеров для каждого класса может быть ограничено, поэтому обучение "с нуля" имеет свои недостатки. Один из подходов для преодоления этой проблемы - использование передачи обучения. Цель передачи обучения заключается в использовании знаний о задаче-источнике и их передаче на новую задачу-приемник. Это предоставляет значительные преимущества по сравнению с обучением "с нуля" (т.е. с случайной инициализацией весов). Одним из

очевидных преимуществ является то, что модель может обучаться более эффективно, так как она начинает с предварительно инициализированной матрицы весов.

В исследовании Yosinski и др. обучали AlexNet на наборе данных ImageNet и обнаружили, что первые три слоя в CNN содержат общие и повторно используемые признаки. За пределами третьего слоя признаки постепенно становятся более специфичными относительно исходного набора данных. Однако авторы не учли размер целевого набора данных, на котором будет обучаться модель с передаваемыми признаками.

Размер целевого набора данных играет важную роль, поскольку он влияет на то, насколько сильно передача обучения повлияет на результаты. Таким образом, логично задаться вопросом, насколько хорошо извлеченные признаки обобщаются на меньшие наборы данных. Было бы полезно знать, при каком размере набора данных передача обучения все еще будет полезной. Более конкретно, на каком слое модель все еще способна обобщаться на небольшой размер набора данных? Поэтому изучение того, при каком размере целевого набора данных передача обучения может предоставить дополнительную ценность, представляет как академический, так и практический интерес.

Кроме того, Yosinski и др. использовали только набор данных ImageNet. Было бы интересно выяснить, отличаются ли свойства передачи обучения при использовании набора данных из другой области.

В данной работе мы расширим исследование и измерим влияние размера целевого набора данных на передаваемость параметров в сверточных нейронных сетях. Нашим основным вкладом является количественная оценка степени обобщаемости признаков на целевой набор данных при систематическом уменьшении его размера. Мы будем исследовать это для каждого отдельного слоя, оценивая точность в зависимости от размера набора данных. У нас будет три варианта. Во-первых, мы получим базовый показатель, не применяя никакой

формы передачи обучения. Во втором варианте мы полностью до обучим все слои сети. В третьем варианте мы заморозим переданные признаки для каждого отдельного слоя. Мы будем исследовать это для разных размеров целевого набора. Более того, мы протестируем это на двух разных подмножествах наборов данных, каждое с разной областью, ImageNet и Places2.

## **2. Передача обучения: обобщаемость признаков и влияние размера набора данных**

Было проведено несколько исследований, посвященных обобщаемости признаков и успешности передачи обучения. Популярная стратегия передачи обучения — это тонкая настройка путем обучения линейного классификатора поверх последнего слоя сверточной нейронной сети (CNN). В работе Zeiler и др. было предварительно обучено CNN на наборе данных ImageNet, а затем обучен линейный классификатор на трех целевых наборах данных: PASCAL VOC 2012, Caltech-101 и Caltech-256. Они варьировали размер целевого набора данных и слой, с которого обучается классификатор. Они обнаружили, что модель обобщается очень хорошо на Caltech-101 и Caltech-256, но менее успешно на PASCAL. Тем не менее, исследование подтвердило преимущества передачи обучения. Подобные хорошие результаты были получены в, используя этот подход передачи обучения. Авторы предварительно обучали на ImageNet в сочетании с классификатором SVM и использовали задачи Pascal VOC и MIT-67 Indoor Scenes в качестве целевых задач.

В работе исследователи изучали, насколько хорошо признаки переносятся на различные целевые задачи в разных областях и исследовали, на каком слое сети это наиболее оптимально. Сначала они обучили AlexNet на наборе данных ImageNet и протестировали эти признаки на простой задаче распознавания объектов с использованием набора данных Caltech-101. Затем они протестировали сеть на адаптацию к другой области, где доступно небольшое

количество данных, используя базу данных Office. В-третьих, они проверили, насколько хорошо их модель работает на более подробном наборе данных с использованием набора данных Caltech-UCSD birds. Поскольку изображения в этом наборе данных очень похожи друг на друга, это довольно сложная задача классификации изображений. Наконец, авторы протестировали свою модель на базе данных SUN-397 Large-Scale Scene Recognition. Эта задача существенно отличается от исходной задачи, где требовалось классифицировать объекты. Целью набора данных SUN-397 является классификация сценических категорий. В каждом эксперименте авторы улучшали показатели результатов, что указывает на то, что признаки, изученные на ImageNet, обладают значительными обобщающими свойствами.

Наше исследование является прямым продолжением работы Yosinski и др. Они исследовали, насколько переносимы признаки между слоями в архитектуре AlexNet. Для этого они обучили две сети, N1 и N2, каждую на случайном разделении набора данных ImageNet, содержащем половину данных, разделы A и B. После обучения обеих сетей на своих соответствующих разделах, признаки первого слоя сети N1, базового, были переданы в первый слой сети N2, целевого. Оставшиеся слои в сети N2 были случайно инициализированы. Наконец, сеть N2 обучается на разделе B набора данных ImageNet. Таким образом, сеть N2 не обучается с нуля, а использует предварительно инициализированные признаки из сети N1. Исследователи делают это для слоев сети от первого до седьмого, передавая признаки как от A к B, так и от B к A. Они обнаружили, что признаки в первых трех слоях достаточно обобщены и могут быть переданы, улучшая производительность. Однако признаки в более глубоких слоях сети более специфичны для исходной задачи и, следовательно, их передача ухудшает производительность.

Предполагаем, что подход передачи обучения путем фиксации первых слоев является более ценным, если целевой набор данных меньше, и что для больших наборов данных обновление всех слоев даст лучшие результаты, и



подтверждаем это на наборах данных из двух разных областей классификации изображений.

### 3. Исследования и результаты

В данном разделе мы представляем обзор нашей экспериментальной настройки и результатов. Мы начнем с общего обзора подхода, а затем предоставим дополнительные детали в дополнительных подразделах.

#### 3.1. Исследования и результаты

Будем передавать признаки из CNN, обученной на задаче-источнике, на задачу-приемник, т.е. на наборы данных с различными классами исходов. Рассмотрим сценарий, где размер целевого набора данных такой же, а также меньший по размеру, чем исходный набор данных. Последнее условие является стандартным в передаче обучения. Гипотетически, ценность передачи обучения должна увеличиваться с уменьшением размера целевого набора данных. Более того, для каждого сценария мы будем исследовать первый случай, когда мы донастраиваем все слои с переданными признаками. Во втором случае мы будем передавать признаки, но замораживать веса сети в первых слоях.

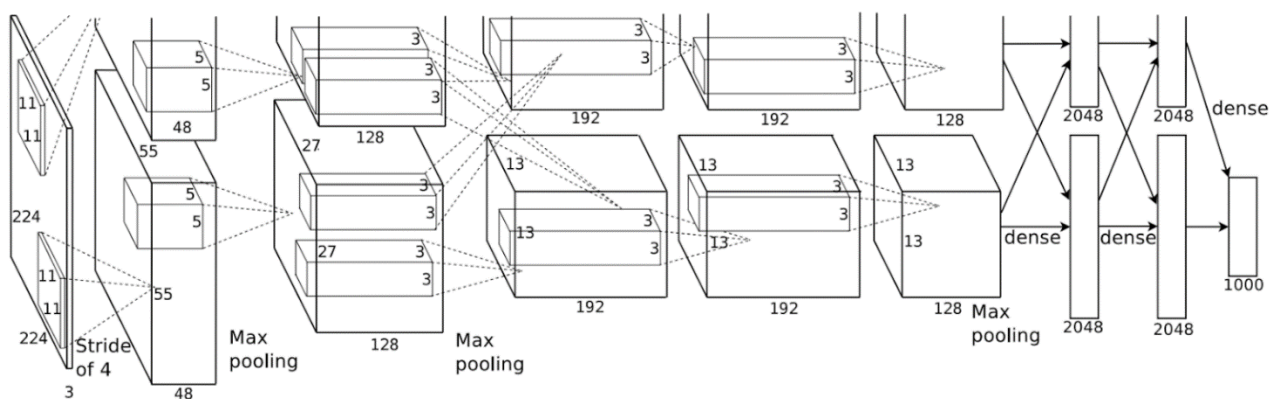


Рис. 1. AlexNet Архитектура.

Архитектура CNN, которую мы будем использовать, — это AlexNet, разработанная Крижевским и др. ал. (см. рисунок 1), которая была

победительницей в соревновании по распознаванию визуальных объектов ImageNet Large Scale в 2012 году. Модель состоит из пяти сверточных слоев и трех полносвязных слоев. Первые два сверточных слоя которыми следует слой максимальной дискретизации и слой нормализации соответственно. Пятый сверточный слой за которым следует только слой максимальной дискретизации. Первые два полносвязных слоя содержат 4096 нейронов. Последний полносвязный слой содержит 1000 нейронов для оценки классов целевой задачи. Интересно отметить, что авторы использовали функции активации с прямым распределением (ReLU) вместо обычной сигмоиды. Более того, они применили технику регуляризации, называемую исключением (dropout), для снижения переобучения.

### **3.2. Предварительная обработка данных**

В наших экспериментах мы использовали подмножество набора данных ImageNet [1] под названием Tiny-ImageNet. Этот набор данных содержит 100 000 изображений, разделенных на 200 классов, где каждый класс содержит 500 изображений размером 64 X 64 пикселя. Набор данных включает изображения широкого спектра объектов, таких как кошки, парковочные счетчики, утесы и регбийные мячи. Валидационный набор содержит отдельно 10 000 изображений.

Кроме того, мы расширяем работу Yosinski и др. , повторяя эксперименты на втором наборе данных, MiniPlaces2. Это уменьшенная версия более крупной базы данных MIT Places . Набор данных состоит из изображений различных сцен, таких как фут-корт, гольф-поле, офис и каток. В наборе данных MiniPlaces2 содержится 100 000 изображений с 100 классами. Каждый класс состоит из 1000 изображений размером 128 X 128 пикселей, но мы изменяем их размер до 64 X 64 пикселей, чтобы сохранить согласованность размеров с Tiny-ImageNet. Опять же, валидационный набор содержит 10 000 изображений.



(a) Lighthouse



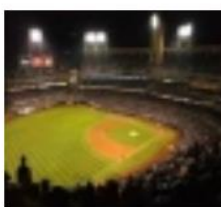
(b) Sulphur butterfly



(c) Umbrella



(d) Museum



(e) Baseball field



(f) Valley

Рис. 2. Верхняя часть: Пример обучающих изображений из набора данных Tiny-ImageNet. Нижняя часть: Пример обучающих изображений из набора данных MiniPlaces2.

На рисунке 2 мы представляем несколько классов изображений обоих наборов данных, чтобы подчеркнуть разницу между двумя доменами.

Для измерения влияния размера набора данных на обобщаемость признаков мы переносим признаки с исходной задачи на целевую задачу, у которой размер может изменяться. Мы будем тестировать это на подмножестве наборов данных ImageNet и Places2. Мы используем подмножество наборов данных вместо обучения на полных наборах данных ImageNet и Places2 (содержащих соответственно 1,2 миллиона и 8,1 миллиона изображений для обучения) из-за ограничений вычислительных возможностей. Мы обозначаем наш целевой набор данных как  $N_{target}$ . Кроме того, мы определяем разбиения набора данных с изменяемым размером как  $M_{target_i}$ , где  $M_{target_i} \subseteq N_{target}$ . Чтобы получить  $M_{target_i}$  из  $N_{target}$ , мы выполняем следующую процедуру:

1. Мы случайным образом разделяем весь набор данных на исходную и целевую части, соответственно  $N_{source}$  и  $N_{target}$ , где каждая часть содержит 50 000 изображений. В обеих частях изображения равномерно распределены по  $k = 100$  классам, с 500 изображениями в каждом классе для Tiny-ImageNet. В

MiniPlaces2 разделение составляет  $k = 50$  классов на каждую часть, с 1000 изображений в каждом классе.

2. Мы искусственно уменьшаем размер  $N_{target}$ , выбирая случайные выборки размером  $M_{target_i}$  из каждого класса  $k$ , где  $i$  принимает значения 5001, 400, 300, 200, 100 и 50 в случае Tiny-ImageNet. Для MiniPlaces2  $i$  равно 10001, 900, 800, 700, 600 и 500.

### 3.3. Передача признаков

В целях создания модели, из которой можно было бы передавать признаки, сначала производится обучение нашей сети на исходных данных ( $N_{source}$ ). Параметры модели источника сохраняются в объекте `Caffemodel` (см. раздел 3.4), который используется для передачи параметров от модели источника к модели целевой задачи.

Для получения базового показателя мы не применяем никакого метода передачи обучения и позволяем модели обучаться на предоставленном наборе обучающих данных. В нашем первом эксперименте мы настраиваем сеть, передавая все признаки от задачи источника к модели и затем продолжаем обучение на новой задаче с использованием обратного распространения ошибки.

Однако, так как нас также интересует, на каком слое  $l$  сети признаки могут обобщаться, мы передаем признаки от задачи источника к задаче цели по одному слою за раз. В AlexNet всего восемь слоев. Поэтому мы передаем признаки, начиная со слоя  $l = 1$  и заканчивая слоем  $l = 7$ . При передаче параметров модели цели мы их фиксируем, то есть не обновляем параметры с помощью градиентного спуска. Оставшиеся 8 -  $l$  слоев сети инициализируются случайным образом, и ошибки распространяются через эти слои.

Наконец, для получения среднего показателя точности мы повторяем эксперименты, следуя той же процедуре, но на этот раз используем  $N_{source}$  в качестве  $N_{target}$  и наоборот.

### 3.4. Обучение

Для проведения экспериментов используется глубокая нейронная сеть Caffe, разработанная в Университете Калифорнии в Беркли. В GPU-режиме активируется Caffe с помощью одной графической карты Nvidia GTX Titan X, что позволяет ускорить время обучения. Используется модель-эталон AlexNet, включенная в Caffe. Подробную информацию об архитектуре модели можно найти в [1]. Кроме того, мы следуем тому же режиму обучения, что и указано в модели-эталоне.

В контексте аугментации данных, в фазе обучения мы производим случайное обрезание и случайное отражение изображений, как указано в Caffe. В фазе тестирования мы делаем центральное обрезание изображений. Так как наши входные изображения имеют размер 64 x 64, мы изменяем размер обрезки до 57, а не масштабируем изображения до размера 256 x 256 и не применяем стандартный размер обрезки 227. Таким образом, мы сохраняем соотношение, используемое в модели-эталоне AlexNet. Кроме того, мы вычитаем среднее изображение из каждого изображения.

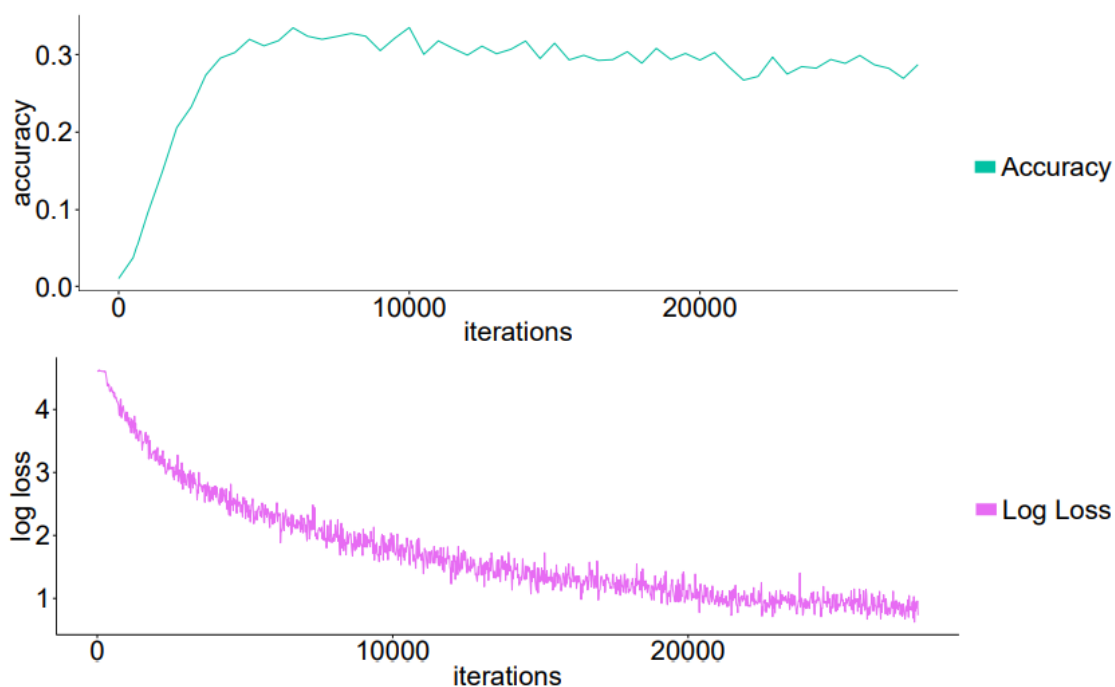


Рис. 3. Верхняя часть: Точность на V source после обучения на Nsource наборе данных TinyImageNet после 25 тыс. итераций. Этот набор содержит 100 классов, по 500 изображений в каждом классе. Нижняя часть: Логарифмическая потеря на обучающем наборе данных с тем же разделением.

Наконец, чтобы определить количество итераций для обучения моделей, мы обучаемся на  $N_{source}$  обоих наборах данных и проводим валидацию на соответствующих наборах  $V_{source}$ , не применяя никаких форм передачи обучения. Мы обнаружили, что модель начинает переобучаться на обучающих данных примерно после 10 000 итераций (см. рисунки 3 и 4). Поэтому для последующих экспериментов было разумно запустить каждую модель на 10 000 итераций, чтобы измерить положительный эффект передачи обучения. Кроме того, чем меньше  $N_{target}$  (размер целевого набора данных), тем быстрее модель достигнет точки переобучения, что подтверждается уменьшающейся точностью базовых условий обучения в наших экспериментах.

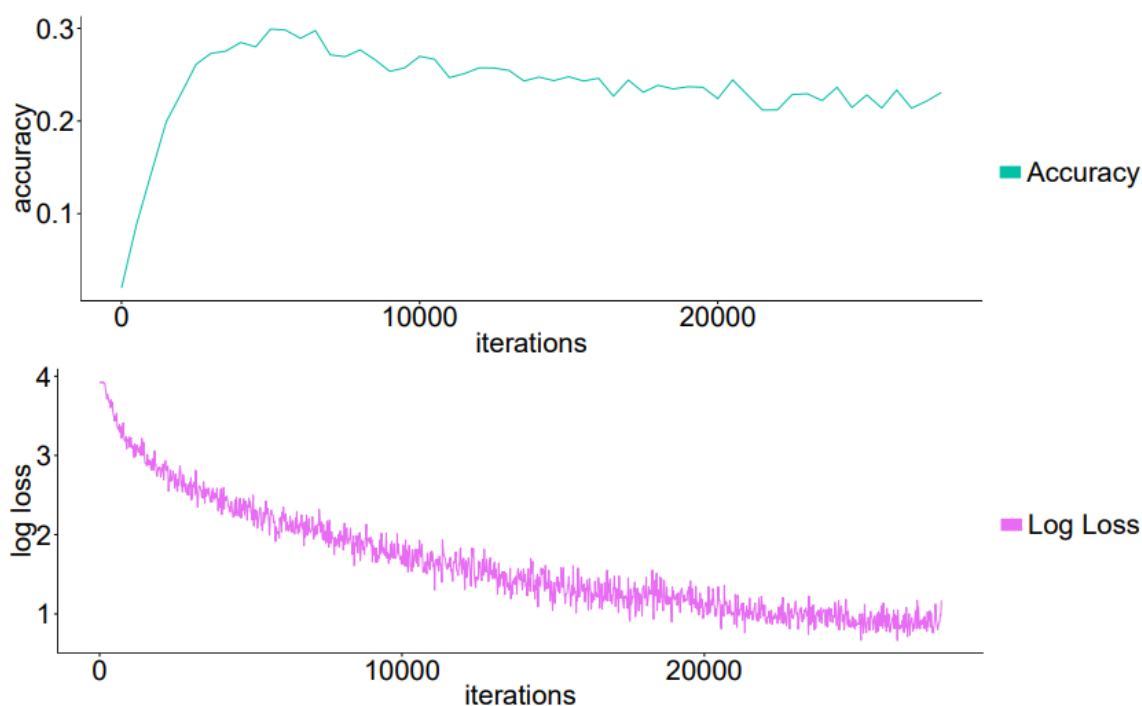


Рис. 4. Верхняя часть: Точность на исходном наборе данных  $V$  после обучения на  $N_{source}$  наборе данных MiniPlaces2 после 25 тысяч итераций. Этот набор данных содержит 50 классов, с 1000 изображений в каждом классе. Нижняя часть: Логарифмическая потеря на обучающем наборе данных с той же самой разбивкой.

### 3.4.1 . Результаты Tiny-ImageNet

На рисунке 5 показаны результаты передачи обучения на наборах данных разных размеров. График показывает точность на валидационном наборе данных после 10 000 итераций обучения.

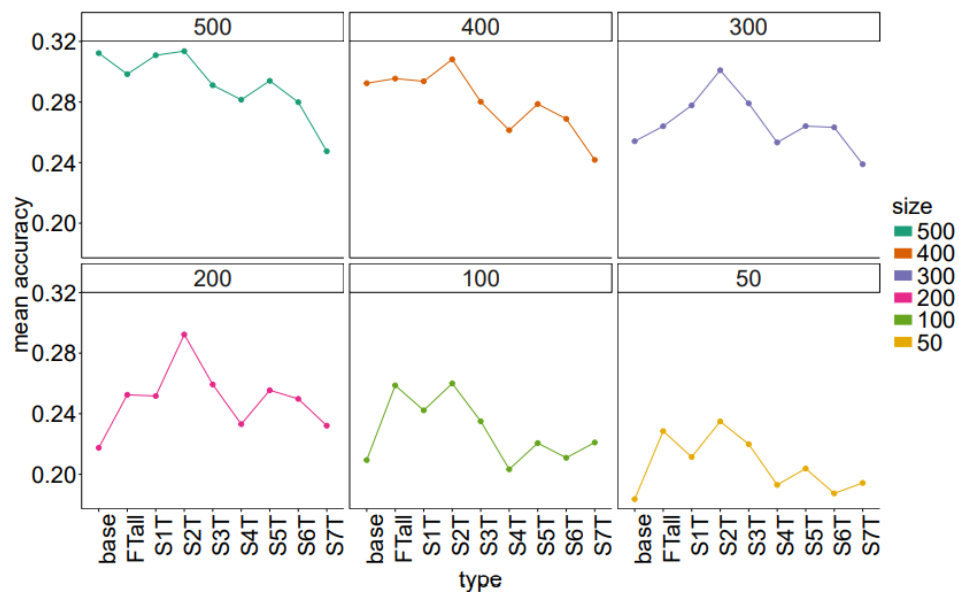


Рис. 5. Средняя точность, полученная после обучения на разделах целевого набора данных TinyImageNet, где  $i$  в  $M_{target_i}$  равно 500, 400, 300, 200, 100 и 50, и валидация проводилась на  $V_{target}$ . Обратите внимание, что мы провели аналогичные эксперименты, но использовали  $N_{source}$  вместо  $N_{target}$  и наоборот. Таким образом, наши средние показатели точности были получены путем усреднения результатов.

Первые два условия — это базовый случай и тонкая настройка всех слоев. Условие "базовый" указывает, что мы не применяли передачу обучения. Условие "FT all" означает, что мы проводили тонкую настройку всех слоев, а обозначение "SnT" указывает до каких слоев мы замораживали передаваемые признаки из исходной модели в целевую модель. Например, "S3T" означает, что мы передали первые три слоя признаков из модели, обученной на наборе данных " $N_{source}$ ", в модель, обученную на наборе данных " $M_{target_i}$ ". Последние семь значений точности представляют собой результаты, где мы передаем параметры для каждого слоя из исходной модели и замораживаем этот конкретный слой. Мы замечаем влияние размера набора данных на точность базового значения. При уменьшении размера набора данных точность также уменьшается. На рисунке 5 мы наблюдаем, что точность ухудшается, поскольку мы фиксируем больше слоев при передаче параметров из исходной задачи.

### 3.5. Результаты MiniPlaces2

Как видно из рисунка 6 несмотря на то, что это задача из другой области, результаты следуют очень похожему шаблону, как и в случае с Tiny-ImageNet. С уменьшением размеров целевого набора данных увеличиваются преимущества блокировки первых нескольких слоев.

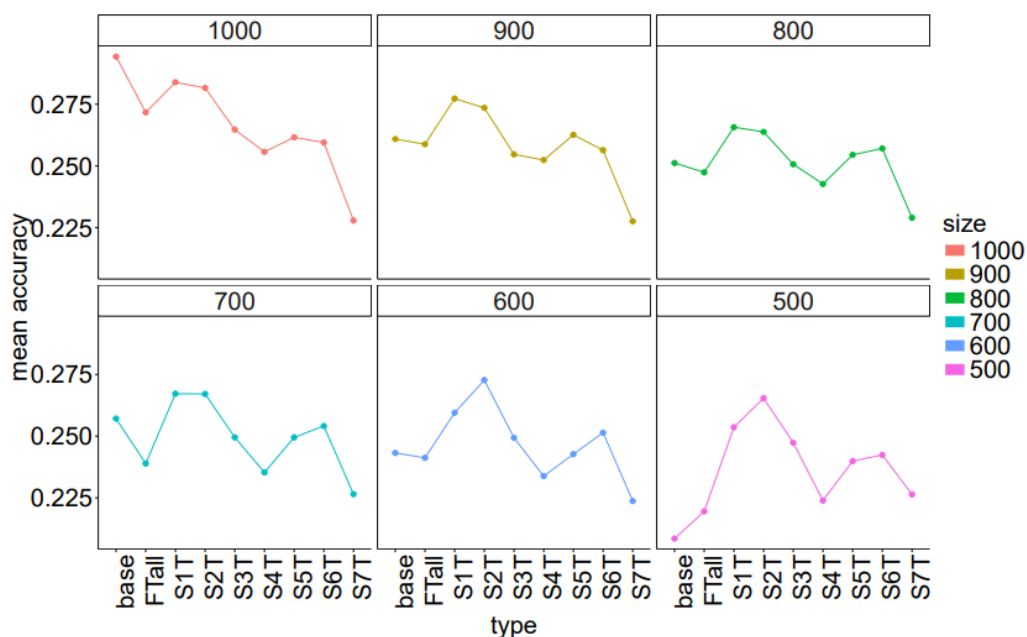


Рис. 6. Средняя точность после обучения на разделах целевого набора данных MiniPlaces2, где  $i$  в  $M_{target_i}$  равно 1000, 900, 800, 700, 600 и 500, и валидация проводилась на  $V_{target}$ .

Только для  $M_{target1000}$  графики указывают на то, что обучение "с нуля" лучше, но это действительно только базовое значение. В реальной ситуации, вероятно, ожидается, что также необходимо распознавать классы исходного набора данных, и производительность настройки всех слоев все равно ниже, чем блокировка некоторых из начальных слоев. Когда все слои сети могут быть адаптированы, инициализация весов, полученных из модели, обученной на большем наборе данных, все равно предоставляет преимущество. Предполагается, что признаки, выученные на исходном наборе данных, также важны для целевого набора данных. Поэтому инициализация модели параметрами из модели, обученной на большем наборе данных, явно демонстрирует свое преимущество. Интересно, что при передаче параметров из первых двух слоев во всех наших экспериментах наблюдается заметный прирост



точности, в то время как передача четырех слоев приводит к существенному снижению по сравнению с передачей первых трех слоев.

Результаты на рисунке 5 соответствуют результатам исследования Йосински и др. [17]. По мере передачи большего количества признаков (слоев) из исходной задачи, точность сначала улучшается, а затем ухудшается. Это можно объяснить специфичностью признаков для исходной задачи. Однако мы наблюдаем второй положительный всплеск точности на слое  $l = 5$  в почти всех наших экспериментах. Этот результат удивительный, поскольку признаки становятся очень специфичными для исходной задачи, но хорошо обобщаются на новую задачу. Очевидно, что передаваемые признаки из исходной задачи в этом слое обладают той же или даже более высокой представительной силой по сравнению с признаками, выученными только из целевого набора данных.

Эти результаты можно суммировать в простое эвристическое правило. Для начальных  $n$  экземпляров нового класса рекомендуется заморозить первые  $l$  слоев сети. Как только количество экземпляров нового класса превышает  $n$ , обучение может быть применено ко всем слоям. Значения  $n$  и  $l$ , очевидно, зависят от конкретных данных и задачи. В наших экспериментах замораживание первых 3 слоев до 300 экземпляров на класс в Tiny-ImageNet и 900 экземпляров на класс в MiniPlaces2 дало лучшие результаты.

Наше исследование могло бы получить дополнительную пользу от большего количества образцов на точку данных путем проведения повторных экспериментов. Поскольку инициализация параметров происходит случайным образом, параметры могут сходиться к различным локальным минимумам при каждом запуске модели, что может повлиять на точность в тестовой фазе. Тем не менее, наши результаты указывают на то, что передача признаков из большего исходного набора данных в меньший целевой набор данных добавляет ценность, уменьшая риск переобучения и улучшая производительность.

В заключение, в этой работе было исследовано влияние размера набора данных на обобщаемость признаков в глубоких сверточных нейронных сетях. Путем передачи признаков из предварительно обученной сети в новую сеть и систематического уменьшения размера целевого набора данных мы обнаружили, что замораживание первых двух или трех слоев существенно повышает производительность по сравнению с базовым показателем, особенно для малых размеров целевого набора данных менее тысячи экземпляров на класс.

## ЗАКЛЮЧЕНИЕ

Результаты показывают, что размер набора данных влияет на точность передачи обучения с помощью глубоких сверточных нейронных сетей. Первое влияние, которое мы замечаем, проявляется в базовом случае (повторим, когда модель обучается с случайной инициализацией весов). Мы видим, что модель начинает переобучаться на тренировочных данных, когда мы искусственно уменьшаем размер набора данных, что приводит к постепенному снижению точности как на Tiny-ImageNet, так и на MiniPlaces2. Это можно объяснить неоптимальной конфигурацией параметров, вызванной переобучением на небольшом размере набора данных.

Кроме того, настройка всех слоев модели показывает положительный эффект только с более маленькими наборами данных для Tiny-ImageNet, где значение  $M_{target_i}$  находится в диапазоне от 400 до 50, и для MiniPlaces2, где  $M_{target_i}$  равно 500. Это интересный результат,

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Girshick R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation //arXiv preprint arXiv:1311.2524. – 2013.
2. Gokhale V. et al. A 240 G-ops/s Mobile Coprocessor for Deep Neural Networks //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. – 2014. – С. 682-687.
3. Schmidhuber J. Deep Learning in Neural Networks: An Overview //arXiv preprint arXiv:1404.7828. – 2014.
4. Pham P. H. et al. NeuFlow: dataflow vision processing system-on-a-chip //Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on. – IEEE, 2012. – С. 1044-1047
5. LeCun Y. et al. Gradient-based learning applied to document recognition //Proceedings of the IEEE. – 1998. – Т. 86. – №. 11. – С. 2278-2324.
6. Everingham M. et al. The pascal visual object classes (voc) challenge //International journal of computer vision. – 2010. – Т. 88. – №. 2. – С. 303-338.
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
8. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 106(1), 59–70 (2007)
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the

- IEEE conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
11. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
  12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
  13. 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)
  14. 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)
  15. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1717–1724 (2014)
  16. 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)
  17. 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)
  - 18.. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
  19. 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)

20. 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)