

УДК 004.89

## ПРИМЕНЕНИЕ МЕТОДОВ ДОМЕННОЙ АДАПТАЦИИ В ЗАДАЧЕ КЛАССИФИКАЦИИ РАСТЕНИЙ ПО ИЗОБРАЖЕНИЯМ ИХ ЛИСТЬЕВ

Резвая Екатерина Петровна<sup>1</sup>, Гончаров Павел Владимирович<sup>2</sup>,  
Ососков Геннадий Алексеевич<sup>3</sup>

<sup>1</sup>Студент;

ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: rezvaya2016@gmail.com.

<sup>2</sup>Аспирант;

ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: kaliostrogoblin3@gmail.com.

<sup>3</sup>Доктор физико-математических наук., профессор;

ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: ososkov@jinr.ru.

Потери урожая из-за болезней растений являются серьезной проблемой для сельского хозяйства и экономики. В ЛИТ ОИЯИ была разработана платформа для определения болезней растений (PDDP). В PDDP для решения проблемы распознавания болезней растений по фотографиям их листьев успешно используются методы глубокого обучения. Такие методы требуют большой обучающей выборки. В то же время существует ряд методов, используемых для решения задач классификации в случае малой обучающей выборки. К таким методам относятся методы доменной адаптации. В работе проведено сравнение трех методов: Domain-Adversarial Training of Neural Networks (DANN), двухэтапный перенос обучения и Unsupervised Domain Adaptation with Deep Metric Learning (M-ADDA). M-ADDA позволил достигнуть точности классификации 92%.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00829.

**Ключевые слова:** классификация болезней растений, глубокое обучение, доменная адаптация, искусственные нейронные сети.

## USING DEEP DOMAIN ADAPTATION FOR IMAGE-BASED PLANT DISEASE DETECTION

Rezvaya Ekaterina<sup>1</sup>, Goncharov Pavel<sup>2</sup>, Ososkov Gennady<sup>3</sup>

<sup>1</sup>Student;

Dubna State University;  
Institute of the system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: rezvaya2016@gmail.com.

<sup>2</sup>PhD student;

Dubna State University;  
Institute of the system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: kaliostrogoblin3@gmail.com.

<sup>3</sup>*Professor, Doctor of Science (Phys & Math);  
Dubna State University;  
Institute of the system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: ososkov@jinr.ru.*

*Crop losses due to plant diseases is a serious problem for the farming sector of agriculture and the economy. Therefore, a multi-functional Plant Disease Detection Platform (PDDP) was developed in the LIT JINR. Deep learning techniques are successfully used in PDDP to solve the problem of recognizing plant diseases from photographs of their leaves. However, such methods require a large training dataset. At the same time, there are number of methods used to solve classification problems in cases of a small training dataset, as for example, domain adaptation (DA) methods. In this paper, a comparative study of three DA methods is performed: Domain-Adversarial Training of Neural Networks (DANN), two-steps transfer learning and Unsupervised Domain Adaptation with Deep Metric Learning (M-ADDA). The advantage of the M-ADDA method was shown, which allowed to achieve 92% of classification accuracy.*

*The reported study was funded by RFBR according to the research project № 18-07-00829.*

**Keywords:** classification of plant diseases, deep learning, domain adaptation, artificial neural networks.

## **Введение**

Болезни растений представляют серьезную угрозу, причем, как для частных фермерских хозяйств, так и для сельскохозяйственного сектора, в целом. Даже в случае, когда заболевание не приводит к полной гибели растения, оно может привести к снижению качества получаемой продукции. По статистике, из-за различных заболеваний теряется более 35% урожая [1]. Своевременно принятые меры и подходящий комплекс средств позволят предотвратить распространение заразы по всей плантации и спасти урожай.

Для подбора комплекса средств для успешной борьбы с болезнью необходимо точно определить ее вид. Но не все люди, занимающиеся сельским хозяйством, имеют достаточно опыта в определении возбудителя заболевания. А допущенная на ранних стадиях ошибка может стать губительной. Поэтому решением может стать специальное программное обеспечение, которое будет помогать фермерам определять наличие заболевания и устанавливать его вид. В наше время развитых IT-технологий, практически у всех есть мобильные устройства с различными мобильными приложениями, в том числе и такими, которые могут анализировать фотографии растений. Такое приложение *Plant Disease Detection Platform (PDDP)* уже было разработано в ЛИТ ОИЯИ [2, 3]. Пользователь должен просто загрузить фотографию больного листа, а приложение определит вид заболевания и может выдать рекомендации по лечению.

В *PDDP* для распознавания видов заболеваний по изображениям листьев заболевших растений успешно используются методы глубокого обучения [3]. Но такие методы требуют большой обучающей выборки, которая должна включать изображения всех изучаемых сортов растений, заболеваний наиболее типичных для каждого сорта, а также изображения листьев здоровых растений. Сбор и маркировка подходящего набора данных – очень сложная и дорогостоящая процедура, поэтому чаще всего собирается только относительно небольшая база размеченных изображений. Хотя уже существуют аналогичные программы [4], но эксперименты с использованием их баз данных показали, что они не позволяют получить требуемой точности на реальных изображениях, так как большинство из них имеют синтетическую природу (однотипные изображения, однотонный фон, строгое позиционирование) и не дают хорошего распознавания при загрузке реального изображения [5]. В *PDDP* была собрана своя база данных с реальными изображениями, но она состоит всего из 765 изображений, поэтому считается малой обучающей выборкой.

Принимая во внимание вышесказанное, потребовалось использовать такие методы, которые позволяют работать с малой выборкой. Среди прочих к таким методам относится метод переноса обучения (*transfer learning*) [5], когда нейросеть предобучается на известной большой выборке, а потом часть весов предобученной нейросети используется для доучивания на имеющейся малой целевой выборке. Другие методы используют самую малую выборку с применением обучения сиаемской нейронной сети [5], или метод для работы с классами, которых не было в обучающей выборке – *zero-shot learning* [5,

б). Наличие больших наборов данных в открытом доступе повлияло на выбор еще одного из таких методов, называемого доменной адаптацией, из числа других методов для решения задачи классификации в условиях малой обучающей выборки.

В настоящей работе для получения хорошего результата классификации изображений при нехватке данных для обучения и тестирования был проведен анализ существующих методов доменной адаптации, применяемых в аналогичных задачах [768]. В качестве основных подходов для исследования были выбраны следующие три: *Domain-Adversarial Training of Neural Networks (DANN)*, двухэтапный перенос обучения и *Unsupervised Domain Adaptation with Deep Metric Learning (M-ADDA)*.

Для лучшего понимания специфики излагаемого материала и стратегии обучения нейросетей при использовании доменной адаптации необходимо вначале описать используемые наборы данных и обучаемые на них глубокие сверточные нейронные сети.

## 1. Обзор используемых наборов данных

### *Plant Village*

В сети в открытом доступе есть большая база изображений Plant Village, состоящая из 54036 размеченных снимков листьев растений 14 различных культур: яблоня, черника, вишня, виноград, апельсин, персик, болгарский перец, картофель, малина, соя, тыква, клубника и томат. Он содержит изображения 17 грибковых заболеваний, 4 бактериальных заболеваний, 2 плесневых заболевания, 2 вирусных заболевания и 1 заболевание, вызванное клещом. 12 видов сельскохозяйственных культур также имеют изображения здоровых листьев, которые не поражены болезнью [8]. Некоторые изображения из базы представлены на Рис. 1.



Рис. 1. Примеры изображений из Plant Village Dataset

Исследования показывают, что, используя этот набор данных для обучения нейроклассификатора, можно достичь хороших результатов. Но все фотографии, содержащиеся в базе, сделаны на сером фоне со строгим позиционированием листа в центре, что мешает получить хороший результат в случае загрузки изображения листа в реальных условиях (смещенный относительно центра лист, яркий фон, посторонние предметы в кадре). Сделать фотографии, подобные изображениям из базы, для пользователя приложения иногда может быть затруднительным, в связи отсутствием необходимого оборудования и условий. Реально пользователь делает фотографию, не отрывая лист растения, а прямо на стебле, с посторонними предметами на фоне и неравномерным освещением.

### *PDD Dataset*

*PDD dataset* [3] был собран в ЛИТ ОИЯИ. Набор содержит 765 размеченных изображений листьев растений 20 классов здоровых и больных листьев 4-х культур: винограда, пшеницы, кукурузы и хлопчатника. С помощью данного набора данных можно установить такие болезни как: черная гниль, апopleксия, мучнистая роса, хлороз (для винограда); черный бактериоз, бурая ржавчина, мучнистая роса, желтая ржавчина (для пшеницы); склероспороз, глазковая пятнистость, бурая пятнистость, ржавчина (для кукурузы); альтернариоз, мучнистая роса, вертициллезное увядание и нехватка элементов (для хлопчатника). Примеры фотографий из наборы представлены на Рис. 2.



Рис. 2. Изображения из набора данных PDD

Все изображения имеют размер 256x256 пикселей. На фоне можно найти посторонние предметы, яркие пятна. Листья сфотографированы под разными ракурсами с неоднородным освещением. Таким образом, все изображения, находящиеся в базе, максимально приближены к тем, что могут получиться у любого пользователя приложения в реальных условиях.

## 2. Сверточные нейронные сети

Так как основная задача исследования – классификация болезней растений по изображениям их листьев, то все рассмотренные далее методы подразумевают работу с изображениями и все они будут основаны на глубоких, то есть многослойных, сверточных нейронных сетях (*Convolution Neural Networks – CNN*) [9]. Поэтому устройство сверточной нейронной сети необходимо рассмотреть более подробно.

Сверткой называют операцию над парой матриц *A* и *B*, результатом которой является матрица *C*. Каждый элемент этой матрицы вычисляется как скалярное произведение матрицы *B* и некоторой подматрицы *A*, такой же размерности. Подматрица определяется положением вычисляемого элемента в *C*. Процесс свертки проиллюстрирован на Рис. 3.

Матрица *B* сканирует матрицу *A* и в каждом положении считается скалярное произведение *B* и той части матрицы *A*, на которую она наложена в данный момент. Получившийся результат записывается в соответствующий элемент матрицы *C*. В этом случае матрица *A* будет называться изображением, а матрица *B* – фильтром.

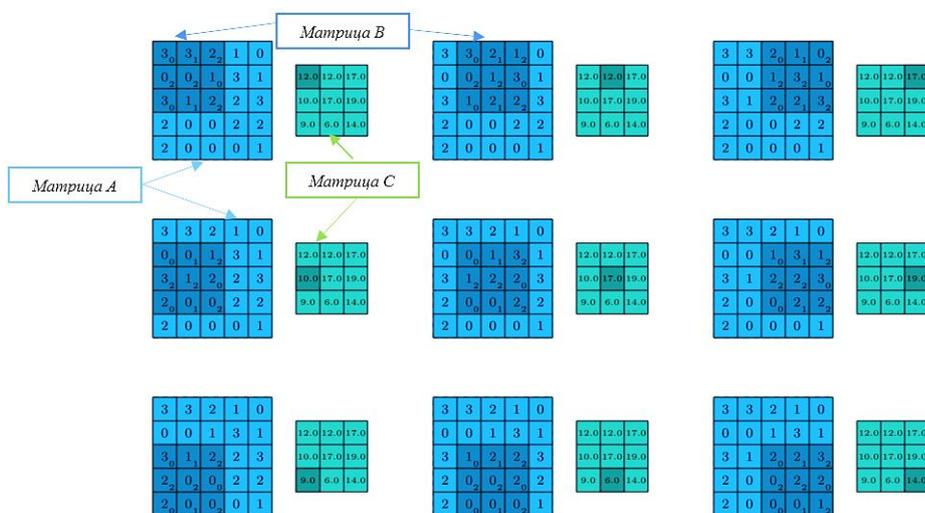


Рис. 3. Операция свертки

Приведенная выше техника работает, когда изображение имеет один входной канал. Но обычно на вход подаются цветные изображения. Цветное изображение чаще всего имеет 3 входных канала (*RGB*). В таком случае фильтр будет иметь 3 уникальных ядра (по одному для каждого входного канала) (см. Рис. 4).

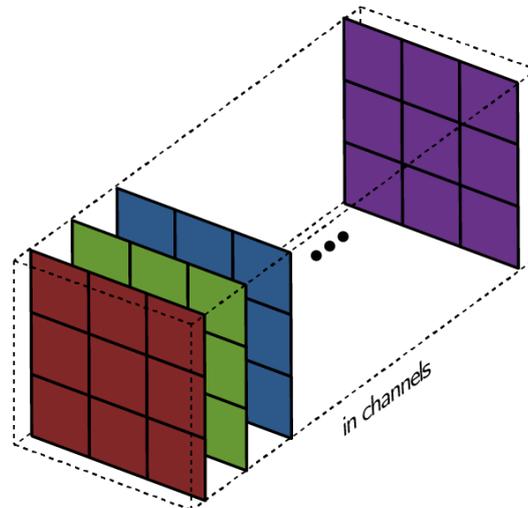


Рис. 4. Фильтр для многоканального изображения

Каждое из ядер фильтра перемещается по соответствующему входному каналу. В результате получается 3 результирующие матрицы  $C$ . Затем они суммируются, образуя на выходе 1 общий выходной канал.

Операция свертки происходит в сверточном слое нейронной сети. Основная задача этого слоя – выделение таких признаков как края, линии, цвет, градиентная ориентация и т. д. В качестве функции активации сверточного слоя обычно используется *ReLU* [9].

Кроме сверточного слоя есть пулинговый (*Pooling*) слой. Он предназначен для уменьшения свернутого элемента. Существуют два основных типа пулинга: максимальный (возвращает максимальное значение из части изображения, покрываемого фильтром) и средний (возвращает среднее). Суть пулинга – извлечение доминирующих признаков, независимых от вращения и позиционирования объекта. Также пулинговый слой подавляет шум. Кроме того, он позволяет уменьшить вычислительную мощность, необходимую для обработки данных.

Таким образом, сверточный и пулинговый слои вместе образуют слой сверточной нейронной сети. В зависимости от подаваемого на вход изображения количество слоев может быть увеличено. В конце на выходе нейросети должен быть слой-классификатор, определяющий, к какому из классов принадлежит проверяемый объект. Поэтому множество признаков, получаемых после слоев свертки и пулинга, соединяются в один вектор для подачи на выходной полносвязный слой нейронов с сигмоидальной или *Softmax* [14] функцией активации. В качестве функции потерь сверточной нейросети обычно используют функцию кросс-энтропии. Описание вышеприведенных терминов и понятий можно найти, например, в монографии [9]. Данный процесс позволяет нейросети выделить признаки изображения, которые, в свою очередь, станут входными данными для классификатора.

Для исследования вначале была выбрана сеть с пятью сверточными и пулинговыми слоями и классификатором, состоящим из 20 нейронов (для распознавания 20 классов). Количество слоев мотивировано размером обучающей выборки. Так как обучение многослойной (10 и более слоев) сети на малой обучающей выборке может грозить переобучением, а малое количество слоев не справится с качественной обработкой признаков изображений. Поэтому было выбрано среднее значение – 5 слоев.

### 3. Способы обучения CNN

Задача обучения *CNN* точно также сводится к задаче минимизации функции потерь, но эта функция очень большой размерности. Поэтому для ее вычисления, как правило, используют адаптивный стохастический градиентный спуск *ADAM* (*Adaptive Moment Estimation*) [9].

Сверточную нейронную сеть нельзя успешно обучить на малой тренировочной выборке сразу с нуля. Так, в конкретной задаче распознавания листьев растений по изображениям их листьев, при обучении *CNN* с 5 слоями результаты классификации листьев растений не превышают 60% точности, что является неудовлетворительным результатом. Поэтому необходимо приблизить значения весов *CNN* уже в начале обучения к ожидаемым. Для этого можно использовать метод переноса обучения.

### Метод переноса обучения

Чем сложнее задача, тем труднее получить удовлетворительный результат, обучая сверточную нейронную сеть с нуля. Поэтому, чаще всего, для получения хорошего результата на практике используют метод переноса обучения (*transfer learning*). Метод предполагает обучение нейросети в два этапа:

1. Предобучение. Сеть обучается на большом датасете с большим количеством разнообразных классов (например, *ImageNet* [10]).
2. Дообучение. Предобученная модель обучается на данных целевой задачи, используя накопленный опыт. Перед этим происходит замена классификатора (так как количество классов разное) и тонкая настройка параметров модели. Иногда несколько слоев замораживают. Это позволяет зафиксировать веса предобученной модели.

Как показали наши вычисления, на первом этапе метода переноса обучения для задачи распознавания болезней растений точность классификации достигает 99% на *Plant Village* – большой размеченной базе изображений листьев растений (подробнее речь о ней пойдет далее). Но результаты переноса обучения на малый набор данных *PDD* для текущего исследования не превышают точности классификации 85%. Поэтому для достижения хорошего результата обучения на *PDD* необходимо дополнять *transfer learning* другими алгоритмами.

## 4. Доменная адаптация

На данный момент, уже реализованы подходы, основанные на сиамских нейронных сетях, дающие на данных *PDD* успешные результаты [11]. Но существуют и другие потенциально успешные методы – методы глубокой доменной адаптации. Суть этих методов в том, что используется сразу 2 набора данных – домен-источник и целевой. Уровень схожести между целевым и исходным наборами данных определяет, насколько успешным будет обучение [12]. Поэтому для успешного обучения необходимо верно подобрать наборы данных.

Целевой набор данных определен постановкой задачи – *PDD*. Для домена-источника больше всего подойдет база изображений – *Plant Village*. У нее есть несколько преимуществ: во-первых, размер. Она содержит больше изображений, чем аналогичные датасеты. Следующий фактор – база размечена специалистами, а значит данные точно не содержат ошибки. Кроме того, *Plant Village* находится в свободном доступе в интернете. Домен-источник имеет в наличии 6 классов из целевого, поэтому можно считать их схожесть удовлетворительной.

Современные методы глубокой доменной адаптации можно разделить на 3 большие группы:

*Discrepancy-based* подходы, основанные на минимизации расстояния между векторными представлениями на исходном и целевом доменах с помощью введения этого расстояния в *loss*-функцию.

*Adversarial-Based* подходы используют состязательную (*adversarial*) *loss*-функцию, появившуюся в генеративно-состязательных нейросетях (*Generative-Adversarial Networks - GAN*) [9], для обучения сети, инвариантной относительно домена.

Смешанные методы, которые не используют *adversarial loss*, но применяют идеи из *discrepancy-based* семейства, а также последние наработки из глубокого обучения [12].

Для дальнейшего использования выбранных наборов данных была проведена нормализация каждого изображения. Нормализация предполагает замену номинальных признаков так, чтобы каждый из них лежал в диапазоне от 0 до 1. Она позволяет всем признакам вносить одинаковый вклад в процесс обучения.

Пошаговое применение выбранных методов с использованием подготовленных данных описано далее.

### Алгоритм Domain-Adversarial Training of Neural Networks (DANN)

Алгоритм состоит из 3-х основных частей: основная сеть, с помощью которой происходит получение векторных представлений (*feature extractor*). На Рис. 5 она представлена зеленым цветом. Часть, отвечающая за классификацию на исходном домене (*label predictor*) представлена синим цветом. И часть, которая обучается отличать изображения из исходного домена от изображений целевого домена (*domain classifier*) выделена розовым цветом на Рис. 5. По результатам исследований, на наборах данных MNIST и его уменьшенной копии *mMNIST* данный метод позволил достичь точности 91% [12].

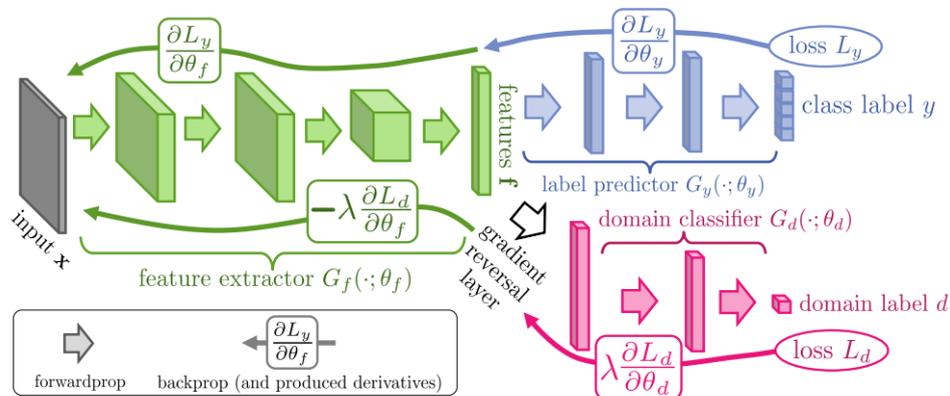


Рис. 5. Алгоритм DANN

Для использования метода *DANN* на задаче классификации растений с базой данных *PDD* потребовались этапы, описанные далее. Так как в алгоритме используется общий классификатор для двух наборов данных, на выходе должно быть одинаковое количество классов. Без предварительной подготовки, в домене источнике 38 классов, а в целевом домене 20. Поэтому необходимо найти пересечение двух наборов данных для того, чтобы количество классов было одинаковым. Для обучения будет использоваться 2 культуры: виноград и кукуруза (так как пшеница не встречается в домене-источнике). Данные разбиты на 6 классов: 2 класса – здоровые листья культур, 1 класс – апоплексия винограда и 3 класса – склероспороз, глазковая пятнистость и ржавчина кукурузы.

*Feature extractor* представляет собой обычную сверточную нейронную сеть, состоящую из 5 слоев, каждый из которых содержит операцию свертки, пулинговый слой, батч-нормализацию [13] и функцию активации нейронов *ReLU*.

Классификатор, определяющий принадлежность изображения к тому или иному классу, состоит из 3-х слоев с функции активации *ReLU*, применением батч-нормализации и функцией активации *Softmax* на выходе. Последний слой классификатора имеет 6 нейронов (для предсказания 6 классов).

Второй классификатор, отличающий домены друг от друга, имеет меньшее количество слоев, а именно, 2 слоя с функцией активации *ReLU* и слой батч-нормализации. На последнем слое 2 нейрона и функция активации *Softmax* на выходе.

Обучение сети с описанной выше архитектурой проходило в 1000 эпох, с использованием оптимизатора *Adam*. Тестирование показало, что результаты в этом случае плохие. Точность не превысила и 23%.

Проблема заключалась в том, что по сравнению с базой данных *MNIST*, где каждое изображение имеет размер 28x28 пикселей, базы данных с листьями растений имеют размер почти в 10 раз больше (256x256). И основная сверточная сеть не справляется с обработкой подаваемых на вход данных. Решением этой проблемы стала замена обычной сверточной сети на заранее предобученную на *ImageNet* сеть *MobileNetv2*. Архитектура классификаторов осталась неизменной. Этот прием позволил увеличить точность классификации до 43%. Но результат до сих пор остался неудовлетворительным.

Для повышения точности была проведена корректировка параметров (подбор шага обучения, использование дополнительных параметров оптимизатора, таких как *amsgrad* и *eps* [15]). Также была использована аугментация данных: то есть в целевой набор данных были добавлены зеркально отраженные дубликаты, повернутые в случайном направлении изображения и экземпляры с искаженной

цветовой палитрой. Обучение сети с новыми параметрами проводилось после предварительной заморозки слоев основной сети (всех, кроме 2-х последних). Этот прием позволяет зафиксировать веса, полученные на этапе предобучения и достигать более высокой точности.

Таким образом точность классификации удалось повысить до 61%. Это значит, что в 39 случаях из 100 сеть допускает ошибку, а это считается большой погрешностью. Поэтому для получения лучших результатов были применены другие методы.

### Доменная адаптация с двухэтапным переносом обучения

Метод переноса обучения не дает хороших результатов на малой выборке, но с другой стороны, на большом наборе данных показывает хорошую точность. Поэтому возникла идея использовать *transfer learning* в два этапа: тренировать предобученную сеть на полном наборе данных *Plant Village* (домен-источник), а затем, зафиксировав полученные на первом этапе веса, дообучить на полном наборе данных *PDD* (целевой датасет).

Цель первого этапа – приблизить распределение весов к нужному на целевой выборке для получения лучшего результата. Классификатор на первом этапе – линейный, с функцией активации *Softmax*.

Второй этап требует замены классификатора в сети, так как количество классов в доменах различно. Классификатор второго этапа имеет такую же архитектуру, как и на первом этапе, отличие только в количестве нейронов. Обучение сети на втором этапе проходит с уменьшением шага в процессе. Это позволяет постепенно замедлять обучение, для получения более точного результата.

Перед обучением была проведена аугментация, описанная в предыдущем методе. Сравнение результатов тестирования обученной сети *MobileNetv2* и *ResNet34* показало перспективность использования именно *ResNet34* (точность 90%). В то время как точность классификации на целевой выборке с использованием *MobileNet2* – 88%.

### Алгоритм *Unsupervised Domain Adaptation with Deep Metric Learning (M-ADDA)*

Также как в двухэтапном переносе обучения, и в отличие от метода *DANN*, *M-ADDA* подразумевает разделение сети для исходного домена и целевого. То есть процесс обучения на каждом домене требует отдельную сеть.

Основная идея алгоритма заключается в сближении векторных представлений на разных доменах. Для того, чтобы классы лучше разделялись в пространстве, при обучении на домене-источнике используется трехчленная функция потерь (см. 1) (*Triplet Loss* см, например, [11]).

$$Loss = \sum_i^N \left[ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \quad (1)$$

*Triplet Loss* одновременно минимизирует расстояние между позитивными примерами  $x_i^p$  (из одного класса) и изображением-якорем (выбранный образец из базы)  $x_i^a$  и максимизирует между негативными  $x_i^n$  (из разных классов) и «якорем». Здесь  $\alpha$  – расстояние между позитивными и негативными образцами. [16]

В результате векторные представления данных стремятся разбиться на  $N$ -кластеров (где  $N$  – количество классов), а для каждого кластера вычисляется его центр. То есть, решается задача кластеризации. Кластеризация, в отличие от классификации, относится к типу обучения нейронной сети без учителя, то есть происходит самообучение сети.

Затем инициализируется нейросеть для целевого домена и сеть-дискриминатор, предназначенная для определения домена, к которому принадлежит изображение. Дискриминатор уже решает задачу классификации. Он обучается при фиксированных весах доменных сетей, а затем замораживается и обучается уже целевая сеть. После этого добавляется регуляризация, обеспечивающая стягивание векторных представлений к ближайшему кластеру, использующая расстояние до центра кластера. Это обеспечивает лучшую разделимость классов в целевом домене [12].

Для применения метода к решению задачи классификации листьев растений, были созданы 3 нейросети. Так как предыдущие эксперименты доказали эффективность использования переноса обучения, было принято решение сразу использовать предобученную сеть в процессе дальнейшего обучения. Исходная ( $S$ ) и целевая ( $T$ ) сети имеют одинаковую архитектуру – *MobileNetv2*. Третья сеть – дискриминатор ( $D$ ) состоит из 3 слоев.

Сеть  $S$  обучалась на всем наборе данных *Plant Village* с использованием трехчленной функции потерь, а затем полученные веса были переданы в  $T$ -сеть. Этап обучения сети с использованием целевого домена выглядит следующим образом: дискриминатор обучается отличать изображения из исходного домена от изображений целевого домена на протяжении 3 эпох, затем замораживается, и целевая сеть в течении 7 эпох учится приближать векторные представления изображений из целевого домена к центрам кластеров, образованных во время обучения на домене-источнике. Таких эпох в процессе обучения было 150.

Причем, этот метод (как и *DANN*) использует только пересечение датасетов, для того, чтобы корректно решалась задача кластеризации.

Вышеперечисленные шаги позволили так разделить классы в пространстве, что точность предсказания достигла 78%.

В процессе обучения и тестирования использовался метод  $k$ -ближайших соседей. Суть его в том, что классифицируемый объект сравнивается с  $k$ -ближайшими кластеризованными объектами обучающей выборки, и ему присваивается класс большинства этих соседей. В случае с задачей распознавания болезней растений количество соседей, по которым определялся класс, было равно 2.

78% все ещё не очень хороший результат. Для улучшения точности классификации был применен еще и метод переноса обучения.

Целевая сеть с трехчленной функцией потерь была дополнена классификатором с одним слоем из 20 нейронов (для классификации всего набора *PDD*, содержащего 20 классов изображений), а затем была обучена на протяжении 300 эпох на всей базе *PDD* с замедлением в 10 раз шага обучения каждые 100 эпох, с замороженными слоями (т.е. с сохранением опыта предыдущего обучения). Этот прием позволил получить удовлетворительный результат. Точность классификации на 20 классах набора данных *PDD* составила 92%.

Таким образом, было рассмотрено 3 метода глубокой доменной адаптации. Лучший результат классификации изображений был получен с использованием метода *M-ADDA*, с дополнительным применением метода переноса обучения. Он показал точность на тестовой выборке 92%. Классификация проводилась по 20 классам изображений.

## Заключение

В рамках данного исследования были изучены общепринятые способы обучения искусственных нейронных сетей. Эксперименты показали их неэффективность относительно задачи классификации болезней растений по изображениям их листьев в условиях малой обучающей выборки на данных *PDD*. Был проведен анализ существующих методов обучения нейронной сети на малых выборках, и в качестве исследуемого подхода была выбрана доменная адаптация.

Сравнение трех методов доменной адаптации показало успешность выбранного подхода.

Однако, один из методов – *DANN* оказался неэффективным при решении поставленной задачи. Он не позволил достичь желаемых результатов, точность классификации составила около 60%. Но, доменная адаптация с двухэтапным переносом обучения позволила получить удовлетворительную точность – 90%.

Лучший результат классификации показал метод *M-ADDA* с последующим использованием метода переноса обучения. Обученная сеть при тестировании делает только 8 ошибочных предсказаний из 100. Точность достигает 92%. Данные результаты получены при распознавании 20 классов изображений целевого набора данных.

Полученные результаты не хуже аналогичных результатов других авторов. Например, классификация болезней риса по изображениям листьев, с использованием сверточных нейронных сетей и метода переноса обучения в исследовании *Shreya Ghosal* и *Kamal Sarkar* [7] достигает точности 92%.

Здесь необходимо заметить, что данная работа была выполнена в рамках комплексного исследования, проведенного в ЛИТ ОИЯИ по сравнению различных методов повышения эффективности нейроклассификации растений по изображениям их листьев. Помимо текущего, было проведено еще исследование [5], где сравнивались возможности улучшения эффективности исходной сиамской сети

[3] методом переноса обучения для одного из ее близнецов и предельно тонкой настройке её самой с использованием *Keras Tuner* [17]. Последний подход дал в итоге 97.5% точность классификации. Также было проведено исследование [11], в котором для сиамской нейронной сети предложена комбинация методов переноса обучения с применением трехчленной функцией ошибки и выполнения финальной классификации при помощи многослойного персептрона, что позволило достигнуть рекордной точности классификации в 99.5%. Тем не менее текущее исследование, хотя его результат уступает двум другим по эффективности классификации, является не менее важным, так как позволяет завершить комплексное исследование по методам повышения эффективности для проекта *PDD* и заполнить весь спектр таких методов.

Все разработанные нами алгоритмы были реализованы на языке *Python*, а программный код размещен в свободном доступе в сети интернет по ссылке [18].

## Список литературы

1. Потеря урожая // Справочник «Экология». — 2019. — [Электронный ресурс]. URL: <https://ru-ecology.info/term/9058/>.
2. Платформа для определения болезней растений // ЛИТ ОИЯИ. — 2019. — [Электронный ресурс]. URL: <http://pdd.jinr.ru/>.
3. *Uzhinskiy A. et al. Multifunctional Platform and Mobile Application for Plant Disease Detection / A. Uzhinskiy, G. Ososkov, P. Goncharov, A. Nechaevskiy // Proceedings of the 27th Symposium on Nuclear Electronics and Computing (NEC 2019), Budva, Montenegro. — 2019. — Vol. 2507. — P. 110-114.*
4. Приложение для определения болезней растений // *Plantix*. — 2019. — [Электронный ресурс]. URL: <https://plantix.net/en/>.
5. *Goncharov P. et al. Deep Siamese Networks for Plant Disease Detection / Pavel Goncharov, Alexander Uzhinskiy, Gennady Ososkov, Andrey Nechaevskiy and Julia Zudikhina // EPJ Web of Conferences. — EPJ Sciences, 2020. — Vol. 226. — P. 03010.*
6. *One Model To Learn Them All // Cornell University. — 2019. — [Электронный ресурс]. URL: https://arxiv.org/pdf/1707.00600.pdf.*
7. Rice Leaf Diseases Classification Using CNN With Transfer Learning. — [Электронный ресурс]. URL: <https://ieeexplore.ieee.org/abstract/document/9106423>.
8. *Dataset of diseased plant leaf images and corresponding labels. — [Электронный ресурс]. URL: https://github.com/spMohanty/PlantVillage-Dataset.*
9. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение // Спб.: Питер, 2018. — С. 480.
10. *ImageNet. — [Электронный ресурс]. URL: http://www.image-net.org/.*
11. Сметанин А. А., Гончаров П. В., Ососков Г. А. Выбор методов глубокого обучения для решения задачи распознавания болезней растений в условиях малой обучающей выборки // Системный анализ в науке и образовании: сетевое научное издание. №1. — Дубна, 2020. — №1. — [Электронный ресурс]. URL: <http://www.sanse.ru/archive/55>.
12. Обзор основных методов *Deep Domain Adaptation* (Часть 1) // Блог компании *Mail.ru Group*. — 2018. — [Электронный ресурс]. URL: <https://habr.com/ru/company/mailru/blog/426803/>.
13. *Batch Normalization* для ускорения обучения нейронных сетей. — [Электронный ресурс]. URL: <https://habr.com/ru/post/309302/>.
14. Функции активации нейросети. — [Электронный ресурс]. URL: <https://neurohive.io/ru/osnovy-data-science/activation-functions/>.
15. *PyTorch optim. — [Электронный ресурс]. URL: https://pytorch.org/docs/stable/optim.html.*
16. *Florian Schroff, Dmitry Kalenichenko, James Philbin // FaceNet: A Unified Embedding for Face Recognition and Clustering. — [Электронный ресурс]. URL: https://arxiv.org/pdf/1503.03832.pdf.*

17. Открытая программная библиотека для машинного обучения *TensorFlow*. — [Электронный ресурс]. URL: <https://www.tensorflow.org/>.
18. Репозиторий *GitHub*. — [Электронный ресурс]. URL: <https://github.com/rezvaya/>.