

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический  
университет имени К.И. Сатпаева

Институт автоматизации и информационных технологий

УДК \_\_\_\_\_

На правах рукописи

Қабышев Айбек Мирболатұлы

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

На соискание академической степени магистра

Название диссертации	Формирование и разметка набора данных полученных с борта БПЛА для решения задач точного земледелия методами машинного обучения
Направление подготовки	7M06102 – Machine Learning & Data Science

Научный руководитель  
Профессор доктор инж.наук  
Р.И. Мухамедиев  
«\_\_» \_\_\_\_\_ 2023 г.

Оппонент  
канд. физ-мат. наук, ст.  
преподаватель  
\_\_\_\_\_ И.М. Уалиева  
«\_\_» \_\_\_\_\_ 2023 г.

Нормоконтроль  
PhD, ассистент-профессор  
\_\_\_\_\_ А.Т. Ахмедиярова  
«\_\_» \_\_\_\_\_ 2023 г.

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой ПИ  
канд. физ-мат. наук, профессор  
А.Н. Молдагулова  
«\_\_» \_\_\_\_\_ 2023 г.

Алматы 2023

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Казахский национальный исследовательский технический  
университет имени К.И. Сатпаева

Институт автоматизации и информационных технологий

Кафедра Программная инженерия  
Специальность: 7М06102 – Machine Learning & Data Science

УТВЕРЖДАЮ  
Заведующий кафедрой ПИ  
канд. физ-мат. наук, профессор  
\_\_\_\_\_ А.Н. Молдагулова  
«\_\_\_» \_\_\_\_\_ 2023 г.

ЗАДАНИЕ  
на выполнение магистерской диссертации

магистранту, обучающемуся Қабышев Айбек Мирболатұлы

Тема диссертации: «Формирование и разметка набора данных  
полученных с борта БПЛА для решения задач точного земледелия методами  
машинного обучения»

Срок сдачи законченной диссертации

«\_\_\_» \_\_\_\_\_

Исходные данные к магистерской диссертации. Дан анализ области  
обнаружения объектов, различных методов обнаружения и существующих  
наборов данных. Поставленные цели и задачи диссертационной работы  
обуславливают анализ основных подходов к решению задачи обнаружения  
объектов; изучение наборов данных; создание собственного набора данных и  
обучение нейронной сети YOLOv4 для решения задачи обнаружению сорняков и  
сорняков на грядках.

Перечень подлежащих разработке в магистерской диссертации вопросов  
или краткое содержание магистерской диссертации: а) ознакомление с  
глубоким обучением; б) исследование существующих работ по данной области;  
в) исследование и сравнение наборов данных; г) создание собственного набора  
данных; д) ознакомление с нейронной сетью YOLO; е) установка компонентов и  
подготовка к обучению ; ж) обучение YOLO и выводы по проделанной работе.

Рекомендуемая основная литература 1. Введение в машинное обучение.  
Мухамедиев Р.И., Амиргалиев Е.Н. – 2022. - 288 стр. 2. Redmon, J., & Farhadi, A.  
(2018). YOLOv3: An incremental improvement. 3. Mukhamediev R. I. et al. From  
Classical Machine Learning to Deep Neural Networks: A Simplified Scientometric  
Review // Applied Sciences. – 2021. – Т. 11. – №. 12. С.5541. 4. Mukhamediev R. I.  
et al. Review of Some Applications of Unmanned Aerial Vehicles Technology in the  
Resource-Rich Country // Applied Sciences. – 2021. – Т. 11. – №. 21. – С. 10171.

ГРАФИК

**подготовки магистерской диссертации**

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Раздел 1. Введение в машинное и глубокое обучение	23.10.2022 г.	Сдал
Раздел 2. Введение в область обнаружения объектов	30.11.2022 г.	Сдал
Раздел 3. Обзор существующих работ по данной области	6.02.2023 г.	Сдал
Раздел 4. Практическая часть диссертационной работы	15.04.2023 г.	Сдал

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант, (уч. степень, звание)	Сроки	Подпись
Нормоконтроль	А.Т. Ахмедиярова, PhD, ассистент-профессор		
Программное обеспечение	Н.К. Мукажанов, PhD, ассоциированный профессор		
Антиплагиат	Б. Аубакиров Ассистент		

Дата выдачи задания " \_\_\_\_ " \_\_\_\_\_ 2022 г.

Заведующий кафедрой \_\_\_\_\_ А.Н. Молдагулова

Научный руководитель Р.И. Мухамедиев

Задание принял к исполнению магистрант \_\_\_\_\_ А.М. Қабышев

Дата " \_\_\_\_ " \_\_\_\_\_ 2023 г.

**АНДАТПА**

Бұл диссертация біздің жеке деректер жиынтығын және оқытылған YOLOv4 нейрондық желісін пайдалана отырып, төсектердегі соя бұршақтары мен арамшөптерді анықтау әдісін дамытуға арналған. Соя және арамшөптерді анықтау мәселесі агротехнологияда өзекті болып табылады және тиімді анықтау жүйелері ауыл шаруашылығы жерлерінің тиімділігі мен басқаруын айтарлықтай жақсартуға мүмкіндік береді.

Жұмыста бейнелерде объектілерді анықтау саласындағы қолданыстағы әдістер мен технологияларға талдау жасалды. Жұмыс барысында тереңдетіп оқытудың теориялық негіздері, сұрақтары қарастырылып, мәліметтер жиынтығын салыстыру қарастырылады. Соя және әртүрлі арамшөп түрлеріне қатысты қолданыстағы тәсілдердің шектеулері анықталды. Бұл мәселені шешу үшін біздің жеке деректер жиынтығын құру ұсынылды. Жұмыс әдістемесі мамандандырылған кескінді аннотациялау құралдарын пайдалана отырып, төсектердегі соя және арамшөптердің суреттерін жинау және таңбалауды қамтиды. Модельді оқыту үшін нысанды анықтау тапсырмаларындағы дәлдігі мен өнімділігімен танымал YOLOv4 нейрондық желісі таңдалды.

Диссертациялық жұмыстың құрылымы мен көлемі: Бұл жұмыс кіріспеден, төрт бөлімнен, қорытындыдан, пайдаланылған әдебиеттер тізімінен тұрады, негізгі мәтіннің 59 беті, 13 сурет және бір кестеден тұрады. Пайдаланылған көздер тізімінде 51 дереккөз атауы бар.

Түйін сөздер: Машинамен оқыту, тереңдетіп оқыту, дәлме-дәл ауыл шаруашылығы, UAV (ұшқышсыз ұшу аппараты), деректер жинағы (DS), конволюциялық нейрондық желілер (CNN), YOLO (тек бір рет қарайсыз).

## **АННОТАЦИЯ**

Данная диссертационная работа посвящена разработке метода обнаружения сои и сорняков на грядках с использованием созданного собственного набора данных и обученной нейронной сети YOLOv4. Проблема обнаружения сои и сорняков является актуальной в агротехнике, и эффективные системы обнаружения могут значительно улучшить эффективность и управление сельскохозяйственными угодьями.

В работе был проведен анализ существующих методов и технологий в области обнаружения объектов на изображениях. В ходе работы рассматриваются теоретическая основа глубокого обучения, проблематика, сравнение наборов данных. Было выявлено ограничение существующих подходов в отношении сои и разнообразных видов сорняков. Для решения этой проблемы было предложено создание собственного набора данных. Методология работы включала сбор и разметку изображений сои и сорняков на грядках с использованием специализированных инструментов аннотации изображений. Для обучения модели была выбрана нейронная сеть YOLOv4, известная своей точностью и производительностью в задачах обнаружения объектов.

Структура и объем диссертации: Данная работа состоит из введения, четырех разделов, заключения, списка использованных источников содержит 59 страницы основного текста, 13 рисунков и одной таблицы. Список использованных источников содержит 51 наименование источников.

Ключевые слова: Машинное обучение (machine learning), глубокое обучение, точное земледелие, БПЛА (беспилотный летательный аппарат), набор данных (dataset - DS), Сверточные нейронные сети (Convolutional Neural Networks, CNN), YOLO(You Only Look Once).

## ANNOTATION

This dissertation work focuses on developing a method for detecting soybean and weeds in seedbeds using a generated proprietary dataset and a trained YOLOv4 neural network. The problem of soybean and weed detection is a hot topic in agricultural technology, and effective detection systems can significantly improve the efficiency and management of farmland.

This paper analyzed existing methods and technologies in the field of object detection in images. The theoretical basis of deep learning, problematics, and comparison of datasets are discussed. A limitation of existing approaches for soybean and a variety of weed species was identified. To address this limitation, the creation of a proprietary dataset was proposed. The methodology involved collecting and partitioning images of soybeans and weeds in beds using specialized image annotation tools. The YOLOv4 neural network, known for its accuracy and performance in object detection tasks, was chosen to train the model.

Structure and volume of the dissertation: This work consists of an introduction, four sections, conclusion, list of references contains 59 pages of the main text, 13 figures and one table. The list of references contains 51 names of sources.

Keywords: machine learning, deep learning, precision agriculture, UAV (unmanned aerial vehicle), dataset (DS), Convolutional Neural Networks (CNN), YOLO (You Only Look Once).

## **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ	8
1. Введение в машинное обучение	10
1.1 Подробное описание методов машинного обучения	12
1.2 Введение в глубокое обучение	15
1.3 БПЛА. Определение характеристики и применение	18
1.4 Сверточные нейронные сети	19
1.4.1 Сверточные слои (Convolutional Layer).	22
1.4.2 Карта признаков (Feature Map)	23
1.4.3 Пулинг (Pooling)	23
1.4.4 Полносвязные слои (Fully Connected Layer)	24
1.4.5 Прямое распространение ошибки (backpropagation)	25
1.4.6 Обратное распространение ошибки (backpropagation)	26
1.4.7 Методы оптимизации	27
1.5 Введение в область обнаружения объектов	28
1.5.1 Различные методы обнаружения объектов на изображениях	28
1.6 Заключение раздела	30
2. Обзор существующих работ по данной области	31
2.1 Литературный обзор	31
3. Практическая часть диссертационной работы	33
3.1 Анализ и сравнение похожих наборов данных	33
3.2 Создание собственного набора данных	35
3.3 Описание собственного набора данных	36
3.3.1 CVAT	37
3.4 Нейронная сеть YOLO	40
3.4.1 YOLOv4	41
3.5 Установка нейронной сети YOLOv4	42
3.6 Подробное описание компонентов, использованных при установке YOLOv4	44
3.6.1 OpenCV	44
3.6.2 Nvidia CUDA	45
3.6.3 CUDNN	46
3.6.4 CMake	47
3.6.5 Darknet	48
3.6.6 Как работают компоненты в контексте с нейронной сети YOLOv4	49
3.7 Обучение нейронной сети и результаты	50
3.8 Второе обучение нейронной сети и результаты	51
ЗАКЛЮЧЕНИЕ	54
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	56

## ВВЕДЕНИЕ

В последние годы, с развитием технологий компьютерного зрения, точное земледелие становится все более интересной и перспективной областью исследований. Технологии компьютерного зрения, основанные на робототехнике и искусственном интеллекте, предлагают возможности автоматизации и оптимизации различных задач в сельском хозяйстве. Они позволяют улучшить эффективность процессов производства, снизить затраты и повысить качество сельскохозяйственной продукции.

Однако одним из основных препятствий для успешного развертывания этих технологий является недостаток общедоступных наборов данных изображений. Создание и разметка больших объемов данных является трудоемким процессом, который требует значительных ресурсов и экспертных знаний. Без доступа к качественным и разнообразным наборам данных затрудняется быстрое создание прототипов и оценка алгоритмов компьютерного зрения.

Сельское хозяйство является одной из важнейших отраслей экономики, поэтому эффективные методы автоматизации процессов в этой сфере могут принести значительные преимущества и повысить эффективность производства. Создание собственного набора данных, специально адаптированного для обнаружения сои и сорняков на грядках, является важным шагом в направлении разработки и оптимизации алгоритмов компьютерного зрения для сельскохозяйственных задач.

**Актуальность темы.** Одной из важных проблем при разработке систем компьютерного зрения для обработки изображений, получаемых с борта БПЛА, является недостаток наборов данных, учитывающих специфику изображений, получаемых с низколетящих платформ. Для обучения глубоких нейронных сетей необходимы такие данные, но их создание требует много времени и планирования, а также различного оборудования, начиная от готовых коммерчески доступных камер до специально разработанных датчиков. Это приводит к тому, что данные разнородны и могут быть скудными, а также могут собираться для слишком специфических исследований, отличающихся по качеству, типу датчиков, количеству и виду.

**Объект исследования:** Набор данных, полученных с борта БПЛА (беспилотного летательного аппарата) в контексте точного земледелия. Грядки с культурой сои.

**Предмет исследования:** Формирование и разметка набора данных полученных с борта БПЛА для решения задач точного земледелия методами машинного обучения. Создание собственного набора данных и обучение нейронной сети YOLOv4 для обнаружения сои и сорняков на грядках.

**Цель.** Создание собственного набора данных для определения сои и сорняков на грядках и обучение нейронной сети YOLO (You Only Look Once) для автоматического обнаружения и классификации этих растений.



Данная диссертация нацелена на разработку и исследование метода определения сои и сорняков на грядках с использованием собственного набора данных и нейронной сети YOLO. Основной задачей является сбор разнообразных данных о сое и сорняках, включая изображения и соответствующие метки классов.

Собственный набор данных играет важную роль в определении сои и сорняков на грядках по нескольким причинам:

- Разнообразие классов: Собственный набор данных позволяет собрать информацию о разных разновидностях сои и сорняков, которые встречаются на грядках. Это позволяет модели обучаться на разнообразных примерах и повышает ее способность точно определять различные виды растений;
- Персонализированное обучение: Обучение модели на собственном наборе данных позволяет персонализировать ее под конкретные потребности и предпочтения. Модель может быть обучена на определенных признаках, которые важны для конкретной цели, и давать индивидуальные рекомендации и решения;
- Улучшение точности: Собственный набор данных позволяет улучшить точность модели, так как данные будут отражать реальные условия и проблемы, с которыми сталкиваются садоводы и фермеры. Это поможет снизить количество ложных срабатываний и повысить эффективность определения сои и сорняков;

В целом, создание собственного набора данных для определения сои и сорняков на грядках позволяет получить более точные и персонализированные результаты, адаптированные к конкретным условиям и потребностям.

#### **Основные задачи**

- Провести анализ основных подходов к решению задачи распознавания объектов на изображении;
- Исследовать понятия глубокое обучение, нейронные сети;
- Обзор существующих работ по данной области;
- Изучить и сравнить похожие базы (датасеты) и выявить их недостатки;
- Описание собственного набора данных;
- Создание собственного набора данных;
- Обучение нейронной сети на созданном наборе данных;

**Новизна.** Новизна исследования заключается в том, что сейчас малое количество набора данных в свободном доступе, для решения задач точного земледелия. Нехватка общедоступных наборов данных изображений остается ключевым препятствием в разработке компьютерного зрения и интеллектуальных систем нового поколения для точного сельского хозяйства. Несмотря на прогресс, достигнутый за последние несколько лет, необходимы значительные усилия для создания новых баз данных изображений, особенно для многих специфических областей применения, где до сих пор нет ни одной специализированной базы данных изображений.

## 1. Введение в машинное обучение

Машинное обучение (Machine Learning) - это раздел искусственного интеллекта, который позволяет компьютеру обучаться на основе данных и опыта, вместо того чтобы задавать жестко определенные правила для решения задач.

В основе машинного обучения лежит идея создания алгоритмов, которые могут находить закономерности в данных и использовать их для прогнозирования, классификации, кластеризации и других задач. Для этого используются различные методы, такие как регрессия, деревья решений, нейронные сети и другие.

Для обучения модели машинного обучения требуется большой объем данных, на которых модель может учиться, а также заданные правильные ответы (метки). После обучения модель может использоваться для решения новых задач, для которых она была обучена.

Вот некоторые ключевые аспекты машинного обучения:

Типы машинного обучения: Существует пять типов машинного обучения[11, 40, 41]. Обучение с учителем (supervised learning) предполагает наличие размеченных данных, где для каждого примера есть соответствующий правильный ответ или метка. Обучение без учителя (unsupervised learning) включает в себя работу с неразмеченными данными, где алгоритмы стремятся выявить скрытые структуры или паттерны. Полу-управляемое обучение (включая самообучение) - Semi-Supervised Learning (SSL) полу-управляемое обучение использует комбинацию размеченных и неразмеченных данных для обучения модели. Обучение с подкреплением (reinforcement learning) основано на обучении агента, который взаимодействует со средой и получает вознаграждение или наказание за определенные действия. Глубокое обучение (Deep Learning) — это подраздел машинного обучения, который использует нейронные сети с большим числом слоев для извлечения высокоуровневых абстракций из данных.

Эти методы машинного обучения решают разные типы задач таких как регрессии, классификации, кластеризации и снижения размерности данных (Рисунок 1).

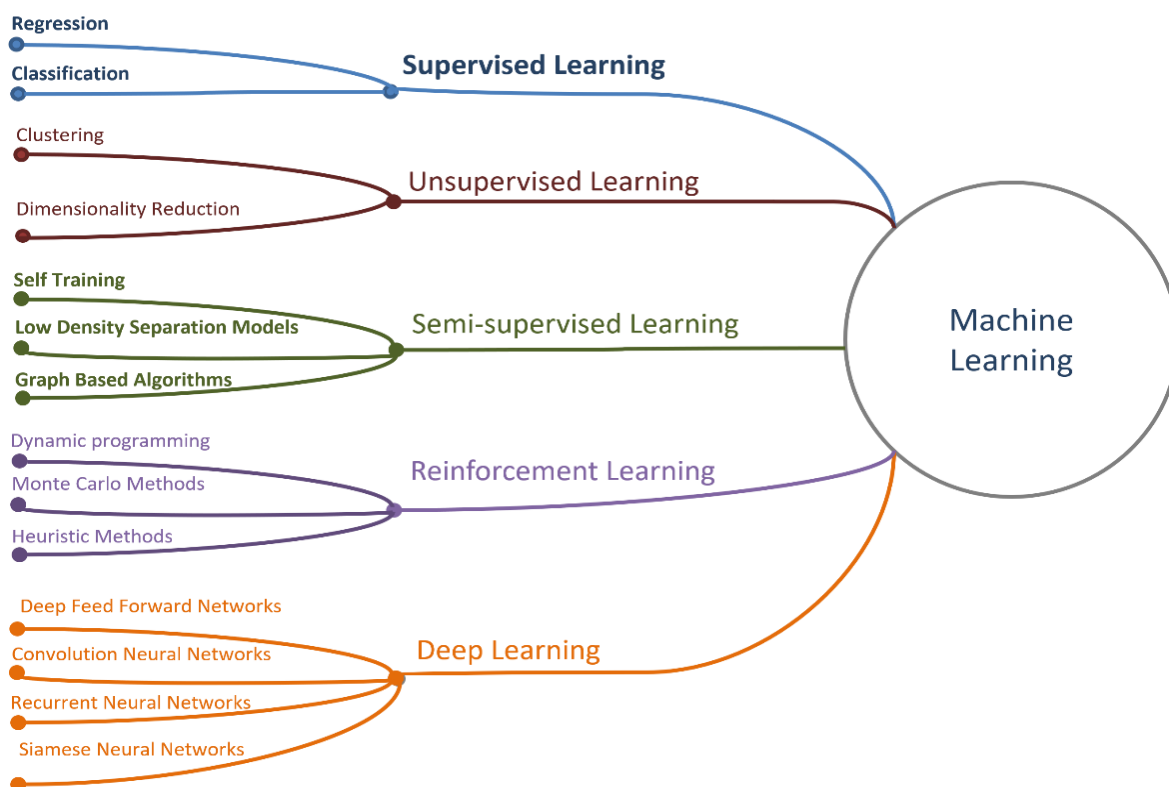


Рисунок 1. Основные методы машинного обучения

**Алгоритмы машинного обучения:** Существует множество алгоритмов машинного обучения, включая линейную регрессию, логистическую регрессию, деревья решений, случайные леса, метод опорных векторов, наивный Байесовский классификатор, алгоритмы кластеризации (например, k-средних), нейронные сети и т. д. Каждый алгоритм имеет свои преимущества, ограничения и области применения.

**Признаки и представление данных:** Хорошо подготовленные признаки данных играют важную роль в успешном применении машинного обучения. Преобразование и представление данных являются важной частью процесса подготовки данных. Это может включать шкалирование, нормализацию, отбор признаков, кодирование категориальных данных и т. д.

**Оценка моделей и обобщение:** Оценка производительности моделей машинного обучения является важной задачей. Это может включать разделение данных на обучающую и тестовую выборки, использование метрик оценки (например, точность, полнота, F1-мера), перекрестную проверку (cross-validation) и анализ показателей ошибок (например, матрицу ошибок).

**Гиперпараметры и настройка моделей:** Многие алгоритмы машинного обучения имеют гиперпараметры, которые не оптимизируются в процессе обучения модели. Они требуют настройки для достижения лучшей производительности модели. Это может включать выбор оптимального количества скрытых слоев нейронной сети, коэффициента регуляризации, скорости обучения и т. д.

**Развитие машинного обучения:** Машинное обучение активно развивается и постоянно появляются новые методы и алгоритмы. Некоторые из них

включают в себя глубокое обучение, обучение с подкреплением, обучение на больших данных (Big Data), обучение с активным участием и метаобучение. Исследователи и практики также работают над развитием методов объяснимого и интерпретируемого машинного обучения.

Машинное обучение широко применяется в различных областях, таких как медицина, финансы, обработка естественного языка, компьютерное зрение, автоматизация производства и многое другое. Оно имеет огромный потенциал для решения сложных задач и принятия умных решений на основе данных. Примеры задач, решаемых с помощью машинного обучения, включают автоматическое распознавание речи, классификацию изображений, определение кредитоспособности клиентов и прогнозирование погоды.

## **1.1 Подробное описание методов машинного обучения**

### **1. Обучение без учителя (Unsupervised Learning):**

В обучении без учителя модель обучается на неразмеченных данных, то есть данных, которые не имеют predetermined меток или целевых переменных. Основная цель этого подхода - найти скрытые структуры, паттерны или зависимости в данных.

В отличие от обучения с учителем, где модель учится на основе правильных ответов, обучение без учителя требует от модели самостоятельно находить внутренние закономерности в данных. Это позволяет модели обнаруживать новые знания и извлекать полезную информацию без прямого руководства.

Основные методы обучения без учителя:

**Кластеризация (Clustering):** Кластеризация — это процесс группировки схожих объектов внутри набора данных на основе их сходства. Методы кластеризации позволяют выделить группы объектов, которые имеют схожие характеристики или поведение. Примеры алгоритмов кластеризации включают k-средних, иерархическую кластеризацию и алгоритм DBSCAN.

**Понижение размерности (Dimensionality Reduction):** Понижение размерности используется для уменьшения размерности данных, сохраняя при этом наиболее значимые признаки. Это помогает упростить анализ данных и уменьшить вычислительную сложность модели. Основные методы понижения размерности включают главные компоненты (PCA), метод t-SNE и автоэнкодеры.

**Ассоциативные правила (Association Rules):** Ассоциативные правила используются для выявления скрытых связей и зависимостей между различными элементами данных. Они позволяют обнаружить, какие элементы данных часто встречаются вместе или какие правила связывают их. Примером алгоритма ассоциативных правил является алгоритм Apriori.

Обучение без учителя широко применяется в различных областях, таких как анализ данных, анализ социальных сетей, обработка естественного языка, компьютерное зрение и многое другое. Оно позволяет выявлять скрытые структуры и получать ценные практические выводы из больших объемов данных без необходимости разметки или руководства.

## 2. Обучение с учителем (Supervised Learning):

В обучении с учителем модель обучается на размеченных данных, где каждый пример данных имеет известную метку или целевую переменную. Цель состоит в том, чтобы построить модель, которая может предсказывать или классифицировать новые неразмеченные данные.

Основные компоненты обучения с учителем:

**Обучающий набор данных:** Это набор данных, состоящий из примеров, каждый из которых содержит набор признаков (входные данные) и соответствующую метку (выходные данные или целевая переменная). Примеры в обучающем наборе используются для обучения модели.

**Модель:** Модель представляет собой алгоритм или функцию, которая принимает на вход набор признаков и пытается предсказать соответствующую метку или целевую переменную. Модель обучается на обучающем наборе данных с целью минимизации ошибки предсказания.

**Функция потерь (Loss Function):** Функция потерь измеряет расхождение между предсказанными значениями модели и фактическими метками в обучающем наборе данных. Она является мерой ошибки модели и используется для оптимизации параметров модели в процессе обучения.

**Оптимизация:** Процесс оптимизации направлен на минимизацию функции потерь путем настройки параметров модели. Для этого применяются различные алгоритмы оптимизации, такие как градиентный спуск, которые позволяют находить оптимальные значения параметров, обеспечивающие наилучшую производительность модели.

**Тестирование:** После завершения процесса обучения модель проверяется на тестовом наборе данных, который не использовался в процессе обучения. Тестирование позволяет оценить производительность модели на новых, независимых данных и сделать выводы о ее способности обобщать.

Обучение с учителем используется во многих задачах, таких как классификация, регрессия, прогнозирование, обнаружение аномалий и других. Примеры алгоритмов обучения с учителем включают линейную регрессию, метод k-ближайших соседей, деревья решений, случайные леса, градиентный бустинг и нейронные сети.

## 3. Полу-управляемое обучение (Semi-Supervised Learning - SSL):

Полу-управляемое обучение объединяет элементы обучения с учителем и обучения без учителя. Оно использует комбинацию размеченных и неразмеченных данных для обучения модели. Основная идея состоит в том, чтобы использовать небольшое количество размеченных данных и большое количество неразмеченных данных для достижения лучшей производительности модели.

Основные идеи и методы полу-управляемого обучения:

Использование размеченных данных: В полу-управляемом обучении используются размеченные данные, где каждый пример имеет известную метку или целевую переменную. Эти данные используются для обучения модели на задачах классификации или регрессии.

Использование неразмеченных данных: Кроме размеченных данных, SSL также использует большое количество неразмеченных данных, где метки или целевые переменные неизвестны. Эти данные используются для извлечения скрытых структур, паттернов или зависимостей в данных, что помогает улучшить производительность модели.

Передача знаний: Одна из основных стратегий в SSL - это передача знаний из размеченных данных на неразмеченные данные. Модель, обученная на размеченных данных, может использоваться для предсказания меток неразмеченных данных. Полученные предсказанные метки могут быть использованы в дальнейшем обучении модели, включая их в обучающий набор данных.

Самообучение (Self-Training): Самообучение - это распространенный метод в SSL. Он начинается с обучения модели на размеченных данных. Затем модель используется для предсказания меток неразмеченных данных. Предсказанные метки добавляются к размеченным данным, и модель дообучается на обновленных данных. Процесс повторяется несколько раз до достижения требуемой производительности модели.

Построение графов: В SSL часто используется построение графов, где вершины представляют примеры данных, а ребра отражают их сходство или близость. Графы позволяют модели учесть структуру данных и использовать информацию из соседних примеров для улучшения предсказаний.

#### 4. Обучение с подкреплением (Reinforcement Learning - RL):

В обучении с подкреплением модель обучается принимать последовательность действий в определенной среде с целью максимизации некоторой награды или достижения заданной цели. Модель обучается через взаимодействие с окружающей средой, получая обратную связь в виде награды или штрафа. Агент стремится оптимизировать свои действия, чтобы получить максимальную награду. Обучение с подкреплением широко применяется в задачах игрового проектирования, управлении роботами и управлении ресурсами.

#### 5. Глубокое обучение (Deep Learning):

Глубокое обучение - это подраздел машинного обучения, который использует глубокие нейронные сети с большим числом слоев для извлечения высокоуровневых абстракций из данных. Глубокое обучение позволяет моделям автоматически изучать иерархические представления данных, что делает его особенно эффективным в обработке изображений, аудио, текста и видео. Более подробно про глубокое обучение будет рассмотрено в следующей главе.

## 1.2 Введение в глубокое обучение

Глубокое обучение - это раздел машинного обучения, который использует искусственные нейронные сети для анализа и решения сложных задач, таких как распознавание изображений, обработка естественного языка и принятие решений на основе больших объемов данных.

Основная идея глубокого обучения заключается в том, чтобы создать модель нейронной сети, которая может обучаться на большом количестве данных, чтобы автоматически выявлять иерархии признаков в данных. Это позволяет модели принимать более сложные решения на основе этих признаков и достигать более высокой точности в предсказании.

Для обучения нейронной сети требуется большой объем данных и достаточно мощных вычислительных ресурсов, поэтому глубокое обучение часто используется для решения задач в области компьютерного зрения, обработки естественного языка и распознавания речи.

Вот некоторые ключевые аспекты глубокого обучения:

**Искусственные нейронные сети:** Основой глубокого обучения являются искусственные нейронные сети (ИНС). Они моделируют работу нейронов в головном мозге, где каждый нейрон принимает входные данные, обрабатывает их и передает результаты следующему слою. Нейроны организованы в слои, и информация передается от входного слоя через скрытые слои до выходного слоя.

**Глубокие нейронные сети:** Глубокие нейронные сети имеют большое количество слоев, что отличает их от более простых нейронных сетей с одним или двумя скрытыми слоями. Глубокие модели могут изучать более сложные представления данных, позволяя автоматически извлекать признаки и анализировать данные на разных уровнях абстракции.

Глубокое обучение представляет собой процесс, в котором данные проходят через нейронные сети с множеством скрытых слоев для извлечения сложных характеристик и принятия решений. На рисунке 2, иллюстрировано простое объяснение основных этапов работы глубокого обучения:

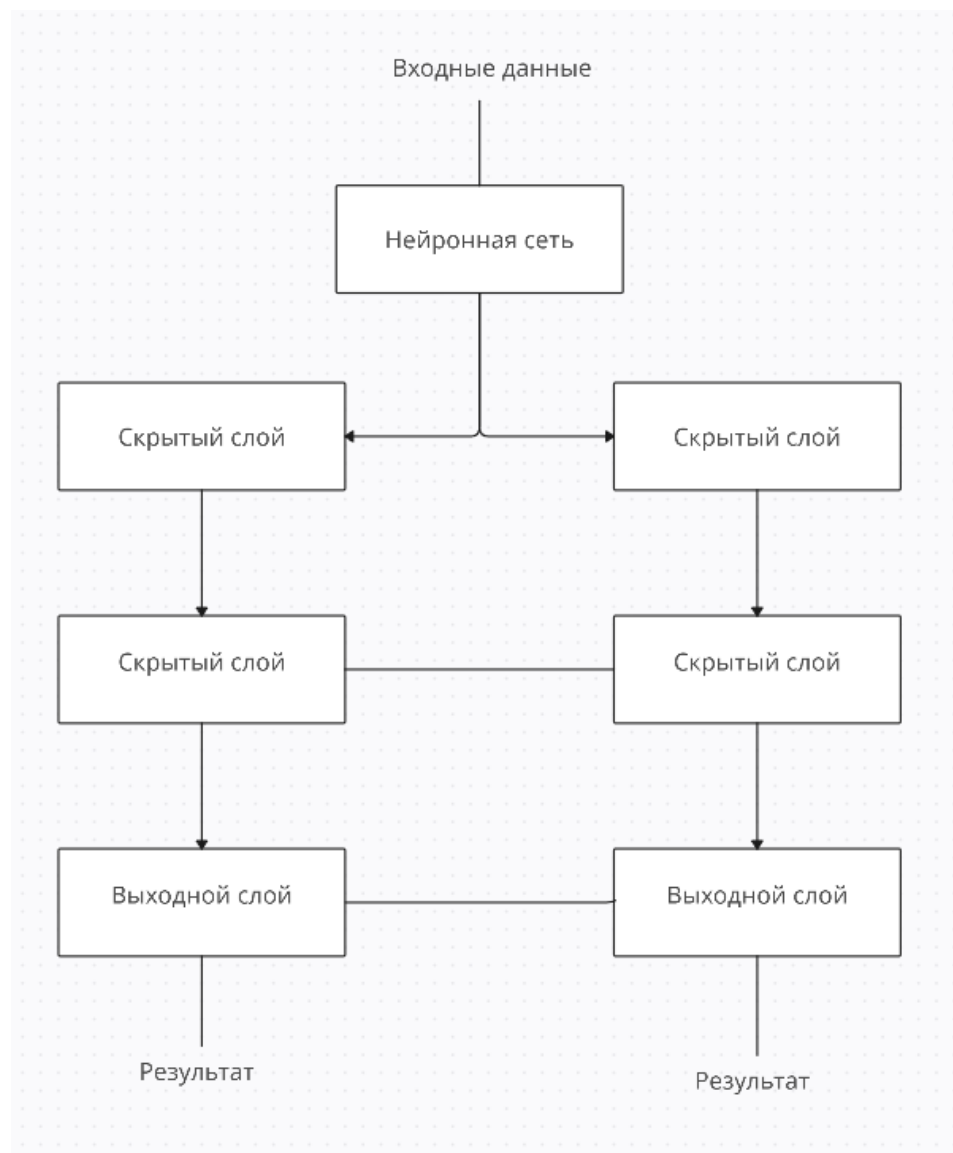


Рисунок 2. Основные этапы работы глубокого обучения

В этой схеме:

Входные данные — это данные, которые подаются на вход нейронной сети, например, изображения для обработки.

Скрытые слои — это слои нейронов между входным и выходным слоями. Каждый скрытый слой обрабатывает входные данные и передает результаты следующему слою.

Выходной слой — это последний слой нейронной сети, который выдает конечный результат или прогноз.

Результат — это вывод или прогноз, полученный от выходного слоя нейронной сети.

В процессе глубокого обучения каждый слой нейронной сети обрабатывает данные, используя веса и активационные функции. Веса определяют, как входные данные связываются с активацией нейронов в каждом слое, а активационные функции обеспечивают нелинейность и обработку сложных взаимодействий между нейронами.



Во время обучения нейронная сеть корректирует свои веса, используя алгоритмы оптимизации и функцию потерь, чтобы минимизировать ошибку между прогнозами и ожидаемыми выходными данными. Обучение может быть проведено с использованием различных методов, таких как обратное распространение ошибки (backpropagation).

Таким образом, глубокое обучение позволяет нейронной сети изучать иерархические представления данных, извлекать сложные признаки и принимать решения на основе обученных параметров и весов.

Архитектуры глубокого обучения: Существует несколько типов архитектур глубокого обучения, каждая из которых применяется в различных сферах. Сверточные нейронные сети (Convolutional Neural Networks, CNN) широко используются для анализа изображений и видео. Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) применяются для работы с последовательными данными, такими как текст и речь. Есть также автоэнкодеры (Autoencoders), генеративные состязательные сети (Generative Adversarial Networks, GAN) и многослойные перцептроны (Multilayer Perceptrons, MLP), среди других.

Обучение с использованием глубоких нейронных сетей: Обучение глубоких нейронных сетей включает в себя два основных этапа: прямое распространение и обратное распространение ошибки. Прямое распространение состоит в передаче входных данных через сеть для получения прогноза или выходных данных. Обратное распространение ошибки используется для оценки ошибки прогноза и корректировки весов нейронов, чтобы улучшить точность модели. Обучение глубоких нейронных сетей требует большого количества размеченных данных и может занимать много времени и ресурсов.

Применение глубокого обучения: Глубокое обучение нашло широкое применение во многих областях. Оно применяется для распознавания и классификации изображений, обработки естественного языка, распознавания речи, рекомендательных систем, анализа данных, генерации контента и др. В глубоком обучении активно исследуются различные задачи, такие как семантическая сегментация изображений, машинный перевод, генерация текста и распознавание объектов.

Преимущества и ограничения: Глубокое обучение обладает несколькими преимуществами, включая способность автоматического извлечения признаков, адаптивность к сложным задачам и обработку больших объемов данных. Однако у глубокого обучения также есть ограничения, включая необходимость большого количества размеченных данных, сложность интерпретации результатов и высокие требования к вычислительным ресурсам.

Глубокое обучение имеет множество применений в различных областях, таких как медицина, финансы, транспорт и многие другие. Например, нейронные сети могут использоваться для диагностики заболеваний, предсказания финансовых трендов или управления транспортными системами.

Глубокое обучение продолжает развиваться, исследователи постоянно работают над новыми методами, архитектурами и подходами для улучшения эффективности и расширения области его применения. Это динамичное поле, которое продолжает вносить значительный вклад в различные сферы науки и технологии.

### **1.3БПЛА. Определение характеристики и применение**

Беспилотный летательный аппарат (БПЛА) - это летательный аппарат без пилота-человека на борту и тип беспилотного транспортного средства. Беспилотные летательные аппараты (БПЛА) представляют собой составную часть беспилотных авиационных систем (БАС), которые включают в себя БПЛА, наземный контроллер и систему связи между ними. Полеты БПЛА могут быть осуществлены с различной степенью автономности: либо с удаленным управлением человеком-оператором, либо автономно с использованием встроенных компьютеров.

В сравнении с пилотируемыми самолетами, изначально БПЛА были задуманы для выполнения задач, которые слишком скучны, грязны или опасны для человека. Несмотря на то, что изначально БПЛА использовались в основном в военных целях, их применение быстро расширяется, включая коммерческие, научные, рекреационные, сельскохозяйственные и другие сферы, такие как охрана порядка и наблюдение, доставка продуктов, аэрофотосъемка.

Поскольку мировой спрос на производство продуктов питания растет экспоненциально, ресурсы истощаются, сельскохозяйственные угодья сокращаются, а сельскохозяйственная рабочая сила становится все более дефицитной, возникает острая необходимость в более удобных и умных сельскохозяйственных решениях, чем традиционные методы, и ожидается, что индустрия сельскохозяйственных дронов и робототехники будет прогрессировать.

В рамках данной диссертационной работы были использованы беспилотные летательные аппараты (БПЛА) для съемки грядок с соей и сорняками. БПЛА являются эффективным инструментом для сбора высококачественных изображений сельскохозяйственных угодий.

Съемка с использованием БПЛА позволяет получать воздушные снимки с высоким разрешением и обеспечивает широкий охват территории. Это позволяет исследователям получить детальные изображения грядок, сои и сорняков, что является важным для анализа и разработки методов обнаружения и классификации растений.

БПЛА оборудованы специальными камерами, такими как RGB-камеры или мультиспектральные камеры, которые позволяют получить многоцветные изображения или изображения в различных спектральных диапазонах. Это дает

возможность получать данные о растениях с разных спектральных каналов и использовать их для дальнейшего анализа и обработки.

С помощью БПЛА было выполнено съемка фотографий грядок с соей и сорняками, которые в дальнейшем были использованы для создания собственного набора данных и обучения модели YOLO. БПЛА позволили получить широкий спектр информации о состоянии грядок, включая плотность растений, распределение сорняков и общую структуру посадки сои.

Использование БПЛА для съемки сельскохозяйственных угодий предоставляет возможность более точного и детального анализа и контроля растений. Это способствует разработке эффективных методов управления сельскохозяйственными процессами и повышению производительности сельского хозяйства.

## **1.4 Сверточные нейронные сети**

Сверточные нейронные сети (Convolutional Neural Networks, CNN) являются мощным инструментом для обработки и анализа изображений. Они были успешно применены во многих областях, таких как распознавание образов, классификация изображений, детектирование объектов и сегментация изображений.

Основная идея сверточных нейронных сетей заключается в том, чтобы изучать признаки изображения на разных уровнях, начиная от простых признаков, таких как линии и углы, до более сложных признаков, таких как текстуры и формы объектов. Это достигается путем применения операции свертки к изображению, которая позволяет выделять локальные признаки в разных частях изображения.

Сверточная нейронная сеть обычно состоит из нескольких слоев, включая сверточные слои, слои пулинга и полносвязные слои. Сверточные слои используются для извлечения признаков изображения, путем применения набора фильтров к каждой части изображения. Фильтры обучаются автоматически в процессе обучения нейронной сети, чтобы выделять различные признаки, такие как границы, углы и текстуры.

Слои пулинга используются для уменьшения размера выходных данных от сверточных слоев и улучшения инвариантности к масштабированию и переносу объектов. Полносвязные слои принимают выходные данные из сверточных слоев и выполняют классификацию объектов на основе изученных признаков.

Сверточные нейронные сети (Convolutional Neural Networks, CNN) широко используются для обработки изображений и анализа визуальных данных. На рисунке 3, иллюстрировано простое объяснение основных этапов работы сверточной нейронной сети:

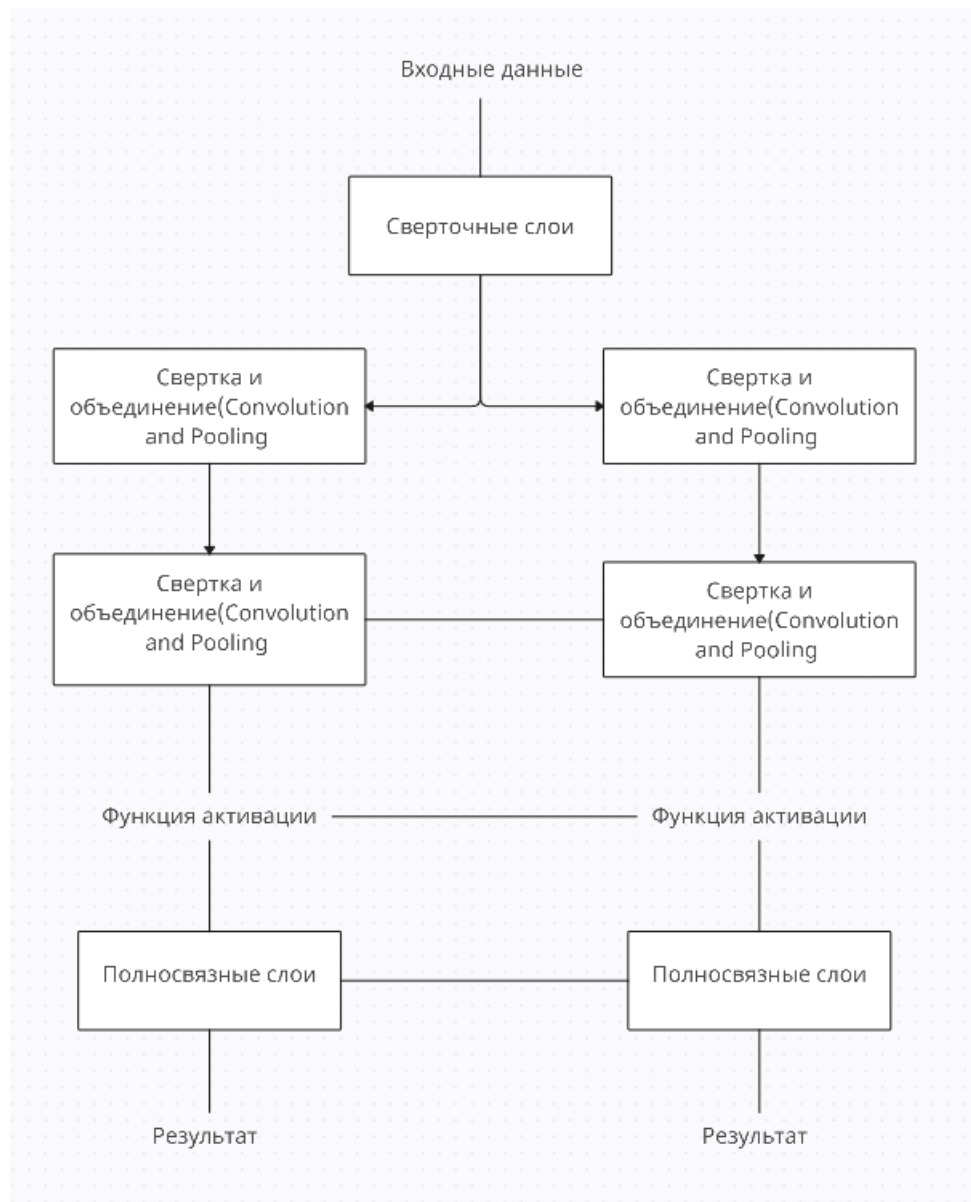


Рисунок 3. Основные этапы работы сверточной нейронной сети

В этой схеме:

Входные данные — это изображение или визуальные данные, подаваемые на вход сверточной нейронной сети.

Сверточные слои — это слои, выполняющие свертку и объединение (пулинг) операции для извлечения важных признаков изображения. Каждый сверточный слой имеет набор фильтров, которые скользят по изображению, выполняя свертку, и затем объединяются, чтобы уменьшить размерность данных и сгруппировать похожие признаки.

Функции активации — это функции, применяемые к результатам свертки, чтобы внести нелинейность в выходные данные.

Полносвязные слои — это слои, в которых каждый нейрон связан со всеми нейронами предыдущего слоя. Они выполняют классификацию или регрессию на основе извлеченных признаков.

Результат — это вывод или прогноз, полученный от последнего полносвязного слоя нейронной сети.

Сверточные нейронные сети эффективно работают с изображениями, так как сверточные слои позволяют извлекать локальные иерархические признаки, а пулинг слои уменьшают размерность данных, сохраняя важные характеристики изображения. Это помогает нейронной сети обучаться на более высоком уровне абстракции и достигать высокой производительности в задачах обработки изображений.

В процессе обучения сверточной нейронной сети используется метод обратного распространения ошибки (backpropagation), который позволяет корректировать веса нейронов и связей между ними на основе ошибки, которая возникает при сравнении предсказанных и фактических значений.

Для достижения высокой точности и производительности сверточных нейронных сетей используются различные техники, такие как регуляризация, дропаут, нормализация пакетов и т.д.

На рисунке 4 представлена сверточная нейронная сеть, которая принимает на вход матрицу изображения. Данные проходят через три последовательных сверточных слоя: Conv1 размером (2,3,3), Conv2 размером (5,2,4) и Conv3 размером (1,3). Использование сверточных сетей позволяет выделять сложные закономерности в представленных данных, которые не зависят от их расположения во входном векторе. Например, они способны выделять вертикальные или горизонтальные линии, точки, а также более сложные фигуры и объекты, такие как глаза, нос и другие.[11]

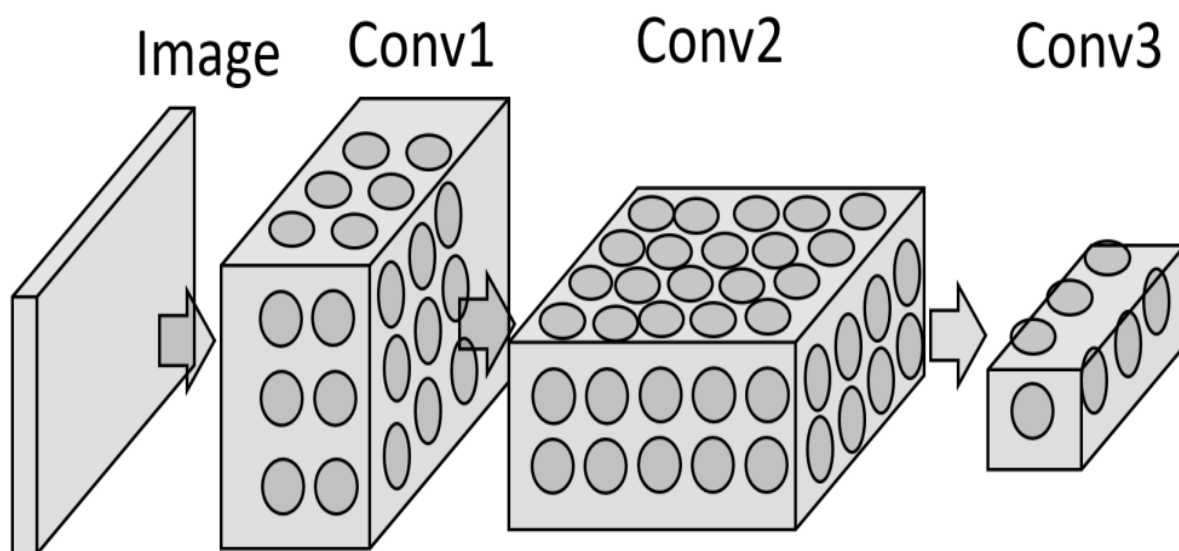


Рисунок 4. Сверточная нейронная сеть (Convolution neural network- CNN)

В целом, сверточные нейронные сети представляют собой мощный инструмент для обработки изображений и достигают высокой точности в задачах классификации, детектирования и сегментации объектов на изображениях. Они успешно применяются во многих областях, таких как

медицина, автомобильная промышленность, анализ изображений со спутников и многие другие.

Одним из наиболее популярных примеров применения сверточных нейронных сетей является сеть VGG (Visual Geometry Group) для классификации изображений. Эта сеть имеет 19 слоев и обучалась на наборе данных ImageNet, который содержит более миллиона изображений. Сеть VGG достигает высокой точности в задаче классификации изображений, превосходя результаты предыдущих алгоритмов.

Другой пример - сеть YOLO (You Only Look Once) для детектирования объектов на изображениях. Она используется для быстрого и точного обнаружения объектов в реальном времени, имеет высокую скорость обработки и точность детектирования.

Существуют и другие модификации сверточных нейронных сетей, такие как ResNet, Inception и другие, которые оптимизируют производительность и точность в различных задачах обработки изображений.

Таким образом, сверточные нейронные сети являются мощным инструментом для обработки и анализа изображений, и продолжают привлекать внимание исследователей и разработчиков в различных областях применения.

### **1.4.1 Сверточные слои (Convolutional Layer)**

Сверточный слой (Convolutional Layer) является основным строительным блоком сверточных нейронных сетей, который позволяет обнаруживать локальные признаки в изображениях.

Сверточный слой принимает на вход тензор изображения и применяет к нему фильтры (ядро свертки), которые проходят по изображению и выполняют операцию свертки. Фильтры могут быть различного размера и определяют, какие признаки будут извлекаться из изображения.

В результате применения фильтров к изображению получается новый тензор, называемый картой признаков (Feature Map), который представляет собой локальные признаки изображения. Каждый элемент карты признаков соответствует определенному участку изображения и является результатом операции свертки фильтра с соответствующим фрагментом изображения.

В сверточных слоях также используются операции пулинга (Pooling), которые уменьшают размер карты признаков и повышают инвариантность к масштабу и перемещению объектов на изображении.

Одним из преимуществ сверточных слоев является то, что они обладают локальной связностью. Это означает, что каждый элемент карты признаков зависит только от небольшой части изображения, а не от всего изображения в целом. Это позволяет сохранять пространственную структуру изображения и извлекать локальные признаки, такие как грани, текстуры и другие.

Сверточные слои в сверточных нейронных сетях обычно объединяются с полносвязными слоями (Fully Connected Layer), которые позволяют связать признаки из разных карт признаков и выполнить окончательную классификацию или сегментацию объектов на изображении.

В целом, сверточные слои являются ключевым элементом сверточных нейронных сетей и позволяют достичь высокой точности в задачах обработки и классификации изображений.

### **1.4.2 Карта признаков (FeatureMap)**

Карты признаков (Feature Maps) являются одним из основных понятий в сверточных нейронных сетях. Каждый сверточный слой в нейронной сети создает набор карт признаков, которые представляют собой набор двумерных массивов.

Каждый элемент карты признаков отвечает за определенный признак на изображении, например, цвет, форму, текстуру и т.д. Количество карт признаков в каждом слое определяется числом фильтров (Filters), которые применяются к входной картинке. Каждый фильтр обучается находить конкретный признак на изображении.

При прохождении входного изображения через сверточный слой, каждый фильтр сканирует входную картинку построчно, находя места, где признаки находятся наиболее сильно. Затем, с помощью операции свертки (Convolution) каждый фильтр создает карту признаков, где каждый элемент карты признаков соответствует значению суммы операций свертки в каждой точке изображения.

Каждый последующий сверточный слой получает на вход карты признаков предыдущего слоя и вычисляет новые карты признаков на основе существующих. При этом количество карт признаков может увеличиваться или уменьшаться в зависимости от параметров сверточного слоя.

Карты признаков играют ключевую роль в распознавании объектов на изображении, поскольку они содержат информацию о наличии и расположении различных признаков на изображении. Чем больше количество карт признаков и чем глубже нейронная сеть, тем выше точность распознавания объектов на изображении.

### **1.4.3 Пулинг (Pooling)**

Операция пулинга (Pooling) в сверточных нейронных сетях используется для уменьшения размера карты признаков (Feature Map), что помогает уменьшить количество параметров в нейронной сети и ускорить обучение.

Операция пулинга заключается в разбиении карты признаков на непересекающиеся прямоугольные области (обычно размером 2x2 или 3x3 пикселя) и выборе максимального (Max Pooling) или среднего (Average Pooling) значения из каждой области.

Применение операции Max Pooling позволяет сохранить наиболее значимые признаки из каждой области, что помогает повысить инвариантность к масштабу и перемещению объектов на изображении. Операция Average Pooling также может использоваться для уменьшения размера карты признаков, но не обладает такой же способностью к сохранению значимых признаков.

Результаты применения фильтров объединения - максимального (max pooling) и среднего (average pooling) - представлены на рисунке 5 и рисунке 6 соответственно. В данном примере используется пулинг-фильтр размером 3x3 и шаг (stride) равный 3.[11]

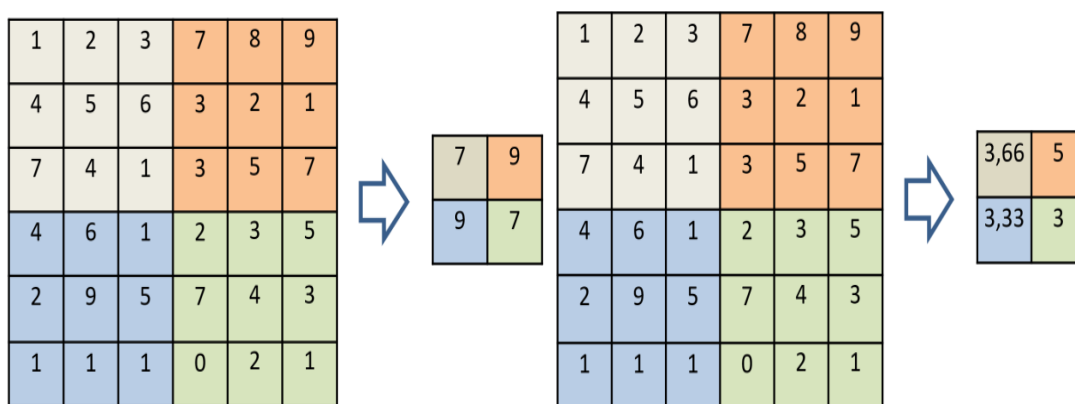


Рисунок5. Max Pooling Рисунок6. Average Pooling

Применение операции пулинга после сверточного слоя позволяет уменьшить размер карты признаков в  $n$  раз (где  $n$  - размер пулинговой области), что может существенно уменьшить количество параметров в нейронной сети и ускорить обучение. Операция пулинга также помогает устранить шум и сгладить изображение, что может повысить качество распознавания объектов.

Однако, слишком частое использование операции пулинга может привести к потере важных признаков изображения и ухудшению качества распознавания объектов. Поэтому, при проектировании сверточных нейронных сетей необходимо учитывать оптимальное соотношение между количеством сверточных и пулинговых слоев, чтобы достичь максимальной точности распознавания объектов на изображении.

#### 1.4.4 Полносвязные слои (Fully Connected Layer)

Полносвязные слои (Fully Connected Layers) в сверточных нейронных сетях представляют собой классические нейронные слои, которые



используются для классификации изображений или выполнения других задач, например, обнаружение объектов на изображении или распознавание речи.

Полносвязные слои принимают на вход выходные данные последнего сверточного слоя, который содержит карты признаков, и преобразуют их в одномерный вектор. Затем каждый элемент вектора проходит через набор нейронов в полносвязном слое, каждый из которых выполняет линейное преобразование входного вектора с помощью весов и смещений. Далее, результат преобразования проходит через функцию активации, например, сигмоидальную или ReLU-функцию, которая добавляет нелинейность в вычисления.

Размерность выходного вектора полносвязного слоя определяется числом нейронов в слое. Количество нейронов в полносвязном слое может варьироваться в зависимости от требований конкретной задачи. В задачах классификации изображений, например, количество нейронов в полносвязном слое соответствует числу классов, на которые нужно разделить изображения.

Полносвязные слои позволяют извлекать более сложные признаки из карт признаков, полученных в сверточных слоях, и выполнять конечную классификацию изображений или другие задачи. Они также являются наиболее вычислительно затратным компонентом в нейронной сети, поэтому часто используются в сочетании с методами оптимизации, такими как Dropout и Batch Normalization, для улучшения обучения и уменьшения переобучения модели.

#### **1.4.5 Прямое распространение ошибки (forward propagation)**

Прямое распространение ошибки (forward propagation) является одной из ключевых операций в обучении нейронных сетей и глубокого обучения. Он представляет собой процесс передачи входных данных через сеть для генерации прогноза или выходных данных.

Вот основные шаги прямого распространения ошибки:

**Входные данные:** На первом слое нейронной сети располагается входной слой, который принимает входные данные. Это могут быть пиксели изображения, значения признаков или любые другие входные значения.

**Вычисления на скрытых слоях:** Входные данные передаются через скрытые слои нейронной сети. Каждый нейрон в скрытых слоях принимает входные значения, выполняет линейную комбинацию с весами и применяет нелинейную активационную функцию. Это позволяет сети моделировать нелинейные зависимости в данных.

**Пропагация через слои:** Результаты вычислений на каждом слое передаются в следующий слой сети. Таким образом, информация постепенно распространяется через все слои сети, от входного слоя к выходному слою.

Вычисление выходных значений: Когда данные достигают выходного слоя, происходит вычисление прогноза или выходных значений нейронной сети. Обычно на выходном слое применяется активационная функция, соответствующая типу задачи (например, сигмоидная функция для бинарной классификации или softmax для многоклассовой классификации).

Вычисление ошибки: После получения выходных значений сравнивают их с правильными ответами (метками) для вычисления ошибки. Ошибка может быть измерена с помощью различных функций потерь, таких как среднеквадратичная ошибка (MSE) или перекрестная энтропия.

Процесс прямого распространения ошибки повторяется для каждого примера в обучающем наборе данных. Затем, в обратном распространении ошибки (backpropagation), эта ошибка будет использоваться для обновления весов нейронов, с целью улучшения прогнозов сети.

#### **1.4.6 Обратное распространение ошибки (backpropagation)**

Обратное распространение ошибки (backpropagation) - это алгоритм, который используется для обучения нейронных сетей методом градиентного спуска. Он позволяет вычислить градиент ошибки функции потерь по каждому весу в сети, и использовать этот градиент для обновления весов таким образом, чтобы минимизировать ошибку сети.

Процесс обратного распространения ошибки начинается с прямого прохода по нейронной сети, когда данные подаются на вход сети, проходят через все слои и вычисляется выходной результат. Затем, вычисляется ошибка функции потерь, которая является мерой расхождения между выходом сети и желаемым выходом.

Далее, градиент ошибки вычисляется путем применения цепного правила дифференцирования к каждому весу в сети. То есть, градиент ошибки на выходном слое вычисляется сначала, а затем он распространяется обратно через все слои до входного слоя, где вычисляется градиент ошибки по каждому входному признаку.

После вычисления градиентов ошибки для всех весов, они используются для обновления весов с помощью метода оптимизации, такого как градиентный спуск. Этот процесс повторяется для каждого батча данных в процессе обучения.

Обратное распространение ошибки является ключевым алгоритмом для обучения глубоких нейронных сетей и позволяет эффективно настраивать параметры сети для достижения высокой точности на задачах классификации, распознавания объектов и других задачах машинного обучения.

### 1.4.7 Методы оптимизации

Методы оптимизации в машинном обучении предназначены для настройки параметров нейронной сети с целью минимизации функции потерь на обучающих данных. Функция потерь описывает разницу между предсказанными и правильными ответами на задачу обучения, например, на задачу классификации изображений.

Некоторые из наиболее распространенных методов оптимизации, используемых в сверточных нейронных сетях, включают:

Градиентный спуск (Gradient Descent) - это метод оптимизации, который находит минимум функции потерь путем поиска направления, в котором функция уменьшается быстрее всего. Градиентный спуск вычисляет градиент функции потерь по параметрам нейронной сети и изменяет параметры в направлении, противоположном градиенту.

Адаптивный градиентный спуск (Adaptive Gradient Descent) - это метод оптимизации, который изменяет скорость обучения (learning rate) в зависимости от изменений в градиенте функции потерь. Это позволяет более эффективно оптимизировать параметры нейронной сети и ускорить сходимость.

Adam (Adaptive Moment Estimation) - это метод оптимизации, который объединяет преимущества градиентного спуска и адаптивного градиентного спуска. Adam также использует моменты первого и второго порядка градиента для адаптивного изменения скорости обучения и обеспечения более стабильной сходимости.

Dropout - это метод оптимизации, который помогает бороться с переобучением нейронной сети. Dropout удаляет случайно выбранные нейроны из сети во время обучения, что приводит к более робастной и эффективной нейронной сети.

Batch Normalization - это метод оптимизации, который нормализует входные данные каждого слоя в нейронной сети и помогает сделать обучение более стабильным. Batch Normalization позволяет быстрее сходить к оптимальным значениям параметров нейронной сети и улучшает ее общую производительность.

L1 и L2 регуляризация - это методы оптимизации, которые помогают контролировать переобучение нейронной сети. L1 и L2 регуляризация добавляют штраф к весам нейронной сети, чтобы ограничить их величину и предотвратить слишком большие значения. Это позволяет избежать переобучения нейронной сети на обучающих данных и улучшить ее обобщающую способность.

Momentum - это метод оптимизации, который добавляет инерцию к обновлению параметров нейронной сети. Momentum позволяет ускорить сходимость и уменьшить колебания при обновлении параметров.

Кроме того, существует множество модификаций и комбинаций вышеупомянутых методов оптимизации, которые позволяют более эффективно настраивать параметры нейронной сети. Каждый из этих методов имеет свои преимущества и недостатки, и выбор конкретного метода зависит от задачи и структуры нейронной сети.

## **1.5 Введение в область обнаружения объектов**

Обнаружение объектов на изображениях — это задача компьютерного зрения, которая заключается в автоматическом обнаружении и локализации специфических объектов или классов объектов на цифровых изображениях или видео. Она включает в себя определение границ объектов и их местоположения в пространстве изображения.

Обнаружение объектов имеет огромное значение во многих областях, включая сельское хозяйство. В сельском хозяйстве задачи обнаружения объектов включают поиск и классификацию растений, животных, вредителей, болезней растений и других факторов, которые могут влиять на урожай и здоровье растений.

Одной из важных задач в сельском хозяйстве является мониторинг урожайных полей. Обнаружение объектов позволяет автоматически определять и классифицировать растения и сорняки на полях, что помогает фермерам принимать решения по уходу за культурами и оптимизировать процессы обработки почвы, полива и удобрения. Также обнаружение объектов может помочь в определении заболеваний растений и позволить раннюю диагностику, что позволяет своевременно предпринимать меры по их контролю и предотвращению распространения.

С помощью современных методов обнаружения объектов, основанных на глубоком обучении и нейронных сетях, стало возможным достичь высокой точности и эффективности в решении задачи обнаружения объектов на изображениях. Эти методы используются в различных областях, включая медицину, автомобильную промышленность, безопасность, робототехнику и многое другое.

### **1.5.1 Различные методы обнаружения объектов на изображениях**

Различные методы обнаружения объектов на изображениях включают в себя методы основанные на признаках, методы на основе машинного обучения и нейросетевые методы. Давайте рассмотрим каждый из этих подходов подробнее:

1. Методы основанные на признаках:

Методы основанные на признаках используют предварительно определенные характеристики или свойства объектов для их обнаружения. Некоторые популярные методы в этой категории включают:

Метод Хаара: Этот метод использует вейвлет-подобные признаки, известные как фильтры Хаара, для обнаружения объектов. Он широко применяется в обнаружении лиц.[35]

Метод Хога (Histogram of Oriented Gradients): Этот метод вычисляет гистограммы направленных градиентов в различных регионах изображения и использует их для обнаружения объектов. Он широко используется для обнаружения пешеходов и автомобилей.[35]

## 2. Методы на основе машинного обучения:

Методы на основе машинного обучения используют алгоритмы обучения на размеченных данных для обнаружения объектов. Некоторые популярные методы в этой категории включают:

Метод опорных векторов (Support Vector Machines, SVM): SVM обучается на размеченных данных и строит гиперплоскость, разделяющую объекты разных классов. Он может использоваться для обнаружения объектов на основе различных признаков.

Алгоритмы случайного леса (Random Forest): Алгоритм случайного леса комбинирует множество решающих деревьев для обнаружения объектов. Каждое дерево обучается на подмножестве данных и признаков.

## 3. Нейросетевые методы:

Нейросетевые методы используют глубокие нейронные сети для обнаружения объектов. Они обеспечивают высокую точность и могут обучаться на больших объемах данных. Некоторые популярные нейросетевые архитектуры для обнаружения объектов включают:

R-CNN (Region-based Convolutional Neural Networks): R-CNN представляет собой метод, в котором изображение разделяется на регионы, которые затем классифицируются с использованием сверточных нейронных сетей. R-CNN был одним из первых методов, который показал высокую точность обнаружения объектов.

Faster R-CNN — это метод обнаружения объектов в изображениях, который комбинирует сверточные нейронные сети (Convolutional Neural Networks, CNN) для извлечения признаков и Region Proposal Network (RPN) для генерации регионов-предложений (region proposals) на изображении.

SSD (Single Shot MultiBox Detector): SSD предлагает эффективное решение для обнаружения объектов в режиме реального времени. Он использует сверточные слои с разными масштабами, чтобы обнаруживать объекты разных размеров.

YOLO (You Only Look Once): YOLO представляет собой ещё один популярный метод обнаружения объектов в реальном времени. Он использует сверточные нейронные сети, чтобы предсказывать границы объектов и их классы в одном проходе через сеть. YOLO имеет высокую скорость работы, но может потерять некоторую точность по сравнению с другими методами.

Mask R-CNN: Mask R-CNN расширяет архитектуру R-CNN, позволяя не только обнаруживать объекты, но и выполнять сегментацию пикселей для каждого обнаруженного объекта. Это полезно, когда требуется точная сегментация объектов на изображении.

EfficientDet: EfficientDet представляет собой эффективную архитектуру для обнаружения объектов с использованием сверточных нейронных сетей. Он достигает высокой точности при меньшем количестве параметров, что делает его эффективным в ресурсоемких сценариях.

Это лишь несколько примеров нейросетевых методов, применяемых в задаче обнаружения объектов. Важно отметить, что с развитием глубокого обучения и нейросетей, появляются новые архитектуры и улучшения в области обнаружения объектов, приводя к ещё более точным и эффективным результатам.

## **1.6 Заключение раздела**

Общее заключение состоит в том, что глубокое обучение и сверточные нейронные сети, в частности, являются мощным инструментом для решения различных задач в обработке изображений, видео, звука и других областях.

Нейронные сети позволяют извлекать сложные признаки из данных, автоматически настраивать параметры и достигать высокой точности на различных задачах, таких как распознавание объектов, классификация изображений, сегментация и т.д. Однако, выбор подходящей структуры нейронной сети, методов оптимизации и регуляризации может существенно влиять на ее производительность и точность. Поэтому необходимо тщательно анализировать данные и задачу, выбирать подходящие алгоритмы и методы оптимизации, и проводить многочисленные эксперименты для достижения наилучших результатов. Также, важным аспектом является подготовка данных, так как качество обучения нейронной сети напрямую зависит от качества данных и их разнообразия. Необходимо проводить предварительную обработку данных, такую как нормализация, аугментация и т.д., а также использовать достаточно большой набор данных для обучения, чтобы избежать переобучения и обеспечить хорошую обобщающую способность.

## **2. Обзор существующих работ по данной области**

В рамках исследования, направленного на создание собственного набора данных сои и сорняков и обучение нейронной сети YOLO, следует обратить внимание на ряд существующих работ в данной области. Литературные источники, представляющие собой ценные ресурсы, которые могут дать обзор и подробности о различных методах и техниках, используемых в данной предметной области.

Один из важных вкладов в область нейронных сетей и компьютерного зрения был сделан Redmon и Farhadi (2018) с представлением модели YOLOv3, которая обеспечивает значительные улучшения в задачах обнаружения объектов.[17] Работа Liu et al. (2016) предлагает альтернативный подход с использованием модели SSD (Single Shot Multibox Detector), которая также позволяет эффективно обнаруживать объекты на изображениях.[18]

Существующие наборы данных, такие как ImageNet (Russakovsky et al., 2015)[19] и Soybean (UCI Machine Learning Repository), предоставляют основу для исследований в области определения сои и сорняков.[25] Кроме того, работы, такие как Li et al. (2019), Marques et al. (2018) и Abdullah и Musa (2020), предлагают обзоры и анализ методов использования глубокого обучения в сельском хозяйстве и обнаружении сорняков.[20, 21, 22]

Исследования, проведенные González-de-Soto et al. (2019) и Simonyan и Zisserman (2014), также предлагают полезные инсайты в области применения глубокого обучения и обнаружения объектов.[23, 24] Кроме того, работы Everingham et al. (2010) и Long et al. (2015) исследуют вопросы семантической сегментации и вызывают интерес в контексте разработки моделей для определения и классификации сои и сорняков.[26, 27]

### **2.1 Литературный обзор**

Определение сои и сорняков на грядках является важной задачей в сельском хозяйстве, которая может быть автоматизирована с использованием методов компьютерного зрения и глубокого обучения. В данном литературном обзоре рассматриваются существующие исследования, связанные с созданием собственного набора данных сои и сорняков и обучением нейронной сети YOLO для эффективного обнаружения и классификации этих растений.

1) Redmon и Farhadi (2018) представили модель YOLOv3, которая является одной из самых популярных архитектур для обнаружения объектов в реальном времени. Их работа внесла значительный вклад в область нейронных сетей и компьютерного зрения, и может быть использована в разработке системы обнаружения сои и сорняков на грядках.[17]

2) Liu et al. (2016) предложили модель SSD (Single Shot Multibox Detector), которая также позволяет эффективно обнаруживать объекты на изображениях. Эта работа представляет альтернативный подход к обнаружению объектов и может быть применена в задаче определения сои и сорняков.[18]

3) Исследование Li et al. (2019) предлагает обзор методов использования глубокого обучения для обнаружения и классификации сорняков в сельском хозяйстве. Они рассматривают различные подходы, включая сверточные нейронные сети, и обсуждают их преимущества и ограничения.[20]

4) Marques et al. (2018) провели обзор применения глубокого обучения в сельском хозяйстве. Они описывают различные задачи, включая обнаружение сорняков, и представляют существующие методы и решения, использующие нейронные сети для решения этих задач.[21]

5) Abdullah и Musa (2020) представили обзор автоматизированного обнаружения и классификации сорняков с использованием глубокого обучения. Они описывают различные алгоритмы и методы, а также рассматривают проблемы и вызовы, связанные с этой задачей.[22]

6) González-de-Soto et al. (2019) провели исследование по использованию глубокого обучения для борьбы с сорняками в сельском хозяйстве. Они представили различные подходы, включая сверточные нейронные сети, и обсудили их эффективность и применимость в реальных условиях.[23]

Литературный обзор указывает на значительное количество исследований, связанных с определением сои и сорняков на грядках с использованием глубокого обучения. Модель YOLO и разные архитектуры представляют эффективные методы обнаружения объектов, которые могут быть применены для решения этой задачи. Представленные обзоры и исследования глубокого обучения в сельском хозяйстве подчеркивают значимость и актуальность разработки собственного набора данных сои и сорняков, и их обучения на нейронной сети YOLO для достижения более точных результатов определения и классификации этих растений на грядках.



### **3. Практическая часть диссертационной работы**

Основные задачи, которые были реализованы, включают в себя анализ и сравнение наборов данных, создание собственного набора данных с использованием специализированного инструмента CVAT, изучение и установку нейронной сети YOLO, а также обучение нейронной сети YOLO на созданном наборе данных.

В рамках исследования был проведен анализ нескольких существующих наборов данных, включая сравнение и выявление недостатков. Были проанализированы такие характеристики наборов данных, как объем, разнообразие объектов, разрешение изображений и другие. На основе проведенного анализа были определены преимущества и недостатки каждого набора данных, что помогло определить основные требования к созданию собственного набора данных.

Для создания собственного набора данных был использован специализированный инструмент CVAT. Были определены требования к данным и произведена разметка объектов на изображениях с помощью CVAT.

Для обучения нейронной сети на созданном наборе данных была изучена и установлена нейронная сеть YOLO. Были изучены основные принципы работы нейронной сети и ее архитектура.

#### **3.1 Анализ и сравнение похожих наборов данных**

Во время исследования разных видов наборов данных была сделано краткое описание каждого датасета характеристики, количество изображений а также их недостатками и было в собрано в таблицу 1.

Таблица 1.  
Описание похожих наборов данных

Название	Характеристики и описание	Кол-во изображений	Недостатки
<b>Sugar Beets 2016 dataset</b>	Набор данных по сахарной свекле ранняя попытка использования полевого робота, оснащенного множеством датчиков для получения данных для борьбы с сорняками. четырехканальных RGB и NIR изображений, которые имеют размер 1296 × 966 пикселей в формате png с аннотациями на уровне пикселей.	>10000	Был снят не с дрона
<b>The DeepWeeds dataset</b>	Этот набор данных включает в себя изображения RGB, которые имеют формат jpg и размером 256 × 256 пикселей, собранные специализированным для восьми видов сорняков и различных несорняковых растений в естественных полевых условиях без контроля освещения. Каждый вид сорняков имеет более 1000 изображений.	>10000	без предоставления аннотаций на уровне пикселей
<b>Ladybird Cobbitty Brassica dataset</b>	Набор данных был собран с помощью автономным роботом Ladybird. Содержит данные мультимодального зондирования как сельскохозяйственных культур, так и окружающей среды. Ежедневное сканирование проводилось для цветной капусты и овощей брокколи в течение 10 недель с момента посадки до сбора урожая. Содержит неизвестное количество изображений но вес датасета составляет 2.9 ТБ	>10000	не аннотированы ни на уровне изображений, ни на уровне пикселей.
<b>Carrot-Weed dataset</b>	Содержит изображения RGB, которые имеют формат jpg и разрешение 1296 × 966 пикселей., собранные на морковном поле, но с использованием недорогой камеры телефона в условиях естественного освещения.	39	Маленький датасет Использование камеры телефона
<b>Plant seedlings dataset</b>	Набор данных проростков растений содержит изображения RGB формата png разного размера, которые были получены от проростков растений, принадлежащих к 12 видам сельскохозяйственных культур и сорняков, в несколько раз в течение 20-дневного периода роста. Без подробной аннотации сорняков.	407	Без подробной аннотации сорняков, этот набор данных не подходит для задач сегментации или обнаружения сорняков.
<b>CWFI dataset</b>	Мультиспектральная камера, установленная на автономном полевом роботе BoniRob. Набор данных состоит из необработанных изображений морковного поля с разрешением 1296 × 966 пикселей в формате png с аннотацией	60	Всего 60 изображений. Маленький датасет

Также был сделан дополнительный обзор некоторых наборов данных. Существуют различные наборы данных, которые можно использовать для определения сои и сорняков на грядках. Вот сравнение некоторых из них:

1) Датасет Soybean (UCI Machine Learning Repository)[25]:

Размер: Небольшой.

Классы: содержит данные о шести различных классах сои и сорняка.

Особенности: включает различные числовые признаки, такие как площадь листа, длина и ширина листа, а также категориальные признаки, такие как цвет и форма листа.

Ограничения: относительно небольшой размер выборки и ограниченное количество классов.

Недостатки: Ограниченное количество классов. Данный набор данных содержит информацию только о шести классах сои и сорняков. Если требуется более широкий спектр классов, этот набор данных может быть ограничен.

2) Датасет Weed Species Identification (Kaggle)[36]:

Размер: Средний.

Классы: включает данные о нескольких классах сорняков.

Особенности: содержит изображения сорняков, а также метки классов. Можно использовать для обучения моделей компьютерного зрения.

Ограничения: не содержит информации о сои и ограничен по количеству классов.

Недостатки: Отсутствие информации о сои. Этот набор данных фокусируется исключительно на сорняках и не содержит информации о сое. Если требуется определение и сои, и сорняков, этот набор данных не будет подходящим.

3) Датасет Weed Detection in Soybean Crops (Kaggle)[37]:

Размер: Большой.

Классы: включает данные о различных классах сорняков.

Особенности: содержит изображения грядок со смешанными классами растений, включая сою и сорняки. Можно использовать для обучения моделей компьютерного зрения.

Ограничения: не фокусируется исключительно на определении сои и содержит разные классы растений.

Недостатки: Смешанные классы растений. Данный набор данных содержит изображения грядок с разными классами растений, включая сою и сорняки. Это может усложнить обучение модели, поскольку необходимо разделять и классифицировать растения на грядке.

Хотелось бы также указать недостаток собственного набора данных, а именно требуется время и ресурсы. Сбор и разметка собственного набора данных требуют времени, труда и ресурсов. Необходимо собрать данные и произвести разметку для обучения модели.

Размер собственного набора данных может быть ограничен, особенно для отдельных садоводов или фермеров. Это может сказаться на точности и обобщающей способности модели.

Учитывая эти недостатки, важно внимательно выбирать набор данных в зависимости от конкретных требований и целей.

Сравнение этих наборов данных позволяет определить их преимущества и ограничения. Выбор наиболее подходящего набора данных зависит от конкретных требований, доступности данных и целей. Собственный набор данных может быть наиболее подходящим, если требуется адаптация к местным условиям и учет конкретных классов растений.

### **3.2 Создание собственного набора данных**

Создание собственного набора данных является необходимостью в контексте точного земледелия, так как существующие общедоступные наборы данных либо не соответствуют требованиям для решения конкретной задачи, либо не обладают достаточным качеством для применения в области точного земледелия. Более того, существующие наборы данных могут быть ориентированы на узкие направления или решение определенных задач, не соответствующих требованиям сельскохозяйственного сектора.

В сфере агропромышленности возникает потребность в разработке автоматизированных систем, способных обнаруживать и классифицировать

сорняки различных видов, а также растения сои, с помощью беспилотных летательных аппаратов. Это позволит эффективно контролировать рост сорняков и растений сои, улучшая процессы возделывания и повышая урожайность.

В связи с этим, активно проводится работа по созданию собственного набора данных, специально адаптированного для обучения систем компьютерного зрения, которые могут решать поставленные задачи обнаружения и классификации сорняков и растений. Этот набор данных будет состоять из разнообразных изображений грядок, собранных в различных условиях, на разных стадиях роста растений сои и с разными видами сорняков. Важно также провести качественную разметку данных, то есть пометить соответствующие области на изображениях.

Создание собственного набора данных является важным этапом исследования в области точного земледелия, так как он позволяет разработать и обучить системы компьютерного зрения, способные эффективно обнаруживать и классифицировать сорняки и растения сои. Это, в свою очередь, открывает перспективы для автоматизации и оптимизации процессов возделывания и контроля культурных растений, снижает необходимость в ручной работе и повышает эффективность сельскохозяйственного производства.

### **3.3 Описание собственного набора данных**

Набор данных состоит из 800 фотографий грядок, на которых изображены соя и сорняки. Каждое изображение было сделано с помощью дрона, находившегося на высоте примерно 6 метров над грядками. Данная перспектива обеспечивает обзор всей грядки и позволяет визуализировать расположение растений и сорняков на ней.

Для разметки набора данных использовался инструмент CVAT (Computer Vision Annotation Tool), который значительно упрощает процесс выделения и маркировки областей интереса на изображениях. Каждая фотография была размечена с выделением соответствующих областей сои и сорняков. Для каждой области была присвоена метка класса, указывающая на то, является ли она соей или определенным видом сорняка. Всего в нашем наборе данных присутствуют 12 классов, чтобы точно идентифицировать различные виды сорняков, которые могут встречаться на грядках.

Примеры изображений из нашего набора данных приведены на (рисунке 6). На них можно увидеть несколько грядок сои и сорняков, отражающих разнообразие растений и видов сорняков, которые встречаются на грядках. Размеченные области, обведенные на изображениях, позволяют моделям нейронной сети понять и выделить соответствующие объекты при обучении и применении алгоритмов обнаружения и классификации.

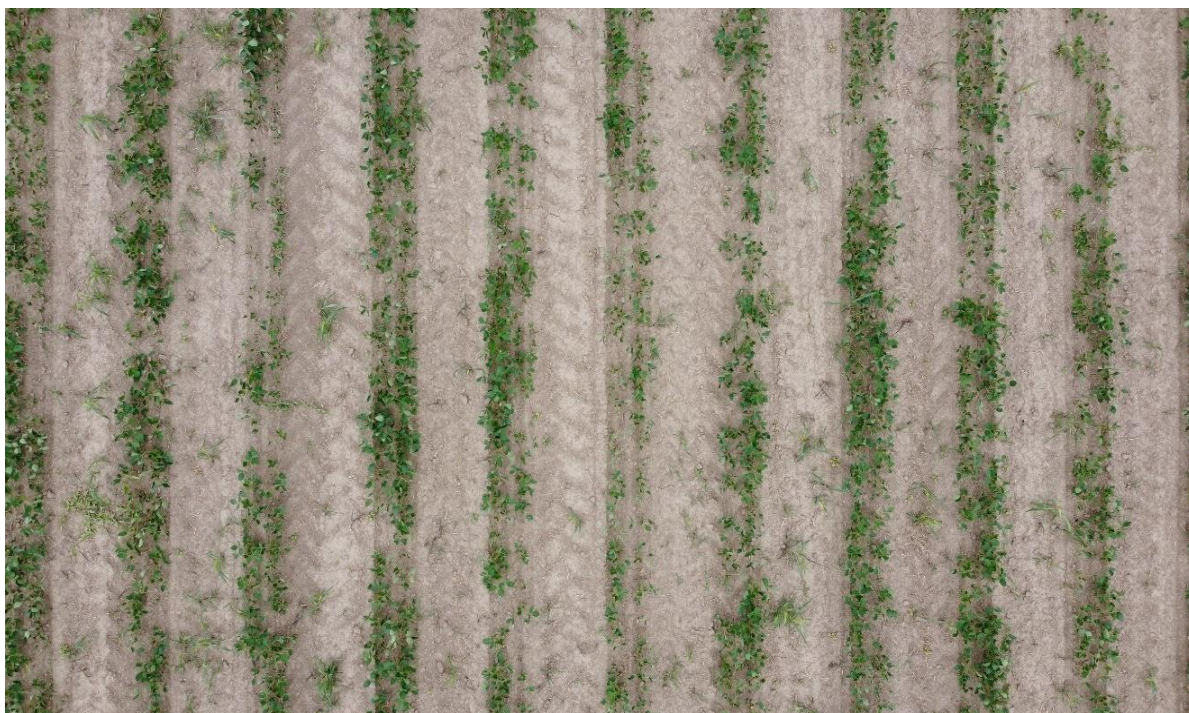


Рисунок 6. Пример изображения собственного набора данных.

Такой собственный набор данных позволяет обучить нейронную сеть, такую как YOLO (You Only Look Once), на конкретных условиях и классах сои и сорняков, которые важны для данной местности и целей. Это увеличивает возможность модели точно определить и классифицировать сою и сорняки на грядках, что поможет в дальнейшем принимать эффективные решения и применять соответствующие меры контроля и управления этими растениями.

### 3.3.1 CVAT

Computer Vision Annotation Tool (CVAT) — это бесплатный веб-инструмент для аннотирования изображений и видео с открытым исходным кодом, который используется для маркировки данных для алгоритмов компьютерного зрения. CVAT позволяет пользователям аннотировать объекты для каждого из этих случаев. Как выглядит интерфейс инструмента CVAT показан на рисунке 7.



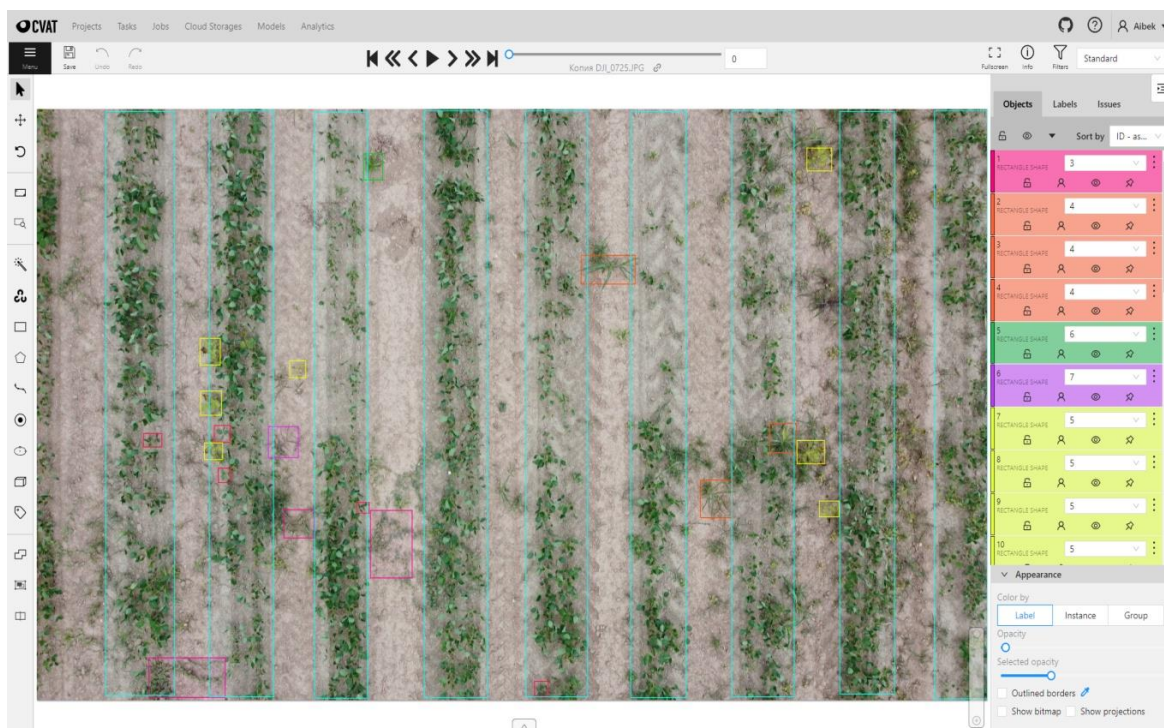


Рисунок 7. Интерфейс инструмента CVAT

Процесс разметки данных с помощью CVAT обычно включает следующие шаги:

1. Создание проекта: Первым шагом является создание нового проекта в CVAT. В этом шаге указывается основные параметры проекта, такие как название, описание и тип задачи (например, детекция объектов, сегментация или классификация).

2. Загрузка данных: Далее загружается набор данных в CVAT. Это могут быть изображения или видео. В моем случае это изображения. Можно загрузить данные с локального компьютера или импортировать их из удаленного источника.

3. Разметка объектов: После загрузки данных переходим к разметке объектов. CVAT предоставляет удобный пользовательский интерфейс для разметки объектов на изображениях или видео. Можно использовать различные инструменты для создания аннотаций, такие как прямоугольники, маски или метки классов, в зависимости от типа задачи.

4. Редактирование аннотаций: После создания аннотаций можно редактировать их при необходимости. Также можно изменять форму или размер аннотаций, добавлять или удалять объекты, а также добавлять дополнительные атрибуты к объектам.

5. Проверка и корректировка: После разметки данных важно проверить и корректировать аннотации, чтобы убедиться в их точности и соответствии требованиям. CVAT предоставляет инструменты для просмотра и анализа аннотаций, а также для исправления возможных ошибок.

6. Экспорт разметки: После завершения разметки вы можете экспортировать аннотации в различных форматах. Это позволяет использовать данные в других приложениях или алгоритмах машинного обучения.

На текущий момент размечено около 800 фотографий грядок сои подготовленных для обучения нейросети YOLO.

После разметки фотографии скачивается архив с файлами формата txt для YOLO как показано на рисунке 8.

```

7 0.259400 0.563760 0.031510 0.052391
0 0.873560 0.496918 0.057535 0.985560
0 0.763900 0.498724 0.065650 0.991573
0 0.653909 0.499927 0.058212 0.991578
0 0.546959 0.498724 0.064983 0.991573
0 0.440686 0.499327 0.066338 0.992778
0 0.320197 0.501131 0.055505 0.991578
0 0.215280 0.498724 0.067690 0.991573
5 0.183390 0.497513 0.022035 0.041107
0 0.108669 0.498122 0.072427 0.987969
5 0.185974 0.579002 0.019043 0.028049
5 0.274519 0.440207 0.016592 0.029018
5 0.823156 0.085473 0.026068 0.038689
5 0.814090 0.580364 0.028790 0.039493
5 0.833584 0.676682 0.020628 0.026600
3 0.158084 0.961973 0.080303 0.066853
3 0.372944 0.736340 0.043873 0.114209
1 0.530771 0.980638 0.014418 0.024662
1 0.343206 0.674753 0.013058 0.018378
1 0.197399 0.619864 0.013058 0.024662

```

Рисунок 8. Данные находящиеся в txt файлах

Первая цифра — это номер класса, который был взят с таблицы (грядка, сорняк или соя) на рисунке 9. Остальные четыре цифры — это координаты прямоугольника, внутри которого объект, который был выделен.



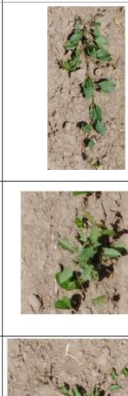









<p>Index - Важно: при создании так разметки в cvat.org добавлять labels в том же порядке, в каком тут индексы, то есть сначала добавляем label грядка, затем label соя, затем label сорняк, затем выенок и так далее.</p>	0	1	2	3	4	5	6	7
name	Грядка сои (1ый полет)	Соя (во втором полете у нее по 3 листика, видимо)	Ширец обыкновенная (Amaranthus retroflexus)	Выенок полевой (Convolvulus arvensis)	Щетинник скзый (Setaria glauca)	Дуришник обыкновенный (Xanthium strumarium)	Осот розовый (Cirsium arvense)	Просо куриное (Setaria viridis)
								
								
								

Рисунок 9. Таблица, содержащая информацию о том, как выглядит каждый объект из каждого класса

### 3.4 Нейронная сеть YOLO

YOLO (You Only Look Once) - это нейронная сеть для обнаружения объектов на изображениях. Основная идея YOLO заключается в том, чтобы разбить изображение на сетку и каждой ячейке сетки назначить класс объекта, который находится в этой ячейке, а также координаты прямоугольника, описывающего этот объект.

YOLO использует сверточную нейронную сеть, которая преобразует изображение во входной тензор и обрабатывает его, выделяя важные признаки. Затем полученный тензор передается через полносвязный слой, который преобразует его в вектор оценок классов и координаты прямоугольника.

Одна из особенностей YOLO заключается в том, что он использует только один прямой проход через сверточную нейронную сеть, что делает его быстрым и эффективным. Кроме того, YOLO может обрабатывать изображения с большим количеством объектов, так как он может определять несколько объектов на одном изображении.

Одним из главных преимуществ YOLO является его высокая точность. YOLO показывает лучшие результаты, чем многие другие алгоритмы для обнаружения объектов, особенно на маленьких объектах. Однако, как и любой другой алгоритм, YOLO также имеет свои недостатки. Он может ошибаться в определении мелких объектов и объектов, находящихся близко друг к другу. Кроме того, он может не обнаружить объекты, которые находятся в тени или частично скрыты другими объектами.

Для обучения YOLO необходимы аннотированные данные, то есть изображения с пометками, указывающими класс объекта и координаты его прямоугольника. Для достижения хороших результатов необходимо использовать большое количество обучающих данных.

В заключении можно отметить, что YOLO - это мощный алгоритм для обнаружения объектов на изображениях, который может использоваться в различных областях, таких как видеонаблюдение, автомобильная промышленность, медицина, обработка изображений и другие области, где необходимо автоматическое обнаружение объектов. YOLO имеет несколько различных версий, каждая из которых имеет свои особенности и улучшения. Например, YOLOv4 имеет более точную архитектуру и оптимизированные алгоритмы обработки изображений, что позволяет достигать лучших результатов.

Однако, как и любой другой алгоритм, YOLO не является универсальным решением для всех задач обнаружения объектов. В зависимости от специфики задачи и требований к точности, могут быть более эффективные и оптимальные алгоритмы. Кроме того, для успешного применения YOLO необходимо правильно настроить гиперпараметры и обучить модель на достаточном количестве разнообразных данных.



В целом, YOLO - это инновационный алгоритм, который имеет широкие применения и обладает высокой точностью. Дальнейшее развитие YOLO и других алгоритмов для обнаружения объектов будет продолжаться, что позволит решать все более сложные задачи в области компьютерного зрения.

### 3.4.1 YOLOv4

Для данной диссертационной работы была выбрана одна из версии YOLO, а именно YOLOv4. В этой главе будет рассказано что представляет из себя версия YOLOv4 и почему была выбрана именно эта версия хоть на данный момент существуют более новые.

YOLOv4 - это одна из версии нейронной сети YOLO (You Only Look Once), представленная в 2020 году. Она представляет собой инновационный алгоритм для обнаружения объектов на изображениях и видео с высокой точностью.

Одной из главных особенностей YOLOv4 является более точная архитектура. Алгоритм использует Darknet-53 - глубокую сверточную нейронную сеть, которая состоит из 53 слоев, что позволяет получить более точные результаты в сравнении с предыдущими версиями YOLO.

В YOLOv4 используется множество оптимизаций, которые позволяют ускорить процесс обработки изображений. Например, алгоритм использования кластеров (clustered anchors) позволяет более эффективно работать с якорными рамками, что ускоряет процесс обработки. Кроме того, были внедрены оптимизации, связанные с параллельной обработкой данных и распределением обучения между несколькими графическими процессорами.

В YOLOv4 также были внедрены новые методы аугментации данных, что позволяет улучшить обучение модели. Алгоритм использует множество различных методов аугментации, таких как случайное искажение цвета, случайное изменение размера и поворот изображения.

YOLOv4 также имеет ряд улучшений в области обработки видео. Например, в алгоритме была внедрена технология FlowNet для предсказания движения объектов на видео, что позволяет более точно определять их положение на следующем кадре.

Кроме того, YOLOv4 обладает высокой точностью обнаружения объектов.

В целом, YOLOv4 - это инновационный алгоритм, который обладает высокой точностью и эффективностью в обработке изображений и видео. Он представляет собой существенное улучшение по сравнению с предыдущими версиями YOLO и является одним из самых быстрых и точных алгоритмов обнаружения объектов на изображениях и видео. YOLOv4 может быть использован в различных задачах, таких как автоматическое вождение автомобиля, системы видеонаблюдения, медицинские системы и т.д.

Кроме того, YOLOv4 является открытым исходным кодом, что позволяет настраивать и улучшать алгоритм в соответствии с конкретными потребностями и задачами. Например, можно изменять архитектуру, параметры обучения, оптимизации и т.д.

Однако, необходимо учитывать, что использование YOLOv4 требует высоких вычислительных мощностей и больших объемов данных для обучения. Для обработки больших объемов данных и достижения высоких результатов необходимо использовать высокопроизводительные компьютеры и графические процессоры.

В целом, YOLOv4 является передовым алгоритмом обнаружения объектов на изображениях и видео с высокой точностью и быстродействием. Он может быть использован в различных задачах, что делает его важным инструментом для исследований и разработки в области компьютерного зрения.

Данная версия была выбрана по ряду причин таких как:

**Быстродействие:** YOLO позволяет быстро обрабатывать изображения и обнаруживать объекты в режиме реального времени, благодаря своей способности обрабатывать изображения целиком за один проход.

**Точность:** YOLO демонстрирует высокую точность обнаружения объектов на изображениях, особенно для крупных объектов. Он также имеет хорошую устойчивость к шуму на изображениях и низкому разрешению.

**Простота и удобство использования:** YOLO имеет простую архитектуру и легко настраивается на новые задачи. Он также имеет открытый исходный код и обширное сообщество пользователей, что делает его доступным и удобным в использовании.

А также в ограниченных ресурсах системы так как предполагается что более новые версии будут работать нестабильно из-за нехватки мощностей системы.

### **3.5 Установка нейронной сети YOLOv4**

В главе представлены подробные сведения о процессе установки нейронной сети YOLOv4, включая перечень программных компонентов, применяемых в этом процессе, а также их функциональное назначение. Установка YOLOv4 является неотъемлемой частью данного исследования, поскольку предоставляет нам возможность провести обширный анализ архитектуры сети и определить ее пригодность для решения поставленных задач.

#### **1. Среда разработки Python:**

**1.1. Загрузка Python:** Первым шагом является загрузка и установка языка программирования Python. Рекомендуется использовать версию Python 3.8.5, так как она совместима с требуемыми зависимостями и библиотеками.

1.2. Установка пакетного менеджера `pip`: после установки Python следует установить пакетный менеджер `pip`, который позволяет управлять сторонними библиотеками и зависимостями Python.

2. Дополнительные зависимости и библиотеки:

2.1. Установка CUDA: CUDA — это программная платформа, предоставляемая компанией NVIDIA, которая позволяет проводить параллельные вычисления на графических процессорах.[29]

2.2. Установка `cuDNN`: `cuDNN` (CUDA Deep Neural Network library) — это библиотека, оптимизированная для глубокого обучения на графических процессорах.[30]

2.3. Установка библиотеки OpenCV: OpenCV (Open Source Computer Vision Library) — это библиотека компьютерного зрения, широко используемая для обработки и анализа изображений и видео.[28]

3. Microsoft Visual Studio:

3.1. Загрузка Microsoft Visual Studio: Первым шагом является загрузка и установка интегрированной среды разработки Microsoft Visual Studio. Это мощное программное обеспечение предоставляет средства разработки, отладки и компиляции приложений для платформы Windows.

4. CMake:

4.1. Загрузка CMake: CMake — это кроссплатформенная система управления сборкой, которая используется для автоматизации процесса сборки программного обеспечения. Необходимо загрузить и установить последнюю версию CMake с официального веб-сайта.[31]

4.2. Установка CMake: после загрузки CMake следует запустить установщик и следовать инструкциям по установке. Рекомендуется добавить CMake в переменную среды `PATH`, чтобы иметь доступ к нему из командной строки.

5. Установка Darknet:

5.1. Загрузка и компиляция Darknet: Darknet — это фреймворк с открытым исходным кодом, используемый для обучения и развертывания нейронных сетей, включая YOLOv4.

5.2. Конфигурация Darknet: после компиляции следует настроить файлы Darknet в соответствии с требованиями.

6. Сборка проекта Darknet:

6.1. Выбор конфигурации сборки: В Visual Studio выберите необходимую конфигурацию сборки, такую как "Debug" или "Release". Это определит параметры сборки проекта Darknet.

6.2. Сборка проекта: выполните сборку проекта Darknet в Visual Studio. Это приведет к компиляции и созданию исполняемых файлов, необходимых для работы YOLOv4.

7. Конфигурационные файлы и веса предварительно обученной модели:

7.1. Загрузка конфигурационных файлов: необходимо загрузить конфигурационные файлы для YOLOv4, включая файл "yolov4.cfg".

Конфигурационные файлы содержат параметры сети, такие как количество классов, размер входного изображения и настройки обучения.

7.2. Загрузка весов предварительно обученной модели: также следует загрузить веса предварительно обученной модели YOLOv4.

В следующей главе хотелось бы расписать эти компоненты более подробно для полного понимания всего процесса.

## **3.6 Подробное описание компонентов, использованных при установке YOLOv4**

### **3.6.1 OpenCV**

OpenCV (Open Source Computer Vision Library) - это библиотека компьютерного зрения с открытым исходным кодом, которая предоставляет широкий набор функций и алгоритмов для обработки изображений и видео. Она разработана для облегчения разработки приложений компьютерного зрения и машинного обучения.

OpenCV предоставляет богатый набор функций, включая обработку изображений, анализ видео, машинное обучение, распознавание образов, отслеживание объектов, калибровку камеры, работу с глубинными картами, сопряжение видео и многое другое. Благодаря этим функциям разработчики могут создавать сложные системы компьютерного зрения с помощью набора инструментов, предоставляемых OpenCV.

Некоторые основные возможности OpenCV:

1. Обработка изображений: OpenCV предоставляет функции для чтения, записи и обработки изображений. Она включает операции изменения размера, поворота, обрезания, фильтрации, коррекции цвета и многое другое.

2. Обработка видео: OpenCV позволяет работать с видеофайлами и потоками видео. Она предоставляет возможности захвата видео с камеры, анализа видео, отслеживания движения, детектирования объектов и других задач, связанных с обработкой видео.

3. Машинное обучение: OpenCV содержит алгоритмы машинного обучения, такие как классификация, кластеризация, регрессия и т.д. Она предоставляет инструменты для создания и обучения моделей машинного обучения на основе изображений и видео.

4. Детектирование и распознавание объектов: OpenCV имеет функции для детектирования и распознавания объектов на изображениях и видео. Она включает в себя алгоритмы распознавания лиц, детектирования объектов, трекинга объектов и другие методы компьютерного зрения.

5. Калибровка камеры: OpenCV предоставляет возможности калибровки камеры, которые позволяют определить параметры камеры, такие как фокусное

расстояние, коэффициенты дисторсии и т.д. Это полезно для задач, связанных с трехмерным видением и реконструкцией сцены.

OpenCV имеет интерфейсы для различных языков программирования, включая C++, Python, Java и другие. Благодаря этому она широко используется в академических и промышленных проектах, связанных с компьютерным зрением и обработкой изображений.

OpenCV имеет обширную документацию, где вы можете найти примеры кода, руководства и дополнительные материалы по использованию библиотеки.

### 3.6.2 NvidiaCUDA

NVIDIA CUDA (Compute Unified Device Architecture) - это архитектура параллельных вычислений, разработанная компанией NVIDIA для использования на ее графических процессорах (GPU). CUDA предоставляет программное и аппаратное окружение, которое позволяет разработчикам использовать возможности параллельных вычислений на GPU для решения широкого спектра задач.

Основные компоненты NVIDIA CUDA:

1. GPU(графический процессор): GPU является высокопроизводительным микропроцессором, который специализируется на обработке графики и выполнении параллельных вычислений. Он состоит из множества ядер (CUDA ядра), которые могут выполнять множество вычислений одновременно.

2. CUDA ядро: CUDA ядро - это программа, которая выполняется на графическом процессоре. Она написана на языке программирования CUDA C/C++ или других поддерживаемых языках программирования и содержит инструкции для выполнения параллельных вычислений на GPU.

3. CUDA Toolkit: CUDA Toolkit - это пакет разработчика, предоставляемый NVIDIA, который включает в себя набор инструментов, библиотек и драйверов для разработки и выполнения программ, использующих NVIDIA CUDA. Он включает в себя компилятор CUDA, библиотеки для работы с памятью GPU, управления потоками, матричными вычислениями и другими задачами.

4. CUDA API: CUDA API представляет собой набор функций и интерфейсов программирования, которые позволяют разработчикам взаимодействовать с аппаратными ресурсами GPU и выполнять параллельные вычисления. CUDA API доступен на различных языках программирования, включая C/C++, Python, Fortran и другие.

5. CUDA язык программирования: CUDA позволяет разработчикам написать параллельный код, используя расширения языка программирования C/C++. Эти расширения включают в себя новые типы данных, директивы компилятора и функции, которые позволяют эффективно использовать возможности параллельных вычислений на GPU.

### Преимущества NVIDIA CUDA:

**Параллельные вычисления:** CUDA позволяет разработчикам распараллеливать задачи и эффективно использовать мощность графического процессора для выполнения больших объемов вычислений одновременно. Это особенно полезно для задач, которые могут быть разбиты на независимые части, такие как обработка изображений, научные вычисления, машинное обучение и другие.

**Ускорение вычислений:** Использование графического процессора для выполнения параллельных вычислений может значительно ускорить выполнение задач по сравнению с центральным процессором (CPU). GPU предлагает значительно большую параллельную вычислительную мощность и высокую пропускную способность памяти.

**Гибкость и универсальность:** CUDA поддерживает различные языки программирования и позволяет разработчикам использовать GPU для выполнения широкого спектра задач, включая графику, научные вычисления, машинное обучение, обработку сигналов и другие.

**Экосистема инструментов:** CUDA Toolkit предоставляет разработчикам инструменты и библиотеки для упрощения разработки параллельного кода и оптимизации его производительности. Это включает в себя библиотеки для линейной алгебры, обработки изображений, машинного обучения и других распространенных задач.

NVIDIA CUDA является мощным инструментом для разработки параллельных вычислений на графических процессорах и находит широкое применение в области научных исследований, машинного обучения, графики, высокопроизводительных вычислений и других областях, где требуется высокая производительность параллельных вычислений.

### 3.6.3 CUDNN

cuDNN (CUDA Deep Neural Network) - это библиотека глубокого обучения, разработанная компанией NVIDIA для оптимизации выполнения нейронных сетей на графических процессорах (GPU) с поддержкой NVIDIA CUDA. Она предоставляет набор функций и алгоритмов, которые позволяют эффективно выполнять операции глубокого обучения, такие как свертки, пулинг, нормализацию и другие, на GPU.

**Основные компоненты cuDNN:**

1. **GPU-оптимизированные функции:** cuDNN предоставляет оптимизированные реализации основных операций глубокого обучения, таких как свертка, пулинг, активация, нормализация и др. Эти функции специально разработаны для эффективного использования аппаратных возможностей графического процессора и максимального ускорения выполнения операций нейронных сетей.

2. Автоматическое масштабирование: cuDNN автоматически оптимизирует выполнение операций глубокого обучения на различных графических процессорах NVIDIA, позволяя использовать все доступные вычислительные ресурсы и достичь максимальной производительности. Он обеспечивает автоматическое масштабирование вычислительной нагрузки на все ядра GPU и распределение памяти для оптимальной работы.

3. Совместимость и интеграция: cuDNN совместим с другими библиотеками глубокого обучения, такими как TensorFlow, PyTorch, Caffe и другие. Он предоставляет специальные API и интеграционные возможности, которые позволяют разработчикам легко использовать cuDNN в своих проектах и интегрировать его с другими инструментами машинного обучения и глубокого обучения.

4. Оптимизация производительности: cuDNN предлагает различные оптимизации производительности, включая многопоточную обработку, использование вычислительных ядер GPU, оптимизированное использование памяти и другие техники. Это позволяет достичь высокой скорости выполнения операций глубокого обучения и ускорить обучение и вывод нейронных сетей.

5. Обновления и поддержка: cuDNN активно развивается и обновляется NVIDIA, чтобы предоставлять последние инновации и оптимизации в области глубокого обучения. Он также предоставляет поддержку разработчикам через документацию, примеры кода и сообщество разработчиков.

cuDNN является важным компонентом в экосистеме инструментов NVIDIA для глубокого обучения и позволяет разработчикам эффективно использовать графические процессоры для обработки нейронных сетей. Он обеспечивает ускорение выполнения операций глубокого обучения и повышает производительность моделей нейронных сетей на GPU.

### 3.6.4 CMake

CMake (Cross-Platform Make) - это кросс-платформенная система автоматизации сборки и конфигурации проектов. Она предоставляет простой и универсальный способ управления процессом сборки программного обеспечения на разных платформах и в различных средах разработки.

Вот некоторые ключевые особенности и преимущества CMake:

1. Кросс-платформенность: CMake разработан для работы на различных операционных системах, включая Windows, macOS и Linux. Он обеспечивает единообразную среду разработки, которая позволяет собирать исходный код проекта на разных платформах без изменений в конфигурационных файлах.

2. Декларативный язык описания проекта: CMake использует простой и понятный язык описания проекта, основанный на синтаксисе CMake. С помощью этого языка разработчики могут описать зависимости проекта, настройки компилятора, опции компиляции и другие параметры сборки.

3. Генерация файлов сборки: CMake автоматически генерирует файлы сборки для выбранной целевой платформы и среды разработки. Это может быть Makefile для Unix-подобных систем, проектные файлы для сред разработки, таких как Visual Studio или Xcode, или скрипты сборки для других сред разработки.

4. Управление зависимостями: CMake облегчает управление зависимостями проекта. Он позволяет легко настроить и включить внешние библиотеки, модули или другие компоненты, необходимые для успешной сборки проекта. CMake автоматически находит и подключает зависимости, что упрощает интеграцию сторонних компонентов.

5. Многоплатформенная поддержка компиляторов: CMake поддерживает различные компиляторы, включая GCC, Clang, Visual C++ и другие. Это позволяет разработчикам свободно выбирать компилятор, который наиболее подходит для их проекта.

6. Расширяемость: CMake предоставляет механизмы для расширения его функциональности с помощью плагинов, модулей и пользовательских скриптов. Это позволяет настроить процесс сборки проекта под конкретные потребности разработчиков.

CMake представляет собой мощный инструмент для автоматизации сборки и конфигурации проектов. Он упрощает процесс разработки программного обеспечения, особенно при работе с крупными проектами, где требуется поддержка разных платформ и сред разработки.

### **3.6.5 Darknet**

Darknet - это открытая платформа глубокого обучения, разработанная для реализации нейронных сетей, включая YOLO (You Only Look Once). Она была создана Джозефом Редмондом и используется для обучения и выполнения различных алгоритмов компьютерного зрения.

Вот некоторые ключевые особенности Darknet:

1. Архитектура: Darknet написан на языке программирования C и C++, что делает его эффективным и быстрым в выполнении вычислений. Он предоставляет гибкую и модульную структуру для создания и настройки различных архитектур нейронных сетей.

2. Поддержка YOLO: Darknet имеет реализацию алгоритма обнаружения объектов YOLO. YOLO — это эффективный алгоритм обнаружения объектов в реальном времени, который анализирует изображение в целом и выполняет обнаружение объектов одновременно в одном прогоне сети. Darknet поддерживает различные версии YOLO.

3. Поддержка различных задач компьютерного зрения: Darknet поддерживает не только обнаружение объектов, но и другие задачи



компьютерного зрения, такие как классификация изображений и сегментация семантики. Он предоставляет возможности для обучения и выполнения различных типов нейронных сетей.

4. Поддержка разных типов архитектур: Darknet позволяет создавать различные типы архитектур нейронных сетей, включая полносвязные сети, сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN). Он также поддерживает сверточные слои, пулинг, активации, объединение и другие операции, необходимые для построения глубоких нейронных сетей.

5. Обучение и выполнение на GPU: Darknet поддерживает использование графических процессоров (GPU) для ускорения обучения и выполнения нейронных сетей. Это позволяет достичь значительного ускорения в вычислениях и повышении производительности, особенно при работе с большими объемами данных и сложными моделями.

### **3.6.6 Как работают компоненты в контексте с нейронной сети YOLOv4**

Хотелось бы описать, как эти компоненты - OpenCV, Nvidia CUDA и cuDNN - работают в контексте нейронной сети YOLOv4:

OpenCV: OpenCV предоставляет мощные инструменты для обработки изображений и видео, что делает его полезным компонентом для работы с нейронной сетью YOLOv4. С помощью OpenCV вы можете загружать изображения или видеофайлы, изменять их размер, выполнять преобразования цветового пространства, а также обнаруживать и извлекать объекты из изображений. В контексте YOLOv4, OpenCV может использоваться для предобработки входных изображений, визуализации результатов обнаружения объектов и выполнения других операций, связанных с обработкой изображений.

Nvidia CUDA: Nvidia CUDA предоставляет архитектуру параллельных вычислений для использования на графических процессорах (GPU). YOLOv4 является вычислительно интенсивной нейронной сетью, требующей большой вычислительной мощности. Использование Nvidia CUDA позволяет распараллелить и ускорить выполнение операций нейронной сети на GPU. Это включает выполнение сверток, активаций, пулинга и других операций, которые составляют основу алгоритма YOLOv4. CUDA позволяет использовать мощности вычисления GPU для ускорения процесса обнаружения объектов и повышения производительности сети.

cuDNN: cuDNN - это библиотека глубокого обучения, оптимизированная для выполнения нейронных сетей на графических процессорах с использованием Nvidia CUDA. Она предоставляет оптимизированные реализации операций глубокого обучения, таких как свертка, активации, пулинг и другие. В контексте YOLOv4, cuDNN может использоваться для ускорения выполнения операций свертки и других операций, которые являются

основой работы алгоритма обнаружения объектов. Оптимизированные функции cuDNN позволяют эффективно использовать вычислительные возможности GPU и достичь высокой скорости выполнения нейронной сети YOLOv4.

Комбинация OpenCV, Nvidia CUDA и cuDNN позволяет эффективно работать с нейронной сетью YOLOv4 на графическом процессоре. OpenCV обеспечивает возможность загрузки и обработки входных изображений, а также визуализацию результатов. Nvidia CUDA и cuDNN вместе позволяют распараллелить выполнение операций нейронной сети и оптимизировать ее производительность на GPU. Эта комбинация обеспечивает высокую скорость и точность обнаружения объектов с помощью YOLOv4.

### 3.7 Обучение нейронной сети и результаты

Обучение сети проводилось на подготовленной выборке данных с использованием GPU для ускорения процесса обучения.

Нейронная сеть была обучена на 150 фотографиях, чтобы исследовать процесс обучения и понять принципы, лежащие в основе YOLO. Были установлены соответствующие параметры процесса обучения, такие как размер изображения, количество итераций и количество классов. Созданы файлы для определения каталога фотографий и файлы с координатами аннотаций. После этого начался тренировочный процесс. Обучение длилось примерно 4-5 дней. После обучения были получены результаты показанные на рисунке 10.



Рисунок 10. Результат обучения нейронной сети YOLOv4

Результаты обучения оказались ниже чем ожидалось поэтому рекомендуется провести анализ результатов обучения и определить, какие проблемы могут привести к снижению точности.

Могу предположить возможные причины такие как:

Недостаточное количество обучающих данных: Если количество обучающих данных недостаточно, то это может привести к недостаточному охвату всех возможных случаев в процессе обучения.

Неправильно подобранные гиперпараметры: Некоторые гиперпараметры модели, такие как коэффициент обучения, веса и размеры сети, могут оказать влияние на точность модели.

Недостаточное количество эпох обучения: Если количество эпох обучения недостаточно, то модель может не иметь достаточного времени для настройки своих параметров.

Однако, стоит отметить, что в процессе обучения нейронной сети возникают определенные сложности, такие как необходимость большого количества размеченных данных, выбор оптимальных гиперпараметров и оптимизация алгоритма обучения, чтобы достичь наилучших результатов. Также, необходимо учитывать, что точность работы нейронной сети может зависеть от условий съемки фотографий, освещения и других факторов.

### **3.8 Второе обучение нейронной сети и результаты**

Было сделано второе обучение нейронной сети YOLOv4 но уже с большим количеством фотографий а именно 800. Обучение проходило 5 дней и ниже на рисунке 11 и рисунке 12 показаны примеры некоторых результатов обучения.



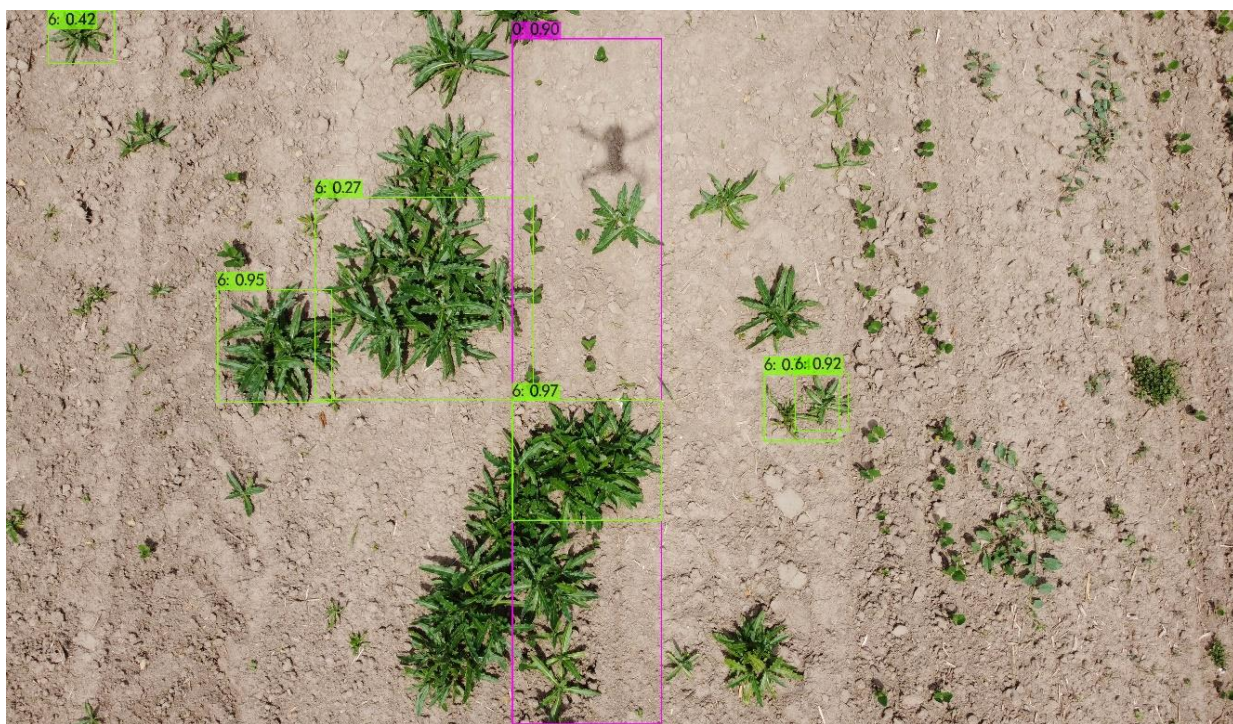


Рисунок 11. Результаты второго обучения нейронной сети YOLOv4

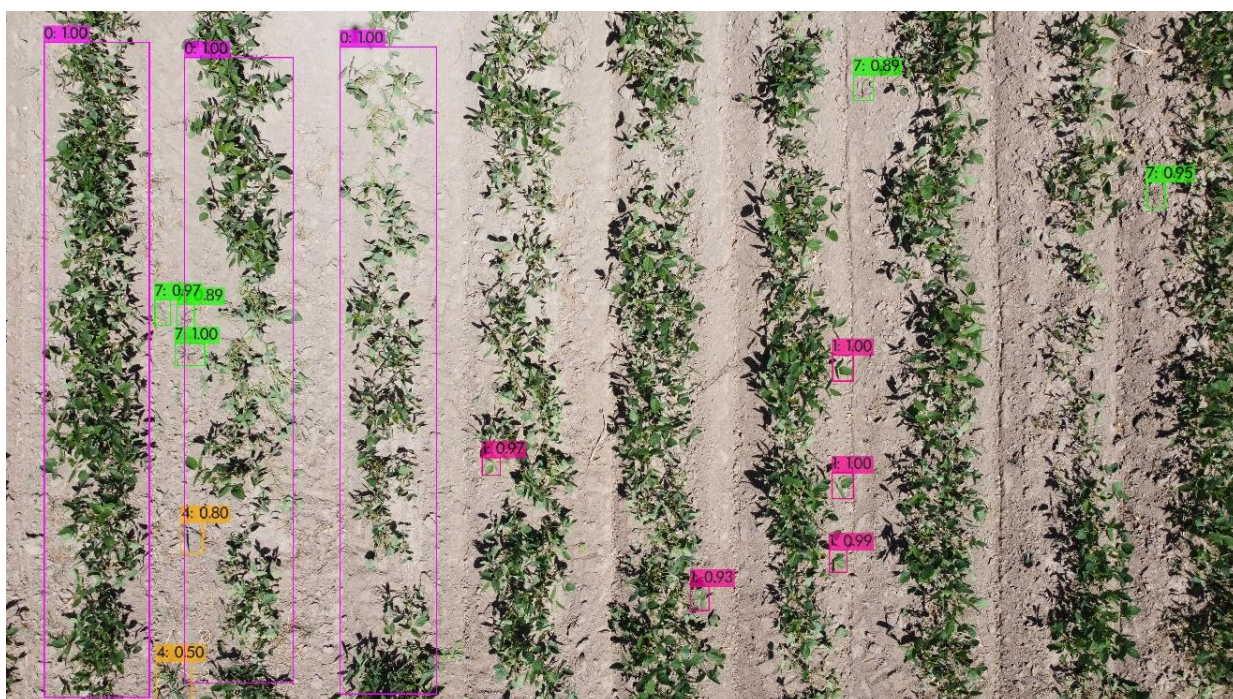


Рисунок 12. Результаты второго обучения нейронной сети YOLOv4

Также на рисунке 13 показан сам график прохождения последних количеств итерации в котором показана средняя ошибка и точность определения объектов.



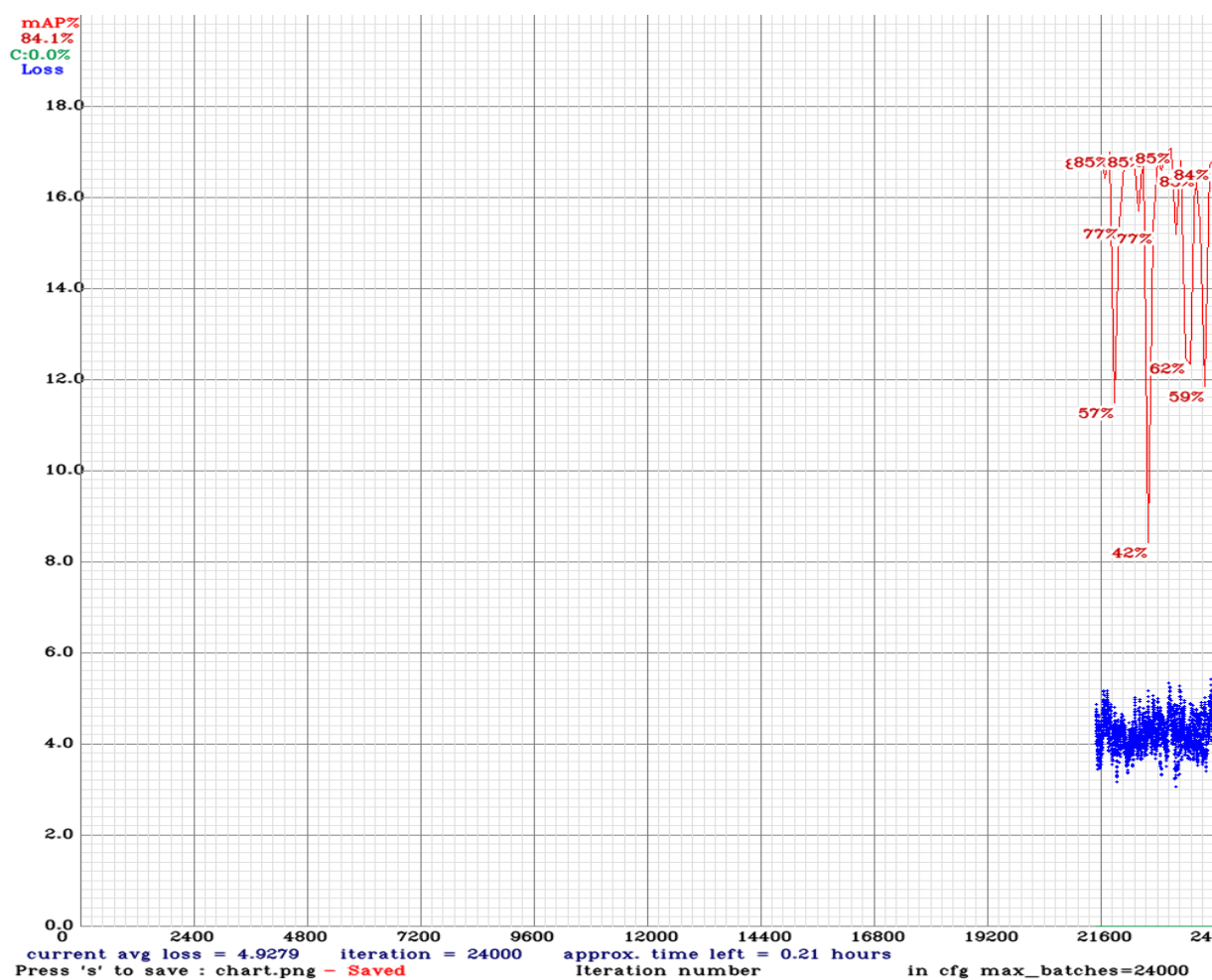


Рисунок 13. Результаты второго обучения нейронной сети YOLOv4

В целом точность модели как показано на графике идет в среднем выше 70 %. В самой модели есть одни из главных проблем в виде того, что на некоторых фотографиях один объект определяется как два разных класса и вторая проблема заключается в том, что бывает обнаружение объектов вовсе не происходит.

Модель требует доработок в плане увеличении точности определения объектов а также в добавление большего количества фотографии для обучения что может улучшить обучение нейронной сети и привести к более точному результату.

## ЗАКЛЮЧЕНИЕ

В заключение диссертационной работы, можно сделать следующие выводы. Результаты исследования показали, что формирование и разметка набора данных, полученных с борта БПЛА, являются важным и неотъемлемым этапом в применении методов машинного обучения для решения задач точного земледелия. Качество данных напрямую влияет на точность и надежность моделей машинного обучения, а, следовательно, на эффективность применения точного земледелия в сельском хозяйстве. В ходе исследования была проведена подробная аналитическая работа, направленная на изучение проблематики обнаружения сои и сорняков в сельскохозяйственных угодьях. Были рассмотрены существующие методы и подходы к решению данной проблемы, а также их преимущества и недостатки. В результате этого анализа было выявлено, что нейронная сеть YOLO представляет собой перспективный инструмент для решения задачи обнаружения объектов на изображениях.

В ходе исследования был создан и аннотирован набор данных, содержащий изображения грядок сои и сорняков. Этот набор данных является ценным ресурсом, который может быть использован для обучения и оценки алгоритмов обнаружения.

YOLO (You Only Look Once), известная своей способностью к быстрой и точной классификации объектов на изображениях. Был проведен процесс обучения с использованием разработанного набора данных, что позволило адаптировать модель к задаче распознавания сои и сорняков на грядках.

Для успешного применения нейронной сети YOLO требуется хорошо сформированный и размеченный набор данных. В рамках работы был разработан и использован специальный подход к формированию и разметке данных для обучения и тестирования модели обнаружения сои и сорняков.

На основе разработанного набора данных и проведения обучения нейронной сети YOLO было осуществлено экспериментальное исследование, направленное на оценку ее эффективности и точности в контексте обнаружения сои и сорняков. В результате экспериментов было установлено, что предложенная модель обладает значимыми показателями эффективности, достигающими значения выше 70%.

В настоящее время нейронные сети широко применяются во многих областях и решают разнообразные задачи. Их развитие и прогрессивное усовершенствование способствуют улучшению и упрощению повседневной жизни. В контексте данного исследования, разработанный подход представляет собой важный пример применения нейронной сети YOLO для оптимизации сельскохозяйственного производства и сокращения ручного труда.

Полученные результаты говорят в пользу значительного улучшения эффективности процесса обнаружения сои и сорняков благодаря предложенной модели. Это может иметь важное значение для сельскохозяйственной отрасли, поскольку позволит сократить затраты на трудоемкие процессы и повысить производительность. Однако следует отметить, что дальнейшие исследования и

оптимизации модели необходимы для ее дальнейшего развития и повышения общей эффективности.

Созданный набор данных сои и сорняков может быть использован в дальнейших исследованиях и разработках в области компьютерного зрения и машинного обучения в аграрном секторе. Он представляет собой ценный ресурс для обучения и тестирования других моделей и алгоритмов, а также может быть расширен и дополнен для анализа других агрокультурных культур и вредителей.

Однако, необходимо отметить, что дальнейшие исследования и улучшения все еще требуются. Возможности нейронных сетей YOLO могут быть расширены путем внедрения более сложных архитектур и оптимизации процесса обучения. Также проведение сравнительного анализа с другими алгоритмами распознавания для оценки эффективности разработанной системы. Также стоит рассмотреть возможность расширения набора данных, чтобы улучшить общую обобщающую способность модели и ее применимость к различным условиям и сценариям.

В целом, результаты работы является важным вкладом в область распознавания сорняков и сои на грядках с использованием нейронных сетей, и ее результаты могут быть полезны как для научного сообщества, так и для сельскохозяйственных предприятий, стремящихся к автоматизации и оптимизации процессов выращивания культурных растений. Также подтверждают потенциал нейронной сети YOLO в области обнаружения сои и сорняков, а разработанный набор данных и методология разметки могут быть использованы в будущих исследованиях и практических приложениях. Это позволит автоматически определять и классифицировать сорняки и сою, а затем принимать соответствующие меры по управлению, такие как автоматическое удаление сорняков или оптимизация использования удобрений и пестицидов.

В целом, дальнейшие исследования в этой области имеют значительный потенциал для улучшения эффективности и устойчивости сельскохозяйственного производства. Применение нейронных сетей, таких как YOLO, для обнаружения сорняков и сои может привести к сокращению затрат и повышению урожайности, что является важным шагом в направлении устойчивого развития сельского хозяйства.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Vittorio Mazzia, Lorenzo Comba, Aleem Khaliq, Marcello Chiaberge, Paolo Gay UAV and Machine Learning Based Refinement of a Satellite-Driven Vegetation Index for Precision Agriculture [Электронный ресурс]: URL: <https://www.mdpi.com/1424-8220/20/9/2530/html>
- 2 Yuncheng Lu, Zhucun Xue, Gui-Song Xia, Liangpei Zhang A survey on vision-based UAV navigation [Электронный ресурс]: URL: <https://www.tandfonline.com/doi/full/10.1080/10095020.2017.1420509?scroll=top&needAccess=true>
- 3 The DeepWeeds dataset [Электронный ресурс]: URL: <https://github.com/AlexOlsen/DeepWeeds>
- 4 Ladybird Cobbity Brassica dataset [Электронный ресурс]: URL: [https://cloudstor.aarnet.edu.au/plus/s/N0giZD5QTMB32Xj?path=%2FJFR\\_2018](https://cloudstor.aarnet.edu.au/plus/s/N0giZD5QTMB32Xj?path=%2FJFR_2018)
- 5 Sugar Beets 2016 (Chebrolu et al., 2017) [Электронный ресурс]: URL: [https://www.ipb.uni-bonn.de/datasets\\_IJRR2017/](https://www.ipb.uni-bonn.de/datasets_IJRR2017/)
- 6 Carrot-Weed dataset [Электронный ресурс]: URL: <https://github.com/lameski/rgbweedddetection>
- 7 Plant seedlings dataset [Электронный ресурс]: URL: <https://vision.eng.au.dk/plant-seedlings-dataset/>
- 8 CWFID dataset [Электронный ресурс]: URL: <https://github.com/cwfid/dataset>
- 9 Mukhamediev R. I. et al. From Classical Machine Learning to Deep Neural Networks: A Simplified Scientometric Review // Applied Sciences. – 2021. – Т. 11. – №. 12. С. 5541. [Электронный ресурс]: URL: <https://doi.org/10.3390/app11125541>
- 10 Mukhamediev R. I. et al. Review of Some Applications of Unmanned Aerial Vehicles Technology in the Resource-Rich Country // Applied Sciences. – 2021. – Т. 11. – №. 21. – С. 10171. [Электронный ресурс]: URL: <https://doi.org/10.3390/app112110171>
- 11 Введение в машинное обучение. Мухамедиев Р.И., Амиргалиев Е.Н. – 2022. – 288 стр.
- 12 Fonder, M.; Droogenbroeck, M. Mid-Air: A Multi-Modal Dataset for Extremely Low Altitude Drone Flights. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 553–562. [Электронный ресурс]: URL: [\[CrossRef\]](#)
- 13 Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
- 14 Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.



- 15Mittal, P.; Sharma, A.; Singh, R. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image Vis. Comput.* 2020, 104, 104046. [Электронный ресурс]: URL: [[CrossRef](#)]
- 16Zhong, Y.; Hu, X.; Luo, C.; Wang, X.; Zhao, J.; Zhang, L. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. *Remote Sens. Environ.* 2020, 250, 112012. [Электронный ресурс]: URL: [[CrossRef](#)]
- 17 Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement.[Электронный ресурс]: URL:<https://arxiv.org/abs/1804.02767>
- 18 Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.[Электронный ресурс]: URL:<https://arxiv.org/abs/1512.02325>
- 19 Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). ImageNet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.[Электронный ресурс]: URL:<https://arxiv.org/abs/1409.0575>
- 20Li, L., Zhang, L., Huang, L., &He, L. (2019). Weed detection using deep learning: A review. *Computers and Electronics in Agriculture*, 162, 206-220.
- 21 Marques, G., Sanchez, A., Sisbot, E. A., & Alenyà, G. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90.
- 22 Abdullah, M. Z., & Musa, N. H. (2020). Automated weed detection and classification using deep learning techniques: A comprehensive review. *Computers and Electronics in Agriculture*, 169, 105167.
- 23 González-de-Soto, M., González-Moreno, Á., López-Granados, F., & Peña-Barragán, J. M. (2019). Deep learning for weed control in agriculture: A review. *Computers and Electronics in Agriculture*, 162, 219-231.
- 24 Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.[Электронный ресурс]: URL:<https://arxiv.org/abs/1409.1556>
- 25 R.S. Michalski and R.L. Chilausky"Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis",*International Journal of Policy Analysis and Information Systems*, Vol. 4,No. 2, 1980.
- 26 Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) challenge. *International journal of computer vision*, 88(2), 303-338.[Электронный ресурс]: URL:[https://www.researchgate.net/publication/220659463\\_The\\_Pascal\\_Visual\\_Object\\_Classes\\_VOC\\_challenge](https://www.researchgate.net/publication/220659463_The_Pascal_Visual_Object_Classes_VOC_challenge)
- 27Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer*

- vision and pattern recognition (pp. 3431-3440).[Электронный ресурс]: URL:<https://arxiv.org/abs/1411.4038>
- 28OpenCV: Computer Vision Projects with Python[Электронный ресурс]: URL: ( <https://opencv.org/> )
- 29 CUDA Toolkit Documentation[Электронный ресурс]: URL: (<https://docs.nvidia.com/cuda/> )
- 30NVIDIAcuDNNDocumentation[Электронный ресурс]: URL:<https://docs.nvidia.com/deeplearning/cudnn/>
- 31CMake Documentation[Электронный ресурс]: URL: (<https://cmake.org/documentation/>)
- 32Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
- 33Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).
- 34Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.[Электронный ресурс]: URL:<https://arxiv.org/abs/2004.10934>
- 35Richard Szeliski (2010): Computer Vision: Algorithms and Applications"(pp. 812)
- 36 Weed Species Identification :[Электронный ресурс]: URL:<https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes>
- 37 Weed Detection in Soybean Crops:[Электронный ресурс]: URL:<https://www.kaggle.com/datasets/fpeccia/weed-detection-in-soybean-crops>
- 38 Как работает сверточная нейронная сеть: архитектура, примеры, особенности - [Электронный ресурс]: URL:<https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/>
- 39 Canny J. A computational approach to edge detection //IEEE Transactions on pattern analysis and machine intelligence. – 1986. – №. 6. – С. 679-698.
- 40 Taiwo Oladipupo Ayodele. Types of Machine Learning Algorithms // New Advances in Machine Learning. – 2010. – P. 19-48.
- 41 Muhamedyev R. Machine learning methods: An overview //CMNT. - 19(6). - 2015. - p. 14-29.
- 42 Hastie T., Tibshirani R., Friedman J. Unsupervised learning // Springer New York, 2009. – P. 485-585.
- 43 Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques.Emerging artificial Intelligence Applications in Computer Engineering. - IOS Press, 2007. - P. 3-24.
- 44 Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
- 45 Darknetoverview - [Электронныйресурс]: URL: <https://github.com/AlexeyAB/darknet#requirements-for-windows-linux-and-macos>

46Ian Goodfellow, Yoshua Bengio, Aaron Courville. "Deep Learning" - The MIT Press; Illustrated edition (2016): 800.

47Rajalingappaa Shanmugamani. "Deep Learning for Computer Vision" - Packt Publishing (2018): 310.

48Li Deng, Dong Yu. "Deep Learning: Methods and Applications" - Foundations and Trends in Signal Processing Vol. 7, Nos. 3–4 (2013) 197–387

49Andriy Burkov. "The Hundred-Page Machine Learning Book"(2019)pp.160

50S. Albawi, T. A. Mohammed, S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), 1–6, IEEE, 2017, doi:10.1109/ICEngTechnol.2017.8308186.

51 Jähne B., Haussecker H., Geissler P. (ed.). Handbook of computer vision and applications. – San Diego : Academic press, 1999. – T. 2. – C. 423-450.