



Лабораторная работа №1

по дисциплине: Технологии нейросетевых вычислений

вариант: Plant Segmentation

Выполнили: Миху Вадим

Чернова Анна

P34301

Преподаватель: Старобыховская Анастасия Александровна

Санкт-Петербург
2024

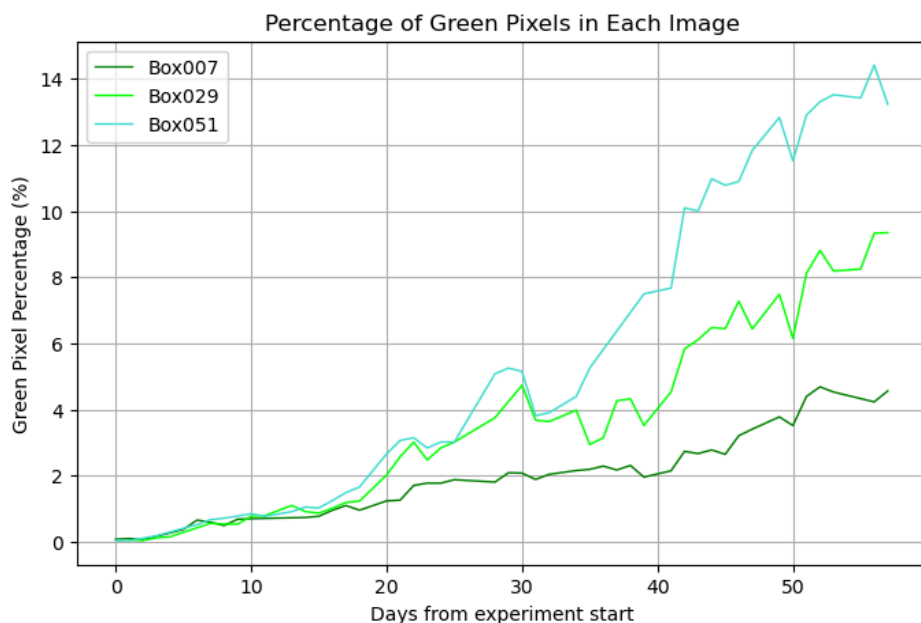
Задание

1. Выбрать задачу из списка (сообщить об этом)
2. Скачать dataset (см. задачу)
3. Сделать анализ данных (не менее 8 графиков/расчетов)
4. Сделать выводы по результатам исследования и описать потенциальные риски и их решения (не менее 3х)
5. Подготовить данные к обучению (очистка, нормализация, сплит)
6. Выбрать сеть согласно задаче и обучить ее (или любую иную open source)
7. Каждый эксперимент должен быть сформулирован как гипотеза
8. Должны быть добавлено отслеживание кривых и отрисовка графиков с помощью трекера
9. Должны быть выведены примеры корректных предсказаний и ошибочных
10. Должны быть рассчитаны метрики и проанализированы полученные результаты
11. Описать выводы + 3 новые гипотезы для повышения качества

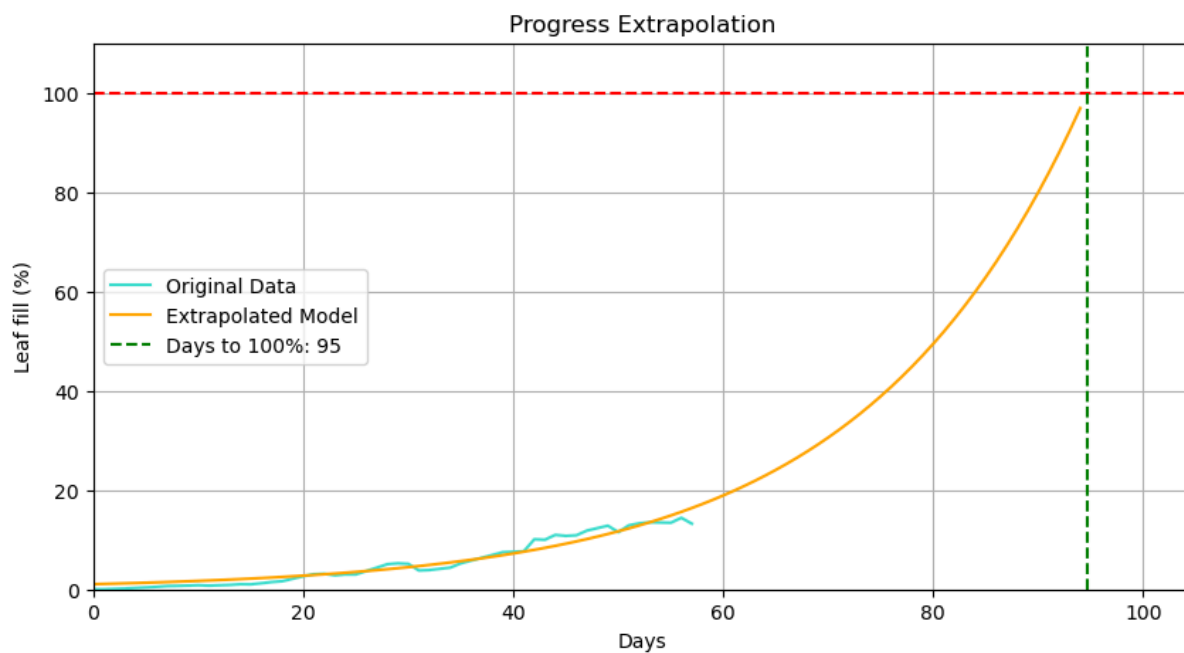
Выполнение

I. EDA

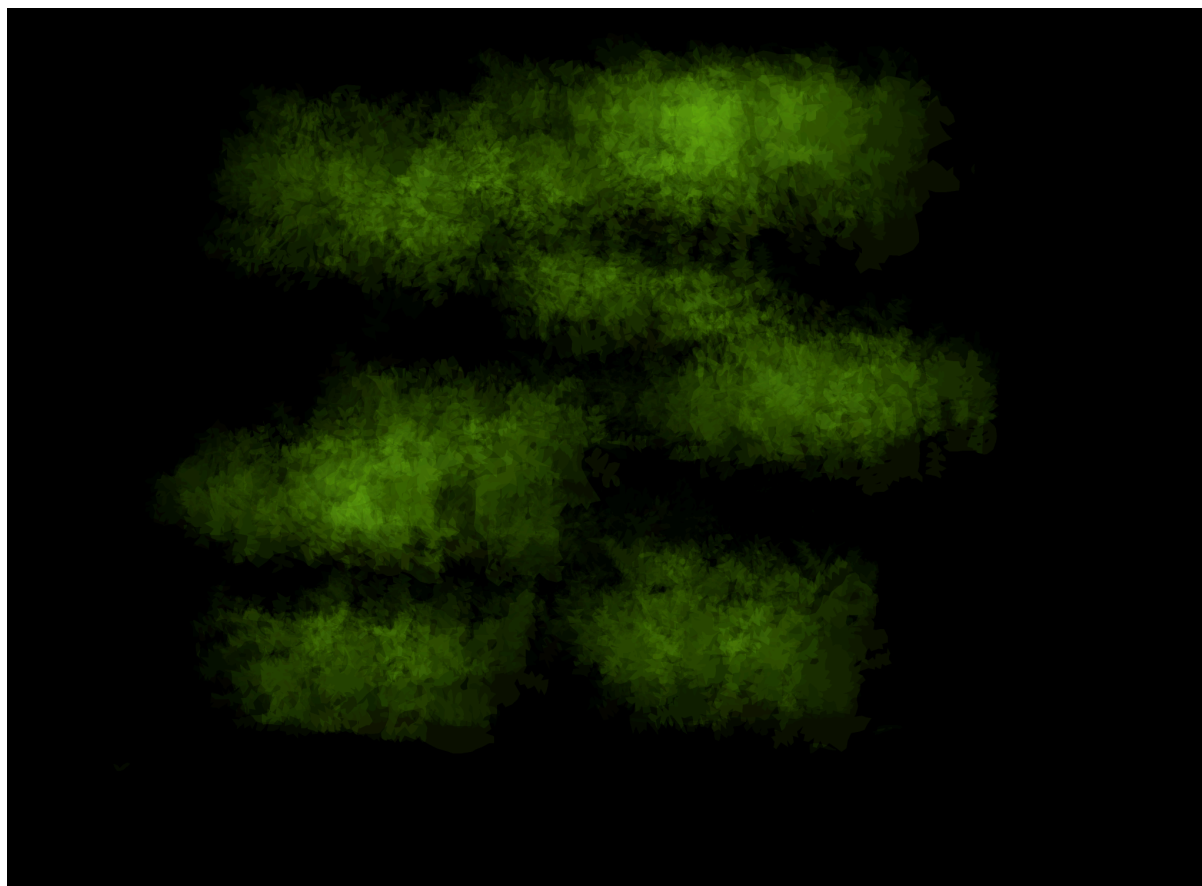
Построим временной ряд заполнения изображения листвой.



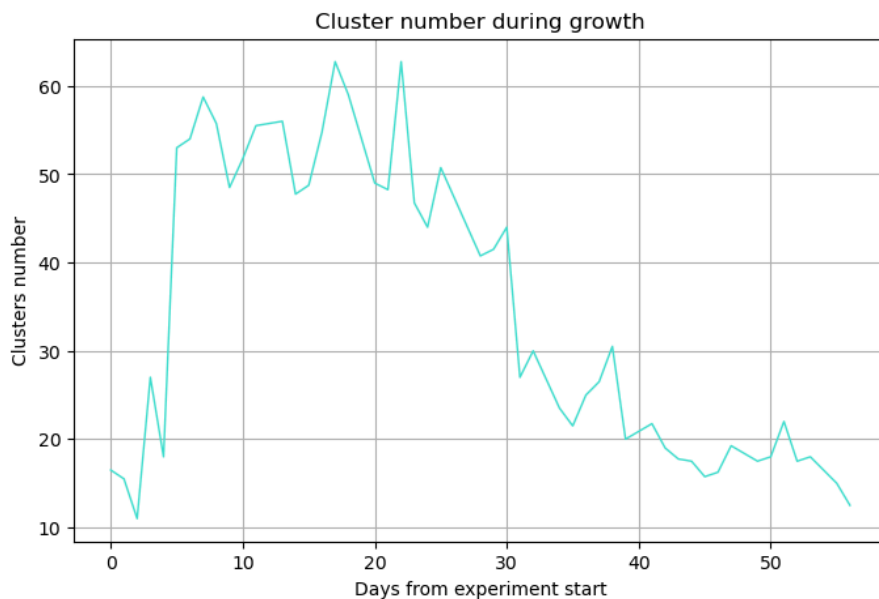
Проделаем экспоненциальную экстраполяцию и определим, через сколько дней после начала эксперимента каждое из растений займет всю площадь.



Построим HeatMap, чтобы увидеть, как распространяются листья. К сожалению, картинки не позиционированы относительно друг друга, поэтому карта не будет точной.

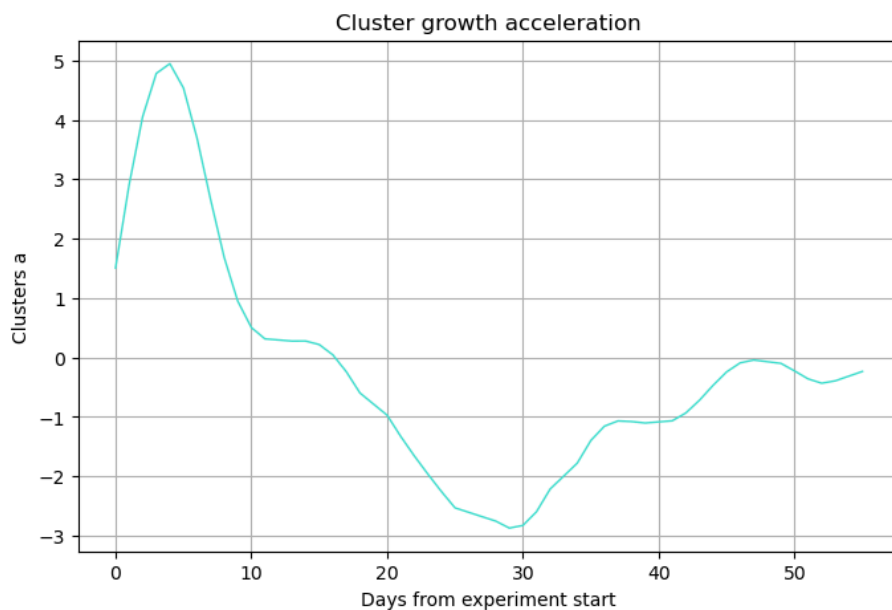


Посчитаем также, сколько отдельных кластеров (независимых ростков) образуется со временем. Усредним график количества кластеров, чтобы получить результаты с меньшим количеством выбросов.



Посчитаем корреляцию и поймем, что количество кластеров не зависит от дня эксперимента, хоть и имеет сильную связанность. Корреляция равна -0.57 . Рассчитаем медиану количества кластеров. В датасете указано, что в каждом контейнере вплоть до 40 семян. Будем считать, что проросло 27 из сорока семян, то есть коэффициент прорастания 0.675

Из сглаженного графика производной видим, что наибольшее прорастание наблюдается в дни около пятого. Итого наибольшее количество прорастаний на четвертый день эксперимента.



II. Риски и решения

1. Датасет содержит сбалансированное количество снимков со сбалансированным количеством этапов прорастания растений, что позволяет не беспокоиться о выборке и обработке входных данных.
2. Объем данных небольшой, модель может переучиться, запоминая обучающие изображения, а не обобщая знания для новых данных. Решение: Применять аугментацию данных (поворот, масштабирование, отражение), чтобы искусственно увеличить объем данных. Использовать dropout, регуляризацию для предотвращения переобучения.

III. Обучение

Архитектура модели

Была использована U-Net — популярная модель для задач сегментации. Она обладает следующими характеристиками:

1. Симметричная структура:
 - Состоит из энкодера (сжатие изображения, выявление ключевых признаков) и декодера (восстановление пространственной информации для построения маски).
2. Скип-соединения (skip connections):
 - Позволяют передавать информацию высокого разрешения с этапов энкодера на соответствующие уровни декодера, улучшая качество прогнозирования границ.

Детали модели

1. Входные данные:
 - Размеры входных изображений — (128, 128, 3) (RGB изображения).
 - Все данные были нормализованы (значения пикселей от 0 до 1).
2. Выходные данные:
 - Маска размером (128, 128, 1) с вероятностями (значения от 0 до 1).
 - Для бинаризации маски используется порог, например `0.5`.
3. Функция потерь:
 - Была использована комбинация Binary Cross Entropy и Dice Loss, что помогает учитывать как пиксельное сходство, так и пересечение

Исходный код модели

```
def create_unet_model(input_size=(IMG_HEIGHT, IMG_WIDTH, 3)):
    inputs = tf.keras.layers.Input(input_size)

    # Contracting path
    c1 = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
    c1 = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(c1)
    p1 = tf.keras.layers.MaxPooling2D((2, 2))(c1)

    c2 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(p1)
    c2 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(c2)
    p2 = tf.keras.layers.MaxPooling2D((2, 2))(c2)

    # Bottleneck
    c3 = tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same')(p2)
    c3 = tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same')(c3)

    # Expanding path
    u1 = tf.keras.layers.UpSampling2D((2, 2))(c3)
    u1 = tf.keras.layers.concatenate([u1, c2])
    c4 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(u1)
    c4 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(c4)

    u2 = tf.keras.layers.UpSampling2D((2, 2))(c4)
    u2 = tf.keras.layers.concatenate([u2, c1])
    c5 = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(u2)
    c5 = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(c5)

    outputs = tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid')(c5)
    model = tf.keras.Model(inputs, outputs)
    return model
```

IV. Выводы по обучению

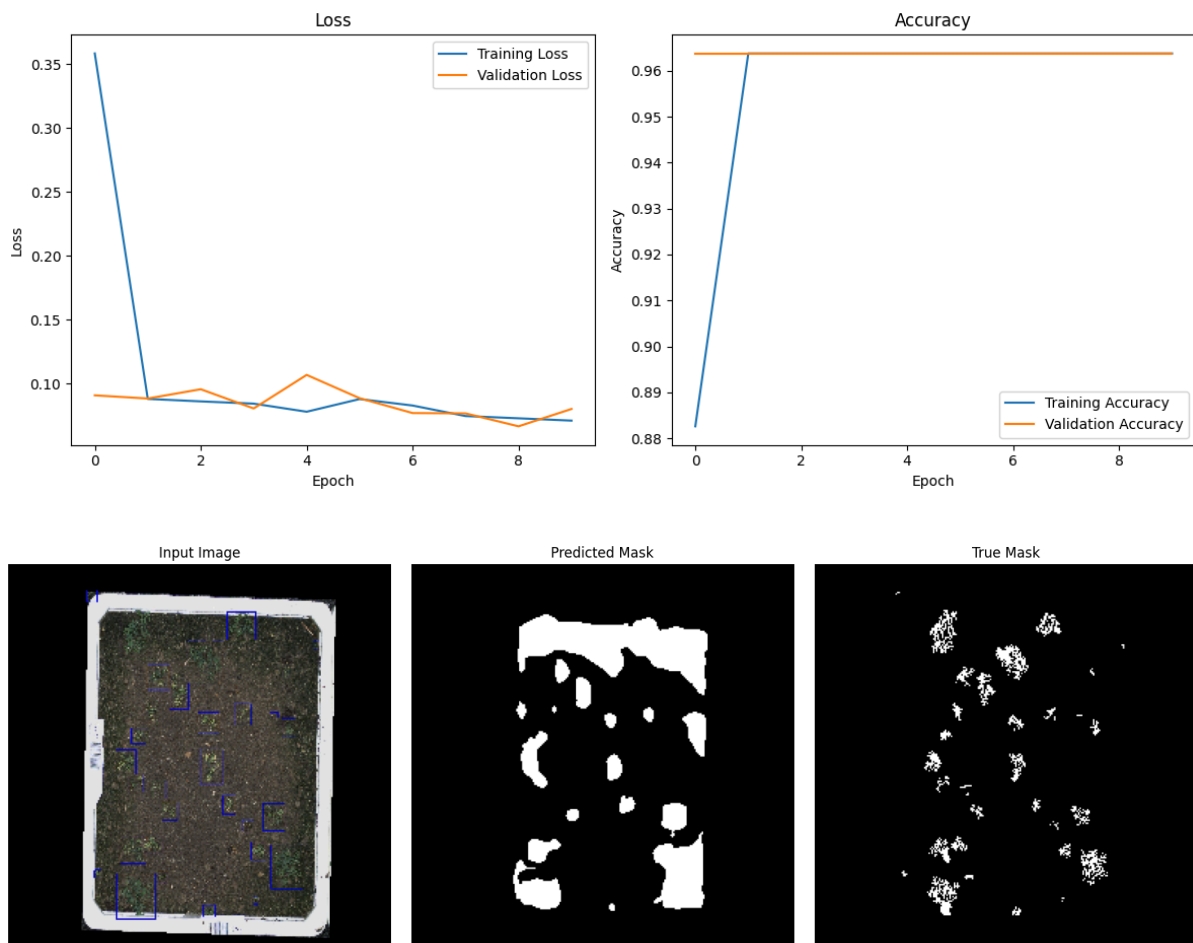
Метрики:

1. Dice coefficient: Показывает схожесть предсказанных и истинных масок.
2. IoU (Intersection over Union): Доля пересечения предсказанной и истинной областей.
3. Ассурасу: Общее количество правильно классифицированных пикселей.

Средний Dice (F1): 0.3263

Средний IoU: 0.2184

Средняя точность: 0.9361



V. Гипотезы по улучшению

Увеличить разрешение входных изображений

Из-за того, что исходное изображение в довольно большом разрешении, его сжатие, учитывая тот факт, что на общей площади растения занимают маленькое пространство, сильно ухудшает видимость растений. Увеличение входного разрешения энкодеров должно повысить общую точность распознавания.

Улучшить качество исходного изображения

Исходные изображения обладают малым контрастом и насыщенностью, из-за чего модели становится труднее различать края маски. Предварительная обработка изображения в виде повышения контрастности или насыщенности зеленого перед нормализацией должно помочь модели быстрее обучаться.

Увеличить количество данных и аугментации

Исходные изображения обладают высоким разрешением и детализацией, которую модель не способна рассмотреть. Разбиение картинки (tiling) и обучение модели на большем количестве получившихся картинок позволит добиться лучшего результата на итоговых данных

Тонкая настройка гиперпараметров

Модель возвращает вероятностные значения маски, которые мы отсекаем определенным значением в бинарное представление. Более тонким подбором параметра отсечки можно увеличить конечные метрики.