Диагностика биотического стресса растений методами объяснимого искусственного интеллекта

М.Р. Алибеков [1]

[1] *ННГУ им. Н.И. Лобачевского, проспект Гагарина 23, Нижний Новгород, 603022, Россия*

Аннотация

Исследованы методы предобработки цифровых изображений, существенно повышающие эффективность применения методов ML, а также ряд методов и моделей ML в качестве основы для построения простых и эффективных XAI сетей для диагностики биотических стрессов растений. Построено комплексное решение, включающее в себя этапы: автоматической сегментации; извлечения признаков; классификации ML-моделями. Выбраны лучшие классификаторы и векторы признаков. Исследование выполнено на открытом датасете PlantVillage Dataset. Лучшим по критерию F1-score=93% стал однослойный персептрон (SLP), обученный на полном векторе из 92 признаков (20 статистических, 72 текстурных). Время обучения на ПК с CPU Intel Core i5-8300H составило 189 минут. По критерию "F1-score/число признаков" лучшим стал также SLP, обученный на 7 главных компонентах, с F1-score=85%. Время обучения - 29 минут. Критерий "F1-score/количество+интерпретируемость признаков" отдает предпочтение отобранным 9 признакам и модели случайный лес, F1-score=83%. Программный комплекс для исследования выполнен в современной версии Python, с использованием библиотек OpenCV и моделей глубокого обучения, и готов для применения в точном земледелии.

Ключевые слова

Объяснимый искусственный интеллект, машинное обучение, обработка изображений, биотический стресс растений, диагностика, сегментация, рекурсивное исключение признаков.

Diagnosis of Plant Biotic Stress by Methods of Explainable Artificial Intelligence

M.R. Alibekov [1]

[1] *Lobachevsky State University of Nizhny Novgorod (UNN), 23 Gagarin Ave, Nizhny Novgorod, 603022, Russia*

Abstract

Methods for digital image preprocessing, which significantly increase the efficiency of ML methods, and also a number of ML methods and models as a basis for constructing simple and efficient XAI networks for diagnosing plant biotic stresses, have been studied. A complex solution has been built, which includes the following stages: automatic segmentation; feature extraction; classification by ML models. The best classifiers and feature vectors are selected. The study was carried out on the open dataset PlantVillage Dataset. The single-layer perceptron (SLP) trained on a full vector of 92 features (20 statistical, 72 textural) became the best according to the F1-score=93% criterion. The training time on a PC with an Intel Core i5-8300H CPU took 189 minutes. According to the criterion "F1-score/number of features", SLP trained on 7 principal components with F1-score=85% also became the best. Training time - 29 minutes. The criterion "F1-score/number+interpretability of features" favors the selected 9 features and the random forest model, F1-score=83%. The research software package is made in a modern version of Python using the OpenCV and deep learning model libraries, and is able for using in precision farming.

ГрафиКон 2022: 32-я Международная конференция по компьютерной графике и машинному зрению, 19-22 сентября 2022 г., Рязанский государственный радиотехнический университет им. В.Ф. Уткина, Рязань, Россия

EMAIL: MuradAlibekov2000@gmail.com (М.Р. Алибеков)

ORCID: 0000-0002-9201-8878 (М.Р. Алибеков)

Keywords
Explainable artificial intelligence, machine learning, image processing, plant biotic stress, diagnostics, segmentation, recursive feature elimination.

1. Введение

Заболевания сельскохозяйственных культур являются большой проблемой международной продовольственной безопасности – для некоторых культур болезни могут снизить урожай более, чем на 60%, и, как следствие, привести к голоду и к банкротству аграриев. Стоимость ежегодных потерь достигает сотен миллиардов долларов. Для своевременного противодействия необходимы интеллектуальные системы, позволяющие автоматически выявлять заболевания растений на возможно ранней стадии путем мониторинга их состояния из космоса или используя БПЛА.

В последние годы стало возможным создание систем, основанных на нейронных сетях ([1], [2], [3]), в том числе систем объяснимого искусственного интеллекта (XAI), решающих задачу ранней диагностики стресса растения на основе изображений, полученных в процессе мониторинга.

2. Описание набора данных

Для обучения высокоточных ML-классификаторов необходимо иметь представительный набор данных изображений, как больных, так и здоровых растений (рисунок 1).

Для этих целей в текущей работе использован открытый датасет “PlantVillage Dataset” [4], из которого были взяты лишь изображения листьев томатов. Это было сделано для оценки возможностей различных методов на количестве данных не выходящих за рамки 10000 примеров.

Классификация должна быть многоклассовой, так как должно определяться, не только здорово ли растение, а ещё и к какому типу заболеваний относится поразившая его болезнь. Разделение выборки на классы выглядит следующим образом:

1) Healthy (1591 изображение)
2) Bacterial_spot (2127 изображений)
3) Late_blight (1909 изображений)
4) Septoria_leaf_spot (1771 изображение)
5) Target_Spot (1404 изображения)
6) Tomato_Yellow_Leaf_Curl_Virus (2534 изображения)

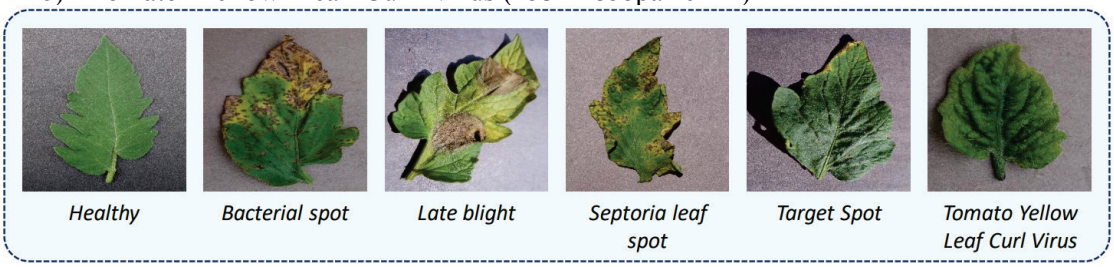


Рисунок 1 – Примеры типичных представителей рассматриваемых классов (болезней)

3. Эксперименты

Основная задача проведения экспериментов – построить наиболее простой, надежный, и, одновременно, объяснимый классификатор (XAI classifier) заболеваний растений по RGB-изображениям листьев. Естественным выбором для построения такого классификатора являются методы традиционного машинного обучения (ML).

Комплексное решение задачи будет включать следующие этапы:

1) Предобработка данных (автоматическое отделение листа от фона)

2) Извлечение признаков (текстурных и статистических)
3) Сокращение количества признаков
4) Обучение исследуемого набора классификаторов на разных наборах признаков
5) Оценка качества работы обученных классификаторов
6) Анализ полученных результатов

3.1. Сегментация листьев и фона

Первоначально даны RGB-изображения (размера 256×256) листьев растения (томата). Но помимо самого листа на изображении представлен фон (например, почва). Цель: получить изображение листа на черном фоне. Для этого будем использовать набор простых и широко известных методов, чтобы научиться проводить автоматическую разметку.

Подход, используемый в данной работе, основывается на методе построения признаков фона и листа, предложенном в статье “Automatic Leaf Extraction from Outdoor Images” [5].

Сегментация носит многошаговый характер.

Сначала избавляемся от “не зелёного” фона. Для этого поочередно избавляемся от белых, черных и синих областей, а затем применяем порог Оцу для разницы между индексами избыточного зеленого и красного [6] (см. рисунок2).

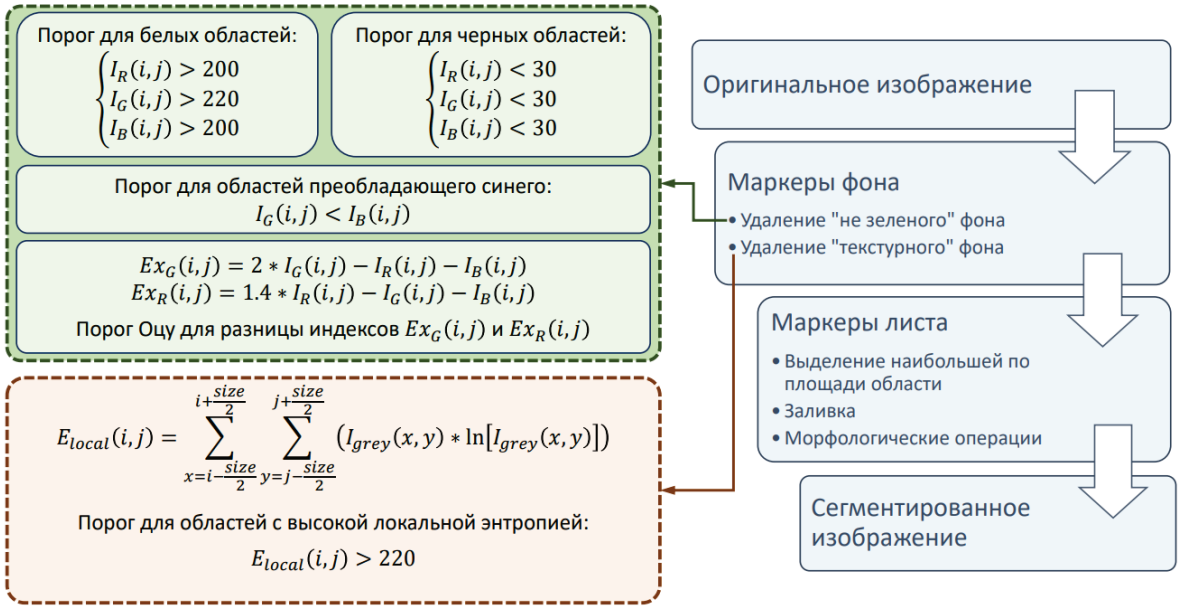
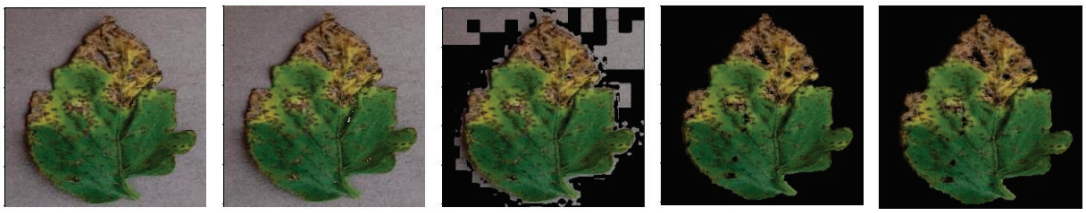


Рисунок 2 – Общая схема сегментации

Однако поражение листа может быть значительным и иметь другой цвет, отличный от здоровой части листа, и тогда изображение будет делиться не на две крупные в основном равномерно яркие области (фон и лист), а на три (фактический фон, здоровая часть листа, больная часть листа). Поэтому использовать только разницу между индексами недостаточно. Также учтем насыщенность (saturation), предварительно переведя изображения в цветовое пространство HSV. Значения насыщенности поделим на 2 группы методом Оцу, объединим маски, полученные на основе разницы индексов (избыточного зеленого и красного) и насыщенности. Далее удалим “текстурный” фон (рисунок 3).

В результате удаления “текстурного” фона получена маска листа с дефектами (“дырами”) внутри области листа, которые необходимо заполнить (Рисунок 4,a). Для этого построено бинарное изображение маски дефектов (0 – фон и лист, 1 – дефект). После чего бинарное изображение маски листа без дефектов получено объединением масок (Рисунок 4,c). Сглаживание излишней изрезанности (зашумленности) границы реализовано морфологическими операциями. Финалом автоматической сегментации является применение полученной маски к исходному изображению (Рисунок 4,e).



a) b) c) d) e)

Рисунок 3 – Результаты шагов: a) после удаления белых областей, b) после удаления черных областей, c) после удаления синих областей, d) после применения метода Оцу, e) после удаления "текстурного" фона

a) b) c) d) e)

Рисунок 4 – Изображения: a) маска листа с дефектами, b) маска дефектов, c) маска листа после объединения масок, d) маска листа после сглаживания операциями матморфологии, e) результат сегментации листа

Исходные классы существенно несбалансированы (например, 1370 сегментированных изображений для Healthy против 2163 – для Tomato_Yellow_Leaf_Curl_Virus). Для обеспечения баланса для каждого из 6 классов взято по 1000 автоматически сегментированных изображений. В результате, наш набор данных будет содержать 1000×6=6000 изображений. На рисунке 5 представлены примеры сегментированных изображений для рассматриваемых классов.

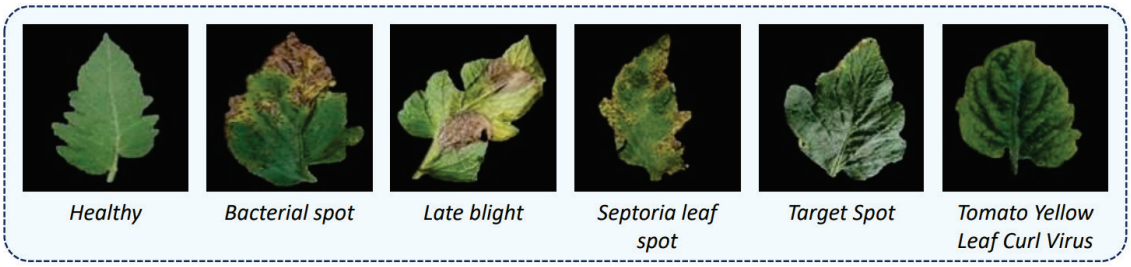


Рисунок 5 – Примеры сегментированных изображений для рассматриваемых классов

Для упрощения отслеживания прогресса выполнения затратной по времени сегментация всех изображений датасета создан временный Telegram бот, которому автоматически отправлялись обновления текущего статуса. Для отправки текущего прогресса использован модуль tqdm.contrib.telegram библиотеки tqdm.

3.2. Извлечение признаков

В качестве признаков извлекались статистические признаки изображения, получаемые на основе различных статистик, и текстурные, получаемые на основе GLCM ([7], [8]).

Для формирования статистических признаков сначала вычислены базовые статистические признаки для каждого канала RGB изображения листа в пределах его маски (фон игнорируется). Далее на их основе могут конструироваться признаки второго уровня. В качестве базовых статистических характеристик RGB изображения листа приняты:

- Среднее значение яркости для каждого канала изображения размера $N \times M$:

$$\mu_R = \frac{1}{NM}\sum_{i=0}^{N}\sum_{j=0}^{M} I_R(i,j), \quad (1)$$

где μ_R – среднее значение яркости для красного канала, $I_R(i,j)$ – значение красного канала пикселя (i,j). Аналогично, μ_G – для зеленого (I_G) и μ_B – для синего канала (I_B).

- Стандартное отклонение яркости для каждого канала изображения размера $N \times M$:

$$\sigma_R = \sqrt{\frac{1}{NM}\sum_{i=0}^{N}\sum_{j=0}^{M}(I_R(i,j) - \mu_R)^2}, \quad (2)$$

где σ_R – стандартное отклонение яркости для красного канала. Аналогично, σ_G – для зеленого и σ_B – для синего канала.

- Максимальное значение яркости для каждого канала изображения размера $N \times M$:

$$M_R = \max_{0\le i\le N, 0\le j\le M} I_R(i,j), \quad (3)$$

где M_R – максимальное значение яркости для красного канала. Аналогично, M_G – для зеленого и M_B – для синего канала.

- Минимальные значения яркости для каждого канала изображения размера $N \times M$:

$$m_R = \min_{0\le i\le N, 0\le j\le M} I_R(i,j), \quad (4)$$

где m_R – минимальное значение яркости для красного канала. Аналогично, m_G – для зеленого и m_B – для синего канала.

Не отрицая применение базовых признаков в чистом виде $(mean, std, max, min)$, можно сформировать из них дополнительные признаки, ориентированные на специфику задачи.

Построим таким образом отношения разности между максимальным и средним и между максимальным и минимальным значениями к стандартному отклонению значению яркости среди всех изображений в выборке в соответствующем канале, введя сюда таким образом признаки каждого из классов в целом:

$$\frac{M_R^{(i)} - \mu_R^{(i)}}{\frac{1}{C}\sum_{i=0}^{C}\sigma_R^{(i)}}, \quad (5)$$

$$\frac{M_R^{(i)} - m_R^{(i)}}{\frac{1}{C}\sum_{i=0}^{C}\sigma_R^{(i)}}, \quad (6)$$

где $\mu_R^{(i)}, \sigma_R^{(i)}$, $M_R^{(i)}$ и $m_R^{(i)}$ – среднее, стандартное отклонение, максимальное и минимальное значения яркости для красного канала (i)–ого изображения, C – количество изображений в выборке. Аналогично для зеленого и синего каналов.

Построим также разностные индексы ([6]):

- Разница между средними значениями яркости в красном и зеленом каналах:

$$\mu_R^{(i)} - \mu_G^{(i)} \quad (7)$$

- Разница между средними значениями яркости в зеленом и синем каналах:

$$\mu_G^{(i)} - \mu_B^{(i)} \quad (8)$$

- Отношение разности между средними значениями яркости в красном и зеленом каналах к модулю разности между средними значениями яркости в зеленом и синем каналах:

$$\frac{\mu_R^{(i)} - \mu_G^{(i)}}{\left|\mu_G^{(i)} - \mu_B^{(i)}\right|} \quad (9)$$

- Индекс избыточного зеленого (для средних значений яркости):

$$2 * \mu_G^{(i)} - \mu_R^{(i)} - \mu_B^{(i)} \quad (10)$$

- Индекс избыточного красного (для средних значений яркости):

$$1.4 * \mu_R^{(i)} - \mu_G^{(i)} - \mu_B^{(i)} \quad (11)$$

Построим также нормированные признаки – отношения средних значений яркости по отдельным каналам к сумме средних по всем 3 каналам, которые можно рассматривать также как барицентрические координаты реализаций класса:

$$\frac{\mu_B^{(i)}}{\sum_{k\in\{B,G,R\}}\mu_k^{(i)}} = \frac{\mu_B^{(i)}}{\mu_B^{(i)} + \mu_G^{(i)} + \mu_R^{(i)}}, \quad (12)$$

$$\frac{\mu_G^{(i)}}{\sum_{k\in\{B,G,R\}}\mu_k^{(i)}} = \frac{\mu_G^{(i)}}{\mu_B^{(i)} + \mu_G^{(i)} + \mu_R^{(i)}}, \quad (13)$$

$$\frac{\mu_R^{(i)}}{\sum_{k\in\{B,G,R\}}\mu_k^{(i)}} = \frac{\mu_R^{(i)}}{\mu_B^{(i)} + \mu_G^{(i)} + \mu_R^{(i)}}, \quad (14)$$

где $\mu_B^{(i)}$, $\mu_G^{(i)}$, $\mu_R^{(i)}$ – средние значения яркости для синего, зеленого и красного каналов (i)–ого изображения соответственно.

Полный список статистических признаков для (i)–ого изображения в итоге составит вектор из 20 признаков.

В качестве текстурных признаков используем признаки Харалика, которые основаны на использовании матриц смежности уровней серого (GLCM [9]).

Для вычисления GLCM, в первую очередь, нужно провести конвертацию исходного изображения I в оттенки серого I_{grey}. Традиционно, вместо 256 уровней серого используют квантование 256 значений на 8–32 уровней, что позволяет одновременно, как резко снизить объем вычислений и используемой памяти, так и избавиться от шума. В нашем случае, примем число уровней серого (L) равным 8. В результате, размер GLCM матрицы будет уменьшен более, чем в 1000 раз до 8x8.

В зависимости от логики обработки, фоновые пиксели могут выделены в особый класс «0» и квантованию в диапазон [1,8], в этом случае, можем подвергнуть диапазон значений с 1 до 255:

$$U_k = \left[1 + \frac{255}{L}(k-1);\ 1 + \frac{255}{L}k\ \right], \quad (15)$$

где L – число уровней серого.

Если $I_{grey}(i,j) \in U_k$ при некотором $k \in \{1,2,\dots,L\}$, то $Q(i,j) = k$ – квантованное значение яркости в пикселе с координатами (i,j).

От квантования изображения, переходим к построению GLCM матриц. В данной работе используются матрицы $G_{r,\theta}$, где $r \in \{1,2,4\}$, а $\theta \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$, что даёт 12 матриц. Для каждой из них вычисляются 6 текстурных признаков Харалика ([10], [11], [12]): контраст, различие, однородность, второй угловой момент, энергия и корреляция, что даёт 6×12=72 признака.

Объединив текстурные и статистические, получим 92 признака для каждого изображения. Для унификации последующего анализа все признаки переведены в диапазон [0, 1].

3.3. Снижение размерности

Количество признаков уменьшали с помощью двух методов: метода главных компонент (PCA) и метода рекурсивного исключения признаков (RFE).

При использовании PCA [13], [14], получаем меньший набор новых признаков (называемых главными компонентами), каждый из которых является линейной комбинацией исходных. Определяем количество главных компонент (m), которое оставим, как значение, при котором δ будет больше 97% (т.е. $t = 0.97$). Иными словами, $m = \underset{\delta(m^*) \geq 0.97}{argmin}\ \{m^*\}$.

Как видно из рисунка 6, взяв в качестве порога значение суммарной дисперсии, равное 97%, получаем: $\underset{\delta(m^*) \geq 0.97}{argmin}\ \{m^*\} = 7$, т.е. оставляем только 7 главных компонент, тем самым более, чем на порядок уменьшаем количество признаков и данных, на которых будем тренировать различные модели машинного обучения, с (6000, 92) до (6000, 7).

Метод RFE, рекурсивного исключения признаков (recursive feature elimination) ([15], [16], [17]), состоит в том, что начиная с полного набора признаков, на каждом шаге алгоритма удаляется один наименее “важный” из признаков, пока не будет достигнуто желаемое количество (размерность нового вектора признаков). После отбора данным методом останутся лишь наиболее релевантные признаки из исходных, причем в том же виде, в каком они были изначально.

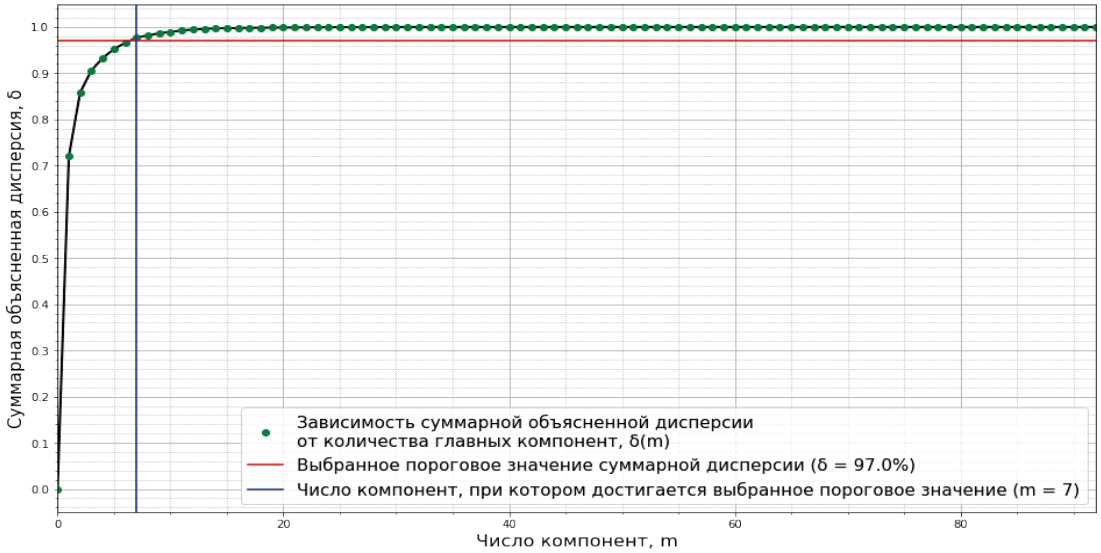


Рисунок 6 – График зависимости суммарной объясненной дисперсии от количества главных компонент, δ(m)

Метод рекурсивного исключения признаков требует много времени и больших вычислительных затрат (из-за большого числа обучений модели–классификатора на разных наборах признаков). Разумным будет перед этим уменьшить число признаков более простыми методами. Для этого избавимся от сильно коррелированных признаков, для которых абсолютное значение коэффициента корреляции Пирсона $|r_{ij}| \geq 0.7$.

Убрав в каждой паре сильно коррелированных признаков по одному из них, получили набор, состоящий из 9 слабо/средне коррелированных:

1) $MEAN\_B$
2) $MEAN\_G$
3) STD_B
4) $RATIO\_DIF\_MAX\_MEAN\_B$
5) $DIF\_MEANS\_R\_G$
6) $DIF\_MEANS\_G\_B$
7) $RATIO\_DIF\_G\_B\_ABS\_DIF\_R\_G$
8) $EXCESS\_R$
9) $CORRELATION\_1\_0$

За счёт удаления сильно коррелированных признаков, удалось заметно снизить количество признаков до достаточно небольшого набора, для которого можно использовать метод RFE.

Для рекурсивного исключения признаков в качестве оценщика взяли модель RF (случайный лес), потому что его можно обучить довольно быстро. Данная модель может переобучиться на тренировочных данных, что не позволит качественно классифицировать реальные данные. Чтобы этого не произошло, вместо RFE использовали его аналог RFECV – метод рекурсивного исключения признаков с перекрестной проверкой. Глядя на рисунке 7, можно увидеть претендентов на количество признаков: 5 и 9. Несмотря на то, что после 5 признаков улучшение небольшое, оно все же есть: при 5-ти признаках значение показателя чуть меньше 0.8, а при 9-ти – чуть больше 0.8. Поэтому рассмотрим оба варианта. В случае 9-ти признаков оставляем все найденные путем удаления сильно скоррелированных, а для 5-ти – оставляем следующий набор:

1) $MEAN\_B$

2) STD_B
3) $DIF\_MEANS\_G\_B$
4) $RATIO\_DIF\_G\_B\_ABS\_DIF\_R\_G$
5) $CORRELATION\_1\_0$

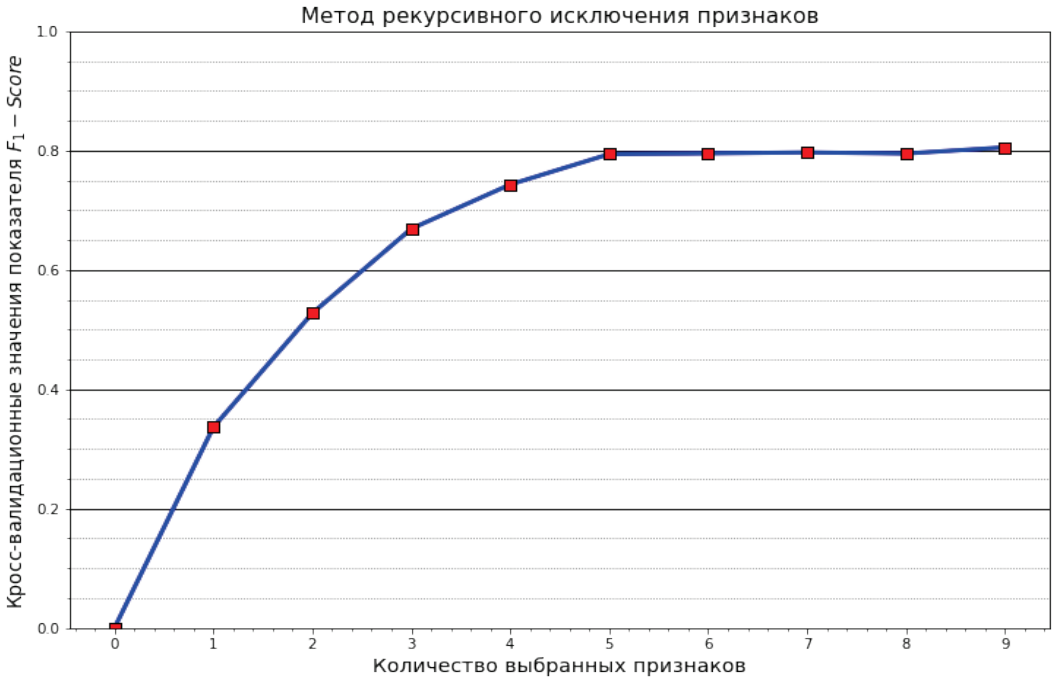


Рисунок 7 – График зависимости значения показателя Average F1–Score для перекрестной проверки от количества выбранных признаков

Таким образом, в результате отбора более, чем на порядок, уменьшилось количество признаков и данных, на которых можно тренировать различные модели машинного обучения, с (6000, 92) до (6000, 9) в одном случае, и с (6000, 92) до (6000, 5) в другом.

3.4. Классификация

В итоге, у нас сформированы данные, которые можно использовать для классификации. Размер данных: (6000, 92), т.е. 6000 изображений, для каждого из которых вычислены 6 наборов признаков.

1) Статистические признаки (20 признаков).
2) Текстурные признаки (72 признака).
3) Все признаки (92 признака).
4) Главные компоненты (7 признаков).
5) Отобранные признаки (5 признаков).
6) Отобранные признаки (9 признаков).

Этот наш набор данных разделяем на тренировочный и тестовый в соотношении 4:1. В тренировочный набор попадут 4800 изображений, а в тестовый – 1200.

При обучении исследуемых классификаторов на тренировочном наборе данных, настраиваем гиперпараметры. Начинаем обучение со значений гиперпараметров по умолчанию, после чего с помощью GridSearchCV (модуля model_selection библиотеки Scikit–Learn) находим для них оптимальные значения.

Исследуемые классификаторы:

1) kNN (метод k–ближайших соседей);
2) DT – Decision Trees (деревья решений);
3) LR – Logistic Regression (модифицированная версия логистической регрессии для мультиклассовой классификации);

4) RF – Random Forest (случайный лес);
5) MLP – Multilayer Perceptron (в данном случае одноуровневый персептрон – SLP, т.е. простейшая полносвязная нейронная сеть).

После обучения и настройки гиперпараметров тестируем классификаторы на тестовой выборке. Т.к. задача мультиклассовой классификации, а не бинарной, то для оценки качества классификации использовано среднее значение метрики F1–score по всем классам [18], вычисляемое следующим образом (14):

$$Average\ F1\text{–score} = \frac{1}{K}\sum_{i=1}^{K} F1\text{–score}_i, \tag{14}$$

где K – число классов, $F1\text{–score}_i$ – значение метрики для i–ого класса.

3.5. Результаты экспериментов

Для указанных выше 6 векторов признаков, натренированы и протестированы 5 исследуемых классификаторов. Их сравнение по показателю Average F1–score приведено на (рисунок 8).

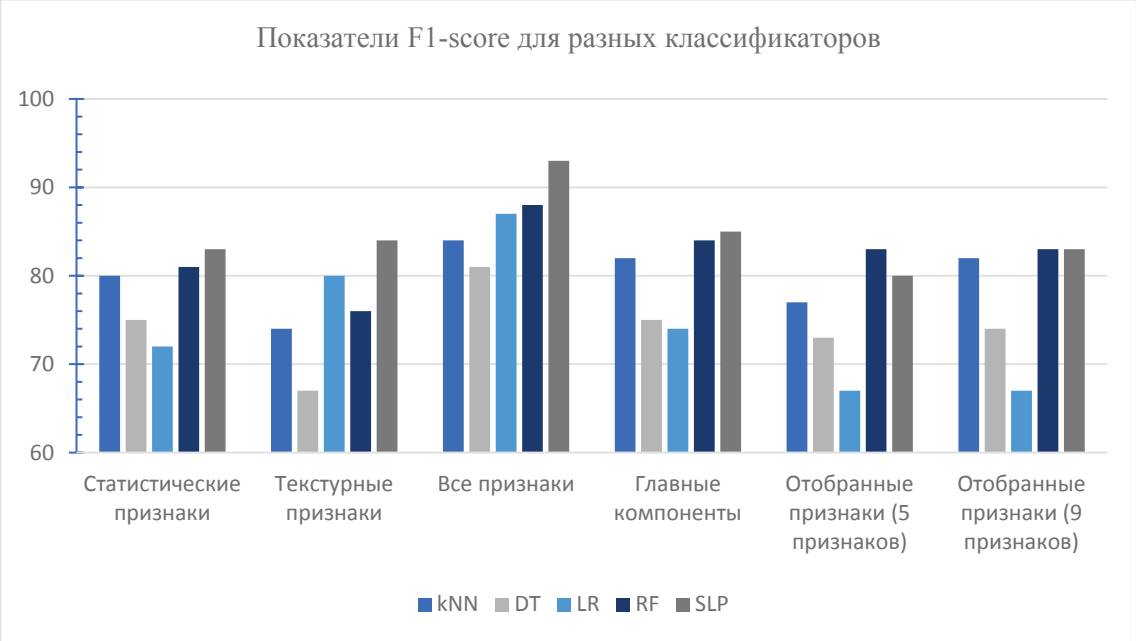


Рисунок 8 – Показатели F1–score для разных классификаторов, обученных на обозначенных 6 вариантах вектора признаков

Абсолютным лидером с Average F1–score = 93% стала SLP модель на полном векторе 92 признаков. При обучении на сокращенном векторе из 7 “новых” PCA–признаков, результат оказался лучше результатов соответствующих моделей, обученных на 20 статистических и 72 текстурных признаках по отдельности. Лучшей моделью по критерию "качество/количество параметров" оказалась SLP, обученная на основе PCA–признаков, с Average F1–score = 85%. Качество классификации упало на 8%, но число признаков уменьшилось в ≈13 раз (с 92 до 7), и, вместе с ним, затраты времени и ресурсов на обучение и настройку гиперпараметров.

Применение метода главных компонент создает проблемы, когда речь идет об интерпретируемости, потому что PCA вынуждает оперировать с трудно интерпретируемыми и контекстнозависимыми признаками, практически исключающими объяснимость результата. Если же целевым критерием будет “качество/количество параметров/объяснимость сети”, то предпочтительны 9 или 5 признаков, оставшихся в результате применения метода рекурсивного исключения. При обучении моделей (кроме LR) на векторе признаков, состоящем из отобранных, можно получить весьма неплохие результаты, превосходящие или не сильно уступающие моделям, обученным исключительно на статистических или текстурных признаках. С точки зрения объяснимости предпочтителен случайный лес, набравший 83%. Несмотря на то,

что качество классификации упало на 10%, по сравнению с полным набором признаков, количество признаков в этом случае уменьшено с 92 до 9 (т.е. в ≈10 раз) или до 5 (т.е. в ≈18 раз) интерпретируемых исходных признаков.

Результаты по гиперпараметрам для лучших моделей приведены в таблице 1.

Таблица 1 – Гиперпараметры, при которых достигается оптимальный Average F1–score (непредставленные параметры принимаются равными значениям по умолчанию в SciKit–Learn)

Метод	Набор признаков	Оптимальные гиперпараметры
SLP	Все признаки	hidden_layer_sizes = (550,), solver = 'lbfgs', alpha = 0.001
SLP	Главные компоненты	hidden_layer_sizes = (160,), activation = 'tanh', alpha = 0.001
RF	9 отобранных признаков	n_estimators = 200
RF	5 отобранных признаков	n_estimators = 210

Сравним наши результаты с представленными в работах последних 5 лет (таблица 2).

Таблица 2 – Сравнение методов классификации стресса растений (датасет PlantVillage)

Метод	Статья	Год	Метрика
SVM	Hlaing C.S., Zaw S.M.M [19]	2017	84.7% (accuracy)
GLSVD + SVM	Zhang, S., et al. [20]	2018	91.2% (accuracy)
VGG–16	Rangarajan A.K., et al. [21]	2018	96.2% (accuracy)
MobileNet	Elhassouny A, Smarandache F. [22]	2019	90.3% (accuracy)
Modified Inception v3	Toda Y, Okura F. [23]	2019	97.1% (accuracy)
DenseNet201	Ngugi L.C., et al. [24]	2020	99.7% (f1–score)
C–GAN + DenseNet121	Abbas A., et al. [25]	2021	97.1% (accuracy)
VGG–19	Ahmed S., et al. [26]	2022	99.5% (accuracy)
EfficientNet–B0	Ahmed S., et al. [26]	2022	96.9% (accuracy)
Customized LSTM	Rana S., et al. [27]	2022	84.3% (accuracy)
RFE + RF	Данная статья	2022	83.2% (f1–score)
SLP	Данная статья	2022	92.6% (f1–score)

4. Заключение

Исследование проведено с целью построения простых и надежных методов XAI для решения задачи диагностики биотических стрессовых состояний (болезней) растений. В качестве исходных данных использован открытый датасет PlantVillage Dataset, включающий 1591 изображение здоровых растений и 9745 примеров изображений для 5 заболеваний: Bacterial spot; Late blight; Septoria leaf spot; Target Spot; Tomato Yellow Leaf Curl Virus. Построено комплексное решение, включающее в себя этапы: автоматической сегментации; извлечения признаков; классификации ML–моделями. Решение акцентирует внимание на тщательной предобработке входных изображений с целью обеспечения полноты сегментации листа и полноты признаков, характеризующих аномалии листа, связанные с заболеваниями. Построен полный вектор признаков, насчитывающий 92 признака и состоящий из двух концептуальных групп: статистической (20 признаков) и текстурной (72 признака) ориентированных на XAI решение. Выделено 6 исследуемых вариантов вектора признаков: полный; концептуальные группы (20 и 72); PCA (7 признаков); лучшие некоррелированные (9 и 5 признаков). В 9 и 5 признаков вошел 1 текстурный и 8 и 4, соответственно, статистических.

В качестве лидеров XAI решений выявлены: 1) однослойный персептрон (SLP) – лучший на всех вариантах вектора признаков, кроме случая 5 признаков; 2) случайный лес (RF) – лучший на 5 признаках. Максимум по критерию Average “F1–score” (93%) показала модель SLP, обученная на полном векторе признаков. Обучение и настройка гиперпараметров на ПК с CPU

Intel Core i5–8300H заняли 189 минут. RF по 5 признакам обеспечивает Average “F1–score” = 83%.

Программный комплекс для исследования, выполненный в современной версии Python с использованием библиотек OpenCV и моделей глубокого обучения, пригоден для применения в точном земледелии. Сравнение с результатами классификации стресса растений в публикациях последних 5 лет показывает, что несмотря на то, что в нескольких публикациях результаты детектирования стресса растений по метрикам F1–score и Accuracy приблизились к 100%, но результаты данной работы остаются интересными и перспективными для развития, особенно для случая именно биологических источников стресса. Планируется дальнейшая оптимизация состава статистических признаков и XAI–классификатора для их обработки.

5. Благодарности

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации, соглашение № 075–15–2020–808.

6. Список источников

[1] Evaluating Plant Disease Detection Mobile Applications: Quality and Limitations [Электронный ресурс] / A. Siddiqua и др. // Agronomy. 2022. № 12(8). URL: https://doi.org/10.3390/agronomy12081869 (дата обращения 12.08.2022).

[2] In–field early disease recognition of potato late blight based on deep learning and proximal hyperspectral imaging [Электронный ресурс] / Chao Qi и др. // 2021. URL: https://arxiv.org/ftp/arxiv/papers/2111/2111.12155.pdf (дата обращения 12.08.2022).

[3] Щетинин Е.Ю. Распознавание заболеваний растений на основе анализа их изображений глубокими нейронными сетями // Всероссийская конференция “Информационно–телекоммуникационные технологии и математическое моделирование высокотехнологичных систем” (Москва, РУДН, 13–17 апреля 2020 г) / Москва: РУДН, 2020. Т. 1. С. 326–328.

[4] David P. Hughes, Marcel Salathé. An open access repository of images on plant health to enable the development of mobile disease diagnostics [Электронный ресурс] // 2016. URL: https://arxiv.org/ftp/arxiv/papers/1511/1511.08060.pdf (дата обращения 23.02.2022).

[5] Leaf Extraction from Outdoor Images [Электронный ресурс] / Nantheera Anantrasirichai, Sion Hannuna, Nishan Canagarajah. Automatic // 2017. URL: https://arxiv.org/ftp/arxiv/papers/1709/1709.06437.pdf (дата обращения 23.02.2022).

[6] Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. / D. M. Woebbecke, G. E. Meyer, K. Von Bargen, D. A. Mortensen // Transactions of the ASAE. 1995. № 38(1). С. 259–269.

[7] P.K. Sethy, N.K. Barpanda, A.K. Rath. Detection & Identification of Rice Leaf Diseases using Multiclass SVM and Particle Swarm Optimization Technique // International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2019. № 8. С. 108–120.

[8] Jing Yi Tou, Phooi Yee Lau, Yong Haur Tay. Computer vision–based wood recognition system // Proceedings of international workshop on advanced image technology. 2007. С. 1–6.

[9] Фраленко В.П. Методы текстурного анализа изображений, обработка данных дистанционного зондирования Земли // Программные системы: теория и приложения. 2014. № 4(22). С. 19–39.

[10] R. M. Haralick, K. Shanmugam, I. Dinstein. Textural features for image classification // Transactions on Systems, Man, and Cybernetics. 1973. № 6. С. 610–621.

[11] Тымчук А.И. О текстурных признаках в задаче сегментации аэрофотоснимков на основе матриц яркостной зависимости // Кибернетика и программирование. 2018. № 6. С. 31–39.

[12] Чехина Е.А. Обзор методов текстурного анализа изображений // Евразийское Научное Объединение. 2020. № 6–2(64). С. 160–162.

[13] Pearson C. On lines and planes of closest fit to systems of points in space // The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901. № 11(2). С. 559–572.
[14] Балабанов А.С., Стронгина Н.Р. Анализ данных в экономических приложениях: учебное пособие // Н. Новгород: Изд–во Нижегородского гос. университета, 2004. 135 с.
[15] Gene selection for cancer classification using support vector machines / I. Guyon, J. Weston, S. Barnhill, V. Vapnik // Machine Learning. 2002. № 46. С. 389–422.
[16] Comparative study of techniques for large–scale feature selection / F.J. Ferri, P. Pudil, M. Hatef, J. Kittler // Machine Intelligence and Pattern Recognition. 1994. № 16. С. 403–413.
[17] Machine Learning to Identify Flexibility Signatures of Class A GPCR Inhibition / J. Bemister–Buffington, A.J. Wolf, S. Raschka, L.A. Kuhn // Biomolecules. 2020. № 10(3). С. 454–476
[18] R.P. Espíndola, N.F.F. Ebecken. On extending F–measure and G–mean metrics to multi–class problems // WIT Transactions on Information and Communication Technologies. 2005. № 35. С. 25–34.
[19] Hlaing C.S., Zaw S.M.M. Model–based statistical features for mobile phone image of tomato plant disease classification. // 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). 2017.
[20] Three–channel convolutional neural networks for vegetable leaf disease recognition [Электронный ресурс] / S. Zhang и др. // Cognitive Systems Research. 2018. URL: https://doi.org/10.1016/j.cogsys.2018.04.006 (дата обращения 11.08.2022).
[21] Rangarajan A.K., Purushothaman R., Ramesh A. Tomato crop disease classification using pre–trained deep learning algorithm. // Procedia Computer Science. 2018. № 133. С. 1040–1047.
[22] Elhassouny A, Smarandache F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. // 2019 International Conference of Computer Science and Renewable Energies (ICCSRE). 2019. № 7(5). С. 1–12.
[23] Toda Y, Okura F. How convolutional neural networks diagnose plant disease [Электронный ресурс] // Plant Phenomics. 2019. № 2019(3). URL: https://doi.org/10.34133/2019/9237136 (дата обращения 11.08.2022).
[24] Ngugi L.C., Abelwahab M., Abo–Zahhad M. Recent advances in image processing techniques for automated leaf pest and disease recognition – A review. // Information Processing in Agriculture. 2020. № 8(1). С. 27–51.
[25] Tomato plant disease detection using transfer learning with C–GAN synthetic images. / A. Abbas и др. // Computers and Electronics in Agriculture. 2021. № 187. С. 106279.
[26] Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification / S. Ahmed и др. // IEEE Access. 2022. № 10. С. 68868–68884.
[27] Tomato Leaf Disease Detection using Customized Transfer Learning Architectures and LSTM [Электронный ресурс] / S. Rana и др. // 2022. URL: http://dx.doi.org/10.13140/RG.2.2.26376.29443 (дата обращения 16.08.2022).