МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Направление подготовки:
«Фундаментальная информатика и информационные технологии»
Профиль подготовки:
«Компьютерная графика»

КУРСОВАЯ РАБОТА

Тема:
«Методы машинного обучения в задаче обнаружения и классификации болезней листьев томатов»

Выполнила:
студентка группы 381706-1
Максимова Ирина Игоревна

_____
(подпись)

Научный руководитель:
проф. каф. МОСТ, д-р техн. наук
Турлапов Вадим Евгеньевич

_____
(подпись)

Нижний Новгород
2020

Оглавление

1. Введение

Болезни растений являются одной из главных угроз глобальной продовольственной безопасности. Ежегодно болезни растений приводят к потере 10–16% мирового урожая сельскохозяйственных культур, стоимость которого оценивается в 220 миллиардов долларов [3]. Одновременно с потерей урожая происходит прирост населения мира. Согласно прогнозу ООН, численность населения мира достигнет 9,8 млрд. человек в 2050 году и 10,7 млрд. в 2100 году [4]. Поэтому, чтобы удовлетворить потребности в продовольствии постоянно растущего населения, необходимо свести к минимуму потери мирового урожая.

Одним из главных шагов на пути к сохранению урожая является использование эффективных методов для выявления заболеваний растений.

Мониторинг сельскохозяйственных культур играет ключевую роль в успешном выращивании урожая. В настоящее время основным подходом, используемым на практике, является наблюдение невооруженным глазом эксперта. Получение заключения эксперта, диагностика заболевания и обращение за консультацией к практикующим врачам - долгая, дорогая и трудоемкая практика. Поэтому в настоящее время возникает острая необходимость в создании быстрых и недорогих современных методов обнаружения заболеваний растений.

Актуальность данной работы заключается в автоматизации процесса мониторинга сельскохозяйственных угодий. **Проблематика** данной работы заключается в обнаружении и классификации болезней листьев томатов.

Основными возбудителями болезней культурных растений являются паразитические бактерии, грибы и вирусы. Данные патогены вызывают отклонения жизненных процессах поражённого растения и приводят к значительным изменениям не только его внутреннего состояния, но и внешнего вида. Внешними признаками болезней растений являются: увядание, разрушение, мумификация и деформация органов растения, отмирание, изменение цвета тканей, появление гнили, пятен, налета, пустулы, нароста или опухоли, выделение камеди.

Поскольку появление побочных внешних эффектов сигнализирует о реакции растения на заболевание, то обнаружение больных растений может проводиться путем выявления зараженной области. С точки зрения анализа изображения всякое внешнее проявление болезни листа сопровождается изменением ее статистических и текстурных признаков. В настоящее время для анализа таких признаков используется машинное обучение, поскольку оно обладает высоким показателем точности и большим потенциалом.

В зависимости от задачи, используются разные методы машинного обучения. Касательно рассматриваемой темы в работе будут применяться алгоритмы классификации. Каждый из них имеет свои преимущества и недостатки в задаче классификации. Поэтому необходимо провести обзор классификаторов и определить наиболее эффективный метод классификации, применяемый в исследуемой области.

Целью данной работы является поиск наилучшего алгоритма классификации болезней листьев растений. **Объектом** исследования являются методы извлечения признаков изображения и базовые алгоритмы машинного обучения, а **предметом** исследования – обнаружение и классификация болезней листьев томатов по внешним симптомам.

В рамках преследуемой цели сформулирован общий подход по созданию автоматизированной системы классификации болезней листьев растений. Предлагаемый подход состоит из следующих этапов:

Подготовка данных. Для начала необходимо найти базу данных, на которой будет выполняться тренировка и тестирование алгоритма машинного обучения. Данные должны содержать размеченные изображения как больных, так и здоровых листьев исследуемого сорта растения. Качество листовых изображений играет важную роль, и поэтому необходимо использовать надежный источник. Выборка должна быть достаточно большой для обеспечения высокой обобщающей способность будущей модели.

Извлечение и анализ статистических и текстурных признаков. На данном шаге мы уменьшаем размер данных путем получения необходимой информации из изображения. Предполагается, что детектировать заболевание, то есть обнаружить пораженную область, возможно с помощью статистических признаков, но для классификации болезни их может оказаться недостаточно. Поверхность листьев растений достаточно репрезентативна, чтобы различать их заболевания с высокой точностью. Главной характеристикой поверхности является текстура. Анализ текстур направлен на поиск уникального способа представления основных характеристик текстур и представления их в некоторой более простой, но уникальной форме. Поэтому, в рамках задачи классификации, будут использоваться текстурные характеристики листьев. Наиболее распространенным вариантом получения статистических характеристик текстуры является матрица совместной встречаемости уровней серого тона (GLCM).

Тестирование алгоритмов машинного обучения. На основе извлеченных признаков осуществляется классификация болезней листьев растений. В данной работе будут рассмотрены следующие классификаторы: деревья решений (DTC), случайный лес (RF), мультиклассовый метод опорных векторов (MSVM), k-ближайших соседей (KNN), одноуровневый персептрон (*1 Layer MLP*).

Вышеизложенные этапы являются базовыми и обязательными в задаче классификации болезней листьев растений. Качественная реализация каждого этапа позволит с высокой точностью диагностировать заболевание растения.

2. Постановка задачи

С учетом подхода, описанного в предыдущей главе, для достижения поставленной цели необходимо:

1. Освоить теоретический минимум. А именно:
   1.1 Изучить статистический метод исследования текстур на базе GLCM.
   1.2 Ознакомиться с методами машинного обучения, используемыми в задачах классификации – DTC, RF, MSVM, KNN и 1 Layer MLP.
   1.3 Рассмотреть способы оценки качества работы классификатора.
2. Подготовить базу данных для обучения и тестирования классификатора.
3. Извлечь признаки для всех изображений из базы данных, на основе которых будет выполняться классификация.
4. Исследовать качество работы различных классификаторов на извлеченных данных и сделать выводы по возможности применения каждого из рассматриваемых классификаторов и признаков для обнаружения и классификации болезни.

3. Теоретическая часть

3.1 Матрица GLCM

Любую текстуру можно описать как пространственное распределение значений яркости локальной области изображения с ростом расстояния между оцениваемыми точками. Такое описание реализует матрица совместной встречаемости уровней серого тона - Grey Level Co-occurence Matrix (GLCM) [8].

Матрица *GLCM* представляет собой оценку плотности распределения вероятностей второго порядка, полученную по изображению в предположении, что плотность вероятности зависит лишь от расположения двух пикселей. Обозначим эту матрицу $P(i,j,d,\varphi)$, где i и j – яркости соседних точек на изображении, расположенных на расстоянии d друг от друга, при угловом направлении φ. Размер матрицы определяется количеством градаций яркости изображения.

Матрица GLCM обычно приводится к одному из двух видов:

- Симметричная матрица:

$$S(i,j,d,\varphi) = P(i,j,d,\varphi) + P(i,j,d,\varphi)^T$$

- Нормализованная матрица (матрица условной вероятности):

$$N(i,j,d,\varphi) = \frac{P(i,j,d,\varphi)}{\sum_{m,k} P(m,k,d,\varphi)}$$

Как только матрицы вычислены, из них извлекаются текстурные признаки данного класса текстуры. Для этой цели Haralick, Shanmugam и Dinstein [8] предложили 14 мер. Позже Conners и Harlow [9] отметили, что многие из них коррелируют друг с другом и только 5 из этих 14 мер являются достаточными. Далее приведем описание этих мер.

Корреляция. Измеряет линейную зависимость интенсивностей пикселей относительно друг друга.

$$\text{корреляция} = \sum_{i,j} P_{ij} * \frac{(i-\mu_i) * (j-\mu_j)}{\sigma_i * \sigma_j} \quad (1)$$

$$\mu_i = \sum_{i,j} i * P_{ij}\,, \qquad \mu_j = \sum_{i,j} j * P_{ij} \quad (2)$$

$$\sigma_i^2 = \sum_{i,j} P_{ij} * (i-\mu_i)^2\,, \qquad \sigma_j^2 = \sum_{i,j} P_{ij} * (j-\mu_j)^2 \quad (3)$$

Энергия. Измеряет текстурную однородность.

$$\text{энергия} = \sqrt{\sum_{i,j} P_{ij}^2} \tag{4}$$

Энтропия. Измеряет беспорядок или сложность изображения.

$$\text{энтропия} = \sum_{i,j} P_{ij} * (-\ln P_{ij}) \tag{5}$$

Контраст. Определяет локальные изменения интенсивности.

$$\text{контраст} = \sum_{i,j} P_{ij} * (i-j)^2 \tag{6}$$

Однородность. Измеряет близость распределения элементов GLCM к диагонали.

$$\text{однородность} = \sum_{i,j} \frac{P_{ij}}{1+(i-j)^2} \tag{7}$$

3.2 Описание классификаторов

В работе рассматривается несколько алгоритмов классификации. В данном разделе кратко опишем каждую модель классификатора, переходя от самой простой, к наиболее сложной.

3.2.1 Дерево решений

Дерево решений (DTC) – модель прогнозирования. На основе входных данных дерево решений прогнозирует значение целевой переменной путем изучения простых правил принятия решений if-then-else. Такие деревья очень схожи с бинарными деревьями поиска.

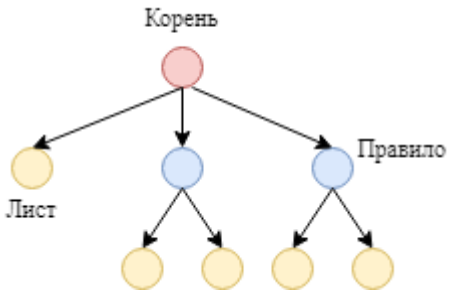

Рисунок 1. Структура дерева решений

В корневом узле располагается множество входных данных, в листовых узлах – значение целевой функции, в остальных узлах – правило перехода, определяющее по какому из ребер идти. Правило перехода, или правило разбиения данных на подмножества на

текущем узле, определятся из условия минимизации среднего значения энтропии внутри этих подмножеств.

3.2.2 Случайный лес

В основе метода *случайного леса (RFC)* лежит использование *ансамбля* деревьев решений. На вход каждому дереву поступает некоторое подмножество тренировочных данных, причем эти подмножества для разных деревьев не пересекаются. После чего, на полученной выборке, происходит построение дерева решений. Результатом прогноза ансамбля деревьев решений будет класс, который набрал наибольшее количество «голосов».

3.2.3 Мультиклассовый метод опорных векторов

Основная цель *метода опорных векторов (SVM)* состоит в том, чтобы найти разделяющую гиперплоскость в N-мерном пространстве признаков, которая четко классифицирует эти признаки.

В классической версии SVM – бинарный классификатор. Чтобы разделить два класса точек данных, может быть выбрано множество гиперплоскостей. Искомой *разделяющей гиперплоскостью* будет гиперплоскость, максимизирующая расстояние до точек данных обоих классов. Чем больше это расстояние, тем меньше будет средняя ошибка классификатора.

Так как в данной работе необходимо классифицировать на несколько классов, то будет использоваться модифицированная версия SVM – *мультиклассовый метод опорных векторов (MSVM).*

3.2.4 Метод k-ближайших соседей

Метод K-ближайших соседей (KNN) — метрический алгоритм для автоматической классификации объектов. Объект присваивается тому классу, который является наиболее распространённым среди k-соседей данного элемента, классы которых уже известны.

Для классификации каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

1. Вычислить расстояние до каждого из объектов обучающей выборки;
2. Отобрать k объектов обучающей выборки, расстояние до которых минимально;
3. Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей.

3.2.5 Одноуровневый персептрон

Одноуровневый персептрон (1 Layer MLP) представляет собой нейронную сеть прямого распространения. Структура 1 Layer MLP:

1. Входной слой. Во входной слой поступают исходные данные, и передаются дальнейшим слоям.
2. Один скрытый слой. В нем происходит взвешенное суммирование выходных сигналов предыдущего слоя и формирование выхода посредством нелинейной функции активации.
3. Выходной слой. Вывод результата. В данной работе выходом будет распределения вероятностей принадлежности к определённому классу.

Сигналы между нейронами разных слоев сети передаются через соединения *синапсов*. У синапсов есть 1 параметр — *вес*. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому.

3.3 Метрики качества алгоритмов классификации

Для оценки качества работы классификатора на тестовой выборке, используются различные численные оценки. В простейшем случае такой метрикой может быть доля верных предсказаний (*Accuracy*):

$$Accuracy = \frac{N_{correct}}{N_{total}} \tag{8}$$

где $N_{correct}$ - число верных предсказаний алгоритма, N_{total} - число объектов в тестовой выборке. Однако стоит отметить, что такая метрика присваивает всем ответам одинаковый вес, вследствие чего ее можно применять только в случае сбалансированных наборов данных.

Для определения качества работы классификатора на несбалансированных данных необходимо использовать точность (*Precision*) и полноту (*Recall*) работы алгоритма:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

где TP - число верных предсказаний, FP - число ложноположительных предсказаний, FN - число ложноотрицательных. В качестве одной из метрик также можно использовать *F-score*, являющийся средним гармоническим между точностью и полнотой:

$$F\text{-}score = \frac{2Precision * Recall}{Precision + Recall} \tag{11}$$

4. Исследовательская часть

4.1 Описание базы данных

База данных изображений больных и здоровых листьев томатов была взята из открытого источника Plant Village [5]. Она включает в себя 5994 предварительно отсегментированных изображений, которые подразделяются на 6 классов, описанных в таблице 1.

Таблица 1. Описание классов исследуемой базы данных

Класс	Тип заболевания	Кол-во изображений
Healthy – здоровые листья	-	1000
Bacterial spot – бактериальная пятнистость	Бактериальное	1000
Early blight – ранняя гниль	Грибковое	1000
Late blight – поздняя гниль	Грибковое	1000
Septoria leaf spot – септория	Грибковое	1000
Yellow Leaf Curl Virus – вирус желтого скручивания листьев	Вирусное	994

Описание изображений:

- Разрешение: 256x256;
- Глубина цвета: 8 бит;
- Формат хранения: .jpeg;
- Цветовой профиль: RGB;

4.2 Инструменты исследования

Разработка программного комплекса ведется на языке *Python*. Такой выбор языка программирования обусловлен наличием готовых библиотек для работы с многомерными массивами (тензорами). В частности, в данной работе используются библиотеки *numpy* и *pytorch,* который позволяет обернуть процесс извлечения признаков в единую модель.

Для вычисления GLCM и ее признаков, описанных в пункте 3.1 «Матрица GLCM», используются функции библиотеки *scikit-image.*

Обучение и тестирование классификаторов происходит с помощью инструментов библиотеки *scikit-learn.*

4.3 Ход работы

Работая с RGB изображениями, мы фактически уже работаем с мультиспектральными изображениями. В этом случае интересно определить можно ли мы ограничиться одним, достаточно информативным каналом.

Признаки болезней проявляются в основном в наличии изменения (потемнении) цвета листа. Это выражается более явно в красном канале. Поэтому возникает предположение, что признаков, извлеченных из красного канала изображения, окажется достаточно, чтобы с успехом диагностировать заболевание.

Еще одним подкреплением такого предположения служит исследование [10], в котором установлено, что на основе показателя NDVI рассчитанного в красном и инфракрасном диапазоне, можно судить о состоянии здоровья растения. Поэтому, есть основания полагать, что красный канал может быть довольно информативным.

4.3.1 Исследование глобальных статистических признаков

В качестве статистических признаков используются: математическое ожидание – *mean*, среднеквадратичное отклонение - *std*, минимальное - *min* и максимальное - *max* значение выборки, выраженные в единицах стандартного отклонения. Под *глобальными* понимаются признаки, извлеченные неким оператором над всем изображением сразу.

Для каждого из глобальных статистических признаков построены гистограммы (рис. 1-4). В качестве первого приближения в решении задачи классификации изображений возможно воспользоваться информацией из данных гистограмм, для выделения самых простых для классификации случаев заболевания.

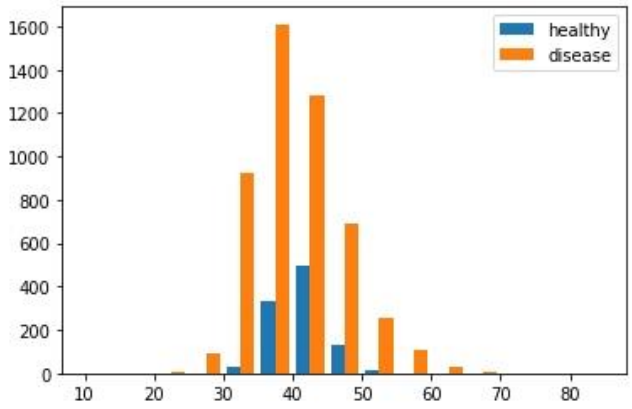

Рисунок 1. Гистограмма признака «mean»

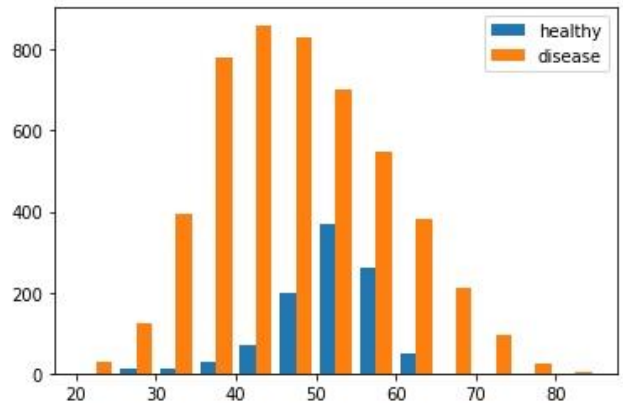

Рисунок 2. Гистограмма признака «std»

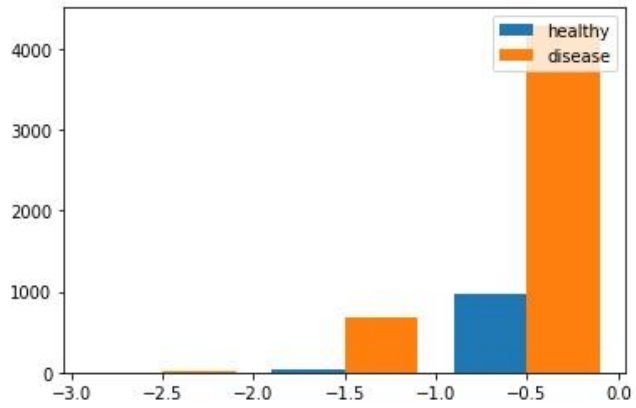

Рисунок 3. Гистограмма признака «min»

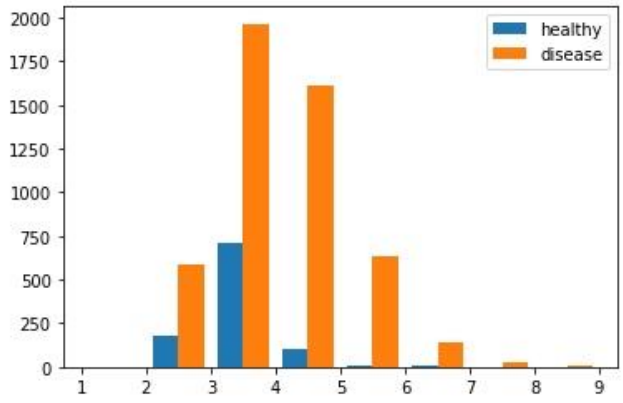

Рисунок 4. Гистограмма признака «max»

Из полученных гистограмм видно, что распределения этих признаков пересекаются, но несмотря на это, можно определить «пороги» бинарной классификации для некоторой доли изображений. *Лучшим* признаком будем называть тот признак, который отсекает как можно больше изображений больных листьев от здоровых, с минимальной долей ошибки.

Очевидно, что наилучшим признаком является «max». Если допустить небольшую долю ошибки, и установить порог по признаку «max» в значение 5, то мы получим частичное решение задачи детектирование болезни. То есть предполагается, что лист, у которого значение признака «max» больше 5, является больным.

Итеративно применяя такой подход для каждого лучшего признака из оставшихся, мы будем увеличивать полноту, одновременно уменьшая точность алгоритма.

Наилучшие результаты по детектированию заболевания, полученные вышеописанным «пороговым» подходом:

$$Precision = 95.64\ \%$$
$$Recall = \ 63.79\ \% \qquad (12)$$
$$F\text{-}score = 76.54\ \%$$

Данных признаков недостаточно для детектирования болезни, поэтому и задачу классификации они решить не способны.

Исследование глобальных статистических признаков показало, что они являются недостаточно информативными в задаче детектирования болезни. Поэтому предполагается, что использование локальных статистических признаков должно оказаться более полезным.

4.3.2 Метод квантования изображений

Извлечение текстурных признаков для изображений, имеющих 256 градаций серого, требует большой вычислительной работы. Как правило точность признаков, полученных таким способом, является избыточной. Поэтому, для уменьшения вычислительной нагрузки в

данной работе было решено сжать изображения путем их квантования на 5 уровней серого. Предполагается, что такой точности признаков хватит для классификации.

Для квантования изображений использовались фиксированные уровни дискретизации, полученные из распределения пикселей с ненулевой интенсивностью среди всех изображений в красном канале:

$$\begin{aligned} meanR^* &= 85.38, \\ stdR^* &= 53.79, \end{aligned} \tag{13}$$

где *mean**, *std** - среднее и стандартное отклонение яркости пикселей по всей выборке. В качестве адаптивного алгоритма квантования применялись следующие границы:

$$Q = \begin{cases} [0] - \text{0-й уровень} \\ [1, meanR^* - stdR^*) - \text{ 1-й уровень} \\ [meanR^* - stdR^*, meanR^*) - \text{ 2-й уровень,} \\ [meanR^*, meanR^* + stdR^*) - \text{ 3-й уровень} \\ [meanR^* + stdR^*, 255] - \text{ 4-й уровень} \end{cases} \tag{14}$$

Для нулевого уровня яркости выделен отдельный уровень квантования. Это сделано для того, чтобы не учитывать вклад фоновых (нулевых) пикселей в гистограмму, подаваемую на вход классификатору, поскольку они в основном не несут полезной информации.

4.3.3 Исследование глобальных текстурных признаков

В начале была исследована идея о том, что разделить растения на здоровые и больные можно с помощью GLCM-матрицы всего изображения. В качестве решающего признака было предложено использовать косинус угла между GLCM матрицей классифицируемого изображения и средней GLCM-матрицей всех здоровых растений.

В данной работе строятся симметричные, нормированные матрицы GLCM, со следующими параметрами d и φ:

$$\begin{aligned} d &= \{1px, 2px, 4px\} \\ \varphi &= \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\} \end{aligned} \tag{15}$$

Всё изображение, согласно (14), квантовалось на 5 уровней. Следовательно матрица GLCM для каждого направления и расстояния на изображении имела 25 элементов. Таким образом, каждое изображение характеризовалось признаком *glcm_feature* из 12 векторов длиной 25:

$$glcm\_feature(img) = \{GLCM_i,\ \ i = \overline{0,11},\ \dim(GLCM_i) = 25\} \tag{16}$$

С целью проверки идеи о том, что больные растения можно отделить от здоровых с помощью косинуса угла между векторами их текстурных признаков, для каждого класса был

вычислен средний вектор *feature_mean* и на одном графике были построены все значения косинусов между средними векторами всех классов (рис. 5).

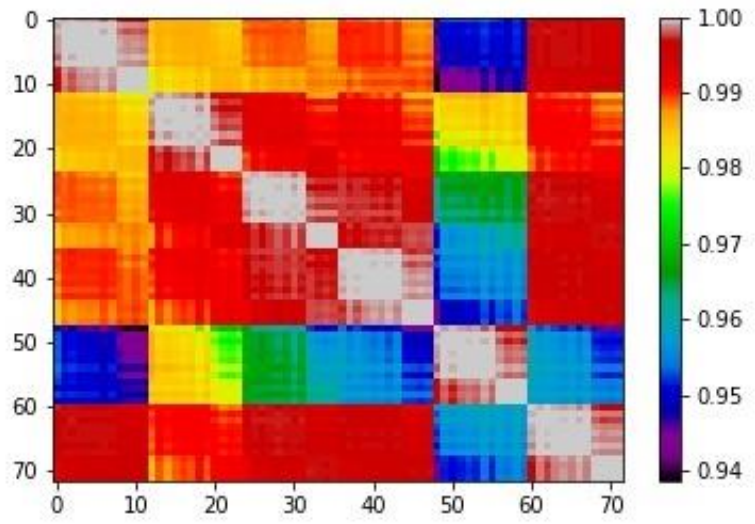

Рисунок 5. Значение косинусов между средними векторами всех классов

На данном рисунке достаточно чётко выделяются квадраты, полученные скалярными произведениями средних признаков классов между собой. Причём первая «строка» и первый «столбец» соответствуют взаимным скалярным произведениям между средними признаками здоровых и больных растений.

Из анализа этого рисунка первоначально было сделано предположение, что скалярные произведения между нормированными признаками исследуемого изображения и средними нормированными признаками здорового изображения могут быть критерием классификации на больное или здоровое растение. В рамках реализации данной идеи было построено решающее дерево, принимающее на вход 12-мерные признаки данных скалярных произведений.

Метрики качества работы построенного алгоритма по обнаружению болезни оказались следующими:

$$Precision = 84\ \%$$
$$Recall = \ 99\ \% \qquad (17)$$
$$F\text{-}score = 91\ \%$$

На первый взгляд, цифры высокие, однако исследуемая выборка является несбалансированной, и в ней вероятность встретить больное растение как раз составляет 84%. Таким образом можно сказать, что почти всегда дерево решений давало предсказание, что растение больное. Поэтому можно утверждать, что, основываясь только на признаках, извлечённых из глобальной GLCM-матрицы изображения, корректный алгоритм классификации построить в рамках модели решающего дерева невозможно.

4.3.4 Описание вектора локальных признаков

Как показали предыдущие исследования, на основе признаков, которые вычисляются глобально по всему изображению, разделение листьев растений на классы оказывается довольно слабым. Ввиду этого было решено вычислять признаки локально, в масштабах малой, по сравнению с размерами изображения, маски инструмента. Только потом информация о локальных признаках собиралась в масштабе всего изображения.

Размер маски был взят равным 17x17. Для анализа участков изображений под маской инструмента выбран следующий вектор признаков:

$$features = [STAT, HIST, GLCM], \tag{18}$$

составленный из:

1. Локальных статистических характеристик изображения в R-канале:

$$STAT = \left[\frac{meanR}{meanR^*}, \frac{stdR}{stdR^*}, \frac{maxR - meanR}{stdR^*}, \frac{meanR - minR}{stdR^*}\right], \tag{19}$$

   где *meanR, maxR, minR* и *stdR* - среднее, максимальное, минимальное значения и стандартное отклонение гистограммы изображения в красном канале, *meanR** и *stdR** обозначены в равенстве (13);

2. Нормированной гистограммы квантованного изображения без учёта нулевых элементов:

$$HIST = [N(Q_i)], \tag{20}$$

   где $N(Q_i)$ - нормированное значения уровня Q_i на рассматриваемом участке изображения, $i = \overline{1,4}$;

3. Признаков, полученных из GLCM матрицы квантованного изображения, для различных расстояний и углов:

$$GLCM = [contrast, homogeneity, energy, corrrelation, entropy] \tag{21}$$

   GLCM-матрица для каждого угла и расстояния квантованного изображения имеет размер 5x5. Всего в работе рассмотрено 3 расстояния и 4 направления (15). Поэтому каждый из 5 текстурных признаков представлен 12 числами.

На рисунках 6 и 7 изображены примеры исходного изображения здорового и больного Bacterial spot листьев томатов, а рядом с ними результаты извлечения 68 признаков *features* под маской инструмента. Данные изображения возможно получить в п. 3.3 (см. таблицу 2) выполнения модели извлечения признаков из оригинальных изображений.

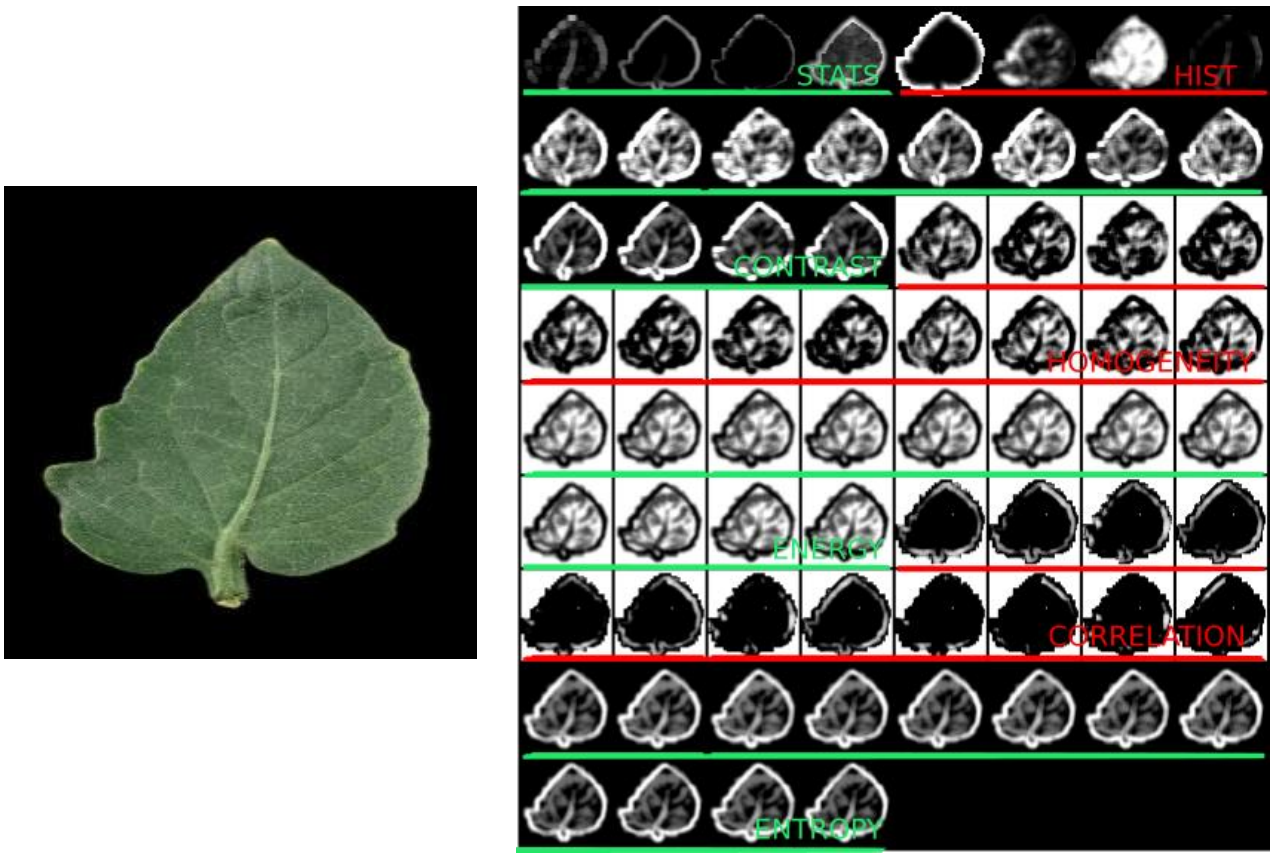

Рисунок 6. Исходное изображение и вектор изображений *features* для здорового листа

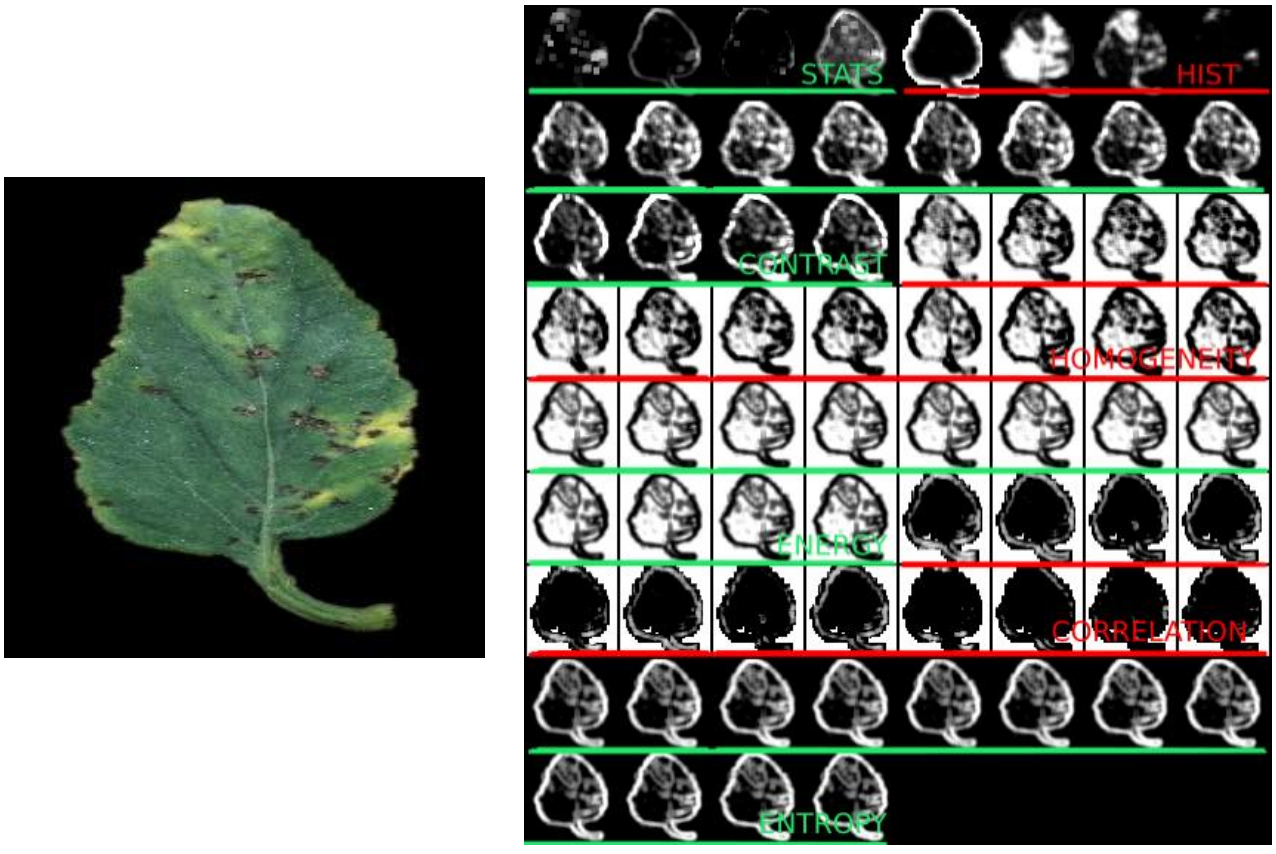

Рисунок 7. Исходное изображение и вектор изображений *features*
для больного Bacterial spot листа

4.3.5 Алгоритм извлечения вектора локальных признаков из изображения

В данном разделе пошагово описан алгоритм извлечения локальных признаков *features* для конкретного изображения.

Таблица 2. Алгоритм извлечения признаков

№ Шага	Действие	Размерность выхода
1	Подготовка данных и вычисление статистических характеристик *STAT* (15)	
1.1	Считывание RGB-изображения.	[3,256,256]
1.2	Извлечение информации из канала R.	[1,256,256]
1.3	Паддинг изображения для подготовки к операции скользящего окна.	[1,257,257]
1.4	Построение массива всех скользящих окон 17x17 с шагом *stride* = 4 для всего изображения.	[1, 61, 61, 17, 17]
1.5	Вычисление характеристик *STAT* (15) в каждом окне.	[4, 61, 61]
1.6	Квантование всего изображения на 4 уровня (13). Получение квантованного изображения *Q*.	[1, 61, 61, 17, 17]
1.7	Изменение размера *Q*.	[3721, 17, 17]
2	Вычисление характеристик *HIST* (16) и *GLCM* (17) в каждом из 3721 окон	
2.1	Вычисление нормированной гистограммы *HIST* (16) для *Q*.	[4]
2.2	Вычисление нормированной, симметричной *GLCM*-матрицы.	[5, 5, 3, 4]
2.3	Вычисление признаков *GLCM*.	[5,3,4]
2.4	Уплощение матрицы *GLCM*-признаков.	[60]
2.5	Конкатенация гистограммы *HIST* с *шага 2.1* и признаков *GLCM* с *шага 2.4*.	[64]
3	Формирование результирующего вектора признаков по данному изображению:	
3.1	Окончание операций под масками инструмента.	[3721,64]
3.2	Приведение формы матрицы признаков, полученной на *шаге 12*, к размеру с *шага 5* и её транспонирование.	[64,61,61]
3.3	Конкатенация *STAT*, *HIST*, *GLCM*.	[68,61,61]
3.4	Усреднение локальных признаков по всем скользящим окнам.	[68]

Блок операций 2 выполняется по всему изображению параллельно. Модель извлечения признаков реализована средствами библиотеки *pytorch*.

5. Результаты исследования

Одной из задач работы являлась проверка возможностей различных алгоритмов классификации на одном и том же наборе данных с целью выбрать наилучший из них. Поэтому далее описаны результаты эксперимента по применению различных алгоритмов классификации и методов обработки полученных данных к разным наборам признаков.

Для изучения были выбраны следующие алгоритмы: дерево решений, случайный лес, метод опорных векторов, метод k-ближайших соседей и одноуровневый персептрон. К тестовой выборке применялись алгоритмы с параметрами, дающими наилучшие результаты при кросс-валидации на тренировочной выборке. Параметры выбранных алгоритмов представлены в таблице 1.

Таблица 1. Параметры алгоритмов классификации

Название метода	Параметры для scikit-learn
DTC	random_state=0, splitter='best', criterion='entropy', max_depth=10
MSVM	gamma='auto', kernel='linear', C=5, shrinking=True
RF	max_depth=10, random_state=0, criterion='entropy', n_jobs=8, n_estimators=250
KNN	n_neighbors=11, n_jobs=8, metric='euclidean'
1 Layer MLP	activation='relu', batch_size='auto', hidden_layer_sizes=(200), learning_rate='constant', max_iter=10000, nesterovs_momentum=True, shuffle=True, solver='adam'

Эти алгоритмы применялись как к всем извлечённым признакам *features* (18), описанным в разделе 4.3.4 «Описание вектора локальных признаков», так и к их частям: *STAT* (19), *HIST* (20) и *GLCM* (21). Также эти признаки брались как нормализованными на собственное стандартное отклонение, так и без процедуры стандартизации. В приложениях A-C представлены таблицы с результатами мультиклассовой и бинарной классификации, в каждой из которых выделены 5 лучших и 5 худших алгоритмов. Лучший набор признаков и методов предобработки данных для каждого алгоритма выделен соответствующим цветом.

После выполнения тестирования различных классификаторов можно сделать несколько выводов. Дерево решений в результате мультиклассовых тестов показало один из худших результатов по сравнению с другими классификаторами. Хороший результат даёт лишь ансамбль деревьев решений – случайный лес. Стабильно лучшие результаты показывали одноуровневый персептрон и метод опорных векторов над полным вектором признаков. Однако стоит заметить, что метод опорных векторов демонстрирует очень низкое качество в случае работы лишь с вектором *STAT*. Из результатов тестов видно, что алгоритмы классификации на основе статистических характеристик гистограммы, таких как её минимальное, максимальное, среднее значения и стандартное отклонение, показывают низкое

качество. Текстурные характеристики вносят заметный вклад в улучшение работы алгоритмов.

Довольно интересной является разница в метриках качества всех рассмотренных алгоритмов при работе с глобальными и локальными признаками. Усреднение локальных признаков *features* улучшает качество работы многоклассового классификатора на 3-4% для почти всех классификаторов, кроме SVM. Классификация на глобальных признаках *GLCM* приобретает до 4% качества относительно классификации на глобальных признаках *GLCM*. Правда при этом она всё ещё остаётся хуже классификации на полном наборе глобальных признаков *features*. Существенный провал до 15% испытывает классификация на глобальных признаках *STAT, STAT+HIST*.

Для бинарной классификации подобной разницы в качестве работы алгоритмов практически нет.

Процедура нормализации стандартного отклонения признаков на единицу в среднем позволяет выиграть несколько процентов качества для методов: SVM, KNN, MLP. Деревья решений на исследованном наборе данных оказались не чувствительны к данной процедуре.

6. Заключение

В процессе выполнения работы был освоен теоретический материал, необходимый как база для дальнейших исследований. В частности, изучен статистический метод исследования с помощью матрицы GLCM. Получены базовые знания в области машинного обучения. Данные темы кратко описаны в главе 3. «Теоретическая часть». Помимо теории, освоен с нуля язык программирования Python. Улучшены навыки работы с многомерными данными и их анализа.

Для обучения и тестирования классификаторов была найдена и подготовлена база данных, содержащая 5994 изображений листьев томатов, включающая в себя 6 классов больных и здоровых растений (см. п. 4 «Описание базы данных»).

Одним из наиболее трудоёмких моментов в работе, с точки зрения реализации, был процесс извлечения признаков из изображений. Всего было получено 2 типа признаков: статистические и текстурные. При этом исследовалось влияние на результат способа получения этих признаков: глобального и локального. Как показали результаты экспериментов (см. прил. А, В), глобальные признаки проигрывают в качестве в среднем на 3 – 4% при классификации с помощью features. Это происходит в основном за счёт потерь на признаках STAT, HIST. Однако глобальные GLCM признаки в рамках рассмотренных моделей наоборот, показывают себя лучше усреднённых локальных. Таким образом, в будущей работе можно будет применять комбинированные локальные и глобальные признаки.

В процессе исследования было доказано, что и в задаче детектирования болезни глобальных статистических или текстурных признаков изображений в смысле угловой близости недостаточно. Доказательства этих утверждений находятся в разделах 4.3.1 «Исследование глобальных статистических признаков» и 4.3.3 «Исследование глобальных текстурных признаков».

Так же в ходе выполнения работы было предположено, что для диагностирования болезни достаточного использовать только красный канал изображений. Высокие результаты в приложении А, доказывают справедливость этого предположения.

В итоге лучшим классификатором оказалась простейшая нейронная сеть – одноуровневый персептрон, показатель F-score которого составляет 84%. При этом для достижения такой точности ему понадобился весь вектор признаков (18). Отмечено, что значения F-score равного 77 %, можно добиться используя лишь текстурные признаки.

Вторым по качеству алгоритмом машинного обучения оказался метод опорных векторов, который дает 76% F-score, при этом используя весь вектор признаков (18).

Было выяснено (прил. А-С), что стандартизация данных является важной для алгоритмов SVM, KNN, MLP, в то время как DTC и RF могут обучаться и без этой процедуры.

Таким образом, все поставленные задачи выполнены. Установлен лучший классификатор болезней листьев томатов – одноуровневый персептрон и определен набор лучших признаков, описанный в разделе 4.3.4 «Описание вектора локальных признаков». Цель исследования достигнута.

7. Список литературы

1. M.A.F. Azlah, L.S. Chua, F.R. Rahmad, F.I. Abdullah, S.R Wan Alwi, “Review on Techniques for Plant Leaf Classification and Recognition”, Computers 2019, *8*, 77. 22 pp, doi: 10.3390/computers8040077
2. 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), ISSN: 2278-3075, Volume-8 Issue-6S2, April 2019, pp 108-120.
3. S. Chakrabortya, A.C. Newton, “Climate change, plant diseases and food security: an overview”, Plant Pathology (2011), Vol. 60, Issue 1, pp 2–14, doi: 10.1111/j.1365-3059.2010.02411.x.
4. United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019: Highlights (ST/ESA/SER.A/423).
5. Dataset of images. Available: https://arxiv.org/abs/1511.08060
6. G. Mathur, H. Purohit, “Performance Analysis of Color Image Segmentation using K-Means Clustering Algorithm in Different Color Spaces”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Vol. 4, Issue 6, Ver. III (Nov - Dec. 2014), pp 1-4. Available: www.iosrjournals.org.
7. G. Jeon, “Color Image Enhancement by Histogram Equalization in Heterogeneous Color Space", International Journal of Multimedia and Ubiquitous Engineering (IJMUE), Vol. 9, No. 7 (2014), pp 309-318, doi: http://dx.doi.org/10.14257/ijmue.2014.9.7.26
8. R. M. Haralick, K. Shanmugam, and I. Dinstein, ‘‘Textural features for image classification,’’ IEEE Trans. Syst., Man, Cybern., vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
9. R. W. Conners and C. A. Harlow, ‘‘A theoretical comparison of texture algorithms’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-2, no. 3, pp. 204–222, May 1980.
10. J.Weier, D. Herring, “Measuring Vegetation”, NASA Earth Observatory, 2000-08-30, pp. 4.

8. Приложение

Приложение А. Показатели F-Score, полученные на локальных признаках

Name	Feature vector	Standartize data	Tomato Bacterial Spot	Tomato Early blight	Tomato Late blight	Tomato Septoria leaf spot	Tomato Yellow Leaf Curl Virus	Tomato healthy	Average F-Score
Decision Tree	FEATURE	Y	0,71	0,44	0,49	0,42	0,72	0,80	0,60
	FEATURE	N	0,70	0,43	0,50	0,41	0,72	0,79	0,59
	STAT	Y	0,57	0,41	0,43	0,40	0,67	0,58	**0,51**
	STAT	N	0,59	0,39	0,41	0,39	0,65	0,58	**0,50**
	STAT+HIST	Y	0,63	0,41	0,43	0,43	0,65	0,64	0,53
	STAT+HIST	N	0,62	0,40	0,44	0,46	0,66	0,64	0,54
	GLCM	Y	0,67	0,36	0,42	0,41	0,56	0,79	0,54
	GLCM	N	0,66	0,36	0,42	0,40	0,57	0,79	0,53
Support Vector Machine	FEATURE	Y	0,85	0,60	0,65	0,69	0,87	0,92	**0,76**
	FEATURE	N	0,75	0,52	0,55	0,58	0,78	0,86	0,67
	STAT	Y	0,58	0,34	0,43	0,31	0,62	0,28	**0,43**
	STAT	N	0,55	0,37	0,47	0,31	0,59	0,27	**0,43**
	STAT+HIST	Y	0,67	0,47	0,47	0,42	0,73	0,65	0,57
	STAT+HIST	N	0,65	0,44	0,47	0,18	0,62	0,64	**0,50**
	GLCM	Y	0,79	0,52	0,58	0,60	0,80	0,89	0,70
	GLCM	N	0,74	0,43	0,54	0,46	0,68	0,82	0,61
Random Forest	FEATURE	Y	0,79	0,53	0,57	0,57	0,81	0,88	0,69
	FEATURE	N	0,79	0,52	0,59	0,57	0,80	0,88	0,69
	STAT	Y	0,69	0,44	0,49	0,45	0,76	0,65	0,58
	STAT	N	0,68	0,44	0,49	0,46	0,76	0,65	0,58
	STAT+HIST	Y	0,71	0,50	0,55	0,53	0,76	0,73	0,63
	STAT+HIST	N	0,71	0,50	0,54	0,53	0,77	0,73	0,63
	GLCM	Y	0,77	0,46	0,56	0,51	0,70	0,85	0,64
	GLCM	N	0,77	0,45	0,56	0,52	0,70	0,85	0,64
K Nearest Neighbors	FEATURE	Y	0,73	0,53	0,53	0,51	0,79	0,86	0,66
	FEATURE	N	0,73	0,50	0,53	0,48	0,76	0,82	0,64
	STAT	Y	0,65	0,44	0,46	0,37	0,76	0,63	0,55
	STAT	N	0,61	0,43	0,42	0,35	0,71	0,64	0,53
	STAT+HIST	Y	0,71	0,45	0,54	0,46	0,74	0,73	0,60
	STAT+HIST	N	0,68	0,44	0,52	0,43	0,70	0,73	0,58
	GLCM	Y	0,71	0,46	0,50	0,42	0,72	0,84	0,61
	GLCM	N	0,69	0,43	0,51	0,43	0,67	0,79	0,59
1 Layer MLP	FEATURE	Y	0,89	0,73	0,77	0,76	0,95	0,94	**0,84**
	FEATURE	N	0,91	0,72	0,77	0,77	0,94	0,95	**0,84**
	STAT	Y	0,69	0,47	0,53	0,44	0,79	0,69	0,60
	STAT	N	0,70	0,48	0,51	0,43	0,79	0,69	0,60
	STAT+HIST	Y	0,79	0,59	0,66	0,60	0,88	0,81	0,72
	STAT+HIST	N	0,77	0,57	0,64	0,60	0,88	0,81	0,71
	GLCM	Y	0,88	0,62	0,66	0,71	0,84	0,92	**0,77**
	GLCM	N	0,86	0,55	0,64	0,63	0,81	0,91	**0,73**

Приложение В. Показатели F-Score, полученные на глобальных признаках

Name	Feature vector	Standartize data	Tomato Bacterial Spot	Tomato Early blight	Tomato Late blight	Tomato Septoria leaf spot	Tomato Yellow Leaf Curl Virus	Tomato healthy	Average F-Score
Decision Tree	FEATURE	Y	0,67	0,38	0,49	0,43	0,61	0,74	0,55
	FEATURE	N	0,68	0,37	0,52	0,45	0,66	0,75	0,57
	STAT	Y	0,59	0,28	0,33	0,16	0,40	0,53	**0,38**
	STAT	N	0,56	0,25	0,36	0,17	0,41	0,53	**0,38**
	STAT+HIST	Y	0,59	0,37	0,41	0,40	0,58	0,60	0,49
	STAT+HIST	N	0,59	0,37	0,41	0,41	0,56	0,62	0,49
	GLCM	Y	0,67	0,38	0,39	0,38	0,67	0,73	0,54
	GLCM	N	0,68	0,40	0,44	0,38	0,66	0,75	0,55
Support Vector Machine	FEATURE	Y	0,83	0,59	0,62	0,68	0,91	0,91	**0,76**
	FEATURE	N	0,74	0,53	0,51	0,58	0,74	0,86	0,66
	STAT	Y	0,59	0,32	0,38	0,21	0,32	0,38	**0,36**
	STAT	N	0,59	0,32	0,38	0,21	0,32	0,38	**0,37**
	STAT+HIST	Y	0,60	0,40	0,38	0,35	0,39	0,53	0,44
	STAT+HIST	N	0,62	0,39	0,39	0,33	0,39	0,52	0,44
	GLCM	Y	0,78	0,56	0,56	0,65	0,84	0,89	0,71
	GLCM	N	0,72	0,48	0,54	0,52	0,75	0,80	0,63
Random Forest	FEATURE	Y	0,75	0,49	0,58	0,53	0,76	0,85	0,66
	FEATURE	N	0,76	0,49	0,58	0,53	0,77	0,85	0,66
	STAT	Y	0,62	0,33	0,40	0,22	0,50	0,58	0,44
	STAT	N	0,63	0,33	0,40	0,21	0,50	0,58	0,44
	STAT+HIST	Y	0,70	0,43	0,49	0,47	0,65	0,68	0,57
	STAT+HIST	N	0,70	0,41	0,49	0,49	0,64	0,69	0,57
	GLCM	Y	0,74	0,48	0,56	0,49	0,75	0,84	0,65
	GLCM	N	0,75	0,48	0,58	0,50	0,75	0,85	0,65
K Nearest Neighbors	FEATURE	Y	0,72	0,48	0,52	0,48	0,76	0,82	0,63
	FEATURE	N	0,65	0,45	0,48	0,37	0,69	0,77	0,57
	STAT	Y	0,57	0,30	0,36	0,23	0,39	0,57	**0,40**
	STAT	N	0,59	0,28	0,39	0,23	0,44	0,57	0,42
	STAT+HIST	Y	0,62	0,41	0,46	0,42	0,62	0,67	0,53
	STAT+HIST	N	0,60	0,40	0,44	0,37	0,59	0,64	0,51
	GLCM	Y	0,70	0,44	0,52	0,44	0,74	0,82	0,61
	GLCM	N	0,67	0,46	0,51	0,38	0,68	0,77	0,58
1 Layer MLP	FEATURE	Y	0,87	0,67	0,73	0,74	0,92	0,95	**0,81**
	FEATURE	N	0,85	0,64	0,73	0,71	0,92	0,94	**0,80**
	STAT	Y	0,65	0,38	0,44	0,24	0,53	0,60	0,47
	STAT	N	0,65	0,37	0,45	0,19	0,53	0,59	0,46
	STAT+HIST	Y	0,71	0,49	0,52	0,50	0,69	0,74	0,61
	STAT+HIST	N	0,69	0,49	0,53	0,50	0,72	0,74	0,61
	GLCM	Y	0,86	0,60	0,70	0,72	0,90	0,93	**0,79**
	GLCM	N	0,84	0,60	0,67	0,68	0,91	0,93	**0,77**

Приложение C. Precision, Recall, F-Score для задачи обнаружения болезни на основе локальных и глобальных признаков

Name	Feature vector	Standartize data	Local			Global		
			Precision	Recall	F-Score	Precision	Recall	F-Score
Decision Tree	FEATURE	Y	0,96	0,96	0,96	0,96	0,96	0,96
	FEATURE	N	0,96	0,96	0,96	0,96	0,96	0,96
	STAT	Y	0,91	0,93	0,92	0,89	0,91	**0,90**
	STAT	N	0,90	0,92	**0,91**	0,90	0,93	**0,91**
	STAT+HIST	Y	0,90	0,95	0,92	0,91	0,94	0,93
	STAT+HIST	N	0,90	0,95	0,93	0,91	0,95	0,93
	GLCM	Y	0,95	0,96	0,96	0,96	0,95	0,95
	GLCM	N	0,95	0,97	0,96	0,96	0,95	0,96
Support Vector Machine	FEATURE	Y	0,97	0,98	**0,98**	0,99	0,98	**0,99**
	FEATURE	N	0,95	0,98	0,96	0,96	0,98	0,97
	STAT	Y	0,83	1,00	**0,91**	0,83	1,00	**0,91**
	STAT	N	0,83	1,00	**0,91**	0,83	1,00	**0,91**
	STAT+HIST	Y	0,83	1,00	**0,91**	0,86	0,99	0,92
	STAT+HIST	N	0,83	1,00	**0,91**	0,86	1,00	0,92
	GLCM	Y	0,97	0,98	**0,98**	0,97	0,98	0,97
	GLCM	N	0,95	0,98	0,96	0,94	0,97	0,95
Random Forest	FEATURE	Y	0,97	0,98	**0,98**	0,96	0,98	0,97
	FEATURE	N	0,97	0,99	**0,98**	0,96	0,99	0,97
	STAT	Y	0,89	0,97	0,93	0,88	0,95	0,92
	STAT	N	0,89	0,97	0,93	0,88	0,95	0,92
	STAT+HIST	Y	0,91	0,99	0,95	0,91	0,97	0,94
	STAT+HIST	N	0,91	0,98	0,95	0,91	0,97	0,94
	GLCM	Y	0,96	0,98	0,97	0,95	0,98	0,97
	GLCM	N	0,96	0,98	0,97	0,95	0,98	0,97
K Nearest Neighbors	FEATURE	Y	0,97	0,97	0,97	0,96	0,97	0,96
	FEATURE	N	0,96	0,97	0,96	0,95	0,97	0,96
	STAT	Y	0,91	0,95	0,93	0,89	0,94	**0,91**
	STAT	N	0,91	0,95	0,93	0,88	0,93	**0,91**
	STAT+HIST	Y	0,94	0,95	0,94	0,92	0,95	0,94
	STAT+HIST	N	0,94	0,96	0,95	0,91	0,94	0,93
	GLCM	Y	0,96	0,98	0,97	0,96	0,97	0,97
	GLCM	N	0,95	0,98	0,96	0,95	0,96	0,96
1 Layer MLP	FEATURE	Y	0,99	0,98	**0,99**	0,99	0,98	**0,98**
	FEATURE	N	0,99	0,99	**0,99**	0,98	0,99	**0,98**
	STAT	Y	0,92	0,96	0,94	0,90	0,94	0,92
	STAT	N	0,91	0,96	0,93	0,89	0,95	0,92
	STAT+HIST	Y	0,96	0,96	0,96	0,94	0,96	0,95
	STAT+HIST	N	0,96	0,96	0,96	0,93	0,96	0,94
	GLCM	Y	0,98	0,99	**0,98**	0,99	0,98	**0,99**
	GLCM	N	0,98	0,98	**0,98**	0,99	0,98	**0,99**