

Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

Кафедра «Прикладная математика»

ОТЧЕТ

по практике по получению профессиональных умений и опыта в профессиональной деятельности

Выполнил:

студент гр. 33631/4 Камалетдинова Ю.А.

Место выполнения:

Кафедра «Прикладная математика»

Руководитель:

старший разработчик алгоритмов Яковлев Д.В.

Оценка руководителя:

Санкт-Петербург

2019

Содержание

1 Введение	3
1.1 Мотивация	3
1.2 Задачи	3
2 Анализ набора данных	4
2.1 Первичный взгляд на данные	4
2.2 Загрузка изображений	5
2.2.1 Представление данных	5
2.2.2 Изменение разрешения	6
2.3 Сегментация	6
2.4 Удаление шума	7
3 Извлечение признаков	10
3.1 Цветовые признаки	11
3.2 Признаки формы	12
3.2.1 Количество ограничивающих контуров	12
3.2.2 Общая площадь	13
3.2.3 Максимальная площадь	13
3.2.4 Периметр	13
3.2.5 Мера прямоугольности	13
3.2.6 Мера округлости	13

4 Классификация	15
4.1 Метод опорных векторов	15
5 Результаты исследований	17
5.1 Метрика	17
5.2 Выводы	18
5.3 Заключение	19

1 Введение

1.1 Мотивация

Потребность в сельскохозяйственных продуктах увеличивается с каждым днем, как и растет население планеты Земля. Часть работ выполняют люди, и силы уходят на контроль качества выращиваемых культур. Мы сможем использовать временные и природные ресурсы более бережно и экономно, увеличим урожай, если научимся дифференцировать благородные культуры и отличать их от сорняков без помощи человека.

В такой ситуации естественным образом приходит мысль об автоматизации процессов, например, классификация саженцев по фотоснимкам. Возникает мысль задействовать нейросети, что обосновано преимуществами, но также они обладают недостатками в виде вычислительных затрат. Возможно использовать менее затратные алгоритмы, но они требуют более тонкой настройки для достижения сопоставимого результата.

1.2 Задачи

Пусть X — множество признаков образцов, Y — множество меток классов, $X^l = \{(x_i, y_i)\}_{i=1}^l$ — обучающая выборка, где $x_i \in \mathbb{R}^n$ — вектор признаков объекта, y_i — метка класса. Существует целевое отображение $y^* : X \rightarrow Y$ со значениями, известными на обучающей выборке. Поставим следующие задачи:

- Реализовать алгоритм классификации для любого вектора признаков $x \in X$ с помощью алгоритмов машинного обучения и компьютерного зрения
- Изучить методы предобработки данных и использовать их в решении задачи классификации

2 Анализ набора данных

2.1 Первичный взгляд на данные

Исследуемый набор данных был собран группой Орхусского университета по обработке сигналов в сотрудничестве в Университетом Южной Дании. Этапы создания коллекции описаны в статье [1]. Набор содержит приблизительно 960 уникальных изображений растений 12 видов, находящихся на разных стадиях роста.

Изучим исходные данные. Построим образцы каждого класса в виде сетки



Рис. 1: Образцы растений каждого вида (построчно)

Проанализировав снимки, можно сделать некоторые выводы:

- Исходные изображения уже кадрированы и не требуют дополнительной обрезки



Рис. 2: Примеры исходных изображений

- Разрешения изображений варьируются от 50x50px до 2000x2000px, поэтому необходимо привести весь набор к единому разрешению, иначе для каждого изображения значения его признаков будут находиться в разных пределах, что не позволит решить задачу классификации
- Данные не сбалансированы: от 221 до 654 размеченных образцов каждого класса
- Фон на снимках различен, необходимо выбрать способ сегментации, наиболее подходящий для большинства

2.2 Загрузка изображений

2.2.1 Представление данных

Библиотека OpenCV использует цветовую модель BGR (Blue Green Red) для представления цветных изображений. Каждый пиксель характеризуется составляющими синей, зеленой и красной компонентами.

2.2.2 Изменение разрешения

Приведем изображения к разрешению 200x200px. Воспользуемся алгоритмом билинейной интерполяции, реализованном в библиотеке компьютерного зрения OpenCV [2]: в случае уменьшения размера новый пиксель изображения представляет собой взвешенную сумму соседних пикселей исходного и наоборот в случае увеличения разрешения.

2.3 Сегментация

Заметим, что представленные растения окрашены в зеленый цвет. Поэтому мы можем создать маску, фильтрующую диапазон зеленых оттенков и игнорирующую остальные пиксели. Для реализации воспользуемся библиотекой OpenCV [2] и библиотекой для вычислений NumPy [3] языка программирования Python.

Рассмотрим цветовую модель HSV (Hue Saturation Value) 3. В формате BGR значение каждой компоненты зависит от количества света, попадающего на объект. HSV же позволяет разграничить информацию о цвете и яркости. Оттенок, насыщенность и интенсивность позволяют задать нижнюю и верхнюю границы оттенков некоторого цвета, в данном случае — зеленого.

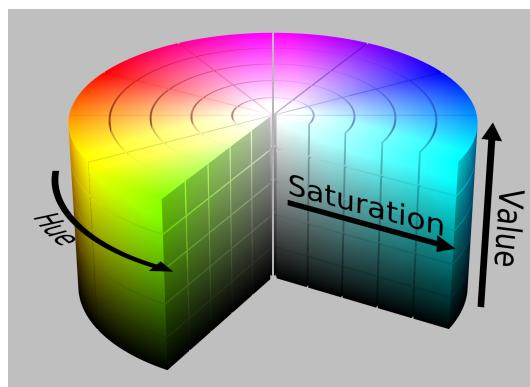


Рис. 3: Цветовая модель HSV

Пометим пиксели, находящиеся в зеленом диапазоне и получим цветовую маску.
Теперь применим операцию логического умножения к исходному изображению, присвоим
значениям пикселей фона значение черного цвета и получим сегментированное растение:

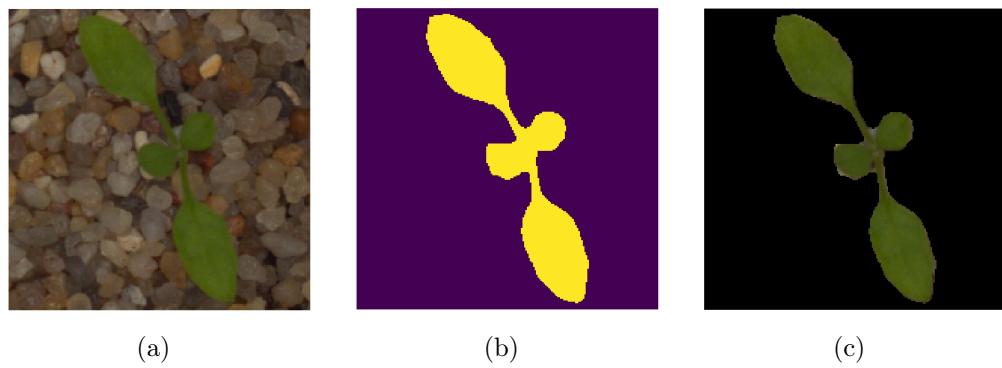


Рис. 4: (а) Исходное изображение; (б) Мaska; (с) Сегментированное изображение

2.4 Удаление шума

Сегментация не всегда происходит хорошо, как на примере 4(с). Небольшие участки
фона могут попадать в диапазон зеленых значений, что вызывает искажение бинарной
маски, и получается эффект, представленный на рисунке ниже



Рис. 5: Искажения при сегментации

Такие недостатки можно устраниТЬ при помощи морфологических операций — нелинейных преобразованиях, связанных с формой и структурой некоторого объекта, в данном случае, изображения. При обработке изображений морфология используется для исследования взаимодействия изображения с определенным структурирующим элементом — ядром — с помощью морфологических операций. Ядро итерируется по всему изображению и сравнивается с окрестностью пикселей, как описано в источнике [4].

Для улучшения сегментации применим операцию морфологического закрытия — комбинацию операций дилатации и эрозии.

Эрозия бинарного изображения f ядром s (обозначается $f \ominus s$) производит новое бинарное изображение $g = f \ominus s$ с единицами на некоторых позициях (x, y) ядра, где оно полностью совпадает с исходным изображением f , то есть $g(x, y) = 1$, если s поэлементно совпадает с участком f и 0 в другом случае, для всех координат пикселей (x, y)

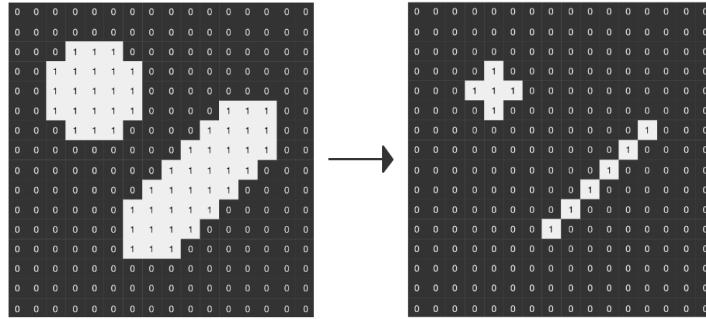


Рис. 6: Эрозия с квадратным ядром 3x3

Дилатация бинарного изображения f ядром s (обозначается $f \oplus s$) производит новое бинарное изображение $g = f \oplus s$ с единицами на некоторых позициях (x, y) ядра, где оно совпадает с исходным изображением f хотя бы в одной позиции, то есть $g(x, y) = 1$, если s совпадает хотя бы в одной позиции с участком f и 0 в другом случае, для всех

координат пикселей (x, y)

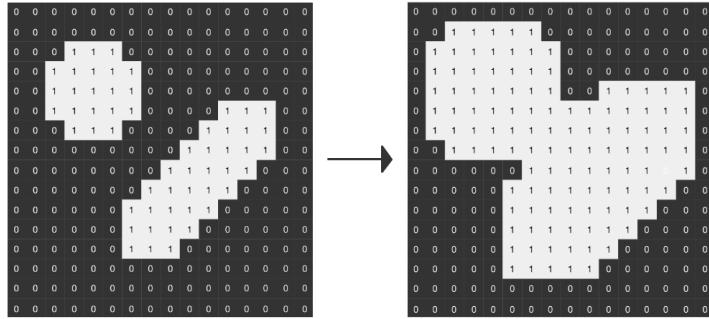


Рис. 7: Дилатация с квадратным ядром 3х3

Теперь можем определить операцию закрытия изображения f ядром s как $f \bullet s = (f \oplus s) \ominus s$. Структурный элемент может быть любой формы, и его выбор зависит от формы недостатков, которые требуется устраниить.

Применим операцию закрытия к изображению 5, выбрав эллиптическое ядро размером 6х6px и удалим оставшиеся объекты площадью менее 160px



Рис. 8: Изображение после закрытия маски

Растение на изображении 8 не имеет полостей, а фон очищен от не относящихся к

растению элементов. Но морфологическое закрытие не всегда однозначно хорошо действует на изображения. Рассмотрим результат работы над изображением класса Метлица обыкновенная (англ. *Loose silky-bent*):

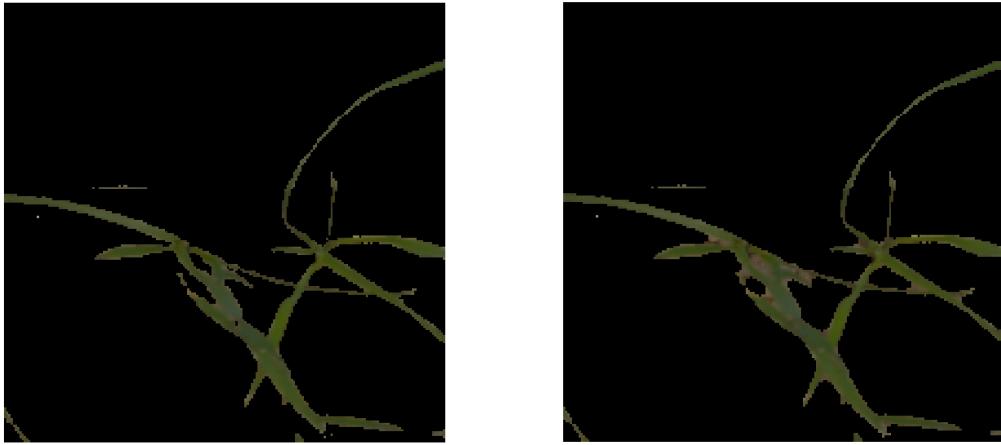


Рис. 9: Пример с ухудшением сегментации

На 9(b) заметно, что полости, соответствующие фону, были восстановлены, что не соответствует желаемому результату. Поскольку характеристики ядра и количество итераций определяются однажды и их нельзя изменять адаптивно, то мы не будем применять операцию морфологического закрытия. Ограничимся удалением объектов, чьи контуры ограничивают малую площадь.

3 Извлечение признаков

Признак в задаче классификации изображений — это информация, позволяющая решить, к какому заданному классу относится объект. Предположим, что каждый пиксель фотографии — это его признак. Тогда каждая фотография разрешением 200x200px будет

иметь 40000 признаков, к тому же, исходный набор данных — это более 4000 образцов. Решение задачи не только требует больших вычислительных затрат, но и влечет неспособность алгоритма к обобщению — переобучение. Более того, часть пикселей вообще не характеризует признаки растения и является фоном.

Для решения этой проблемы необходимо выбрать признаки с такими свойствами:

- Небольшая размерность пространства признаков
- Признаки не должны сильно коррелировать между собой
- Набор признаков позволяет классифицировать объект

3.1 Цветовые признаки

Чтобы установить сходство, рассчитаем цветовые моменты, характеризующие распределение цвета на изображении. Пусть $\{x^{(k)}\}_{i=1}^N$, где $k = 1 (= R), 2 (= G), 3 (= B)$ — номер канала цветового пространства RGB, N — число пикселей изображения, $x^{(k)_i}$ — i -ый пиксель k -го канала. Определим характеристики

$$\overline{x^{(k)}} = \frac{1}{N} \sum_{i=1}^N x_i^{(k)} \text{ — выборочное среднее} \quad (1)$$

$$s^{(k)} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i^{(k)} - \overline{x^{(k)}})^2} \text{ — выборочное стандартное отклонение} \quad (2)$$

Разделим изображение поканально, как на рисунке 10, и рассчитаем (1), (2)

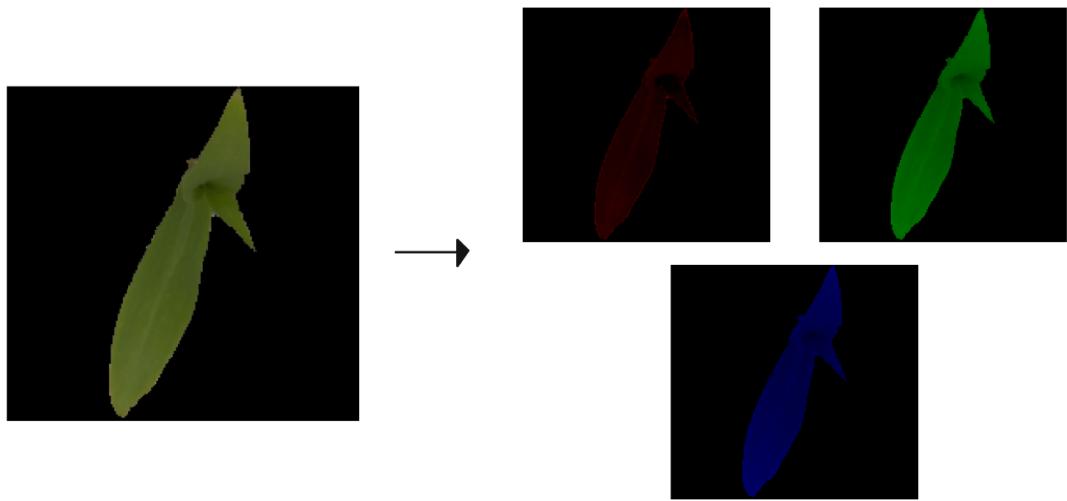


Рис. 10: Разложение по каналам R, G, B

3.2 Признаки формы

Не только цвет является важным признаком при классификации. Больше информации можно узнать, выделив признаки формы.

3.2.1 Количество ограничивающих контуров

В результате сегментации некоторые классы образцов разделяются на несколько объектов. Количество таких объектов — один из признаков класса. Для извлечения контуров воспользуемся алгоритмом трассировки границ, описанном в статье [6] и реализованном в библиотеке OpenCV [2] для языка программирования Python. Конуры и все использующие их далее характеристики не учитываются, если ограничиваемая площадь менее 160px^2 (установлено эмпирически для изображения $200\times 200\text{px}$).

3.2.2 Общая площадь

Величина составляет сумму всех площадей, ограниченных контурами. Площади вычисляются по формуле Грина по замкнутому контуру, алгоритм реализован в библиотеке OpenCV [2].

3.2.3 Максимальная площадь

Представляет собой максимальную из площадей, ограниченных контуром.

3.2.4 Периметр

Вычисляется для контура максимальной длины среди найденных.

3.2.5 Мера прямоугольности

Для вычисления меры прямоугольности требуется построить наименьший ограничивающий прямоугольник — множество точек двумерного пространства с наименьшей площадью, включающего в себя все точки объекта-растения. Тогда рассчитаем характеристику:

$$A = \frac{d_{min}}{d_{max}} \text{ — мера прямоугольности} \quad (3)$$

, где d_{min} и d_{max} — меньшая и большая стороны прямоугольника соответственно.

3.2.6 Мера округлости

Характеристика показывает, насколько большую площадь ограничивает периметр объекта. Мера округлости круга достигает максимального значения и равна единице.

$$f_{circ} = \frac{4\pi A}{P^2} — мера округлости, \quad (4)$$

где P — периметр контура, ограничивающего объект, A — площадь объекта

Построим корреляционную матрицу описанных признаков и изобразим ее в виде тепловой карты:

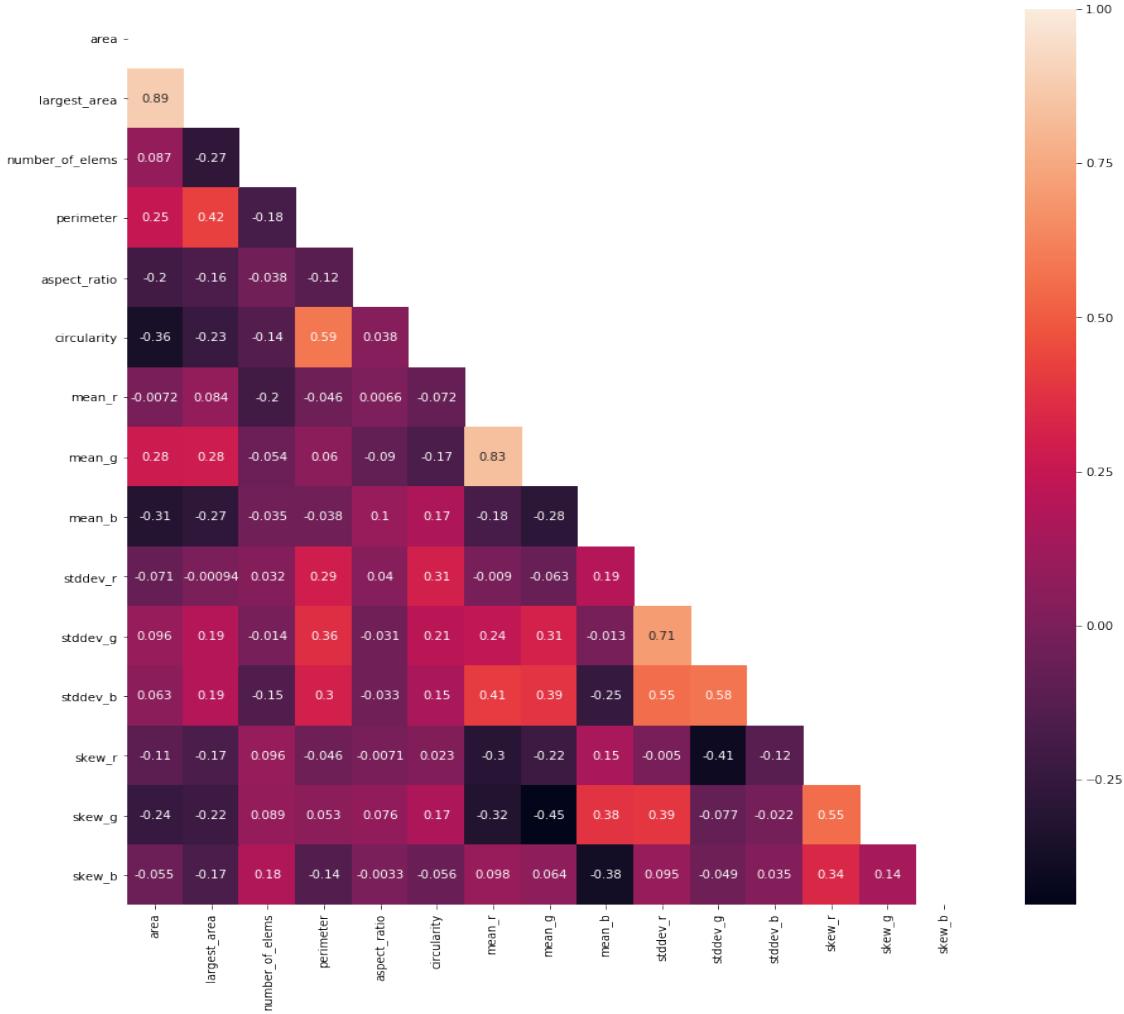


Рис. 11: Матрица корреляций признаков

На основе данных на рисунке 11 заключаем, что наиболее линейно зависимы призна-

ки общей площади (area) и максимальная площадь (largest area), но это справедливо не для всех классов в виду преобладания растений, ограниченных одним контуром, поэтому из рассмотрения признак максимальной площади не исключается.

4 Классификация

4.1 Метод опорных векторов

Метод опорных векторов (*англ. Support vector machine, SVM*) — алгоритм бинарной классификации, основанный на построении разделяющей гиперплоскости с зазором. Алгоритм метода описан в библиотеке Scikit-learn [7] для языка программирования Python.

Пусть поставлена задача классификации из пункта 1.2. Так как в общем случае гарантировать линейную разделимость выборки невозможно, сформулируем задачу минимизации с дополнительным параметром (*мягким зазором*):

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (5)$$

при условиях

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad (6)$$

где $\phi(x_i)$ — отображение пространства признаков в пространство большей размерности, ξ_i — поправка, ослабляющая границы классов. Коэффициент $C > 0$ определяет, насколько велик зазор разделяющей гиперплоскости.

Определим непрерывную функцию ядра как $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, представляющую собой пары скалярных произведений. В данной работе выберем гауссово ядро с радиальной базовой функцией (*англ. Radial basis function, RBF*):

$$K(x_i, x_j) = \exp(\gamma \|x_i - x_j\|^2), \quad (7)$$

где γ — параметр ядра

Ядро RBF обладает преимуществами перед другими ядрами:

- Позволяет решить задачу (5) с условиями (6) в случае, когда выборка не разделима линейно
- Один параметр — γ , контролирует степень влияния признаков на границу решения
- Значения ядра $K(x_i, x_j)$ лежат в пределах $(0, 1]$, не обращаются в 0 и не уходят в бесконечность

Определим гиперпараметры — это параметры, которые определяются однажды и не меняются в ходе обучения классификатора. В данной задаче такими параметрами являются C и γ . Подбор гиперпараметров осуществляется с помощью кросс-валидации и поиска по сетке, описанных в статье [8].

Алгоритм SVM чувствителен к неотмасштабированным данным, особенно в случае использования ядра RBF, представляющего собой Евклидово расстояние (7). В случае, когда значения свойств находятся в разных интервалах, незначительное отличие в одном может вывести за пределы значений второго свойства. Переведем все значения на отрезок $[0, 1]$:

$$z_{i_j} = \frac{x_{i_j} - \min_{i=1,\dots,l}(x_{i_j})}{\max_{i=1,\dots,l}(x_{i_j}) - \min_{i=1,\dots,l}(x_{i_j})} \quad (8)$$

5 Результаты исследований

5.1 Метрика

Качество полученным результатов будем оценивать с помощью микро-усредненной F-меры.

F-мера — это гармоническое среднее между точностью и полнотой:

$$F = 2 \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (9)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (10)$$

$$\text{recall} = \frac{TP}{TP + FN}, \quad (11)$$

TP (True Positives) — истинно положительные,

FP (False Positives) — ложно положительные,

FN (False Negatives) — ложно отрицательные классифицированные объекты

В случае микро-усреднения характеристики TP, FP, FN усредняются для каждого класса, далее вычисляется F-мера по формуле (9). Выбор такой метрики обоснован тем, что классы не сбалансированы по объему данных, и в данном случае влияние классов уменьшается за счет усреднения по характеристикам классификации, а не по F-мерам.

Представим общие результаты в виде таблицы:

	precision	recall	f-score
Sugar beet	0.91	0.98	0.94
Fat Hen	0.91	0.94	0.92
Scentless Mayweed	0.86	0.87	0.87
Charlock	0.89	0.94	0.96
Small-flowered Cranesbill	0.95	0.97	0.96
Maize	0.92	0.90	0.91
Shepherds Purse	0.81	0.73	0.77
Common wheat	0.85	0.85	0.85
Common Chickweed	0.92	0.92	0.92
Cleavers	0.87	0.82	0.84
Loose Silky-bent	0.88	0.82	0.85
Black-grass	0.72	0.70	0.71
micro F-score	0.89		

Рис. 12: Поклассовые характеристики и общая метрика

Ссылка на репозиторий с исследованиями — https://gitlab.com/JDima/plant_seed_classification

5.2 Выводы

Обратимся к таблице 12. Менее успешно удалось классифицировать объекты класса *Black-grass*. Результат обусловлен тем, что визуально отличить его образец от образца *Loose Silky-bent* довольно затруднительно:

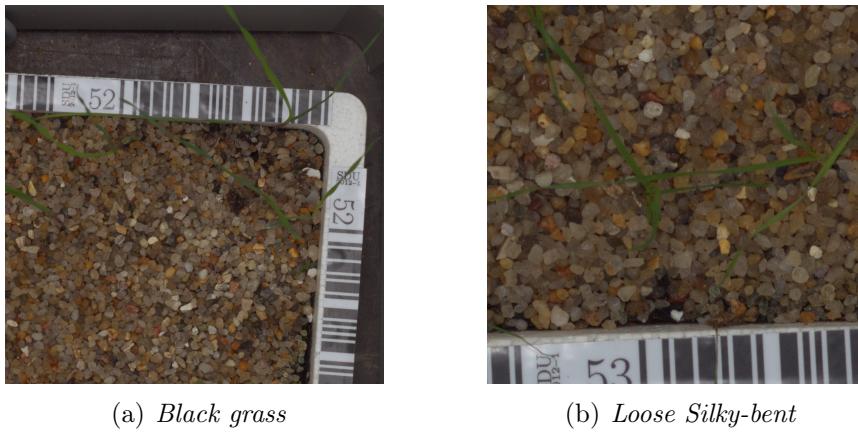


Рис. 13: Схожесть образцов классов

Предполагается, что качество классификации можно существенно улучшить, изменив метод сегментации. На данный момент не удается выделить части стебля растения, близкие к земле. Задать порог цвета для основания растения затруднительно, поэтому более тонкие различия между растениями, как на примере 13 (13(а) обладает не зеленым основанием), могут затеряться.

Также можно рассмотреть другие группы признаков, к примеру, текстурные, но возникает проблема с изображениями очень низкого разрешения, где отличительной текстуры не наблюдается. Поэтому необходимо найти подход, учитывающий несовершенства исходных данных.

5.3 Заключение

Задача классификации объектов остается открытой, алгоритмы машинного обучения и сегментации изображений развиваются ежедневно. Сделанные выводы говорят о достигнутом результате, а дальнейшие исследования особенностей задачи могут помочь поднять качество классификации.

Список литературы

- [1] Giselsson, T., Jørgensen, R., Jensen, P., Dyrmann, M., and Midtiby, H. (2017). A Public Image Database for Benchmark of Plant Seedling Classification Algorithms.
- [2] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
- [3] Oliphant, T. E. (2006). A guide to NumPy (Vol. 1). Trelgol Publishing USA.
- [4] Panja, D., Poppe, R. (2018). INFOIBV. Image Processing course, Universiteit Utrecht.
- [5] Wojnar, L., Kurzydłowski, K. J. et al. (2000). Practical Guide to Image Analysis, ASM International.
- [6] Suzuki, S. and Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following.
- [7] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python
- [8] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin (2003). A Practical Guide to Support Vector Classification.