

Распознавание поражений листьев

Введение в глубокое обучение. Проект, использующий анализ изображений листьев для распознавания болезней растений

Цель проекта

Цель этого проекта – осуществить первый подход к компьютерному зрению и познакомиться с алгоритмами, лежащими в основе обработки и извлечения признаков из изображений. Вам предстоит обучить классификатор заболеваний растений на основе изображений их пораженных листьев.

Основные инструкции

Язык программирования – Python. Вы можете организовывать и именовать свои файлы по своему усмотрению, соблюдая при этом требования к проекту и ограничения, перечисленные ниже.

В данном проекте предусматривается использование специализированных библиотек Python, реализующих различные методы МО для классификации изображений, а также инструменты для первичной обработки изображений, вывода графиков и т.д. Однако, при сдаче проекта будет проверяться знание алгоритмов, реализуемых данными библиотеками, поэтому ваш отчет должен включать в себя внушительную теоретическую часть, на которую вы будете опираться при реализации и сдаче проекта.

Вы должны уметь объяснять понятия аугментации, ее задачи и алгоритмы. Баллы будут начисляться в зависимости от ясности ваших объяснений. Кроме этого вы должны уметь рассказать об алгоритмах обработки изображений, извлечения из них признаков, об архитектуре нейронной сети, ее составных компонентах (если вы используете нейронные сети), или о других методах МО, которые вы задействуете, а также об особенностях, связанных с решением задачи классификации изображений. Вы должны суметь пояснить, как «компьютер» видит изображение и что лежит в основе распознавания изображений.

Для вдохновения, полезной информации и для понимания командного проекта можно ознакомиться с [проектом](#)

Данные

Набор данных в данном проекте представляет собой набор изображений листьев, распределенных по типу заболевания, а также изображения здоровых листьев. Данные хранятся в архиве leaves.zip

Основная задача

Часть 1: Анализ набора данных

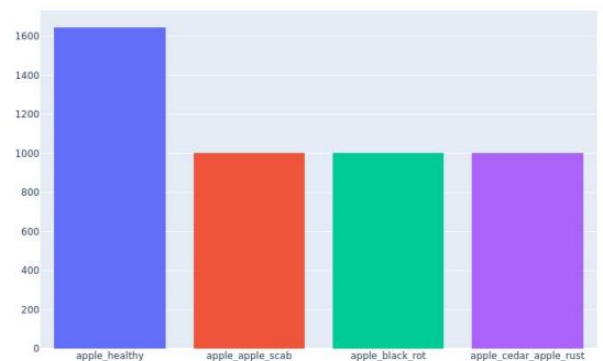
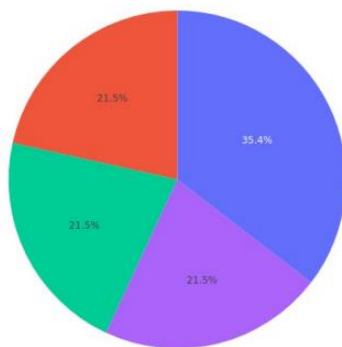
Напишите программу с именем Distribution.[расширение], которая принимает в качестве аргумента каталог и извлекает изображения из его подкаталогов. Эта программа, а значит и вы, должны извлечь и проанализировать/понять набор данных из изображений и реализовать круговые и столбчатые диаграммы для каждого типа растений — ваша программа также должна получить имя каталога, чтобы соответствующим образом назвать диаграммы.

```
$> find . -maxdepth 2
./Apple
./Apple/apple_healthy
./Apple/apple_apple_scab
./Apple/apple_black_rot
./Apple/apple_cedar_apple_rust
```

```
$> ./Distribution.[extension] ./Apple
```

Пример ожидаемого вывода:

apple class distribution



Часть 2: Аугментация (дополнение) данных

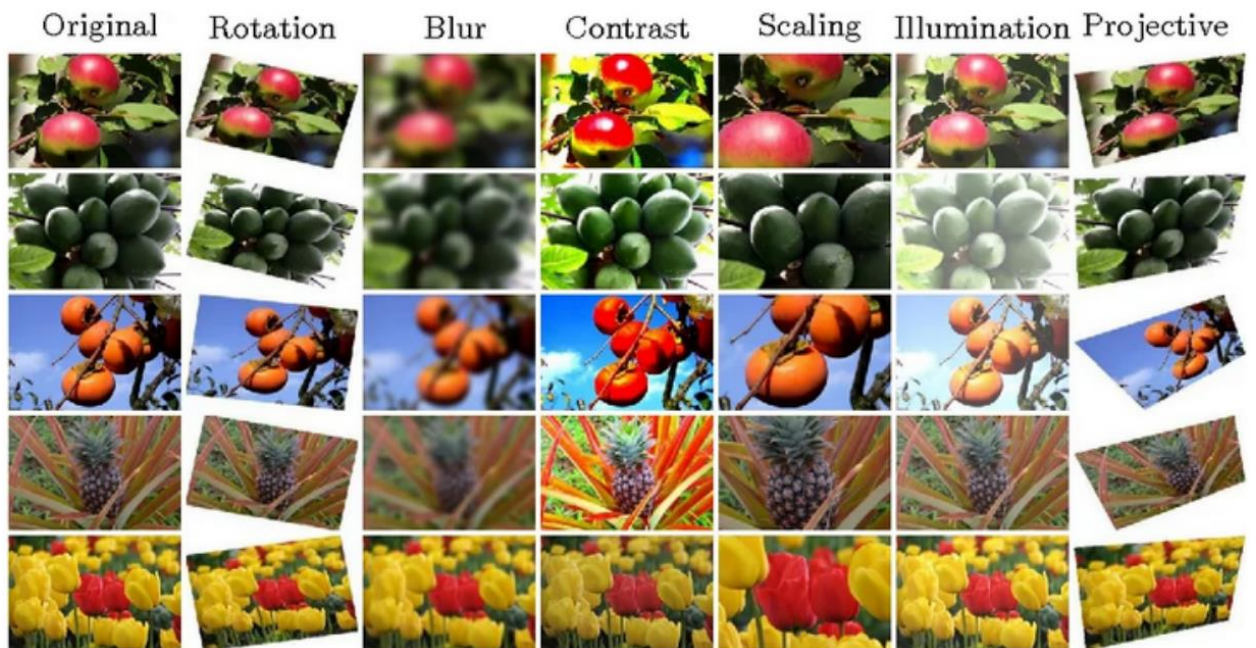
Можно заметить, что набор данных не сбалансирован. Вы должны сбалансировать количество изображений для каждого сорта и заболевания. Вот пример списка методов, которые вы можете использовать (также вы можете использовать любые другие методы):

- Переворот (flip)
- Поворот (rotate)
- Наклон (skew)
- Сдвиг (shear)
- Кадрирование (crop)
- Искажение (distortion)

Вы должны написать программу Augmentation.[расширение] чтобы отобразить 6 типов дополнения данных для каждого изображения, предоставленного вашей программе, они также должны быть сохранены с исходным именем файла, за которым следует имя типа дополнения.

```
$> ./Augmentation.[extension] ./Apple/apple_healthy/image (1).JPG
```

Пример ожидаемого вывода:



```
$> ls
image (1)_Flip.JPG
image (1)_Rotate.JPG
image (1)_Skew.JPG
image (1)_Shear.JPG
image (1)_Crop.JPG
image (1)_Distortion.JPG
```

Вам необходимо разместить новые изображения в существующем каталоге, чтобы сбалансировать ваш набор данных.

Часть 3: Обработка изображений

В данном разделе рекомендуется выбрать библиотеку, которая облегчает рабочий процесс обработки изображений, например, [plantCV](#). Необходимо реализовать различные методы прямого извлечения характеристик из изображения листа. Опять же, вы должны отобразить не менее 6 преобразований изображения.

```
$> ./Transformation.[extension] ./Apple/apple_healthy/image (1).JPG
```

Пример ожидаемого вывода:

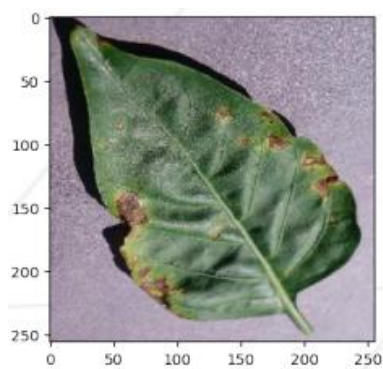


Figure IV.1: Original

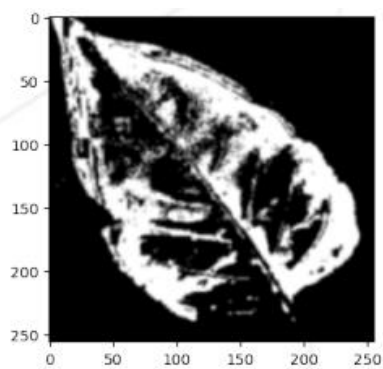


Figure IV.2: Gaussian blur

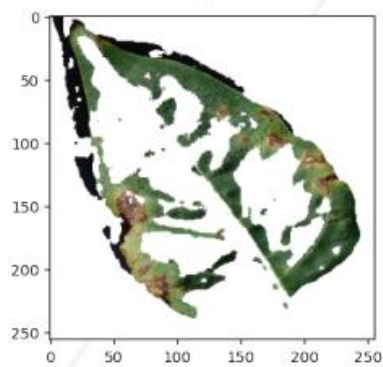


Figure IV.3: Mask

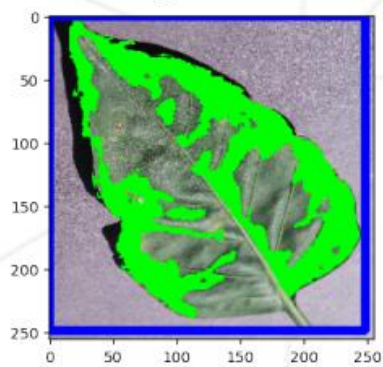


Figure IV.4: Roi objects

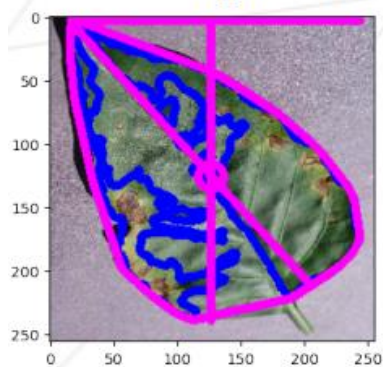


Figure IV.5: Analyze object

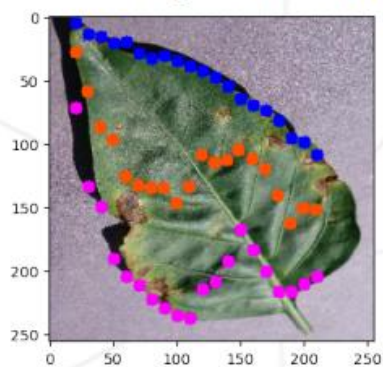
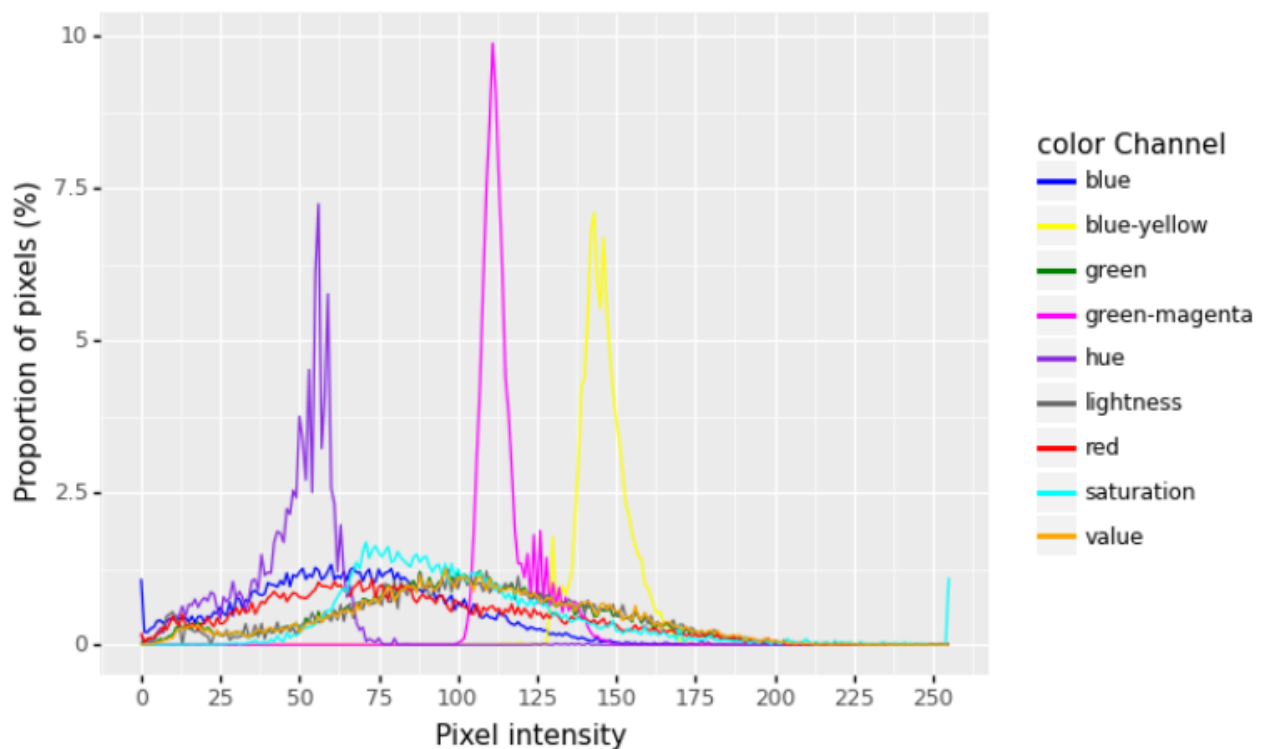


Figure IV.6: Pseudolandmarks



Если вашей программе дан прямой путь к изображению, она должна отобразить ваш набор преобразований изображений. Однако, если ей дан исходный путь к каталогу, заполненному несколькими изображениями, она должна затем сохранить все преобразования изображений в указанном целевом каталоге. Например,

```
$> ./Transformation.[extension] -src Apple/apple_healthy/ -dst dst_directory -mask
```

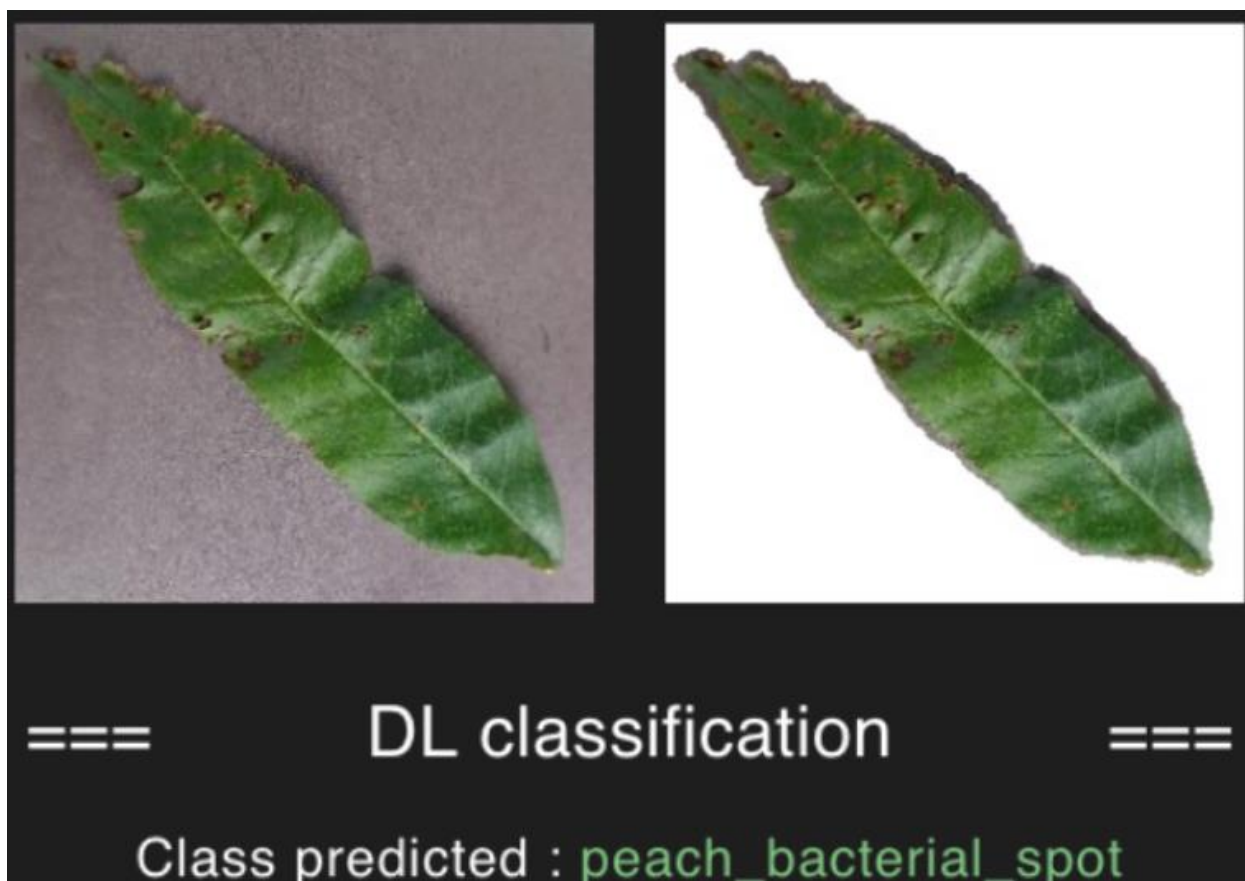
Часть 4: Классификация

В этой части вы должны написать программу с именем `train.[extension]`, которая принимает в качестве параметра каталог и извлекает изображения из его подкаталогов. Затем она должна аугментировать/изменить эти изображения, чтобы изучить характеристики заболеваний листа. Эти знания должны быть сохранены и возвращены в .zip архиве, который также включает ваши аугментированные/измененные изображения.

```
$> ./train.[extension] ./Apple/
```

Для оценки вам нужно будет написать вторую программу, которая использует полученные вами признаки листа. Она должна принимать в качестве аргумента изображение из вашей базы листьев и предсказывать его заболевание или отсутствие заболевания. Вам, конечно, нужно будет разделить ваш набор данных на две части, одну для обучения и одну для проверки. Желательно, чтобы ваши прогнозы имели точность более 90%.

```
$> ./predict.[extension] ./Apple/apple_healthy/image (1).JPG
```



Сдача проекта

Поскольку проект командный, то он выполняется совместно всеми участниками команды. Однако должен быть выбран тимлид, в папке которого будут располагаться все материалы проекта. Рабочие программы с исходным кодом должны быть размещены в индивидуальной папке тимлида на Яндекс Диске (под студенческой учетной записью) Основы Python_Фамилия\Классификация листьев. В этой же папке необходимо разместить отчет по проекту, оформленный средствами LaTeX или Word с описанием вашей работы (необходимо включить формулы, код, выкладки и рассуждения, описание инструментов, которыми вы пользовались при реализации проекта). Отчет обязательно должен быть оформлен отдельным файлом. Также необходимо подготовить и разместить в папке презентацию проекта с обязательным указанием участников команды и тимлида. Защита проекта проходит очно, вы должны иметь возможность показать на ноутбуке программный код, отчет и презентацию, кроме этого все презентации перед защитой нужно скинуть в общую группу в телеграм, чтобы во время защиты все студенты группы могли видеть и следить за вашей презентацией, так как проектора у нас не будет.

Время на выполнение – 3 недели. Защита проектов у гр.09-121 6.02.2025 с 8:30, у группы 09-122 защиты 5.02 и 7.02 по расписанию, (присутствие всей группы на этих парах обязательно).