**Задача двухклассовой классификации изображений**

*Отчет по исследовательскому проекту.*

Автор: Гущин Александр

Группа: 174, МФТИ

Руководитель: В. В. Стрижов

Консультант: Василий Лексин

Дата: 25.11.2014

**Содержание**

# Введение

## Описание проекта

### Аннотация

В работе решается задача двухклассовой классификации изображений. Задача разбивается на два этапа - локализации текста на изображении и определении, содержит ли текст запрещенную информацию. Для нахождения символов на изображении производится локализация текста с помощью алгоритма Stroke Width Transform. Для распознавания символов используется OCR (Optical Recognition System) Tesseract. Качество работы алгоритма оценивается с помощью Area Under Curve.

Ключевые слова: computer vision, Stroke Width Transform, text localizaton, Optical Charaster Recognition, Tesseract, text recognition.

### Мотивация

1. **Цель проекта**

Главной целью проекта является обучение классификатора, способного выполнять двухклассовую классификацию изображений.

1. **Примение результатов**

Результаты проекта могут быть использованы для фильтрации изображений с определенным текстом. Классификатор самостоятельно определяет, какой текст необходимо фильтровать, обучаясь на тренировочной выборке.

1. **Описание входных данных**

Обучающий и тестовый набор изображений в формате jpeg, полученных с сайта avito.ru (данные можно скачать по ссылке <http://inclass.kaggle.com/c/searching-for-contacts-in-the-picture/data>). По правилам Avito некоторая информация (например, номера телефонов, запрещена к размещению на изображениях). Для обучающей выборки известно, запрещено изображение или нет.

1. **Критерий качества**

В проекте решаются несколько задач в следующем порядке:

1. локализация текста
2. распознавание текста
3. выделение фич для классификатора
4. классификация изображений

Критерием качества классификации является AUC.

1. **Требования для успешного завершения проблемы**

Достижение AUC порядка 0.75

1. **Осуществимость и риски проекта**

Проект является реально осуществимым,так как основывается на реальных алгоритмах, достигающих значительного качества при решении своих подзадач.

При локализации текста и распознавании символов возникают проблемы с текстом, имеющим полупрозрачные символы и/или расположенным на неоднородном фоне, а также текстом с нечеткими границами.

1. **Алгоритмы и методы решения поставленной задачи**

Проект выполнен на языке программирования Python 2.7

В проекте использованы:

1. алгоритм Stroke width Transform [1,2] на этапе локализации текста
2. OCR Tesseract на этапе распознавания текста [3]
3. SVM с линейным ядром из пакета sklearn

Переформулируем задачу в терминах компьютерного зрения и машинного обучения. В начале нужно выявить все положения текста на этих фотографиях, распознать текст и определить, содержит ли распознанный текст запрещенную информацию. На основе полученных в процессе работы с фотографией признаков (которыми являются, например, количество локализованных областей текста, распознанный текст, наличие запрещенной информации в нём) построить двухклассовый классификатор, определяющий, содержит ли изображение запрещенный текст.

### Литература

[1] E. O. Epshtein Boris and Y. Wexler, “Detecting text in natural scenes with stroke width

transform,” 2010.

[2] K. Saurav and A. Perrault, “Text detection on nokia n900 using stroke width transform,” 2010.

[3] J. Canny, “A computational approach to edge detection,” 1986.

[4] R. Smith, “An overview of the tesseract ocr engine,” 2007.

### Подготовка системы

Для Linux:

1. установить программу для установки библиотек питона:

sudo apt-get install pip

1. Установить openCV:

https://help.ubuntu.com/community/OpenCV

1. установить необходимые пакеты для Python:

install ipython

sudo apt-get install pip scipy

sudo pip install pandas scikit-learn matplotlib pytesseract nltk

1. Установить реализацию SWT:

http://libccv.org/doc/doc-swt/

1. Установить Tesseract OCR и скачать данные для английского и русского языков:

https://code.google.com/p/tesseract-ocr/

## Архитектура системы

### Постановка задачи

Математическая постановка задачи может быть найдена здесь:

### A1 diagram

Архитектура системы, описанная в формате IDEF0.

## Структура проекта

Файлы расположены в папках:

* [doc](http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Voronov2013TextRecognition/doc/) – документация по проекту;
* [code](http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Voronov2013TextRecognition/code/) – функции выполняющие алгоритм;
* [web](http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Voronov2013TextRecognition/web/) – файлы, относящиеся к веб-разработке;
* [data](http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Voronov2013TextRecognition/data/) – синтетические и реальные данные.

## Basic data structures

Базовые структуры данных:

### Изображение

|  |  |  |
| --- | --- | --- |
| img | [m, n, [3]] | Описание изображения в матричном виде: координаты пикселя и 3 компоненты цвета (в формате RGB). |

### Text Localization

|  |  |  |
| --- | --- | --- |
| raw\_features | list | Экстремальные области, обнаруженные SWT, в формате (x1, y1, x2-x1, y2-y1) |
| file\_list | list | Список файлов из директории |

### Text recognition

|  |  |  |
| --- | --- | --- |
| rawdf | pandas.DataFrame | Экстремальные области, обнаруженные SWT, в формате (x1, y1, x2-x1, y2-y1). |
| nbox | int | Количество регионов с текстом, найденных SWT на фотографии |
| r1, r1 | (int, int) | (x1,y1), (x2,y2) |
| crop | [m,n,[3]] | регион, который был обнаружен SWT |
| eng, rus | str | Распознанный текст |
| fold | int | Увеличение полей crop, используется для дополнительных попыток распознавания |
| trigrams |  | триграмы - всевозможные сочетания трех подряд идущих букв из распознанного текста на фотографии |
| alltrigrams | Dict | содержит число раз, сколько был распознан каждые триграмм |
| toban | Dict | сколько раз триграм был распознан на фотографиях, которые необходимо забанить |
| probability | Dict | доля раз, которые триграм встречался на забаненных фотографиях |
| banned\_prob | list | фича для классификатора, для каждой фотографии высчитываемая следующим образом: сумма probability[trigram] по всем триграмам trigram, полученным из распознанного текста |

### Классификация

|  |  |  |
| --- | --- | --- |
| X\_train | np.array | фичи, выделенные из данных |
| y\_train | list | ответы для обучающих данных: 0 - изображение не содержит запрещенной информации, 1 - содержит |

### Выходные данные

|  |  |  |
| --- | --- | --- |
| Text positions |  | Изображение, где выделены похожие на текст ER |
| Report |  | Технические подробности запуска: время работы, параметры и т.д. |

# Module Interfaces (for Synthetic Data)

This section provides list of the basic algorithms. See more detailed description at the section “Algorithms description”.

Folder: [code](http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Voronov2013TextRecognition/code/)

|  |  |
| --- | --- |
| **Название алгоритма** | **Короткое описание** |
| SWT | Алгоритм, который находит регионы с текстом  images -> raw\_features, text regions |
| Tesseract | Алгоритм, который распознает текст на регионах с текстом (text regions)  text regions -> rec |
| Trigrams | Разбиение текста на триграммы и подсчёт частоты (эмпирически оцененной вероятности) их наличия в забаненных изображениях.  rec -> banned\_prob |
| LinearSVC | Алгоритм SVM с линейным ядром, предсказывающий для изображения вероятность оказаться забаненным.  raw\_features, banned\_prob -> pred\_norm |

# Real Data Processing

This section describes modules for real data processing and conversion

## Data download

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Input** | **Function inputs** |  |
| avito\_test/, avito\_train/ | folders with images | jpeg images |
| **Output** | **Function outputs** |  |
| submission.csv | the desired probabilities | float numbers in [0,1] |

## Saves for intermediate results

|  |  |
| --- | --- |
|  |  |
| **file name** | **result** |
| features/raw\_test\_CCV, features/raw\_train\_CCV | SWT results |
| features/rec\_test\_CCV\_0\_20\_40, features/rec\_CCV\_0\_20\_40 | Tesseract results |
| features/X\_train,  features/X\_test | features for LinearSVC (SVM) |

# Test Modules List

# System Launch And Testing

## System launch

## System testing

## System performance

# System tests visualisation

# Project Review

## Literature

September 25:

## Statement

October 2:

## IDEF

October 9:

## Code

October 16:

## Unit-tests

October 23:

## Data

October 30:

## Tests

November 6:

## Profiler

November 13:

## Report

November 20:

## Web

November 27:

# The Project Is Located ON: