

1 時間で画像検索エンジンを作る

2017/3/13, CGVI研究会@東京

松井勇佑
国立情報学研究所



1時間で画像検索エンジンを作る

www.simple-image-search: x


www.simple-image-search.xyz

Simple image search engine


ファイルを選択 選択されていません

送信


Query:




Results:






0.962598



0.977681



1.02553
Alamy stock photo



1 時間で画像検索エンジンを作る

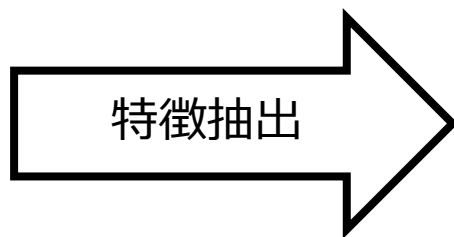
- 近年, 様々なライブラリが整備され, 簡単に画像処理が行える
- 画像をクエリとした画像検索のエンジンを作るデモ
- サーバを立ち上げるところから始め, 全pythonで0からライブコーディング
- AWS EC2 + Flask + Keras (VGG16)
- html **24**行, python **62**行 (うちimport 19行)



<https://github.com/matsui528/sis>

DEMO: <http://www.simple-image-search.xyz/>

画像表現


$$\begin{bmatrix} 0.21 \\ 0.43 \\ 0.18 \\ 0.02 \\ 0.59 \end{bmatrix}$$

- 画像を一本のベクトルに
- ベクトルが似ていれば画像も似ている

画像表現



特徴抽出

$$\begin{bmatrix} 0.21 \\ 0.43 \\ 0.18 \\ 0.02 \\ 0.59 \end{bmatrix}$$

- 画像を一本のベクトルに
- ベクトルが似ていれば画像も似ている

- かつてはSIFT+BoF. 様々な改良が提案された [1, 2]
- 最近ではCNN [3, 4]
- 実は、識別用の事前学習モデルを再利用するだけでも、かなり説得力のある特徴表現になっている
 - 例：猫画像ベクトルに対し、猫や猫に近い動物のベクトルが似る（直感に沿う）

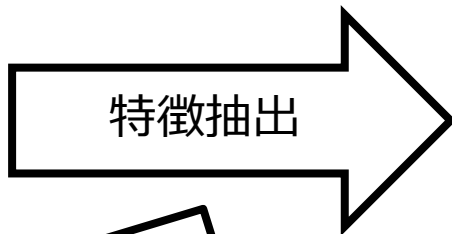
[1] J. Sivic and A. Zisserman, "Video Google: a Text Retrieval Approach to Object Matching in Videos," ICCV 2003

[2] H. Jégou and A. Zisserman, "Triangulation Embedding and Democratic Aggregation for Image Search," CVPR 2014

[3] A. Babenko and V. Lempitsky, "Aggregating Deep Convolutional Features for Image Retrieval," ICCV2015

[4] A. Gordo et al., "Deep Image Retrieval: Learning Global Representations for Image Search," ECCV 2016

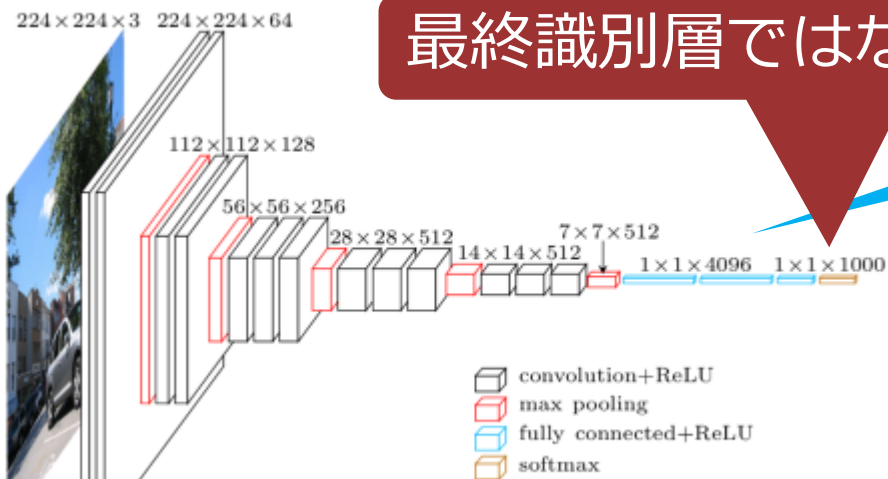
画像表現



$\begin{bmatrix} 0.21 \\ 0.43 \\ 0.18 \\ 0.02 \\ 0.59 \end{bmatrix}$

- 画像を一本のベクトルに
- ベクトルが似ていれば画像も似ている

- 今日はVGGモデル[5]のfc6層を特徴ベクトルとして用いる

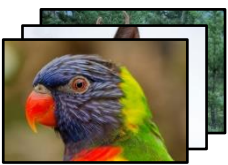


<https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/>

最近のライブラリでは、
ImageNetで事前学習
されたVGGモデルが
関数一つで呼び出せる

offline

offline.py



img

Feature Extractor
(Keras with Tensorflow)

$\begin{bmatrix} 0.21 \\ 0.43 \\ 0.02 \\ 0.59 \end{bmatrix}$ $\begin{bmatrix} 0.17 \\ 0.84 \\ 0.22 \\ 0.86 \end{bmatrix}$...
feature

online

server.py

$\begin{bmatrix} 0.21 \\ 0.43 \\ 0.02 \\ 0.59 \end{bmatrix}$ $\begin{bmatrix} 0.17 \\ 0.84 \\ 0.22 \\ 0.86 \end{bmatrix}$...

Search

Web Server
(Flask)

Feature Extractor

$\begin{bmatrix} 0.42 \\ 0.73 \\ 0.31 \\ 0.07 \end{bmatrix}$

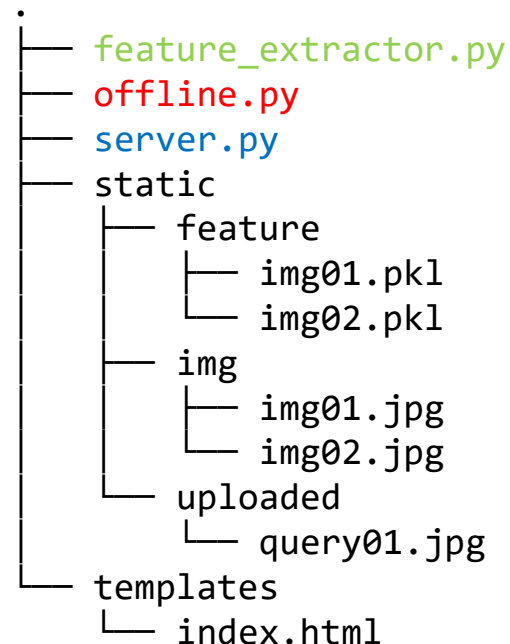
Browser

Simple image search engine

ファイルを選択 選択されていません

送信

Query:



ここから

ライブコーディング

知見

- 特徴抽出（一回の推論）は，2017年現在のCPUで1秒前後
 - 1.6 s
m4.large: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
 - 0.9 s
t2.large: Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
- 1000枚の画像に対する探索は10 ms
- 自然画像に対する検索では，VGG事前学習モデルでもかなり直感に沿う検索が出来る
- 技術の進歩が早いので，昨日動いたライブラリが今日動かないということはよくあるので注意
 - 今回の依存関係：numpy==1.11.3, Pillow==4.0.0, h5py==2.6.0, tensorflow==1.0.0, Keras==1.2.2, Flask==0.12

次のステップ

実装に関すること

- オフラインの特徴抽出は別に行う
- 「通信部分」と、「クエリ画像から特徴抽出・検索」を切り分ける (RPC: thrift, gRPC, etc)
- セキュリティ考慮

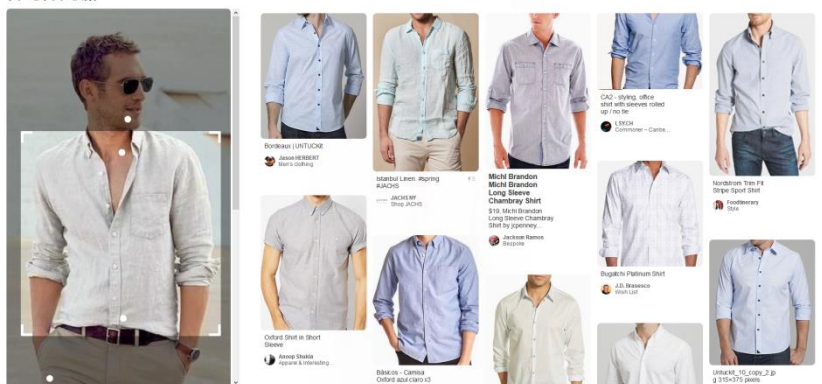
画像検索に関すること

- 特徴量抽出器をファインチューン
- 探索を高速化 (近似最近傍探索: FLANN, PQ, etc)
- クエリ拡張, 位置補正, etc

発展的な内容

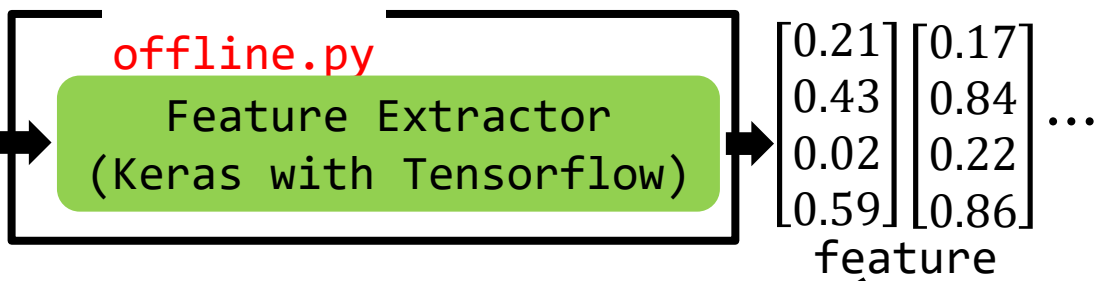
- 領域検索
- *Pinterest*
“Visual Discovery at Pinterest”
A. Zhai, et al., WWW 2017

ズームイン検索

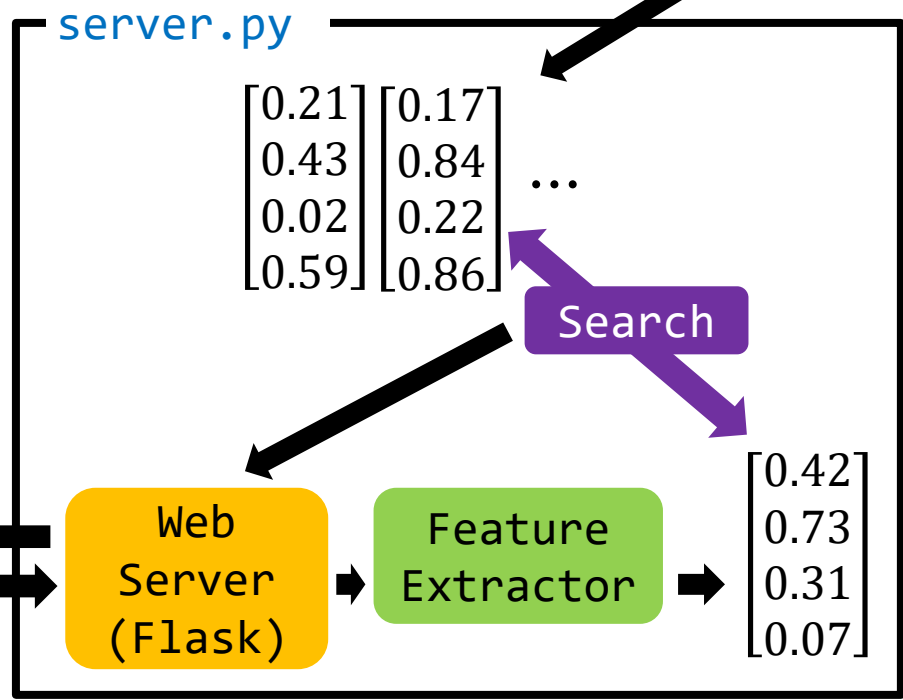
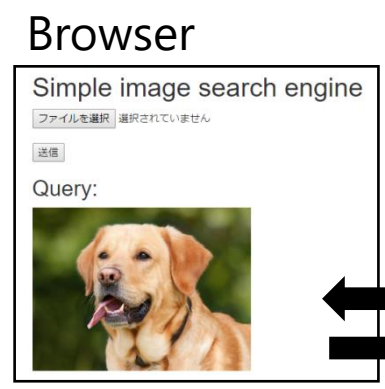


<https://jp.pinterest.com/>

offline



online



```

feature_extractor.py
offline.py
server.py
static
├── feature
│   ├── img01.pkl
│   └── img02.pkl
├── img
│   ├── img01.jpg
│   └── img02.jpg
├── uploaded
│   └── query01.jpg
└── templates
    └── index.html
  
```

謝辞：本発表内容はJSPS科研費16H07411，およびJST ACT-Iの支援を受けたものです。また，ドワンゴメディアヴィレッジから技術的アドバイスを頂きました。