

bash+AI で 画像テキスト検索

- Google Cloud Vision の場合 -

bash と Cloud Vision API

- bash は古いが安定しているので
確実に動作させたい場合に安心感がある
- Windows 10 にも bash が搭載される
- UNIX, Mac, Windows で共通して使える

bash から利用するには

(コマンドライン)

- JSON 形式のファイルを用意し
curl を使って呼び出す
- Shell Script を利用することで
便利に使えるようになる

実際の curl コマンド

```
curl -v -k -s -H
```

```
"Content-Type: application/json"
```

```
https://vision.googleapis.com/v1/images:
```

```
annotate?key=★ --data-binary
```

```
@~/temp.json
```

- ★は API キーで各自登録したものを指定
- @以降には JSON ファイルパスを指定
- v は確認のために指定している

実際の JSON ファイル

```
{"requests": [{"image": {"content": "
```



```
"},"features": [{"  
    "type": "TEXT_DETECTION",  
    "maxResults": 5  
}]}]}
```

● は画像を BASE64 エンコードした文字列

戻ってくるデータは JSON

- Cloud Vision API を curl で呼び出すとテキストデータが戻ってくる。

テキストデータは JSON 形式

実際に戻ってきた JSON データ例

```
{
  "responses": [
    {
      "textAnnotations": [
        {
          "locale": "fi",
          "description": "C fi min 117 kinoko\n~",
          "boundingPoly": {
            "vertices": [
              {
                "x": 286,
                "y": 59
              },

```

(こんな感じで続く)

指定した画像ファイルを AI 検索

●Mac OS X 用

```
cd ~/Desktop
echo '{"requests":[{"image":{"content":"" > ./temp.json
openssl base64 -in $1 | cat >> temp.json
echo ""},"features":[{"
    "type":"TEXT_DETECTION",
    "maxResults":5
}}]}' >> temp.json
curl -v -k -s -H "Content-Type: application/json"
https://vision.googleapis.com/v1/images:annotate?key=★
--data-binary @./temp.json
```

コマンドラインから呼び出す

- ai_text.sh がシェルスクリプトファイル名
パラメーターは1つで画像ファイル名になる
正しいパスを指定しないと駄目

```
./ai_text ~/Desktop/1.png
```

返されたデータ

```
{
  "responses": [
    {
      "textAnnotations": [
        {
          "locale": "fi",
          "description": "C fi min 117 hatenablo\nbash\n~",
          "boundingPoly": {
            "vertices": [
              {
                "x": 286,
                "y": 59
              },

```

とりあえず JSON を grep 検索

```
./ai_text.sh ~/Desktop/1.png | grep "kinoko"
```



```
"description": "C fi min 117 hatenablo\nbash\nGoogle  
Cloud Vision\nTwitter, Inc. [US] https://twitter.com\nRED18  
\nEa MacReti\nenss base64 -n oso-El-capitan\nAdobes cs6  
\n00000\n00Nam..vert jsx\n00Nam...ert2.jsx\nM 51524. Jpo  
1940.10 14866\n00zlnDesignnDf\n",  
  "description": "kinokol",
```

wc でいくつ該当したか確認

```
./ai_text.sh ~/Desktop/1.png | grep "kinoko" | wc -l
```



2

フォルダ内画像を一括 AI 検索

- シェルスクリプトを改良してフォルダ内にある画像をまとめて AI 検索すると便利
- 結果はテキストファイルで保存。
後で確認したり再利用できる。
また、呼び出しコストが削減できる。

```
cd $1
file=*.*png
for i in ${file}
do
echo '{"requests":[{"image":{"content":"" > ./temp.json
openssl base64 -in $i | cat >> temp.json
echo ""},"features":[{"
    "type":"TEXT_DETECTION",
    "maxResults":5
}}]}' >> temp.json
curl -k -s -H "Content-Type: application/json" https://vision.
googleapis.com/v1/images:annotate?key=★ --data-binary
@./temp.json > "$i.txt"
done
```

コマンドラインから呼び出す

- ai_text_dir.sh がシェルスクリプトファイル名
パラメーターは1つでフォルダ名になる

```
./ai_text_dir ~/Desktop/images
```

処理した結果

●テキストファイル

(JSON 形式) が作成される

-  1.png
-  1.png.txt
-  2.png
-  2.png.txt
-  3.png
-  3.png.txt
-  4.png
-  4.png.txt
-  5.png
-  5.png.txt
-  6.png
-  6.png.txt
-  temp.json

もう一工夫して grep 検索も

```
cd $1
file=*.png
for i in ${file}
do
echo '{"requests":[{"image":{"content":"" > ./temp.json
openssl base64 -in $i | cat >> temp.json
echo ""},"features":[{"
    "type":"TEXT_DETECTION",
    "maxResults":5
}}]}' >> temp.json
curl -k -s -H "Content-Type: application/json" https://vision.
googleapis.com/v1/images:annotate?key=★ --data-binary
@./temp.json > "$i.txt"
done
grep $2 *.txt
```

2 番目に検索文字列を指定

```
./ai_text_dir_find.sh ~/Desktop/img "kinoko"
```



```
1.png.txt: "description": "C fi min 117~",  
1.png.txt: "description": "kinokol",  
6.png.txt: "description": "Google Cloud~",  
6.png.txt: "description": "l/kinokolinares-jp1",
```

－ 応用例 －

- 画面キャプチャーからの文字検索
- アップロード前にアダルト画像チェック
- 過去のデジカメ画像内の文字を検索
- 処理速度が上がれば映像内の文字なども
検索できる

終わり