



『SPY×FAMILY謎解き』のらくえんを Pythonで解いてみた（光学文字認識）



Suzuka22

2025年3月3日 20:35

...

見出し画像の出典先：[【『SPY×FAMILY謎解き』ネタバレ】全問題の答えと解説まとめ【5巻キャンペーン】| 二次元メモ](#)

▼ 目次

やること

参考文献

問題

答え

解説

Pythonで問題を解くための方針

使用した画像

1. 光学文字認識でORDEという4文字を特定

4文字を特定するソースコード

実行結果

すべて表示

やること

こんにちは、Suzu22です。前回はスパイファミリーのたぬき暗号の記事を書きました。引き続き、今回も2020年9月に開催されたキャンペーンの「SPY x FAMILY謎解き」に出てくる謎解き問題を解いてみました。今回は謎解き問題の上級3の一部を光学文字認識で解いてみました

参考文献

光学文字認識とはテキストの画像を機械で読み取り可能なテキストに変換するプロセスのことを言います

OCR とは? - 光学文字認識の説明 - AWS

OCR (光学式文字認識) とは、その仕組み、および Amazon Web Services での使用方法をご覧ください。

[aws.amazon.com](https://aws.amazon.com/ja/rekognition/)

以下は光学文字認識を使う際に参考にした記事です。環境構築を行うときに参考してください

Tesseract OCR をWindowsにインストールする方法 - ガンマソフト

OCRをPythonで操作できれば大量の紙資料の読み取りも自動化できます。特に郵便番号や請求書番号など定型書類の番号を読み

[gammasoft.jp](https://www.gammasoft.jp/tesseract-windows.html)

<https://qiita.com/ryuka0610/items/20fbea83fa2b193e2d84>

Tesseract OCRにて画像からの文字認識を試したメモ - Qiita

Anacondaにて tesseract-ocr を試したメモ。環境Anacondaを使って、OpenCVも入れている前提
qiita.com

<https://openillumi.com/pytesseract-tesseract-not-found-error-solution-guide/>

Pytesseractの「TesseractNotFound」エラーを解決する方法ガイド

Pytesseractの「TesseractNotFound」エラーを解決するためのステップバイステップガイド。Tesseractopenillumi.com

以下は画像処理で文字認識を行う際に参考にした記事です

【Python入門】画像処理で文字認識をしてみる【VS Code/PyOCR/Tesseract】 | 気ままなシュミログ

この記事では、Pythonを使ってOCR(光学的テキスト認識)で画像から文字を抽出する手順を解説しています。PyOCR、Tesseracttakaaki-hobby-blog.com

【Python入門】画像から読み取った文字の位置を取得する【VSCode/OpenCV】 | 気ままなシュミログ

takaaki-hobby-blog.com

<https://shikaku-mafia.com/opencv-line/>

【OpenCV】cv2.line関数の使い方【線分を描画する】

OpenCVのcv2.line関数について解説しています。cv2.line関数の使い方や引数などをサンプルコードと合わせて

shikaku-mafia.com

問題

こちらの問題です。問題としては新聞と広告の画像から何かのキーワードを抽出するとのことです。（ソース：[SPY×FAMILY SPECIAL MISSION](#)）

フォージャー一家が写っている広告と新聞の画像があります。

広告と新聞を照らし合わせて見る必要がありそうですね。ここで早速ヒントですが広告に写っているロイドに注目してください



図1 新聞の画像



図2 フォージャー一家が写っている広告の画像

答え

読者の皆さんには解けましたか？？ちなみに筆者は解けませんでした(汗)

答えはらくえんです。なぜらくえんなのか解説していきます

解説

まず、広告のロイドの近くに写っている**ORDER**という文字に注目します。この**ORDER**という文字が存在する箇所を新聞から探します。すると縦並びの文章にO,R,D,Eという文字が存在することが分かります。そして、**ORDER**の順番で辿っていくと、矢印を一筆書きしたような筆跡が出力されます。出力された筆跡の矢印が指し示す延長線上の文字を読むと「らくえん(樂園)」という単語になります。この単語は意味が通る単語なので正解である確率が高そうです。そのため、答えは「らくえん(樂園)」です

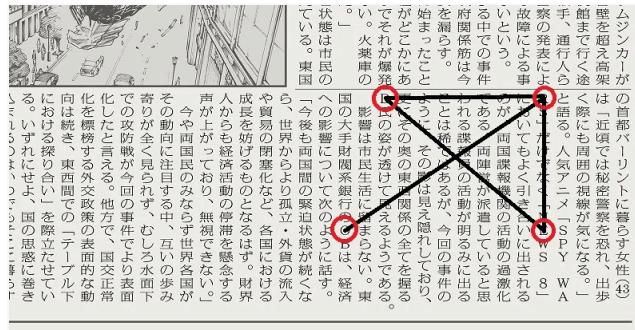


図3 問題の答えの画像

Pythonで問題を解くための方針

問題の解き方が分かりました。ここからはPythonで問題を解いていきます。下記はPythonで問題を解く手順です

1. 光学文字認識で**ORDE**という4文字を特定
2. 1で特定したそれぞれの4文字の座標値を縦並びの文字が写っている画像から取得
3. O→R→D→E→Rの順番で線を結ぶ

使用した画像

下記は使用した画像です

新聞から縦並びの文章のみを切り取った画像を使用しました



図4 光学文字認識を行う際に使用した画像

1. 光学文字認識でORDEという4文字を特定

光学文字認識で縦並びの文章を抽出します。新聞から縦並びの文章が写っている箇所を切り取ります。

4文字を特定するソースコード

このソースコードを実行したら果たして文字が抽出されるでしょうか
(ドキドキ)

```
#システムの利用を宣言する
import sys
import os

#PyOCRを読み込む
from PIL import Image
import pyocr

#OpenCVの利用を宣言する(画像に四角を書き込む際に使用)
import cv2

#Tesseractのインストール場所をOSに教える
tesseract_path = "C:\Program Files\Tesseract-OCR"
if tesseract_path not in os.environ["PATH"].split(os.pathsep):
    os.environ["PATH"] += os.pathsep + tesseract_path

#OCRエンジンを取得する
tools = pyocr.get_available_tools()
if len(tools) == 0:
    print("OCRエンジンが指定されていません")
    sys.exit(1)
```

```

else:
    tool = tools[0]

#画像の読み込み
file_path = "ここに文字を抽出したい画像のパスを入力"
img = Image.open(file_path)
img2 = cv2.imread(file_path)

box_builder = pyocr.builders.WordBoxBuilder(tesseract_layout=6)
text_position = tool.image_to_string(img, lang="jpn", builder=box_builder)

# 取得した座標と文字を出力し、画像に枠を書き込む
order_dict = {"O": None, "R": None, "D": None, "E": None, "R": None} # ORDER の順番
for res in text_position:
    if res.content in order_dict:
        print(res.content)

```

実行結果

D
R
O

文字が抽出されました！しかし、Eが抽出されていません。なぜでしょう
Eが抽出されていない原因を探るために全ての文字を抽出します

画像から全ての文字を抽出するソースコード

```

#システムの利用を宣言する
import sys
import os

#PyOCRを読み込む
from PIL import Image
import pyocr

#OpenCVの利用を宣言する(画像に四角を書き込む際に使用)
import cv2

#Tesseractのインストール場所をosに教える
tesseract_path = "C:\Program Files\Tesseract-OCR"
if tesseract_path not in os.environ["PATH"].split(os.pathsep):
    os.environ["PATH"] += os.pathsep + tesseract_path

```

```
#OCRエンジンを取得する
tools = pyocr.get_available_tools()
if len(tools) == 0:
    print("OCRエンジンが指定されていません")
    sys.exit(1)
else:
    tool = tools[0]

#画像の読み込み
file_path = "ここに文字を抽出したい画像のパスを入力"
img = Image.open(file_path)
img2 = cv2.imread(file_path)

box_builder = pyocr.builders.WordBoxBuilder(tesseract_layout=6)
text_position = tool.image_to_string(img, lang="jpn", builder=box_builder)

print(" ".join([res.content for res in text_position]))
```

縦並びの文章の文字が縦に表示されました

P
ア
暮
どす
る
で
一
後
と
政
る
計
中
下
走
1
ん
ら
東
真
る
状
は
東
の
い
東
で
と

4
だへ行西しに当仙し一ブー)ーし下んか態な西情う西あい名つと中国王且軒較、生届口ト、『に問中)?い間有夫こと間り車がただ落

に
外
還
用
た
ー
有
1
も
に
に
座
何
が
の
悪
と
の
丶
の
死
外
下
交
月
記
紀
同
/
B
・
人
PE
影
お
いも
が
い
確
化
ち
緊
事
普
選
交
し
屋
官
全
車
出
昨
叫
/
ブリ
償
寺
主
計
紋

ける
分
き
常
執
へ
あ
迫
件
レ
官
だ
た
外
が
中
層
較
剛
骨
胡
紳
<
ゞ
軸
凍
る
ま
め
つ
に
は
の
り
が
が
仁
1
た
と
用
乗
中
則
販
ン
克
M
の
AND
た
作
財
う
ら
か
火
今
不
い

高
は
キ
?
運
大
防
る
導
電
クソ
NN
】
え
着
だ
な
け
種
に
安
政
ま
な
の
警
転
使
音
リ
ンジ
。
で
て
状
?
い
で
が
始
を
府
る
い
改
家
手
館
壁
ム
凍
4
条
。
クリン
リ
似
い
態
二
?

そ
ど
ま
漏
閥
中
と
陣
の
、
い
ま
を
ジ
の
じ
ペ
ニ
ジ
ラ
「
。
(AN)
る
は
火
れ
と
こ
つ
ら
係
で
い
に
発
通
で
超
ン
2
の
う
ル
ン
た
：
N
爆
に
と
?
?
る
に
高
に
フ
ン
生
ニ

プ
し
川
NNNA
一澤のの発あと今件事よら途梨がはき国込るに向化化で寄そ声人成やう二へ國D更よとわでのにRとくはのいいゝ民ま?おは

を
し
の
り
の
今
が
か
長
錦
\
今
の
の
影
氏
に
うと
ど
れ
あ
が
お
8
語
際
一
首
つ
両
で
れ
い
け
続
標
た
攻
が
動
や
上
ら
を
易
世
後
影
大
響
の
そ
に
は
る
る
い
る
に
近

都
訪
国
ある
ずる
き構
ど防
全向
両が
や妨
の界も
才手は
雇の、
稀課
く両て
だ“も
頃バ
れを
るのれ
探しす
言戦
くに
国つ
経げ
閉か
両に
財市
が奥
をで
報

両国
もやけ信周で！ると？はにり東るえが見注民でて済る塞ら国つ内民透のは員陣恋よで気囲はりのり両いせ合西外る今ら目のお活

も化よ
間い系
生け東
影あ
の営報
くなア
の秘
ンだな
国つ
よい間
交回れ
すみり動
のなりの
てでて銀
活てで西
はる活が
機引<
二二視密
トろす政

で
ヽ
一
で政他
の
する
な
、
いの
とど
孤緊次行
に見
閥見
が動
派閥
きつ
メ
線警
に
う一
府も
国を
の策
方事
い中
ら無
合な
ヽ立
迫の
の劉
え係
えい
が萌

の
合
N
一
が
察
暮
か
絆
が
そ
の
際
一
の
で
件
む
、
ず
視
滞
る
各
・
状
よ
〇
ま
る
の
隠
今
明
し
活
い
E
S
氣
を
ら
?
一
そ
と
思
立
テ
表
い
で
し
互
世
で
を
は
国
外
態
う
B

らよ
全れ
回る
て動
にWP
に恐
すを
れに
員た
1面
国よ
ろい
界き
懸す
に貨
がに
はな
うて
レの
みい
の出
S8Y
なれ
女育
に暮
にせ
普的
交り
水の
各な

念
?
おの
続語
^
いで
をて
事に
る過
ぎ
る、
條
お氣
ら
巻て
ルな
正表
面歩
国い
す財
け流
くす
経?
あ握
め件
出と
激れ
8W
“
出人
目づ
すぎ
い下
動常
画

下
み
が
ご
○
る
界
る
入
な
？
済
東
る
る
へ
の
る
思
化
る
一
ム
一
歩
ぞ

文字が区切られて表示されています。しかし、表示された文字数が多すぎます。そこで、Eを検索します

ES

検索に引っ掛かりました！しかし、抽出された文字がEではなくてESになっています。はっきりとした原因は分かりませんが縦並びの文章が関係しているかもしれません、、、仕方がないので1のソースコードの

```
order_dict = {"O": None, "R": None, "D": None, "E": None, "R": None}
```

のEから

```
order_dict = {"O": None, "R": None, "D": None, "ES": None, "R": None}
```

ESに変更してみます。変更した場合は

```
D  
R  
O  
ES
```

Sも表示されていますが特定したい文字の1つであるEが抽出されました！

2 1で特定したそれぞれの4文字の位置を新聞から取得

とりあえず、ORDEの4文字が抽出されました。次は新聞の画像からORDEの位置を取得します

新聞の画像から文字の位置を取得するソースコード

```
#システムの利用を宣言する
import sys
import os

#PyOCRを読み込む
from PIL import Image
import pyocr

#OpenCVの利用を宣言する(画像に四角を書き込む際に使用)
import cv2

#Tesseractのインストール場所をOSに教える
tesseract_path = "C:\Program Files\Tesseract-OCR"
if tesseract_path not in os.environ["PATH"].split(os.pathsep):
    os.environ["PATH"] += os.pathsep + tesseract_path

#OCRエンジンを取得する
tools = pyocr.get_available_tools()
if len(tools) == 0:
    print("OCRエンジンが指定されていません")
```

```

        sys.exit(1)
    else:
        tool = tools[0]

    #画像の読み込み
    file_path = "C:/Users/homur/Downloads/mission.jpg"
    img = Image.open(file_path)
    img2 = cv2.imread(file_path)

    box_builder = pyocr.builders.WordBoxBuilder(tesseract_layout=6)
    text_position = tool.image_to_string(img, lang="jpn", builder=box_builder)

    # 取得した座標と文字を出力し、画像に枠を書き込む
    order_dict = {"O": None, "R": None, "D": None, "ES": None, "R": None} # ORDER の順番
    for res in text_position:
        #print(res.content) # 画像から抽出した文字を表示
        if res.content in order_dict:
            print(res.position)
            order_dict[res.content] = res.position
            cv2.rectangle(img2, res.position[0], res.position[1], (0, 0, 255), 2)

```

実行結果

```

((1002, 496), (1052, 518))
((1328, 497), (1347, 517))
((935, 753), (962, 807))
((1331, 772), (1386, 792))

```

文字の座標値が出力されました。次は出力された座標値がそれぞれの文字と合致しているのかopencvを使って確認します

文字の座標値を可視化するソースコード

```

#システムの利用を宣言する
import sys
import os

#PyOCRを読み込む
from PIL import Image
import pyocr

#OpenCVの利用を宣言する(画像に四角を書き込む際に使用)
import cv2

```

```
#Tesseractのインストール場所をOSに教える
tesseract_path = "C:\Program Files\Tesseract-OCR"
if tesseract_path not in os.environ["PATH"].split(os.pathsep):
    os.environ["PATH"] += os.pathsep + tesseract_path

#OCRエンジンを取得する
tools = pyocr.get_available_tools()
if len(tools) == 0:
    print("OCRエンジンが指定されていません")
    sys.exit(1)
else:
    tool = tools[0]

#画像の読み込み
file_path = "文字を抽出したい画像のパスを入力"
img = Image.open(file_path)
img2 = cv2.imread(file_path)

box_builder = pyocr.builders.WordBoxBuilder(tesseract_layout=6)
text_position = tool.image_to_string(img, lang="jpn", builder=box_builder)

# 取得した座標と文字を出力し、画像に枠を書き込む
order_dict = {"O": None, "R": None, "D": None, "ES": None, "R": None} # ORDER の順番
for res in text_position:
    #print(res.content) # 画像から抽出した文字を表示
    if res.content in order_dict:
        print(res.position)
        order_dict[res.content] = res.position
        cv2.rectangle(img2, res.position[0], res.position[1], (0, 0, 255), 2)

#四角を書き込んだ画像を表示する
cv2.imshow("image", img2)
cv2.imwrite("出力させたい画像のパスを入力", img2)
cv2.waitKey(0)
```

座標値を可視化した結果



図3 文字の座標位置を可視化した画像

4文字の位置を特定することがほぼ出来ています。しかし、Oを囲んでいる赤枠の位置が少々ずれています。また、Dを囲む赤枠が右隣に存在する文字の更まで囲んでいます。縦並びの文章が写っている画像を使用しているのが影響しているのかもしれません

3 O→R→D→E→Rの順番で線を結ぶ

とりあえず、文字の座標値の件は大目に見ることにします。(え)

4文字の座標値を特定することが出来ているとしてORDERの順番で線を結んでいきます。

opencvを使います

特定した文字を線で結ぶソースコード

```
#システムの利用を宣言する
import sys
import os

#PyOCRを読み込む
from PIL import Image
import pyocr

#OpenCVの利用を宣言する(画像に四角を書き込む際に使用)
import cv2

#Tesseractのインストール場所をosに教える
tesseract_path = "C:\Program Files\Tesseract-OCR"
if tesseract_path not in os.environ["PATH"].split(os.pathsep):
    os.environ["PATH"] += os.pathsep + tesseract_path

#OCRエンジンを取得する
tools = pyocr.get_available_tools()
if len(tools) == 0:
    print("OCRエンジンが指定されていません")
    sys.exit(1)
else:
    tool = tools[0]

#画像の読み込み
file_path = "文字を抽出したい画像のパスを入力"
img = Image.open(file_path)
img2 = cv2.imread(file_path)

box_builder = pyocr.builders.WordBoxBuilder(tesseract_layout=6)
text_position = tool.image_to_string(img, lang="jpn", builder=box_builder)

# 取得した座標と文字を出力し、画像に枠を書き込む
order_dict = {"O": None, "R": None, "D": None, "ES": None, "R": None} # ORDER の順番
for res in text_position:
    #print(res.content) # 画像から抽出した文字を表示
    if res.content in order_dict:
        print(res.position)
```

```

order_dict[res.content] = res.position
cv2.rectangle(img2, res.position[0], res.position[1], (0, 0, 255), 2)

# ORDERの順番で線を引く
order_keys = ["O", "R", "D", "ES", "R"]
prev_pos = None
for key in order_keys:
    if key in order_dict and order_dict[key] is not None:
        center_pos = ((order_dict[key][0][0] + order_dict[key][1][0]) // 2,
                      (order_dict[key][0][1] + order_dict[key][1][1]) // 2)
        if prev_pos is not None:
            cv2.line(img2, prev_pos, center_pos, (255, 0, 0), 2)
        prev_pos = center_pos

#四角を書き込んだ画像を表示する
cv2.imshow("image", img2)
cv2.imwrite("出力させたい画像のパスを入力", img2)
cv2.waitKey(0)

```

特定した文字を線で結ぶソースコードの実行結果

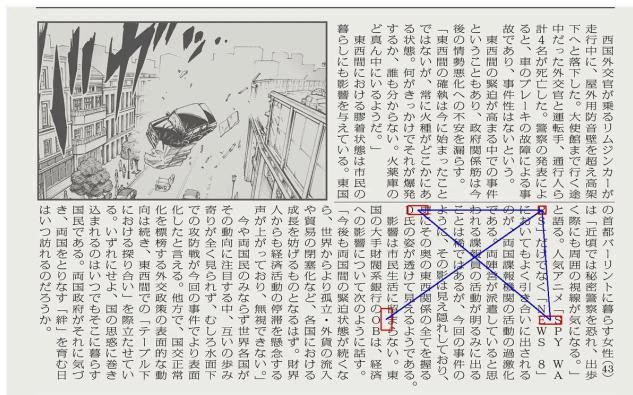


図4 光学文字認識と画像処理で求めた回答

矢印を一筆書きしたような結果が出力されました！！！！！矢印の延長線上の文字を読むとらくえんであることが確認できました

さいごに

今回は謎解き問題の上級3の問題の一部をPythonで解きました。本記事では新聞を画像として扱いました。そして、光学文字認識を使って画像から文字を抽出する手法で問題を解きました。新聞をテキストと扱ってOREDという4文字を探索する方法でも解けそうです。

今後の課題としては光学文字認識の精度を向上させる、新聞をテキストと扱って解くことが挙げられます

ここまで記事を読んでいただきありがとうございました！！！！！！！

[宣伝] 発信中のアカウントについて

※ウマ娘の声をフーリエ変換した記事や水栓の開閉判定をLineに通知する記事も書いています。suzuが私です。興味がある方はぜひご覧ください！

31-1. ウマ娘の声をフーリエ変換して比較してみた

(①準備編)

やることここにちは、生粋のウマ娘ファンです | • ω •)ノ2021年に公開した「画像のフーリエ変換」の記事がロングセラーになつ

vigne-cla.com

※Qiitaもやっています。

[備忘録] Youtubeを見ながらUnityの勉強 8日目 -

Qiita

はじめに ここにちは、Umamusume22です。しばらくunityの勉強をサボっていました。業務でUnityを使っている

qiita.com

※ zennもやっています

suzuka22さんの記事一覧

メーカー勤務のロボットエンジニア。ソフトウェア開発担当。短大から国公立大学に編入した珍しい経歴の持ち主。情報工学専攻。RO

zenn.dev

アニメを題材にテック記事を書いています。
UnityやITなどの備忘録も発信しています。

ぜひフォローよろしくお願ひします!