

画像を題材とした競技プログラミング用コンテンツの開発

松本 貴裕[†] 福井 昌則[‡] 高見 友幸[†]

[†] 大阪電気通信大学大学院 総合情報学研究科

[‡] 兵庫教育大学

キーワード：競技プログラミング，学習サイト，Python

1 はじめに

現在，プログラミング学習やプログラミングスキルの証明，高度な IT 人材を採用する手段として，競技プログラミングが活用されている。競技プログラミングとは，複数の参加者全員に同時に同一課題が複数出題され，その課題を素早く要求に応えられるプログラムの作成することを競うものである。

国内の代表的な競技プログラミングサイトとして，AtCoder[1]や Aizu Online Judge[2]などが挙げられる。また，各サイトはオンラインジャッジシステムが採用されており，ソースコードをアップロードすることでプログラムの実行結果の判定を行なっている。

しかし，競技プログラミングでは，コンピュータサイエンスや数学の問題が出題されており，論理的思考力を測ることができるが，画像処理やデータベース，Web スクレイピングといったエンジニアスキルを網羅的に測ることが出来ない。

また，競技プログラミングは C や C++，Java，Python などといった様々なプログラミング言語で問題を回答することが可能であるが，それにより問題がそれぞれの言語でも解ける問題に限定される。そのため，問題はより抽象的になる。

本研究では，画像を題材とした競技プログラミング用コンテンツの開発を行った。従来のコンピュータサイエンスや数学の問題がこの問題と置き換わる。画像を題材とした問題は競技プログラミングやプログラミング学習サイトでは，取り扱われてないため，現在の競技プログラミングシステムでは扱うことができない。そのため，画像を扱うために新たな競技プログラミングシステムを設計・構築を行った。また，プログラミング言語を

Pythonに限定して，より実践的な問題を構築した。

2 Python

Python とは，インタプリタ型言語であり，近年話題の Deep Learning や Internet of Things(IoT)，ビッグデータ処理などの開発を得意としている。特徴として C 言語や C++ などと比べシンプルに記述できることや標準で豊富なライブラリがある。競技プログラミングにも使用されている他，プログラミング学習の入門などといった幅広く用いられており，IEEE の調査[3]によれば，Python は近年最も注目されているプログラミング言語である。

本研究では，Python の豊富なライブラリを活用して画像処理，画像認識，Web スクレイピングのコンテンツを開発した。

3 競技プログラミングシステム

3.1 システム概要

競技プログラミングシステムは，学習サイト[4]との相互連携を目指すため学習サイトの機能を元に拡張を行った。

ユーザーは始めに自身のアカウントの作成を行いログインする。ログインするとフィールドマップに遷移する。ユーザーはそのページ競技プログラミングが開催される時刻まで待機する。開催時刻となるとコンテスト開催機能によりフィールドマップに競技プログラミング用リンクが表示され，ユーザーはそこから競技プログラミングのページへと遷移する。競技プログラミングページでは問題が一覧になっており，そこから回答する問題を選択していく。問題の回答方法は学習サイトと同様に Web サイト上に構築されたプログラミングエディターを用い，Web サイト上でコーディング

を行い回答する。回答ボタンを押すことで回答のソースコードを提出する。競技プログラミングページではこの工程を繰り返すことにより競技を行っていく。また、競技プログラミングページでは全てのページに常時残り制限時間を表示することでユーザーに制限時間を認識させている。

制限時間が終了した時、コンテスト開催機能によりユーザーは強制的にフィールドマップへと遷移させられる。また、制限時間外の回答を防ぐため回答ページは全てアクセス制限を行っている。

競技プログラミングが終了した時に再度リンクを押すことで採点結果とユーザーの順位が確認できるページへと遷移する。

以上の機能を有する競技プログラミングシステムの構築を行った。システム構成図を図1に示す。

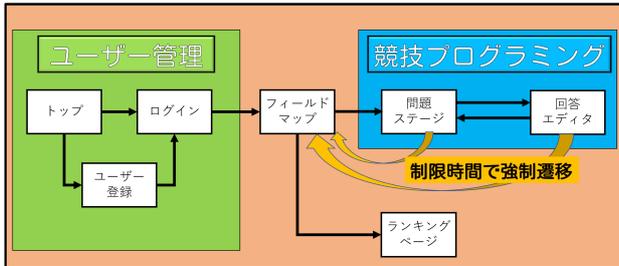


図1. システム構成図.

3.2 コンテスト開催機能

コンテスト開催機能は、競技プログラミングサイトのリンクの表示・非表示と時間外のアクセス制限により実現した。

リンクの表示・非表示は JavaScript を用い実装を行った。Date 関数で開催日時、終了日時を指定し show メソッド、hide メソッドでリンクの表示・非表示を行った。

次に時間外のアクセス制限は Python を用いて実装を行った。datetime モジュールを用い開催時間を指定した。そして時間外にリンクへ接続した場合 Flask の abort 関数を用い HTTP 403 エラーの表示を行った。

3.3 制限時間機能

制限時間機能は、競技プログラミングページである問題一覧ページ、問題詳細ページ、回答ページ全てに制限時間の表示を行うものである。

制限時間機能の実装には、JavaScript を用いた。Date 関数を用い現在時刻と終了時間を設定し、制限時間を計算しカウントダウン表示した。制限時間が終了した場合、window.location.href を用い、トップページへと強制遷移させる。これにより制限時間外の接続をアクセス制限と合わせて防ぐ。制限時間機能を図2に示す。



図2. 問題詳細ページの制限時間表示.

3.4 回答機能

学習サイトの回答エディターを改良し回答機能を実装した。Web サイト上のプログラミングエディターは Ace.js の CDN を用いて実装し、実行、回答処理は Python を用いて実装した。また、画像の問題を取り扱うために、ファイル参照ボタンで画像を確認することができる。エディター画面を図3に示す。



図3. Web 上で動作する回答エディター.

問題に回答するには、実行ボタンでエディター上のソースコードを実行し、プログラムが動作するか確認する。そして回答ボタンでソースコードを提出して問題に回答する。学習サイトでは、不正解の場合は問題に回答することができなかったが、競技プログラミングサイトでは正解・不正解問わず回答することが可能である。また、回答を行った時、問題一覧のページで回答済みかを確認することができる。問題一覧を図4に示す。



図 4. 問題一覧ページの回答確認.

その他、保存ボタンで現在のソースコードを保存や、リセットボタンで現在のソースコードを初期化することができる。

3.5 ランキング機能

ランキング機能では、制限時間終了後に競技プログラミングのサイトで自身のランキングと問題の正解・不正解を確認することが可能である。実装には、NoSQL データベースである TinyDB からユーザーの回答状況、正解状況を元に Python で処理を行い順位と正解・不正解の出力を行った。順位に関しては同率順位の場合も考慮するため pandas モジュールの rank メソッドを用いて出力を行った。ランキング機能を図 5 に示す。

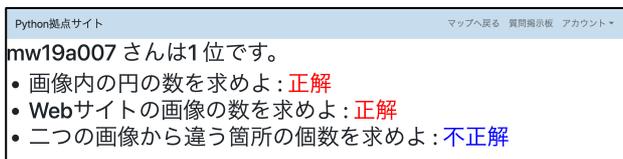


図 5. ランキングページ.

4 画像を題材としたコンテンツ

4.1 コンテンツ概要

本研究では、他サイトでは扱うことが出来ない画像を題材としたコンテンツを開発した。コンテンツとして、画像処理、画像認識、Web スクレイピングのプログラミングスキルを確認できるものを用意した。また、他サイトでは扱うことが出来ない Python の外部モジュールを用い Python の特徴が出るコンテンツ構成とした。

4.2 画像内の円の数を求めよ

この問題は入力値として与えられた円を含んだ画像から円の数を求める問題である。この問題を解くことで画像処理、画像認識のプログラミングスキルを確認することが可能である。入力画像を図 6、解答ソースコードを図 7 に示す。

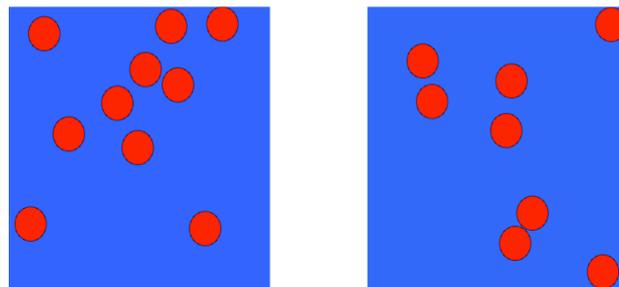


図 6. 左)入力画像, 右)採点時の入力画像.

```

1 import cv2 # opencv-pythonモジュール
2 import numpy as np # numpyモジュール
3
4 def getCircle(path):
5     img = cv2.imread(path) # ファイル読み込み
6     bgrLower = np.array([ 0, 0, 100]) # 抽出する色の下限(BGR)
7     bgrUpper = np.array([255, 255, 255]) # 抽出する色の上限(BGR)
8
9     mask = cv2.inRange(img, bgrLower, bgrUpper) # 読み込んだ画像からマスクを作成
10    result = cv2.bitwise_and(img, mask) # 読み込んだ画像とマスクを合成
11    gray = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY) # グレースケール化
12    # 輪郭抽出
13    contours, _ = cv2.findContours(gray, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
14
15    ans = len(contours) # 輪郭の数を抽出
16    return ans
17
18 assert getCircle('./user/files/a000718.png') == 10

```

図 7. 円の数を求めるプログラム.

4.3 画像から多種類の図形の数を求めよ

この問題は前問題の応用問題である。前問題では、画像内の円の数を求めるものだったが、この問題は画像内の三角形、四角形、五角形、円の数を求める問題となっている。この問題を解くことでより深い画像処理、画像認識のプログラミングスキルを確認することができる。入力値である多種類の図形を含んだ画像を図 8、解答のソースコードを図 9 に示す。

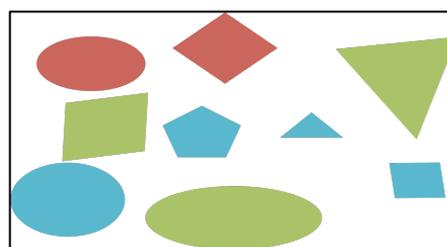


図 8. 多種類の図形を含んだ画像.

```

1 import cv2
2 import numpy as np
3
4 def countShape(path):
5     img = cv2.imread(path, cv2.IMREAD_GRAYSCALE) # ファイルを読み込み グレースケール化
6     contours, _ = cv2.findContours(img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) # 輪郭抽出
7     ShapeList = [0]*4 # 図形をカウントするためのリスト
8
9     for cnt in contours:
10        approx = cv2.approxPolyDP(cnt, 0.01*cv2.arcLength(cnt, True), True) # 周囲長
11        # 輪郭の近似の計算
12        if len(approx) == 3: ShapeList[0] += 1 # 点の数が3個の場合、三角形とする
13        elif len(approx) == 4: ShapeList[1] += 1 # 点の数が4個の場合、四角形とする
14        elif len(approx) == 5: ShapeList[2] += 1 # 点の数が5個の場合、五角形とする
15        elif 6 < len(approx) < 14: ShapeList[3] += 1 # 点の数が7個~13個の場合、円とする
16        else: ShapeList[3] += 1
17    return ShapeList
18
19 if __name__ == "__main__":
20    assert countShape('./user/files/140646.png') == [2, 3, 1, 3]

```

図 9. 画像の種類を判別するプログラム.

4.4 Web サイトから白を含む画像の数を求めよ

この問題は入力値として画像を含んだ Web サイトから白色を含む画像の数を求める問題である。Web サイトは画像と文字がランダムに配置されている。この問題を解くことで Web スクレイピング、正規表現、画像処理のプログラミングスキルを確認することが可能である。入力値の Web サイトを図 10、解答のソースコードを図 11 に示す。

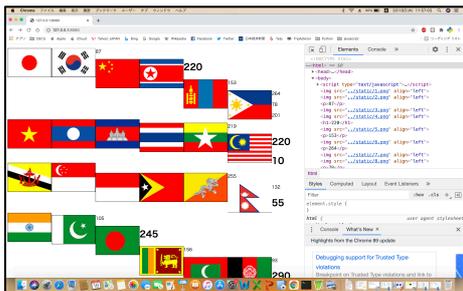


図 10. 入力値の画像を含んだ Web サイト。

```
1 from selenium import webdriver
2 from selenium.webdriver.chrome.options import Options
3 from PIL import Image
4 import re, io, requests
5
6 def getImage(URL):
7     options = Options()
8     options.add_argument('--headless')
9     driver = webdriver.Chrome(options=options)
10    driver.get(URL) # URLのサイトに接続
11    html = driver.page_source # サイトのソースコードを取得
12    ans = re.findall(r'<img src="(.*?)>.*?>',html) # 正規表現で画像のみを抽出
13    ans = [URL[-1]-i[2:] for i in ans] # 画像を保存するために文字列を整形
14    count = 0
15    for i in ans:
16        img = Image.open(io.BytesIO(requests.get(i).content)).convert('RGB') # バイナリデータを変換して開く
17        for x in range(img.size[0]):
18            for y in range(img.size[1]):
19                r,g,b = img.getpixel((x,y)) # 画像の高さと幅から全面高アクセスを行う
20                if r == 255 and g == 255 and b == 255: # 色が白か判定を行う
21                    count += 1
22            else: continue
23    return count
24
25 if __name__ == '__main__':
26     assert getImage('http://127.0.0.1:8080/') == 142
```

図 11. Web の画像の数を求めるプログラム。

4.5 二つの画像から違う箇所の個数を求めよ

この問題は入力値として上下別の画像が結合された画像を元に上下の画像の差分をとり違う箇所の個数を求める問題である。この問題を解くことで画像処理のプログラミングスキルを確認することが可能である。入力画像を図 12、解答ソースコードを図 13 に示す。



図 12. 上下別の画像が結合された画像。

```
1 import cv2 # opencv-pythonモジュール
2 import numpy as np # numpyモジュール
3
4 def diffImage(path):
5     img = cv2.imread(path) # 画像ファイルを読み込む
6     h, w = img.shape[:2] # 画像の高さと幅
7     img1 = img[0:h//2,0:w] # 画像の半分から上部分
8     img2 = img[h//2:h,0:w] # 画像の半分から下部分
9
10    img_diff = cv2.absdiff(img1, img2) # 画像の差分
11
12    gray = cv2.cvtColor(img_diff, cv2.COLOR_RGB2GRAY) # グレースケール化
13    contours, _ = cv2.findContours(gray, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
14    ans = len(contours)
15    return ans
16
17 if __name__ == '__main__':
18     assert diffImage('./user/files/lena3083030.png') == 2
```

図 13. 画像の差分を取るプログラム。

5 おわりに

本研究では、画像処理、Web スクレイピングなどのプログラミングスキルを確認できる画像を題材とした競技プログラミング用コンテンツの開発を行った。また、コンテンツを扱うために競技プログラミングシステムの設計・構築を行った。

コンテンツでは、画像を題材とすることでより実践的なコンテンツを扱うことが可能である。また、プログラミング言語を Python に限定することで幅広い分野でのコンテンツの作成を行うことが可能である。

競技プログラミングシステムでは、コンテスト開催機能、制限時間機能、回答機能、ランキング機能の実装を行った。各機能は Python 学習サイトの拡張であり、問題や正答率などといったデータベースは学習サイトと共通のものを使用している。そのため、今後の学習サイトとの連携をスムーズに行うことが可能である。

今後の展望として、サイトを運用し競技プログラミングサイトと学習サイト双方のデータを集め、それを元に競技時間の設定や問題の選定を行うことでより発展したサイトになることを期待する。

参考文献

- [1] 「AtCoder」<https://atcoder.jp>, (参照 2021年3月17日)
- [2] 「Aizu Online Judge」,<https://judge.u-aizu.ac.jp/onlinejudge/>, (参照 2021年3月17日)
- [3] 「The Top Programming Languages 2019」,<https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019>, (参照 2020年9月19日)
- [4] 尾有栄光, 平塚太一, 松本貴裕, 高見友幸, 「Python 競技プログラミングサイトの設計と開発」, 第8回 国際 ICT 利用研究 研究会講演論文集, pp.24-27, 2020.