

以下、for(1~1000) で回していた所。⇒入力画像の部分を引数にとって1枚1枚でできるようにする。

App.py の中の form のところに新しい関数を作って、  
llah.create\_db(Cyan,./templates/IMG/Cyan.jpg)  
llah.create\_db(Magenta,./templates/IMG/Magenta.jpg)  
llah.create\_db(Yellow,./templates/IMG/Yellow.jpg)  
という風にして、従来の llah の中の create\_db の機能呼び出す。

```
#include <opencv2/opencv.hpp>
```

```
#include <llah/llah.hpp>
```

読み込むライブラリとしては、opencv2(すでに app.py で実装中になっている。)

Llah/llah.hpp の部分に関しては作ってくれたクラス構造を一度に import できるようにしてもらう。

```
llah::Database dbCyan((256)*5 + 27, 7, 6, 4);
```

```
llah::Database dbMagenta((256)*5 + 27, 7, 6, 4);
```

```
llah::Database dbYellow((256)*5 + 27, 7, 6, 4);
```

ここで、作成した LLAH クラスの Database のインスタンスを生成している。

```
(Database::Database(size_t hashTableSize, unsigned int n, unsigned int m,  
unsigned int l) {
```

```
    size_t elementSize = sizeof(double) * 2 + sizeof(DocumentID) +  
sizeof(unsigned long long);
```

```
    _hashTable = new HashTable(elementSize, hashTableSize);
```

```
    _descriptorExtractor = new llah::DescriptorExtractor(n, m, l);} という構造になっ  
ている。)
```

```
dbCyan.setRemoveRedundancy(false);
```

```
dbCyan.setSampleKeypoints(false);
```

```
dbCyan.setNumberOfKeypointsToSample(200);
```

```
dbCyan.setSampleFeatures(false);
```

```
dbCyan.setNumberOfFeaturesToSample(4000000000);
```

```
dbCyan.setListLengthLimit(50000000000);
```

ここで、LLAH の諸々のパラメーターの設定をしている。

各色の For 文の部分の解析

(1) 0 詰をして、ファイルパスを指定している。

(2) image という画像行列にファイルパスの画像を取り込んでいる。

(3) リサイズしている。

```
(4) vector<llah::Keypoint> keypoints;
```

```
    llah::detectFeatures(image, keypoints);
```

特徴点の抽出をしている。

```
(5) vector<cv::Rect> rects = llah::dividedRects(image, 4, 2);
```

```
    dbCyan.addDocument(keypoints, i, &rects);
```

ここでは、特徴点の行列、画像 id 番号、&rects この引数はサンプリングのために、各長方形領域の決められた個数だけ取るための指定。

(6) keypoints.size が特徴点の個数としてコマンドに出力されている。

