

Retrieve.cpp の python 実装

App.py の中に新しい関数と行列 search[9] を作って、

```
search[0~2] = llah.retrieve(firstsecondthird のどれか,Cyan,./templates/IMG/  
Cyan.jpg)
```

```
search[3~5] = llah.retrieve(firstsecondthird のどれか,Magenta,./templates/IMG/  
Magenta.jpg)
```

```
search[6~8] = llah.retrieve(firstsecondthird のどれか,Yellow,./templates/IMG/  
Yellow.jpg)
```

という風にして (もっと効率の良い方法があればそちらに変える。)、従来の llah の中の retrieve の機能呼び出す。

その引数を使って、App.py の中に実装した AKAZE のコードを for 文を使って実行する。

create_db の時と同様に以下の文を for 分で実行する。

(1) 0 語をして、ファイルのパスをしてする。

(2) imread でファイルを読み込む

(3) リサイズする。

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

ここで、特徴点の行列を抽出している。

(5) llah::DocumentID CyandocumentID = -1; unsigned int Cyanscore = 0; この後使う行列を宣言している。

```
(6) CyandocumentID = dbCyan.findDocument(keypoints, nullptr, &Cyanscore ,  
0); ここで特徴点、射影変換行列 (空)、スコアを入れる行列を引数として候補画像を検索  
している。(同時に時間も計測している。)
```