**고급소프트웨어실습**

**중간 리포트**

**(CSE 4152)**

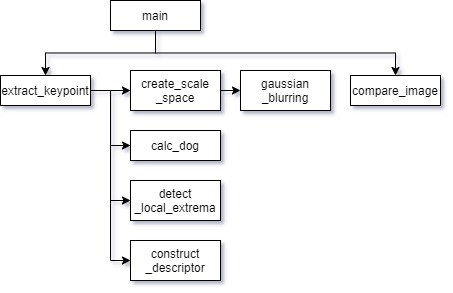
**Due: 2020년, 10/19일, 23:59분**

**학번 : 20171678**

**반 번호 : 2반**

**이름 : 이찬희**

**1. 이미지 검색 프로그램의 구현에 필요한 주요 함수들의 관계도 그래프를 자유로운 형식으로 그리고, 각 함수들의 역할에 대하여 설명하시오. (이미지는 모두 로컬 하드 드라이브에 저장되어 있는 것으로 가정. 함수의 개수는 주요 기능 위주로 5개 이상 10개 이하로 선정할 것)**

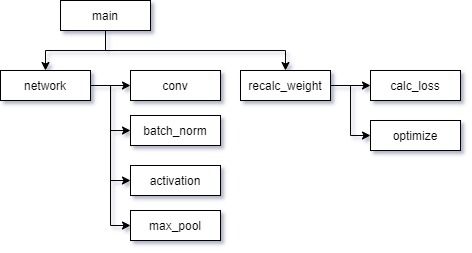


이미지 검색 프로그램에서는 찾고자 하는 이미지와 대상이 되는 이미지의 특징점 검출과 이 두 이미지의 특징점들을 비교하는 절차가 필요하므로 필요한 주요 함수들은 위와 같다.

각 함수의 역할은 다음과 같다.

* **main:** 입력과 출력을 담당한다. 검색할 이미지와 대상이 되는 이미지를 입력 받고 extract\_keypoint를 사용하여 특징점을 검출, compare\_image를 호출하여 특징점간의 비교를 하여 대상이 되는 이미지에서 검색할 이미지를 포함하는 이미지를 찾아낸다.
* **extract\_keypoin**t**:** 함수create\_scale\_space, calc\_dog, detect\_local\_extrema, construct\_descriptor를 차례대로 호출하여 입력받은 이미지의 특징점을 검출한다.
* **create\_scale\_space:** 입력된 이미지로부터 여러 개의 스케일로 표현되는 이미지를 생성한다. 이때, gaussian\_blurring 함수를 호출하여 가우시안 흐림을 수행하는 방식으로 스케일 변화 효과를 달성한다.
* **gaussian\_blurring:** 이미지에 가우시안 흐림 필터를 적용한다.
* **calc\_dog:** Scale 공간에서 픽셀값의 변화량을 찾기 위해 같은 ocatave 안에 인접한 scale 이미지간의 차이(Difference of Gaussian)을 계산한다.
* **detect\_local\_extrema:** 스케일 공간에서 각 픽셀을 중심으로 인접한 26개 픽셀의 극대값과 극소값을 뽑아 Dog로부터 특징점을 찾는다.
* **construct\_descriptor:** 특징점 주변의 gradient를 계산하여 이를 특정 길이의 디스크립터로 표현한다.
* **compare\_image:** 두 이미지의 특징점들을 비교하여 찾고자 하는 이미지가 대상 이미지에 존재하는지를 확인한다.

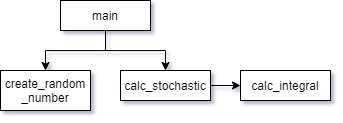
**2. 얼굴 인식 프로그램의 구현에 필요한 주요 함수들의 관계도 그래프를 자유로운 형식으로 그리고, 각 함수들의 역할에 대하여 설명하시오. (이미지는 모두 로컬 하드 드라이브에 저장되어 있는 것으로 가정. 함수의 개수는 주요 기능 위주로 5개 이상 10개 이하로 선정할 것)**



충분히 많은 라벨링된(사진에 얼굴이 존재함/존재하지 않음) 이미지 데이터를 가지고 있다고 가정하고, 이를 이용하여 ResNet을 training 시켜 검사할 이미지에서 얼굴이 존재하는지 여부를 찾는 프로그램을 만들고자 한다면 이미지를 Network에 통과시키고, weight 값을 다시 계산하는 함수가 필요하며 이를 도식으로 나타내면 위와 같다.

* **main:** 입력과 출력을 담당한다.
* **network:** 입력된 이미지를 네트워크에 통과시킨다. conv, batch\_norm, activation, max\_pool을 적절히 호출하여 image를 라벨링하고 그 결과를 반환한다.
* **conv:** 입력된 이미지를 주어진 필터와 convolution 한다.
* **batch\_norm:** convolution 결과를 normalize 한다.
* **activation:** 입력받은 normalized 데이터가 다음 단계에서 활성화 되는지를 결정한다.
* **max\_pool:** input data의 크기를 줄이기 위해 최댓값을 뽑아내는 max pooling을 수행한다.
* **recalc\_weight:** network 결과를 보고 calc\_loss 와 optimize를 호출하여 가중치를 다시 계산한다.
* **calc\_loss:** network를 통해 도출된 결과와 실제 label 값을 비교하여 loss를 측정한다.
* **optimize:** loss 값을 최적화하는 방향으로 가중치를 최적화한다.

**3. 주어진 확률 밀도 함수 p(x)를 통계적으로 따르는 stochastic process, X0, X1, X2, ... 를 생성해주는 소프트웨어를 설계한 후, 구현에 필요한 주요 함수들의 관계도 그래프를 자유로운 양식으로 그리고, 각 함수들의 역할에 대하여 설명하시오.**



확률 밀도 함수 p(x)를 통계적으로 따르는 stochastic process Xi는 구간 (0,1)에 존재하는 난수 Ui에 대하여 를 만족한다. 따라서, stochastic는 를 만족하는 x이다. 이를 계산하기 위해서는 0과 1 사이의 난수를 생성하고 이를 사용하여 stochastic을 계산하는 함수들이 필요하며, 이를 도식으로 나타내면 위와 같다.

각 함수의 역할은 다음과 같다.

* **main:** 입력과 출력을 담당한다. create\_random\_number 와 calc\_stochastic을 m번 호출하여 stochastic m개를 생성한다.
* **create\_random\_number:** 0과 1 사이의 난수를 생성한다.
* **calc\_stochastic:** 생성된 난수를 이용하여 의 근 x를 구하는 방식으로 주어진 확률 밀도함수를 따르는 stochastic을 계산한다. 이 때 는 비선형 방정식이므로, 이 방정식의 풀이를 위해 Newton-Rapson 방법 (이 때의 초기값은 가장 근으로 등장할 확률이 높은, p(x)의 값이 최대인 x를 고르는 것이 합당하다.). Secant 방법（이 때의 초기값은 근을 반드시 x0와 xn 사이에 있으므로 그 둘로 주는 것이 합당하다.) 그리고 Bisection 방법 (이 때 는 반드시 0과 1 사이의 값을 가지므로 는 와 1 사이의 의 값을 가지고 이 때의 x 값은 x0와 xn이므로 초기값 또한 그 둘로 주는 것이 합당하다.) 는 calc\_integral을 호출하여 구한다.
* **calc\_integral:** 주어진 x에 대하여를 계산한다.