숙제 1번

제출기한 9월 29일 24시까지. HW1\_가반\_이름\_학번\_시간.zip 형식으로 [dongsunghw@gmail.com](mailto:dongsunghw@gmail.com) 과 [dongsung@ssu.ac.kr](mailto:dongsung@ssu.ac.kr) 로 제출하세요.(나반인 학생은 나반으로 하시고요. 다른 반으로 조교에게 forwarding되면 곤란하겠죠. 시간은 다음 예제처럼 2018-09-29-0700 의 형태로 표기하세요) 특별히 이메일 제목을 위의 형식으로 작성해서 보내주세요. 제발! spam filtering 되어서 사라지지 않게요. 보낼때는 vs 2017 기준으로 debug에 관련된 파일은 제거하고 zip해서 보내주세요.(아마 100k 넘으면 이상한 것 넣어진거예요)

목적: recall 과 precision 을 계산하는 프로그램을 작성하자.

<https://en.wikipedia.org/wiki/Precision_and_recall>

알고리즘 설명: 영상인증은 원래 영상과 전송되어진 영상사이에 변형이 없는지를 살피는 것이다. 그 변형감지 하는 방법이 A방법과 B방법이 있는데 그 방법의 성능을 분석하고자 한다. 성능분석을 위해서 약 15개의 영상을 사용했다. 먼저 한 개의 영상을 읽고 A 방법의 결과를 파일에 적고 그 다음 B이 결과를 적었다. 그 다음에 계속해서 나머지 영상에서 처리했다. 각 방법의 결과는 전송전 원영상에서 검출된 feature의 개수(nFeatureOriginal), 전송후 채널오류등으로 변경된 영상에서 검출된 feature의 개수(nFeatureReceived), 그리고 공통으로 검출된 feature의 개수(nFeatureCommnon) 이다.

영상1, nFO1\_A, nFR1\_A, nFC1\_A,

영상1, nFO1\_B, nFR1\_B, nFC1\_B,

영상2, nFO2\_A, nFR2\_A, nFC2\_A,

영상2, nFO2\_B, nFR2\_B, nFC2\_B,

....

한 개의 영상에서 각 방법에서 recall = nFC/nFO 그리고 precision = nFC/nFR 로 정의하자.

각 문제는 모두 함수로 구성해야 한다.

1. 15개의 영상의 데이터를 array에 읽어들이고 화면에 출력한다.(1점)
2. 방법 A와 B에 대해서 recall값과 precision 값을 출력한다.(1점)
3. 그리고 recall과 precision의 평균값을 출력한다. (1점)
4. 영상의 개수가 n 개가 되고 방법의 개수가 m 개가 됐을때도 수행될 수 있게 한다(2점). 단 n과 m은 파일의 가장 앞부분에 적혀있다. 이의 검증을 위해서 n이 17개 m 이 3인파일도 수행해서 결과를 보인다.. -> 동적 메모리

결과 파일은 2개 (15/2) (17/3)

Input file… (input file(data file)을 각자 만든다.)

그 파일을 base로 해서 main함수와 sub함수를 만든다.

Tip) A 방법이 좋은지, B 방법이 좋은지의 일관성이 있게끔 data 파일을 만든다.

< cin은 없다고 생각한다. 파일을 입력으로 받아서 처리하면 된다. >

Ex)

17 3

Summervac.jpg 1000 970 900 (영상 1. 방법 a)

Summervac.jpg 700 800 650 (영상 1. 방법 b)

Summervac.jpg 495 203 495 (영상 1. 방법 c)

Summervac2.jpg 385 493 102 (영상 2. 방법 a)

….

…

…

문제를 풀기 전에, Design을 한다.

몇 시간 고민한 뒤 modeling을 한다

Recall이 1이 되면 제일 좋음, 근데, false positive도 많아지면..?

* Precision으로 판별!

둘다 1이 되는게 제일 좋겠지?

풀이 (영상 인증 – 정밀도와 재현율) (image authentification)

첫번째 함수

1. 입력: 파일 이름
2. 출력: int nImage, nMethod;
3. String str\_Filename[];  
   int pnfO[], pnfR[], pnfc[];

두번째 함수

입력 : 첫번째 함수의 출력

출력 : double pRecall[], pPrecision[];

세번째 함수

입력; nImage, nMethod, pRecall[], pPrecision[]

출력: AvgRecall[nMethod(=방법의 개수)], AvgPrecision[nMethod];