생소할 수 있는 문법들에 대해 간단히 설명 드리겠습니다.

vector<….> 변수 : c++ 기본 형식이며, 동적 배열과 같은 의미를 지닌다.(기하학적 벡터와 의미와 쓰임새가 다름!) 초기 크기는 0이며, 추후 데이터를 추가할 때 배열과 다르게 크기 재할당이 필요 없어 자유롭다. <iostream>에 선언되어 있으며 using namespace std; 선언 후 호출가능하다.

vector를 이용한 표현 간소화의 예시

float value[ ]; value = new float[size];

=>vector<float> value ;

float value[ ][ ]; \*value = float[size]; for(){ value[i] = new float[size]};

=> vector<vector<float>> value;

배열은 불편한 접근성과 크기 고정성 때문에 잘 쓰이지 않는다. 배열 대체의 대안으로써 vector 가 쓰이고 있으며, 비슷한 성격의 형식은 deque(큐)가 있다.

벡터로의 데이터 추가는 value.push\_back(float데이터); 의 과정으로 이루어진다. 추가된 데이터는 배열과 똑같이 value[ i ] 형식의 문법으로 접근 가능하다.

Mat 변수 : opencv에서 행렬, 벡터, 이미지를 처리하는 통합된 클래스이며, using namespace cv 선언 후 호출 가능합니다. 벡터 또한 표현할 수 있습니다.

Mat mat : 일반 행렬 선언

행렬은 vector 와 비슷한 구조로 이루어져 있다. 이차원 동적 배열과 성격이 비슷하며, push\_back()을 통해 column을 추가할 수 있다.

deque<Mat> mat : 행렬의 큐

vector<Mat> mat : 행렬의 동적 배열