

숫자 검출 및 인식

-컴퓨터 비전 미니 프로젝트



컴퓨터공학부 안희성

INDEX

1. 프로젝트 정의

2. 프로젝트 진행 과정

알고리즘 절차

전처리

인식

3. 프로젝트 결과 및 보완점

4. 프로젝트 계획서

5. 프로젝트 개발환경

6. 참고 자료

7. QnA

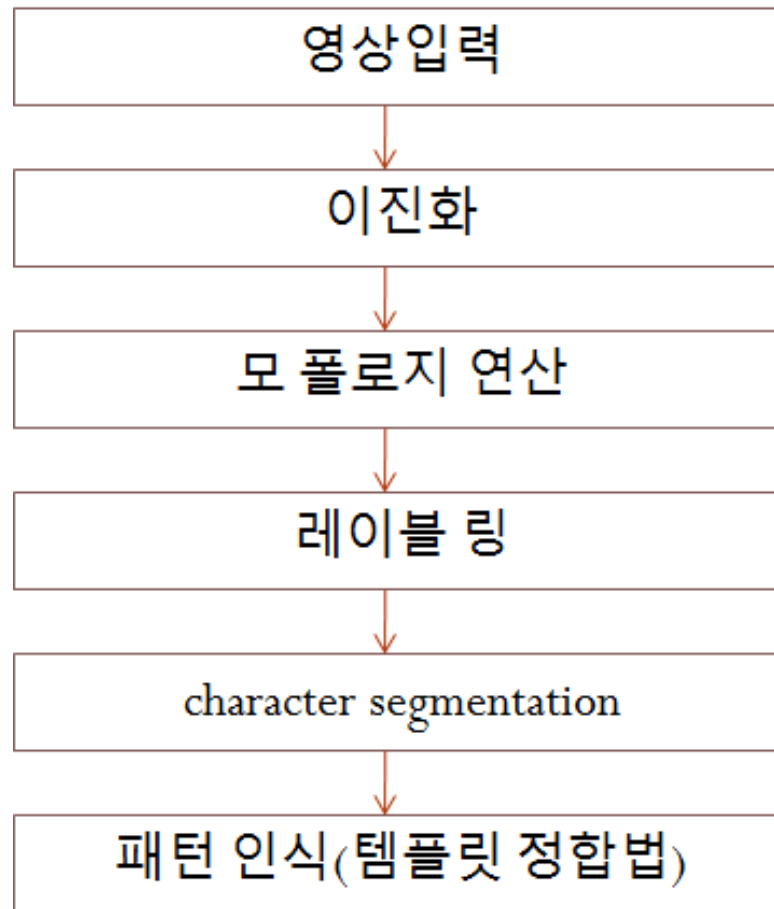
프로젝트 정의

프로젝트 정의

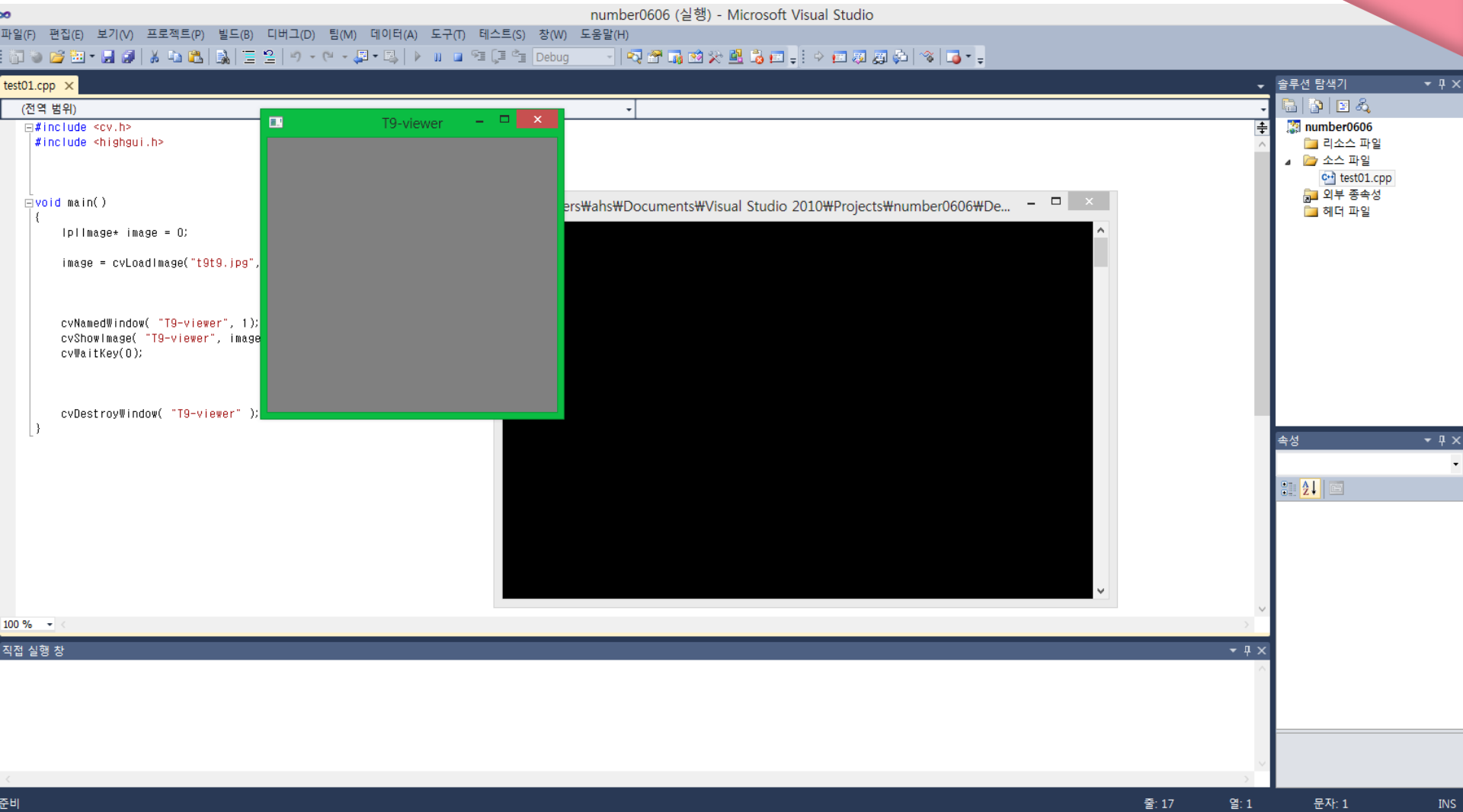
Open CV와 MFC를 사용해서 패턴인식의 한 분야인 숫자 검출 및 인식 프로그램을 인쇄되거나 손으로 쓴 문자를 판독·식별하여 컴퓨터가 이해할 수 있는 코드로 전환할 수 있게 만들어봄으로써 영상처리 기법을 이해하고 배워 본다.



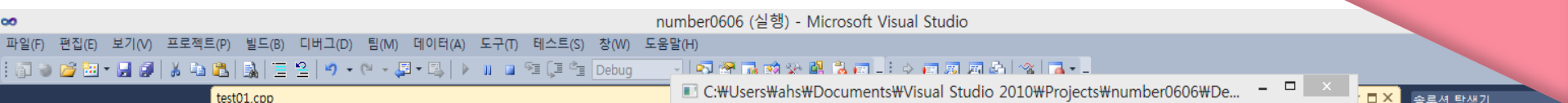
알고리즘 절차



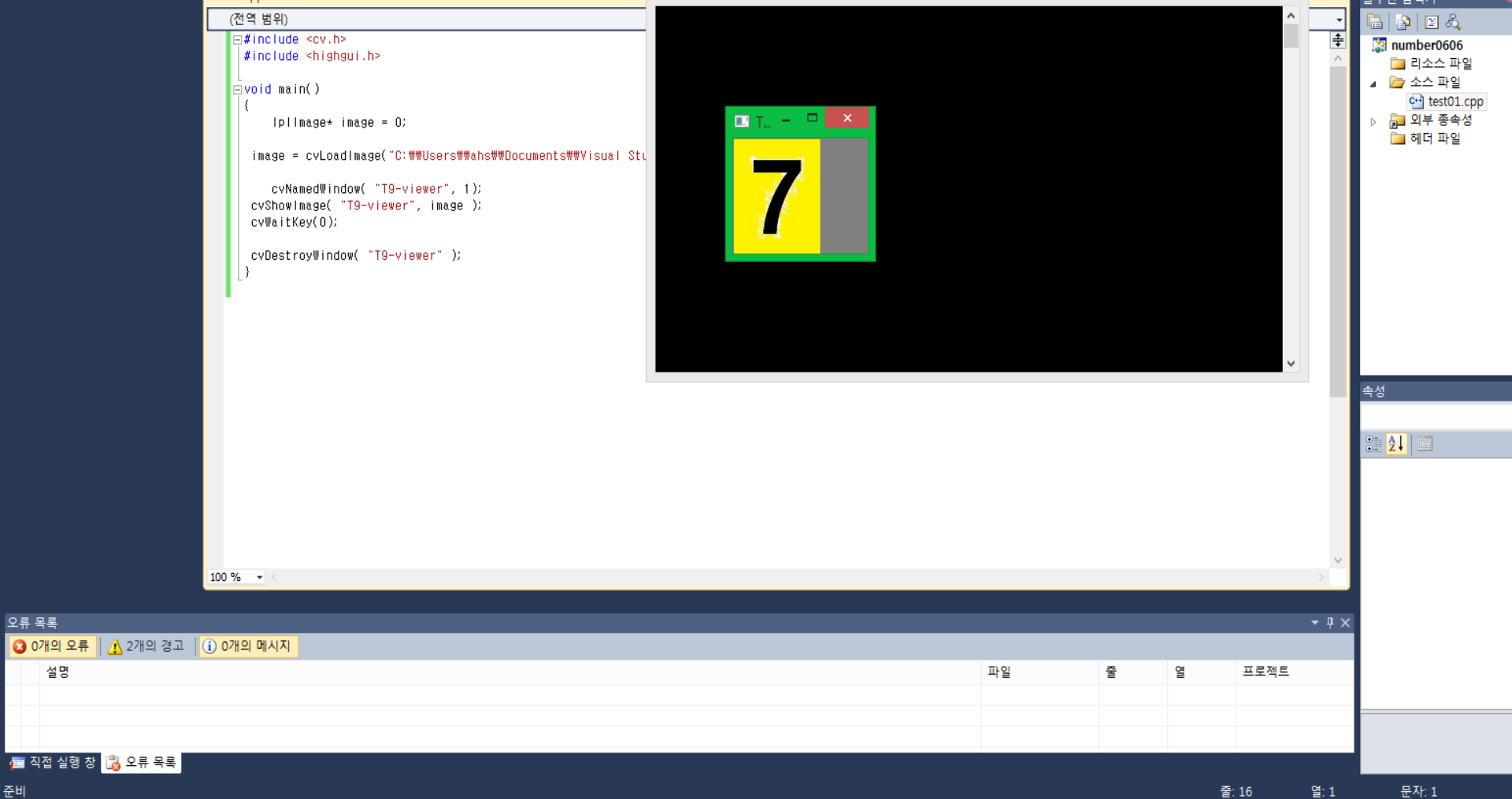
전처리



전처리



The screenshot shows the Microsoft Visual Studio 2010 environment. The title bar indicates the project is 'number0606 (실행) - Microsoft Visual Studio'. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '프로젝트(P)', '빌드(B)', '디버그(D)', '팀(M)', '데이터(A)', '도구(T)', '테스트(S)', '창(W)', and '도움말(H)'. The toolbar shows various icons for file operations, editing, and debugging. The 'Debug' window is active, showing a list of files including 'test01.cpp'. The status bar at the bottom indicates the current file is 'C:\Users\Wahs\Documents\Visual Studio 2010\Projects\Wnumber0606\De...'. The taskbar on the right shows the '솔루션 탐색기' (Solution Explorer) icon.



전처리

```
#include <cv.h>
#include <highgui.h>

void main()
{
    IplImage* image = 0;          //영상을 위한 IplImage 포인터 변수 선언.

    image = cvLoadImage("C:\\Users\\ahs\\Documents\\Visual Studio 2010\\Projects\\number0606\\qw.jpg", 1); //내컴퓨터에 저장된 이미지 불러오기

    cvNamedWindow( "T9-viewer", 1); //윈도우 캡션이름 "T9-viewer" 1플래그로 윈도우 생성
    cvShowImage( "T9-viewer", image ); //cvloadimage 함수로 읽어온 영상을 t9-viewer에 보여줌
    cvWaitKey(0); //지연 값 0 이므로 사용자가 아무 키보드나 누를 때까지 멈추고 기다림

    cvDestroyWindow( "T9-viewer" ); //cvnamedWindow 함수에 의해 생성된 "T9-viewer" 윈도우 파괴
}
```

- ① Gray스케일의 1채널 이미지로 변환
- ② 임계 값을 기준으로 임계 값을 초과하는 픽셀과 임계 값 이하의 픽셀을 구분
- ③ 단순 임계치 방법, 평균 이진화, 반복 이진화, 적응 이진화, OTSU 등이 있다



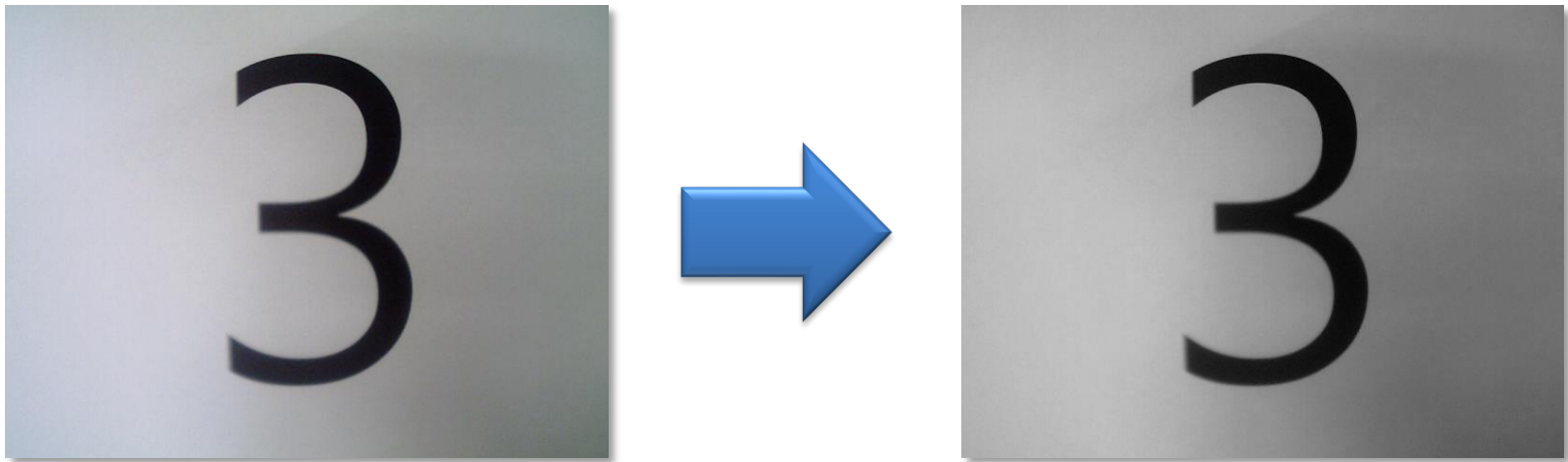
원영상
OSTU



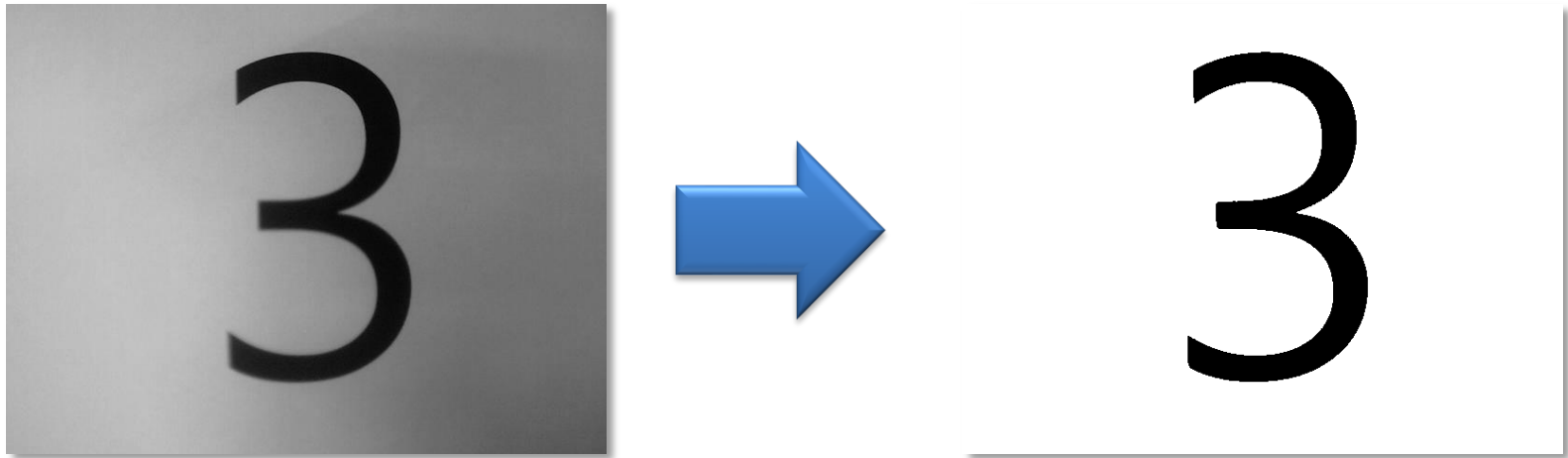
그레이 스케일



단순 임계치



- Cam에서 입력 받은 이미지를 RGB 컬러공간에서 GRAY 컬러공간으로 변환



- GRAY 이미지에 Threshold를 적용

이진화

전처리

number0606 (실행) - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 데이터(A) 도구(T) 테스트(S) 창(W) 도움말(H)

Debug

test01.cpp

```
(전역 범위)
#include <cv.h>
#include <highgui.h>

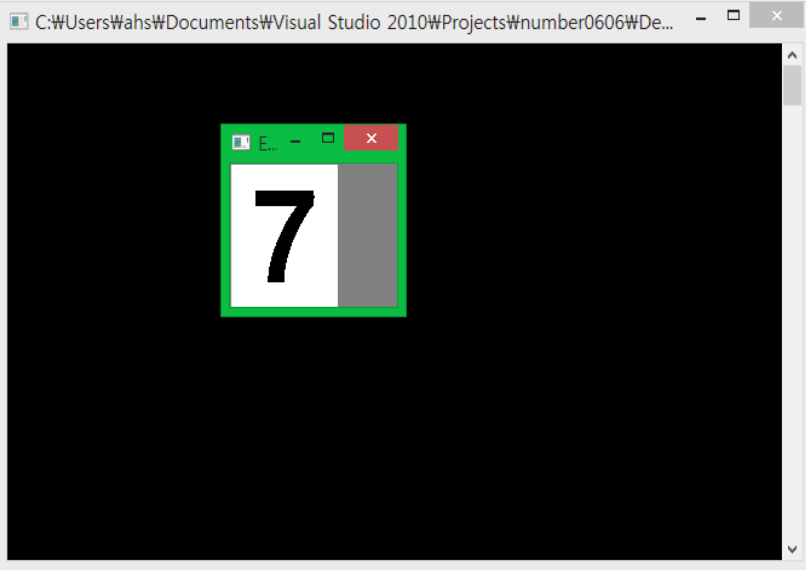
int main(int argc, char** argv)
{
    IplImage* img = cvLoadImage("C:\\Users\\Wahs\\Documents\\Visual Studio 2010\\Projects\\number0606\\De...");

    int threshold = 128;
    cvNamedWindow("Example1", CV_WINDOW_AUTOSIZE);
    IplImage *gray = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
    IplImage *output = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
    cvCvtColor(img, gray, CV_RGB2GRAY);
    cvThreshold(gray, output, threshold, 255, CV_THRESH_BINARY);
    cvShowImage("Example1", output);
    cvWaitKey(0);

    cvReleaseImage(&img);
    cvReleaseImage(&gray);
    cvReleaseImage(&output);
    cvDestroyWindow("Example1");

    return 0;
}
```

C:\\Users\\Wahs\\Documents\\Visual Studio 2010\\Projects\\number0606\\De...



솔루션 탐색기

- number0606
 - 리소스 파일
 - 소스 파일
 - test01.cpp
 - 외부 종속성
 - 헤더 파일

속성

오류 목록

0개의 오류 2개의 경고 0개의 메시지

설명	파일	줄	범위	프로젝트

직접 실행 중 오류 목록

준비

줄: 27 열: 1 문자: 1 INS

오전 5:20

전처리

```
#include <cv.h>
#include <highgui.h>

int main(int argc, char** argv)
{
    IplImage* img = cvLoadImage("C:\\Users\\wahs\\Documents\\Visual Studio 2010\\Projects\\number0606\\qw.jpg", 1);

    int threshold = 128; //임계값설정
    cvNamedWindow("Example1", CV_WINDOW_AUTOSIZE); // 윈도우 캡션이름 "Example1" CV_WINDOW_AUTOSIZE로 윈도우생성
    IplImage *gray = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1); //영상크기가 img인, 영상에서 화소의 비트수를 unsigned 8비트, 1개의 채널로 생성하여 *gray에 저장
    IplImage *output = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1); //영상크기가 img인, 영상에서 화소의 비트수를 unsigned 8비트, 1개의 채널로 생성하여 *output에 저장
    cvCvtColor(img, gray, CV_RGB2GRAY);
    cvThreshold(gray, output, threshold, 255, CV_THRESH_BINARY); //cvThreshold함수의 반환값은 인수로 제공한 임계값이 128
    cvShowImage("Example1", output); // cvLoadImage함수로 읽어온 이미지 "Example1"에 보여줌
    cvWaitKey(0);

    cvReleaseImage(&img); //할당된 메모리 제거
    cvReleaseImage(&gray); //할당된 메모리 제거
    cvReleaseImage(&output); //할당된 메모리 제거
    cvDestroyWindow("Example1"); // 생성된 "Example1"윈도우 파괴

    return 0;
}
```

모폴로지 연산

① 침식(erosion) 연산

이진영상에서는 객체영역을 흰색으로 볼 때 이들 영역이 줄고 검은색 영역이 늘어나며, 그레이 영상(또는 컬러영상)에서는 밝은 영역이 줄고 어두운 영역이 늘어나는 쪽으로 연산 결과가 변한다.

② 팽창(dilation) 연산

침식, 팽창연산은 잡영을 제거하는데 주로 사용된다.

③ 열기(opening) 연산

침식연산을 먼저 수행하고 팽창연산을 수행하여 잡영들을 제거할 수 있다.

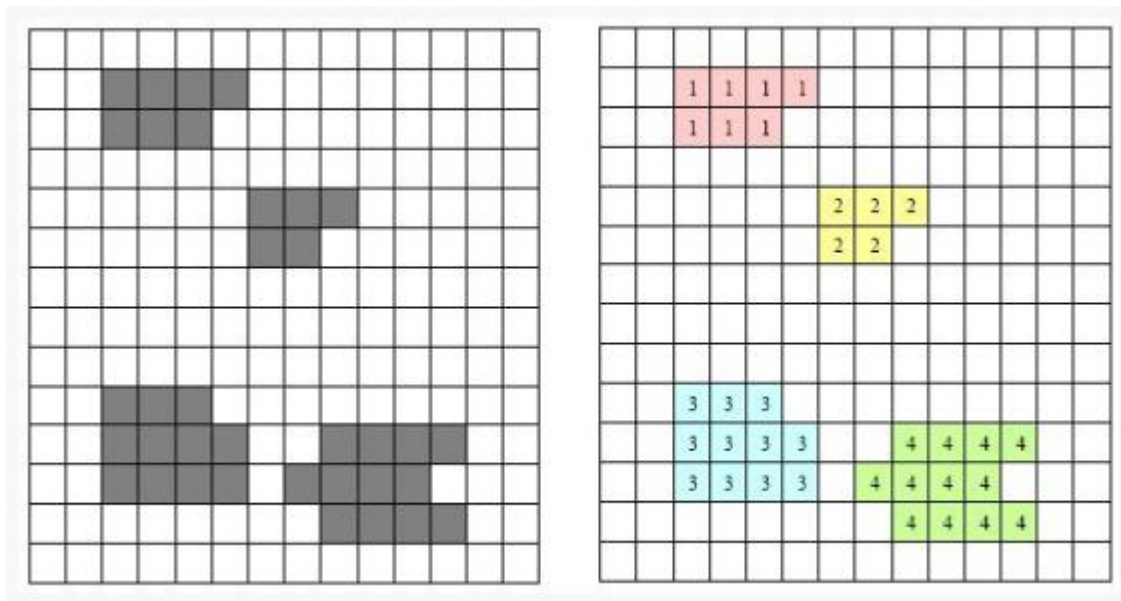
④ 닫기(closing) 연산

팽창연산을 먼저 수행하고 침식연산을 수행하여 구멍들을 제거할 수 있다.

⑤ Top-Hat (Black-Hat) 연산

이와 같이 열림 연산(또는 닫힘 연산)을 수행한 후 원본 영상과 열림 연산(또는 닫힘 연산)의 결과의 차이를 구하면 영상의 텍스처 정보(또는 윤곽선 정보)를 얻을 수 있다.

인접한 화소에 모두 같은 번호(Label)를 붙이고 연결되지 않은 다른 성분에는 다른 번호를 붙이는 일
이진화된 영상의 픽셀수의 범위를 정하여 원하는 부분을 얻을 수 있다



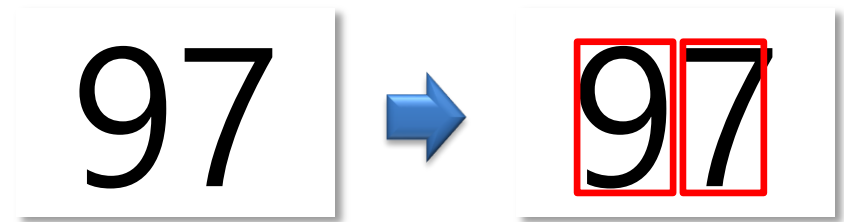
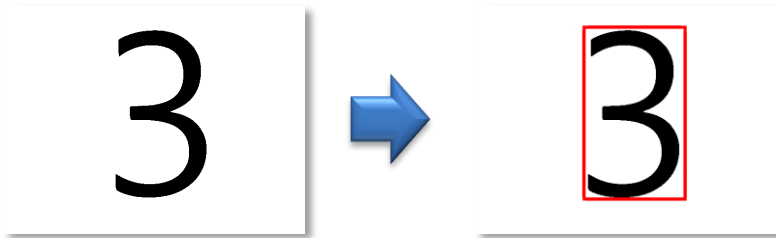
- 모폴로지한 이미지에 레이블링(라벨화)를 적용

character segmentation

인식

숫자를 한 단위로 분할하는 과정

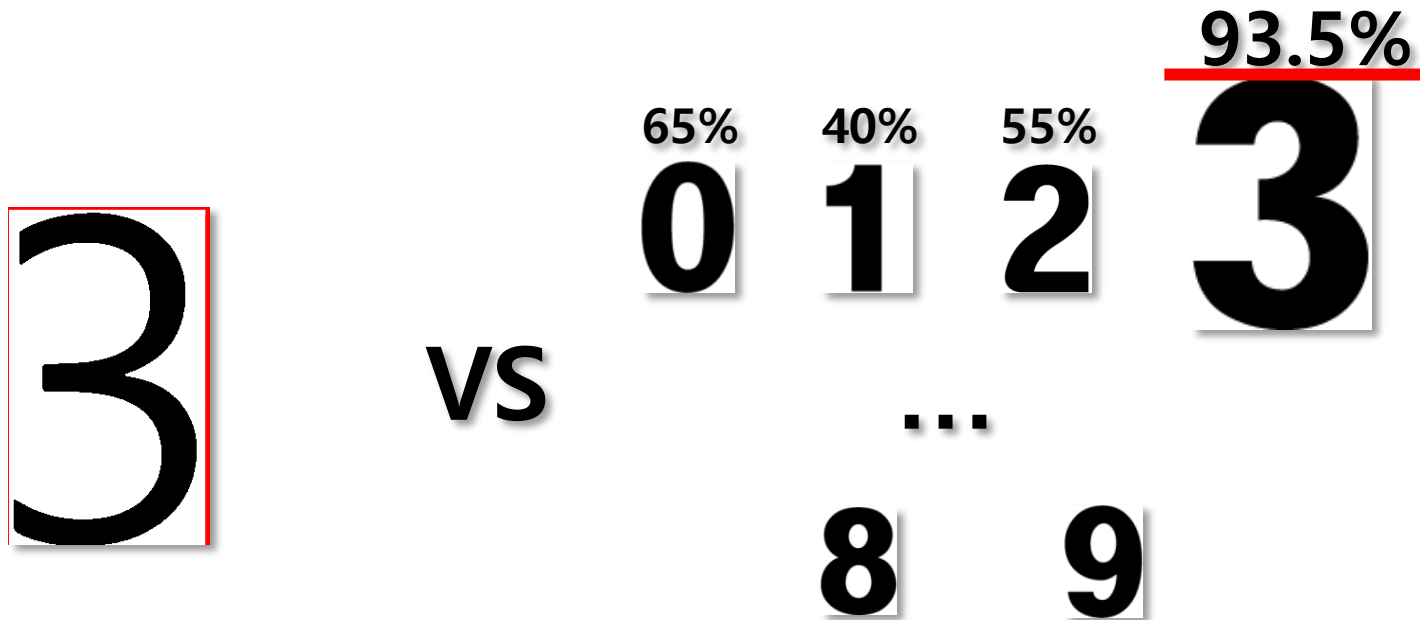
숫자(문자)인식은 한 글자 단위로 인식 하기 때문에 한 글자 영역을 구분해 낸다.



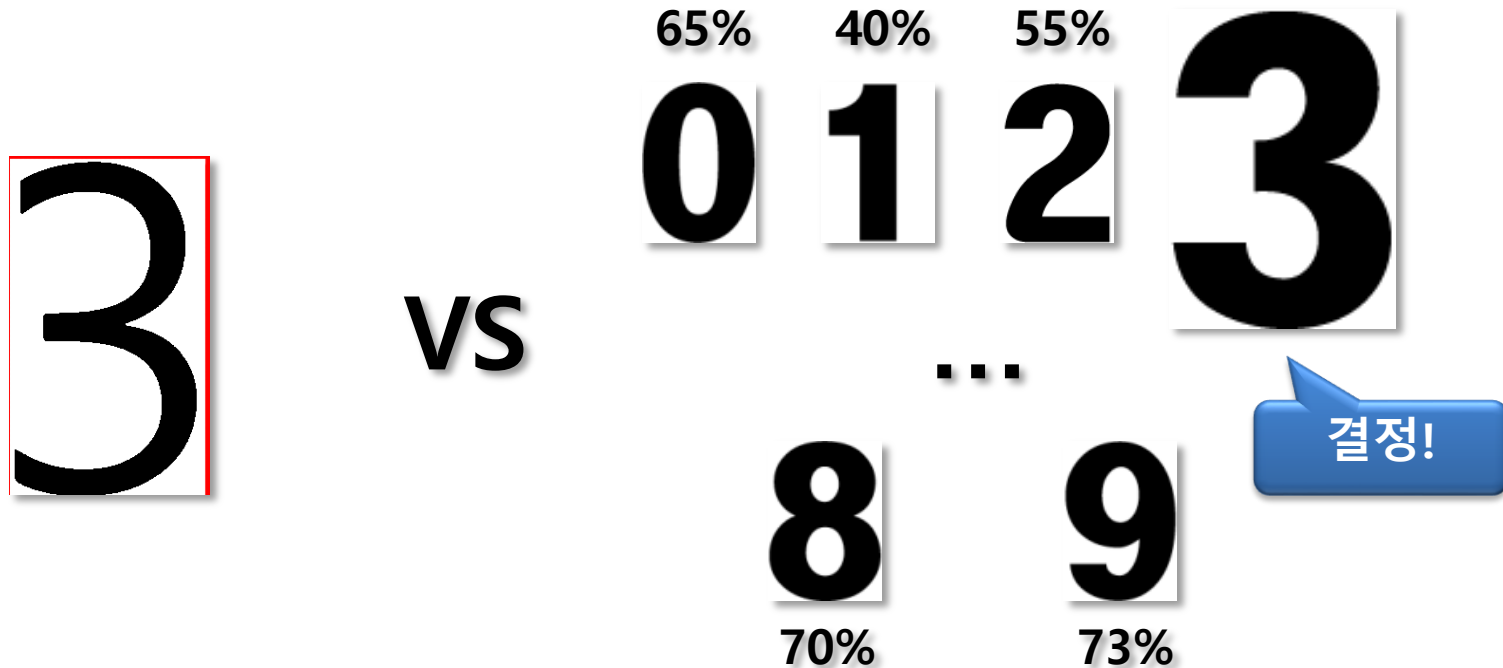
- 이미지에서 숫자 영역만 추출

- 이미지에서 한 단위로 분할

표준 패턴과의 비교를 통해 일치도, 유사도가 가장 높은 표준 패턴을 찾는다

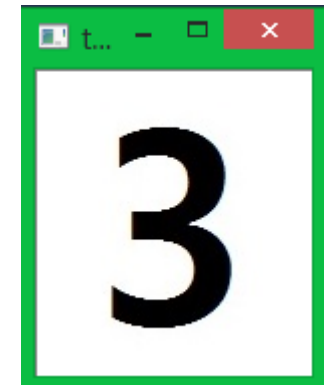
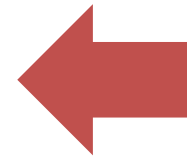
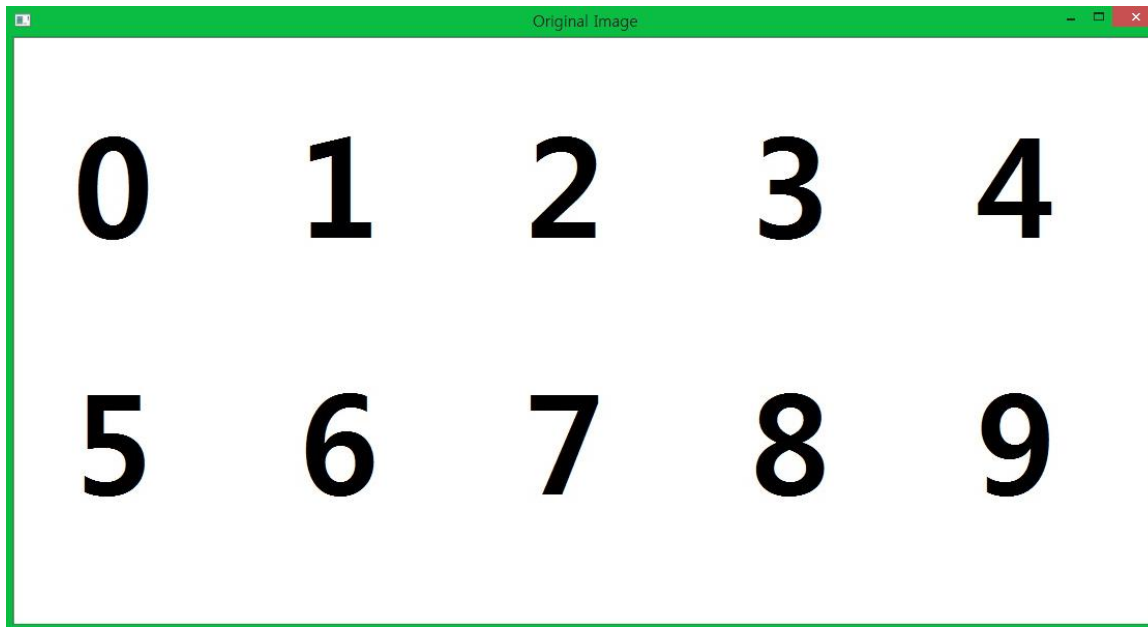


- 입력 영상에서 추출한 숫자 이미지를 기본 숫자 이미지들과 하나씩 비교
- 비교 결과를 토대로 가장 적합한 부분 인식



- 비교 결과를 토대로 가장 적합한 이미지를 결정

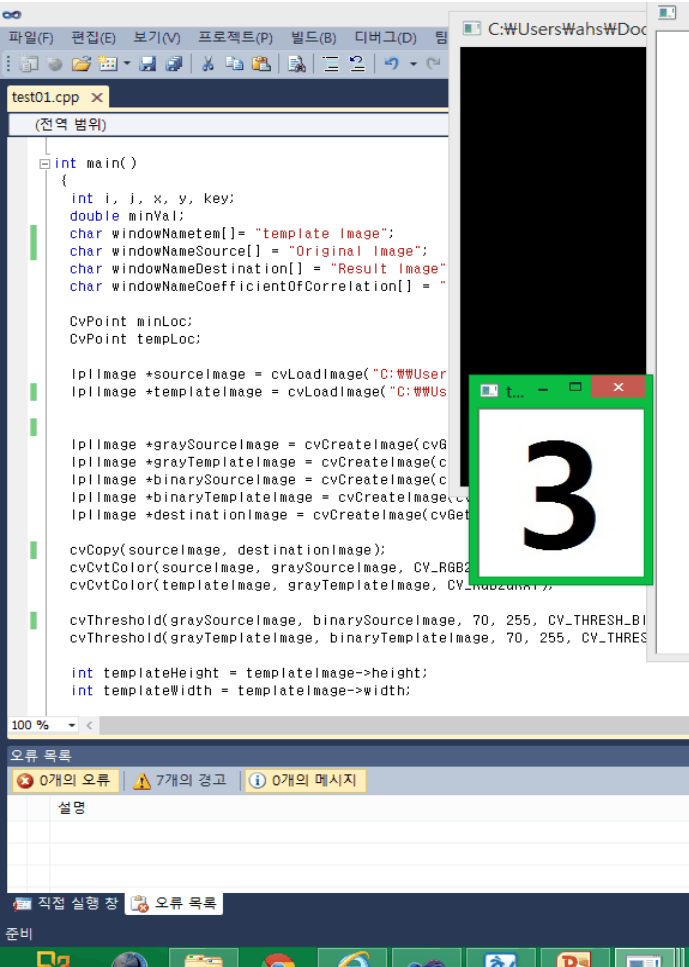
패턴 정합법(추가사항)



전체 0-9 숫자 이미지 파일(original image)와 숫자 이미지(3,template image)를 이진화 시켜서 original image에 template image를 비교하여 93% 이상 유사한 부분을 인식합니다.

패턴 정합법(추가사항)

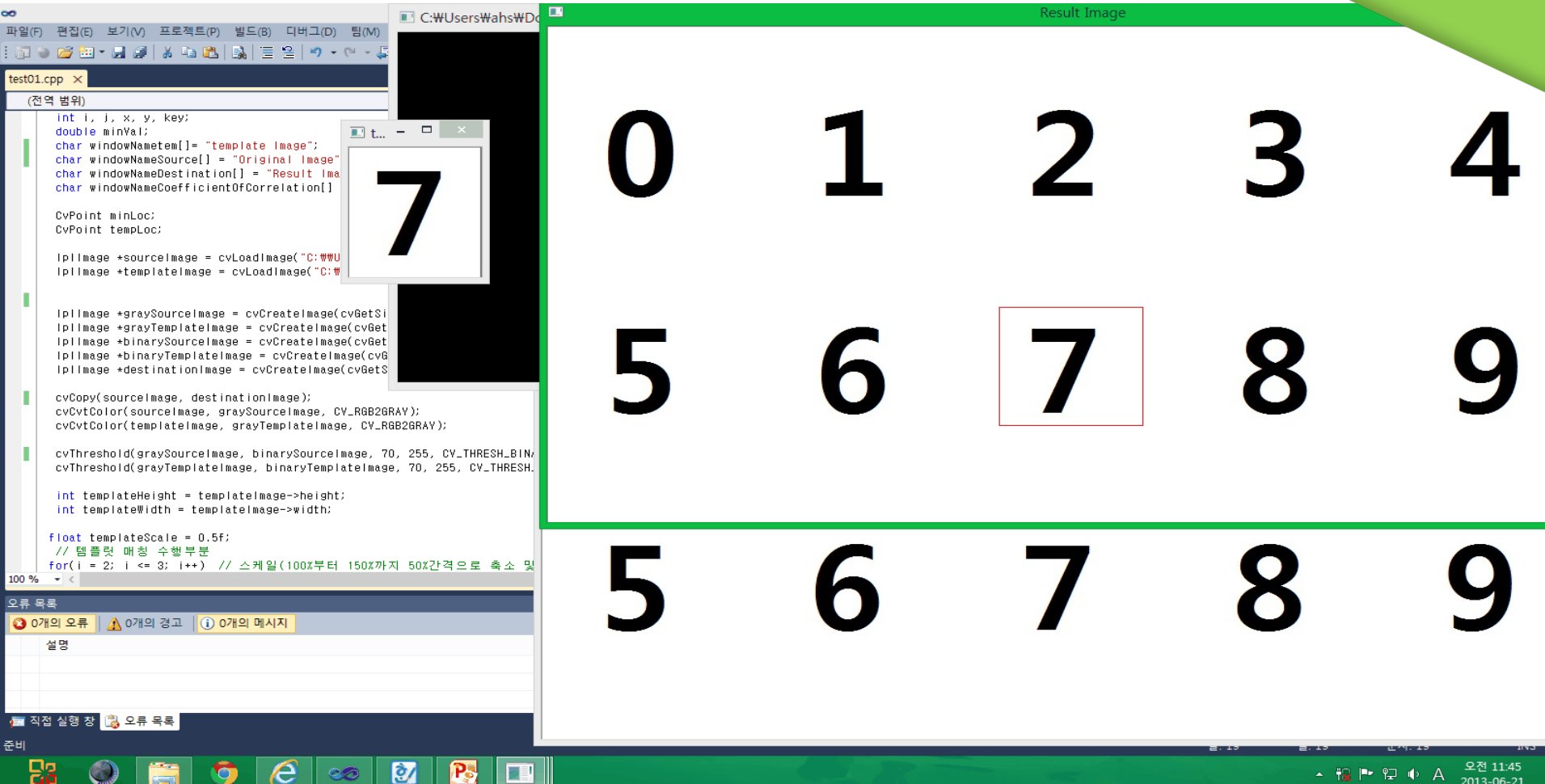
인식



숫자 “3” 의 이미지를 인식

패턴 정합법(추가사항)

인식



숫자 “7”의 이미지를 인식

- ✧ 오류 해결 능력 부족 및 Opencv 지식 부족..
- ✧ 언어 및 코딩 학습 필요
- ✧ 영상 처리에 대한 지식 향상



◆ 개발 툴

visual studio 2010

Opencv2.2

cmake2.8.6

◆ 개발 환경

삼성전자 센스 NT550P5C-S85L

운영체제 OS Windows8

프로세서 CPU Intel Core Ivy Bridge i7-3630QM

◆ 참고 자료

1. 기초부터 활용까지 패턴인식과 기계학습 , 박 혜영 이 관용 저
2. visual c++와 openCV로 배우는 디지털 영상처리, 강 동중 하 종은 저
3. Development of an Image Processing Algorithm for the Recognition of Car Types and Number Plate, 서울시립대학교 대학원 제어계측공학과 이 평원 저
4. (openCV로 배우는 디지털 영상처리) openCV programming ,김 동근 저
5. OpenCV KOREA 대한민국 최고의 컴퓨터비전 커뮤니티, naver 카페
6. <http://blog.naver.com/PostView.nhn?blogId=dozero4&logNo=70031923735>
7. 인식기술과 알고리즘
8. 차량 번호판 검출 및 인식 알고리즘 , 경일대 전자공학과 김 진호 저
9. <http://www.crazymind.net/>

Q & A



Thank you.

