

# 머신러닝을 이용한 필기체 계산기

2001224 정유나

# 차례

1. 구현 결과
2. UCC
3. 결과 고찰

# 구현결과

3

목표사양	구현결과	구현함수명
필기체 입력 기능	100% 완성	<code>void onMouse(int event, int x, int y, int flags, void* userdata);</code>
문자 학습 기능	100% 완성	
모델 파일 변환	100% 완성	
=(등호)버튼	100% 완성	<code>void Object_Recognition(Mat img, vector&lt;Rect&gt; &amp;r);</code>
문자 인식 기능	100% 완성 (인식률 77)%	<code>void tm_machine(Mat dst, vector&lt;Rect&gt; &amp;r);</code>
계산 기능	100% 완성	<code>Void calc_op(double* val1, double* val2, int op_index, string str); string calc1(); double calc2(string str);</code>
AC 기능	100% 완성	<code>void onMouse(int event, int x, int y, int flags, void* userdata);</code>
CE 기능	100% 완성	<code>void ce_button(Mat img);</code>
ANS기능	100% 완성	<code>void onMouse(int event, int x, int y, int flags, void* userdata);</code>
출력 기능	100% 완성	<code>void draw(Mat img); void draw_q(Mat img, Mat dst, double value, int ox);</code>

# 구현결과

4

항목	결과
머신러닝 종류	Cnn - google teachable machine
훈련데이터 개수	18 클래스당 훈련데이터수 : 200개
GPU모델	
소스코드 라인수	408
직접 작성한 라인수	408
사용한 오픈 소스라이브러리	opencv, teachable machine
사용언어	c++

# 구현결과

5

gitHub주소	<a href="https://github.com/yn0212/Handwriting_Calculator_using-machine-learning">https://github.com/yn0212/Handwriting_Calculator_using-machine-learning</a>
----------	---

- 작성방법 : 기능별로 함수로 작성하고 모든 소스코드는 깃허브에 올리고 소스가 저장된 사이트 주소를 작성하면 됨, 소스코드에는 라인 단위로 주석이 포함되어야함,
- 발표시 깃허브 소스코드로 설명하면됨, 훈련데이터도 깃허브에 업로드할 것

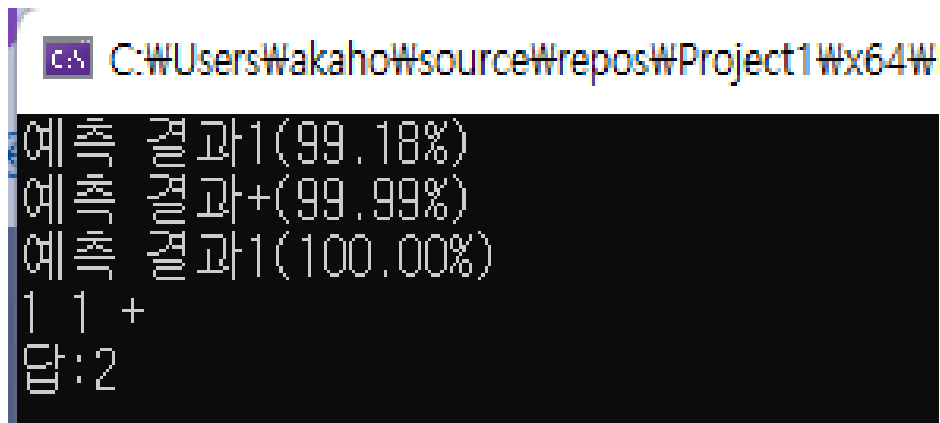
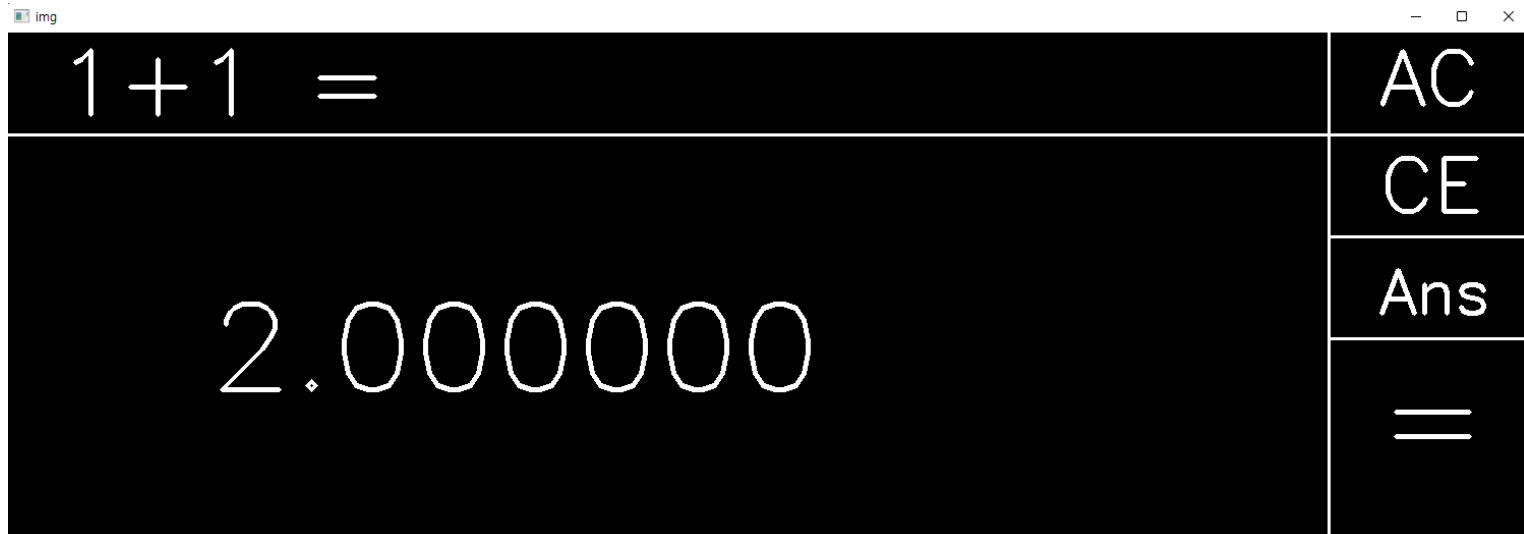
# 구현결과 : 필기체 입력 기능

6



# 구현결과 : 문자 인식 기능, 계산기능

7

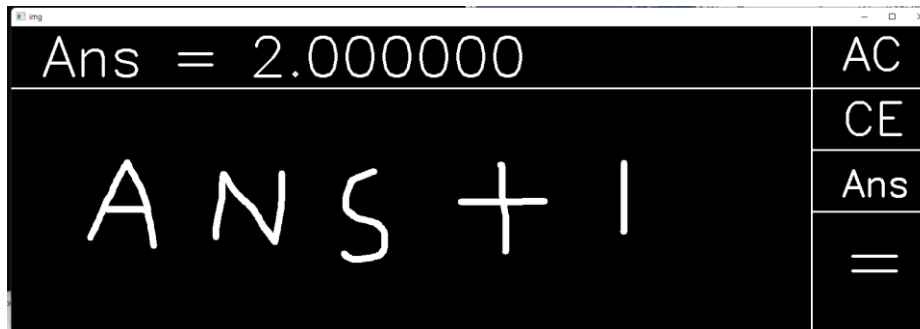


# 구현결과 : ANS기능 , AC기능

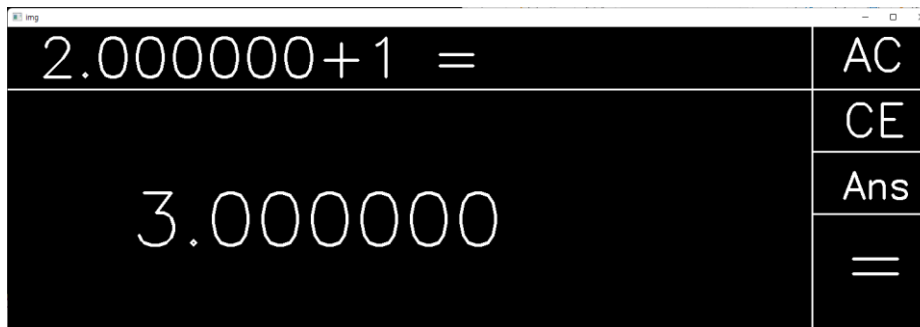
8



AC 버튼 : 입력한 필기체 지우기



ANS 문자 인식시: 저장된ans값 대입



계산



# 구현결과 : CE기능

9



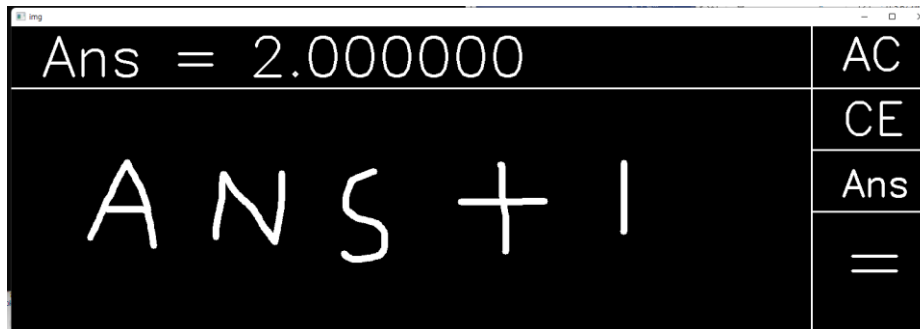
CE 버튼 : 입력한 필기체 지우기

# 구현결과 : ANS기능 , AC기능

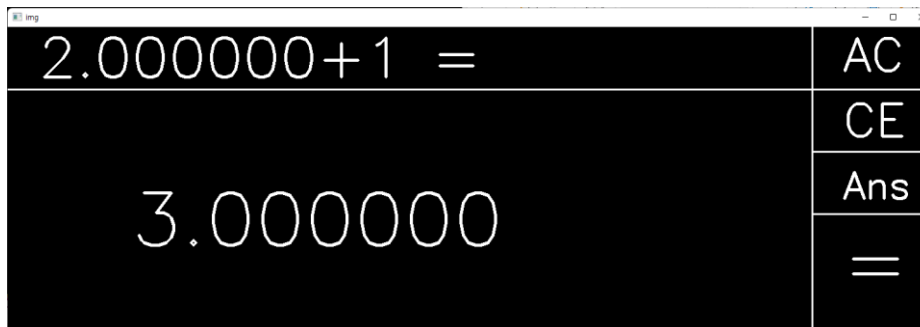
10



AC 버튼 : 입력한 필기체 지우기



ANS 문자 인식시: 저장된ans값 대입



계산

유튜브주소

<https://www.youtube.com/watch?v=JaJqFwNpuyE>

# 결과고찰

12

자체평가결과	처음엔 인식률이 낮았지만 여러 번의 시행착오를 통해 인식률을 높일 수 있었다. teachable 머신과같은 cnn을 직접 구현해보고 싶어 책을 열심히 공부한 것은 뿌듯하지만, 구현에 실패한 점이 아쉽다.
졸업작품 계획	머신 러닝 , 딥 러닝을 사용한 작품