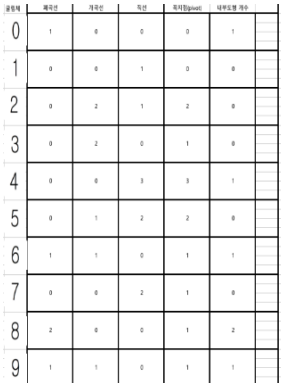
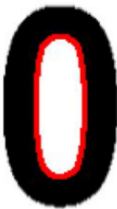


모듈1 수행 중간보고서

팀번호: 2조(코로나끝나조) 팀원명:안재현(팀장), 박세환, 윤강민, 이상혁

이름	주차	학습내용	진행중간결과	문제점	진행계획
안재현	1	<p>opencv 라이브러리를 통하여 이미지를 입력받고 숫자를 판별하기 위해 외곽선 검출을 우선적으로 생각하고 있다. opencv에 내장된 canny 알고리즘을 통하여 외곽선을 검출하였다. 각 숫자를 판별하기 위해 특징들을 찾았으며, 이를 기반으로 컴퓨터가 숫자를 인지하도록 프로 그래밍을 진행 중이다.</p> 	좌측의 사진처럼 각 숫자마다 특징들을 분류해놓았으며, 결정트리를 만들어 각 특징마다 구별할 예정이다.	각 폰트마다 검출되는 직선, 폐곡선, 개 곡선의 개수가 달라진다. 함수 내 파라미터를 조정하였지만, 이를 일반화시키는데는 시간이 걸릴 거 같으며, 다른 방법을 모색해 볼 예정이다. 또한 인풋 이미지로 중앙이 아닌 다른 위치에 숫자가 있거나, 그 크기가 달라지면 어떻게 해야 할지 아직 고민 중이다.	각 이미지가 인풋으로 들어오면 이것을 컴퓨터가 검출할 수 있도록 숫자를 중앙에 놓고 크기를 일반화시킬 예정이다. 이는 opencv를 통하여 할 예정이며, 이것이 완료되면 현재 생각으로는 일정한 격자 모양을 만들어 확률적으로 숫자를 판단하는 방법도 모색을 해봐야 할 거 같다.
박세환	1	<p>현재 글씨체에 따른 숫자 이미지들을 구분하는 과제를 진행하고 있다. opencv의 cv2 라이브러리를 이용하여 과제를 수행하고 있으며 곡선, 원, 직선, 꼭짓점 등 여러 가지 특징을 이용하여 다양한 글씨 이미지들을 구분이 가능하도록 시도하고 학습하고 있다.</p>	직선, 원, 곡선을 발견할 수 있도록 코드를 작성해보았다. 서체의 종류에 따라서 결과값이 매우 다양하고 생각지도 못한 부분에서 도형이 발견되는 모습도 보였다.	모듈의 파라미터를 조정함에 따라서 원, 직선 등의 개수가 매우 다양하게 나오는 것으로 확인되었다. 이는 적절한 파라미터를 이용해서 구분이 가능하도록 하였다.	현재 서체에 따른 숫자를 구분하기 위해서 존재하는 기준이 부족하다는 느낌이 들었다. 원이 발견되는 위치에 따라서 숫자 구분이 가능할 것으로 보이고 다른 기준들도 더하여 분류가 수월해지도록 할 예정이다.

이름	주차	학습내용	진행중간결과	문제점	진행계획
윤강민	1	<p>3개의 다른 서체로 작성된 숫자 이미지 (0~9)를 입력받아 Python을 통해 어떤 숫자인지 출력하는 OCR 프로그램을 개발하는 과제이다. 숫자마다 가지고 있는 특성들을 비교하여 비슷하거나 차이를 보이는 기준을 찾고, 이 기준(예: 꼭짓점, 폐곡선, 개곡선, 외곽선, 곡률에 따라 생성된 원 개수 등)에 따라 숫자를 분류하는 과정이 진행되고 있다.</p>	<p>현재 각 숫자의 꼭짓점, 외곽선, 곡률에 따라 생성되는 원 개수 등에 따라 분류가 되었다.</p>	<p>숫자 [6, 9] 또는 [2, 5]가 중복되는 상황이 발생하여 추가적인 기준을 찾고 있다.</p>	<p>교차점, 함수 사용 등의 기준을 추가하여 기존에 분류했던 기준에 의해 중복되는 것을 줄여나갈 계획이다.</p>
이상혁	1	<p>탐색 범위를 좁히기 위해 숫자의 내부에 도형의 포함 여부로 0~9까지의 숫자들이 두 부류의 클래스로 나눌 수 있다. 예를 들어, 숫자 0 같은 경우 내부에 둥근 도형 1개를 포함하고 있다.</p> 	<p>OpenCV 라이브러리를 통해 이미지 윤곽선을 검출해 검출된 크기가 2 이상이면 도형을 하나 가지고 있는 것으로 판단하여 내부에 도형이 있는 숫자(0, 4, 6, 8, 9) 클래스, 도형이 없는 숫자(1, 2, 3, 5, 7) 클래스 이렇게 두 부류로 나눌 수 있다.</p>	<p>탐색 범위를 반으로 좁혀서 각 클래스 내에서 더 세부적으로 분류를 위해 직선, 꼭짓점, 교점 등을 이용하려고 하나 모듈이 우리가 원하는 대로 검출해주지 못하고 있다.</p>	<p>검출이 확실하지 않은 직선, 꼭짓점 외에 확실하게 숫자 각을 나타낼 수 있는 특징과 모듈을 더 찾아볼 계획이다.</p>