

[final term] business card recognition (2/2)

Dept. Applied AI, SEOULTECH
강준(23102308), 김나현(22102304)

Content

- Introduction
- Data
- Preprocessing
- Line Detection
- Corner Detection
- Other Case Method
- Result

Introduction

- Business Card Recognition
- 목표: **다양한 배경에서 명함을 인식하는 것**
- 기본적인 이미지 처리
- 사각형 검출 알고리즘
- 예외 처리를 통한 대체 알고리즘

Data

- 총 이미지 수: 21장 (BC1 – BC21)
- 촬영 환경
 - 실내 책상 위: BC1, BC2, BC3, BC4, BC7, BC8, BC15, BC19
 - 청색 매트/판 위: BC5, BC6, BC9, BC10, BC11, BC21
 - 패턴 배경 위: BC12, BC13, BC16, BC17
 - 흰 종이 위: BC14
 - 실외 바닥: BC18
 - 손에 들린 상태: BC20



Data

- 조명 환경
 - 형광등 아래 **직사광 및 간접광**이 혼재
 - BC18(실외)처럼 **자연광 및 그림자가 섞인 상태**
- 촬영 각도 및 기울기
 - 대부분 정면이 아닌 **기울어져 있는 상태로 촬영**
- 이미지 RGB
 - 색상이 선명한 **흰색 배경(명함)**과 대비되는 **원색·채도가 강한 물체** (파란배경, 핑크 패턴 등)
 - 일부 이미지는 명함과 **유사한 톤의 배경**(회색 책상, 비슷한 흰색 종이) 위에 놓여 있어 인식 난이도가 높음

Preprocessing

- BGR -> Grayscale
- **CLAHE 추가 적용**
 - 반사광 또는 대비 왜곡이 있는 경우에는 적용해 완화
- 노이즈 제거
 - **Median Blur:** 주변 픽셀의 중간값으로 현재 픽셀 대체, 텍스처 노이즈 등을 제거
 - **Gaussian Blur:** 글자 획, 배경의 과도한 디테일 부드럽게 처리
 - **Bilateral Filter:** 명함 경계처럼 급격한 색 변화가 있는 부분 보존, 불필요한 노이즈 감소
 - **edgePreserving Filter:** Edge 보존 효과 강화, 노이즈 제거 후에도 명함 윤곽선 유지

Preprocessing

- **Canny Edge Detector**

- 저역 임계값 60, 고역 임계값 180
- 약한 Edge와 강한 Edge 구분, 선명한 테두리 검출

- **Morphological Dilation**

- 끊어진 Edge 연결, 연속된 하나의 선처럼 보이도록

- **Corner mask와 Edge를 OR 연산 수행**

- Corner Detection으로 얻은 특징을 Canny Edge에서 제거하면 명함 테두리만 남게 됨

Line Detection

- **Hough Transform**
- Line의 기울기와 위치, 선의 각도와 거리 기준으로 중복 선 제거
- 필터링 된 선을 이용해 교차점 계산
- 교차점 4개를 샘플링해 사변형 생성 (Random Sample Consensus)

Line Detection

- **Weighted Sum** 방식 사용해 최종 사각형 선택
 - 면적 점수 (Area Score)
 - 직사각형 점수 (Rectangularity Score)
 - 중심성 점수 (Centricity Score)
 - 종횡비 점수 (Aspect Ratio Score)
 - $\text{weights}=(0.2, 0.45, 0.2, 0.15)$
- **Perspective Transform**
 - 명함 이미지 출력
 - 가로형, 세로형 명함을 구분해 900x530 사이즈로 출력

Corner Detection

- **Harris Corner Detection**
- **Canny Edge** 검출, **Edge map** 생성
- Contours를 RETR_LIST 모드, CHAIN_APPROX_SIMPLE 방법으로 존재하는 **모든 윤곽선과 계층 구조를 탐색**
- **각 윤곽선 면적이 5000픽셀 이상인 것만 남겨둠**
- 명함 테두리와 무관한 잡음, 텍스트 윤곽선이 **필터링 됨**
- 필터링 된 윤곽선을 이용해 **다각형 근사(Polygon Approximation)** 적용
 - 윤곽선 둘레 길이 **2% 허용 오차**

Corner Detection

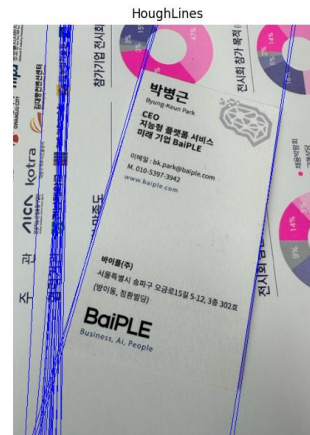
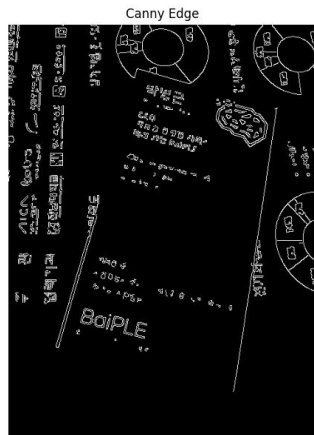
- cv2.approxPolyDP 실행, 네 개의 꼭짓점 갖고 있을 때 사각형 후보 추가
- 어떠한 윤곽선도 근사되지 않았을 경우
 - **예외처리 1. GrabCut 기반 전경 분리 기법 사용**
 - reproc 함수 호출해 명함 영역 분리, 분리 후 과정 반복
 - 8번 재시도 후 사각형 찾지 못할 경우 **Convex Hull을 추출해 사각형 후보로 사용**
 - 이후에 없으면 해당 이미지 건너뛰고 "exception" 출력
 - **하나 이상 있을 경우, 가장 큰 면적을 사각형 후보로 선택**
(면적이 큰 사각형일수록 명함 영역일 가능성이 높다는 가정)

Other Case Method

BC12 – BC20

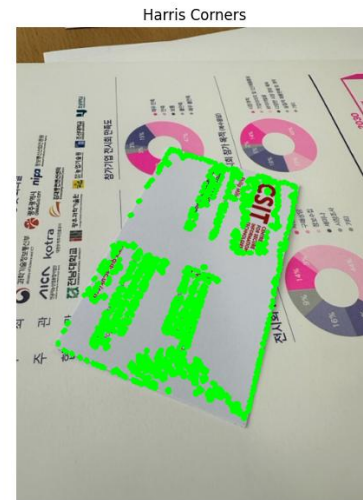
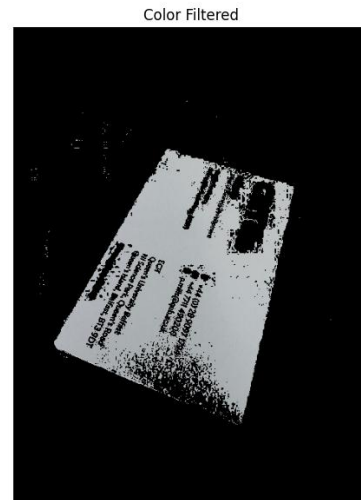
BC12

- Grayscale에 팽창(Dilate) 두 번 적용해 배경 잡음 억제
- HoughLines와 HoughLinesP를 모두 사용
- 전체 직선과 선분 모두 추출
 - 기울기 기준 평행선 필터링, 교차점 탐색
- 4개의 점으로 Affine Transform



BC13

- 명함과 배경 색상 유사, 컬러 필터링 필요
- get_mode_hsv 함수 통해 “흰색” 픽셀만 남기는 mask 적용
- MedianBlur 이후 Harris Corner Detection
- 4개의 극점 추출해 시계방향 정렬 후 Perspective Warp



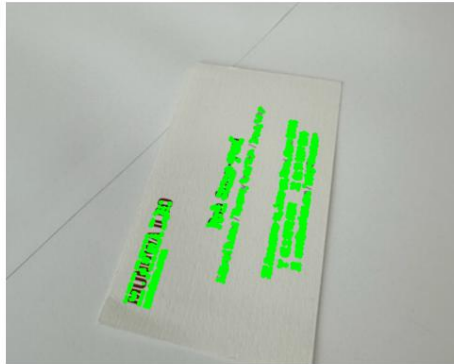
BC14

- 첫 번째 방법
- **명함이 있는 부분 추출** (상하단 20% 제거)
- Grayscale 변환, Harris Corner Detection, 텍스트 영역 인식
- 인식한 Corner 기반으로 **최소 사각형** 찾은 후 명함 인식

20% Cropped



Harris Corners



MinAreaRect on Full Image



Warped Result



BC14

- 두 번째 방법
- 첫 번째와 Corner Detection까지는 동일
- 인식된 Corner 바탕으로 Convex Hull
- 명함에서 코너가 있는 부분을 잘라냄
- 코너 주변 색을 이용해 새로운 배경을 만들고 인식한 부분을 복사함

Convex Hull & Approximated Quadrilateral

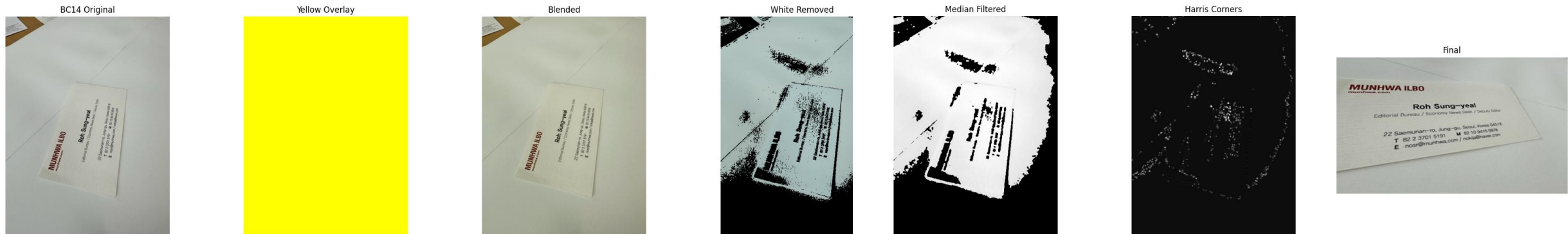


Card on Blurred Patch BG (Rotated)



BC14

- 세 번째 방법
- **Yellow color overlay alpha blending**
- MedianBlur, Harris Corner Detecion으로 주요 코너 좌표 탐색
- Corner 탐색 시 **최적 임계값 조사 위해 0.001~1까지 실험**
- 결과 임계값 0.5로 코너 픽셀 탐색 후, 명함 인식



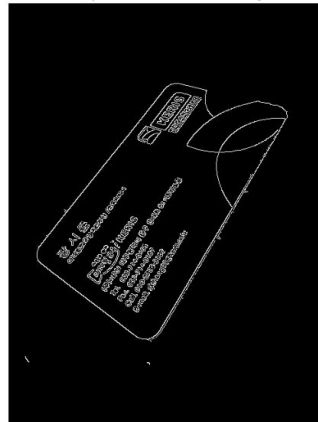
BC15

- 첫 번째 방법
- 명함에서 배경과 비슷한 녹색, 초록색 구분 위해 HSV로 색공간 변환
- 해당 색상 범위 마스킹 후 Edge 추출
- 추출된 Edge 이용해 큰 윤곽선을 찾고, approxPolyDP로 네 꼭짓점 근사 후, 시계 방향으로 정렬, 이후 명함 인식

BC15 Original



Warped -> Threshold -> Canny



Detected Contours

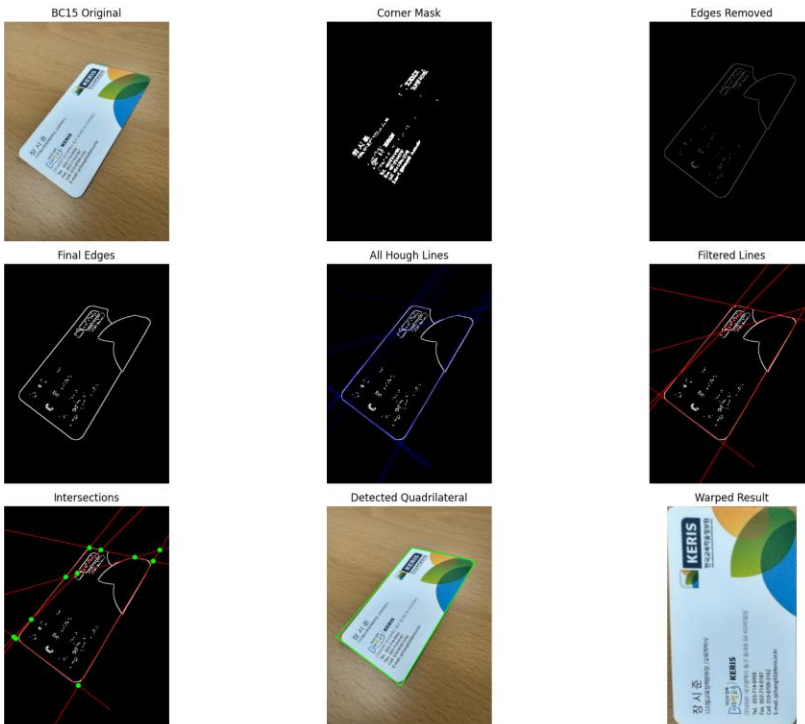


Warped Color



BC15

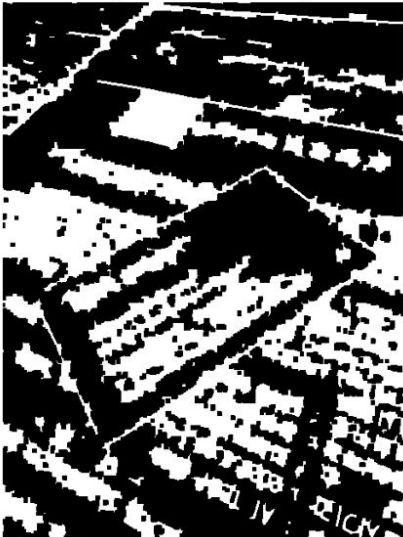
- 두 번째 방법
- 기존 Line Detection 과정에서 **HoughLine**의 임계값을 130으로 조정



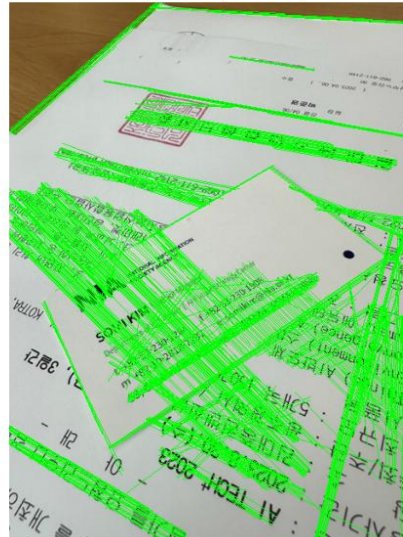
BC16

- 텍스트와 배경 노이즈가 섞인 인쇄물 위에 명함이 있어 인식에서 어려움
- CLAHE -> Laplacian -> Canny -> Morp 순서대로 적용
- Hough Lines을 뽑고 교차점 계산 결과 잘 인식되지 않음

Edges (Laplacian + Canny)



Hough Lines (423 lines)



Nearly Perpendicular Intersections (2627)



BC16

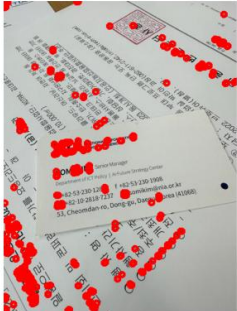
- AdaptiveThresholding으로 **조명 편차 보정 이진화** 수행
- findContours로 가장 큰 Contour 선택
- approxPolyDP로 네 꼭짓점 근사, 실패 시 Convex Hull로 볼록 껍질 생성해 사각형 후보 탐색
- 최종 사각형을 reorderPts로 정렬하고 Perspective Transform
- Otsu 이진화 후 이미지 사이즈 조절해 명함 인식



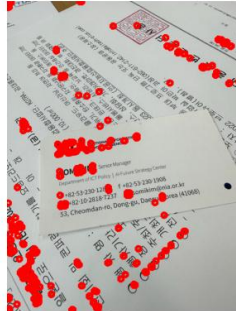
BC17

- Corner Detection을 이용해 다양한 임계값 시도
- 최적의 임계값 탐색에는 실패함

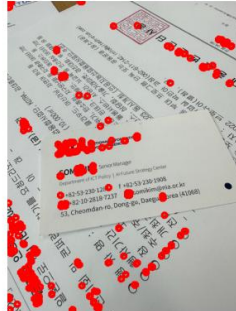
Threshold: 0.002
Corners: 2983



Threshold: 0.0025
Corners: 2651



Threshold: 0.003
Corners: 2412



Threshold: 0.0025
Corners: 2651



Threshold: 0.00275
Corners: 2511



Warped



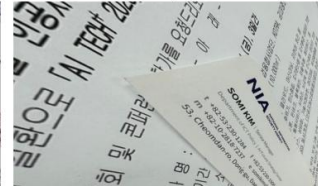
Warped



Warped



Warped

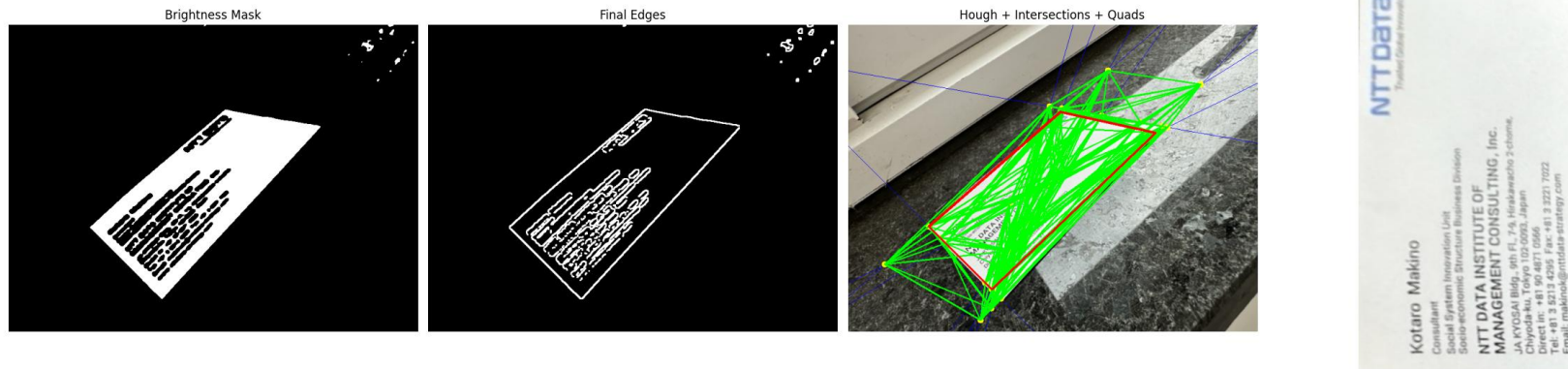


Warped



BC18

- 첫 번째 방식
- 기존 Line Detection 방법에서 빛이 반사된 부분을 제거하기 위해 추가적으로 **밝기가 230~255인 값만 하얀색으로 만드는 밝기 mask** 생성
- 이를 적용한 이미지에서 Edge 검출
- Line Detection, 교차점, 사각형 선택 후 명함 인식



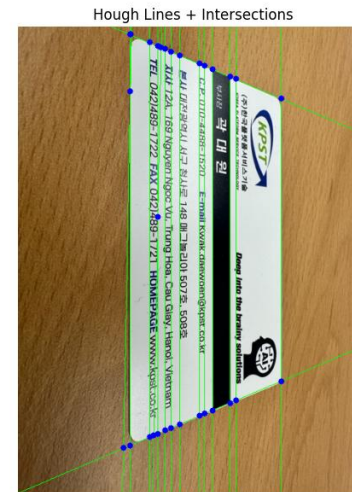
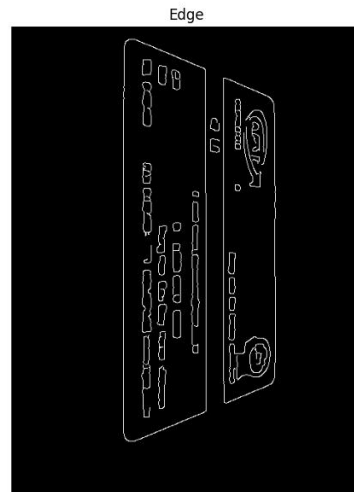
BC18

- 두 번째 방식
- **5x5, 9x9 GaussianBlur**를 연속으로 적용
- **HoughLinesP**로 찾은 **장거리 선분만**을 선별
- 선분 간 교차점 4개를 찾아 **getAffineTransform**과 **warpPerspective**를 이
용해 명함 인식



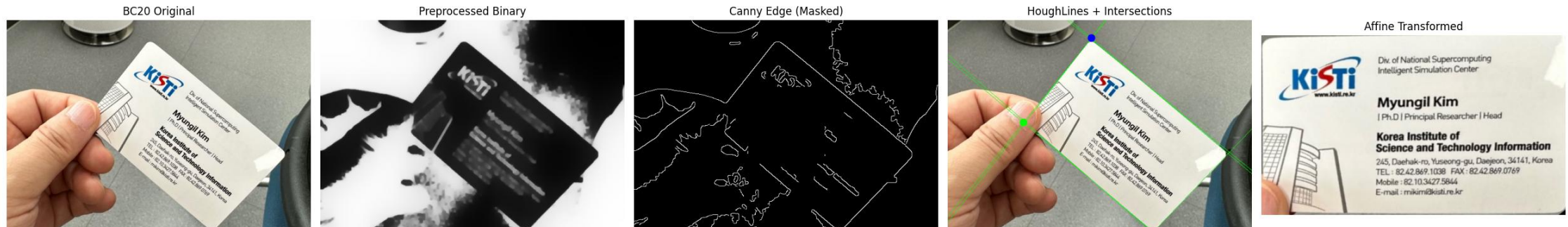
BC19

- 명암이 겹쳐있고, 명함 내부에 상자, 선이 있어 잘 인식되지 않았음
- GaussianBlur와 bilateral Filter, edgePreserving Filter 적용
- **HoughLines**로 선분을 선별해 선분 간 교차점 4개를 찾아 명함 인식



BC20

- 손가락에 가려진 부분 때문에 명함 잘 인식되지 않음
- GaussianBlur, bilateral Filter, edgePreserving Filter 적용
- 다른 알고리즘과 달리 **Affine Matrix** 계산 후, 이미지 변환, 이진화 및 Edge 검출해 명함 인식



Result

- 총 21장 중 20장 성공
- 전처리 단계에서 **노이즈와 텍스처를 효과적으로 억제함**으로써 Edge와 Corner 검출 안정성 확보
- Line Detection과 Corner Detection을 이용해 명함 테두리를 찾아냄
- 뚜렷한 배경에서 촬영된 이미지는 첫 시도에서 명함이 정확하게 인식 되었음
- 배경 노이즈, 가려짐 등으로 실패한 케이스에 대해서는 **하이퍼파라미터 튜닝, 새 알고리즘 등 추가 기법**을 적용해 명함을 인식함
- 팀 프로젝트 협업을 통해 중간 과제와 비교해 **향상된 성능 및 연산속도**를 확인할 수 있었음

Result

BC1



BC2



BC3



BC4



BC5



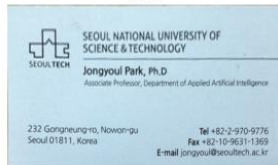
BC6



BC7



BC8



BC9



BC10



BC11



BC12



BC13



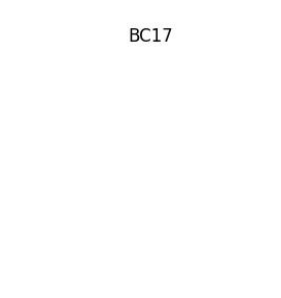
BC14



BC16



BC17



BC18



BC19



BC20



BC21



BC15



참고문헌 및 역할 분담

- 신현섭, & 김차중. (2014). 모바일 환경에서의 명함인식 성능 향상에 관한 연구. 한국정보통신학회논문지, 18(2), 318-328.
- [opencv/modules/imgproc/src at 4.x · opencv/opencv](#)
- 강준
 - **Line Detection** 알고리즘
 - **이미지 필터링**(노이즈 감소, 글자 제거) 및 Edge 검출 구현
 - Edge 검출 결과에서 사각형 미인식 문제 개선 및 **명암 문제 해결**
 - **명암 대비와 휘도 조정**을 통한 성능 향상 실험
 - 개별 작업을 전체 코드로 통합
- 김나현
 - **Cornor Detection** 알고리즘
 - 인식 실패 사례에 대해 **다양한 개선 방법 실험**
 - **흰 배경(12~14, 16, 17번)** 이미지 명함 인식 구현 및 결과 공유
 - **Occlusion이 존재하는(20번)** 이미지 명함 인식 구현 및 결과 공유
 - **컬러 필터와 하이퍼파라미터 튜닝** 등 새로운 방법을 최대한 시도하여 성능 향상 모색

Thank you

Dept. of Applied Artificial Intelligence

강준(23102308), 김나현(22102304)