

곡식 알갱이 개수 추정 알고리즘 개발과 성능평가

Performance Analysis on Grain Seeds Counting Algorithms

박승렬*, 최길튼[†], 홍진욱[†], 조환규[†]

*부산대학교 공과대학 전기컴퓨터공학부 정보컴퓨터공학과

†부산대학교 정보의생명 공학대학 정보컴퓨터공학과

{qkxk123, gilteun, hju927, hgcho}@pusan.ac.kr

요약

컴퓨터를 이용한 이미지 분석은 다양한 분야에서 응용되고 있으며 최근에는 딥러닝을 이용한 분석이 주류를 이루고 있다. 본 알고리즘은 부산대학교 정보의생명 공학대학에서 열린 “콩 세는 밤” 경진대회에 출품하기 위하여 개발한 것으로 대회 규정에 따라 공개된 데이터를 기반으로 개발하였다. 이 대회의 목적은 제공된 5장의 접시 위에 올려져있는 콩(백태) 사진을 이용하여 가장 정확한 콩의 개수를 추정하는 것이다. 사용된 방법론은 OpenCV기반의 이미지 segmentation, 허프 원 검출(Hough Circles Detection), 선형 회귀(Linear Regression)와 K-최근접 이웃(K-NN) 알고리즘 등이다. 본 논문은 대회에 최초 제출한 시스템을 4단계로 개선하였다. 최종 모델은 평가용 데이터에 대하여 평균 6% 정확도를 보였다.

1. 연구동기

이미지 분석 알고리즘은 그 대상 물체의 특성에 따라서 구분된다. 예를 들어 안면(face)을 인식하는 알고리즘과 길거리 영상에서 글자(text)를 인식하는 방법, 또는 자율 주행 차량에서 위협이 될 만한 문제 요소를 실시간으로 감지하는 방법은 많은 차이를 보인다.

몇몇 이미지 분석 알고리즘은 이미 그 형태가 잘 규정된 물체를 대상으로 한다. 대표적인 예로서, 알갱이 형태의 고체(solid) 개수를 추정하는 이미지 처리 문제가 있다. 그 중 한 사례는, 의료용으로 사용되는 알약(pills)의 개수를 정확하게 세는 알고리즘으로써 해당 알고리즘을 활용한 안드로이드 응용 프로그램이 시중의 일부 약국에서 활용되고 있다.

이보다 물체의 형태가 덜 정형화 된 경우로는, 접시 위에 무작위로 쌓여 있는 곡식의 개수를 사진을 통해 파악하는 문제를 예로 들 수 있다. 이 경우, 각각의 곡식 알갱이의 형태가 알약의 경우보다 더 비정형적이고, 사진 내부에서 접시에 해당하는 영역을 별도로 추출해야 하며, 곡식 알갱이가 쌓여 있어 모든 곡식 알갱이를 사진 속에서 확인할 수 없다는 점에서 정형화된 알약과는 다른 접근 방식을 가진 알고리즘이 필요하다.

본 논문은 부산대학교 정보의생명 공학대학에서 실시한 경진대회인 “콩 세는 밤”에 출품하여 최우수상을 수상한 본 연구팀의 결과를 소개하고 대회 이후 알고리즘의 성능 개선 결과에 관한 내용을 다룬다.

2. 관련 연구

곡물의 개수를 세는 과정은 대부분 시간이 많이 걸리고 인적 오류의 가능성이 높은 육안으로 수행하거나 도입 과정에서 비용이 많이 드는 전자 또는 레이저 기반 입자 계수기를 사용한다. 이러한 문제들을 해결하기 위하여 여러 연구가 진행되었다.

먼저 종자 발아율에 대한 검사를 자동화하기 위한 알고리즘에 관한 연구가 있다[1]. 해당 연구에서는 발아율을 확인하기 위해 종자의 본체와 뿌리를 세분화하여 탐지하고, 이들이 연결된 상태인지 확인하였다. 그러나 [1]에서 제시한 알고리즘은 종자가 모두 분리된 상태가 보장되어야 한다는 한계가 있다.

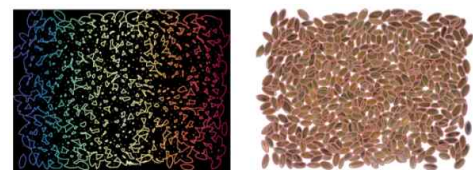


그림 1. Ref[3]의 쌀알 개수 세기 과정의 일부

곡식을 계수하는 여러 오픈소스 알고리즘에 대하여 곡식 종류 별 정확도와 편향성을 분석한 연구도 있다[2]. 또한 최근에는 딥러닝을 응용한 방법도 공개되고 있다. Ref.[3] 연구에서는 이전 연구와 달리 쌓여 있는 곡식 알갱이의 중복되는 형태와 개수를 면밀히 파악하기 위하여 곡식이 쌓여 있는 형태의 등고선을 파악하는 인공신경망 기술을 활용하여 정확도를 향상시키는 방식이 제시되어 있다[3].

한 연구에서는 곡식을 계수하는 알고리즘의 정확도가 사진을 촬영하는 모바일 디바이스와 주변의 조명 여건에 따라 어떤 영향을 받는지를 파악하였고, 조명의 상태가 계수 정확도에 큰 영향을 준다는 사실을 입증하였다[4].

최근에는 이 문제에서 CNN(Convolution Neural Network)도 활용되고 있음이 확인할 수 있다. 연구 [5]는 two-column convolution neural network를 응용하여 쌓인 곡식의 밀도 추정을 통한 개수 측정 알고리즘을 새롭게 제시하였다. 해당 연구에서는 쿵이 그 껍질에 쌓여 있는 사진을 활용하여 낱알의 개수를 파악하였다. 해당 논문에서는 이를 위해 개별 껍질의 개수를 파악하는 알고리즘과, 껍질 속 쿵 낱알의 개수를 파악하는 알고리즘을 융합한 해결책을 제시하였다[5].

이상의 연구들에서는 곡식의 낱알이 겹쳐 있지 않거나, 약한 정도로 겹쳐 있는 경우에 대한 알고리즘을 중점적으로 다루었다.

약학 분야에서도 물체 계수에 관한 많은 연구가 진행되었다. 의료용 알약(pills)의 개수를 목적으로 한 일부 연구들은 다음과 같다.

이미지 분석이 아닌 계수기를 이용하여 알약을 계수하는 연구 역시 진행된 바 있다. Ref.[6] 연구에서는 백색광의 LED와 포토다이오드 사이로 낙하하는 알약에 대하여 알약 낙하 유형이 포토다이오드 센서에 미치는 영향을 분석하여 해당 센서를 통한 알약 계수 알고리즘을 제시하였다. 이러한 연구는 체계화된 알약 계수기를 필요로 한다는 점에서 운용 환경의 제약이 있다[6].

이미지 분석을 기반으로 한 Ref.[7] 연구에서는 알약이 펼쳐진 사진 속 다양한 알약의 형태와 크기를 파악하고 이를 종류별로 식별 및 계수하는 알고리즘을 제시하였고 이 결과를 널리 알려져 있는 알고리즘인 SIFT등과 비교하였다[7].

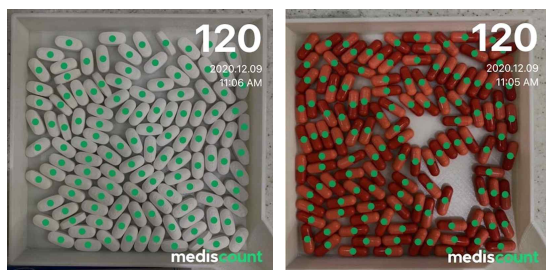


그림 2. Pilleye 어플리케이션의 실행 화면

또한 위 논문과 유사한 연구로써, (주)메딜리티의 Pilleye 어플리케이션은 알약의 식별 및 계수 서비스를 제공한다. 해당 어플리케이션 역시 모든 알약이 겹쳐져 있지 않은 상황을 가정하고 작동한다. 해당 연구에서는 반으로 쪼개진 알약을 찾아내고 알약 데이터베이스와 연동하여 어떤 종류의 알약이

사진에 있는지 파악하는 기능을 제공하는 등의 발전을 이루었다[8].

그러나 본 논문에서 다루고자 하는 곡식 사진의 경우 그 특성상 쿵 낱알들이 다층적으로 쌓여 있기 때문에 이상의 연구들에서 제시한 방법을 적용하는 것에 한계가 있다. 곡식 낱알과 같은 대상 물체가 쌓여(piling)있는 연구는 상대적으로 덜 진행되어 있다.

다층적(multi layers)으로 겹쳐 쌓여있는 곡식 알갱이의 개수 추정 알고리즘 개발을 목표로한 본 연구는 물체를 내려다보는 방향에서 찍은 사진만을 사용함에서 벗어나 다양한 각도의 사진을 사용함으로 개수 추정 알고리즘의 정확도를 높이하고자 하였다.

3. “쿵 세는 밤” 대회 개요

3.1 대회에 사용한 데이터

“쿵 세는 밤”(이하 쿵밤)은 이미 label이 부여된 사진을 모든 참가팀에게 제공한 뒤 대회 당일 학생들에게 비공개 데이터를 공개하여 접시에 쌓인 쿵 개수를 가장 정확하게 계산한 팀을 선정하는 대회이다. 참가자들에게 제공된 사진은 최소 2개에서 최대 1600개까지의 쿵(백태)이 흰색 종이 접시에 올려있는 사진들이다. 쿵 개수가 700개를 넘는 몇몇 경우에는 쿵이 하나 이상의 층(layer)으로 쌓여 있기도 하였다.

공개된 데이터 set에는 정답이 있는 30개의 case가 제공되었다. 각 case별로 Perspective view로 촬영된 4장의 사진(0도, 90도, 180도, 270도로 접시를 회전)과 Top view로 측정된 사진 1장이 제공되었다. 사진은 일반 스마트폰 카메라를 사용하여 촬영되었다. 모든 사진들은 삼각대에 올린 스마트폰 카메라로 촬영되었기 때문에 접시에 대한 orientation 이 모두 일정하게 조정되어 있다.

3.2 개발 환경에 대한 제한 사항

대회에 참가한 팀이 사용할 수 있는 개발 환경은 다음과 같았다. 사용 언어는 Python, C/C++, JAVA로 한정하였으며 OpenCV, Numpy 이외의 다른 라이브러리는 사용이 금지되었다. 이에 따라 AI 기반의 플랫폼인 Tensorflow나 Pytorch 등은 허용되지 않았다. 또한 최종 평가가 이루어지는 환경은 인터넷과 분리되어 있어 모든 제출 결과물은 네트워크 연결 없이 Windows 10 기반의 standalone PC에서 수행되어야 하였다. 최종 평가 시에도 이미 공개된 150장의 공개된 데이터가 계산 과정에서 활용될 수 있도록 제공되었다. 소스 코드의 크기는 20MB를 넘지 않도록 강제되었으며 계산은 5분 이내에 수행되어야 했다.

3.3 참가팀 등위 결정방법

팀별 등위를 결정하는 방법은 다음과 같았다. test case t_i 의 접시에 올린 실제 콩 개수를 b_i 라고 하고 참가팀이 5개의 사진을 이용하여 추정한 (guessing) 콩 개수를 g_i 라고 할 때 그 오차(error)는 보편적인 오차비율의 제곱차를 이용하여 다음과 같이 정의하였다.

$$e_i = (100 \cdot |b_i - g_i| / g_i)^2$$

각 참가팀의 결과는 30개 Hidden data에 대하여 계산한 e_i 를 모두 더한 값으로 평가된다. 이 값을 기준으로 낮은 순위대로 등위를 결정하였다. 만일 이 값이 동일할 경우에는 위 L_2 metric이 아닌 L_3 metric을 사용하였다.

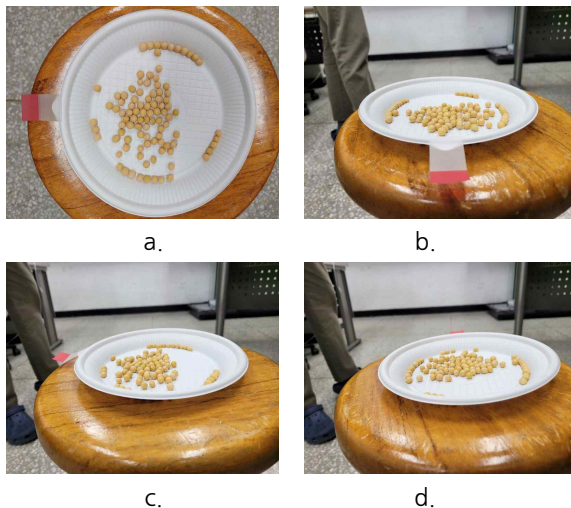


그림 3. 콩 개수를 정확하게 맞춘 테스트 케이스

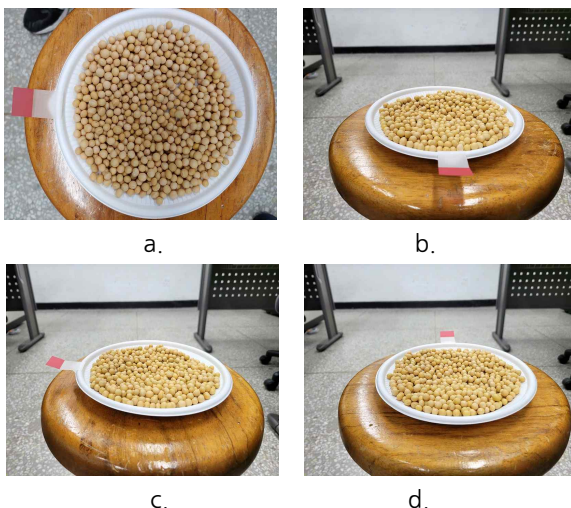


그림 4. 가장 큰 오차율(53%)이 발생한 데이터

위 그림 3은 본 제안하는 시스템을 사용하여 평가용 데이터의 콩 개수를 정확하게 추정한 case를 보여주고 있다. 그리고 그림 4는 가장 큰 오차율을 보인 사진(testing case)을 보여주고 있다.

그림 3과 그림 4의 a는 콩이 담긴 접시를 위에서 촬영한 Top view 사진이다. 그림 3과 그림 4의 b, c, d는 각각 측면 0도, 90도, 180도에서 촬영한 Perspective view 사진이다. 사진에 나타난 빨간 포스트잇은 접시 방향의 기준점을 알리기 위하여 “콩 세는 밤” 주최 측에서 표시한 것이다.

4. 제안하는 계수(Counting) 알고리즘

이번 장에서는 본 연구팀이 개발한 4가지 알고리즘 4.1, 4.2, 4.3, 4.4에 대해서 설명할 것이다.

4.1 Hough Circle 검출과 HSV 마스크

Step1. 주어진 전체 이미지를 800*600 해상도로 변환하여 코드의 실행 시간을 줄이고 이미지의 노이즈를 제거하였다.

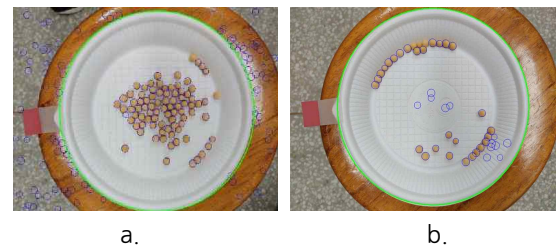


그림 5. 접시 범위를 고려한 허프 원 검출

Step2. Top View 이미지에 대해 OpenCV 패키지의 허프 원 검출 알고리즘으로 콩을 검출하였다. 접시 밖의 바닥 무늬나 의자에서도 원이 검출이 되는 것을 확인하였다.

Step3. 원 형태인 접시의 크기를 검출하여 접시의 내부에 있는 콩만을 검출하도록 했다. 접시 검출은 허프 원 검출을 사용하여 특정 크기의 원을 검출하였다. 비슷한 크기의 오검출한 원들을 배제하기 위해 원 중심이 중앙에서 가장 근접한 원을 접시로 선택하였다.

Step4. 접시 테두리에 원이 오검출되는 경우가 있어 접시 크기의 97프로 이내에서 검출하도록 수정하였다. 콩 개수의 최대치인 1600개의 경우에도 접시 테두리까지 콩이 쌓이지 않음을 확인하였다.

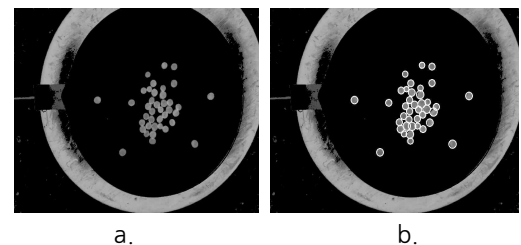


그림 6. HSV 색상 마스크와 검출 결과.

그림 6-b에서 작은 원은 그림 6-a의 이미지를 사용해 검출된 콩을 나타낸다.

Step5. 그림 5-b에서 보이는 오검출을 줄이기 위해 그림 6-a와 같이 톤 색깔의 HSV 색상 범위를 이용하여 톤 색깔을 제외한 부분 마스킹하고 톤 검출을 시도하였다. HSV 색상 범위는 색상 표를 참고하여 톤 색상의 채도, 명도 값을 포함하는 범위를 경험적으로 선택하였다.

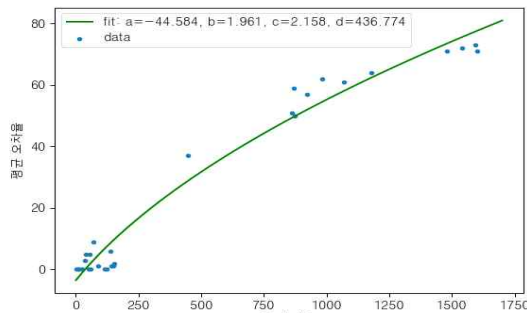


그림 7. 알고리즘 4.1을 사용하여 예측된 톤 개수에 따른 평균 오차율. (가로축은 실제 톤의 개수)

그림 7은 실제 톤 개수에 따른 예측치의 평균 오차율에 대해 커브 피팅(Curve fitting)을 한 결과이다. 그림 7을 보면 톤 개수가 증가함에 따라 오차율이 90%에 수렴하고 있음을 확인할 수 있다. 그림으로 커브 피팅 모델 함수 $f(x)$ 는 다음과 같이 추정하였다. 조절변수는 {a, b, c, d}이다.

$$f(x; a, b, c, d) = a + b \cdot \sqrt{c \cdot x + d}$$

본 과정을 통해 약 200개 정도까지의 톤 개수는 오차율 평균 2% 이내로 검출되었다. 톤 개수가 200개보다 많은 경우 겹치는 부분과 보이지 않는 부분을 파악하지 못하여 오차율이 높아지는 경향을 보였다. 이를 해결하기 위해 알고리즘 4.2, 4.3, 4.4는 알고리즘 4.1을 실행한 결과로 얻은 추정 값과 이미지를 사용하여 200개 이상의 데이터에 대해서 추가적인 개선을 적용한 알고리즘이다.

4.2 중첩도 개념과 선형 보간

알고리즘 4.2에서는 기존 알고리즘 4.1이 해결하지 못한 200개 이상의 데이터에 대해서 중점적으로 개선하였다. 겹치는 부분과 보이지 않는 부분을 파악하기 위해서 톤의 중첩도를 추정하고 이를 초기 추정 값의 선형 보간에 사용하였다.

먼저 Top view 이미지에서의 층(layer)별 거리에 따른 톤의 크기의 차이를 통해 톤의 층(layer)을 파악하려고 하였다. 하지만 이미지에서 각 톤의 크기 차이를 확인했을 때, 최대 톤의 크기와 최소 톤의 크기의 차이가 거리에 따른 층(layer)별 톤의 크기 차이보다 매우 큰 것을 확인하고 제외하였다.

다음으로 톤의 중첩도(overlap_rate)개념을 도입하여 이미지에서 톤이 쌓여있는 공간 복잡도를 계산하고자 하였다. 정확한 중첩도 계산을 위해 모든 톤들에 대해 각 톤의 반지름과 다른 톤들과의 거리를 비교하려고 하였으나 시간 복잡도가 $O(N!)$ 으로 실행시간이 너무 길어지기 때문에 제외하였다.

마지막으로 허프 원 검출 알고리즘의 검출 민감도(minDist)를 조절하였다. 톤이 층층이 쌓여 있는 경우에는 민감도 조절에 따른 검출 개수가 크게 변화하는 반면, 한 층으로 쌓인 경우 검출 개수의 변화가 작았다. 그래서 한 층 이상 쌓였을 것으로 생각되는 초기 추정 값이 200개 이상인 case에 대해 다음과 같이 중첩도를 추정하였다. HoughCircles() 함수의 파라미터 중 minDist, param1, param2 값을 (가장 큰 톤의 반지름*0.95), 80, 9로 설정했을 때의 접시 안 톤의 개수를 h, (가장 큰 톤의 반지름*0.65), 80, 9로 설정했을 때의 접시 안 톤의 개수를 l이라 했을 때 overlap_rate는 다음과 같다.

$$\text{overlap_rate} = \frac{h}{l}$$

위 식을 바탕으로, overlap_rate가 0.05 이상인 경우 톤이 한 층 이상으로 쌓여 있다고 판단했다.

초기 추정 값이 200개 이상인 case에 대한 오차율 개선을 위해 선형회귀 (Linear Regression)를 적용하려고 했지만, 대회 제약 조건에 따라 Pytorch와 같은 머신러닝 모듈을 사용할 수 없어 선형 회귀 식은 위의 overlap_rate에 대한 실제 톤 개수를 나타내는 그래프를 보며 결정하였다.

Open Case 30번의 경우 검출 민감도별 검출치의 차이가 커서 선형 보간을 통해 오차율을 굉장히 낮출 수 있었던 반면 Open Case 22번은 검출 민감도별 검출치 차이가 없었고 최초 검출치가 실제 톤의 개수와 유사하였다.

4.3 다차원 선형 회귀

알고리즘 4.2은 AI 기반 platform 모델을 사용하지 않고 Top view 이미지와 overlap_rate에 따른 선형 보간을 사용한다. 알고리즘 4.3은 보다 정교한 선형 보간을 위해 Scikit-learn의 Linear regression 모델을 사용한다. 그리고 Hidden 데이터의 Top View를 포함한 Perspective View 사진에 대해서도 알고리즘 4.1의 방법을 사용하였다. 이를 통해 나온 5개의 초기 추정 값 벡터들을 overlap_rate 대신 사용하였고, 이 이미지 각도 별 톤 개수에 대한 5개의 추정 값 벡터를 Linear regression 모델에 학습시켰다. 이를 위해서 본 연구팀은 Python의 Scikit-learn(version 1.0.2) 패키지의 LinearRegression()을 사용하였다.

4.4 K-최근접 이웃

알고리즘 4.4에서는 알고리즘 4.3과 동일한 5개의 추정 값 벡터를 선형 회귀 모델이 아닌 K-최근접 이웃(KNN) 모델에 사용하였다. 이미지 각도별 초기 추정 값이 비슷한 case는 실제 쿡의 개수 또한 비슷할 것이고 데이터가 많아질수록 선형 회귀보다 K-최근접 이웃(KNN) 모델이 오차율을 줄일 것이라고 판단하였다. 이를 위해서 본 연구팀은 Python의 Scikit-learn (version 1.0.2) 패키지의 KNeighborsClassifier()을 사용하였다.

5. 실험과 성능평가

이 장에서는 “쿡 세는 밤” 대회에서 제공한 데이터를 사용하여 다양한 조건에서 계수 추정 알고리즘의 성능을 평가한 결과를 설명한다. 그림 8은 실제 쿡 개수가 2개에서 100개, 100개에서 900개, 900개에서 1600개 사이와 전체 데이터에 대한 알고리즘별 평균 오차율 그래프이다. 각 개선마다 모든 쿡 개수 구간에서 오차율이 감소하고 있음을 확인할 수 있다.

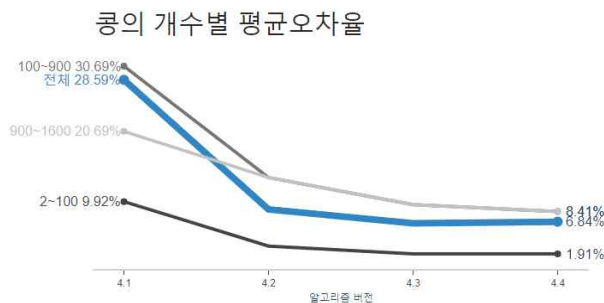


그림 8. 각 알고리즘 버전의 실제 쿡 개수 범위 별 추정 값들에 따른 평균오차율

알고리즘 4.1. Top view의 사진에서 허프 원 검출과 HSV 마스킹을 실행한 알고리즘 4.1의 Hidden data 30개에 대한 오차(error)의 합은 54998 이고, 평균 오차율은 28.59% 이다. 200개 이하에 대한 검출 오차율은 2%이고 200개 이상에 대한 검출 오차율은 60%이다. 이는 사진 속의 쿡 개수가 200개를 넘어가면서 겹치는 쿡을 검출하지 못하였기 때문이다.

알고리즘 4.2. 중첩도 개념과 선형 보간을 적용한 알고리즘 4.2의 Hidden data 30개에 대한 오차(error)의 합은 7247 이고, 평균 오차율은 8.72% 이다. 이는 대회에서 실제 사용했던 알고리즘으로 AI Platform의 모델을 사용하지 않고 overlap_rate에 따른 선형 보간으로 쿡의 개수를 추정한다.

알고리즘 4.3. 다차원 선형 회귀를 적용한 알고리즘 4.3의 Hidden data 30개에 대한 오차(error)의 합은 3919 이고, 평균 오차율은 6.59% 이다. 선형 회귀 모델을 사용함으로써 알고리즘 4.2에 비해 오차율과 오차의 합을 개선할 수 있었다.

알고리즘 4.4. K-최근접 이웃을 적용한 알고리즘 4.4의 Hidden data 30개에 대한 오차(error)의 합은 3018 이고, 평균 오차율은 6.83% 이다. 알고리즘 4.3보다 0.24%P만큼 높은 오차율이지만 오차(error)의 합을 901만큼 개선했다. 추가적인 데이터를 확보하고 모델을 개선한다면 다른 모델보다 더 좋은 성능을 낼 것으로 판단된다.

표-1. 단계별 알고리즘의 성능 비교

t_i	v 4.1	v 4.2	v 4.3	v 4.4	b_i
T18	4	4	4	4	4
T08	6	6	6	6	6
T01	10	10	10	10	10
T25	14	14	14	14	14
T23	24	24	24	24	24
T04	36	36	36	36	36
T30	37	37	37	37	39
T15	52	52	52	52	52
T05	54	54	54	54	57
T03	59	59	59	59	59
T14	63	63	63	63	69
T10	90	90	90	90	89
T16	116	116	116	116	116
T11	125	125	125	125	125
T06	132	132	132	132	140
T19	142	142	142	142	143
T29	152	152	152	152	150
T26	157	157	157	157	154
T27	284	373	483	379	449
T21	420	420	1139	1032	860
T09	355	823	833	691	870
T13	436	1336	1228	1032	873
T24	393	1064	1087	1032	982
T07	373	937	849	1032	982
T02	412	1184	1205	1190	1070
T28	424	1260	1302	1429	1178
T20	429	1292	1371	1429	1480
T12	431	1304	1416	1323	1541
T22	436	1336	1463	1323	1594
T17	461	1494	1635	1375	1600
E1	54998	7247	3919	3018	
E2	28.59	8.72	6.59	6.83	

E1은 오차인 $\text{error}(e_i)$ 의 합을 의미하고, E2은 평균오차율을 의미한다. 표-1을 보면 200개 이하의 데이터에 대해서는 모든 알고리즘에서 정확한 추정을 하고 있고, 알고리즘의 개선이 진행될수록 200개 이상의 데이터에 대해 추정 값이 정확해지고 있는 것을 확인할 수 있다.

6. 결론

본 연구팀은 부산대학교 정보의생명 공학대학에서 개최한 “콩 세는 밤” 대회에 참가하기 위하여 접시 위에 담긴 콩 개수를 계수하는 시스템을 개발하였다. 이 개발 과정에서 본 연구팀은 4번의 알고리즘 개선을 진행하였으며 각 개선 단계마다 유의미한 성능의 향상이 있음을 확인할 수 있었으면 그 결론은 다음 3가지로 요약될 수 있다.

- 최초 알고리즘을 점진적으로 개선한 결과, 가장 낮은 오차율을 보인 알고리즘 4.3 다차원 선형 회귀 알고리즘을 적용했을 때 6.59%의 가장 낮은 오차율을 보였다.
- 오차(Error)를 고려한 경우 알고리즘 4.4 K-최근접 이웃 알고리즘을 적용했을 때 6.83%의 평균 오차율과 3018의 오차(error)의 합으로 가장 좋은 성능을 보였다.
- 학습 데이터의 개수와 종류가 많아졌을 때 알고리즘 4.4 경우 다른 모델보다 더 정확한 추정을 할 수 있을 것으로 기대된다.

일반적으로 씨앗을 판매하는 업체들은 각 봉지에 실제로 들어있는 씨앗이 제품에 표기된 개수의 10%내외인 경우를 정상 제품의 합격기준으로 보고 있다. 즉 만일 특정 씨앗이 500개 단위로 판매될 때 각 용기에 실제로 들어있는 씨앗의 개수는 450개에서 550개 사이임을 업체가 보장한다는 의미이다. 이런 기준으로 볼 때 본 시스템에서 보인 평균 오차율 6%는 충분히 실용성이 있음을 나타낸다. 본 연구 이후의 연구 주제는 다음과 같다.

- 30여개의 Open Data를 확장하여 최소 1000개 이상의 데이터를 확보하고자 한다. 이를 위하여 이번 대회와 같은 수동적 작업이 아니라 자동적으로 정답 데이터를 생성하는 장치(콩을 붓고 5개의 사진을 기록하는)를 개발하고자 한다.
- 오차(Error)를 고려한 경우 알고리즘 4.4 K-최근접 이웃 알고리즘을 개선하고자 한다.
- 콩 이외의 다른 종류의 곡식, 즉 크기가 다른 예를 들어 매우 작은 크기의 좁쌀, 더 큰 크기의 완두콩이나 더 나아가 밤(chestnut)을 대상으로 측정해보는 것에 흥미로운 주제가 될 것이다. 크기에 따른 오차율의 변화가 있는지 알아보려고 한다.
- 여러 종류의 곡식이 섞인 환경에서 각각의 알갱이를 추정하는 문제 역시 매우 흥미로운 것으로 예상된다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음 (2016-0-00019)

참고문헌

- [1] Khoenkaw, Paween. "An image-processing based algorithm for rice seed germination rate evaluation." Int. Comp. Sc. and Eng. Conference (ICSEC). IEEE, 2016.
- [2] Mussadiq, Zohaib, et al. "Evaluation and comparison of open source program solutions for automatic seed counting on digital images." Computers and Electronics in Agriculture no. 117 pp. 194-199, 2015.
- [3] Sun, Jin, et al. "Deep learning optimization method for counting overlapping rice seeds." Journal of Food Process Engineering 44.9 : e13787, 2021.
- [4] Komyshev, Evgenii, et al. "Evaluation of the Seed Counter, a mobile application for grain phenotyping." Frontiers in plant science vol. 7, 2017.
- [5] Li, Yue, et al. "Soybean seed counting based on pod image using two-column convolution neural network." IEEE Access 7 , pp. 64177-64185, 2019.
- [6] Lee, Soon-Geul, et al. "Development of Pill Counting Algorithm and Pill Counting Machine Using Non-contact Photo Sensor." Journal of Institute of Control, Robotics and Systems 11.9 pp. 810-815, 2005.
- [7] Palenichka et al. "Visual attention-guided approach to monitoring of medication dispensing using multi-location feature," Proc. of the IEEE International Conference on Computer Vision Workshops. 2015.
- [8] Medilty Inc., "Pilleye - tablet, pill counter." www.pilleye.com. Accessed 12, Dec 2021.