

선택적 탐색을 이용한 포도송이 검출

Finding Grapevines with Selective Search

이성현

고려대학교 전기전자공학부

(kate1301@naver.com)

본 연구는 선택적 탐색 기법을 적용하여 어떠한 객체가 있을 것으로 예상되는 영역들을 모두 찾고, 각 영역들에 대해 포도송이를 담고 있는 window 이미지들과의 유사도 비교를 통해 해당 영역이 나타내는 객체가 포도송이인 영역만을 결과로 도출한다. 이후 각 영역의 중심 사이의 거리를 가로(x), 세로(y) 방향으로 각각 계산하여 threshold보다 가까이 위치하고 있다면 두 영역을 통합하여 입력 이미지 안에 있는 총 포도송이의 개수 및 각각의 대략적인 위치를 최대한 중복이 없도록 찾아낸다. 본 알고리즘의 시간적, 연산적 장점을 바탕으로 포도송이와 관련된 소비자, 농가, 공장 및 기업 등 다양한 분야에서의 효율적인 적용을 기대할 수 있다.

1. 서론

컴퓨터 기술이 발전하면서 사람들은 직접 하는 것보다 훨씬 빠르고 효율적으로 많은 일들을 처리할 수 있게 되었다. 특히, 인간의 눈이 하는 역할과 더 나아가 시각 정보에 대한 처리 및 인식 과정을 컴퓨터로 구현하고 분석하려는 노력은 계속 이루어지고 있다. 이러한 객체 인식 분야는 영상 감시, 얼굴 인식, 로봇 제어, 자율 주행, 제조업 보안 등 산업 전반에서 핵심 기술로 사용되고 있어 그 중요성이 크다. 객체 인식 연구는 객체의 특징을 설계하고 검출하여 해당 객체를 찾아내는 방식에서 시작하여 합성곱 신경망인 CNN과 같은 네트워크로 지역적인 정보도 고려하는 방식으로 인식율을 크게 높이며 연구가 활발하게 이루어졌다.

한편, 주어진 이미지 속 객체가 무엇인지 인식하는 문제와 주어진 이미지 속 객체가 어디에 존재하는지를 알아내는 문제는 다르다. 따라서 객체의 위치를 검출하는 방법에 대한 연구도 이루어지기 시작했다. R-CNN은 이를 딥러닝의 회귀 방법으로 해결했으나, 속도가 매우 느려서 이를 보완하기 위해 Fast R-CNN이 개발되었다. Fast R-CNN의 개발로 객체 인식 속도도 매우 개선되었고 딥러닝을 이용할 수 있게 되었다. 그 이후에는 R-FCN이 등장하여 지역적 정보에 지나치게 의존적인 단점을 개선했다. 하지만 객체 인식이 필요한 로봇, 자율 주행 등에 적용하기 위해선 실시간에 가까운 처리 속도가 필요했기에 YOLO(You Only Look Once)와 SSD(Single Shot MultiBox Detector) 등의 아주 빠른 검출 속도를 보

이는 방법들이 제안되었다.

본 연구는 객체 위치 인식의 기초 원리를 담고 있는 선택적 탐색 알고리즘을 기반으로 포도송이의 위치를 예측함으로써, 전체 인식 모델의 성능보다는 알고리즘 작동 원리에 대한 이해를 바탕으로 포도송이라는 특정 객체만을 선택하는 효율적이고 정확도 높은 방법을 고안하고자 한다. 선택적 탐색 알고리즘으로 통해 얻어낸 제안 영역에 대해 포도송이 window와의 유사도 비교를 통해 포도송이의 특성을 나타내는 영역만을 결과로 도출하고자 한다.

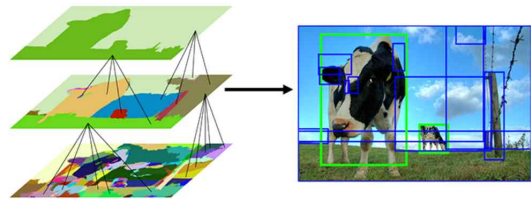
많은 종류의 객체들 중에서 포도송이를 선택한 이유는 여러 송이가 겹쳐 있을 경우 쉽게 구분하기가 어렵기 때문이고, 이 문제를 해결했을 경우 해당 기술을 적용할 분야가 많을 것으로 예상되기 때문이다. 소비자는 포장되어 있는 포도송이의 개수를 간단하고도 빠르게 파악할 수 있고, 농민의 경우에는 본인의 밭에 몇 송이의 포도가 자라고 있는지 빠르게 알 수 있다. 또한 공장 안에서는 포장 과정에서 포도의 수량의 파악이 쉬워지고 각 위치를 빠르게 알 수 있기 때문에 감시에 이점이 있다. 그 외에 마트나 백화점 등의 기업에서는 지나가는 카트나 장바구니 안의 포도송이의 수량을 빠르게 파악할 수 있을 것으로 기대된다.

2. 이론적 배경

2.1. 선택적 탐색 (Selective Search)

선택적 탐색은 sliding windows 및 image pyramids를 보완하기 위해 등장한 방법으로 더 빠르고 더 효율적이면서 객체를 포함하고 있는 영역을 정확하게 탐지하는 영역 제안 알고리즘을 고안하고자 만들어졌다. 이 알고리즘은 superpixel을 활용하여 이미지를 과도하게 segmentation 하는 방법에서 시작했다. 이때 superpixel은 edge

를 잘 표현하는 patch를 의미한다. 찾아낸 superpixel들을 특정 조건에 따라 병합하여 객체를 포함하고 있을 것으로 추정되는 영역을 만들어낸다. 특히 superpixel들을 병합하는 과정에서 네 가지의 주요 유사도를 측정하여 기준으로 활용하는 계층적 방법을 사용한다.



(그림 1) 선택적 탐색

계층적 그룹핑 알고리즘은 입력으로 이미지를 받아 객체의 위치에 대한 예측들의 집합을 출력한다. R 을 선택된 영역 후보들의 집합인 $\{r_1, r_2, \dots\}$ 라고 하고 S 를 그 영역들의 유사도 집합인 $\{s(r_i, r_j), \dots\}$ 라고 할 때, 학습 방법은 다음과 같다. 먼저 R 을 초기화하고, 가장 유사도가 높은 $s(r_i, r_j)$ 을 선택하여 r_i 와 r_j 을 r_t 로 병합하며 S 에서 $s(r_i, r_*)$, $s(r_j, r_*)$ 을 제거한다. 그런 다음 병합하여 새로 만들어진 r_t 에 대해 다른 영역들과의 유사도 집합인 S_t 을 계산한다. r_t 는 R 에, S_t 은 S 에 포함시키고 여기까지의 과정을 하나의 영역이 될 때까지 반복한다. 이때 유사도 $s(r_i, r_j)$ 은 $[0, 1]$ 의 값으로 정규화된 4가지 요소인 color, texture, size, fill 값의 가중합으로 계산한다. 식으로 표현하면 다음과 같다.

$$s(r_i, r_j) = a_1 s_c(r_i, r_j) + a_2 s_t(r_i, r_j) + a_3 s_s(r_i, r_j) + a_4 s_f(r_i, r_j)$$

이때 $a_1 = a_2 = a_3 = a_4 = \frac{1}{4}$ 로 모든 요소들의 가중치가 같다고 설정한다.

color 요소는 이미지의 색에 관한 유사도이다. 영역마다 각 컬러 채널에 대해 25개의 bin으로 구성된 히스토그램을 구한 다음, 영역끼리의 히스토그램 교집합 영역을 구해 유사도를 계산한다. 여기서 한 영역당 히스토그램의 총 차원 수는 $25 \times 3(\text{RGB})$ 의 75가 된다. 추가로, 합쳐진 영역에 대한 유사도는 사이즈를 고려하여 업데이트하는 방식을 활용한다.

texture 요소는 주변 픽셀들의 변화량에 관한 유사도이다. 각 컬러 채널에 대해 8개의 방향에 대한 가우시안 미분을 구한다. 그런 다음 10개의 bin으로 구성된 히스토그램을 구하고, 히스토그램 교집합 영역을 이용해 유사도를 계산한다. 총 차원 수는 $8 \times 10 \times 3(\text{RGB})$ 의 240이 된다.

size 요소는 각 영역들의 사이즈에 관한 유사도로, 작은 영역일수록 더 빨리 그룹화시키는 것을 의미한다. 다시 말해 사이즈가 작을수록 유사도가 높아진다. 이 유사도를 포함시킴으로써 작은 영역을 모두 포함해서 덮어 버리는 클러스터들이 많아지는 것을 예방할 수 있다.

fill 요소는 후보 bounding box와의 크기 차이에 관한 유사도이다. 후보 박스와 영역들의 크기 차이가 적을수록 유사도가 높다. 영역들은 서로 'compatible'해야 하는데, 이는 두 영역이 fit 해서 영역 제안 생성에서 생길 빈 공간들을 채우는 것이다. 즉, 서로 닿지 않는 영역은 합쳐지면 안 된다는 의미로 해석할 수 있다.

이러한 네 가지 요소에 대한 각 계산 결과를 가중치를 고려한 선형 결합을 통해 특정 두 영역에 대한 유사도를 계산하고, 이를 바탕으로 유의미한 후보 영역들을 제안한다.

선택적 탐색 알고리즘은 입력 받은 이미지에 대해 객체를 포함할 것 같은 영역들을 생성한다. 그러나 그 영역에 포함된 객체가 무엇인지에 관한 정보는 전혀 알 수 없다. 하지만 이미지 피라미드 알고리즘과 sliding

windows 알고리즘 같이 모든 경우에 대해 하나씩 비교해 보며 특정 객체가 포함되었다고 판단되는 영역들을 찾아내는 비효율적인 방법에 비해 계산적 비용을 크게 줄일 수 있다. 즉, 선택적 탐색은 SVM와 CNN 같은 최종 분류 단계에 해당하는 연산에 대해서 미리 가능성 있는 영역들을 도출해내는 방법론이라고 할 수 있다.

2.2. SSIM (Structural Similarity Index Measure)

흑백 이미지는 크게 세 가지 요소인 휘도, 대비, 구조로 구성된다.

휘도는 빛의 밝기로, 이미지 x 의 평균 휘도는 식 (1)과 같이 쓸 수 있다. 이때 x_i 은 해당 픽셀의 밝기이고 N 은 이미지 x 의 전체 픽셀 개수이다.

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

대비는 빛의 밝기가 극적으로 바뀌는 성질을 의미한다. 이미지 x 의 대비의 표준편차는 식 (2)와 같다. 표본 표준편차이기 때문에 N 대신 $N-1$ 로 나눈 값을 적용한다.

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (2)$$

또한 구조는 픽셀들의 상대적 위치에 대한 성질이다. 구조는 이미지에서 평균 밝기를 뺀 후에 표준편차로 나눈 것으로, 수식으로는 식 (3)처럼 표현할 수 있다.

$$(x_i - \mu_x) / \sigma_x \quad (3)$$

SSIM은 두 개의 이미지 혹은 이미지와 원도우를 앞에서 언급한 세 요소인 휘도, 대비, 구조를 바탕으로 비교하여 그 유사도

를 수치화한다. 이때 최종 비교 함수의 조건은 symmetric해야 하고, 그 값이 1보다 작거나 같아야 하며, 두 이미지가 같을 때만 1의 최댓값을 가져야 하는 것이다. 이 조건들을 고려하여 두 이미지 x 와 y 에 대해서 식 (4)와 같이 계산한다.

$$SSIM(x, y) = l(x, y) c(x, y) s(x, y) \quad (4)$$

휘도 비교 함수는 식 (5)와 같이 계산한다. 이때 C_1 은 $(0.01L)^2$ 을 의미하는데, L 은 픽셀들의 범위 사이즈로, 식으로 표현하면 $2^{(\# \text{ of bits per pixels}) - 1}$ 과 같다.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (5)$$

또한 대비 비교 함수는 식 (6)으로 계산하며 C_2 은 $(0.03L)^2$ 을 의미하고 L 은 휘도 비교 함수에서의 값과 동일한 값이다.

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (6)$$

구조 비교 함수는 식 (7)과 같이 계산하고, 이때 σ_{xy} 는 식 (8)과 같다.이다.

$$s(x, y) = \frac{\sigma_{xy} + C_2/2}{\sigma_x\sigma_y + C_2/2} \quad (7)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (8)$$

이 세 비교 함수를 적용하면 두 이미지의 유사도를 계산하기 위한 수식은 다음과 같이 쓸 수 있다.

$$SSIM(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \times \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

2.3 IoU (Intersection over Union)

머신 러닝 등에서 이미지의 class를 분류하는 문제에 대해서는 맞다, 틀리다의 두 가지로 명확하고 간단하게 구분할 수 있기 때문에 성능을 평가하는 것이 쉽다. 하지만 객체 탐지, 특히 객체 영역 탐지의 경우에는 예측된 영역이 정답 영역과 정확히 일치하는 경우가 거의 없기 때문에 단순히 맞다, 틀리다고만 성능을 평가하는 것에는 다소 무리가 있다.

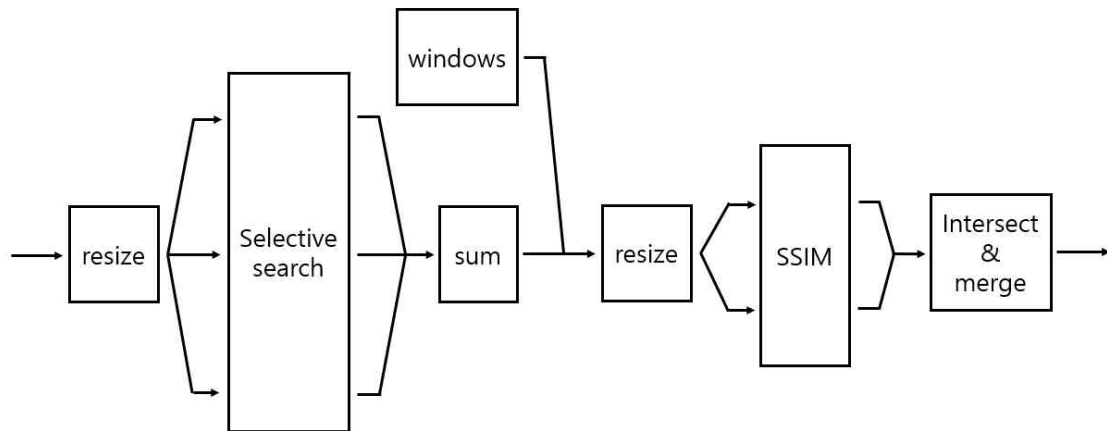
따라서 두 영역이 얼마나 일치하는지 그 정도를 수식적으로 파악하여 성능을 평가하는 IoU가 객체 탐지 모델의 정확도 측정에 있어서 많이 사용된다. 이 방식을 적용하기 위해서는 두 가지가 필요한데, 정답 영역을 나타내는 ground truth bounding box와 모델에서 예측된 영역을 나타내는 predicted bounding box가 주어져야 한다.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IoU 값은 두 박스의 교집합에 해당하는 영역의 면적을 합집합에 해당하는 영역의 면적으로 나누어서 구한다. 따라서 0과 1 사이의 값을 가지며, 그 값이 1에 가까울수록 두 박스의 위치 및 크기가 일치하는 것이기 때문에 성능이 좋은 것이라고 해석할 수 있다.



(그림 2) IoU 값에 따른 성능 평가



(그림 3) model diagram

3. 모델

본 연구는 선택적 탐색 기법을 적용하여 색상, 크기 등의 특성에 근거하여 어떠한 객체가 있을 것으로 예상되는 영역 후보들을 모두 찾고, 예측된 영역들에 대해 포도송이를 담고 있는 window 이미지들과의 일대일 유사도 비교를 통해 영역이 나타내는 객체가 포도송이로 추정되는 영역만을 결과로 도출한다. 이후 결과 영역의 중심 사이의 거리를 가로, 세로 방향을 기준으로 각각 계산하여 특정 threshold보다 가까이 위치하고 있다면 해당하는 두 영역을 통합한다. 이러한 방식으로 입력 이미지에 포함되어 있는 포도송이의 총 개수 및 그 대략적인 위치를 최대한 중복 없이 찾아내는 것이 최종 목표이다.

window 이미지 생성 및 모델의 테스트를 위해서 사용한 포도송이 이미지 및 라벨 데이터는 Embrapa Wine Grape Instance Segmentation Dataset(이하 WGISD)를 crop하여 사용했다. WGISD는 밭을 배경으로 한 다섯 종류의 포도송이에 대한 데이터로 포도의 모양, 색상, 형태 및 카메라 초점 등이 다양하게 구성되어 있다. window 이미지는

포도송이가 담긴 라벨 박스를 기준으로 잘라내어 생성한 다양한 크기로 구성된 200장으로 정의했다.



(그림 4) window 이미지 예시

전체적인 모델 구성은 그림 3과 같다.

포도송이가 포함되어 있는 512x512 사이즈의 컬러 이미지 한 장을 입력으로 받는다. 입력 이미지에 대해 resize를 적용하여 512x512, 256x256, 128x128의 세 사이즈의 이미지로 재구성한다. 그 후, 세 개의 이미지에 각각 선택적 탐색을 적용하여 객체가 있을 것으로 예측되는 후보 영역을 도출한다. 이때 선택적 탐색의 경우 같은 변수에 대해 입력 이미지의 사이즈가 작을수록 결과로 나오는 영역의 개수가 적어지는 경향

을 보인다. 이를 보정하여 한쪽으로 치우치는 것을 예방하기 위해 scale 변수의 값을 각각 200, 150, 100으로 설정해 결과의 개수를 비슷하게 맞춰 주었다. 또한 선택적 탐색으로 얻은 결과 영역에 대해 512x512 이미지를 기준으로 복구하기 위해 각각 1, 2, 4를 곱한 다음 세 결과를 통합한다. 그 과정에서 영역의 사이즈(면적)가 픽셀을 기준으로 9000보다 작거나 영역의 높이(h)와 너비(w)의 비율 즉, $h \div w$ 또는 $w \div h$ 의 값이 1.2 이상인 길쭉한 영역은 제거한다.

이렇게 얻어낸 후보 영역들의 값을 기준으로 512x512 입력 이미지를 자른다. 이렇게 생성한 이미지들을 window로 정의했던 비교용 포도송이 이미지 200장과 일대일로 SSIM에 적용하여 유사도 값을 구한다. 이때 유사도 값 계산을 위해서는 잘라진 입력 이미지의 사이즈와 window 이미지의 사이즈를 동일하게 맞춰줘야 하기 때문에, 두 가지 기준으로 resize한다. 후보 영역의 높이 및 너비, window의 높이 및 너비 총 네 개의 값 중 최솟값을 한 변으로 하는 정사각형으로 resize하는 것과 256을 한 변으로 하는 정사각형으로 resize하는 것으로 구분하여 기준을 적용한다.

또한 각 영역과 window 이미지를 흑백으로 변환하여 유사도 계산해야 한다. 포도송이 이미지의 특성상 포도는 초록색과 보라색으로 종 및 성숙도에 따라 다양한 색상이 존재하며, 특히 초록색의 경우는 잎과 줄기 또는 발과 같은 주변 환경과 색만으로는 구분하기 어렵기 때문에 흑백으로 변환하여 색상 요소를 제외하는 것이 더 유의미하다.

유사도 계산 과정에서 입력으로 넣어 준 이미지의 크기가 클수록 그 결과 값이 비교적 높은 영역이 많이 나오는 경향을 보이기 때문에 이 역시도 너무 한 쪽에만 치우치지 않도록 보정이 필요하다. 최솟값으로 resize하여 비교한 결과에는 0.25, 256으로 resize하여 비교한 결과에는 0.4의 서로 다른 threshold를 적용해 각각의 결과로 나오는

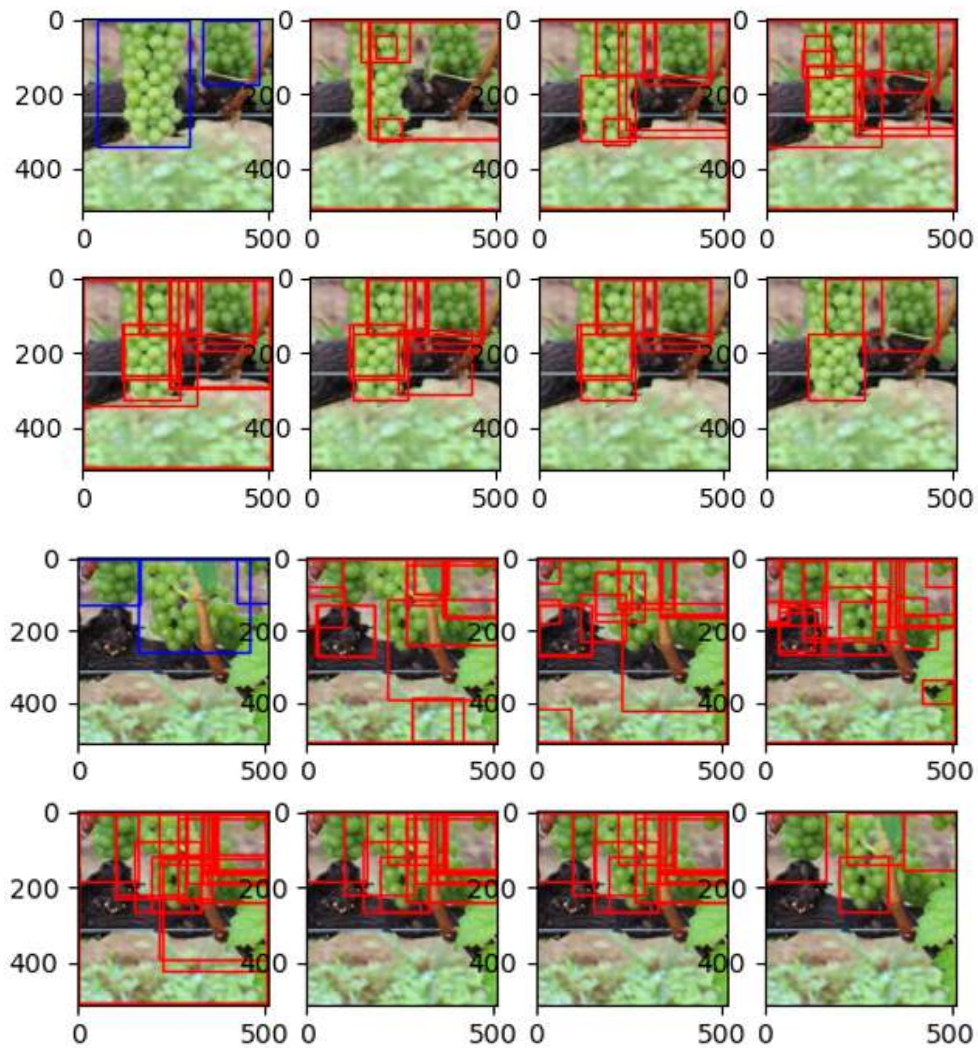
영역의 개수를 비슷하게 맞춰 준다.

마지막으로, 두 기준에 따라 계산된 유사도가 threshold 이상인 두 결과 집단에 대해서 두 집단에 공통적으로 포함된 영역만을 최종 결과로 뽑는다. 이때 이미지에 포함되어 있는 포도송이의 정확한 위치보다는 총 개수와 대략적인 위치를 구하는 것이 목표이기 때문에 각 결과 영역의 중심 픽셀과 가로, 세로 방향 기준의 거리가 100 이하에 위치한 다른 영역(중심)이 있을 경우 그 둘을 하나로 통합하여 해당 포도송이를 대표하는 영역 하나만 남겨 둔다.

4. 연구 결과

생성한 모델에 대한 테스트를 위해서 WGSD 데이터를 활용했다. 이때 window 생성에 있어서 사용되지 않은 데이터만 사용하여 총 150개의 테스트 데이터를 만들었다. 1365X2048 사이즈의 이미지를 포도송이 박스가 포함되어 있도록 그 위치를 고려하여 512x512의 이미지로 crop했고, 이에 맞추어 라벨 데이터인 이미지 내의 포도송이 위치 좌표도 재구성했다. 이때, 검토를 통해 자른 이미지에 모든 포도송이가 유의미하게 담긴 이미지만을 테스트 데이터로 사용했다. 유의미하게 담겼다는 것은 포도송이임을 육안으로도 알아볼 수 있으면서 한 영역에 한 개의 포도송이가 적절하게 들어 있으며, 영역들이 너무 구석에만 치우치지 않은 것을 의미한다.

그림 5를 보면 두 가지 입력 이미지에 따른 모델의 예측 결과를 8개의 단계로 확인할 수 있다. 각각 왼쪽 위부터 차례대로 정답 영역을 나타낸 파란색 박스, 세 사이즈의 입력 이미지에 각각 대한 선택적 탐색 결과 영역 (입력 사이즈는 왼쪽부터 차례로 512x512, 256x256, 128x128), 두 가지 resize 기준에 따른 다른 사이즈의 입력에 대한 window 이미지와의 평균 유사도가



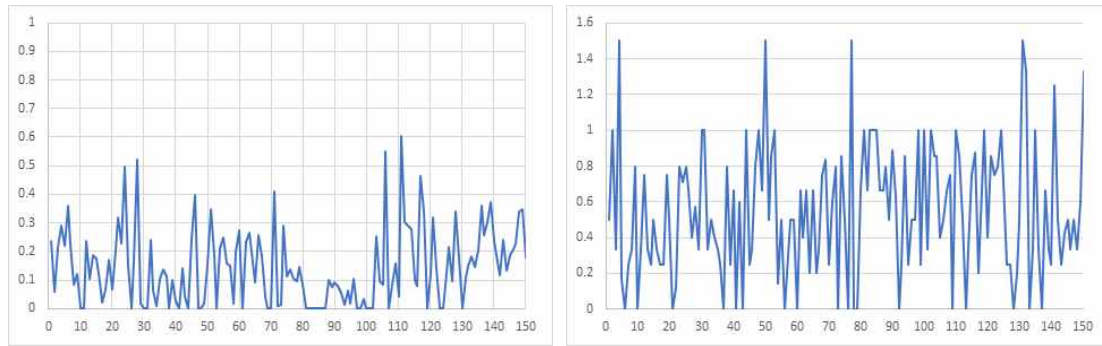
(그림 5) 모델의 예측 결과

(위쪽 8개는 입력 이미지 1에 대한 결과, 아래쪽 8개는 입력 이미지 2에 대한 결과)

threshold 이상인 영역 (resize 기준은 왼쪽 부터 min, 256), 두 유사도 연산 결과에 공통으로 등장한 영역, 각 중심을 기준으로 거리가 가까운 영역을 합친 최종 결과 영역을 나타내고 있다. 두 경우 모두 포도송이의 대략적인 위치를 비교적 정확하게 나타내고 있으며, 예측한 포도송이의 개수도 1개씩의 오차만 존재하기 때문에 모델이 잘 찾고 있음을 알 수 있다.

수치 계산을 통한 성능 측정은 총 두 가지 방식으로 진행했다. 첫 번째로는 결과 영역과 라벨 데이터의 비교를 위해 객체 탐지 및 segmentation의 대표 성능 측정 방

법인 IoU를 사용했다. 그 결과 전체 테스트 이미지에 대한 평균 IoU 값은 0.1447로 다소 낮았지만 일부 이미지는 0.5에 근접한 비교적 높은 값으로 계산되었다. 보통 IoU의 값이 0.5 이상이면 제대로 검출되었다고 판단하기 때문에 일부 이미지에 대해서는 좋은 성능을 보였다고 할 수 있다. 하지만 비교적 낮은 성능을 보인 이미지도 많았는데, 이는 이미지에 줄기와 잎들 또는 배경으로 발이 많이 등장해 질감 등의 측면에서 포도송이와의 뚜렷한 구별이 어려웠을 것으로 예상된다. 또한 일부 이미지는 다소 화질이 떨어져서 window 이미지들과의 대비



(그림 6) 150개 데이터에 대한 성능 측정 결과
(왼쪽부터 예측 영역에 대한 IoU, 예측 개수의 차이 비율)

값에서 차이가 많이 났기 때문에 threshold를 통과하지 못했을 것으로 예상된다.

두 번째로는 본 연구의 목적이었던 포도송이의 개수를 알아내는 것에 대한 성능 파악을 위해 라벨 데이터의 영역 개수와 예측 데이터의 영역 개수의 차이를 구해 비교했다. 테스트 이미지마다 라벨 데이터에 있던 영역의 개수가 다양한 폭으로 존재했기 때문에 단순히 개수의 차이로만 판단하는 것 보다는 그 차이 값을 라벨 데이터 영역의 개수로 나눈 값으로 비교하는 것이 타당하다. 전체 테스트 이미지에 대한 평균은 0.5452로 본 모델이 보통 0.5배에서 1.5배의 예측 영역을 결과로 도출하는 것으로 해석된다. 한 포도송이에 대해 두 개 이상의 영역이 도출되었으나 거리 차이에 의해 통합되지 못한 영역들에 의해 오차가 커졌을 것으로 예상된다.

5. 결론

본 연구에서는 비교적 구조가 간단한 선택적 탐색 알고리즘을 활용하여 얻은 예측 결과에 대해 포도송이 이미지의 특징을 대표하는 window들과의 휘도, 대비, 구조적 유사성을 비교하는 부분만을 추가한 모델을 생성함으로써 빠르고 연산 부담이 적으면서도 신뢰성 있는 포도송이 영역의 검출이 가

능하다는 것을 확인했다.

기존에 객체 탐지의 예측 정확도를 높이기 위해 사용되었던 복잡한 딥러닝 및 네트워크 구조를 사용하지 않고도 어느 정도 원하는 객체의 개수 파악 및 각각의 대략적인 위치 파악이 가능하다는 것은 시간과 연산 등 다양한 자원적으로 봤을 때 큰 장점이다. 더 나아가 거의 실시간의 빠른 속도로 계산 및 처리 되어야 하는 감시와 감지 등의 분야에 대해서도 객체 인식 관련 기술이 시간적 패널티 없이 충분히 적용될 수 있을 것이라는 가능성을 제시하고 있다.

추후 연구에서는 본 모델의 정확성을 보다 높이기 위한 시도가 이루어져야 할 것이다. 우선 최종 결과 영역을 구하는 과정에서 가까운 영역과 결합할 때, 둘 중 하나로 대표하는 방식이 아닌 두 영역을 포함하는 새 영역을 생성하는 방식을 사용한다면 모델의 성능도 더 향상될 뿐만 아니라 본 연구에서 해결하지 못한 같은 포도송이 객체를 나타내지만 통합되지 않았던 부분에 대한 답을 찾을 수 있을 것이라고 예상된다. 또한 본 모델의 유사도 계산 부분을 응용하여 나뭇잎 및 줄기 및 배경 요소와 매우 유사한 영역은 결과에서 제외하는 부분을 추가한다면 포도송이가 아님에도 최종 예측 결과에 남아있었던 잘못 예측된 영역들도 제거되어 정확도를 더 높일 수 있을 것으로 기대된다.

참고문헌

- F. F. Pedro and P. H. Daniel, "Efficient Graph-Based Image Segmentation", International Journal of Computer Vision, 2004.
- H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression", In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.
- J. S. Lee, S. K. Lee, D. W. Kim, S. J. Hong and S. I. Yang, "딥러닝 기반 객체 인식 기술 동향", Electronics and Telecommunications Trends, Vol.33(2018), 23-32.
- J. Uijlings, K. van de Sande, T. Gevers and A. Smeulders, "Selective Search for Object Recognition", International Journal of Computer Vision, 2013.
- K. E. Van de Sande, J. R. Uijlings, T. Gevers and A. W. Smeulders, "Segmentation as Selective Search for Object Recognition", In 2011 international conference on computer vision, IEEE, 2011.
- T. T. Santos, L. L. de Souza, A. A. dos Santos and S. Avila, "Grape Detection, Segmentation, and Tracking using Deep Neural Networks and Three-dimensional Association", Computers and Electronics in Agriculture, Vol.170(2020).
- Z. Wang, C. B. Alan, R. S. Hamid and P. S. Eero, "Image Quality Assessment: From Error Measurement to Structural Similarity", IEEE, Vol.13, No.1(2004).