**평가하기**

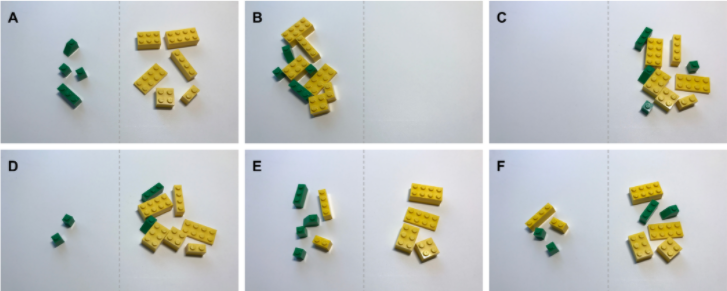
**1. 분류하기**

아래 레고 블록 중 상단 돌기의 수가 홀수인 것을 골라 왼쪽으로 놔두고, 짝수는 오른쪽으로 놔두세요.



홀수가 4개, 짝수가 6개

우리가 평가할 모델이 여섯 개이고, 각 모델의 결과가 다음과 같다고 가정해보자.



우리가 평가할 모델이 여섯 개이고, 각 모델의 결과가 다음과 같다고 가정해보자.

어떤 모델의 결과가 가장 좋을까? 정답은 A모델이다.

홀수 블록과 짝수 블록을 정확하게 구분하고 있다.

그럼 다음으로 좋은 모델은 무엇일까? 각 모델 결과의 특징을 보자.

B모델 : 모두 홀수로 분류

C모델 : 모두 짝수로 분류

D모델 : 홀수라고 분류한 것에는 홀수만 있지만, 일부 홀수를 놓침

E모델 : 짝수라고 분류한 것에는 짝수만 있지만, 일부 짝수를 놓침

F모델 : 홀짝이 모두 섞여있음

**정확도**(정확도를 평가할 때에는 클래스의 분포도를 꼭 확인해야 한다.)

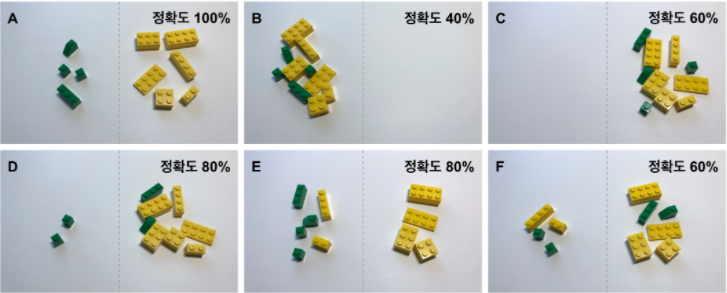
정확도란 전체 개수 중 홀수를 홀수라고 맞추고(양성을 양성이라 말하고), 짝수를 짝수라고 맞춘(음성을 음성이라고 말한) 개수의 비율이다.

B모델은 전체 10개 중에 홀수 블록 4개만 맞추었으므로 정확도는 40%이다.

C모델은 전체 10개 중에 짝수 블록 6개만 맞추었으므로 정확도가 60%이다.

무조건 한 쪽으로 분류하더라도 클래스의 분포에 따라 높게 나올 수 있다.

만약 남자고등학교에서 남녀를 구분하는 모델을 개발한다고 했을 때, 그 모델이 무조건 남자인 결과를 내놓는다고 가정해보자. 실제 여자가 있더라고 모두 남자라고 분류를 하겠지만 정확도는 90%가 넘을 것이다. 그렇다고 이 모델이 좋다고는 할 수 없다.

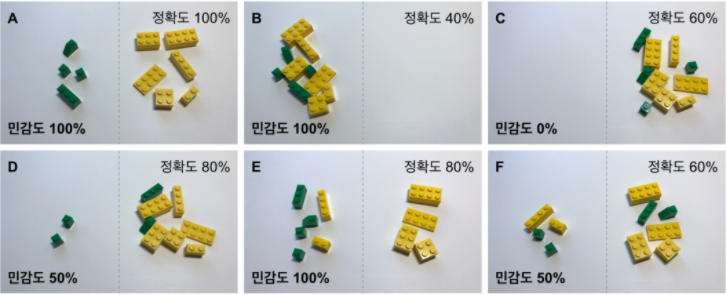


D모델과 E모델은 같은 정확도를 가지고 있지만 만약 홀수 블록(양성)은 빠짐없이 모두 골라내길 원한다면 E모델이 더 적합하다. 양성을 잘 골라낼 수 있는 능력을 평가하기에는 정확도가 아닌 다른 기준이 필요하다.

바로 민감도입니다.

**민감도**(민감도 = 판정한 것 중 실제 양성 수 / 전체 양성 수)

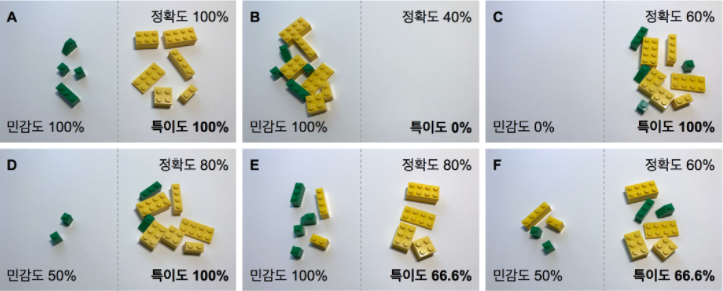
민감도는 양성에 얼마나 민감하냐?라는 의미입니다. 양성을 양성이라고 판정을 잘 할수록 이 민감도가 높습니다.



다시 D모델와 E모델을 보겠습니다. D모델은 홀수를 절반만 골랐으므로 50%이고, E모델은 홀수를 모두 골랐으므로 100%입니다. 정확도가 동일한 모델 중에 홀수를 잘 골라내는 모델을 선정하고 싶다면, 민감도가 높은 것을 선택하시면 됩니다. 주의깊게 봐야할 부분은 모두 홀수라고 판정하는 B모델도 민감도가 100%이기 때문입니다.

**특이도**(특이도 = 판정한 것 중 실제 음성 수 / 전체 음성 수)

특이도는 얼마나 특이한 것만 양성으로 골라내느냐?입니다. 이말은 특이한 것만 양성으로 골라내니 반대로 음성을 음성이라고 잘 판정한다고 볼 수 있습니다.



D모델과 E모델을 다시 보자. D모델은 E모델에 비해 민감도는 낮지만 특이도는 높다. 만약 음성을 음성이라고 잘 골라내는 모델이 필요하다면 D모델을 선정해야 한다.

**정확도, 민감도, 특이도 비교**

| **구분** | **모델 A** | **모델 B** | **모델 C** | **모델 D** | **모델 E** | **모델 F** |
| --- | --- | --- | --- | --- | --- | --- |
| 맞춘 홀수(전체 4개) | 4개 | 4개 | 0개 | 2개 | 4개 | 2개 |
| 맞춘 짝수(전체 6개) | 6개 | 0개 | 6개 | 6개 | 4개 | 4개 |
| 정확도 | 100% | 40% | 60% | 80% | 80% | 60% |
| 민감도 | 100% | 100% | 0% | 50% | 100% | 50% |
| 특이도 | 100% | 0% | 100% | 100% | 66.6% | 66.6% |

맞춘 개수는 다르지만 같은 평가지수도 있고, 맞춘 개수는 같지만 평가지수가 다른 것들이 보인다. 어떤 모델이 적합한 지는 문제에 따라 다르니 곰곰히 생각해 보자. 도움될 만한 몇가지 예를 들어봤습니다.

- 공항검색기기는 일반물건을 위험물건이라도 잘못 판정하더라도 위험물건은 반드시 찾아야 합니다. 즉 민감도가 높아야 한다.

- 쇼핑 시에는 꼭 필요한 물건만 사야한다. 사야할 물건도 경우에 따라 사지 않을 수 있지만 사지 않아야 하는 물건을 반드시 안 사야한다. 즉 특이도가 높아야 한다.

- 지진이 나고 나면, 다음날 지진을 느낀 사람과 그렇지 않은 사람이 있을 것이다. 어떤 사람(A)은 민감해서 지진도 아닌 진동도 느끼지만 왠만한 지진은 모두 느끼는 사람이 있는 반면, 어떤 사람(B)은 정말 강도가 높은 지진이 아니고서야 왠만해서는 느끼지 못하는 사람이 있을 것이다. 이 경우 다음과 같이 생각할 수 있다.

#A가 지진을 못 느꼈다고 하면, 그날은 지진이 발생하지 않은 것입니다. 왜냐하면 A는 민감도가 높아 왠만한 지진은 다 알아내기 때문이다.

#B가 지진을 느꼈다고 하면, 그날은 지진이 발생한 것입니다. 왜냐하면 B는 특이도가 높아 지진이 발생하지 않은 것은 다 알아내기 때문이다.

**좀 더 살펴보기**

각 블록을 판정할 때는 통상적으로 모델에서는 해당 블록일 확률로 결과가 나온다. 즉 이 블록은 홀수일 확률이 60%이야 또는 40%이야 이런식으로 말이죠. 이 확률로 판정결과를 나타내기 위해서 50%가 기준이 되어, 50% 이상이면 홀수 블록이다 라고 얘기하는 것이죠. 우리는 이 50%를 임계값(threshold)라고 부른다. 지금까지 위에서 봤던 결과들은 모두 확률값을 임계값을 기준으로 판정을 한 것이다.

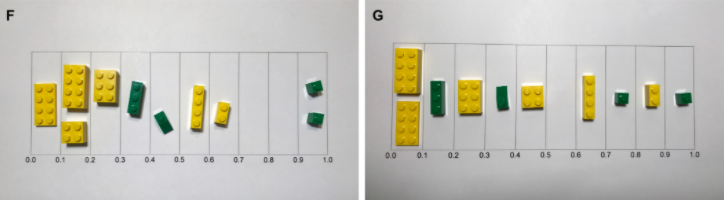
F모델 결과를 보도록 하겠다.

총 10개 블록 중 홀수 2개, 짝수 4개를 맞추었으므로 60%의 정확도를 가지고 있다. 민감도는 총 4개의 홀수 블록 중 2개를 맞추었으니 50%이다.

특이도는 총 6개의 짝수 블록 중 4개를 맞추었으니 66.6%이다.

그리고 이와 동일한 정확도, 민감도, 특이도를 가진 모델G가 있다고 가정해보자.

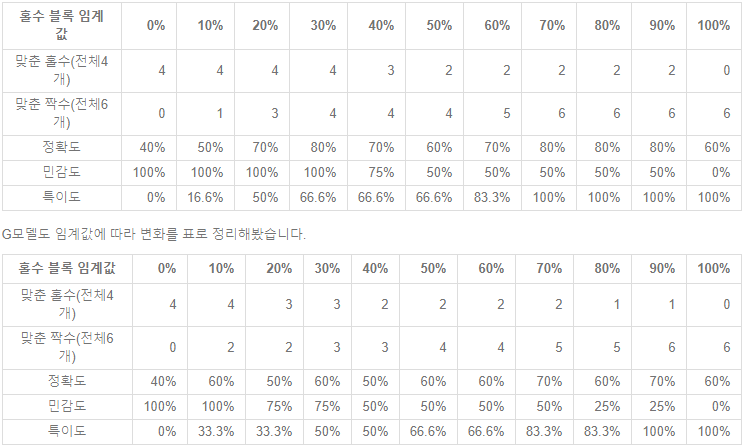
F모델과 G모델이 주어진 블록에 대해 홀수라고 판정할 확률값을 이 오름차순으로 나열한 뒤 10% 단위로 표시된 칸에 배치를 해봤다.



먼저 F모델을 보겠다. 왼쪽의 첫번째 블록이 홀수 블록일 확률이 5%라고 가정해보자. 그래서 0.0과 0.1 사이 칸에 위치 시켰다. 0.5가 임계값이라고 한다면, 맞춘 홀수 블록은 0.5 임계값에서 오른쪽에 있는 2개이고, 맞춘 짝수 블록은 0.5 임계값 왼쪽에 있는 4개이다.

이 0.5인 임계값을 조정하면 어떻게 될까? 임계값이 0.0이라면, 모두 홀수 블록이라고 하는 것이므로, 맞춘 홀수 블록은 4개가 되고, 맞춘 짝수 블록은 0개가 된다. 따라서, 임계값을 조정하면 정확도, 민감도, 특이도가 바뀐다.

10% 단위로 임계값를 변화시키면서 바뀌는 정확도, 민감도, 특이도를 표로 정리하면 다음과 같다.



어느 모델이 더 좋을까?

대충보면, 모델 F가 더 좋아보인다. 모델 F가 모델 G보다 홀수 블록이 홀수일 확률이 높은곳에 배치되어 있고, 짝수블록이 홀수일 확률이 낮은 곳에 배치되어 있기 때문이다. 이런 패턴을 보기 위해서 많이 사용되는 것이 ROC(Receiver Operating Characteristic) curve 이다.

이는 민감도와 특이도가 어떤 관계를 가지고 변하는 지 그래프로 표시한 것이다. 이러한 ROC curve 아래 면적을 구한 값을 AUC(Area Under Curve)이라고 하는데, 하나의 수치로 계산되어서 성능비교를 간단히 할 수 있다.

ROC curve를 그리는 방법은 간단하다. 각 임계값별로 민감도와 특이도를 계산하여 x축을 (1-특이도), y축을 민감도로 두어서 이차원 평면 상에 점을 찍고 연결하면 된다.

모델 F와 모델 G에 대해서 ROC Curve를 그리는 소스코드는

「DeeplearningConception\_06.py」에 있다.

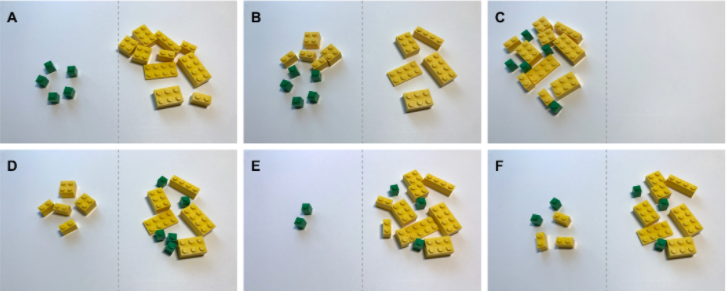
**2. 검출 및 검색하기**

아래 레고 블록 중 상단 돌기의 수가 하나인 것만 골라 왼쪽으로 놔두세요.



상단돌기가 하나인 블록은 5개.

우리가 평가할 모델이 여섯 개이고, 각 모델의 결과가 다음과 같다고 가정해보자.



어떤 모델의 결과가 가장 좋을까? 정답은 A모델이다.

하나인 블록을 모두 골랐다. 각 모델 결과도 살펴보자.

B모델 : 하나인 것을 모두 골랐지만 아닌 것도 있음

C모델 : 모두 하나인 것이라고 고름

D모델 : 하나인 것이라고 고른 것 중 진짜 하나인 것은 없음

E모델 : 하나라고 고른 것은 모두 진짜이나 하나인 것을 모두 고르지는 못함

F모델 : 하나인 것을 모두 고르지 못했고, 고른 것들에도 진짜가 아닌 것도 있음

A모델과 B모델 중 어느 것이 좋을까?

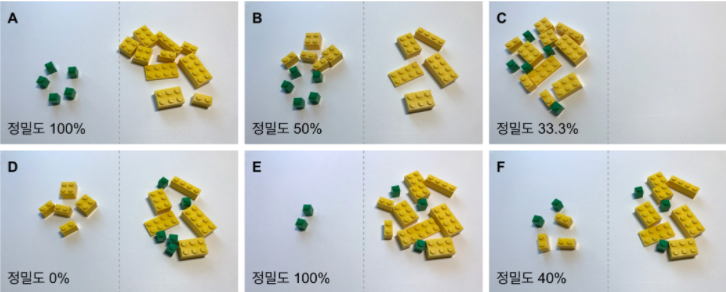
B모델도 전체 양성 5개 중 5개를 모두 골랐지만 하나가 아닌 것들도 골랐다.

이를 구분하기 위한 기준으로 정밀도라는 것을 사용한다.

**정밀도**(정밀도 = 실제 양성 수 / 양성이라고 판정한 수)

정밀도는 모델이 얼마나 정밀한가?이다. 즉 진짜 양성만을 잘 고를 수록 정밀도가 높다.

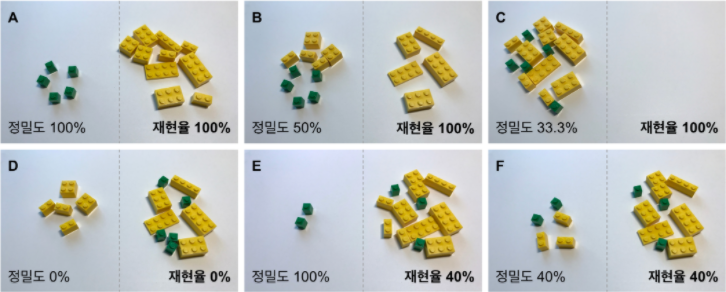
A모델은 5개 고른 것 중 실제 양성이 5개이므로 정밀도가 100%이고, B모델은 10개 고른 것 중 실제 양성이 5개이므로 정밀도는 50%가 된다. E모델은 A모델와 같이 정밀도가 100%이지만 E모델에서 양성을 모두 검출하지는 못했다. E모델보다 A모델이 더 좋은 성능을 가지고 있지만 정밀도만으로는 성능차이를 나타낼 수 없다.

이를 나타내기 위해 재현율이라는 기준을 사용한다.

재현율(재현율 = 검출 양성 수 / 전체 양성 수)

재현율은 양성인 것을 놓치지 않고 골라내는가?이다.

양성을 많이 고를수록 재현율이 높다.



앞서 본 E모델의 재현율은 전체 양성 5개 중에 2개만 검출하였으므로 40%이다. F모델도 전체 양성 5개 중 2개만 검출하였으므로 재현율은 똑같이 40%입지만, 정밀도에서 차이가 난다. 두 개의 모델만 비교한다면 F모델이 E모델보다 더 좋은 성능을 가졌다고 볼 수 있다. 또 하나 짚고 넘어가야 할 것이 C모델입니다. 재현율은 양성을 얼마나 잘 검출하는가를 평가하는 것이기에 모두 양성이라고 하는 C모델인 경우에도 100% 재현율을 가지게 된다.



「**검출 문제**」는 「**분류 문제**」와 어떤 차이가 있을까?

-> 검출 문제에서는 검출되지 않은 진짜 음성에 대해서는 관심이 없다.

Ex) 초등학교 소풍가서 하던 보물찾기 게임을 예를 들어 복습을 해보자.

우리 고생하시는 선생님들이 보물 10개를 숨겨두었다고 가정해보자.

철수는 보물을 5개가지고 왔는데, 모두 보물이었다. -> 정밀도 100%, 재현율 50%

영희는 보물이라고 100개를 가지고 왔는데, 그 중 보물은 하나였다.

-> 정밀도 1%, 재현율 10%

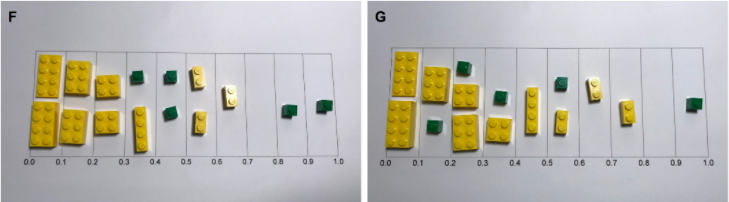
**좀 더 살펴보기**

앞서 언급한 것과 같이 각 블록을 판정할 때는 통상적으로 모델에서는

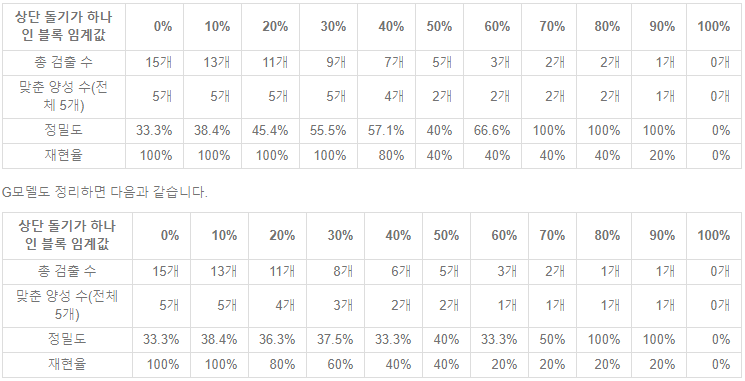
한개짜리 블록일 확률로 나오고, 이 확률값으로 판정을 하는데,

판정 기준을 임계값(threshold)이라고 불렀다. 지금까지 위에서 봤던 결과들은 모두 확률값을 임계값을 기준으로 판정을 한 것이다.

F모델과 동일하게 정밀도 40%, 재현율 40%을 가진 G모델이 있다고 가정해보자. F모델과 G모델의 판정결과 이전에 확률값을 오름차순으로 나열한 뒤 10% 단위로 표시된 칸에 배치를 해봤다.



F모델의 판정 기준인 임계값을 10% 단위로 변화시키면서 바뀌는 정밀도와 재현율을 표로 정리하면 다음과 같다.



어떤 모델이 더 좋을까?

검출 문제에서 이런 패턴을 보기 위해 사용되는 것이 「Precision-Recall Graph」 이다.

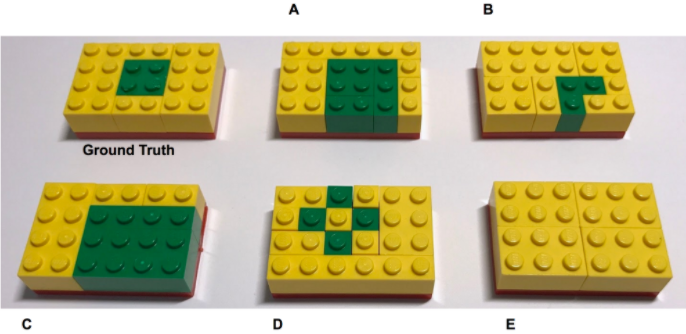
이 그래프는 x축을 재현율로 y축을 정밀도로 두어 이차원 평면상에 결과를 표시한 것이다.

이러한 그래프를 그리기 위한 소스코드는

「DeeplearningConception\_07.py」에 있다.

**3. 분할하기**

아래 사진의 Ground Truth을 보고 똑같이 만들어보세요. 즉 전체 영역에서 녹색블록과 노란블록을 구분해보세요.



어느 모델의 결과가 가장 좋을까?

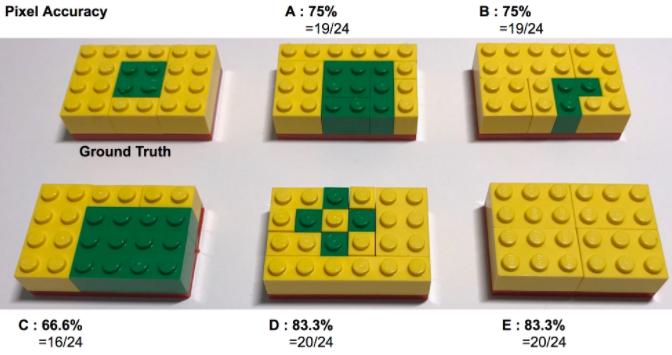
가장 쉽게 픽셀 정확도(Pixel Accuracy)를 가지고 판단할 수 있다.

픽셀은 이미지 처리에서 나오는 용어지만 여기서는 블록 하나라고 해보자.

여기서 다루는 클래스는 녹색과 노란색으로 두 개이다.

픽셀 정확도 계산

Pixel Accuracy = (녹색 블록 맞춘 수 + 노란색 블록 맞춘 수) / 전체 블록 수



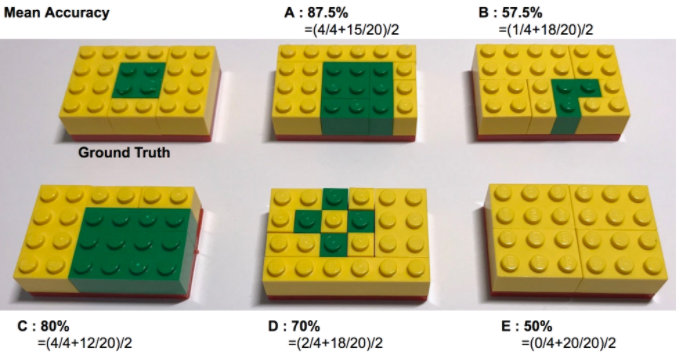
단순히 맞춘 개수를 보고 계산하기 때문에 서로 다른 패턴이라도 동일 정확도로 나오는 경우가 많다. E모델은 녹색 블록이 하나도 없는데도 83.3%이나 결과가 나온다. 색상별로 어느정도 맞춰야 좋은 평가를 얻게 되려면 어떻게 해야할까?

클래스 별로 픽셀 정확도를 계산하는 방법인 평균 정확도(Mean Accuracy)가 있다. 이는 색상별로 픽셀 정확도를 계산하여 평균을 구한 값이다.

Mean Accuracy 계산

Mean Accuracy = ((녹색 블록 맞춘 수 / 전체 녹색 블록 수) +

(노란색 블록 맞춘 수 / 전체 노란색 블록 수)) / 2



앞서 봤던 E모델은 가장 낮은 평가를 받았다. 노란색 블록의 픽셀 정확도는 100%이지만 녹색 블록의 픽셀 정확도 0%이기 때문에 평균인 50%가 되었다.

A모델과 C모델이 상대적으로 높은 평가를 받았다.

녹색 블록의 비중이 낮지만 픽셀 정확도는 100%이기 때문에 전체 평균값이 올라갔다. C모델인 경우에는 노란색 블록이 많이 틀렸음에도 80%나 되는 평가를 받았다.

이러한 정확도는 맞춘개수만 카운팅이 되고 틀린개수에 대한 패널티가 없기 때문이다. 그럼 틀린블록에 대한 패널티는 어떻게 고려할까?

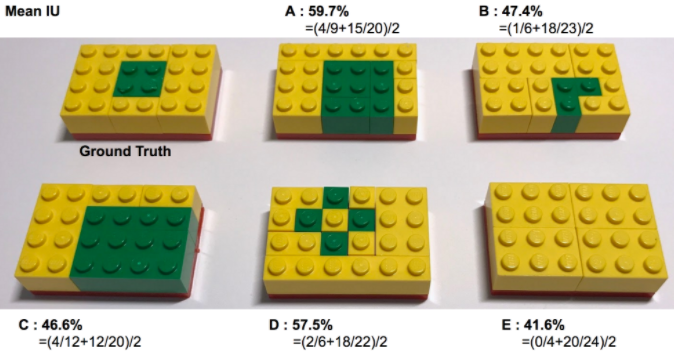
이를 고려한 평가방법으로 MeanIU라는 것이 있다.

MeanIU계산

Mean IU = (녹색 블록 IU + 노란색 블록 IU) / 2

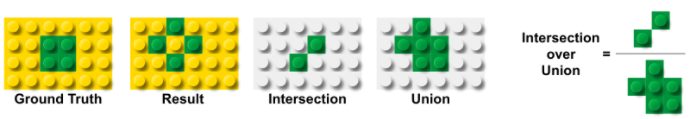
IU는 Intersection over Union의 약자로 특정 색상에서의 실제 블록과 예측 블록 간의 합집합 영역 대비 교집합 영역의 비율이다.

MeanIU는 색상별로 구한 IU의 평균값을 취한 것이다.



IU 개념에 대해서 조금 더 살펴보자.

D모델을 예를 들어, 아래 그림을 보시면 Ground Truth의 녹색 블록 영역과 모델 결과 녹색 블록 영역에서 서로 겹치는 블록 수가 2개이고, 서로 합한 영역의 블록 수가 6개이기 때문에, 녹색 블록 IU는 2/6이다. 노란색 블록 IU는 18/22입니다. 따라서 Mean IU는 이를 평균 취한 57.5%가 된다.

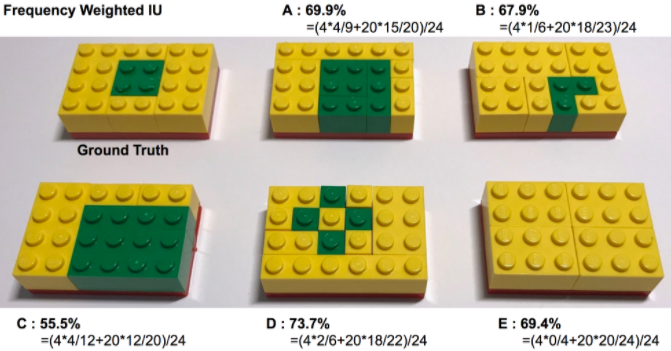


Pixel Accuracy와 Mean Accuracy인 경우에는 틀린블록에 대한 고려가 없었지만, Mean IU인 경우에는 틀린블록 수가 많을 수록 분모가 커지기 때문에 전체수치는 낮아진다. 클래스별로 IU를 구한 뒤 평균을 취하기 때문에 비중이 낮은 클래스라도 IU 수치가 낮으면 Mean IU 값도 떨어진다. 만약 클래스별로 픽셀 수가 다를경우 픽셀 수가 많은 클래스에 더 비중을 주고 싶다면 Frequency Weighted IU를 사용한다.

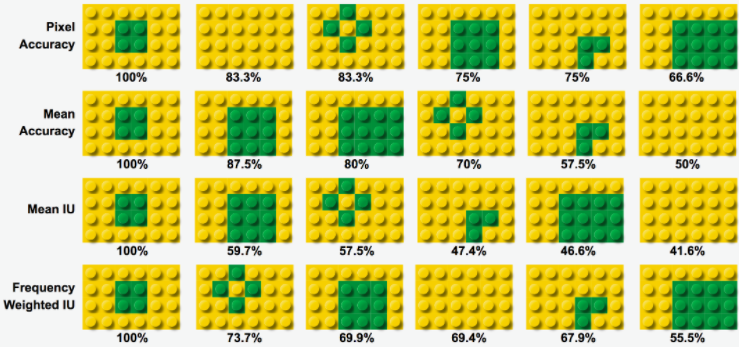
Frequency Weighted IU 계산

Frequency Weighted IU = ((녹색 블록수 \* 녹색 블록 IU) +

(노란색 블록수 \* 노란색 블록 IU))/전체수



앞서 살펴본 4가지 평가기준에 대해서 정리



결론

「**분류**」, 「**검출 및 검색**」, 「**분할**」 문제를 정의해보고, 임의의 모델의 결과를 기초적인 기준 몇가지로 평가해봤다. 딥러닝 모델을 기존 도메인에 적용하기 위해서는 아무리 강조해도 지나침이 없는것이 평가인 것 같다. 도메인마다 사용하는 용어도 다르고 평가 기준도 다르지만, 왜 그런 평가 기준이 나왔는지에 대해서 곰곰히 생각해보고 차근차근 계산해보면, 자신이 개발한 딥러닝 모델이 인정받을 날이 온다.