**Box loss**

**Box loss**, modelinizin tahmin ettiği sınırlayıcı kutuların (bounding boxes) gerçek kutularla ne kadar hizalı olduğunu ölçer. Box loss'un 1.379 olması, modelinizin tahminleri ile gerçek değerler arasında belirli bir fark olduğunu gösterir. Ancak, bu değer modelin iyi veya kötü olduğunu belirlemek için tek başına kullanılamaz. Genel olarak, daha düşük bir box loss değeri, modelin sınırlayıcı kutuları daha doğru bir şekilde tahmin ettiği anlamına gelir.

İyi bir eğitim için **box loss**'un belirli bir aralıkta olması gerektiği söylemek zordur çünkü bu, kullanılan veri setine, modelin karmaşıklığına ve belirli bir uygulamanın gereksinimlerine bağlıdır. Ancak genel bir kural olarak, box loss'un eğitim süresince düşmesini ve bir noktada sabitlenmesini bekleyebiliriz. Bu, modelin verilerden öğrendiği ve performansının iyileştiği anlamına gelir.

**Cls\_loss**

**Cls\_loss**, modelinizin tahmin ettiği nesne sınıflarının gerçek sınıflarla ne kadar hizalı olduğunu ölçer. Bu, genellikle bir çapraz entropi kaybı (cross entropy loss) ile hesaplanır. Eğitim sürecinde, modelin amacı bu kaybı minimize etmektir.

Cls\_loss'un 1.125 olması, modelinizin tahminleri ile gerçek sınıflar arasında belirli bir fark olduğunu gösterir. Ancak, bu değer modelin iyi veya kötü olduğunu belirlemek için tek başına kullanılamaz. Genel olarak, daha düşük bir cls\_loss değeri, modelin sınıfları daha doğru bir şekilde tahmin ettiği anlamına gelir.

İyi bir eğitim için cls\_loss'un belirli bir aralıkta olması gerektiğini söylemek zordur çünkü bu, kullanılan veri setine, modelin karmaşıklığına ve belirli bir uygulamanın gereksinimlerine bağlıdır. Ancak genel bir kural olarak, cls\_loss'un eğitim süresince düşmesini ve bir noktada sabitlenmesini bekleyebiliriz.

**DFL (Distribution Focal Loss)**

DFL loss, modelin gerçek dağılım ile tahminleri arasında belirli bir fark olduğunu gösterir. Daha düşük bir DFL loss değeri genellikle modelin dağılımı daha doğru bir şekilde tahmin ettiği anlamına gelir.

İyi bir eğitim için DFL loss'un belirli bir aralıkta olması gerektiğini söylemek zordur çünkü bu, kullanılan veri setine, modelin karmaşıklığına ve belirli bir uygulamanın gereksinimlerine bağlıdır. Ancak genel bir kural olarak, DFL loss'un eğitim süresince düşmesini ve bir noktada sabitlenmesini bekleyebiliriz. Bu, modelin verilerden öğrendiği ve performansının iyileştiği anlamına gelir.

**Instances**

Instances değeri, belirli bir eğitim döneminde (epoch) işlenen örneklerin (nesnelerin) sayısını temsil eder. Bu değer, eğitim sürecindeki ilerlemenizi ve işlenen veri miktarını anlamanıza yardımcı olabilir.

37 instances, modelinizin belirli bir epoch boyunca 37 farklı nesneyi işlediğini gösterir. Bu, toplam veri setinizin büyüklüğüne ve belirli bir eğitim döngüsünde işlenen batch sayısına bağlıdır.

Instances değerinin "iyi" veya "kötü" olup olmadığını belirlemek için kullanılmaz. İyi bir eğitimi yansıtıp yansıtmadığına dair bir aralık belirlemek mümkün değildir, çünkü bu değer genellikle eğitim sürecindeki ilerlemeyi takip etmek için kullanılır, modelin performansını ölçmek için değil.

**Precision**

"Box(P)" metriği genellikle "Precision" (Hassasiyet) için kullanılır. Hassasiyet, doğru tahmin edilen pozitiflerin (gerçekten de pozitif olan ve pozitif olarak tahmin edilen örnekler), toplam pozitif tahminlerin (gerçek pozitifler ve yanlış pozitifler) sayısına oranıdır. Yani, modelinizin tahminlerinin ne kadar doğru olduğunu ölçer.

Hassasiyet, genellikle 0 ile 1 arasında bir değerdir. **1, mükemmel bir hassasiyeti ifade eder - yani model, tüm tahminlerinde doğru. 0 ise tam tersi durumu ifade eder - model, hiçbir tahmininde doğru değil**. Genelde, hassasiyet değeri ne kadar yüksekse, modelin performansı o kadar iyidir.

**Recall**

"Box(R)" metriği genellikle "Recall" (Duyarlılık veya Geri Çağırma) anlamına gelir. Recall, doğru tahmin edilen pozitiflerin (gerçekten de pozitif olan ve pozitif olarak tahmin edilen örnekler), tüm gerçek pozitiflerin (gerçek pozitifler ve yanlış negatifler) sayısına oranıdır. Yani, modelinizin tüm pozitif örnekleri ne kadar iyi "hatırladığını" veya tespit ettiğini ölçer.

Recall değeri de genellikle 0 ile 1 arasında bir değerdir. 1, mükemmel bir recall değerini ifade eder - yani model, tüm gerçek pozitif örnekleri doğru bir şekilde tespit eder. 0 ise tam tersi durumu ifade eder - model, hiçbir gerçek pozitif örneği doğru bir şekilde tespit edemez. Genellikle, recall değeri ne kadar yüksekse, modelin performansı o kadar iyidir. Ancak, "iyi" bir recall değeri belirlemek, kullanılan veri setine, modelin karmaşıklığına ve belirli bir uygulamanın gereksinimlerine bağlıdır.

**Mean Average Precision**

"mAP" terimi, "mean Average Precision"ın kısaltmasıdır ve genellikle bir nesne algılama modelinin performansını ölçmek için kullanılır.

mAP50: Bu metrik, Intersection over Union (IoU) eşiği 0.5 olarak belirlendiğinde tüm sınıfların ortalama Precision değerini ifade eder. Yani, bir tahminin doğru olarak kabul edilmesi için, tahminin ve gerçek değerin kesişim bölgesinin, birleşim bölgesine oranı en az 0.5 olmalıdır.

mAP50-95: Bu metrik, IoU eşiği 0.5'ten 0.95'e 0.05 adımlarla artırıldığında tüm sınıfların ortalama Precision değerini ifade eder. Yani, bir tahminin doğru kabul edildiği IoU eşiği değerleri çok daha sıkıdır. Bu, genellikle COCO veri seti benzeri benchmarklarda kullanılan bir metriktir ve genellikle modelin algılama performansını daha kapsamlı bir şekilde ölçer.

Her iki metrik için de, değerler 0 ile 1 arasında olup, 1'e daha yakın olması daha iyi bir model performansı anlamına gelir. Bununla birlikte, "iyi" bir mAP değeri belirlemek, kullanılan veri setine, modelin karmaşıklığına ve belirli bir uygulamanın gereksinimlerine bağlıdır.

**Semantic segmentation** ve **instance segmentation**, görüntüleri analiz etmek ve anlamak için kullanılan iki önemli tekniktir. Bu iki yaklaşım arasındaki temel fark, nesnelerin ayrıştırılmasının nasıl yapıldığıdır.

**Semantic Segmentation:** Bu teknikte, görüntüdeki her piksel belirli bir sınıfa atanır. Örneğin, bir otomobilin resmi semantic segmentation kullanılarak işlendiğinde, tüm otomobil pikselleri "otomobil" olarak işaretlenir. Ancak, semantic segmentation tekniği bir görüntüde birden fazla aynı sınıfa ait nesnenin bulunduğu durumlarda onları ayrı ayrı tanımlayamaz. Yani, aynı görüntü üzerinde birden fazla otomobil olduğunda, tüm otomobiller "otomobil" etiketi altında birleştirilir ve aralarında fark gözetilmez.

**Instance Segmentation:** Bu teknik, semantic segmentation'ın bir adım ötesine giderek, her bir nesneyi ayrı ayrı belirler. Yani, bir görüntüde birden fazla otomobil olduğunda, her otomobil ayrı bir instance (örnek) olarak işaretlenir. Bu durumda, her bir otomobilin kendine ait, birbirinden farklı bir maske (mask) oluşturulur. Bu sayede aynı sınıfa ait farklı nesneler birbirinden ayrıştırılır.