

Transfer öğrenmenin üç çeşidi vardır:

- 1 Kaynak ve hedef görevlerinin farklı olduğu endüktif
- 2 Kaynak ve hedef alanının farklı, görevlerin aynı olduğu transdüktif
- 3 Hedef öğrenmenin farklı ve verinin etiketli olmadığı gözetimsiz

Çalışmada kaynak ve hedef görevleri farklıdır, ayrıca her iki alanda da etiketli veri bulunmaktadır bu yüzden endüktif transfer öğrenme tekniği uygulanacaktır. Bu çalışmada transfer öğrenmenin tercih edilmesindeki iki ana sebep; (i) transfer öğrenmenin çok büyük veri setlerine ihtiyaç duymaması, (ii) sadece son katman ağırlıkları eğitildiği için düşük hesaplama gücü ve eğitim süresinin oluşmasıdır.

Çalışmadaki transfer öğrenme, aşağıdaki adımlar ile uygulanmaktadır:

- 1 Kaggle'dan kask veri setini edin.
- 2 Ön eğitilmiş ağı seç.
- 3 Son katmanı probleme uygun olarak değiştir.
- 4 Eğitim görüntülerindeki yeni sınıf sayısını belirt (çalışmada 2).
- 5 Ağa uygun olarak görüntüleri yeniden boyutlandır, eğitim ayarlarını tamamla (learning rate, batch size, epoch, weight decay, momentum, optimizer vs.) ve eğitimi gerçekleştir.
- 6 Eğitilen ağı test veri setinde değerlendir.

Transfer öğrenme işleminde ön eğitilmiş modelin seçilmesi ve problem boyutu, benzerliği adımlarına dikkat edilmelidir. Seçim işlemi ön eğitilmiş modelin hedef problemle ne kadar ilgili olduğuna bağlıdır. Hedef veri seti küçük (1.000 görüntüden daha az) ve kaynak veri setine benzer (inşaat alanı, şapkalar vs.) ise aşırı öğrenme şansı yüksektir. Bu yüzden çalışmada, hedef veri seti (inşaat alanında kasket) kaynak veri setinden (doğal resimler) farklıdır ve 1.000 sayısından fazla tutulmuştur.

2.6 Karşılaştırma metrikleri

Modellerin başarımlarını değerlendirmek ve literatürdeki diğer çalışmalarla karşılaştırmak için nesne tespiti bağlamında uygulanan makine öğrenmesi metriklerinden; TP (True Positive) görüntüden tespiti yapılması istenen nesneyi tespit etme, FP (False Positive) görüntüden tespiti yapılması istenmeyen nesneyi veya arka plan görüntüsünü tespit etme, FN (False Negative) görüntüden tespiti yapılması istenen nesneyi tespit edememesidir. Bu 3 değer, test veri setindeki tüm görüntüler için hesaplanarak toplandıktan sonra sistemin başarı değerlerini veren aşağıdaki eşitlikler uygulanmaktadır.

$$\text{Doğruluk} = \frac{TP}{TP + FP + FN} \quad (6)$$

Doğruluk değeri, modelin doğru tahmin ettiği nesnelerin toplam nesne sayısına oranı ile hesaplanmaktadır. Bu değer dengesiz veri setlerinde tek başına yeterli değildir.

$$\text{Hassaslık} = \frac{TP}{TP + FP} \quad (7)$$

Denklem (7)'de verilen Hassaslık, doğru tespitlerin gerçekte kaçının doğru olduğunu göstermektedir. Ayrıca yanlış tespit oranını veren FP 'nin ağırlıklı olduğu değerdir.

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (8)$$

Denklem (8)'deki Duyarlılık ise doğru tespit edilmesi gereken nesnelerin ne kadarının doğru tespit edildiğini göstermektedir. Gözden kaçırılan tespitler olan FN bu eşitlikte ağırlıklıdır.

$$F1 \text{ Skor} = 2 * \frac{\text{Hassaslık} * \text{Duyarlılık}}{\text{Hassaslık} + \text{Duyarlılık}} \quad (9)$$

Son olarak **Denklem (9)**ile hesaplanan F1 Skor, Hassaslık ve Duyarlılığın harmonik ortalamasıdır ve sadece FP veya FN 'nin oluşturduğu hatayı değil tüm modelin hata değerini görme açısından değerlidir. Literatürde doğruluk yerine F1 Skorun temel alınması da bu sebepten kaynaklanmaktadır.

3 Bulgular ve tartışma

Veri sayısının fazla olduğu eğitim ve derin öğrenme modeli epok değerinin yüksek olması aşırı öğrenmeye neden olabilir [72]. Bu yüzden görüntü sayısı 5.000, epok sayısı her 3 model için de 200 olarak belirlenmiştir. Model eğitime başladığında, algoritma kayıp (yitim) değeri grafiğinin kademeli olarak azalması beklenir. Eğer azalma durursa model, öğrenmeyi kesmiş veya öğrenebileceği her şeyi zaten öğrenmiş demektir [73]. Bu durum gerçekleştiğinde eğitim, kullanıcı tarafından durdurulabilir.

Epok sayısının 200 tutulmasının yeterli olduğu erken durdurma kriteri kullanılarak kanıtlanmıştır. Erken durdurma kriterinin kullanım amacı, model eğitimi kullanıcı yerine keserek eğitim boyunca oluşan en iyi ağırlığı bulmaya yöneliktir. YOLO V5 için denenen bu yöntemde 89. epoktan sonraki 111 epokun gereksiz olduğu ortaya çıkmıştır. Oluşan ağırlığın en iyi olduğunu söylemiştir yani bu epoktan sonraki eğitim, iyileşme yapmamıştır.

Deneylerin eşit şartlar altında gerçekleşmesi için YOLO V4, V5 ve Faster R-CNN transfer öğrenmeli/öğrenmesiz;6 model eğitimi, Intel i7-8750H işlemci, 16 GB RAM özelliklerine sahip kişisel bilgisayarda PyTorch [74] çerçeve yazılımı kullanılarak gerçekleştirilmiştir. PyTorch sayesinde kişisel bilgisayarın işlemci thread'leri paralel çalıştırılabilmiştir.

Şekil 3,4 ve **5**'teki grafiklerde sırayla transfer öğrenme tekniğinin uygulanmadığı ve uygulandığı Faster R-CNN, YOLO V4 ve YOLO V5 modellerinin eğitim boyunca hesapladıkları yitim değerinin değişimini görmekteyiz. Her 3 grafikten anlaşılabileceği üzere transfer öğrenmenin uygulanmadığı modellerin yitim değerleri her epok aralığında (10'luk), transfer öğrenmenin uygulandığı modellerin yitim değerlerine göre daha fazladır. Bu duruma bakarak transfer öğrenme tekniğinin, olması gereken (gerçek) ile tahmin edilen arasındaki farkın değeri olan yitimi düşürmede daha başarılı olduğu söylenebilir.