



Transfer öğrenmeli ve transfer öğrenmesiz derin ağlar ile inşaat alanında kask tespiti

Helmet detection on the construction site with transfer learning and without transfer learning deep networks

Mehmet Uğur Türkdamar^{1,*}, Murat Taşyürek², Celal Öztürk³

^{1,*}Niğde Ömer Halisdemir Üniversitesi, Bilgisayar Mühendisliği Bölümü, 51240, Niğde Türkiye

²Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü, 38280, Kayseri, Türkiye

³Erciyes Üniversitesi, Bilgisayar Mühendisliği Bölümü, 38039, Kayseri, Türkiye

Öz

İnsan yaşamını önceleyen sistemlerin yaygınlaşması toplumlara bütüncül fayda sağlamaktadır. Solunum yoluyla bulaşıcı hastalıklardan sakınmak için ağız-burun maskesi takmanın Covid-19 pandemisi ile zorunlu hâle geldiği gibi yapı inşaatında çalışan işçilerin inşaat alanında kafa kaskı takması zorunludur. İnşaat alanlarında çalışan işçilerin kaskını takıp takmadığının kontrolünü göz ile yapmak yorucu ve hataya açıktır. Yapay zekâ tabanlı bilgisayar teknolojilerinin geliştiği bu çağda hayatımızı her anlamda kolaylaştıran sistemlerin varlığı ümit vaat etmektedir. Bu çalışmada görüntü verisinin anlamlandırıldığı evrişimli sinir ağı (ESA) tabanlı derin öğrenme ile kask takma kontrolünün otomatik yapılması önerilmiştir ve YOLO V4, V5 ve Faster R-CNN modellerine uygulanan transfer öğrenme tekniği ile kısıtlı veri seti probleminin üstesinden gelinmiştir. Deneylerde transfer öğrenme uygulanmayan eğitimlere de yer verilerek yöntemin etkinliği incelenmiştir. Sonuçta transfer öğrenmeli YOLO V5 modelinin %98 f1 skor ile 6 farklı model eğitimi arasında en başarılı olduğu gözlemlenmiştir.

Anahtar kelimeler: Kask tespiti, Derin öğrenme, Transfer öğrenme, ESA, Yapay zekâ

1 Giriş

İnşaat alanlarında çalışanlar, ihtiyaca yönelik yapı işlerini gerçekleştirirken kendi sağlıkları için dikkatli olmak durumundadır. İnşası devam eden yapıdan baş bölgesine düşebilecek herhangi bir cisim, ciddi yaralanmalara hatta hayat kaybına sebebiyet verebilir. Bu nedenle inşaat sahası içinde çalışanların kafa koruyucu ekipmanı olarak kask takması ciddi önem teşkil etmektedir. Yakın zamanlı bir istatistiğe göre Çin'deki kazaların yaklaşık %67'si kask takmamaktan meydana gelmiştir [1]. Mevcut prosedürde çalışanların kask takma kontrolünün, ekip başı tarafından yerine getirilmesi vardır, fakat gözetleme yoluyla yapılan bu kontrolde yorulma dolayısıyla gözden kaçırmalar meydana gelebilmektedir. Bu sebeple insan yaşamını önceleyen

Abstract

The widespread use of systems that prioritize human life provides holistic benefits to societies. In order to avoid respiratory contagious diseases, wearing a mouth-nose mask has become mandatory with the Covid-19 pandemic, and workers working in building construction are required to wear a head helmet at the construction site. It is tiring and error-prone to visually check whether the workers working on the construction sites are wearing their helmets. In this age, where artificial intelligence-based computer technologies are developed, the existence of systems that make our lives easier in every sense is promising. In this study, it is proposed to make helmet wearing control automatic with convolutional neural network (CNN) based deep learning in which the image data is meaningful. The limited data set problem was overcome with the transfer learning technique applied to the YOLO V4, V5 and Faster R-CNN models. The effectiveness of the method was examined by including the trainings in which transfer learning was not applied in the experiments. As a result, it was observed that the YOLO V5 model with transfer learning was the most successful among 6 different model trainings with an f1 score of 98%.

Keywords: Helmet detection, Deep learning, Transfer learning, CNN, Artificial intelligence

sistemlerin tasarımında aktif rol oynayan derin öğrenme ağları, insanı belli bir noktaya kadar devre dışı bırakarak probleme çözüm olarak sunulabilir.

İnternetin gelişmesiyle veri setlerinin, bilimsel çalışmaların artışıyla algoritmaların, donanımın güçlenmesiyle de verinin paralel işlenebilirliğinin yaygınlaşması sonucu derin öğrenme çalışmaları hız kazanmıştır. Derin öğrenme son yıllarda görüntü tanıma, doğal dil işleme, ses tanıma, mobilde virtüs tespiti gibi çok çeşitli alanlarda yaygın olarak kullanılmaktadır [2-5]. Görüntü kaynaklarındaki bilginin anlamlandırılması işlemi, derin öğrenme altyapısını kullanan bilgisayarlı görü teknolojisi sayesinde olmaktadır. Nesne tespiti ise görüntüde, tespiti yapılması istenen nesnelere hangilerinin olduğunu ve bu nesnelerin görüntüde nerede olduklarını

* Sorumlu yazar/ Corresponding author, e-posta / e-mail: ugurturkdamar@gmail.com (M. Turkdamar)

Geliş / Received: 12.09.2022 Kabul / Accepted: 14.11.2022 Yayınlanma / Published: 15.01.2023

doi: 10.28948/ngumuh.1173944

keşfeden bir bilgisayar görme uygulamasıdır. Tespit işlemi, nesnenin görüntüdeki konumunu da bulmaya çalıştığından sınıflandırma problemlerine göre daha komplekstir. Işığın azlığı, arka planla tespiti yapılacak nesnenin iç içe geçmesi, nesnelerin birbirini üzerine gelerek çakışması, görüntüdeki uzak veya yakın nesnelere gibi faktörlerden dolayı; tespit işlemi oldukça zor bir süreçtir. Buna karşın ESA tabanlı derin öğrenme modelleri kullanılarak gerçekleştirilen nesne tespiti, sayılan çoğu faktörü otomatize ettiği için geleneksel metotlara göre daha başarılıdır.

Bir yapay zekâ tekniği olan derin öğrenme, veriden çeşitli özellikleri çıkarma işini iyi organize olmuş çeşitli doğrusal olmayan katmanlar aracılığı ile beyin veri işleme taklit ederek gerçekleştirilmektedir. Veriden öğrenmesi yönüyle insan öğreniminden esinlenen makine öğrenmesinin; veriyi kabul ettiği girdi, özellik çıkardığı gizli ve isteneni verdiği çıktı olmak üzere 3 katmanı varken, başarılı olmak için daha fazla bilgiye ihtiyaç duyan derin öğrenmenin makine öğrenmesinden ana farkı; görüntü, ses, video gibi karmaşık veriden daha fazla anlam çıkarabilmek için birden çok gizli katmana sahip olmasıdır. Veri, sıralı yapıda olan katmanlardan geçtikçe bilginin yayılımı gerçekleşmekte ve problemin çözümüne bir adım daha yaklaşılmaktadır.

Derin öğrenme tekniklerini içeren makine öğrenmesi, önceden toplanmış etiketli (gözetimli) veya etiketsiz (gözetimsiz) veri setinde eğitilerek önceden görmediği veriyi keşfetmede kullanılır. Makine öğrenmesindeki el ile özellik çıkarma (verinin verilen özelliklerinden hangilerinin sınıflandırma problemine uygun olduğunun belirlenmesi) zaman alıcı ve zorlayıcıdır. Ayrıca el ile özellik çıkarmından yüksek seviyeli özelliklerin eldesi güçtür [6]. Buna karşın derin öğrenme ağları özellik çıkarmayı, katmanlarının içeriğindeki algoritmalar sayesinde otomatikleştirmektedir. Derin öğrenme, çok girdi ile eğitildiğinden görüntü alanında el ile özellik çıkarmaya göre avantajı; ışığa, bozulmaya, çevirmeye ve döndürmeye karşı daha dirençli olmasıdır. Derin öğrenme ağlarından birisi olan evrişimli sinir ağı (ESA); son yıllarda bilgisayar ağları [7], tarım arazileri görüntüleme [8], tıpta hastalık teşhisi [9] gibi farklı alanlarda sıklıkla tercih edilmektedir. ESA'nın başarılı olması için gerek şart çok sayıda veridir. ESA tabanlı derin öğrenme, doğal resimler (her yerde karşımıza çıkan) içeren ImageNet [10], COCO [11] ve PASCAL VOC [12] gibi büyük veri setleri üzerinde dikkate değer başarılar yakalasa da spesifik alanlarda (inşaat kaskı gibi) verinin azlığı sebebiyle yüksek başarı elde edememektedir. Bu handikapı aşmak için insan öğrenim aktarmasından ilham alan transfer öğrenme tekniği işe koşularak çok sayılı veri alanında öğrenilen bilgi, az sayılı veri alanına aktarılabilir. Büyük veri setlerindeki nesnelerin kenar, leke, doku gibi genel özellikleri; küçük veri setindeki nesnelerin çoğu özelliğiyle benzeşmektedir. Bu sayede probleme özgü kurulan ağı; ilk katmanlarında ek bir öğrenmeye ihtiyaç duyulmadan transfer edilen bilgi kullanılabilirken, ileriki katmanlarda alana ait daha detay nesne özellikleri öğrenilir. Örneğin bir işi yapmayı öğrenmiş birisi, daha önce yapmadığı benzer başka bir işi yapmayı öğrenmeye çalıştığında fazla zorlanmayacaktır çünkü önceki işte öğrendiği bilgisini, yeni işe rahatlıkla

uygulayabilecektir. Aynı, insanın yeni bir işi; en baştan öğrenmesine göre bilgi transferiyle öğrenmesi sayesinde öğrenim süresinin kısalması ve daha doğru sonuçları alması gibi, derin ağları eğitmede kullanılan bu teknik ile de eğitim süresi ciddi oranda düşmekte ve çoğu zaman, en baştan eğitime göre daha yüksek skorlar elde edilebilmektedir.

Kaskın takılıp takılmadığını kontrol etmek için RFID; reader ve etiketi kullanan yöntem Kelm vd. [13] ve Torres vd. [14] tarafından önerilmiştir. HOG transform yöntemi ile iki aşamalı tespit işlemi Rubaiyat vd. [15] gerçekleştirmiştir. Malisiewicz vd. [16] SVM kullanarak aday kutularının, tespiti istenen nesnelere denk gelmediğinin kontrolünü sağlamıştır. Liu vd. [17] ise bir görüntü sahnesindeki tüm yüz kaskını bulmak için Canny'sınır tespitine tam uyan çemberler kullanmıştır. Bu metotlar geleneksel makine öğrenmesi kapsamına girmektedir ve genelde özelliklerin; güçlü yetilerle, kompleks tasarım işlemiyle ve düşük genelleme kabiliyetiyle çıkarılması ile sonuçlanmaktadır.

Yakın yıllarda ortaya çıkan nesne tespiti algoritmaları ESA tabanlı olup hız ve doğruluk açısından büyük bir potansiyele sahiptir [18]. Kask tespitinde ESA tabanlı derin öğrenme modellerini kullanan çalışmalardan; Fangbo vd. [19] YOLO V5'in tüm modellerini (x, l, m, s) transfer öğrenmeli ve öğrenmesiz eğittikten sonra karşılaştırarak en yüksek mAP değerini transfer öğrenmeli YOLO V5x'ten %94,7 ile almıştır. Fan vd. [20] kask tespiti için kullandığı YOLO V3'ün omurga ağını DenseNet ile değiştirerek orijinal YOLO V3'e göre %2,44 doğruluk artışı elde etmiştir. Yange vd. [21] 3.261 kask görüntülü veri setinde SSD-MobileNet algoritması işe koşarak %95 hassaslık yakalamıştır. Choudhury vd. [22] trafikte iki tekerlekli araç kullananların kask takma önemine vurgu yapmak için trafik video görüntülerinden kask tespitini SSD ve MobileNet kullanarak gerçekleştirmiştir. Madhuchanda vd. [23] YOLO V3 modeli ile trafikteki motorluların kask tespitini yaptığı çalışmada %96,23 doğruluk yakalamıştır. Wei vd. [24] trafik video kayıtlarından YOLO V5 ile kask tespitinde F1 skoru %92,7 bulmuştur. Shilei vd. [25] görüntüdeki inşaat kaskı tespiti için YOLO V5 modelinde NMS yerine DloU-NMS kullanarak daha küçük nesnelerin tespitini, orijinal YOLO V5'e göre daha iyi yapmıştır. Rui vd. [26] inşaat alanındaki kask tespitinde, görüntülere bulanık Gaussian ön işlemi uygulayarak YOLO V3 modelinden daha başarılı sonuçlar elde etmiştir. Lin vd. [27] ResNet50 omurga ağı RetinaNet kullanarak trafikteki motorlarda kask takmayan fazla yolcuları tespit etmiştir.

ESA tabanlı derin öğrenme modellerini farklı veri setleri üzerinde uygulayan çalışmalara baktığımızda ise Zicong vd. [28] YOLO V4'te CSPBlock yerine ResNet ağı modülleri kullanarak ağı hesaplama maliyetini düşürmüştür. Ayrıca yazarlar YOLO V4'ün gerçek zamanlı bir versiyonunu geliştirerek orijinal YOLO V4'e yakın bir skora ulaşmıştır. Chethan vd. [29] trafikteki araba, kamyon, insan ve iki tekerli tespiti için görüntüler üzerine YOLO V3, videolar üzerine ise YOLO V4 uyguladığı çalışmada sırasıyla %98 ve %99 doğruluğa ulaşmıştır. Yanfen vd. [30] YOLO V4 kullanarak trafik ortamında bulunan 10 farklı tipteki nesnenin tespitini gerçekleştirmiştir. Liu vd. [31] deniz

yüzeyindeki nesnelerin tespiti için insansız yüzey araçlarının topladıkları görüntü veri setini kullanan çalışmada YOLO V4 işe koşulmuştur. Roy vd. [32] verimli tarım yapmak için bitki hastalıklarını zamanında teşhis etmenin öneminden yola çıkarak YOLO V4'ün omurga ağında DenseNet, aktivasyon fonksiyonunda Hard-Swish kullanarak domates bitkilerindeki 4 farklı hastalığı %93.64 F1 skoru ile tespit etmişlerdir. Yu ve Zhang [33] koronadan korunmaya yönelik maske takmanın önemini vurgulamak için insanların maske takma kontrolünü, YOLO V4 kullanarak %98.3 mAP skoru ile sağlamışlardır. Li vd. [34] KITTI yaya veri setini kullanarak kızılötesi görüntülerden nesne tespiti için YOLO V5'in CSP modülünü genişleten çalışmada YOLO V4'e göre %21 mAP artışı gözlemlenmiştir. Wu vd. [35] YOLO V4 ile doğadaki elma çiçeklerini tespit etmiştir. Yu ve Choi [36] otonom araç sürüşündeki risk ataması için nesne başına tek derinlik tespiti için KITTI yaya veri seti üzerinde YOLO V4 çalışmıştır.

Taranan çalışmalarda kullanılan CNN tabanlı derin öğrenme modelleri farklı veri setlerinde de benzer başarıyı yakalaması açısından genelleme yetenekleri yüksektir yorumu yapılabilir. Kasketin; takılı veya değil sınıflandırmasının farklı ve güncel CNN tabanlı derin öğrenme modelleri aracılığıyla karşılaştırmalı çalıştırılması ihtiyacının yanında transfer öğrenme tekniği araştırma alanının hâlâ gelişmekte olduğu görülmüştür. İncelenen çalışmalarda kullanılan; donanımın uzak mesafeden çalışmaması, makine öğrenmesinin el ile özellik çıkarımının başarısının düşük olması gibi sebeplerden ötürü çalışmada derin öğrenme yöntemleri tercih edilmiştir.

Bu çalışmada inşaat sahasındaki işçilerin kask tespitini otomatik gerçekleştiren CNN tabanlı derin öğrenme modelleri kullanılmıştır. Ağların eğitiminde kullanılan görüntüler, niş alana ait olduğundan 5.000 adetle sınırlıdır. Dolayısıyla çalışmada, küçük veri seti probleminin üstesinden gelen transfer öğrenme tekniği uygulanmıştır. Ayrıca transfer öğrenme tekniğinin; farklı ESA'lardaki ve en baştan eğitim ile arasındaki etkililiği (eğitim süresi ve doğruluk bakımından) karşılaştırılmıştır.

Çalışmanın; 2. Kısımında ESA'nın genel özellikleri, kullanılan veri seti ve uygulanan ön işlem, derin öğrenme modelleri ve ayarları, transfer öğrenme uygulaması, nesne tespiti bağlamında karşılaştırma metrikleri sunulmuştur. 3. kısımda deneysel çalışmalar ve 4. kısımda genel bir değerlendirme yapılarak gelecekte planlanan çalışmalara değinilmiştir.

2 Materyal ve metot

Derin öğrenme konseptini önerenler [37]; büyük miktardaki yazı, ses, görüntü veya video verisinde yer alan nesnelerin (eşsiz; kelime, kayıt veya şekil) güçlü özelliklerini çıkarmak ve dolayısıyla daha başarılı sınıflandırma işlemi gerçekleştirmek için derin sinir ağları kullanmışlardır.

Probleme göre özelleşen derin öğrenme ağlarından ESA, özellikle görüntü ve ses sınıflandırmadaki başarısıyla bilinmektedir. Derin öğrenme ağları, sıralı katmanlardan oluşmaktadır. Verinin her katman geçişinde daha derin özellikleri çıkarılmaktadır. İlk çok katmanlı ESA olan LeNet-5 ağı, 7 katmandan oluşmaktadır [38]. Rakamların

sınıflandırılmasında kullanılan bu ağı, birkaç banka el yazısı rakamlarının otomatik tanınmasında uygulamıştır. ESA'ların; GPU gelişimine bağlı olarak derinleşmesi ile görüntü sınıflandırma problemlerindeki başarıları da artmaktadır.

ESA; temelde evrişim, havuzlama ve tam bağlı katmanlardan oluşmaktadır. Evrişim ve havuzlama katmanları nesne özelliklerini çıkarabildiği için ağın en önemli kısımlarıdır ve sıkça art arda kullanılmaktadır. Evrişim katmanı sahip olduğu klasik görüntü işleme algoritma filtreleri sayesinde görüntüden çeşitli nesne özelliklerini çıkarabilmektedir. Ağın ilk evrişim katmanlarında, görüntüden tespiti yapılması istenen nesnelerin kenarı, kıvrımı, rengi gibi genel özellikleri çıkarılırken daha ileriki katmanlarda bu nesnelerin daha ince, detaylı ve karmaşık özellikleri çıkarılmaktadır. Evrişim katmanının filtre sayısı ve filtre boyutu olmak üzere iki ana parametresi vardır. Bunlardan filtre sayısı, yukarıda değinilen klasik görüntü işleme algoritma filtre çeşitlerinden, evrişim katmanının kaçını kullanacağını belirtir. Filtre sayısı çok tutulursa ağın hesaplama yükü de bir o kadar artar fakat görüntüden detay nesne özellikleri çıkarılacaksa sayıyı artırmak, başarıyı da artıracaktır. Diğer parametre olan filtre boyutu ise kullanılan iki boyutlu filtrenin en ve boy oranıdır. Filtre; görüntü üzerinde kaydırıldığından görüntünün her noktasına bakarak matris değerlerinin görüntüdeki nesnelerin matris değerleriyle eşleşip eşleşmediğine bakılır. Bu yüzden küçük boyutlu filtreler, görüntüde; az piksel harcayan küçük nesneleri daha yüksek başarıyla tespit ederken büyük boyutlu filtre seçimi, görüntüde tespiti yapılacak fazla piksel harcayan büyük nesnelere varsa yapılır. Evrişim katmanından havuzlama katmanına geçen özelliklerin (matris) sadece güçlü (yüksek) değerleri ağda tutularak özelliğin içeriği, önceden belirlenen adım sayısına göre azaltılır, dolayısıyla da ağın; hesaplama maliyeti düşürülür ve aşırı öğrenmesi kontrol altında tutulur. Ardından, iki boyutta olan bu özelliklerin tam bağlı katmanlardan geçebilmesi için tek boyuta çevrimi yani matris tipinden vektör tipine dönüşümü gerçekleştirilerek tespiti yapılacak nesnelerin sınıflandırma çıktı skorları elde edilir.

Tipik bir ESA ileri ve geri yönlü geçiş aşamalarına sahiptir. Yukarıda bahsedilen ileri yönlü geçiş, çıktı ve hesaplama kaybı (gerçek ve tahmin edilen çıktı arasındaki fark) tahminini içerirken sadece eğitim esnasında uygulanan geri yönlü geçiş, hesaplanan hata bazında öğrenilebilir filtreleri ve ağırlıkları (girdi özellikleriyle birleşen) geri yayılım algoritması ile düzeltmeyi içerir. Ayrıca ESA'lar paylaşılan ağırlık, alıcı alanı ve yerel bağlantı konseptlerine bağlıdır. Ağda eğitilmesi gereken ağırlıkların sayısını düşüren ağırlık paylaşma tekniği, LeCunn tarafından 1998'de önerilmiştir [39]. Bu teknik, ağdaki düğümlerin (nöron, birim) ürettiği ağırlıkların yakın başka düğümlerde de kullanılmasıdır. Yani görüntünün herhangi bir bölgesinde tespiti yapılan motifin (nesnenin), başka bir bölgede de yer alabileceği göz önünde bulundurularak öğrenilen bilgi (ağırlık), ağ arasında paylaşılmaktadır. Böylece ağın hem eğitim süresi hem de hesaplama maliyeti düşmektedir. Üst paragrafta verilen ESA mimarisinin temel katmanlarının

farklı sırada ve sayıda kullanılmasıyla veya atlama gibi yeni katmanların da ağa dâhil edilmesiyle işin özü birçok ESA modeli çıkmıştır. Beyindeki görüş açısına benzetilen alıcı alanı tekniğini; küçük (dar) tutan ESA modelleri, görüntüdeki; küçük nesnelere tespit daha başarıyla büyük (geniş) tutan modeller büyük nesnelere tespit daha başarılıdır. Alıcı alanındaki bu mantık, evrişim katmanında kullanılan filtrenin boyut seçimiyle benzerdir. ESA'nın kullandığı son teknik ise, herhangi bir katmandaki tüm nöronların (birimlerin-düğümünlerin) bu katmandan bir önceki katmandaki tüm nöronlara tam bağlanmayarak hesaplama maliyetini düşürmesidir. ESA'nın, klasik sinir ağlarından en büyük farkı da tam olarak budur. Bir klasik sinir ağından geçen 3 kanallı (RGB) ve 256 piksel en, boy uzunluğuna sahip bir görüntü için bu ağdaki tüm katmanlarda bias (hata değeri) olmadan 196.608 (256x256x3) adet ağırlık miktarı (hesaplama yükü) hesaplanacaktır.

ESA'nın gelişimini; takip etmek ve artırmak için Stanford Vision Lab tarafından her yıl ImageNet Large Scale Visual Recognition Challenge (ILSVRC) organize edilmektedir [40]. Bu çalışmada 1.000 kategorili, 1.2 milyon görüntülü ImageNet veri setinde eğitilmiş ağırlıklı 3 farklı ESA tabanlı derin öğrenme modeline transfer edilerek modellerin, rasgele ağırlıklar ile eğitime başlaması yerine gündelik hayatta karşılaştığımız birçok nesnenin şekil, kenar, leke, doku gibi özelliklerini öğrenmiş ağırlıklı ile eğitime başlaması sağlanmıştır. Probleme özgü tespiti istenen nesne sayısı iki olduğundan (kafa-kask), son tam bağlı katmanın 1.000 (sınıf sayısı) çıktı vermesi yerine iki çıktı vermesi sağlanarak, ağ eğitilmiştir. Ayrıca, aynı 3 ESA tabanlı derin öğrenme modelinin rasgele ağırlıklar ile eğitime başlaması da çalışmada uygulanarak transfer öğrenmenin etkinliği incelenmiştir.

2.1 Veri seti ve ön işleme

Eldeki veriyi eğitim, validasyon ve test için önceki makine öğrenmesi çalışmalarında ve deeplearning.ai'den Andrew Ng'nin derslerinde [41] 6: 2: 2 oranında bölündüğü ve tavsiye edildiği görülmektedir. Buna karşın çalışmada kullanılan toplam görüntü verisi, 70: 15: 15 oranında bölünerek ağırlık eğitime biraz daha fazla ağırlık verilmiştir. Kaggle'dan edinilen 5.000 görüntünün 3.500'ü eğitime, 750'si validasyona geri kalan 750'si ise teste ayrılmıştır [42]. Eğitim veri seti; modeli eğitmede dolayısıyla ağırlıkların oluşmasında, validasyon veri seti; modelin parametrelerini ayarlama ve modelin kapasitesini değerlendirmede, test veri seti ise modelin genelleme yeteneğini ve tespit başarısını hesaplamada kullanılmıştır [43].

Çalışma, gözetimli makine öğrenmesi (olması gereken değerlerin makineye tanıtılması) başlığına girdiği için eğitilecek görüntüler etiketlenmiştir. Bu işlem için açık kaynak kodlu makesense.ai tercih edilmiştir [44]. Kullanımı herkese açık olan bu program GPLv3 lisanslıdır. Etiketleme ile görüntülerdeki nesnelere konumları ve sınıfları (kategorileri) bilgisayarın anlayabileceği formatta sunulmuştur. Çalışmada kullanılan YOLO modelleri için oluşturulan etiketlerin (COCO veri seti formatı) içeriği sırasıyla; nesne sınıfının eşsiz rakam karşılığı, nesnenin etrafına, görüntüdeki yerini belirtmek için çizilen sınırlayıcı

kutunun; merkez x, y noktası ile genişlik, yüksekliği iken Faster R-CNN (PASCAL VOC veri seti formatı) modeli için sırasıyla nesne sınıfı, sınırlayıcı kutunun; sol alt köşesinin yataydaki konumu olan x min, sağ üst köşesinin yataydaki konumu olan x max, sol alt köşesinin dikeydeki konumu olan y min, sağ üst köşesinin dikeydeki konumu olan y max'tır. Oluşturulan etiketlerin dosya formatları, bilgisayarda fazla yer kaplamayan txt ve xml'dir.

2.2 YOLO

Nesne tespiti mimarilerinin ortak noktası, girdi görüntüsünü sırayla omurga, boyun ve kafa bölümlerinden geçirmesidir. Boyun; omurganın çıkarmış olduğu görüntü özelliklerini, karıştırıp birleştirdikten sonra tespit için kafaya hazırlayan özellik toplayıcı olarak çalışmaktadır. Kafa bölümü ise her sınırlayıcı kutunun tespitinden ve sınıflandırılmasından sorumludur. Nesne tespitindeki tek, çift aşamalı mimari farkı, bu noktada ortaya çıkmaktadır; iki aşamalı mimari, sınırlayıcı kutu; tespiti ve sınıflandırması görevlerini ayrı ayrı çalıştırıp (Seyrek Tespit) sonuçlarını daha sonra birleştirirken, tek aşamalı mimari, bu işlemleri tek seferde (Yoğun Tespit) yapmaktadır. Bir nesne tespiti mimarisi olan YOLO, görüntü; sınıflandırma ve konumu belirlemek için ESA ağında yalnızca bir ileri geçiş gerektirdiği için tek aşamalıdır ve bu yüzden yalnızca bir kere bakarsın mânâsına gelmektedir.

YOLO eğitiminde kullanılan girdi görüntüleri öncelikle Şekil 1'deki gibi karesel ızgaralara bölünmektedir.



Şekil 1. YOLO ızgara

Görüntü örneğinin daha iyi anlaşılması için 9 ızgaraya bölünmüştür, gerçekte ağırlık girişte kabul ettiği çözünürlük değerince, daha fazla sayıda ızgaraya bölünmektedir. Daha sonra etiketleme yolu ile YOLO'ya tanıttığımız tespiti yapılması istenen nesnelere merkez noktaları içinde kaldığı ızgara tarafından tespit edilmeye çalışılmaktadır. ızgara tespit işlemini, 4 parametrelilik sınırlayıcı kutuyu ve bu kutunun güven skorunu tahmin ederek gerçekleştirmektedir. Sınırlayıcı kutunun parametrelerinden x, y; kutunun merkezi w, h; kutunun genişlik ve yüksekliğidir. Kutunun güven skoru Denklem (1)'de verilen ızgaranın içinde nesne olma olasılığı $p(Nesne)$ ve olması gereken ile tahmin edilen kutu

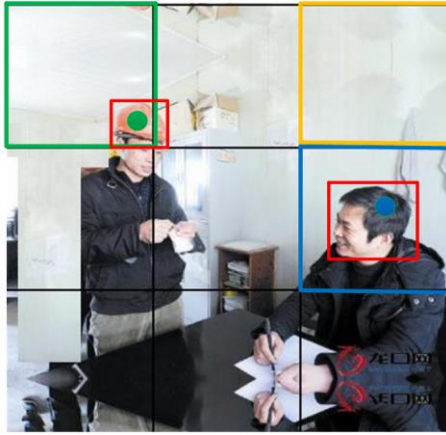
arasındaki fark IOU_{pred}^{truth} 'un çarpımıyla hesaplanmaktadır. $p(Nesne)$ 0 ile 1 arasında değer almaktadır; ızgara içinde nesne var ise $p(Nesne)$ 1 değerini alarak güven skoru hesabını IOU_{pred}^{truth} 'e bırakmaktadır. ızgara içinde nesne yok ise güven skoru 0 olmaktadır.

$$güvenskoru = p(Nesne) * IOU_{pred}^{truth} \quad (1)$$

Güven skoru ile sınırlayıcı kutunun parametre sayısı esasen 5 olmaktadır. Her ızgaranın oluşturduğu tahmin edilen görüntü konumunu belirten kutu vektörü \hat{y} ile gösterilirken gerçekteki (olması gereken) görüntü konumunu belirten kutu vektörü, **Denklem (2)**'de verilen y 'dir.

$$y = [p_c b_x b_y b_h b_w c_0 c_1] = p(Nesne) * IOU_{pred}^{truth} \quad (2)$$

Denklem (2)'deki p_c güven skorunu, $b_x b_y b_h b_w$ sınırlayıcı kutunun yataydaki, dikeydeki merkez konumunu ve genişlik, yüksekliğini, $c_0 c_1$ sınıf tahminlerini ifade etmektedir. **Şekil 2**'de girdi görüntüsü üzerinde tahminlerin gerçekleştiği bir örnek sunulmuştur. Şekildeki sarı ızgara, tespiti yapılması istenen herhangi bir nesneyi içermediğinden oluşturmak istediği sınırlayıcı kutunun güven skoru 0 çıkmaktadır, dolayısıyla bu ızgara sınırlayıcı kutu üretememektedir. Yeşil ızgarada tespiti yapılması istenen kask nesnesi ve mavi ızgarada tespiti yapılması istenen kafa nesnesi olduğundan bu ızgaralar için üretilen çıktı vektörleri sırasıyla $\hat{y} = [1 b_x b_y b_h b_w 1 0]$, $\hat{y} = [1 b_y b_h b_w 0 1]$ olmaktadır.



Şekil 2. YOLO çalışma mantığı

YOLO algoritması son olarak NMS (Non-Maximum Suppression) tekniği ile nesne içermeyen veya diğer sınırlayıcı kutularla aynı nesneyi içeren sınırlayıcı kutuları, IoU (Intersection over Union) eşik değeri yardımıyla elemektedir. Bu sayede hesaplama maliyeti düşerken gözünü yormayan bir sonuç elde edilmektedir.

2.2.1 YOLO V4 ve V5

YOLO algoritmaları önceki bölümde de değinildiği üzere, tek aşamalı CNN tabanlı derin öğrenme modellerindedir. Hızlı tespit yetenekleri, tek aşamalı olmalarından kaynaklanmaktadır. YOLO V4 algoritmasının

omurga ağı CSPDarknet53, boynu SPP ve PAN, kafa bölümü YOLO V3'ten oluşmaktadır. Algoritmanın yapısındaki elemanlar için kullandığı metotlar çok çeşitlidir. Omurgası için kullandığı Ücretsizler Torbasının (Bag of Freebies) temel amacı çıkarım maliyetini artırmadan, nesne mimarisinin daha doğru sonuçları alması adına daha iyi eğitim metotları geliştirmektir. Diğer bir ifadeyle sadece eğitim stratejisini değiştiren veya sadece eğitim maliyetini artıran metotlara denmektedir. Nesne tespiti tarafından kabul edilen bu tanımla birleştiği kısım veri artırmadır. Veri artırmanın amacı girdi görüntülerini çeşitlendirerek tasarlanan nesne mimari modelinin, farklı çevrelerden gelen görüntülere karşı daha güçlü olmasını sağlamaktır. İki sık kullanılan veri artırma metodu görüntüsel ve geometrik bozmalar, nesne tespiti görevine bu yönden fayda sağlamaktadır. Görüntüsel bozmalarla uğraşırken görüntünün parlaklığı, kontrastı, rengi, doygunluğu ve gürlüğü değiştirilirken geometrik bozmalarda rasgele ölçekleme, kırpma, çevirme ve döndürme yapılmaktadır.

Mimarisi için çıkarım maliyetini küçük miktarda artırırken tespit görevinin doğruluğunu önemli ölçüde artıran, eklemeli modüllere ve rötuş metotlarına ise Özeller Torbası (Bag of Specials) denmektedir. Eklemeli modüller; alıcı alanını genişletmeye, motivasyon mekanizmasını tanıtmaya veya özellik entegrasyonu kabiliyetini güçlendirmeye yararken rötuş metotları, model tespit sonuçlarını görüntüleme içindir. Alıcı alanı artırma işlemi önceden bahsedildiği gibi, görüntüdeki büyük nesnelerin tespitini kolaylarken motivasyon mekanizması, girdi ile katmanlardan geçen özellikler arasında bağlantılar oluşturmaktadır.

YOLO V5'te omurga ağı olarak, derin ağlarla işlem süresinde önemli bir gelişme sağlayan CSPNet (Cross Stage Partial Network) ve 3 adet YOLO V4 katmanı yerine geçen Odak modülü kullanılmaktadır (Jocher, 2020). Odak modülü, eğitim için gerekli CUDA (C ile yazılmış algoritmaların GPU üzerinde çalışmasını sağlayan geliştirme araçları kümesidir) hafızasını azaltırken ileri ve geri yönlü yayılmayı artırmaktadır [45]. Mimarideki boyun ise genelde modellerin nesne ölçeklemesindeki genelleşirmesine yardım eden, özellik piramitlerinin üretimini sağlamaktadır. PANet kullanılarak geliştirilen özellik piramitleri bu bölümde mevcuttur. Boyunda ayrıca ağı sabit boyut kısıtlamasını ortadan kaldıran havuzlama katmanı, SPP (Spatial Pyramid Pooling) bulunmaktadır. SPP katmanı, görüntü özelliklerini havuzlayarak tam bağlı katmanlara veya diğer sınıflandırıcılara girdi olarak verilecek çıktıları üretmektedir. Modelin kafa bölümü, YOLO V3 ve V4 ile aynıdır. Tespit aşamasında kullanılan kafa, özelliklere ançor kutuları uygulayarak nihai çıktı vektörlerini (sınıf olasılıkları, nesnesizlik skorları, sınırlayıcı kutuları; içeren) üretmektedir.

YOLO V5'in YOLO V4'ten bir farkı, IoU-yitim yerine genelleştirilebilir yitim (GIoU-yitim) algoritmasını işe koşması, diğer bir farkı ise oto öğrenme sınırlayıcı kutuları kullanarak genel doğruluğu artırmasıdır [46]. Her iki algoritmanın; omurgasında da CSPDarknet53 ağı kullanıldığından görüntüden derin özellikleri çıkarma yetenekleri benzerdir, kullandıkları yöntemler (BoF, BoS,

mozaik veri çoklama) de aynı olduğundan doğrulukları benzer oranda artış göstermektedir.

YOLO'nun önceki versiyonları C programlama diliyle yazılmışken YOLO V5, Python programlama diliyle yazılmıştır. Böylelikle YOLO V5'i IoT cihazlara yüklemek ve farklı cihazlarla entegre etmek kolaylaşmıştır. YOLO V5'in boyutu sadece 27 MB iken DarkNet kullanan YOLO V4'ün boyutu 244 MB'tır [47]. Öğrenme verimini artırmak için derinleşen ağların parametre sayıları da artmaktadır. YOLO V5'in farklı versiyonlarında, n'den x'e doğru parametre sayısı ve doğruluğu artarken tespit hızı düşmektedir. Mevcut çalışma gerçek-zamanlı olmadığı için en yüksek hızlı versiyonlar (n ve s gibi) yerine en yüksek doğruluk veren 444 katmanlı, 86.186.872 parametrelili versiyon (x versiyonu) kullanılmıştır.

2.3 Faster R-CNN

Bu algoritmanın oluşmasına katkı sağlayan önceki versiyonları R-CNN ve Fast R-CNN'dir. R-CNN görüntü üzerinde bir demet kutu (bölge) önerip kutuların nesneye denk gelip gelmediğine bakmaktadır. Bu kutuların üretiminde ise Seçici Arama algoritması kullanılmaktadır. Bu algoritma nesnelere tanımlamak için görüntüdeki komşu pikselleri; doku, renk veya yoğunluğa göre gruplandırmaktadır. Daha sonra bölgeleri, standart kare boyutuna ayarlayarak AlexNet'in değiştirilmiş bir versiyonuna girdi olarak vermektedir [48]. Ağın bitişine SVM ekleyerek nesne mi değil mi, nesne ise hangi sınıfa ait sınıflandırması yapmaktadır. Son olarak bölge önerilerine lineer regresyon işlemi uygulayarak daha dar sınırlayıcı kutu koordinatları elde etmektedir.

R-CNN aşağıdaki dezavantajlara sahiptir:

- Görüntünün tüm bölge önerileri için ESA'da birer ileri yönlü geçiş gerektirmesi (Her görüntü için yaklaşık 2.000 ileri geçiş).

- 3 farklı modeli birbirinden bağımsız eğitmesi (CNN ile görüntü özellikleri üretme, sınıflandırıcı ile sınıf tahmini, regresyon modeli ile sınırlayıcı kutuyu daraltma).

Gelişimin ikinci basamağındaki Fast R-CNN, RoIPool ile her bir görüntü için sadece bir CNN çalıştırıp bunu, 2.000 öneri ile paylaşmayı öngörerek hesaplama maliyetini düşürmüştür. Ağın uyguladığı ikinci iyileştirme ise ESA, sınıflandırıcı ve sınırlayıcı kutu düzenleyicisini tek modelde birleştirmesidir. Ek olarak SVM sınıflandırıcısını, çoklu sınıf problemlerinde sıkça uygulanan Softmax katmanı ile değiştirmiştir.

Fast R-CNN'deki tüm bu gelişmelere rağmen bölge önerici problemi çözülememiştir. R-CNN ve Fast R-CNN'de bir demet sınırlayıcı kutu ile olası nesne konumlarını tespit etmeye çalışma işini, GPU'dan faydalanamayan yavaş ve maliyetli Seçici Arama algoritması yapmaktadır. Bölge önerilerinin bağlı olduğu görüntü özellikleri, sınıflandırmanın ilk adımındaki ileri yönlü CNN geçişi ile üretilmektedir. Faster R-CNN'in ana fikri ise ayrı bir seçici arama algoritması çalıştırmak yerine sonuçları tekrar kullanarak bölge önermek olmuştur. Bu yüzden özelliklerin tepesine RPN, Türkçe adıyla bölge öneren ağ adlı tam evrişim ağı eklemiştir. Bu ağ, özellik haritası üzerinde kayan pencere gezdirilerek potansiyel sınırlayıcı kutuyu ve

bu kutuların ne kadar iyi olması gerektiği skorunu vermektedir. İki katmandan oluşan RPN'nin *cls* katmanında nesne var/yok sınıflandırılması yapılmakta ve $2 \times k$ sayıda skor üretilmekte, *reg* katmanında ise sınırlayıcı kutu koordinatları tahmini için $4 \times k$ (sınırlayıcı kutu; merkez x , y , genişlik, yükseklik) sayıda vektör üretilmektedir. Bu aşamadan sonra Faster R-CNN'in kullandığı kritik eleman ROI Havuzlaması'dır. CNN'deki havuzlama mantığına sahip olan bu teknik ile RPN'den gelen bölge önerilerinin sadece en güçlü özellikleri seçilerek sınıflandırma için tam bağlı katmanlara gönderilir. Bu sayede gereksiz özellikler ağdan atılarak hesaplama maliyeti ciddi oranda düşürülür.

2.4 Model ayarları

Çalışmada orta boyutlu görüntüler kullanıldığından YOLO V4 ağının kabul ettiği görüntüler için çözünürlük değeri 416x416 seçilmiştir. YOLO V5 algoritması, omurga ağının genişlik ve derinliğinin ayarlanmasına yarayan *depth_multiple* ve *width_multiple* parametrelerine sahiptir. Derin sinir ağları daha fazla katmana (alıcı alanı artırmak için) ve kanala (daha çok ince-taneli desenleri yakalamak için), büyük boyutlu görüntüler için ihtiyaç duymaktadır. Görüntülerden tespiti istenen kask ve kafa nesnelere ne çok küçük ne de çok büyüktür, bu yüzden YOLO V5 modelinin bu iki parametresinin varsayılan değerleri değiştirilmemiştir. Aynı şekilde, Faster R-CNN'in ağ girişinde kabul ettiği görüntü çözünürlük değeri sabit tutulmuştur.

Ağın iç değişkenlerini ayarlama işlemi, eğitim sürecinin özüdür. Çalışmada optimum ağ mimarisi hiper parametrelerini belirlemek için kapsamlı izgara araması (tüm kombinasyonları deneyen kaba kuvvet yöntemi) kullanılmıştır. En kritik hiper parametre olan learning rate [49,50], yitim fonksiyonunu minimum yapmaya çalışan optimizasyon'un adım büyüklüğünü tanımlamaktadır [51,52]. Küçük verilirse eğitim uzun sürecek, büyük verilirse modelin yakınsaması atlanabileceğinden YOLO V4 için belirlenen learning rate değeri 0.0013, YOLO V5 ve Faster R-CNN için 0.001'dir. Kullanılan GPU/CPU'nun hafızasına bağlı olarak ayarlanan batch değeri YOLO V4 için 128, YOLO V5 için 16, CPU'da çalışan Faster R-CNN için 2 verilmiştir. Ayrıca sadece YOLO V4'te olan *subdivisions* değeri 128 verilerek batch/subdivision eşitliği ile her bir iterasyonda sadece 1 adet görüntünün YOLO V4 modeline kabul edileceği ifade edilmiştir. Ağırlıkların gelecekteki değişimine, geçmişin ne kadar etki edeceğini belirleyen momentum YOLO V4 için 0.949, YOLO V5 ve Faster R-CNN için 0.9 seçilmiştir. Tipik özellikler için zayıf ağırlık güncellemeleri yaparak veri setindeki dengesizliği elimine eden *weight decay*, 3 modelde de [53] çalışmasındaki gibi 0.0005'e ayarlanmıştır. Momentum ve ağırlık kaybının model performansına olumlu katkı yaptığı birçok çalışmada görülmüştür.

Sinir ağlarını eğitmede ikinci önemli nokta, kayıp fonksiyonu tanımlamak ve kullandığı geri yayılım algoritmasının optimizasyonu için doğru algoritmayı seçmektir. Global minimum arayışında optimizasyon algoritmaları genelde gradyan azalma stratejisini kullanır. YOLO V4 ve V5'te varsayılan SGD yerine Adam optimizator'ü [54] seçilerek gerçek çıktıyla tahmin edilen çıktı

arasındaki mesafeyi temsil eden, çoklu sınıflandırma problemlerinde sıkça kullanılan Cross Entropy Loss fonksiyonu minimum yapılmaya çalışılmıştır. Yitim fonksiyonu gradyanının hesaplanması için her örneğin hatası toplanmaktadır. SGD (Stochastic Gradient Descent) tüm eğitim hatalarını (yitimi) toplamak yerine her iterasyonda tek bir örneğin hata gradyanını kullanmaktadır [55]. Fakat standart SGD yavaştır ve çalıştığı ağı, lokal optimaya yakalandırabilmektedir. Bundan dolayı Faster R-CNN için optimizasyon algoritması olarak önceki ağırlıkları dikkate alan, SGDM (momentumlu SGD) tercih edilmiştir.

2.5 Transfer öğrenme

Derin öğrenme, geleneksel makine öğrenmesi metodlarına göre yüksek sayıda eğitim verisine ihtiyaç duymaktadır. Bu sebepten ötürü işin özü görüntü sınıflandırma derin ağları, genelde milyon sayıda görüntü içeren veri setleri üzerinde aralıksız birkaç hafta süren çoklu bilgisayar sunucularında eğitilmektedir [56]. Örneğin Google tarafından yakın zamanda geliştirilen dil sunum modeli BERT, aralıksız 4 gün boyunca 16 Bulut TPU (tensör işleme birimi)'sunda (yaklaşık 60 masaüstü bilgisayarda) eğitilmiştir. Ayrıca MIT çalışmasına göre 200 milyon veya daha üstü parametrelili bir derin sinir ağının Bulut TPU'daki eğitimi, 6 arabanın kullanım ömrü boyunca ürettiği CO₂'ye eşdeğerdir [57].

Her derin öğrenme ağını eğitirken ihtiyaç duyulan bolca veriyi bulamadığımız için bir makine öğrenmesi tekniği olarak transfer öğrenme geliştirilmiştir [58]. Bu teknik, uzun eğitim sürelerine, gereksiz kaynak harcamasına ve doğaya zararlı karbondioksit salınımına karşı kısıtlı veri durumlarında kullanılmaktadır. Geçmişte karşılaşılan probleme karşı üretilen çözümün bilgisi, daha sonra karşılaşılan farklı ama benzer problemlere de uygulanabilir. Örneğin daktilo yazmayı öğrenmiş birisi, ileride bilgisayar kullanmaya çalıştığında fazla zorlanmayacaktır.

Transfer öğrenme araştırmalarının ana motivasyonu; insanların önceden öğrendiği bilgiyi, yeni problemleri zekice daha hızlı ve iyi çözmesinde kullanmasıdır. Teknikte sıkça kullanılan iki kavram; kaynak ve hedeftir. Yüksek miktarlı veride çalışmış derin öğrenme modelini temsil eden kaynak, düşük miktarlı veride çalışacak modeli temsil eden ise hedeftir. Transfer öğrenme işlemi basitçe, kaynak çıktısı ağırlıklarının hedef girdisine transferidir [59-62]. Transfer işlemi; hedef verisi küçük, kaynak verisi çok büyükse anlam kazanmaktadır. Ek olarak uygun transferi yapabilmek için kaynak ve hedefteki girdinin tipi (yazı, ses, görüntü, video) ve boyutu (1, 2, 3) aynı olmalıdır [63]. Veri, hiper spektral görüntüler [64,65] veya SAR [66,67] gibi internet görüntülerinden tamamen farklı bir karaktere sahipse modele transfer öğrenme uygulamadan, en baştan eğitim düşünülebilir.

Günlük hayatta karşılaşılabileceğimiz envai çeşitte nesnenin görüntüsünden oluşan COCO [68], Pascal VOC [69] ve ImageNet [70] gibi büyük veri setlerinde eğitilmiş sinir ağları, transfer öğrenme yaklaşımı için ESA tabanlı derin öğrenme modellerinde kullanılmaktadır. Çalışmada kullanılan ön eğitilmiş ImageNet ağırlıkları ile ilklendirme, eğitim süresini kısaltırken modelin genelleme

performansında, rasgele ağırlıklar ile ilklendirmeye göre iyileşmeler gösterebilir. Bu veri setlerinde eğitilen ESA'lar; ilk katmanlardan nesnelere kenar, leke, renk gibi genel (ortak) özelliklerini öğrenirken ileriki katmanlardan probleme özgü, nesnelere detaylı özelliklerini öğrenmektedir. Metot kısmında bahsedildiği üzere ESA'ların son katmanı olan sınıflandırıcı (Softmax'li tam bağlı katman) ise öncesindeki katmanlardan gelen görüntü özelliklerinin hangi nesneye ait olduğunu belirleyerek ağı sonlandırmaktadır. Sınıflandırıcıdan önceki katmanlarda öğrenilen bilgi, transfer öğrenme tekniği ile yeni ağı eğitmede kullanılmaktadır. Dolayısıyla yeni ağıdaki öğrenme hızı artacak, eğitim süresi düşecektir. Son olarak yeni ağıdan çıkarılan sınıflandırıcı katmanı yerine probleme özgü sınıflandırıcı (çalışmada iki çıktılı) eklenerek tespit işlemi tamamlanacaktır.

Matematiksel formülizasyon ile tekniği açıklayacak olursak; [71] D çalışma alanı, özellik uzayı X ve marjinal uzayı $P(X)$ 'ten oluşmaktadır.

$$P(X), X = \{x_1, x_2, \dots, x_n\}, x_i \in X \quad (3)$$

Denklem (3)'teki x_i özellik vektörünü göstermektedir. İş T ; etiket uzayı Y ve özellik vektörü, etiket çiftlerinden öğrenen tahmin fonksiyonu veya $P(Y|x)$ 'ten oluşmaktadır. Yani n , çalışma alanındaki her özellik vektörü için karşılığındaki etiketi tahminlemektedir. Bu durum Denklem (5) ile gösterilmiştir.

$$Y = \{y_1, y_2, \dots, y_n\}, y_i \in Y \quad (4)$$

$$n(x_i) = y_i \quad (5)$$

Hedef etiketli verisi, kaynak etiketli verisinden çok daha küçükken kaynak çalışma alanı D_S ve karşılığındaki kaynak işi T_S , hedef çalışma alanı D_T ve hedef işi T_T olsun. Transfer öğrenmenin amacı $D_S \neq D_T$ veya $T_S \neq T_T$ durumlarında D_S ve T_S 'den bilgi eldelerken D_T 'deki $P(Y_T|X_T)$ hedef durumsal olasılık dağılımını öğrenmedir.

Transfer öğrenme çalışmasında üç ana araştırma sorusu vardır, bunlar:

- 1 Ne transfer edilmeli?
- 2 Nasıl transfer edilmeli?
- 3 Ne zaman transfer edilmeli?'dir.

Üçüncü sorunun cevabı hedefin, kaynaktan küçük olduğu durumlardır. İlk iki soru olan ne ve nasıl transfer etmeliye cevap ise esasen şu 4 kategoriye ayrılmaktadır: (1) örnek tabanlı transfer öğrenme: kaynak görevdeki etiketli verinin hedef görevde kullanılması için yeniden ağırlık verilmesi, (2) özellik tabanlı transfer öğrenme: kaynak alanı, hedef alanı ve model hataları arasındaki farkları azaltma, (3) parametre tabanlı transfer öğrenme: derin öğrenmede popüler olan paylaşılan parametrelerin ve birincil bilginin görevler arası keşfedilmesi, (4) ilişkisel bilgi tabanlı transfer öğrenme: daha az keşfedilmiş olan kaynak görevden hedef göreve, ilişkiyi veri boyunca transfer etme. Çalışmada bilginin yayılımı, 3. kategori olan parametre paylaşımı ile sağlanacaktır.

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 edememektir. 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 nesnelere 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 nesnelere ne kadarının doğru tespit edildiğini göstermektedir. Gözden kaçırılan tespitler olan FN bu eşitlikte ağırlıklıdır.

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

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.

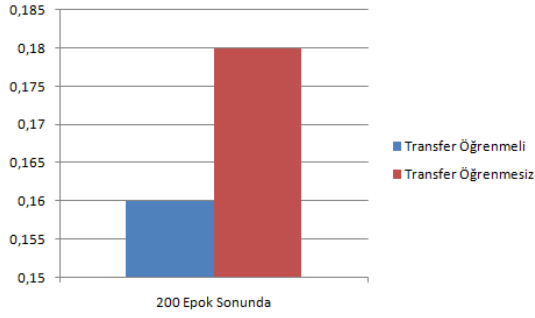
3 Bulgular ve tartışma

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.

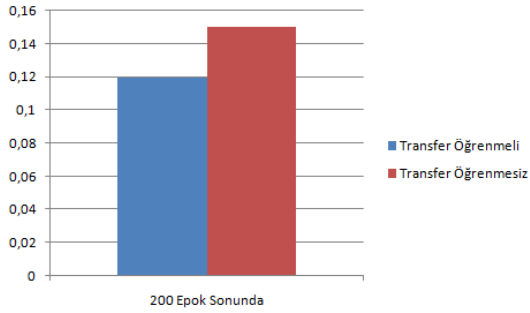
Epok sayısının 200 tutulmasının yeterli olduğu erken durdurma kriteri kullanılarak kanıtlanmıştır. Erken durdurma kriterinin kullanım amacı, model eğitimini kullanıcı yerine keserek eğitim boyunca oluşan en iyi ağırlığı bulmaya yöneliktir. YOLO V5 için denenen bu yöntemde 89. epoktan sonraki 111 epokun 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.

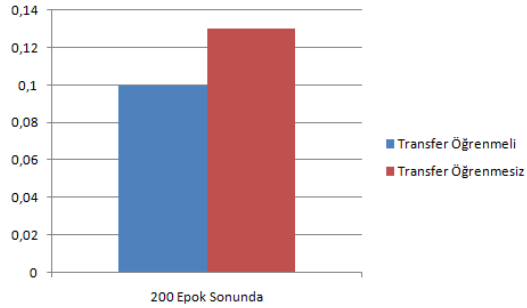
Şekil 3,4 ve 5'teki grafiklerde sırayla transfer öğrenme tekniğinin uygulanmadığı ve uygulandığı Faster R-CNN, YOLO V4 ve YOLO V5 modellerinin eğitim boyunca hesapladıkları yitim değerinin değişimini görmekteyiz. Her 3 grafikten anlaşılacağı üzere transfer öğrenmenin uygulanmadığı modellerin yitim değerleri 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.



Şekil 3. Faster R-CNN transfer öğrenmenin uygulanmadığı/transfer öğrenmeli 200 epokluk eğitim sonucundaki yitim değeri



Şekil 4. YOLO V4 transfer öğrenmenin uygulanmadığı/transfer öğrenmeli 200 epokluk eğitim sonucundaki yitim değeri



Şekil 5. YOLO V5 transfer öğrenmenin uygulanmadığı/transfer öğrenmeli 200 epokluk eğitim sonucundaki yitim değeri

Kayıp değeri, bizim problemimiz için kask tespit hatasını göstermektedir. Yani ne kadar büyük bir yitim değeri çıkarsa o kadar yanlış tespit yapılmış demektir. 200 epok sonucunda, 6 model arasında 0.1 ile en düşük (başarılı) yitim değerine ulaşan model, transfer öğrenmeli YOLO V5'tir. Transfer öğrenmeli YOLO V5 modeli %98 fl skor ile en yüksek başarıma ulaşmıştır.



Şekil 6. Transfer öğrenmeli YOLO V5 modelinden birkaç başarılı örnek

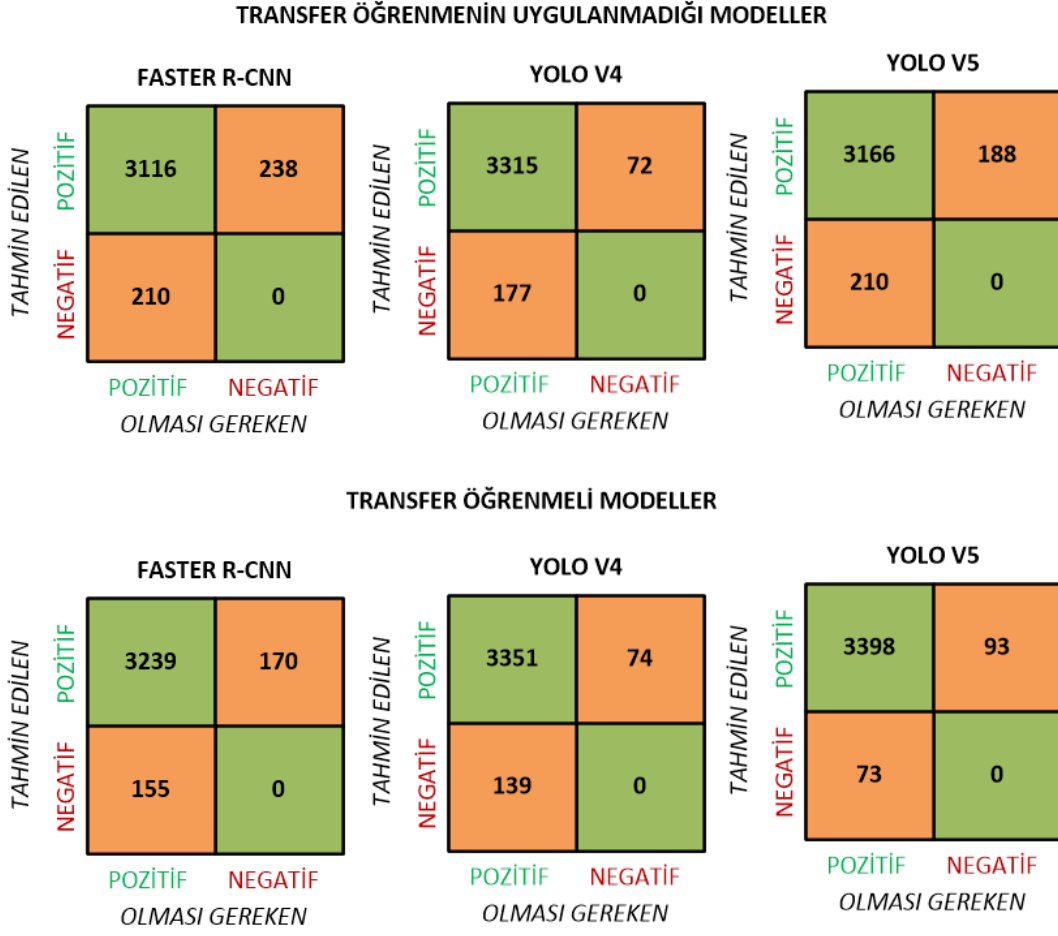
Tablo 1'den görüleceği üzere transfer öğrenmenin uygulandığı modeller, 3.500 adet eğitim ve validasyon görüntüsü üzerindeki 200 epokluk eğitimlerinde, süre açısından bir avantaj sağlamamış aynı zamanda kayda değer bir zaman kaybına sebep de açmamıştır. Tüm modellerin en baştan eğitimleri, transfer öğrenmenin uygulandığı modellere göre az da olsa daha kısa sürede tamamlanmıştır.

Transfer öğrenmenin uygulanmadığı modellerin doğru tespit oranları transfer öğrenme uygulananlara göre daha az çıkmıştır. Transfer öğrenme modellerin yüksek başarımlı çıkmasındaki sebep yanlış tespit oranlarının daha az olmasıdır.

6 modelin 750 test görüntüsündeki 3.564 adet nesneyi tespit başarımlı karşılaştırmasını sunan karışıklık matrisleri Şekil 7'de gösterildiği gibidir.

Tablo 1. Modellerin eğitim süresi karşılaştırması

Eğitim modelleri (200 Epok)	Eğitim süreleri (saat)
Transfer öğrenmesiz faster R-CNN	64.53
Transfer öğrenmeli faster R-CNN	65.58
Transfer öğrenmesiz YOLO V4	15.13
Transfer öğrenmeli YOLO V4	15.30
Transfer öğrenmesiz YOLO V5	4.65
Transfer öğrenmeli YOLO V5	4.84



Şekil 7. Modellerin karışıklık matrisleri

Tablo 2. Modellerin doğruluk karşılaştırması

Model/Metrik	Doğruluk	Hassaslık	Duyarlılık	F1 Skor
Transfer öğrenmeli YOLO V4	0.94	0.98	0.96	0.97
Transfer öğrenmeli YOLO V5	0.95	0.97	0.98	0.98
Transfer öğrenmeli faster R-CNN	0.91	0.96	0.95	0.95
Transfer öğrenmesiz YOLO V4	0.93	0.98	0.95	0.96
Transfer öğrenmesiz YOLO V5	0.89	0.94	0.94	0.94
Transfer öğrenmesiz faster R-CNN	0.87	0.93	0.94	0.93

Transfer öğrenmenin uygulanmadığı modellerin doğru tespit oranları transfer öğrenme uygulananlara göre daha az çıkmıştır. Transfer öğrenme modellerin yüksek başarımlarını çıkarmadaki sebep yanlış tespit oranlarının daha az olmasıdır.

4 Sonuçlar

Görüntülerdeki kaskların tespiti güvenliği artırmakta bu sayede toplumsal fayda sağlanabilmektedir. Gün geçtikçe değişen dünyada inşaat alanları hep aynı kalmaktadır. Yapı işlerini hallenen işçilerin kafa yaralanmasına karşı önlem şeklinde alınan kask materyali bu açılardan değerlidir. Göz ile kaskın tespitinin yapılması zor olacağından otomatik tespit bu çalışmaya ile önerilmiştir.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı: %5

Kaynaklar

- [1] X. Chang and X. M. Liu, Fault tree analysis of unreasonably wearing helmets for builders. *Journal of Jilin Jianzhu University*, 35, 67–71, 2018.
- [2] L. Perezand and J. Wang, The effectiveness of data augmentation in image classification using deep learning. *arXiv*, 2017. <https://doi.org/10.48550/ARXIV.1712.04621>.
- [3] V. Sorin, Y. Barash, E. Konen, E. Klang, Deep-learning natural language processing for oncological applications. *The Lancet Oncology*, 21(12), 1553-1556, 2020.
- [4] H. S. Bae, H. J. Lee and S. G. Lee, Voice recognition based on adaptive MFCC and deep learning. *IEEE 11th Conference on Industrial Electronics and Applications*, pp. 1542-1546, Chengdu, China, 2006.
- [5] Z. Yuan, Y. Lu, Z. Wang and Y. Xue, Droid-sec: deep learning in android malware detection. *Proceedings of the 2014 ACM conference on SIGCOMM*, pp. 371-372, New York, United States, 2014.
- [6] Y. Sun, X. Wang, X. Tang, Hybrid deep learning for face verification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38, 1489–1496, 2013. <https://doi.org/10.1109/ICCV.2013.188>.
- [7] R. Vinayakumar, K. P. Soman and P. Poornachandran, Applying convolutional neural network for network intrusion detection. *Proceedings of the 2017 International Conference on Advances in Computing*, pp. 1222-1228, Udipi, India, 2017.
- [8] K. Teja, L. Jens, S. Felix, H. Stefan, Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 24-49, 2021. <https://doi.org/10.1016/j.isprsjprs.2020.12.010>.
- [9] S. Moccia, L. Romeo, L. Migliorelli, E. Frontoni, P. Zingaretti, Supervised CNN strategies for optical image segmentation and classification in interventional medicine. *Deep Learners and Deep Learner Descriptors for Medical Applications*, 213-236, 2020. https://doi.org/10.1007/978-3-030-42750-4_8
- [10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211-252, 2015. <https://doi.org/10.1007/s11263-015-0816-y>
- [11] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*, pp. 740-755, Zurich, Switzerland, 2014.
- [12] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303-338, 2010.
- [13] A. Kelm, L. Laußat, A. Meins-Becker, D. Platz, M. J. Khzaee, A. M. Costin, M. Helmus, J. Teizer, Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. *Automation in Construction*, 36, 38–52, 2013. <https://doi.org/10.1016/j.autcon.2013.08.009>.
- [14] S. Barro-Torres, T. M. Fernandez-Carames, H. J. Perez-Iglesias, C. J. Escudero, Real-time personal protective equipment monitoring system. *Computer Communications*, 36, pp. 42–50, 2012. <https://doi.org/10.1016/j.autcon.2013.08.009>.
- [15] A. H. M. Rubaiyat, T. T. Toma, M. Kalantari-Khandani, S. A. Rahman, L. Chen, Y. Ye and C. S. Pan, Automatic detection of helmet uses for construction safety. *Proceedings of the 2016 IEEE ACM International Conference on Web Intelligence Workshops*, Omaha, USA, 2016.
- [16] T. Malisiewicz, A. Gupta, and A. A. Efros, Ensemble of exemplar-svms for object detection and beyond. *2011 IEEE International Conference*, 89-96, 2011.
- [17] C. C. Liu, J. S. Liao, W. Y. Chen and J. H. Chen, The full motorcycle helmet detection scheme using canny detection, *18th IPPr Conf. On CVGIP*, 2005.
- [18] M. H. Wuand and J. Zhao, Automated visual helmet identification based on deep convolutional neural networks. *Proceedings of the 13th International Symposium on Process Systems Engineering*, San Diego, USA, 2018.
- [19] Z. Fangbo, Z. Huailin and N. Zhen, Safety helmet detection based on YOLOv5. *2021 IEEE International Conference on Power Electronics, Computer Applications*, pp. 6-11, 2021.
- [20] W. Fan, J. Guoqing, G. Mingyu, H. E. Zhiwei and Y. Yuxiang, Helmet detection based on improved YOLO V3 deep model. *2019 IEEE 16th International Conference on Networking, Sensing and Control*, pp. 363-368, 2019.
- [21] L. Yange, W. Han, H. Zheng, H. Jianling, W. Weidong, Deep learning-based safety helmet detection in engineering management based on convolutional

- neural networks. Hindawi Advances in Civil Engineering, 2020.
- [22] T. Choudhury, A. Aggarwal and R. Tomar, A deep learning approach to helmet detection for road safety. NISCAIR-CSIR, India, 2020.
- [23] D. Madhuchhanda, B. Oishila and C. Sanjay, Automated helmet detection for multiple motorcycle riders using CNN. 2019 IEEE Conference on Information and Communication Technology, pp. 1-4, IEEE, 2019.
- [24] J. Wei, X. Shiquan, L. Zhen, Z. Yang, M. Hai, L. Shujie, Y. Ye, Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. IET Image Processing, 15(14), 3623-3637, 2021. <https://doi.org/10.1049/ipr2.12295>.
- [25] T. Shilei, L. Gonglin, J. Ziqiang and H. Li, Improved YOLOv5 network model and application in safety helmet detection. 2021 IEEE International Conference on Intelligence and Safety for Robotics, pp. 330-333, IEEE, 2021.
- [26] G. Rui, M. Yixuan and H. Wanhong, An improved helmet detection method for YOLOv3 on an unbalanced dataset. 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communication, pp. 328- 332, IEEE, 2021.
- [27] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, Microsoft COCO: Common Objects in Context. European Conference on Computer Vision, pp. 740-755, Zurich, Switzerland, 2014.
- [28] Z. Jiang, L. Zhao, S. Li and Y. Jia, Real-time object detection method for embedded devices. computer vision and pattern recognition, 2020.
- [29] C. Kumar and R. Punitha, YOLOv3 and yolov4: Multiple object detection for surveillance applications. 2020 Third international conference on smart systems and inventive technology, pp. 1316-1321, IEEE, 2020.
- [30] L. Yanfen, W. Hanxiang, L. M. Dang, T. N. Nguyen, D. Han, A. Lee, I. Jang, H. Moon, A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access 8, 194228-194239, 2020.
- [31] T. Liu, B. Pang, L. Zhang, W. Yang, X. Sun, Sea surface object detection algorithm based on YOLO v4 fused with reverse depth wise separable convolution (RDSC) for USV. Journal of Marine Science and Engineering, 9(7), 753. <https://doi.org/10.3390/jmse9070753>.
- [32] A. M. Roy, R. Boseand, J. Bhaduri, A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. Neural Computing and Applications, 34(5), 3895-3921, 2022. <https://doi.org/10.48550/arXiv.2111.00298>.
- [33] J. Yu and W. Zhang, Face mask wearing detection algorithm based on improved YOLO-v4, sensors. Sensors, 21(9), 3263, 2021. <https://doi.org/10.3390/s21093263>.
- [34] S. Li, Y. Li, Y. Li, M. Liand and X. Xu, YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. IEEE Access, 9, 141861-141875, 2021.
- [35] D. Wu, S. Lv, M. Jiang, H. Song, Using channel pruning-based YOLOv4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 178, 105742. <https://doi.org/10.1016/j.compag.2020.105742>.
- [36] J. Yuand and H. Choi, YOLO MDE: Object detection with monocular depth estimation. Electronics, 11(1), 76, 2021. <https://doi.org/10.3390/electronics11010076>.
- [37] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507. doi: 10.1126/science.1127647.
- [38] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11)i 2278-2374, 1998.
- [39] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11)i 2278-2374, 1998.
- [40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211-252, 2015. <https://doi.org/10.1007/s11263-015-0816-y>
- [41] CS230 Deep Learning, Class for learners. <https://cs230.stanford.edu/>, Accessed 03 November 2022.
- [42] Make ML, Hard Hat Dataset. <https://makeml.app/datasets/hard-hat-workers>, Accessed 15 April 2022.
- [43] J. Grum, Book review: pattern recognition and neural networks by B.D. Ripley. International Journal of Microstructure and Materials Properties, 4(1), 146-147, 2009. <https://doi.org/10.1504/IJMMP.2009.028441>.
- [44] make-sense, a labelling tool. <https://github.com/SkalskiP/make-sense>, Accessed 13 March 2022.
- [45] glenn-jocher, YOLOv5 Focus Layer, Explanation of Focus Layer. <https://github.com/ultralytics/yolov5/discussions/3181m1>, Accessed 22 February 2022.
- [46] Mseddi, W. Souidine, M. A. Sedrine and R. Attia, YOLOv5 based visual localization for autonomous vehicles, Proceedings of the 2021 29th European Signal Processing Conference, Dublin, Ireland, 2021.
- [47] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Guiand and W. Wang, Face mask recognition system with YOLO V5 based on image recognition. 2020 IEEE 6th International Conference on Computerand Communications, pp. 1398-1404, IEEE, 2020.
- [48] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink and J. Schmidhuber, LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222-2232, 2017.
- [49] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, Cambridge, 2016.

- [50] C. M. Bishop and N. M. Nasrabadi, Pattern recognition and machine learning. Springer-Verlag, Vol 4. No 4., New York, 2007.
- [51] K. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge, 2012.
- [52] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst, 60(6), 84-90, 2017. <https://doi.org/10.1145/3065386>.
- [53] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization. arXiv, 2014. <https://doi.org/10.48550/arXiv.1412.6980>.
- [54] P. D. Lima and K. Marfurt, Convolutional neural network for remote-sensing scene classification: transfer learning analysis. Remote Sensing, 12(1), 86, 2019. <https://doi.org/10.3390/rs12010086>.
- [55] C. Szegedy, W. Liu, Y. Jia et al, Going deeper with convolutions. arXiv, 2014.
- [56] K. Haoarchive, Training a single AI model can emit as much carbon as five cars in their lifetimes. <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/>, Accessed 8 March 2022.
- [57] L. Torrey and J. Shavlik, Transfer Learning Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Publishing, Hershey, 2009.
- [58] Y. Gao, Y. Ruan, C. Fang and S. Yin, Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. Energy and Buildings, 223, 110156, 2020. <https://doi.org/10.1016/j.enbuild.2020.110156>.
- [59] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang and C. Liu, A survey on deep transfer learning. International conference on artificial neural networks, pp. 270-279 Springer, Cham, 2018. <https://doi.org/10.48550/arXiv.1808.01974>.
- [60] B. Koçer, Transfer öğrenme yeni yaklaşımlar. Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü. Türkiye, 2012.
- [61] Transfer learning & fine-tuning, explanation of layer-base fine-tune and transfer learn methods <https://keras.io/guides/transfer-learning/>, Accessed 27 December 2021.
- [62] F. İ. Eyiokur, Deep convolutional neural network based unconstrained ear recognition. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Türkiye, 2018.
- [63] M. J. Afridi, A. Ross, E. M. Shapiro, On automated source selection for transfer learning in convolutional neural networks. Pattern recognition, 73, 65-75, 2018. <https://doi.org/10.1016/j.patcog.2017.07.019>.
- [64] M. Tharani, N. Khurshid, M. Taj, Unsupervised Deep Features for Remote Sensing Image Matching via Discriminator Network. Remote Sensing Image Retrieval, 2018.
- [65] S. J. Pan and Q. Yang, A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359, 2009.
- [66] M. Gong, H. Yang, P. Zhang, Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. ISPRS Journal of Photogrammetry and Remote Sensing, 129, 212-225, 2017.
- [67] L. Hughes, M. Schmitt, L. Mou, Y. Wang, X. Zhu, Identifying Corresponding Patches in SAR and Optical Images With a Pseudo-Siamese CNN. IEEE Geoscience and Remote Sensing Letters, 15(5), 784-788, 2018.
- [68] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, Microsoft COCO: Common Objects in Context. European Conference on Computer Vision, pp. 740-755, Zurich, Switzerland, 2014.
- [69] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338, 2010
- [70] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei. IEEE Computer Vision and Pattern Recognition (CVPR) (2009)
- [71] S. J. Pan and Q. Yang, A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359, 2009.
- [72] S. Wang, L. Niu and N. Li, Research on image recognition of insulators based on YOLO algorithm, 2018 international conference on power system technology (POWERCON), IEEE, 2018.
- [73] M. Kartal, O. Duman, Ship detection from optical satellite images with deep learning. In 2019 9th International Conference on Recent Advances in Space Technologies (RAST), 2019.
- [74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimsheine, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 32, 2019. <https://doi.org/10.48550/arXiv.1912.01703>.

