



Derin Öğrenme Algoritması (YOLO) ile Dinamik Test Süresince Süspansiyon Parçalarında Çatlak Tespiti

Muhammed Abdullah Özel^{1*}, Selim Sefa Baysal², Mustafa Şahin³

¹ Aydınlar Yedek Parça San. ve Tic. A.Ş., Konya, Türkiye, (ORCID: 0000-0003-3056-6438), ozel.muhammed@aydtr.com

² Aydınlar Yedek Parça San. ve Tic. A.Ş., Konya, Türkiye, (ORCID: 0000-0002-7439-4087), baysal.selimsefa@aydtr.com

³ Aydınlar Yedek Parça San. ve Tic. A.Ş., Konya, Türkiye, (ORCID: 0000-0002-1803-1599), sahin.mustafa@aydtr.com

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(DOI: 10.31590/ejosat.952798)

ATIF/REFERENCE: Özel, M. & Baysal, S. & Şahin, M. (2021). Derin Öğrenme Algoritması ile Dinamik Test Süresince Süspansiyon Parçalarında Çatlak Tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 1-5.

Öz

Günümüzde Derin Öğrenme yöntemlerinin kullanımı her alanda artmıştır. Yapay Zekâ, bilgisayarın veya bilgisayar kontrolünde bir makinenin zeki canlılara benzer mekanizmalarla karar verme yeteneğine sahip olmasıdır. Kısaca Yapay Zekâ, bilgisayarın insanlar gibi düşünmesini sağlar. Derin Öğrenme bir veya daha fazla gizli katman içeren yapay sinir ağları ve benzeri makine öğrenme algoritmalarını kapsayan çalışma alanıdır. Yani en az bir adet yapay sinir ağının kullanıldığı ve birçok algoritma ile, bilgisayarın elindeki verilerden yeni veriler elde etmesidir. Derin Öğrenme alanında kullanılan birçok algoritma vardır. Bu algoritmalarından YOLO (You Only Look Once) algoritması ve Darknet modeli, yüksek işleme hızından dolayı yüksek FPS (Frame Per Second) sağlamakta ve daha net sonuçlar vermektedir. Bu sebeple bu çalışmada YOLO algoritması tercih edilmiştir. Algoritmanın 4 farklı versiyonu için denemeler yapılmış, sonuçlar karşılaştırılmış, tespit doğruluğu ve hız olarak en iyi sonuç Versiyon-4 algoritmasında sağlanmıştır. OpenCV, NumPy, SciPy gibi Python programlama dili kütüphaneleri kullanılarak tespit edilen deformelerin sayısı tespit edilmiş, tespit anları kayıt altına alınmış, bu algoritmaların test kontrol sistemi ile bilgi alışverişi sağlanarak test sisteminin durdurulması sağlanmıştır. Deformasyonlar için büyük bir veri seti oluşturulmuş ve oluşturulan bu veri seti 4 farklı algoritma versiyonlarıyla eğitim yapılmış ve uygulamaya geçilmiştir. Otomotiv yan sanayinde üretilen süspansiyon sistem bileşenlerin performansları dinamik testlerle test edilmektedir. Bu testlerde parçalar plastik deformasyona maruz kalana kadar ya da ani kırılma başlangıcı olan çatlak oluşumunda, test sisteminin durdurulması insan kontrolünde yapılmaktadır. Ancak bu durumda, parçaların ilk deformasyon anı tespit edilememektedir ve aynı zamanda zaman kaybına yol açmaktadır. Bu çalışmada derin öğrenme algoritmalarından olan YOLO ile deformasyon ve çatlak oluşumu tespit edilmeye çalışılmıştır. Çalışmada ayrıca derin öğrenme, görüntü işleme ve Python kütüphanelerinin nesne tespitinde kullanımları ayrıntıyla incelenmiş, uygulama oluşturulmuş, Python programı ve Yapay Sinir Ağları kullanılarak sonuçların elde edilmesi sağlanmıştır.

Anahtar Kelimeler: Derin Öğrenme, YOLO, Görüntü İşleme, Süspansiyon Parçaları, Deformasyon.

Crack Detection in Dynamic Test Drive Suspension Parts with Deep Learning Algorithm (YOLO)

Abstract

Nowadays, the use of Deep Learning methods has increased in many areas. Artificial Intelligence, which includes deep learning, is the ability of a computer or a computer-controlled machine to make a decision similar to intelligent creatures. In short, Artificial Intelligence enables the computer to think like a human. Deep Learning is a field of study that includes neural networks with one or more hidden layers and similar machine learning algorithms. In other words, in deep learning, the computer uses at least one artificial

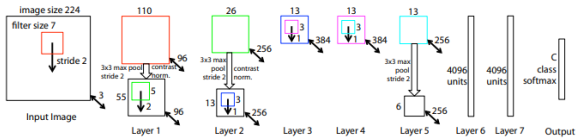
* Sorumlu Yazar: Aydınlar Yedek Parça San. ve Tic. A.Ş., Konya, Türkiye, ORCID: 0000-0003-3056-6408, ozel.muhammed@aydtr.com

neural network and obtains new data from the data it has with different algorithms. There are many algorithms used in the Deep Learning field. Among these algorithms, YOLO (You only look once) algorithm and Darknet model provide higher FPS (Frame Per second) due to high processing speed and give clearer results. For this reason, the YOLO algorithm has been preferred in the application. Trials have been made for 4 different versions of the algorithm, the results have been compared, the best result in terms of detection accuracy and speed has provided in the Version-4 algorithm. Using Python programming language libraries such as OpenCV, NumPy, and SciPy, the number of detected deformities were determined, their detection moments were recorded, and the test system was stopped by providing information exchange with the test control system of these algorithms. A large dataset has been created for deformations and this data set has been trained and implemented with 4 different algorithm versions. The performances of the suspension system components produced in automotive reflectivity are tested with dynamic tests. In these tests, the test system is stopped under human control until the parts are subjected to plastic deformation or in case of crack formation that is a sudden breakout. However, in this case, the first moment of deformation of the parts cannot be detected and at the same time causes a waste of time. In this study, the use of deep learning, image processing and Python libraries in object detection has been examined in detail, the application has been created, and the results have been obtained by using the Python program and Artificial Neural Networks.

Keywords: Deep Learning, YOLO, Image Processing, Suspension Parts, Deformation.

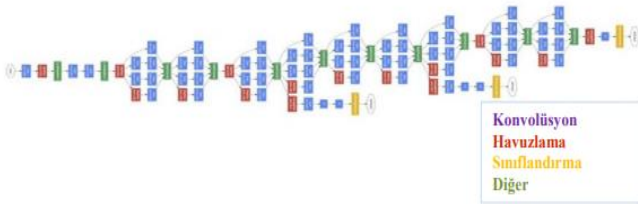
1. Giriş

2012 yılında Alexnet'in ImageNet yarışmasını kazanmasının ardından yapılan yarışmalarda derin öğrenme modelleri kullanılmaya başlandı. Matthew Zeiler ve Rob Fergus tarafından tasarlanan ZFNet (Zeiler and Fergus 2014) 2013 yılında ImageNet yarışmasının kazananı olmuştur. Bu model ile nesne tanımda hata oranı yüzde 11,2'ye indirilmiştir. 7x7 boyutundaki filtreleri ve havuzlama katmanında 2 adım kayma miktarı kullanılmıştır. Bu değişikliğin arkasındaki mantık, birinci konvolüsyon katmanındaki daha küçük bir filtre boyutunun, giriş boyutundaki birçok orijinal piksel bilgisinin korunmasına yardımcı olmasıdır[6].



Şekil 1. ZF Net mimarisi (Zeiler and Fergus 2014)

GoogLeNet (Szegedy, Liu et al. 2015) yapısındaki Inception modüllerinden dolayı karmaşık bir mimaridir. GoogLeNet 22 katmanlı ve yüzde 5,7 hata oranı ile ImageNet 2014 yarışmasının kazananı olmuştur. Bu mimari genel olarak, ardışık bir yapıda konvolüsyon ve havuzlama katmanlarını üst üste istiflemekten uzaklaşan ilk CNN mimarilerinden biridir. Ayrıca bu yeni model bellek ve güç kullanımı üzerinde önemli bir yere sahiptir. Katmanların hepsini yığılmak ve çok sayıda filtre eklemek, hesaplama ve bellek maliyeti getirir ve ezberleme olasılığını artırır. GoogLeNet bu durumun üstesinden gelmek için paralel olarak birbirine bağlı modüller kullanılmıştır[6].



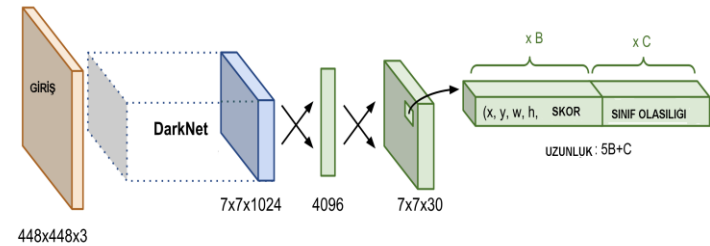
Şekil 2. GoogLeNet ağ mimarisi (Szegedy, Liu et al. 2015)

ResNet (He, Zhang et al. 2016) şu ana kadar olan tüm mimarilerden daha derin olarak tasarlanan bir mimaridir. 152 katmandan oluşmaktadır. Aynı zamanda yüzde 3,6 hata oranı ile ImageNet 2015 yarışmasının kazananı olmuştur. Becerileri ve uzmanlıklarına bağlı olarak, insanlar genelde yüzde 5-10 hata oranına sahiptir[6].



Şekil 3. Microsoft ResNet ilk 34 katmanın ağ mimarisi (He, Zhang et al. 2016)

YOLO ImageNet'te eğitilmiş 53 katmanlı bir ağa sahip olan bir Darknet çeşidi kullanır. Bu bizim backbone katmanımızdır. Algılama görevi için, üzerine 53 katman daha yığılır ve bu da bize YOLO için 106 kat tamamen evrişimli bir mimari sağlar. ResNet ile karşılaştırıldığında, Darknet-53 daha iyi performansla sahiptir ve 1,5 kat daha hızlıdır. Kısayol bağlantılarına sahip Darknet-53 ve özellik harita örnekleme ve birleştirme özelliklerine sahiptir.



Şekil 4. DarkNet-53 Backbone gösterimi[1].

YOLO algoritması araba, kamyon, yaya, bisikletli ve hatta hayvanları tanımlama yetisine ve son derece hızlı bir algoritmaya sahiptir. YOLO modeli görüntüleri 416x416 pikselde 22 milisaniyede çalışır. SSD (Sigle Shot Detection)

kadar hassas ve 3 kat daha hızlıdır. YOLO modeli diğer ağ yapılarına göre çok daha iyi bir ağ yapısına sahiptir. Diğer sistemlere göre yaklaşık 4 kat daha hızlı olan bu model sadece modelin boyutunu değiştirerek kolayca hız ve doğruluk arasında geçiş yapmaya olanak tanır. En önemli özelliği ise yeniden eğitim gerektirmez. Diğer CNN tabanlı sistemlerin çoğu görüntüdeki bir nesneyi algılamak için sınıflandırıcıları yeniden kullanır. Yaygın olan nesne algılama yöntemleri ile karşılaştırıldığında, nesnenin sınıflandırılması ve konumlandırılması için tek CNN uygular. Görüntüleri yaklaşık 30-45FPS hızında işleyebilir, bu yüzden oldukça hızlıdır. Bu, video akışının gerçek zamanlı olarak işlenebileceği ve birkaç milisaniye ihmal edilebilir bir gecikme süresi olduğu anlamına gelir[15].

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	77.6	93.8	29.4	1090	37
Darknet-53	77.2	93.8	18.7	1457	78

Şekil 5. Backbone karşılaştırma[9].

2. Materyal ve Metot

2.1. Algoritma

YOLO artan bir iyileştirilme için geliştirilmiş bir algoritmadır. Bir süredir pek çok nesne algılama algoritması olduğu için rekabet, nesnelere ne kadar doğru ve hızlı bir şekilde tespit edildiğiyle ilgilidir. Çalışmamızda en yeni, doğru ve hızlı tespit edilebilirliği barındırdığı için YOLO Darknet-53 modeli kullanılmıştır. Bu çalışma kapsamında geliştirilen modelde, convolution layer, shortcut layer, contaminated layer, detection layer, batch normalize ve upsample vardır.

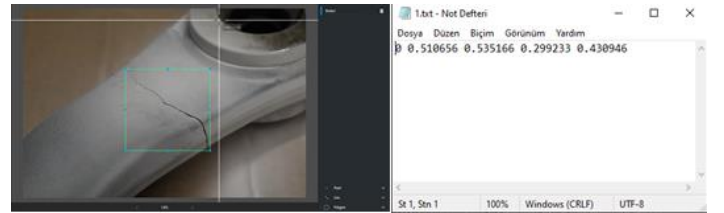
2.1.1. Faaliyetler

Şekil 6'da dinamik test sistemine bağlanmış ve kırılmış olan Salıncak parçası görülmektedir. Dinamik test sistemine bağlanmış olan parçanın deformasyonu, operatör kontrolünde gerçekleştirilmektedir. Bu sebeple gözden kaçırılabilen ve deformasyon artabilmekte, operatör insiyatifinde dinamik test durdurulmaktadır. Dinamik test esnasında salıncak parçasında oluşan çatlak ilk anda tespit etmek gerekmektedir. Deformasyonun tespit edilme süresinin uzaması, dinamik teste tabii tutulan parçanın verilerinde ve yorumlanmasında hatalara neden olabilmektedir. Bu çalışmada Yapay Sinir Ağları kullanılarak deformasyonun tespit işleminin gerçekleştirilmesi planlanılmış, literatür araştırması ve çalışmalar incelenmiş ve irdelenmiştir.



Şekil 6. Süspansiyon parçasında oluşan örnek deformasyon.

Derin öğrenme yöntemlerinin kullanılması için büyük bir veri setine ihtiyaç duyulmaktadır. Önceden test edilmiş olan parçalarda oluşan deformasyonlar, dinamik test sistemine bağlanacak ve yeni test edilecek parçalarda oluşacak bütün deformasyonlar kayıt altına alınmıştır. Kaydedilen video ve resim verileri tamamlandıktan sonra deformasyon bölgesi 'Broken' olarak etiketlenmiştir, yani koordinatları belirlenmiştir. Belirlenen bu koordinatlar, düzenlenecek ve konfigürasyon yapılacak olan YOLO algoritmasına uygun formata dönüştürülmüş (skala edilmiş) ve veri formatı JPG olarak güncellenmiştir. Elde edilen bu verilerin %80'i eğitim için, %20'si ise test için kullanılmak üzere ayrılmıştır. Oluşturmuş olduğumuz veriseti için; data, names ve cfg dosyaları oluşturulmuştur. Data dosyası; kaç sınıfta veri eğiteceğimizi, test ve eğitim verilerinin dosya yollarını, eğiteceğimiz sınıfın etiket adının bulunduğu dosya yolunu ve eğitim sırasında ve sonunda yedekleme dosyasının bulunması istenilen dosya yolu bilgilerini içermektedir. Darknet ana klasörü altında depoladığımız, 'data', 'cfg' ağ dosyası ve 'weights' ağırlık dosyası belirtilerek eğitim işlemi komut ile Google Colab platformu üzerinden başlatılmıştır. Eğitim sırasında elde edilen sonuçlar arasından ortalama hata değeri yani AVG değeri oldukça düşük olduğu zaman eğitim durdurulmalıdır. Ortalama hata değerinin düşmesi uzun zaman almıştır ve AVG değeri 0,08 değerine ulaştığı zaman eğitim işlemi durdurulmuştur. Spyder programı kullanılarak Python programlaması için gerekli olan bilgiler tespit edilmiştir. Bu bilgiler; Deformasyon tespit bölgesinin belirlenmesi ve çizdirilmesi, kaç adet çatlak tespit edildiği, çatlak tespit edildiği an Test Kontrol Sistemi'nin durdurulması için gönderilecek veri, tespit anı görüntüleri olarak OpenCV kütüphanesinde başta olmak üzere birçok kütüphane programlama içerisinde kullanılmıştır. Eğitim işlemi sonucunda elde edilen bu ağırlık dosyamızı yazmış olduğumuz program ile Real-Time olarak çalıştırabilecek, dilediğimiz zaman fotoğraf ve video ile de geçmişe yönelik tespit çalışmaları yapılabilecek, sistem içerisinde geçiş yapabilmeye yönelik modüler bir şekilde hazırlanmıştır.



Şekil 7. Deforme parçanın etiketlenme işlemi ve skala edilmiş koordinatları.

Test ve eğitim veriseti içerisinde bulunmayan başka bir resim ve video yüklenerek, aynı zamanda gerçek zamanlı test sistemi izlenerek en son kaydedilen ağırlık dosyası sisteme yüklenerek test aşamasına geçilmiştir. Deformasyon bölgelerinin tespit edilip çerçeve içerisine alınıp alınmayacağı, tespit halinde deformasyon ve sayısının doğruluğu, Test Kontrol Sistemi'ne durdurma bilgisinin gidip gitmediği 4 farklı versiyon için test edilmiştir.

Tablo 1. YOLO versiyon sonuç karşılaştırmaları

ALGORİTMALAR	mAP	FPS
YOLO-Tiny	%44.2	4 - 4.2
YOLO-v2	%76	0.02 - 0.1
YOLO-v3	%81	1 - 1.2
YOLO-v4	%96.3	0.85 - 1

3. Araştırma Sonuçları ve Tartışma

3.1. Versiyon Denemeleri ve Sonuçlar

Derin Öğrenme algoritmaları sayesinde bu proje ile süspansiyon parçalarından ‘‘Salıncak’’ parçasının dinamik test esnasındaki deformasyonlarının tespit edilmesini ve kontrolünü sağlamak, test edilen ürünlerde deformasyon ilerlemeden test sisteminin durdurulması ve ilk tespit anında görüntü ve verilerin kaydedilmesi hedeflenmiştir. Yapay Sinir Ağları ile nesne tespiti yapabilmek için YOLO algoritması ve DarkNet modeli kullanılmıştır. Veriseti oluşturmak için dinamik test esnasında deformasyon başlamadan kayıt alınmış ve bu kayıtlar işlenerek, önceden test edilmiş parçalardaki çatlakların videoları ve görüntüleride verisetine dahil edilmiştir. Elde edilen verilerden çatlakların belirlendiği noktalar etiketlenmiş ve eğitim işlemi tamamlanmıştır. Görüntü işleme ve deformasyon tespiti için uygun formda Python üzerinden algoritma yazılmış ve kodlama yapılmıştır. Eğitim işlemi bittikten sonra testler yapıлып, çatlak tespitinin farklı YOLO algoritma versiyonları ile kıyaslanarak en doğru sonuçlar alınan algoritma tespit edilmiş, sistem için bu algoritma kullanımı belirlenmiştir. Tiny algoritması daha az katmanı ve ağ yapısı sebebiyle FPS değeri olarak yüksek değerler sağlamıştır fakat bunu yanı sıra mAP değeri olarak diğer algoritmalarından düşük kalmıştır. Versiyon 2 algoritması mAP değeri istenilen değerin üzerinde (%70) sonuç vermiş fakat FPS değeri istenilen aralıkta değer sağlayamamıştır. Versiyon 3 algoritması ile hem mAP değeri hem de FPS değeri olarak istenilen değerler elde edilmiştir. Versiyon 4 algoritması ile de mAP ve FPS değerleri olarak istenilen değerler elde edilmiş olup, en yüksek doğruluk değeri ve mAP/FPS korelasyonu içerisinde bulunması, uygulama için Versiyon-4 algoritmasının kullanımı belirlenmiştir.



Şekil 8. Real-Time Deformasyon Tespit Denemeleri



Şekil 9. Real-Time Deformasyon Tespit Denemeleri

4. Sonuç

Derin öğrenme ve yapay sinir ağları nesne tanıma algoritmaları geliştirilerek süspansiyon parçalarından ‘Salıncak’ parçasının dinamik test esnasında oluşabilecek çatlak veya kırılma deformasyonunun; görüntü, video ve gerçek zamanlı olarak kamera üzerinden tespitinin yapılması başarıyla sağlanmıştır. Yapılan çalışma sayesinde dinamik test esnasında ve test sonrasında istenilen mod dahilinde çatlak tespiti yapılmış olup, parçada oluşan deformasyon ilerlemeden dinamik test durdurulabilecek kabiliyete ulaşıp, oluşan deformasyon personel kontrolünde ve insiyatifinde olmaksızın kesin sonuçları yüzdelik tahmin ve görüntü üzerinden ekrana aktarabilmektedir. Derin öğrenme ağları ile başarılar elde edildiği görülmektedir. Yaygın bir şekilde kullanılmasının sebebi açık kaynak kodlu yazılım kütüphanelerin bulunmasıdır. En büyük dezavantajları modellerinin eğitilmesinde donanımsal kaynakların yetersiz gelebilmesidir.

5. Teşekkür

Bu çalışma AYD0720-03 nolu proje ile desteklenmiştir.

Kaynakça

- Available: [www.arxiv.org/\(2015\). \"abs/1512\"](http://www.arxiv.org/(2015). \) [Accessed:2019].
- Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE international conference on computer vision.
- Girshick, R., J. Donahue, T. Darrell and J. Malik (2014). \"Rich feature hierarchies for accurate object detection and semantic segmentation.\" 2014 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr): 580-587.
- He, K., X. Zhang, S. Ren and J. Sun (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet

- classification. Proceedings of the IEEE international conference on computer vision.
- He, K. M., X. Y. Zhang, S. Q. Ren and J. Sun (2016). "Deep Residual Learning for Image Recognition." 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr): 770-778.
- İNİK ve ÜLKER. /GBAD, 2017, 6(1), 85-104.
- Justin Johnson" from "CS231n Convolutional Neural Networks (2016).
- "Loss Function" from Available: <https://pjreddie.com> (2016), [Accessed:2019].
- Jonathan Hui from SSD object detection: Single Shot MultiBox Detector (2018).
- LeCun, Y. (1987). Modèles connexionistes de l'apprentissage, Université de Paris VI. 18, 504, 517.
- LeCun, Y., Y. Bengio and G. Hinton (2015). "Deep learning." Nature 521(7553): 436-444.
- Lecun, Y., L. Bottou, Y. Bengio and P. Haffner (1998). "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86(11): 2278-2324.
- Szegedy, C., W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich (2015). "Going Deeper with Convolutions." 2015 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr): 1-9
- Sik-Ho Tsang from YOLOv3 You Only Look Once (2019).
- WILDML."UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS FOR NLP."(2016).
- Zeiler, M. D. and R. Fergus (2014). "Visualizing and Understanding Convolutional Networks." Computer Vision - Eccv 2014, Pt I 8689: 818-833.