

YAPAY SİNİR AĞLARI İLE ÜÇ BOYUTLU OBJE SINIFLANDIRMADA KONVOLÜSYON FİLTRELERİNİN ETKİSİNİN İNCELENMESİ

Ahmet OKUDAN¹, Ömer Sinan ŞAHİN²

¹Konya Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Konya Türkiye

²Konya Teknik Üniversitesi, Makine Mühendisliği Bölümü, Konya Türkiye
ahmetokudan27@gmail.com, ossahin@ktun.edu.tr

Özet

Sonlu elemanlar analizi (FEA), yapıların fiziksel davranışlarını tespit etmek için bilgisayar ortamında tasarlanan ve belirli formülasyonlar ile çözülen simülasyon yöntemidir. Problemin büyüklüğüne veya karmaşıklığına göre hazırlık ve çözüm süresi artmaktadır. Sonlu elemanlar analizi için olması gerekenler, geometri ve sınır koşullarıdır. FEA'nın zor ve yorucu olmasının sebebi hazırlık aşamasının çok fazla girdi istemesidir. Ayrıca FEA yapabilmek için ileri düzeyde mühendislik bilgisi de gereklidir. Üç boyutlu yazıcı kullanımının artması da FEA'ya olan ihtiyacı arttırmıştır. Cep telefonu ve tabletlerin işlem gücünün gelişmesi ile sonlu elemanlar analizinin, girdiye ihtiyaç duymadan telefon kamerası ile yapılabilişliğinin mümkün kılınması adına altyapı oluşturulmuştur. Basit analizlerin otomatik olarak yapılabilmesi için derin öğrenme algoritmaları tasarlanmıştır ve eğitilmiştir. Hayatımızın her alanına giren yapay zekânın, sonlu elemanlar analizi alanında ne gibi çözümler sunacağı incelenmiştir. Ağ örgüsü (mesh) olmadan çözüm yapma hacimsel piksellendirme metoduna dayandığı için bu çalışmada vokselizasyon tabanlı öğrenme üzerine inceleme yapılmıştır. Bu çalışmada, sonlu elemanlar analizi girdisi için gerekli olan üç boyutlu geometrik verilerin, bir kamera vasıtası ile taranan objelerin sınıflandırılmasına yarayacak yapay sinir ağı (YSA) oluşturulmuştur ve doruluğu test edilmiştir. Üç boyutlu geometrilerin sınıflandırılabilmesi için özelleştirilmiş bir evrişimli (konvolüsyonel) sinir ağı tasarlanmıştır. Çalışmada, üç boyutlu bilgisayar geometrilerinin konvolüsyon filtresine sahip yapay sinir ağları ile konvolüsyon fitresi bulundurmeyen yapay sinir ağları arasındaki parametre sayısı, öğrenme süresi, öğrenme doğruluğu ve maliyet değerleri karşılaştırılmıştır. Konvolüsyon filtreli YSA'nın validasyon değeri %97,92'dir. Konvolüsyon filtresiz YSA %97,08 validasyon değeri ile öğrenme işlemini

tamamlamıştır. Konvolüsyon filtresiz YSA eğitimi 440 saniyede tamamlarken, Konvolüsyon filtrelili YSA'nın eğitimi 6260 saniye sürmüştür.

Anahtar Kelimeler: Derin öğrenme, Yapay sinir ağı, Üç boyutlu obje sınıflandırma, Evrişimli sinir ağı, Konvolüsyon filtresi, Sonlu elemanlar analizi

INVESTIGATION OF THE EFFECT OF CONVOLUTIONAL FILTERS ON THE THREE DIMENSIONAL OBJECT CLASSIFICATION BY ARTIFICIAL NEURAL NETWORKS

Abstract

Finite elements analysis (FEA) is a simulation method designed in a computer environment to determine the physical behavior of structures and solved with specific formulations. Depending on the size or complexity of the problem, preparation and solution time increases. The requirements for finite element analysis are geometry and boundary conditions. The FEA is difficult and tiring because the preparatory phase requires too much input. In addition, advanced engineering knowledge is required to perform FEA. The increase in the use of three-dimensional printers has increased the need for FEA. With the development of the processing power of mobile phones and tablets, an infrastructure was established to enable the feasibility of finite element analysis with telephone camera without the need for input. Deep learning algorithms are designed and trained in order to perform simple analysis automatically. What kind of solutions will be presented in the field of finite element analysis of artificial intelligence, which is included in every field of our lives, has been examined. Since the solution without mesh is based on the volumetric pixelation method, voxelization-based learning was investigated in this study. In this study, an artificial neural networks (ANN) was created and used for the classification of three-dimensional geometric data required for finite element analysis input and scanned objects by a camera. In order to classify three-dimensional geometries, a customized convolutional neural network was designed. In this study, the number of parameters, learning time, learning accuracy and cost values were compared between artificial neural networks with convolution filter of three dimensional computer geometries and artificial neural networks without convolution filter. The validation value of ANN with convolution filter is 97.92%.

Without convolution filter ANN has completed the learning process with a validation value of 97.08%. While the ANN without convolution filter completed the training in 440 seconds, the ANN with convolution filter took 6260 seconds.

Keywords: Deep learning, Artificial neural network, Three-dimensional object classification, Convolutional neural network, Convolution filter, Finite elements analysis

1. Giriş

Yapay zekâyı oluşturan temel algoritmalarından biri olan öğrenme iki farklı şekilde gerçekleşir. Belirli bir çıktının veya verinin sisteme öğretildiği durum gözetimli öğrenme (supervised learning), algoritmanın çıktıya kendisinin kadar vererek sınıflandırma yapması istenen duruma ise gözetimsiz öğrenme (unsupervised learning) adı verilir [1]. Bu tez çalışmasında statik ve modal analiz sonuçları sinir ağına öğretileceği için gözetimli öğrenme ile algoritmalar optimize edilmiştir.

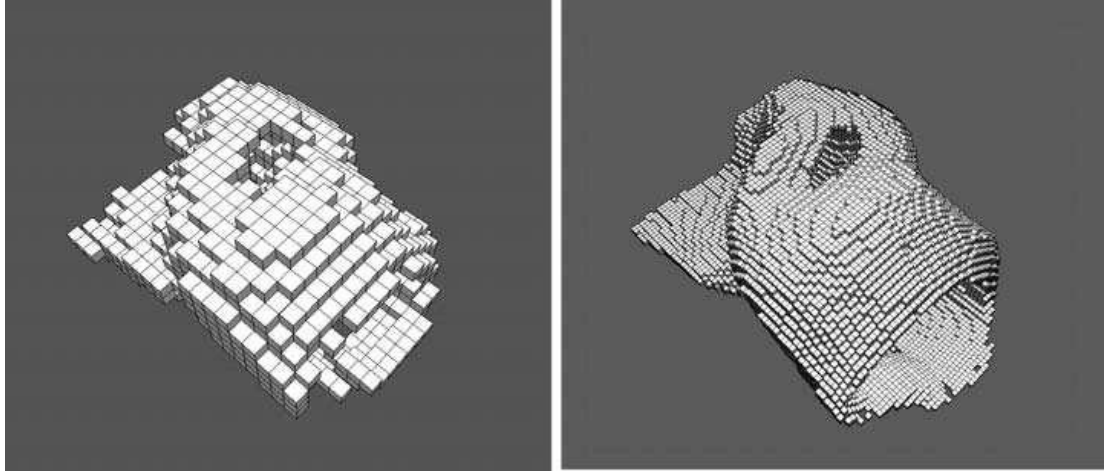
Öğrenme işlemi, yapay sinir ağlarında bir algılayıcı (perceptron) ile gerçekleştirilir. Algılayıcı girdileri toplayarak bir aktivasyon fonksiyonundan geçirir ve bir çıktı üretir. Bu çıktının istenen çıktıya yakınlık derecesi, öğrenme işleminin doğruluğunu gösterir. Derin öğrenme kavramı, algılayıcıların önüne gizlenen fazladan algılayıcılar ile, çok katmanlı algılayıcı yapısının bulunuşu ile ortaya çıkmıştır. Tez çalışmasında çok katmanlı algılayıcı modelleri kullanılmıştır.

İnsan beyin hücrelerinde nöronların çalışma mantığı baz alınarak tasarlanan algılayıcılar çok katmanlı olarak inşa edildiğinde, İleri beslemeli sinir ağı ya da tam bağlı sinir ağı olarak da söylenen gizli katmanlı algılayıcı ağları, derin öğrenme ve yapay zekâ uygulamalarında çokça kullanılan yapılardır [2]. Tek bir algılayıcı (perceptron) doğrusal bir fonksiyon olması sebebiyle sadece doğrusal sınıflandırma problemlerinde başarılı sonuç vermektedir. Ancak XOR kapısı modeli gibi doğrusal olarak sınıflandırılmayan modellerde çok katmanlı yapay sinir ağları tercih edilmektedir.

Konvolüsyonel olarak da anılan evrişimli sinir ağları, çok katmanlı yapay sinir ağlarının özelleştirilmiş bir versiyonudur [3]. Veri, sinir ağlarından geçirilmeden önce, konvolüsyon filtreleri tarafından bazı ön matematiksel fonksiyonlardan geçirilir. Böylece görüntü, ses veya üç boyutlu verilerin baskın özellikleri daha iyi saptanabilir.

Bu çalışmada, üç boyutlu görüntü işleme ile katı cisimlerin sınıflandırılması yapılmıştır. Evrişimsel sinir ağlarının sınıflandırma doğruluğuna ve sınıflandırma eğitim süresine etkisi incelenmiştir.

Hacimsel pikselleme (Voxelization), temeli ikili piksellemeye (binary voxelization) dayanan, bilgisayar ortamındaki eni, genişliği ve derinliği olan üç boyutlu geometrilerin birim küplere bölünmesi ve bir uzay sınırında, geometriyi kapsayan birimlere 1, boşlukta kalan birimlere 0, sayılarının atanması ile gerçekleştirilen işlemdir. Hacimsel pikselleme işleminden sonra geometri, yüzey eğriliğini kaybedeceği için, piksellerin birim kenar uzunluğu, geometrinin ne kadar gerçeğe yakın olacağını etkiler. Fakat birim boyutu küçüldükçe bilgisayar ortamında piksellenen geometrinin saklandığı dosyanın boyutu ve pikselleme işleminin işlem süresi de artar. Şekil 1’de görüldüğü üzere, pikselleme işlemi daha küçük ölçüdeki birim küpler ile seçilirse, geometrinin doğruluk oranı artar ancak dönüştürme süresi de bu oranda artar. Bu yüzden çalışmada, yapay sinir ağlarının, bir geometriyi tanımlarken mümkün olan en optimum piksel genişliği saptanmıştır.



Şekil 1. Hacimsel pikselleme işleminde, piksel sayısı arttıkça geometrinin doğruluk oranının artması [4].

Üç boyutlu nesne sınıflandırma ve tanımlama teknolojisi, üretim hatlarının kalite kontrol bölümlerinde ve otonom sürüş sistemlerinde aracın etraftaki üç boyutlu uzayı tanıması amacı için kullanılmaktadır. LIDAR (Light Detection and Ranging – Değişken Genlikli Işık Algılama) veya derinlik kameraları gibi teknolojileri kullanan nesne tanıma sistemlerinde üç boyutlu uzay, bir koordinat eksenine göre piksellenerek bilgisayar yazılımı tarafından görülebilir hale getirilir [5]. Piksellenmiş veri daha önce

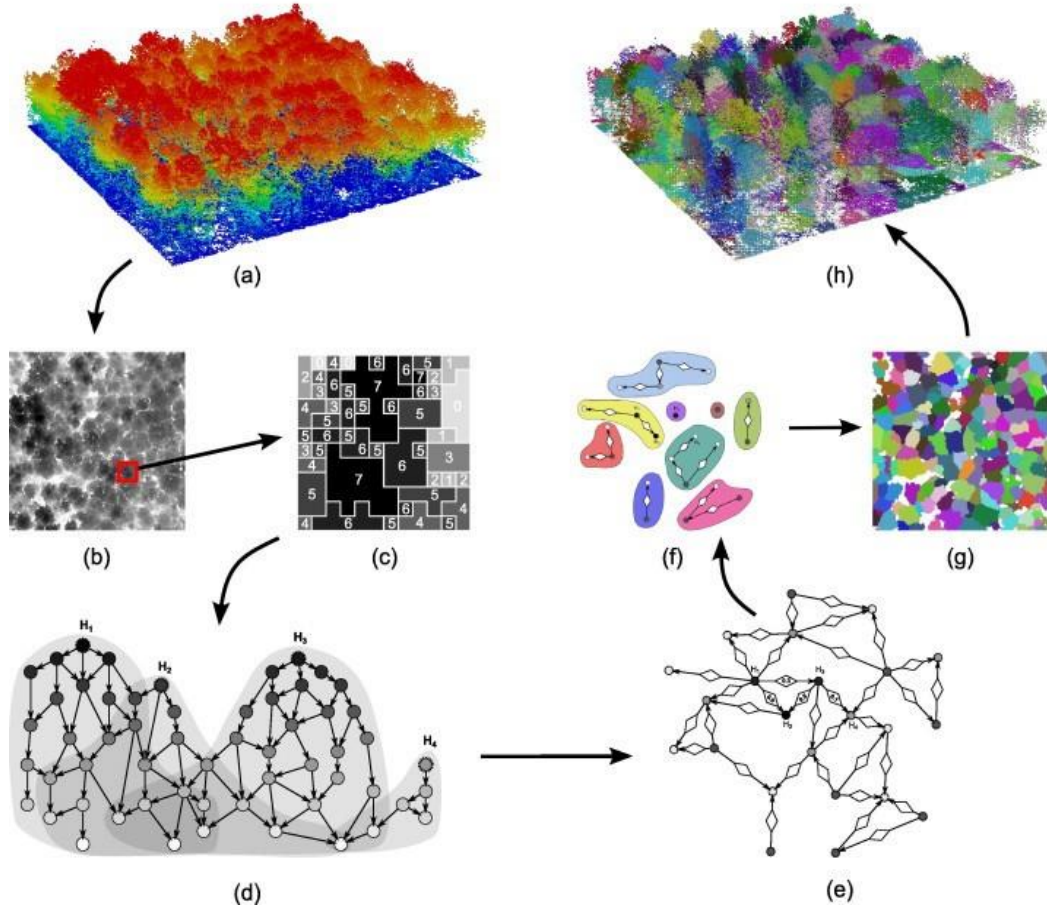
eğitilmiş yapay sinir ağlarından geçirilerek objelerin sınıflandırılması ve ayrıştırılması işlemlerini gerçekleştirerek objeyi tanımlar. Üç boyutlu sınıflandırma, görüntü işleme yönteminde derinlik algısının olmamasından dolayı tercih edilebilmektedir. Birbirine çok yakın renklere cisimler, derinlik bilgisi sayesinde görüntü işleme yöntemine göre daha doğru şekilde tanımlanmaktadır.

Çalışmada, FEA için gerekli koşullardan olan geometri gereksinimi bir üç boyutlu kameraya gösterilecek cismin hangi sınıfa ait olduğunu anlayabilen bir derin öğrenme algoritması eğitilmiş ve konvolüsyon filtreleri farkları saptanmıştır. Bu çalışmanın FEA'yı otomatik olarak yapabilen yapay zeka uygulamalarına alt yapı oluşturması amaçlanmıştır.

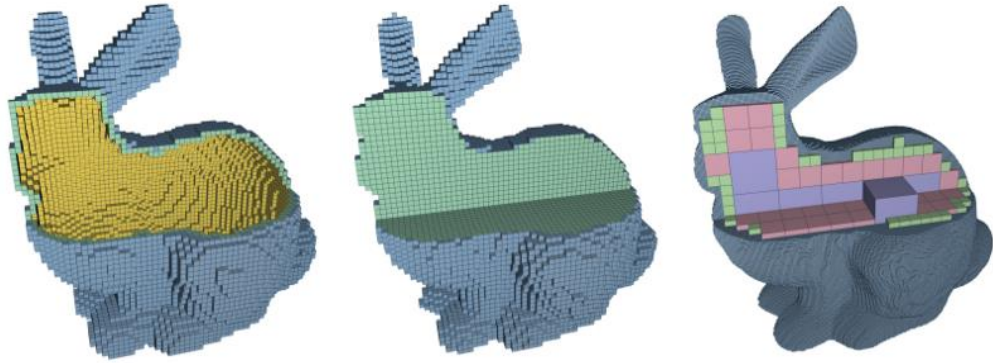
2. Materyal ve Metot

Bu çalışmada üç boyutlu cisim sınıflandırma belirli dönüşümler ile gerçekleştirilecektir. Görüntü işleme yöntemlerinde resimler OpenCV kütüphaneleri ile matrise dönüştürülebilmektedir. Ancak bilgisayar ortamındaki üç boyutlu geometrilerin matrislere dönüştürülebilmesi için önce hacimsel olarak piksellenebilmesi gerekmektedir. Şekil 1'de gösterilen çalışmada bir arazideki bitki örtüsü LIDAR ile taranmış, hacimsel pikselleri elde edilmiş ve piksel verisi hiyerarşik derin öğrenme algoritmaları ile segmente edilmiştir.

Hacimsel pikselleme katı geometriyi veya yüzeyi, uzayda işgal ettiği alana göre bölümleyerek temsil etmektedir. Voxelization; nesnelerin voxel (volümetrik piksel) kümeleri olarak sunulmasını sağlayan bir süreçtir. Tam olarak geometrik nesnelerin nesnenin doğasına/yapısına en yakınsayacak şekilde, sürekli geometrik temsillerden bir voxel kümesi haline dönüştürülmesi safhası olarak tanımlanmıştır. Bilgisayar programlarında katı geometriler çoğunlukla, parasolid, step, wrml, stl gibi dosya formatları ile depolanır. Her dosya formatının kendine özgü bir depolama biçimi vardır ve bu biçim yapay sinir ağları için direkt olarak veri beslemeyi engeller. Bu sebeple çalışmada ücretsiz ve açık kaynak kodlu olarak sağlanan binvox.exe yazılımından destek alınmıştır.



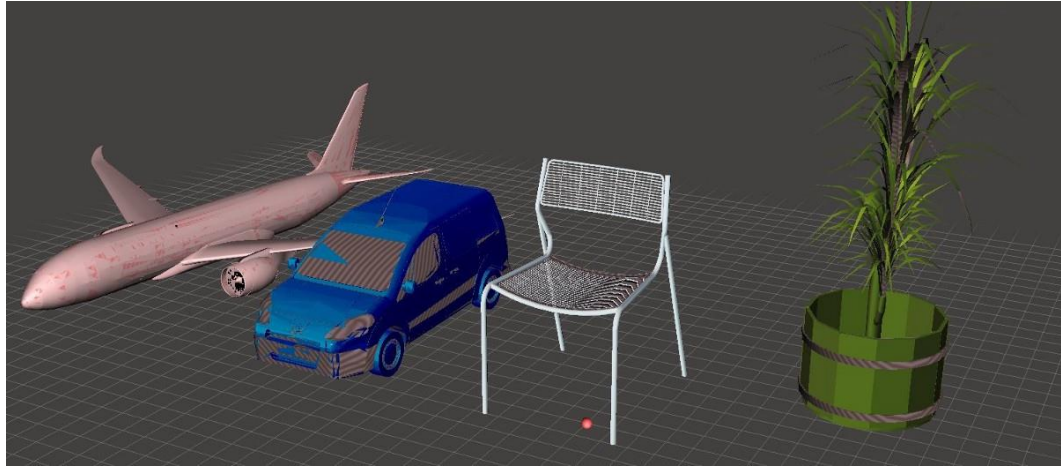
Şekil 1. LIDAR verisinin segmente (bölümleme) edilmesi [6]



Şekil 2. Bir katı geometrinin farklı yöntemler ile hacimsel piksellenmesi

Üç boyutlu obje sınıflandırma ve obje segmentasyonu ile ilgili uzun süredir çalışma yapılmaktadır. Princeton üniversitesi, derin öğrenme alanında çalışanlar için ücretsiz bir veri kütüphanesi hazırlamıştır. Bu çalışmada ModelNET10 kütüphanesinden 4 adet obje seti seçilerek ESA eğitilecektir. ModelNET kütüphanesi verileri *.off formatındaki dosyalar ile sağlamaktadır.

OFF dosya formatı, üç boyutlu geometrileri, yüzey renkleri ve tam uygunluk yüzeyi ile saklayabilen dosya formatlarıdır. Parasolid, STEP ve diğer CAD yazılımlarının üç boyutlu obje formatından farklı olarak yüzeyler üçgenlerin birleşimi ile elde edilir. Bu sebeple çözünürlük, STL, OBJ, POLY ve OFF dosyalarında geometriye yakınlık derecesini temsil eder. ModelNET kütüphanesinden elde edilen geometriler OFF formatı ile sunulmaktadır. OFF dosyaları, görüntü işlemede JPEG dosyalarının direkt olarak okunması gibi okunamayacağı için binvox (binary voxel) yani ikili volümetrik piksel dosyalarına dönüştürülecektir. Çalışmada, ModelNET kütüphanesinden indirilen verilerin görsel olarak doğrulanması ücretsiz olarak sunulan Autodesk Meshmixer programında yapılmıştır. Şekil 3’de gösterildiği üzere, her sınıftan bir cisim rastgele seçilerek görselleştirilmiştir.



Şekil 3. ModelNET kütüphanesinden temin edilen verilerin görselleştirilmesi

Üç boyutlu parçaların eğitiminde 4 farklı sınıf için 300’er adet dosyadan dönüştürülen veriler kullanılacaktır. Tablo 1’de kullanılacak sınıflar ve sınıf numaraları belirtilmiştir. Üç boyutlu veri seti sınıfları ikili dönüştürme yöntemi ile sınıflandırılmıştır.

Tablo 1 Yapay sinir ağı eğitiminde kullanılacak dosya çeşitleri ve sınıfları

Veri Tipi	Obje Sınıfı	Sınıf No	Veri Adedi	Test Oranı
OFF Dosyası	Uçak Çeşitleri	1	300	%20
OFF Dosyası	Otomobil çeşitleri	2	300	%20
OFF Dosyası	Sandalya çeşitleri	3	300	%20
OFF Dosyası	Saksılı ev bitkisi çeşitleri	4	300	%20

Eğitim dosyaları, sadece dolu cisimlerden oluşmaması ve yapay sinir ağlarını dolu cisimleri öğrenmeye baskılamaması için çiçek ve bitki gibi dağılım matrisi düzensiz olacak geometrilerden de seçilmiştir. Bu sayede, matris içerisinde değerlerin dağınık halde bulunsa bile, derin öğrenme algoritmasının yorumlama ve bütünleme tahmini yeteneği kazanmıştır.

OFF dosyaları yapay sinir ağları ile eğitilebilmesi için hacimsel piksel dosyalarına çevrilecektir. Uzantısı binvox olarak isimlendirilen ikili sınıfa sahip volümetrik piksel dosyaları ücretsiz olarak sunulan “binvox.exe” programı ile Windows komut satırı üzerinden çalıştırılarak dönüştürülecektir. Python programlama dilinde yazılan algoritma ile OFF dosyaları sıra ile okunarak binvox.exe programında binvox dosya formatına çevrilecek ve bilgisayara kayıt edilmiştir. OFF dosyaları, geometri özelliklerinin belirgin olması için 128 x 128 x 128 birimlik hacimsel piksellere çevrilmiştir.

Çalışmanın yapıldığı bilgisayardaki ekran kartı donanımı, 4GB belleğe sahip Nvidia GTX 1050i modeldir ve 128 birim küplük dosyalar 4 ve 10 saniye aralığında oluşturulmuştur. Dönüştürme işlemi toplamda 102 dakika sürmüştür.

Binvox dosyaları, 3 boyutlu veri setindeki 1200 ayrı obje için tek tek oluşturulmuştur. Programında doğru çalışırılığının kontrolü için binvox dosyalarından rastgele seçilen otuz adet dosya viewvox.exe yazılımı ile görselleştirilmiştir. Şekil 4’de dosyaların görselleştirilmiş halleri gösterilmektedir.

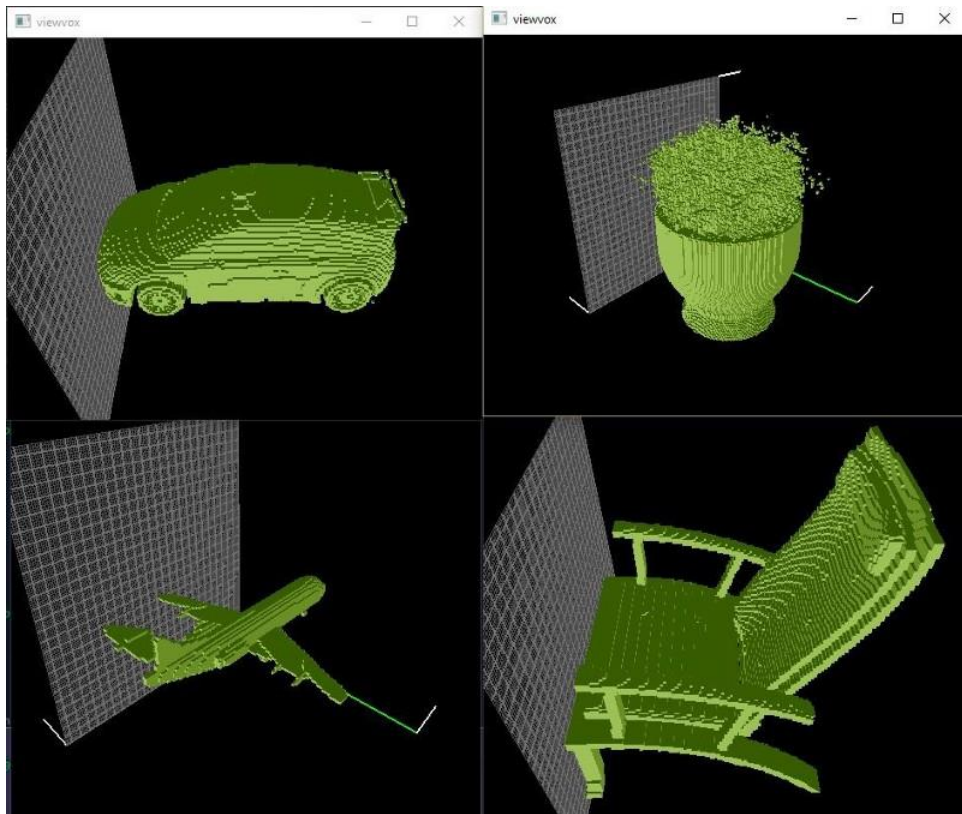
Farklı kenar uzunluğu birimlerinin dosya pikselleme ve pikselleme sürelerinin değişimleri arasındaki farklar bulunmuştur. Farklı derecelerin toplam piksel adedine etkisi, dönüştürülen dosya boyutuna etkisi, piksellenem süreleri ve piksellenen katıların orijinal geometri ile uyuma oranı Tablo 2’de verilmektedir.

Tablo 2. Farklı kübik değerlere ait dosya özelliklerinin karşılaştırılması

Pikselleme derecesi	Toplam piksel adedi	Piksel dosyası boyutu (kb)	Pikselleme süresi (sn)	Katı uyuma oranı (%)
8	512	1	1	54
32	32768	5	1	79
128	2097152	62	3	92
256	16777216	258	6	94
512	134217728	1506	12	97
1024	1073741824	10129	60	98

Çalışmada veri setinin 128 birim küp ile piksellenmesine karar verilmiştir. Dosya boyutu, toplam piksel adedi ile doğru orantılı artacağı için, yapay sinir ağı eğitimi sırasında bellek hatası alınmaması için 128 birimlik pikselleme de karar kılınmıştır. Toplamda 1200 adet off dosyası sıralı bir işlem algoritması ile binvox dosyalarına çevrilmiş ve Şekil 4’de gösterildiği üzere dosyalar görselleştirilerek kontrol edilmiştir.

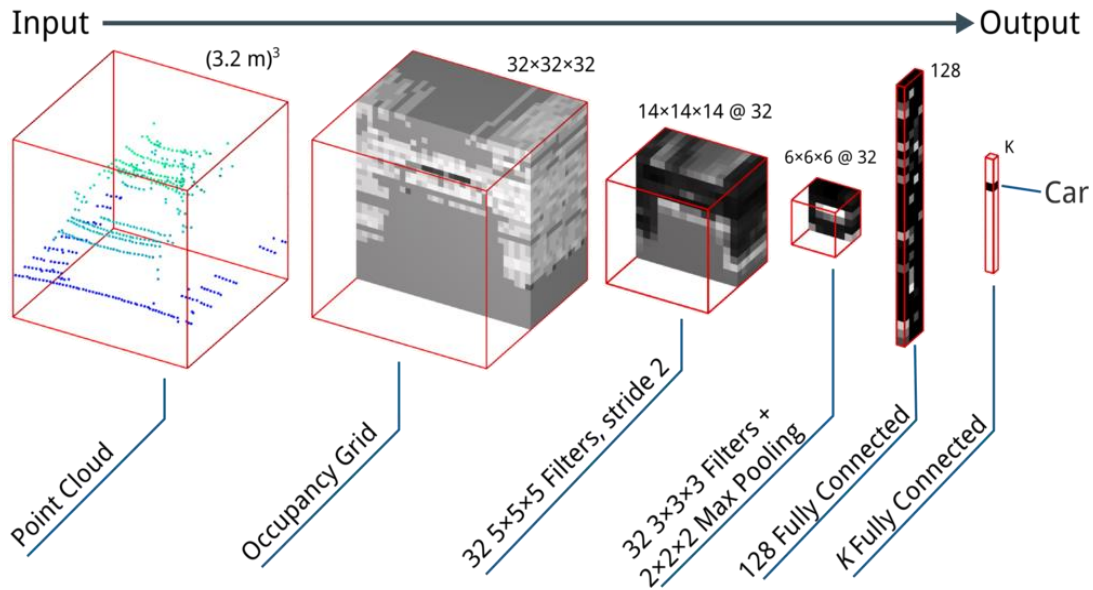
Binvox dosyalarında veri mantıksal (boolean) sınıflandırma ile saklanmaktadır. Dolayısı ile her bir pikselde Doğru (True) veya Yanlış (False) değerleri bulunur. Binvox dosyaları python programlama dilinde Numpy cebirsel kütüphaneden yardım alınarak ikili sınıflandırma matrislerine dönüştürülmüştür. Dönüştürülen matrisler eğitim tensörü içinde toplanmıştır. Ek’de kodları verilen algorithmada, ilk olarak 1200 x 128 x 128 x 128 boyutunda bir tensör oluşturulmuş ve 1200 adet binvox dosyası sıra ile eğitim veri çerçevesine yazdırılmıştır. Değeri boolean olan binvox dosyaları alım esnasında “float32” tipine çevrilmiştir.



Şekil 4. Binvox dosyasına çevrilen OFF dosyalarının görsel kontrolü

Binvox dosyalarının matrislere yazılması ile yapay sinir ağlarının eğitilebileceği veri çerçevesi oluşmuştur. Bu aşamada Google firmasının ücretsiz olarak sunduğu

Tensorflow derin öğrenme kütüphanesinin üzerine geliştirilen Keras Kütüphanesi ile python programlama dilinde yazılacak algoritma ile derin öğrenme modeli oluşturulmuştur. YSA mimarisi, VoxNET mimarisine yakın bir mimari ile oluşturulmuştur. VoxNET mimarisi Şekil 5’de gösterilmiştir. Oluşturulan modelin VoxNET modelinden farklı üç boyutlu girdiyi 128 x 128 x 128 boyutunda almaktadır. Bu hali ile daha büyük bir alanı daha küçük kareler ile tanımlamaktadır. Bu sayede öğrenme işleminde doğruluk oranının artırılacağı düşünülmüştür.



Şekil 5. VoxNET mimarisinin şematik gösterimi, VoxNET mimarisi voxel veriyi 32 x 32 x 32 boyutlarında yapay sinir ağına beslemektedir [7]

Şekil6’da konvolüsyon filtreli olarak inşa edilen YSA mimarisi gösterilmektedir. 128 x 128 x 128 boyutunda alınan üç boyutlu tensör, atlama adedi 6 olan 2 adet konvolüsyon filtresinden geçerek 123 x 123 x 123 boyutlarındaki tensöre dönüşmektedir. Model, veriyi iki adet sıralı bağlı maksimum havuzlama ve bir adet ortalama havuzlama işleminden sonra 51 x 1 boyutlarındaki vektöre dönüştürmektedir. İlk katmanda 32 adet nöron, ikinci gizli katmanda 16 adet nöron ve çıkış katmanında sınıf sayısı olan 4 adet nöron bulunmaktadır. Bu mimarinin eğitilebilir parametre sayısı 2790’dır. Eğitilebilir parametre sayısı, yapay sinir ağında bulunan, türevi alınabilen ve değeri her döngüde (iterasyonda) güncellenebilen ağırlık değerlerinin toplam sayısıdır. Bir yapay sinir ağında giriş vektörü ne kadar büyük olursa eğitilebilir parametre sayısı da fazla olacaktır. Diğer yandan gizli katmanlardaki nöron adedi ve gizli katman sayısı da eğitilebilir parametre sayısı ile ilişkilidir. Şekil 6.’da gösterilen maksimum ortaklama

(max pooling) ve ortalama ortaklama (average pooling) işlemleri eğitilemez parametreler olduğu için varlıkları eğitilebilir parametre sayısını etkilememektedir.

3 boyutlu obje sınıflandırmanın endüstriyel üretim sistemlerinde kullanımı ile alakalı yapılan çalışmada, yerel tarama sistemlerinde parametre sayısının optimize edilmesi sinir ağının sistem bellek doldurma süresine etkisi incelenmiştir. Parametre sayısının fazlalığı sinir ağının sınıflandırma esnasında sistemi yormasına sebep vermektedir. Çalışmada parametre sayısı mümkün olan en optimum sayıda seçilmiştir.

[8]

Layer (type)	Output Shape	Param #
conv3d_5 (Conv3D)	(None, 123, 123, 123, 2)	434
activation_48 (Activation)	(None, 123, 123, 123, 2)	0
max_pooling3d_24 (MaxPooling)	(None, 20, 20, 20, 2)	0
max_pooling3d_25 (MaxPooling)	(None, 6, 6, 6, 2)	0
activation_49 (Activation)	(None, 6, 6, 6, 2)	0
average_pooling3d_12 (AveragePooling3D)	(None, 3, 3, 3, 2)	0
activation_50 (Activation)	(None, 3, 3, 3, 2)	0
flatten_13 (Flatten)	(None, 54)	0
dense_28 (Dense)	(None, 32)	1760
activation_51 (Activation)	(None, 32)	0
dropout_20 (Dropout)	(None, 32)	0
dense_29 (Dense)	(None, 16)	528
activation_52 (Activation)	(None, 16)	0
dropout_21 (Dropout)	(None, 16)	0
dense_30 (Dense)	(None, 4)	68
activation_53 (Activation)	(None, 4)	0
=====		
Total params: 2,790		
Trainable params: 2,790		
Non-trainable params: 0		

Şekil 6. Keras kütüphanesi ile oluşturulan ve özet görünümü verilen konvolüsyon filtreli yapay sinir ağı mimarisi özeti

Şekil 7’de özet görünümü gösterildiği üzere, konvolüsyon filtresi bulunmayan YSA, 128 x 128 x 128 boyutundaki tensörü, konvolüsyonlu modeldeki aynı havuzlama filtrelerinden geçirmekte ve 343 x 1 boyutundaki vektör elde edilmektedir. Konvolüsyon filtreleri veriyi küçülttüğü için bu modelde giriş vektörü daha büyüktür. Konvolüsyon filtresi olmayan YSA’da gizli katmanlar aynı şekilde inşa edilmiş ve eğitilebilir parametre sayısı 11,604 olarak gözlemlenmiştir.

Layer (type)	Output Shape	Param #
max_pooling3d_16 (MaxPooling)	(None, 42, 42, 42, 1)	0
activation_24 (Activation)	(None, 42, 42, 42, 1)	0
max_pooling3d_17 (MaxPooling)	(None, 14, 14, 14, 1)	0
activation_25 (Activation)	(None, 14, 14, 14, 1)	0
average_pooling3d_8 (Average)	(None, 7, 7, 7, 1)	0
activation_26 (Activation)	(None, 7, 7, 7, 1)	0
flatten_9 (Flatten)	(None, 343)	0
dense_16 (Dense)	(None, 32)	11008
activation_27 (Activation)	(None, 32)	0
dropout_12 (Dropout)	(None, 32)	0
dense_17 (Dense)	(None, 16)	528
activation_28 (Activation)	(None, 16)	0
dropout_13 (Dropout)	(None, 16)	0
dense_18 (Dense)	(None, 4)	68
activation_29 (Activation)	(None, 4)	0
Total params: 11,604		
Trainable params: 11,604		
Non-trainable params: 0		

Şekil 7. Keras kütüphanesi ile oluşturulan ve özet görünümü verilen Konvolüsyon filtresi bulundurmeyen yapay sinir ağı mimarisi özeti

3. Sonuçlar ve Tartışma

Çalışmada, ModelNET kütüphanesinden elde edilen OFF formatındaki 4 sınıfa ait 1200 adet dosya önce hacimsel piksellenme ile piksel veriye dönüştürülmüş ve ardından yapay sinir ağları girdisi için gerekli matris formuna dönüştürülmüştür. Eğitimler sonucunda konvolüsyon filtresi bulunduran ve bulundurmeyen YSA mimarilerinin, eğitilebilir parametre sayıları, eğitim maliyet değerleri, eğitim doğruluk değerleri, validasyon maliyet değerleri, validasyon doğruluk değerleri ve eğitim süreleri karşılaştırılmıştır. Tablo 3’de karşılaştırma değerleri verilmiştir. Eğitim, değerlerin sağlıklı karşılaştırılabilmesi adına, 100 iterasyon (epoch) ile eğitilmiştir.

Tablo 3. Üç boyutlu obje sınıflandırma için konvolüsyonlu ve konvolüsyonsuz

YSA’ların eğitim değerleri karşılaştırılması

Tip	Eğitilebilir Parameter Sayısı	Eğitim Maliyet Değeri	Eğitim Doğruluk Değeri	Validasyon Maliyet Değeri	Validasyon Doğruluk Değeri	Eğitim Süresi (sn)
Konvolüsyonlu	2790	0.007	0.9812	0.039	0.9792	6260
Konvolüsyonsuz	11604	0.056	0.9667	0.078	0.9708	440

Tablo 3’de eğitim ve validasyon maliyet değerleri (cost function) olarak adlandırılan değerler, her döngüdeki (iterasyondaki) yapay sinir ağı çıktısının, gerçek değerler ile karşılaştırılarak, lineer regresyon analizinde doğruya uzaklığın ortalamasıdır. Burada verilen değerler son döngünün değerleridir. Eğitim ve validasyon doğruluk değerleri olarak adlandırılan değerler, 1200 adet veriden kaçının doğru tahmin edildiğinin yüzde olarak hesaplanmış değerleridir.

Konvolüsyon filtresi bulunduran mimaride, eğitilebilir parametre sayısının az olmasına rağmen, filtreleme işlemi zaman aldığı için eğitim süresi konvolüsyonsuz mimarinin eğitim süresine göre oldukça fazladır. Fakat konvolüsyon filtresi işlemi 128 x 128 x 128 piksel büyüklüğündeki obje matrisi oldukça küçüldüğü için eğitilebilir parametre sayısı da azalmıştır. Bu yüzden konvolüsyonsuz mimari daha fazla parametre eğiterek eğitim ve validasyon değerlerinde, konvolüsyonlu mimariye yakın değerler vermiştir.

Görüntü işleme yöntemlerinde konvolüsyonlu mimariler oldukça fazla kullanılmaktadır. Ancak iki boyutlu resimlerde kısa zaman alan konvolüsyon filtresi işlemi, üç boyutlu modellerde oldukça zaman almaktadır. Piksel genişliği düşük modeller için konvolüsyon filtresi seçimi eğitim zamanını çok fazla yükseltmeyecektir.

Bu sonuçlar ile piksel genişliği büyük modellerde, konvolüsyon filtresi uygulanmadan da doğru sonuçlar elde edilebileceği görülmüştür.

Teşekkür

Merve Ayyüce KIZRAK ve Arda MAVİ'ye yapay sinir ağları konusunda sağladıkları makale, yayın, destek ve tavsiyeler için çok teşekkür ederim.

Kaynaklar

- [1] Hinton, G., "Deep learning," Nature, 2013, 521, p. 436–444.
- [2] Hinton, G., A Fast Learning Algorithm For Deep Belief Nets, Neural Computation, 18, 2013, p. 1527–1554.
- [3] Goodfellow, I., Bengio, Y., and Courville, A., 2016, Deep Learning [online]. MIT Press, <http://www.deeplearningbook.org>. [Ziyaret Tarihi: 15 Ekim 2018]
- [4] Bier, H., Heule, M., Prototypes for automated architectural 3d-layout, 13th International Conference Virtual Systems and Multimedia, VSMM 2007, Australia, 2007, p. 203-214.
- [5] Mutlu, H., E., Classification of Hyperspectral and LiDAR Data with Deep Learning, Master's Thesis, Hacettepe University, Ankara, Turkey, 2018
- [6] Strimbu, V., F., Strimbu, B., M., A Graph-Based Segmentation Algorithm For Tree Crown Extraction Using Airborne Lidar Data, ISPRS Journal of Photogrammetry and Remote Sensing, 2015, p. 24-25.
- [7] Maturana, D., and Scherer, S., VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition, International Conference on Intelligent Robots and Systems, 2015, p. 922-928.
- [8] Alcaer, V., Cruz, M., Scanning the Industry 4.0: A literature Review on Technologies for Manufacturing Systems, Engineering Science and The Technology, 2019, p. 889-919

Ek – OFF dosyasını Binvox dosyasına çevirme ve binvox dosyalarını matris içine alma python kodları bütünü:

```
import os
os.chdir(r'D:\deep_learning\voxnet\voxgen1')
print(os.getcwd())
import time
```

```
k = 1
while k < 1201:
    print(k)
    command = r"binvox -c -d 128 -t binvox -cb modeldata3\x(" + str(k) + ").off"
    print(command)
    os.popen(command)
    time.sleep(5) #10 second waiting for process time
    k = k + 1
import numpy as np
import binvox_rw
a = np.arange(0,1200)
x_class = np.zeros(shape=(1200,128,128,128), dtype=float)
liste = list()
for i in a:
    with open('modeldata3/x(' + str(i) + ').binvox', 'rb') as f:
        liste.append("nplist/x(" + str(i) + ").npy")
        model = binvox_rw.read_as_3d_array(f)
        narray = np.ndarray(shape=(1,128,128,128), dtype=float)
        narray = model.data
        narray = narray.astype(float)
        x_class[i,:] = narray
    print(i)
np.save("x_class.npy",x_class)
import os
command2 = r"viewvox modeldata2\x(29).binvox"
os.popen(command2)
import pandas as pd
y_label = pd.read_csv("modeldata2/dataset2.csv",sep = ";")
y_class = y_label.classif.values
y_class = np.array(y_class, dtype=float)
y_class = y_class.reshape(30,1)
y = y_class.T
y_save = np.save("y_class",y)
```