



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**HACİMSSEL PİKSELLEME YÖNTEMİ İLE ÜÇ
BOYUTLU NESNE TANIMLAMA VE
GERÇEK ZAMANLI SONLU ELEMANLAR
ANALİZİ YAPABİLEN DERİN ÖĞRENME
ALGORİTMASI GELİŞTİRİLMESİ**
Ahmet OKUDAN

YÜKSEK LİSANS TEZİ

Makine Mühendisliği Anabilim Dalı

Eylül-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ahmet OKUDAN tarafından hazırlanan “HACİMSEL PİKSELLEME YÖNTEMİ İLE ÜÇ BOYUTLU NESNE TANIMLAMA VE GERÇEK ZAMANLI SONLU ELEMANLAR ANALİZİ YAPABİLEN DERİN ÖĞRENME ALGORİTMASI GELİŞTİRİLMESİ” adlı tez çalışması 06/09/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Makine Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

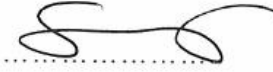
Başkan

Doç. Dr. Necati ATABERK



Üye / Danışman

Prof. Dr. Ömer Sinan ŞAHİN



Üye

Doç. Dr. Mehmet BAĞCI



Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü



TEZ BİLDİRİMİ

Bu tezdaki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Ahmet OKUDAN

Tarih: 06.09.2019

ÖZET

YÜKSEK LİSANS TEZİ

Hacimsel Pikselleme Yöntemi ile Üç Boyutlu Nesne Tanımlama ve Gerçek Zamanlı Sonlu Elemanlar Analizi Yapabilen Derin Öğrenme Algoritması Geliştirilmesi

Ahmet OKUDAN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Makine Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ömer Sinan ŞAHİN

2019, 110 Sayfa

Jüri
Prof. Dr. Ömer Sinan ŞAHİN
Doç. Dr. Necati ATABERK
Doç. Dr. Mehmet BAĞCI

Sonlu elemanlar analizi, 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. SEA'nın zor ve yorucu olmasının sebebi hazırlık aşamasının çok fazla girdi istemesidir. Ayrıca SEA yapabilmek için ileri düzeyde mühendislik bilgisi de gereklidir. Üç boyutlu yazıcı kullanımının artması da SEA'ya olan ihtiyacı arttırmıştır. Cep telefonu ve tabletlerin işlem gücünün gelişmesi ile sonlu elemanlar analizinin, girdiye ihtiyaç duymadan telefon kamerası ile yapılabilirliğinin 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 ve gerekli yöntemler için farklı algoritmalar yazılmıştır.

Bu çalışmada, görüntü işleme yöntemi ile, analiz edilecek parçayı tanımlayan ve malzemesini sınıflandırabilen bir algoritma oluşturulmuştur ve yüksek başarı oranı ile çalışmıştır. Üç boyutlu geometrilerin sınıflandırılabilmesi için özelleştirilmiş bir evrişimli sinir ağı tasarlanmıştır. Üç boyutlu geometrilerin sinir ağlarına öğretilmesi için katı pikselleme yöntemi kullanılmış ve sinir ağı başarı ile eğitilmiştir. Sonlu elemanlar analizinin hızlandırılması ve geometrilerin sınıflandırılarak SEA'ya girdi sağlaması için 3888 farklı analiz sonucu yapay sinir ağlarında eğitilmiş ve doğruluk oranları tayin edilmiştir. Sonlu elemanlar analizi ile çözüm süresi 48 saati geçen problemler, yapay sinir ağı ile 10 saniye içinde %91 doğrulukla sonuç vermiştir.

Anahtar Kelimeler: Derin öğrenme, Evrişimli sinir ağı, Görüntü işleme, Hacimsel pikselleme, Modal frekans analizi, Statik mukavemet analiz

ABSTRACT

MS THESIS

DEVELOPING A DEEP LEARNING ALGORITHM THAT CAN PERFORM THREE DIMENSIONAL OBJECT IDENTIFICATION AND REAL-TIME FINITE ELEMENT ANALYSIS BY VOLUMETRIC PIXELATION METHOD

Ahmet OKUDAN

**Konya Technical University
Institute of Graduate Studies
Department of Mechanical Engineering**

**Advisor: Prof. Dr. Ömer Sinan ŞAHİN
2019, 110 Pages**

**Jury
Prof. Dr. Ömer Sinan ŞAHİN
Assoc. Prof. Dr. Necati ATABERK
Assoc. Prof. Dr. Mehmet BAĞCI**

Finite element analysis is a simulation method which is designed in a computer environment to determine the physical behavior of structures and solved with certain 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. The solutions of artificial intelligence in all areas of our lives in the field of finite element analysis have been investigated and different algorithms have been written for the necessary methods.

In this study, an algorithm that identifies the material to be analyzed and classifies its material has been created by the image processing method and has worked with high success rate. A customized convolutional neural network was designed to classify three-dimensional geometries. In order to teach three dimensional geometries to neural networks, solid pixelation method was used and neural network was successfully trained. In order to accelerate the finite element analysis and classify the geometries and provide input to the FEA, 3888 different analysis results were trained in artificial neural networks and accuracy rates were determined. Finite element analysis showed that the problems with a solution time exceeding 48 hours yielded 91% accuracy in 10 seconds with artificial neural network.

Keywords: Convolutional neural networks, deep learning, image processing, modal analysis, static strength analysis, volumetric pixelation

ÖNSÖZ

Bu tez çalışmasının yürütülmesinde yardımlarıyla yol gösteren danışman hocam Sayın Prof. Dr. Ömer Sinan ŞAHİN ve meslektaşım İbrahim YOZGATLI'ya, bizleri yapay zeka ile tanıştırap bu konuda araştırma yapmayı sevdiren değerli hocam Sayın Arş. Gör. Merve Ayyüce KIZRAK'a ve çalışmam boyunca bana maddi ve manevi destek olan Numesys İleri Mühendislik Hizmetleri A.Ş. personeline teşekkürlerimi sunuyorum.

Ayrıca tez çalışmam süresince büyük bir sabır ile beni destekleyen sevgili eşime de en içten teşekkürlerimi sunuyorum.

Ahmet OKUDAN
KONYA-2019



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
1.1. Statik Yükleme Analizi.....	3
1.2. Doğal Frekans Analizi	3
1.3. Derin Öğrenme	4
1.4. Yapay Sinir Ağı	4
1.5. Evrişimli Sinir Ağları.....	4
1.6. Hacimsel Pikselleme.....	5
1.7. Üç Boyutlu Nesne Sınıflandırma	6
1.8. Tezin Amacı.....	6
2. KAYNAK ARAŞTIRMASI	9
3. MATERYAL VE YÖNTEM.....	11
3.1. Sonlu elemanlar yöntemi	11
3.1.1 Statik Analiz	13
3.1.2 Doğal Frekans Analizi	14
3.1.3. Yapay Sinir Ağı için Eğitim Verisi oluşturma.....	18
3.1.4. Parametrik Tasarım.....	18
3.1.5. Malzeme Mühendislik Verileri	24
3.1.6. Analiz sonucu değerlerinin parametreye bağlanması	26
3.2. Derin Öğrenme	27
3.2.1. Algılayıcı (Perceptron)	27
3.2.2. Çok katmanlı algılayıcı (Multi Layer Perceptron – MLP)	28
3.2.3. Evrişimli Sinir Ağları (Convolutional Neural Networks).....	31
3.2.4. Etkinleştirme Fonksiyonları (Activation Functions)	32
3.3. Görüntü İşleme	35
3.3.1. Veri Seti	35
3.3.2. Veri setlerinin veri çerçevesine (data frame) dönüştürülmesi	37
3.3.3. Evrişimli sinir ağı mimarisinin oluşturulması	39
3.4. Üç Boyutlu Cisim Sınıflandırma	44
3.4.1. Hacimsel pikselleme (Voxelization).....	45
3.4.2. Üç boyutlu obje veri dosyası temini	45
3.4.3. Eğitimde kullanılacak veri dosyaları	49
3.4.4. OFF dosyalarının hacimsel piksel dosyalarına çevrilmesi	49
3.4.5. Binvox dosyalarının elde edilmesi ve görselleştirme	50
3.4.6. Binvox dosyalarının veri seti matrisi içine yazdırılması	54

3.4.7. Üç boyutlu evrişimli sinir ağı mimarisi oluşturulması	55
3.5. Sonlu Elemanlar Analizi Regresyon Sinir Ağı	58
3.5.1. SEM için YSA modeli	58
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	60
4.1. Görüntü İşleme YSA'sının Eğitim Sonuçları	60
4.2. Katı Objeye Sınıflandırma YSA'sının Eğitim Sonuçları	61
4.3. Sonlu Elemanlar Analizleri Çözüm Sonuçları	62
4.3.1. Çelik malzemesi için statik analiz sonuçları	62
4.3.2. Çelik malzeme için modal analiz sonuçları	67
4.3.3. Alüminyum malzeme için statik analiz sonuçları	72
4.3.4. Alüminyum malzeme için modal analiz sonuçları	77
4.3.5. Abs plastik malzemesi için statik analiz sonuçları	82
4.3.6. Abs plastik malzemesi için doğal frekans analizi sonuçları	87
4.3.7. Doğal frekans analizi 1. Modların karşılaştırılması	93
4.3.8. Analiz sonuçlarının yapay sinir ağlarına öğretilmesi	94
5. SONUÇLAR VE ÖNERİLER.....	95
5.1 Sonuçlar	95
5.2 Öneriler	95
KAYNAKLAR	97
EKLER	99
ÖZGEÇMİŞ	110

SİMGELER VE KISALTMALAR

Simgeler

$\{\emptyset\}$: Özvektör
ω	: Özdeğer (Hz)
n	: Serbestlik derecesi
[C]	: Sönüm matrisi
[K]	: Rijitlik matrisi
[M]	: Kütle matrisi
$\{\dot{u}\}$: Hız vektörü
$\{\ddot{u}\}$: İvme vektörü
$\{u\}$: Yer değiştirme vektörü
{F}	: Uygulanan yük

Kısaltmalar

3B-ESA	: Üç boyutlu evrişimli sinir ağı
ABS	: Akrilonitril bütadien strien
AGA	: Aplikasyon geliştirme arayüzü
BINVOX	: Binary voxelization (İkili hacimsel pikselleme)
CAD	: Computer aided design (Bilgisayar destekli tasarım)
ESA	: Evrişimli sinir ağı
KVS	: Katı Veri Seti
LIDAR	: Light detection and ranging (Lazer tespiti ve mesafeleyici)
MVS	: Malzeme Veri Seti
OVS	: Objeye Veri Seti
RELU	: Rectified linear units (Düzeltilmiş doğrusal birimler)
SEM	: Sonlu elemanlar metodu
SEA	: Sonlu elemanlar analizi
STL	: Stereolithography (Stereolitografi)
TBYSAs	: Tam bağlantılı yapay sinir ağı
XOR	: Exclusive or (Özel veya)
YSA	: Yapay sinir ağı

1. GİRİŞ

Analiz ve doğrulama, mühendislikte olduğu gibi pek çok ana bilim dalında yapılan işin veya tasarlanan modelin yapacağı göreve uygunluğunun ölçüldüğü ve test edildiği yöntemlerdir. Günümüzde teknolojik gelişmeler ile mühendisliğin ve mühendislik hesaplamalarının bilgisayar destekli yazılımlar ile kolayca yapılabilmesi ile, analiz ve doğrulamanın da önemi fazlaca artmıştır. On sene öncesinde, tasarlanan ve üretilmesi düşünülen bir ürün, önce tasarım mühendisleri tarafından tasarlanmakta, ardından analiz ve simülasyon uzmanları tarafından bilgisayar ortamında test edilerek doğrulanmaktaydı. Ancak bilgi seviyesi ve hesaplamalı bilgisayar yazılımlarının kullanım kolaylığının artması ile her mühendis tasarım yaparken eş zamanlı olarak analiz ve doğrulama da yapacak seviyeye erişti. Geçmişte, bilgisayar destekli analiz ve simülasyon yazılımları, kullanıcıdan fazlaca teorik bilgi istemekteydi. Ancak günümüzde hazır olarak pek çok mühendislik modeli ve teorisi bilgisayar yazılımlarının içine gömüldüğü için pek çok mühendis bu yazılımları kolayca kavramakta ve kullanabilmektedir. Gelişen teknoloji ve mühendislerin bilgisayar yazılımlarına olan merakının artması ile bilgisayar destekli yazılımların kullanımı da fazlaca artmıştır. Ayrıca bilgisayar mühendislerinin yanı sıra, makine ve imalat mühendislerinin de yapay zekâ ve derin öğrenmeye merakı artmış, bu sayede farklı bilim dalları birleşerek Üretimsel Tasarım (Generative Design) adı altında farklı bir mühendislik dalı doğmuştur.

Üretimsel tasarım, bilgisayar destekli tasarım programına, yapılacak tasarımın birkaç girdisinin belirtilmesi ile, algoritmalarından kendi kendine tasarımı oluşturup bu tasarım üzerinden analizler yürütmesi ve ortaya pek çok parametreye bağlı pek çok tasarım ortaya çıkarmasına dayanmaktadır. Böylece mühendisler ve tasarımcılar, pek çok farklı tasarımı elleri ile tek tek çizmekten, algoritmanın ortaya çıkardığı yüzlerce tasarım içinden imalat yöntemine veya müşteri isteğine en yakın tasarımı iyileştirerek nihai tasarımı ortaya çıkarabilmektedir. Bu nebzedeki birkaç hafta ile birkaç ay aralığında süren tasarım ve prototipleme işlemleri birkaç saat aralığında tamamlanabilmektedir.

Bilgisayar destekli tasarım ve analiz yazılımları, tasarımcı ve mühendislere bilgisayar ortamındaki süreçlerde yardımcı olduğu gibi, tasarım sonrası prototipleme ve imalat kısmında da yardımcı olmaktadır. Gelişen teknoloji ve bilgi birikimi ile imal edilecek parçanın imalat prosedürüne uygun olarak da analizleri yapılmakta ve imalatta

doğacak sorunlar önceden tespit edilebilmektedir. Bu yöntemler ile hem zamandan hem de maliyetten kâr edilebilmektedir.

Mühendisliğin pek çok alanında kullanılan simülasyon programlarında, model doğrulama ve oluşabilecek riskleri görmek için tekil parça veya sistemlerin analiz simülasyonları yapılmaktadır. Katı parçaların statik, mekanik, titreşim, şok, termal ve yorulma analizleri, akış içeren sistemlerin basınç, akış hızı, debi, kavitasyon ve türbülans analizleri, elektrikli sistemlerin manyetizma ve radyasyon analizleri, sıvı-katı etkileşimli sistemlerin mekanik-akışkanlar etkileşim analizleri çoğunlukla “Sonlu Elemanlar Yöntemi” veya “Hesaplamalı Akışkanlar Dinamiği” ile çözüm yapabilen simülasyon yazılımları ile yapılmaktadır.

“Sonlu Elemanlar Yöntemi” veya “Hesaplamalı Akışkanlar Dinamiği” ile simülasyon yapan yazılımlar, bir tasarım üzerinden analiz yapıp sonuç çıktısı verebilmeleri için, geometri, sınır şartları, analiz koşulları ve ağ örgüsü gibi girdilere ihtiyaç duyar. İleri düzey simülasyon programlarında doğru analizi koşturabilmek için yüksek derece mühendislik bilgisi, analizi yorumlayabilmek içinse mühendislik tecrübesi gerekir. Tekil parçaların mekanik veya akışkanlar analizi bir ile on saat arasında sürerken, sistemler büyüdükçe bu süre katlanarak artmaktadır. Tez çalışması statik sistemlerin sonlu elemanlar analizleri üzerinde yapılacaktır. Statik sistemlerin statik yükleme ve doğal frekanslarının belirlemeleri üzerinden bir veri seti oluşturulacaktır. Statik yükleme analizleri bölüm 1.1’de, doğal frekans analizleri ile ilgili kısaca bilgi de bölüm 1.2’de verilecektir.

Yapay zekâ, insanoğlu için her zaman ulaşılması istenilen bir olgu olmuştur. Transistörlerin icat olunması ile yapay zekanın en basit çalışma mantığı olan sınıflandırma araştırılmaya başlanmış, bilgisayarların veri saklayabilme ve hesaplama gücünün artması ile hız kazanmış ve derin öğrenme kavramının ortaya çıkması ile günümüzdeki kullanım kolaylığına ve yaygınlığına erişmiştir. Cep telefonumuzdaki yüz tanıma sistemleri, araçlardaki mesafe algılayıcılar, oto pilotlar, bankalardaki sesli asistanlar, gelişmekte olan insansı robotlar yapay zekayı oluşturan algoritmalar bütünü ile çalışmaktadır. Yapay zekâ insanlığın çoğu işini kolaylaştırdığı ve yapılan işleri de öğrenerek, tekrarlı işlerde yorum yapabilme yetisini kazanmaya başladığı için, mühendislik alanlarından da giderek etkisini göstermeye başlamıştır. Yapay zekayı oluşturan algoritmalarından olan derin öğrenme bölüm 1.3’de, yapay sinir ağı ve derin öğrenme bölüm 1.4’de, evrişimli sinir ağları bölüm 1.5’de kısaca anlatılacaktır. Çalışmada üç boyutlu geometrilerin yapay sinir ağları için piksellenmiş veriye

dönüştürülmesi teknolojisinden kısaca bölüm 1.6'da, piksellenen veri için gereken yapay sinir ağı yapısı bölüm 1.7'de kısaca anlatılacaktır. Bölüm 1.8'de tezin amacından ve tez çalışmasının özgün taraflarından bahsedilecektir.

1.1. Statik Yükleme Analizi

Statik analiz, bir sistemin denge hali analizi olarak tanımlanabilir. 1980'lere kadar mühendisler cisimlerin fiziksel davranışını el ile hesaplamaktaydı. Ancak makinelerin karmaşıklaşması ve çoklu-fizik çözümlerine ihtiyaç duyulması mühendislerin bilgisayar destekli analiz yazılımlarını kullanmasını mecbur kıldı.

Bir cismin katı haldeki davranışı, bir noktasının uzayda sabitlenmesi ve başka bir noktasından yük uygulanması statik analizin temellerini oluşturur. Bilgisayar destekli analiz yazılımları, katı geometriyi sonlu matematiksel modellere bölerek çözer ve bu çözüm modellerini toplayarak kümülatif sonucu verir. Statik yükleme analizi bir cismin durağan (denge) halinde ne kadar sehim yapacağını ve üzerinde ne kadar gerilme oluşacağını sonucunu hesaplar. Statik yükleme analizi doğrusal denklemler ile hesaplanır. Ancak geometri belirli bir uzunluğun üzerinde ise doğrusal olmayan denklemlerle hesaplanması gerekir. Bu yüzden analiz algoritmaları yakınsama yaparak sehim ve gerilmeyi adımlara bölerek hesaplar. Böyle büyük modellerde çözüm süresi bilgisayar hızına bağlı olarak 1 saatten fazla zaman alabilir. Bu tez çalışmasının asıl amaçlarından birisi de, statik analiz sonuçlarını bir yapay sinir ağına öğretmekle doğrusal olmayan modellerin 1 dakikadan az bir süre içinde çözülmesini hedeflemektedir.

1.2. Doğal Frekans Analizi

Her cismin bir doğal frekansı vardır. Doğal frekans, kabaca bir cismin sınır koşullarına bağlı olarak bir salınımı tamamladığı saniye aralığıdır. Örnek olarak bir saatin sarkacı her salınımı bir saniyede tamamlar ve frekansı 1 Hz'dir. Eğer bir cisme uygulanan kuvvet aralığı, cismin doğal frekansı ile çakışırsa, cismin salınımı sonsuza gidebilir. Bu durumda parça veya sistemde hasarlar meydana gelir. Bu yüzden tasarımcılar ve mühendisler, tasarladıkları ve üretecekleri sistem veya parçaların çalışma koşullarında doğal frekansının tetiklenmesini incelemek için doğal frekans (modal) analizi koştururlar.

1.3. Derin Öğrenme

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 (Hinton,2013). Bu tez çalışmasında statik ve modal analiz sonuçları sinir ağına öğretileceği için gözetimli öğrenme ile algoritmalar optimize edilecektir.

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 kullanılacaktır.

1.4. Yapay Sinir Ağı

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 zeka uygulamalarında çokça kullanılan yapılardır (Hinton,1990). 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.

1.5. Evrişimli Sinir Ağları

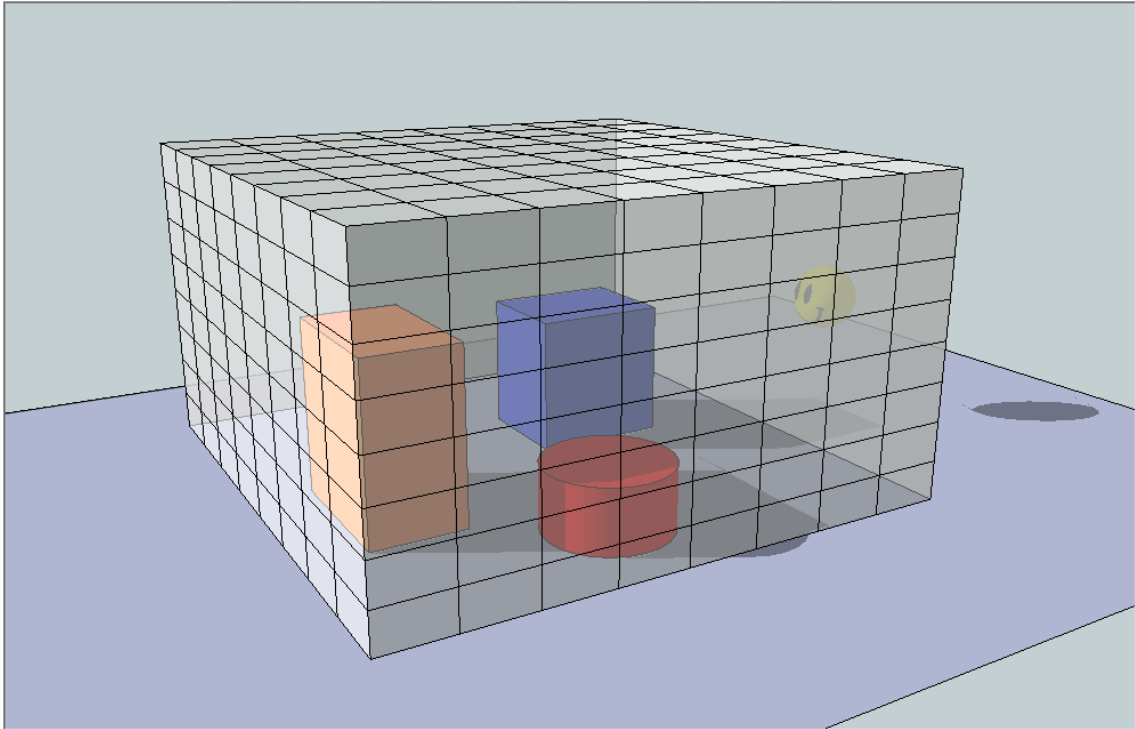
Konvolüsyonel olarak da anılan evrişimli sinir ağları, çok katmanlı yapay sinir ağlarının özelleştirilmiş bir versiyonudur (Googfellow,2016). Veri, sinir ağlarından geçirilmeden önce, konvolüsyon filtreleri tarafından bazı ön matematiksel fonksiyonlardan geçirilir. Böylece görüntü, ses veya üç boyutlu verilerin baskın özellikleri daha iyi saptanabilir.

Tez çalışmasında, görüntü işleme ile katı cisimlerin hangi malzemeden imal edildiğinin tespiti yapılacaktır. Evrişimsel sinir ağları ile, parlama, yüzey pürüzü, matlık, ışık geçirgenliği ve renk gibi özelliklerin konvolüsyon filtreleri ile daha anlaşılır ve öğrenilir bir hale getirilmesi hedeflenmektedir.

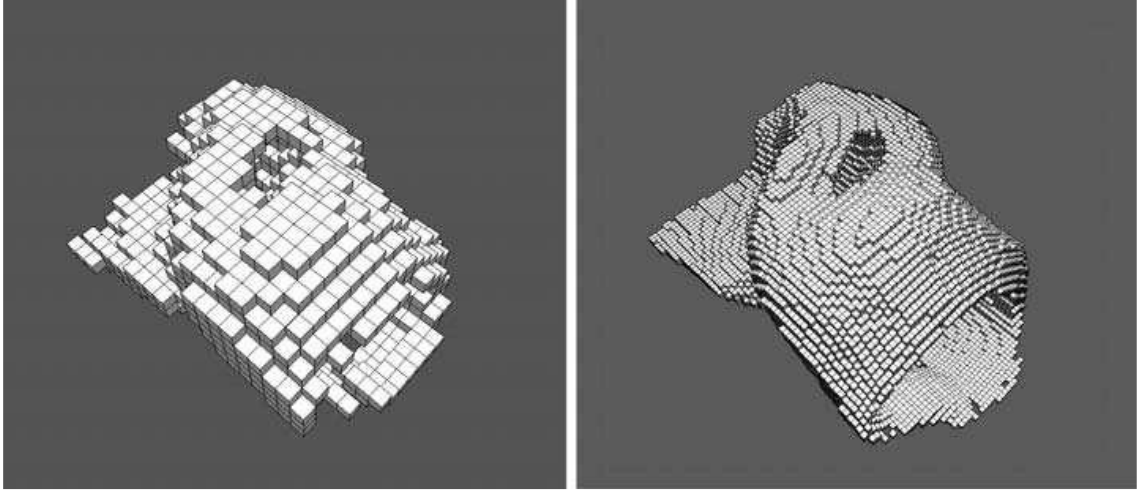
1.6. Hacimsel Pikselleme

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.

Şekil 1.1’de uzayda yer işgal eden geometriler ve geometrileri kapsayan hacimsel piksel birimleri gösterilmiştir. Hacimsel pikselleme işleminin ardından geometri, yüzey eğriliğini kaybedeceği için, piksellerin birim kenar uzunluğu, geometrinin ne kadar gerçeğe yakın olacağını etkiler. Şekil 1.2’de aynı geometri üzerindeki farklı piksel derecelerinin etkisi gösterilmiştir. Birim boyutu küçüldükçe bilgisayar ortamında piksellenen geometrinin saklandığı dosyanın boyutu ve pikselleme işleminin işlem süresi de artar. Bu yüzden bu tez çalışmasında, yapay sinir ağlarının, bir geometriyi tanımlarken mümkün olan en optimum piksel genişliği saptanacaktır.



Şekil 1.1. Hacimsel Pikselleme işleminin uzaydaki birim piksellerinin gösterimi, 8x8x8 birimlik uzay sınırı belirtilmiştir



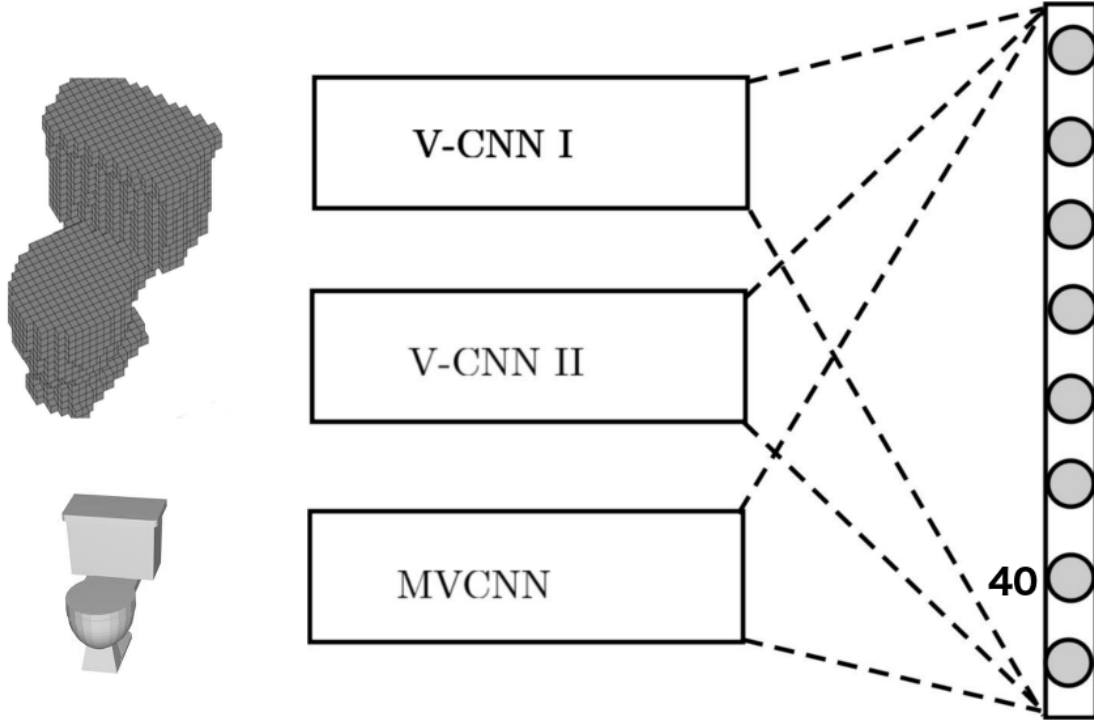
Şekil 1.2. Hacimsel pikselleme işleminde piksel sayısı arttıkça, geometrinin doğruluğa yaklaşma durumu gösterilmiştir (Bier, 2007)

1.7. Üç Boyutlu Nesne Sınıflandırma

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 piksellenecek bilgisayar yazılımı tarafından görülebilir hale getirilir. Piksellenmiş veri daha önce eğitilmiş yapay sinir ağlarından geçirilerek objelerin sınıflandırılması ve ayrıştırılması işlemlerini gerçekleştirerek objeyi tanımlar (Şekil 1.3). Üç boyutlu sınıflandırma, görüntü işleme yönteminde derinlik algısının olmamasından dolayı tercih edilebilmektedir. Birbirine çok yakın renklerde cisimler, derinlik bilgisi sayesinde görüntü işleme yöntemine göre daha doğru şekilde tanımlanmaktadır.

1.8. Tezin Amacı

Bilgisayar destekli analizlerde en büyük problemler, hazırlık ve çözümleme süresinin beklenenden uzun sürebilmesidir. Çoğu firma, tekrarlı işlerinde hız kazanmak için, analiz proseslerine özel kod dizinleri yazarak yapılan işleri otomatikleştirmeye çalışmaktadır. Ancak çözüm, yine formüller üzerinden iterasyonlarla yapıldığı için, model karmaşası arttıkça çözüm süresi de uzamaktadır. Bilgisayarda üç boyutlu çizimi olmayan parça ve sistemler de analiz edilemediği için yapılan işin büyük bir bölümü bu hazırlık süresinde geçmektedir.



Şekil 1.3. Bir geometrinin hacimsel pikselleme verisinin ve evrişimli sinir ağı mimarisini (Hedge,2016)

Çalışmada, ilk olarak parçaların görüntülerinden neye benzedikleri (dişli, kasnak, piston vb.) ve hangi malzemeden yapıldıklarının (çelik, alüminyum, bronz vb.) tespit edilmesi için çok katmanlı yapay sinir ağları eğitilecektir. Bu problemde, parça sınıflandırma ve malzeme tayini için optimum yapay sinir ağının oluşturulması için optimizasyon yapılacaktır.

Çalışmada ikinci adım olarak, Üç boyutlu çizim geometrilerini sınıflandırabilen birçok katmanlı konvolüsyonel sinir ağı oluşturulacaktır. Üç boyutlu geometrileri yapay sinir ağları direkt olarak veri olarak alamayacağı için, CAD geometrisinden volümetrik pikselleme (Voxelization) verisi elde etme algoritması ve volümetrik piksel verisini, üç boyutlu eğitim tensörüne çevirme algoritması da yazılacaktır. Elde edilen veriler, üç boyutlu bir konvolüsyonel sinir ağı ile eğitilecektir.

Çalışmanın üçüncü adımında, statik ve modal analizlerden elde edilen veriler, yapay sinir ağlarına öğretilecektir. Yapay sinir ağları genelde tek çıkışlı ikili sınıflandırma (binary classification) düzeninde oluşturulmaktadır. Ancak analiz verileri birden çok sınıfa sahip olduğu için yapay sinir ağı çok girişli ve çok çıkışlı olarak dizayn edilecektir. Analiz sonuçları ile eğitilen yapay sinir ağının hangi aktivasyon fonksiyonu ile en doğru tahmini yaptığı test edilecektir.

Çalışmanın bütünü, LIDAR, STL Kamera, Derinlik kamerası gibi cihazlar ile görüntüsü ve uzay örüntüsü elde edilen verilerde otomatik olarak simülasyon yapabilecek bir sistemin alt yapısını oluşturması hedeflenmektedir.



2. KAYNAK ARAŞTIRMASI

Bu bölümde tez çalışmasında tasarlanacak olan algoritmaları bütününde kullanılması gereken yöntemler ve daha önce yapılan çalışmalar hakkında inceleme yapılmıştır. Çalışmanın hızlandırılması ve deney sayısının azaltılması adına literatür araştırmasında, tasarlanan algoritmalar veya yapılan deneylerden elde edilen verilere bakılarak eski veya tutarsız yöntemlerin denenmeden güncel algoritma ve fonksiyonların öncelik olarak denenmesine karar verilmiştir.

Rosenblatt (1958), yapay zeka'yı oluşturan algoritmalar bütününden birisi olan "Yapay Sinir Ağı"nın en küçük işlem birimi olan Perceptron (Algılayıcı)'yı ifade edilmiştir. Rosenblatt, algılayıcı fonksiyonunu biyolojik bir sinir hücresinin çalışma mantığından esinlenerek tasarlamıştır. Algılayıcının klasik sınıflandırıcılardan farkı ikili ayırım yapabilmesidir.

Hinton (2013)'in bulgularına göre algılayıcı, tek katmanlı bir yapay sinir hücresidir ve doğrusal problemlerin sınıflandırmasını yapabilmektedir. Fakat doğrusal olmayan problemlerin sınıflandırılmasında yetersiz kalmaktadır. Hinton ve ark. yaptıkları çalışmada, algılayıcıları birbirine bağlayarak, çok katmanlı bir yapay sinir ağı elde etmişlerdir. Bu yöntemle doğrusal olmayan sınıflandırmaları da başarı ile yapabilmişlerdir. Çok katmanlı yapay sinir ağları görüntü işleme ve çok girdili, çok çıktılı yapılarda kullanılmaya başlanmıştır.

Otaïr ve ark. (2006), çok katmanlı yapay sinir ağını, doğrusal olmayan XOR problemini çözmek için kullanmışlardır. Bu çalışma ile Çok katmanlı yapay sinir ağının, çözüm uzay düzlemini bükerek, sınıflandırmayı doğrusal bir çizgi üzerine getirdiği tespit edilmiştir.

Hinton ve ark. (2012), evrişimli sinir ağlarını (ESA) tasarlayarak tanıtmışlardır. Çok katmanlı yapay sinir ağı üzerine inşa edilen modelde, resim gibi 2 boyutlu veriler, yapay sinir ağına girdi olarak beslenmeden önce çeşitli filtrelerden geçirilerek hem boyutları indirgenmiş, hem de kenar, köşe, çizgi, renk ayrımları gibi özelliklerin ön plana çıkması ile öğrenme hızı artarak gereken işlem gücü azaltılmıştır. Hinton ve arkadaşlarının yaptığı bu çalışma ile görüntü işleme teknolojilerindeki ilerleme hızı kazanmıştır.

Maturada ve ark. (2015), ESA'ların iki boyutlu resimlerde kullanıldığı gibi, üç boyutlu nokta bulutu ve radar verilerinde de başarı ile çalışmıştır. Maturana ve ark, yaptıkları çalışmada, radar verisini voxelize ederek (hacimsel pikselleme) 3 boyutlu bir

matrise yazdırmış, matrisi CNN ağlarından geçirerek sınıflandırma işlemini başarı ile yapmışlardır.

Schmidhuber ve Hochreiter (1997) yılında LSTM (Long Short Term Memory) – Uzun Kısa Devirli Hafıza’yı tanıtmışlardır. Yapay sinir ağları, girdi ve çıktılarla eğitildiği için, ileriye dönük tahmin yapamamaktadır. Bu çalışmada, LSTM algoritması ile, eğitim verilerinde, gelecek veriyi tahmin edebilecek şekilde hücreler eğitilmektedir. Çalışma, bu anlamda eğitim verisindeki veri aralığının dışında kalan sınıflandırmaları da yapabilmek için yol gösterici özelliğindedir

Wang ve ark. (2018), iki boyutlu akış problemlerini çözerek, iki boyuttaki akış alanlarını Çok katmanlı yapay sinir ağı ve uzun-kısa devirli hafıza ağlarına öğretmişlerdir. Böylece önceden tanımlı akış problemlerinde uzun iterasyonlara gerek kalmadan, hem belirli zaman aralığındaki akışı yapay sinir ağı ile çözümleyip, hem de zaman aralığının dışında kalan iterasyonları uzun-kısa devirli hafıza algoritması ile çözümlenmişlerdir.

Liang ve ark. (2018), iki boyutlu bir tüpteki akış problemini, tüpün üstüne gelen kuvvetleri gözlemlemek amacı ile “sonlu elemanlar yöntemi” ile analizini modelleyerek çözdürmüş ve 2 boyutlu ağ örgüsü (mesh) üzerindeki düğüm noktalarından (nodes) gerilme değerlerini alarak, bu gerilme değerini 2 boyutlu bir matrise yazdırmışlardır. Elde ettikleri matrisi çok katmanlı yapay sinir ağına öğreterek, gerilme analizinin eğitilmiş YSA ile çözülmesini sağlamışlardır.

Mutlu (2018), hiperspektral görüntü ve lidar verisini kullanarak, 3 boyutlu Evrişimsel Sinir Ağı (ESA) eğitmiştir. lidar verisinin ESA ile tahmininde ilk başta %56,78 isabet oranı yakalanmıştır. softmax fonksiyonu ile çıkış verisini yorumlayan sinir ağı, modele eklenen karar fonksiyonları ve yönlendirme mekanizmaları ile %93,88 isabet oranına ulaşmıştır.

3. MATERYAL VE YÖNTEM

Bu tez çalışmasında ortam şartlarını girdi olarak kullanıp, bir cismin statik analiz ve doğal frekans analizini, sonlu elemanlar metodu formüllerini kullanmadan derin öğrenme algoritması çıktıları ile saptayabilen alt yapı algoritması tasarlanmıştır. Bu yüzden farklı alanlardaki materyal ve yöntemler farklı başlıklarda incelenecektir.

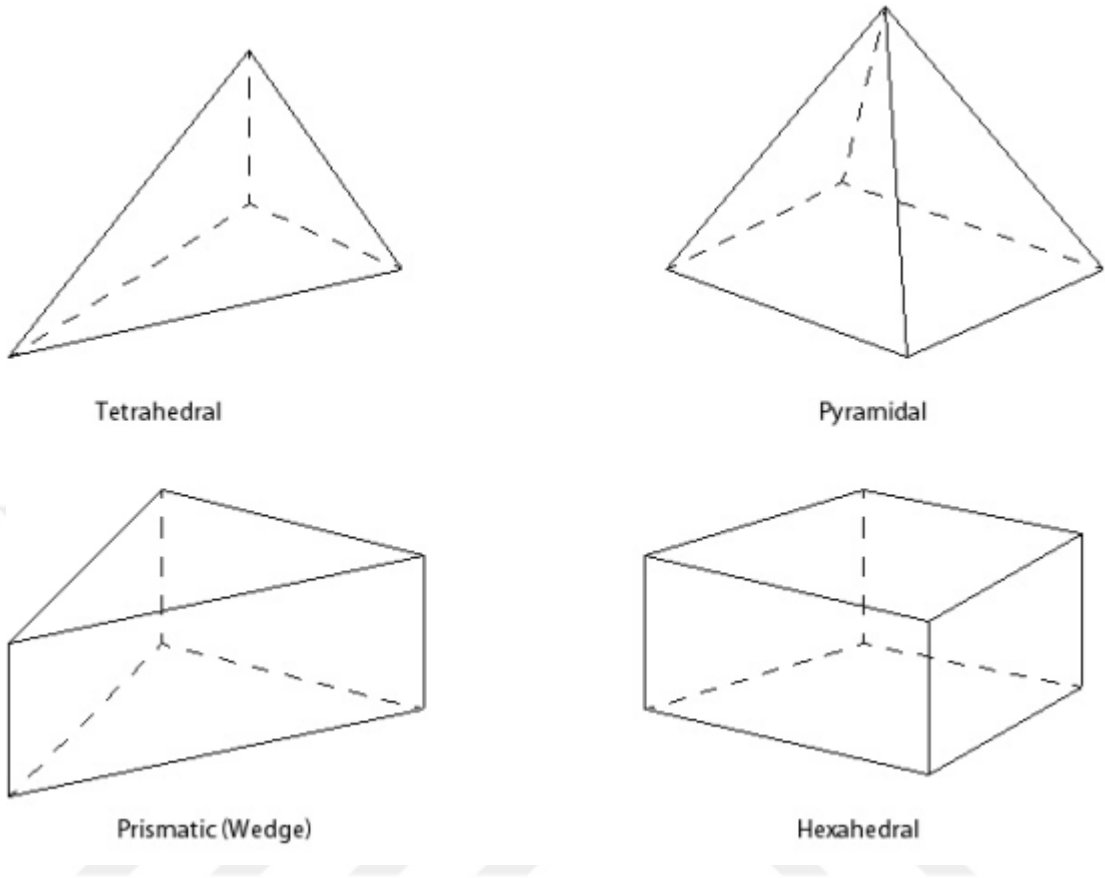
Parametrik olarak oluşturulacak olan bir kirişin farklı boyutlardaki geometrileri için statik ve doğal frekans analizleri Bölüm 3.1’de, görüntü işleme ve üç boyutlu nesne tanımlama yöntemleri Bölüm 3.2’nin altında anlatılacaktır.

3.1. Sonlu elemanlar yöntemi

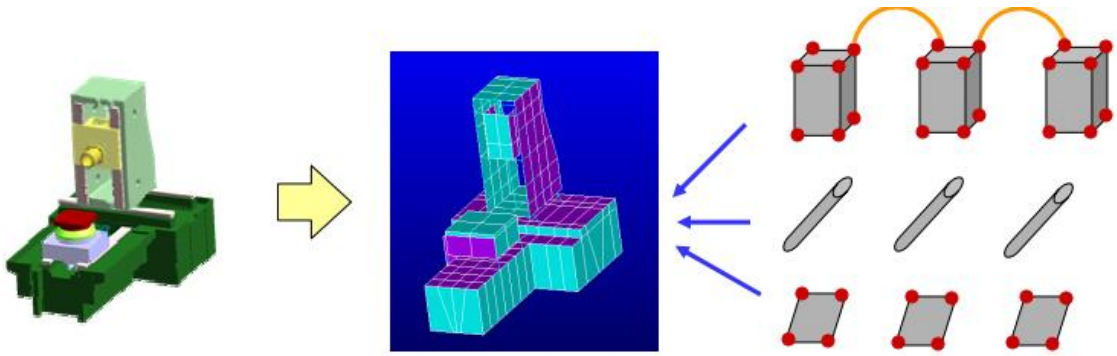
Pek çok mühendislik probleminde, sistemin davranışını incelemek ve sistemi analiz etmek için bu sistemin matematiksel modelinin çıkarılması ve bu matematiksel modelin sınır şartlarına uygun olarak çözümlenmesi gerekmektedir. Çözümün basitleştirilmesi için bazı kabuller yapılır ve diferansiyel denklemi oluşturulan sistem modeli kurulur. Ancak bu denklemlerin çözülmesi zor ve uzun zaman alıcıdır. Bilgisayarların hesaplama kabiliyetinin artması ve depolama alanlarının büyümesi ile çok büyük sistemlerin matematiksel modelleri hızlıca çözümlenebilmektedir. Mühendislik problemlerinin çözümlenmesinde farklı matematiksel modeller elde edilmektedir. Çoklu-fizik simülasyonları, farklı matematiksel modellerin farklı çözüm yöntemleri ile çözümlenme ve çözüm birleştirme yöntemi ile gerçekleştirilir.

Sonlu elemanlar yöntemi, mühendislik çözümlerinde sistemleri analiz etmek için en çok kullanılan yöntemlerden birisidir. Sonlu elemanlar yöntemi, bir cismin sonsuz sayıdaki bilinmeyenini, sonlu sayıdaki elemanlara (elements) bölerek, elemanların birleşim noktalarına düğüm noktası (node) ataması ilkesine dayanır. Sonlu elemana bölünen cisim ve sistem, her eleman için elemanı temsil eden şekil fonksiyonu ile çözümlenir.

Elemanların bir araya gelmesi ile ağ elemanı (mesh) oluşur. Problemin veya geometrinin farklılığına göre farklı ağ elemanları tercih edilebilir. Şekil 3.1’de farklı ağ elemanları gösterilmiştir. Şekil 3.2’de geometilerin ağ elemanlarına dönüşümü gösterilmiştir. Tez çalışmasında, Ansys ticari yazılımında bulunan ve daha doğru sonuç veren, 20 düğüm noktalı, polinomial fonksiyona sahip kübik (hexahedral) ağ eleman tipi tercih edilmiştir.



Şekil 3.1. Ansys ticari yazılımında farklı tiplerdeki üç boyutlu ağ elemanın gösterimi



Şekil 3.2. Ağ örgüsü oluşturulan bir sistemin gösterimi

Sonlu elemanlar yöntemi analizleri genel olarak aşağıdaki unsurlar ile oluşturulur;

- Geometrilerin yazılım içinde entegrasyonu
- Geometri veya yüzeylerin sonlu sayıda elemana bölünmesi
- Her bir eleman için eleman direngelik (rijitlik) matrisinin oluşturulması

- Elemanların direngenlik matrisinin global (küresel) direngenlik matrisi içine yerleştirilmesi
- Sınır şartlarının belirtilmesi
- Çözümleme yönteminin belirtilmesi
- Zaman adımlarının belirtilmesi
- Temas / mafsal olması durumunda temas kriteri belirtilmesi
- Denklemlerin çözülmesi
- Sonuçların yazdırılması

3.1.1 Statik Analiz

Statik analizler, cisimlerin durağan (denge hali) davranışlarını elde etmek için gerçekleştirilen analiz çeşididir. Statik analizler kendi içinde Çizelge 3.1’de verilen analiz çeşitlerine ayrılırlar

Çizelge 3.1. Statik analizlerde geometri, malzeme ve çözücünün doğrusallık farklarının gösterimi

Numara	Geometri	Malzeme	Çözücü
1	Doğrusal	Doğrusal	Doğrusal
2	Doğrusal olmayan	Doğrusal	Doğrusal olmayan
3	Doğrusal	Doğrusal olmayan	Doğrusal olmayan
4	Doğrusal olmayan	Doğrusal olmayan	Doğrusal olmayan

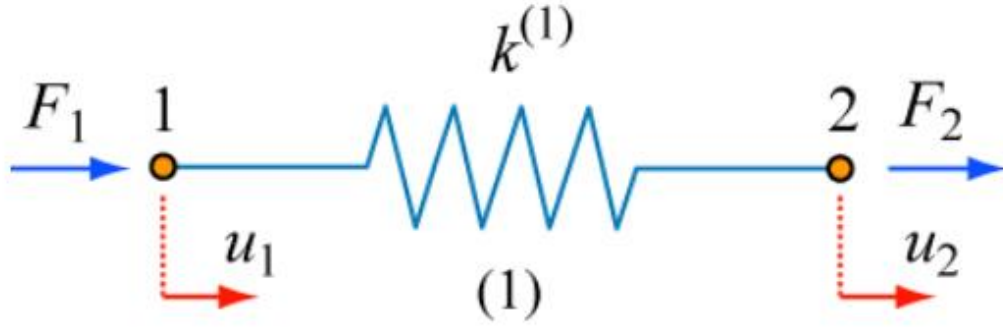
Çizelge 3.1.’de gösterildiği üzere, geometri ve malzemenin doğrusallığına göre çözücü farklılık göstermektedir. Bu tez çalışmasında geometri ve malzeme doğrusal seçilmiştir. Bu yüzden çözücü olarak “direkt yaklaşım” veya diğer adı ile “direkt rijitlik metodu” yöntemi tercih edilmiştir.

Direk yaklaşım metodu, basit geometri ve problemler için kullanılır ve analiz çözümü, tek bir yakınsama ile gerçekleştirilmektedir.

3.1.1.1. Direkt yaklaşım yöntemi

Şekil 3.3’de gösterilen, tek serbestlik derecesine sahip, bir rijitliği bulunan elastik eleman için, bir düğüm noktasının ankastre kabul edildiği durumda kuvvet denklemi şu şekilde ifade edilmektedir.

$$dF = -kdu \quad (1.1)$$



Şekil 3.3. İki serbestlik dereceli, rijitliğe ve esnekliğe sahip iki düğüm noktalı eleman gösterimi (Arslan,2014)

Denklem 1.1.'de gösterilen eşitlik, tek serbestlik dereceli elastik eleman için yazılabilir. Ancak şekil 3.3'deki gibi iki serbestlik dereceli eleman için eşitlik şu şekilde olmalıdır.

$$k(e) u_i - k(e) u_j = F_i (e) \quad (1.2)$$

$$-k(e) u_i + k(e) u_j = F_j (e) \quad (1.3)$$

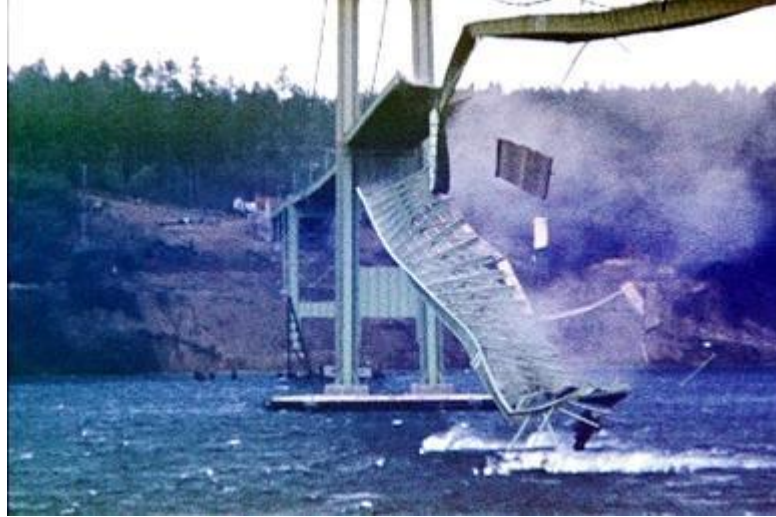
Bu eşitlikten eleman rijitlik matrisine dönüşüm yapılırsa, matris simetrik olacağı için eşitlik şu halde yazılabilir:

$$[K(e)]\{ u(e) \} = \{ F(e) \} \quad (1.4)$$

Denklem 1.4.'de, K , elemanın rijitlik matrisini, u düğüm (nodal) deplasman vektörünü, F ise düğümsel (nodal) kuvvet vektörünü ifade etmektedir.

3.1.2 Doğal Frekans Analizi

Doğal frekans analizinin her tasarım için yapılması gerekliliğinin en acı örneği, 1940 yazında Washington'da yıkılan Tacoma Köprüsü'dür (Şekil 3.4). Tacoma Köprüsü, yatayda etkiyen rüzgar yükü altında salınım yapmaya başlamış ve harmonik salınım, köprünün yatay mod frekanslarından birisi ile çakıştığı için salınma genliği sonsuza gitme eğilimi göstermiştir. Salınımın artması sonucu, doğal olarak halat ve konstrüksiyon üzerindeki gerilme arttığı için hasar meydana gelmiştir ve köprü yıkılmıştır.



Şekil 3.4. 1940 yılında doğal frekansı tetiklenen Tacoma Köprüsü'nün yıkılış anı (Harish,2019)

Sonlu elemanlar yönteminde doğal frekans analizi ile mod tayin etmek için kullanılan pek çok yöntem vardır. Modal analizler, sönümlü ve sönümsüz olarak ikiye ayrılır. Ancak yazılım alt yapısında 5 farklı çözüm yöntemi bulunmaktadır.

- Block Lanczos (Direkt yöntem), 40'dan fazla mod bulunması gereken, katı ve levha elemanlar ile de çözümleme yapabilen, simetrik metot.
- PCG Lanczos (İteratif yöntem), 100'den fazla modu bulunan ve 500,000'den fazla serbestlik dereceli sistemler için ideal çözücüdür. Katı geometriler için elverişlidir, çözümde burkulma sonucu okunamaz.
- Unsymmetric, K ve M matrislerini simetrik olmayan çözüm yöntemi ile tanımlar ve çözer
- SuperNode (Süper Düğüm), 10,000'den fazla mod için ve yapıda çok fazla iki boyutlu eleman bulunuyorsa.
- Subspace (Alt çözüm uzayı), Direkt yöntem ile aynı fakat burkulma hesabı yapılabilmekte.

Tez çalışmasında Ansys ticari yazılımında doğal frekans analiz çözücüsü Direkt Yöntem'dir.

Teorik altyapısı ile Direkt Yöntem'i açıklamak istersek, genel hareket denklemi;

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \quad (1.5)$$

Burada;

$[M]$ = Kütle matrisi

$[C]$ = Sönüm matrisi

$[K]$ = Rijitlik matrisi

$\{F\}$ = Uygulanan yük

$\{\ddot{u}\}$ = İvme vektörü

$\{\dot{u}\}$ = Hız vektörü

$\{u\}$ = Yer değiştirme vektörü

Sönümsüz (undamped) doğrusal bir serbest titreşim hareketinde sönüm matrisi $[C]$ ve uygulanan yük $\{F\}$ sıfır olacağından 1.5'de ki denklem aşağıdaki gibi ifade edilebilir. $\{F\}$ matrisi harmonik analizlerde sıfırdan farklı olmaktadır.

$$[M]\{\ddot{u}\} + [K]\{u\} = \{0\} \quad (1.6)$$

Doğal titreşim sonucunda oluşan rezonans durumu harmonik hareket olarak kabul edilebilir, 1.6'da ki denklemde şu harmonik hareket kabullerini yaparsak:

$$\{u\} = \{\emptyset\}_i \sin(\omega_i t + \theta_i) \quad (1.7)$$

$$\{\dot{u}\} = \omega_i \{\emptyset\}_i \cos(\omega_i t + \theta_i) \quad (1.8)$$

$$\{\ddot{u}\} = -\omega_i^2 \{\emptyset\}_i \sin(\omega_i t + \theta_i) \quad (1.9)$$

Ve denklemimizde $\{u\}$ ve $\{\ddot{u}\}$ harmonik hareket kabullerimizi 1.6'daki denklemde yerine yazacak olursak:

$$\begin{aligned} -\omega_i^2 [M] \{\emptyset\}_i \sin(\omega_i t + \theta_i) + [K] \{\emptyset\}_i \sin(\omega_i t + \theta_i) &= \{0\} \\ (-\omega_i^2 [M] + [K]) \{\emptyset\}_i &= \{0\} \end{aligned} \quad (1.10)$$

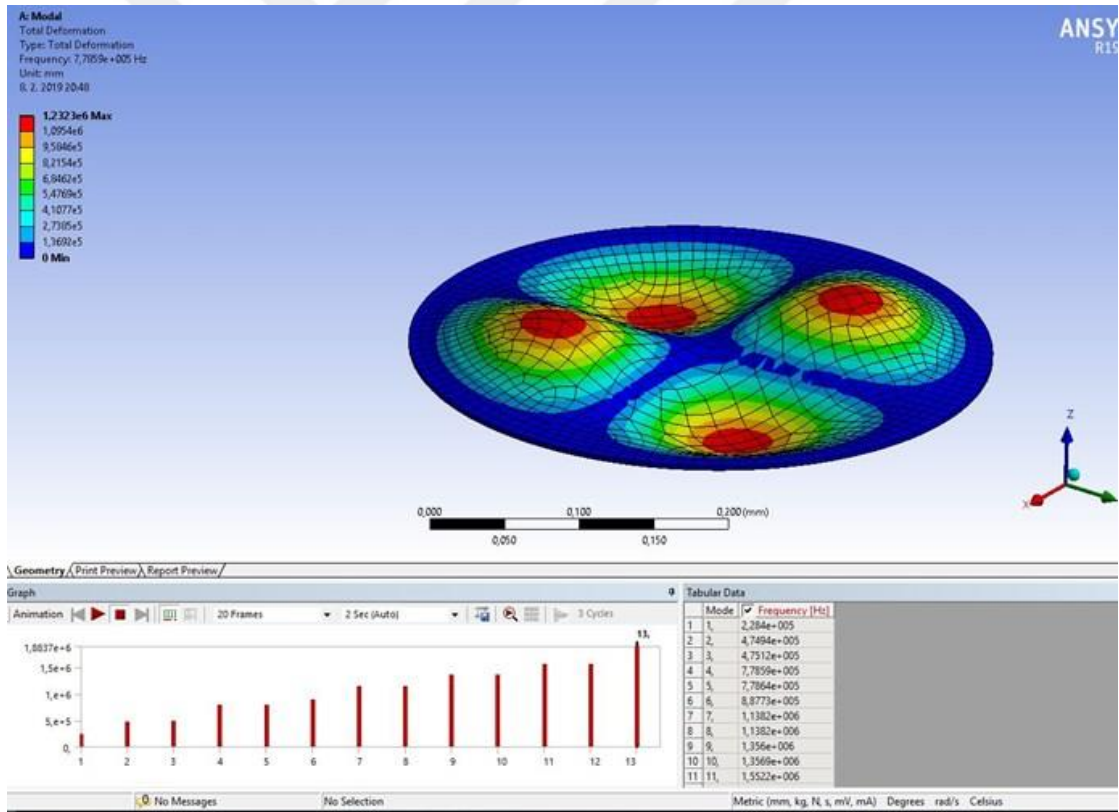
1.10'da ki denklemin determinantını alarak çözdüğümüzde özdeğerleri (eigenvalue) elde ederiz.

$$\det([K] - \omega_i^2 [M]) = \{0\} \quad (1.11)$$

Olmalıdır.

1.11’de ki denklemin çözümünden $\omega_1, \omega_2, \dots, \omega_n$ özdeğerlerini yani doğal frekanslarını (mod frekanslarını) bulmuş oluruz. Burada ‘n’ sayısı yapının serbestlik derecesi (degrees of freedom) sayısıdır.

Sönümsüz bir serbest titreşimde, kütle ve direngelik matrisi ile özdeğerler değiştiği için, bu parametreler öğrenimi yapılacak yapay sinir ağlarına girdi olarak verilmemiştir. Sebebi, yapay sinir ağlarının, bağımlı giriş parametreleri arasındaki bağlantıyı kendisi kurabilmesidir. Burada kütle boyuta, direngelik’de malzemeye ve yine boyuta bağlı olduğu için eğitim verilerine dahil edilmeyecektir. Böylece YSA’ların bağımlı değişkenleri saptama ve çıkarım yapma konusundaki başarıları da test edilmiş olacaktır.



Şekil 3.5. Ansys Ticari paket yazılımda bir cismin doğal frekans modlarının tespitinin gösterimi

3.1.3. Yapay Sinir Ağı için Eğitim Verisi oluşturma

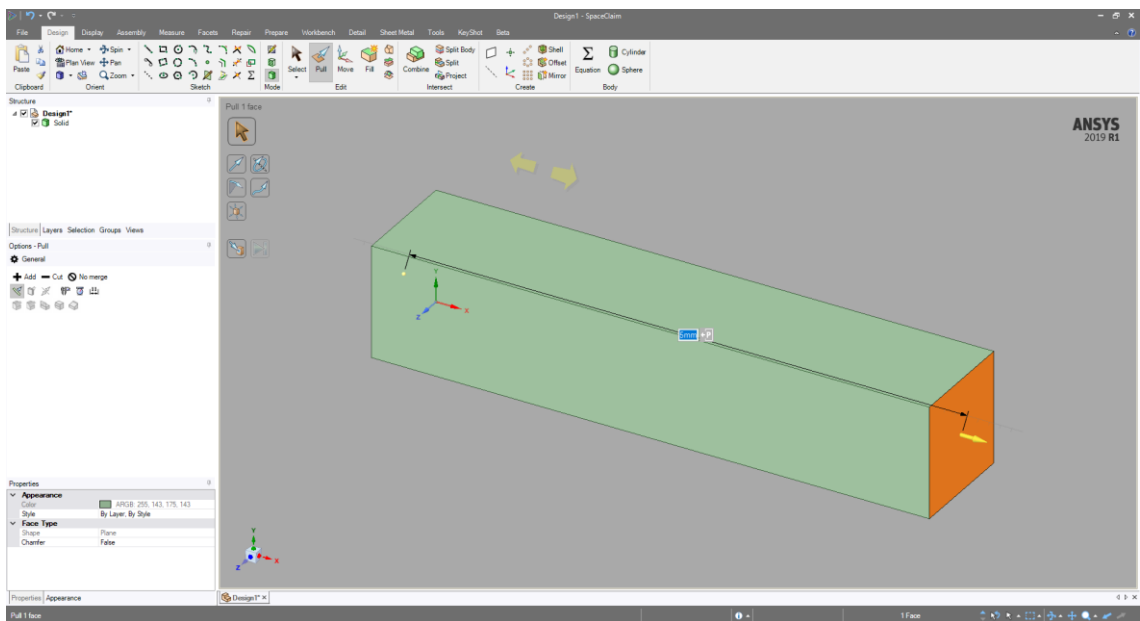
Tez çalışmasında, statik ve doğal frekans analizlerinin sonuçları ile bir yapay sinir ağı eğitilecektir. Yapay sinir ağlarında öğrenme veri sayısı arttıkça daha verimli olacağı için analizler parametrik olarak tasarlanacak ve parametre permütasyonu kadar analiz yapılacaktır. Analizler, üç farklı malzeme için, altı farklı en, yükseklik ve uzunluk verisi ile koşturulacaktır. Çizelge 3.2.'de analizi yapılacak modelin malzeme ve parametre bilgileri verilmiştir.

Çizelge 3.2. Analiz çeşitliliği parametreleri

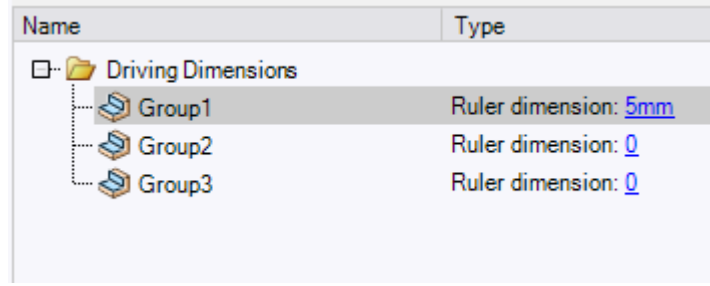
Malzeme	En (mm)	Yükseklik (mm)	Uzunluk (mm)
Çelik	2	2	5
Alüminyum	3	3	10
ABS Plastik	5	5	25
	10	10	100
	25	25	250
	100	100	1000

3.1.4. Parametrik Tasarım

Analizler için gereken parametrik geometri Ansys ticari paket yazılımının, geometri editörü olan SpaceClaim paket yazılımında tasarlanacak ve gerekli parametreler oluşturulmuştur



Şekil 3.6. SpaceClaim ticari paket yazılımında modelin parametrik olarak çizilmesi



Şekil 3.7. SpaceClaim ticari paket yazılımında çizimi yapılan modelin parametre gruplarına bağlanması

SpaceClaim yazılımında parametreye bağlanan en, yükseklik ve uzunluk ölçüleri ile parametre tablosu oluşturulmuştur. Bu durumda 3 farklı uzunluk parametresinin 6 farklı parametre değeri için parametre permütasyonu yapıldığında 216 adet varyans ortaya çıkmıştır.

Çizelge 3.3. Parametre permütasyonundan oluşturulan varyans tablosu

Tasarım Numarası	En (mm)	Yükseklik (mm)	Uzunluk (mm)
1	1	1	5
2	1	1	10
3	1	1	25
4	1	1	100
5	1	1	250
6	1	1	1.00E+03
7	1	2	5
8	1	2	10
9	1	2	25
10	1	2	100
11	1	2	250
12	1	2	1.00E+03
13	1	4	5
14	1	4	10
15	1	4	25
16	1	4	100
17	1	4	250
18	1	4	1.00E+03
19	1	9	5
20	1	9	10
21	1	9	25
22	1	9	100
23	1	9	250
24	1	9	1.00E+03

25	1	24	5
26	1	24	10
27	1	24	25
28	1	24	100
29	1	24	250
30	1	24	1.00E+03
31	1	99	5
32	1	99	10
33	1	99	25
34	1	99	100
35	1	99	250
36	1	99	1.00E+03
37	2	1	5
38	2	1	10
39	2	1	25
40	2	1	100
41	2	1	250
42	2	1	1.00E+03
43	2	2	5
44	2	2	10
45	2	2	25
46	2	2	100
47	2	2	250
48	2	2	1.00E+03
49	2	4	5
50	2	4	10
51	2	4	25
52	2	4	100
53	2	4	250
54	2	4	1.00E+03
55	2	9	5
56	2	9	10
57	2	9	25
58	2	9	100
59	2	9	250
60	2	9	1.00E+03
61	2	24	5
62	2	24	10
63	2	24	25
64	2	24	100
65	2	24	250
66	2	24	1.00E+03
67	2	99	5
68	2	99	10
69	2	99	25

70	2	99	100
71	2	99	250
72	2	99	1.00E+03
73	4	1	5
74	4	1	10
75	4	1	25
76	4	1	100
77	4	1	250
78	4	1	1.00E+03
79	4	2	5
80	4	2	10
81	4	2	25
82	4	2	100
83	4	2	250
84	4	2	1.00E+03
85	4	4	5
86	4	4	10
87	4	4	25
88	4	4	100
89	4	4	250
90	4	4	1.00E+03
91	4	9	5
92	4	9	10
93	4	9	25
94	4	9	100
95	4	9	250
96	4	9	1.00E+03
97	4	24	5
98	4	24	10
99	4	24	25
100	4	24	100
101	4	24	250
102	4	24	1.00E+03
103	4	99	5
104	4	99	10
105	4	99	25
106	4	99	100
107	4	99	250
108	4	99	1.00E+03
109	9	1	5
110	9	1	10
111	9	1	25
112	9	1	100
113	9	1	250
114	9	1	1.00E+03

115	9	2	5
116	9	2	10
117	9	2	25
118	9	2	100
119	9	2	250
120	9	2	1.00E+03
121	9	4	5
122	9	4	10
123	9	4	25
124	9	4	100
125	9	4	250
126	9	4	1.00E+03
127	9	9	5
128	9	9	10
129	9	9	25
130	9	9	100
131	9	9	250
132	9	9	1.00E+03
133	9	24	5
134	9	24	10
135	9	24	25
136	9	24	100
137	9	24	250
138	9	24	1.00E+03
139	9	99	5
140	9	99	10
141	9	99	25
142	9	99	100
143	9	99	250
144	9	99	1.00E+03
145	24	1	5
146	24	1	10
147	24	1	25
148	24	1	100
149	24	1	250
150	24	1	1.00E+03
151	24	2	5
152	24	2	10
153	24	2	25
154	24	2	100
155	24	2	250
156	24	2	1.00E+03
157	24	4	5
158	24	4	10
159	24	4	25

160	24	4	100
161	24	4	250
162	24	4	1.00E+03
163	24	9	5
164	24	9	10
165	24	9	25
166	24	9	100
167	24	9	250
168	24	9	1.00E+03
169	24	24	5
170	24	24	10
171	24	24	25
172	24	24	100
173	24	24	250
174	24	24	1.00E+03
175	24	99	5
176	24	99	10
177	24	99	25
178	24	99	100
179	24	99	250
180	24	99	1.00E+03
181	99	1	5
182	99	1	10
183	99	1	25
184	99	1	100
185	99	1	250
186	99	1	1.00E+03
187	99	2	5
188	99	2	10
189	99	2	25
190	99	2	100
191	99	2	250
192	99	2	1.00E+03
193	99	4	5
194	99	4	10
195	99	4	25
196	99	4	100
197	99	4	250
198	99	4	1.00E+03
199	99	9	5
200	99	9	10
201	99	9	25
202	99	9	100
203	99	9	250
204	99	9	1.00E+03

205	99	24	5
206	99	24	10
207	99	24	25
208	99	24	100
209	99	24	250
210	99	24	1.00E+03
211	99	99	5
212	99	99	10
213	99	99	25
214	99	99	100
215	99	99	250
216	99	99	1.00E+03

Çizelge 3.3.'de gösterildiği üzere, 216 farklı geometri program tarafından otomatik olarak oluşturulmuştur. Bu tasarım noktalarının her birine üç farklı malzeme için statik ve doğal frekans analizi yapılacaktır. Statik analiz sonuçlarında maksimum gerilme ve maksimum sehim (deformation) değerleri, doğal frekans analizinde ise parçanın bir tarafı ankastre olarak mesnetli iken yerçekimsiz olarak ilk 6 doğal frekans değeri tayin edilecektir. Tasarım noktalarında, en ve yükseklik değerlerinin parametre değerlerinden “1” eksik olmasının sebebi, SpaceClaim yazılımının düzlem çiziminde parametre bağlanmasını ölçüsel değil çizimsel olarak algılamasıdır. Çizimsel parametreler, analiz koşturma esnasından yazılımın hata vermesine sebep olacağı için, en ve boy değerleri başlangıç olarak 1 mm kabul edilmiştir. Parametre değerleri, başlangıç değerlerinin üzerine eklendiğinde Çizelge 3.2.'deki değerler sağlanmıştır.

3.1.5. Malzeme Mühendislik Verileri

Analiz sonuçları ile eğitilecek YSA'ya, elastikiyet modülü ve poisson oranı değerlerinin de öğretilmesi için 3 farklı malzeme verisi tanımlanacak ve analizler her tasarım noktası için 3 farklı malzeme üzerinden koşturulacaktır. Tek malzeme ile eğitilen YSA'nın, farklı malzemelerin mekanik özelliklerini öğrenemeyeceği için bu yöntem tercih edilmiştir. Malzemeler Ansys ticari programının “mühendislik verisi” kütüphanesinden temin edilmiştir.

Statik analizler, direkt yaklaşım metodu ile, doğrusal formülasyon kullanılarak çözüleceği için, malzemenin plastik şekil değiştirme bölgesine girmeyeceği kabul edilmektedir. Bu sebeple mühendislik verisinde, statik analiz koşturabilmek için Elastikiyet modülü (Young modulus) ve poisson oranının bulunması yeterlidir. Doğal frekans analizi için ise, frekanslar kütle ile değişeceği için malzeme yoğunluğunun

bilinmesi gerekmektedir. Kullanılan malzemelerin mühendislik değerleri çizelge 3.4.'de verilmiştir.

Çizelge 3.4. Analizlerde kullanılacak malzemelerin mühendislik özellikleri

Malzeme Cinsi	Elastikiyet Modülü	Poisson Oranı	Yoğunluğu
Çelik	200	0.3	7800
Alüminyum	71	0.3	2770
ABS Plastik	1.1	0.42	950

The image shows two screenshots from the Ansys Engineering Data interface. The top screenshot, titled 'Outline of Schematic A2: Engineering Data', displays a table with columns A, B, C, D, and E. Row 4 is highlighted, showing 'Material' as 'ABS'. The bottom screenshot, titled 'Properties of Outline Row 4: ABS', shows a table of material properties for ABS. The table has columns A (Property), B (Value), C (Unit), D, and E. The properties listed include Density (950 kg m⁻³), Isotropic Secant Coefficient of Thermal Expansion, Coefficient of Thermal Expansion (0.00023 C⁻¹), Isotropic Elasticity, Young's Modulus (1.1E+09 Pa), Poisson's Ratio (0.42), Bulk Modulus (2.2917E+09 Pa), Shear Modulus (3.8732E+08 Pa), Tensile Yield Strength (2.5E+07 Pa), Compressive Yield Strength (0 Pa), Tensile Ultimate Strength (3.3E+07 Pa), Compressive Ultimate Strength (0 Pa), Isotropic Thermal Conductivity (0.28 W m⁻¹ C⁻¹), and Specific Heat, C_p (2300 J kg⁻¹ C⁻¹).

Property	Value	Unit	D	E
Material Field Variables	Table			
Density	950	kg m ⁻³		
Isotropic Secant Coefficient of Thermal Expansion				
Coefficient of Thermal Expansion	0.00023	C ⁻¹		
Isotropic Elasticity				
Derive from	Young's Modu...			
Young's Modulus	1.1E+09	Pa		
Poisson's Ratio	0.42			
Bulk Modulus	2.2917E+09	Pa		
Shear Modulus	3.8732E+08	Pa		
Tensile Yield Strength	2.5E+07	Pa		
Compressive Yield Strength	0	Pa		
Tensile Ultimate Strength	3.3E+07	Pa		
Compressive Ultimate Strength	0	Pa		
Isotropic Thermal Conductivity	0.28	W m ⁻¹ C ⁻¹		
Specific Heat, C _p	2300	J kg ⁻¹ C ⁻¹		

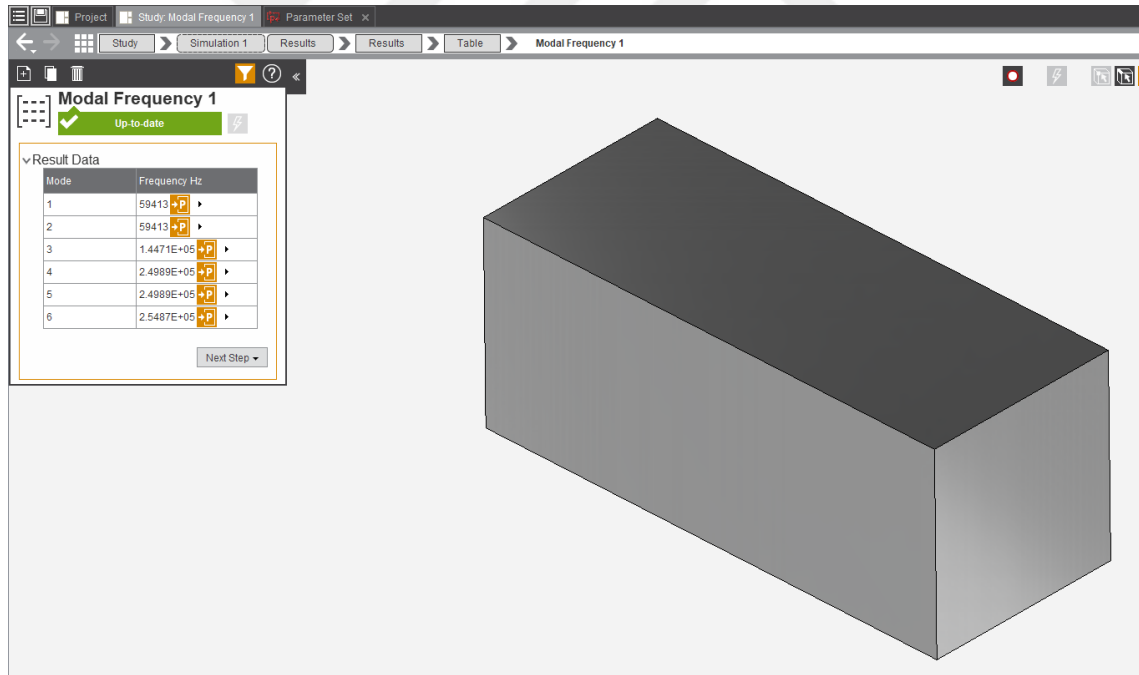
Şekil 3.8. Ansys ticari paket yazılımında malzemelerin seçilmesi ve değerlerin girilmesi

Çizelge 3.4'de ABS (Akrilonitril Bütadien Stiren) plastiğinin elastikiyet modülünün 1.1 MPa olduğu görülmektedir. Hafif ama sert bir polimer olan ABS, sanayinin pek çok alanında kullanılmaktadır. Üç boyutlu yazıcılar ile üretimi de mümkündür. Fakat elastikiyet modülü çok düşük olduğu için, kuvvet altında çelik ve

alüminyuma göre çok daha fazla sehim yapacaktır. Yapay sinir ağlarında eğitilecek veriye ABS polimerinin analiz sonuçlarının da eklenmesi, öğrenme algoritmasının rijitliği çok düşük malzemeleri de tanıyabilmesi için fayda sağlayacaktır ve tahmin yapabilme yeteneğini artıracığı düşünülmektedir.

3.1.6. Analiz sonucu değerlerinin parametreye bağlanması

Çizelge 3.3 ve çizelge 3.4’de belirtildiği üzere, her malzeme için 216 adet, toplamda 648 adet statik ve 648 adet doğal frekans analizi gerçekleştirilecektir. Statik analizlerden maksimum deformasyon ve maksimum sehim değerleri alınacaktır ve toplamda 1296 farklı sonuç bulunacaktır. Doğal frekans analizlerinde ise ilk 6 doğal frekans modu sonucu alınacağı için 648 farklı analizden 3888 farklı frekans modu değeri alınacaktır. Değerlerin parametrik olarak bir tabloya yazdırılabilmesi için, şekil 3.9’da gösterildiği gibi Ansys ticari paket yazılımında, sonuçlar parametre tablosuna bağlanmıştır.



Şekil 3.9. Ansys AIM ticari programında doğal frekans modu sonuçlarının parametreye bağlanması

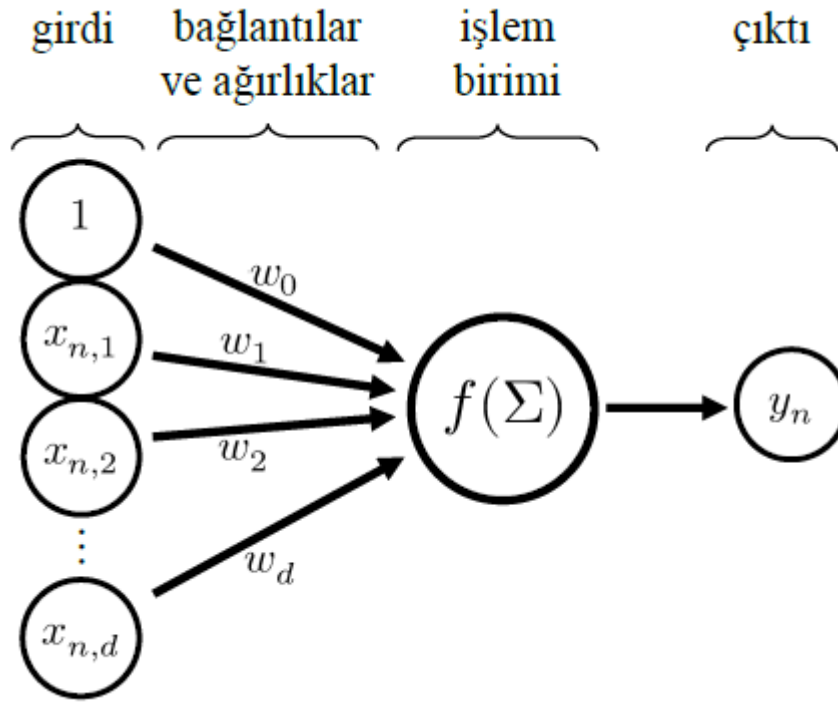
Bu işlemde yapıldıktan sonra program çoklu çözüm yapmaya bırakılacaktır. Alınan sonuçlar yapay sinir ağı için eğitim verisini oluşturacaktır. Yapay sinir ağı için hangi mimarinin kullanılacağı bölüm 3.2’de anlatılacaktır.

3.2. Derin Öğrenme

Tez çalışmasında, farklı alanlardaki veriler ile farklı YSA'lar eğitilmiştir. Cisimlerin sınıflandırılması ve malzemelerinin tanımlanması için çok filtreli evrişimli sinir ağı, üç boyutlu cisimlerin sınıflandırılması için iki filtreli evrişimli sinir ağı ve sonlu elemanlar analizi verilerinin eğitilmesi için çok katmanlı yapay sinir ağı kullanılmıştır. Farklı mimarilerdeki bu YSA'ların daha iyi anlaşılması ve tez çalışmasında kullanılan materyal ve yöntemlerin gösterilmesi için bu bölümde çeşitli bilgiler verilecektir.

3.2.1. Algılayıcı (Perceptron)

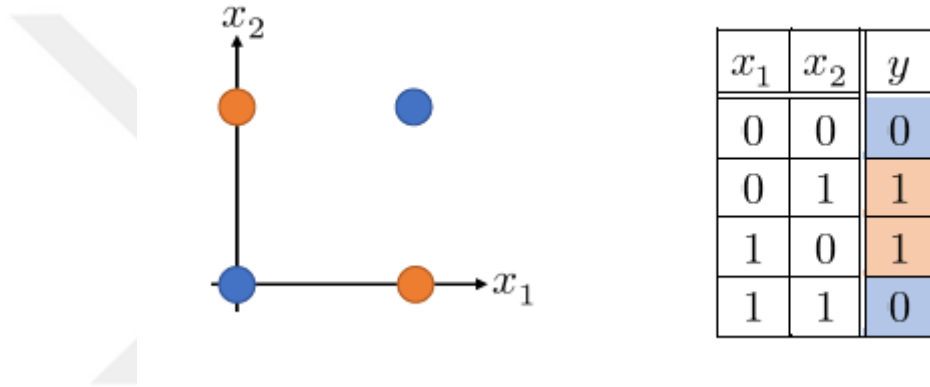
Algılayıcı, YSA'nın en küçük işlem birimidir. İnsan sinir hücresinin yapısı baz alınarak bulunan bu yapı ikili (binary) sınıflandırma becerisine sahiptir. Algılayıcı, basitçe girdi vektörünün bulunduğu uzayı doğrusal şekilde ikiye böler. Bu yüzden doğrusal olmayan problemlerin çözümünde veya öğrenilmesinde kullanılması yetersiz sonuç vermektedir.



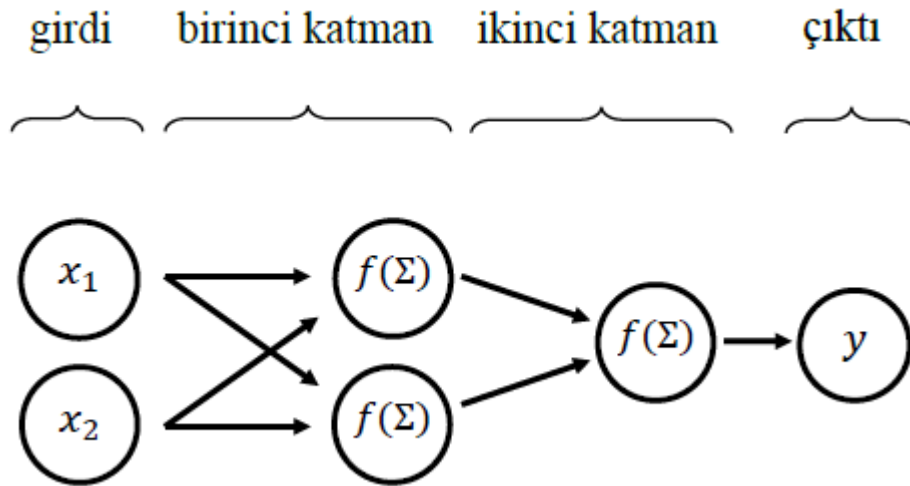
Şekil 3.10. Şematik olarak algılayıcının yapısının gösterimi

3.2.2. Çok katmanlı algılayıcı (Multi Layer Perceptron – MLP)

İleri beslemeli sinir ağı veya derin yapay sinir ağı olarak da adlandırılan, katmanların birbiri ile tam bağlandığı algılayıcı ağları, derin öğrenmede en çok kullanılan yapılardır. Doğrusal problemlerin sınıflandırılmasında tek bir algılayıcı yeterli olurken, doğrusal olmayan problemlerin sınıflandırılmasında çok sınıflı çıktı verebilen çok katmanlı algılayıcılar tercih edilmektedir. XOR kapısı problemi, doğrusal olmayan problemlerden birisidir (Şekil 3.11). XOR problemi, bir algılayıcının girdi katmanı ve aktivasyon fonksiyonu arasına, iki elemanları ara (gizli) katman eklenmesi ile çözülebilir (Şekil 3.12).

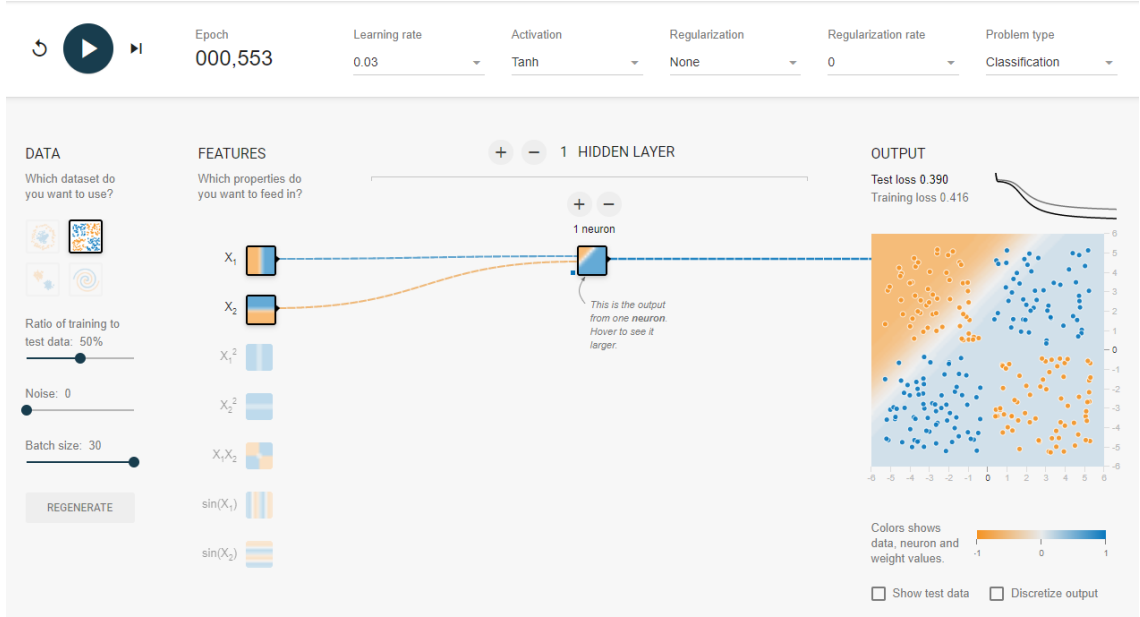


Şekil 3.11. Doğrusal olmayan XOR probleminin gösterimi, renkler aldıkları değeri temsil etmektedir.

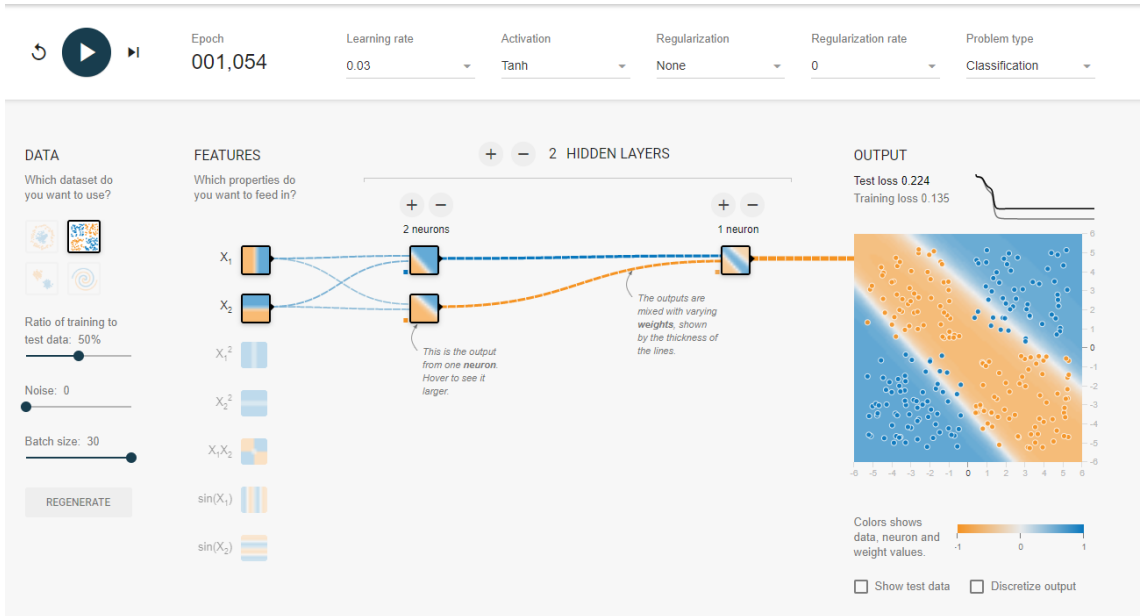


Şekil 3.12. XOR problemini çözebilen örnek bir derin yapay sinir ağı

Yapay sinir ağlarının ve gizli katmanların yaptığı görevin daha iyi anlaşılması için Google şirketinin açık kaynak kodlu olarak yayınladığı. Şekil 3.13’de Tensorflow kütüphanesinin web arayüzünde XOR problemi tanımlıdır. Burada istenilen YSA inşa edilerek problemin çözümü denenmiştir. XOR problemini çözmek için tek bir algılayıcı yeterli olmazken, Şekil 3.14’de görüldüğü üzere, iki nörona sahip gizli katman problemi sınıflandırabilmiştir.

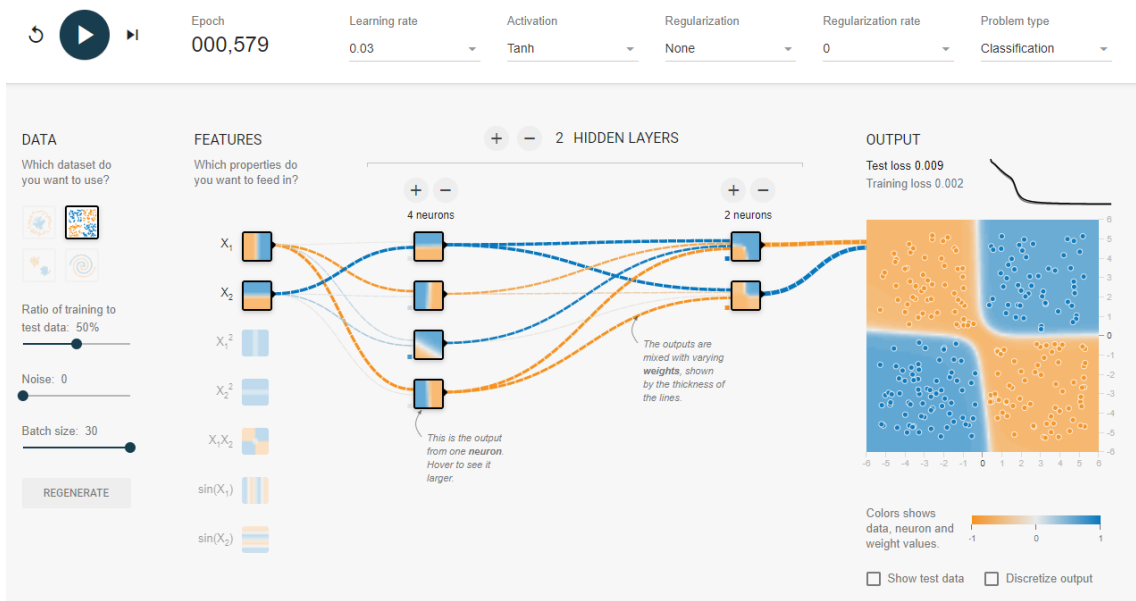


Şekil 3.13. Tensorflow kütüphanesinin Web arayüzünde XOR probleminin tek algılayıcı ile çözümü

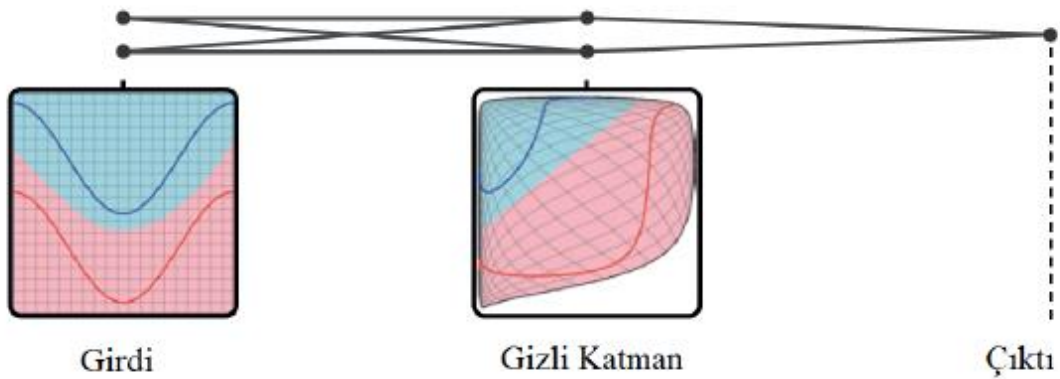


Şekil 3.14. Tensorflow kütüphanesinin Web arayüzünde XOR probleminin çok katmanlı algılayıcı ile çözümü

XOR problemi, değişkenlerin sapmaları da dikkate alındığı için Tensorflow sitesinde gürültü (noise) ihtiva etmektedir. Bu yüzden problemin daha iyi sınıflandırılması için nöron sayısı artırılabilir. Nöron sayısını artırdığımızda, çözüm uzayının nasıl büküleceğini gözlemleyebiliriz. Şekil 3.16'da anlatıldığı ve şekil 3.15'de tecrübe edildiği gibi, ara katmanların ve nöron sayısının artması girdi uzayını bükerek/bozarak çıkış cevabınının doğrusal olmama durumunu doğrusallığa indirgemektedir.



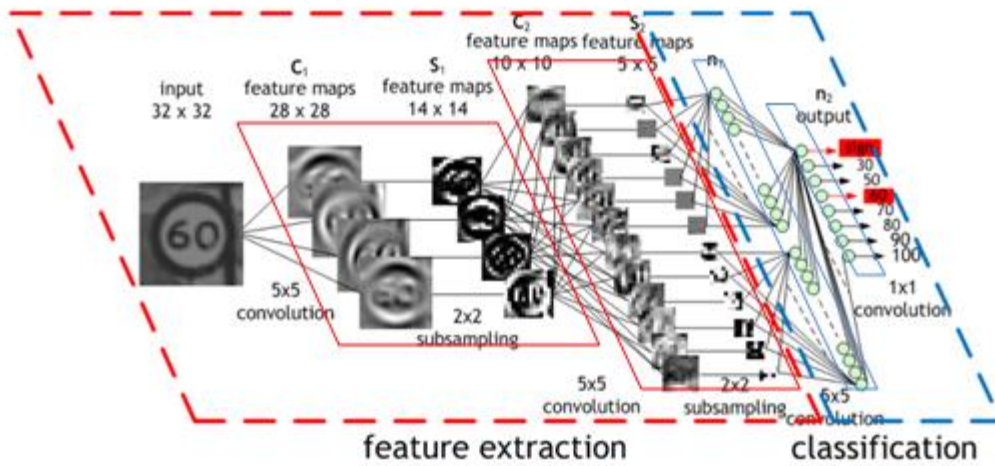
Şekil 3.15. XOR probleminde sınıflandırmanın daha iyi yapılması için gizli katmana iki, çıkış katmanına da bir adet nöron eklenmiştir. Test 0.009 hata içermektedir.



Şekil 3.16. Çok katman ve katmanlardan birden fazla nöron bulunduran yapay sinir ağları girdi uzayını bozar ve farklı sınıfların veri bölgelerini doğrusal olarak ayrılabilir hale getirir. (Hinton, 2012)

3.2.3. Evrişimli Sinir Ağları (Convolutional Neural Networks)

Konvolüsyonel olarak da adlandırılan ESA'lar, 2012'de yayınlandığında, dünya çapında yapılan görüntü işleme yarışmalarında büyük yankı uyandırdı ve bir çok teknolojiye kullanılmaya başlandı. ESA'lar girdileri direkt olarak yapay sinir ağına beslemektense, ön filtreleme, eliminasyon, ağırlık ortaya çıkarmak gibi işlemleri yaptıktan sonra ilk katmana verileri aktarmaktadır (Şekil 3.17). Bu yöntem ile, resim ve video gibi verilerin işlenmesinde, insanın gözünün ayrıntılardan çok objeye odaklanması mantığına benzeyen filtreleme işlemleri sayesinde daha hızlı ve daha doğru sonuçlar üretilmeye başlandı. Öte yandan ESA'ların kullanılması ile, yapay zeka algoritmalarının daha önce hiç görmediği verilere karşı tahmin yeteneği de artmış oldu.



Şekil 3.17. Konvolüsyon filtreleri resmin kenar, köşe, geçiler gibi özelliklerini ortaya çıkaran filtrelere sahiptir. Filtreler öğrenme boyunca optimize edilir. (Khalid,2018)

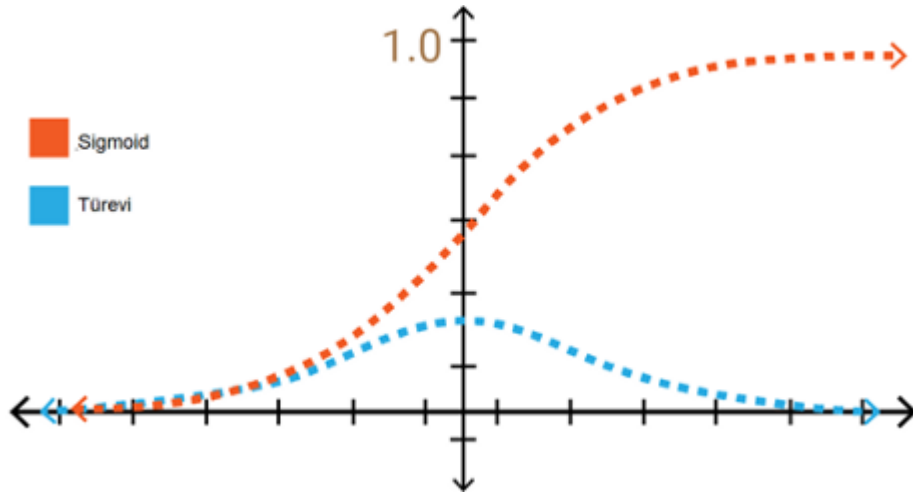
Tez çalışmasında kullanılacak olan evrişimli sinir ağları modellerinin uygunluğu hakkında fikir vermesi için, Tensorflow kütüphanesinin web arayüzünden faydalanılmıştır. ESA'lar Tensorflow kütüphanesinin Python programlama dilinde yazılan bir AGA (Aplikasyon Geliştirme Arayüzü – API) olan Keras kütüphanesi ile oluşturulacaktır.

3.2.4. Etkinleştirme Fonksiyonları (Activation Functions)

Şekil 3.10'da gösterilen tek bir algılayıcının işlem birimi bir etkinleştirme fonksiyonudur. Etkinleştirme fonksiyonu, kendisinden önceki birimde girdi değerlerini bir ağırlık ile çarpımlarının toplandığı, bu toplamın eşik değerini aşması durumunu kontrol eden, ve eşik değeri aşıldığında belli bir çıktı veren birimdir. İkili sınıflandırma ile çalışan bir algılayıcının etkinleştirme fonksiyonu basamak (step) fonksiyonudur ve çıktı olarak 0 veya 1 değerini üretir. Eğer girdiler çarpımı toplamı eşik değerinden büyükse, 1 çıktısı alınır. Bu bölümde derin öğrenme uygulamalarında en çok kullanılan aktivasyon fonksiyonları anlatılacaktır.

3.2.4.1. Sigmoid fonksiyonu

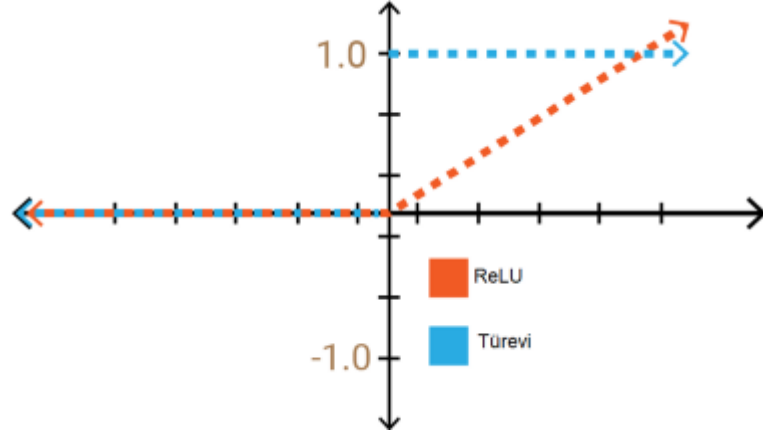
Basamak fonksiyonundan farklı olarak, sigmoid fonksiyonları 0 ile 1 arasındaki değerleri verebilir. Fonksiyon $x = -2$ ve $x = 2$ aralığındaki küçük değişimlerde büyük tepkiler verdiği için bir sınıflandırıcı olarak kullanılabilir. Fakat sigmoid fonksiyonu çoğu uygulamada tercih edilmez. Bunun sebebi çok büyük değerlerde sürekli olarak 1'e yakın değerler üretmesi ve bunun sonucunda gradyan ölmesi (vanishing gradient) olayının yaşanmasına neden olur. Bu durum gradyan ölümü yaşayan nöronun da ölümüne sebep olacağı için ilgili nöronda öğrenme gözlenmez. Sigmoid fonksiyonu Şekil 3.18'de gösterilmiştir.



Şekil 3.18. Sigmoid fonksiyonu ve fonksiyonun türevinin gösterimi

3.2.4.2. Doğrultulmuş lineer birim (Rectified linear unit – ReLU) fonksiyonu

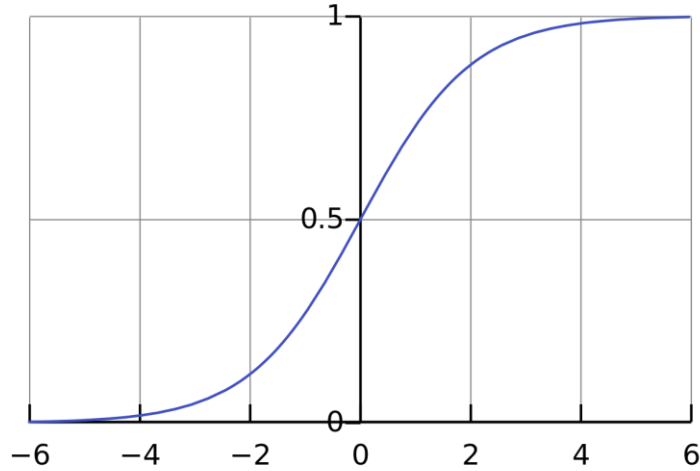
Relu fonksiyonu $[0, +\infty)$ aralığında deper almaktadır. Çok fazla nöronlu büyük yapay sinir ağlarında Sigmoid ve Hiperbolik Tanjant aktivasyon fonksiyonları tüm nöronların aynı şekilde aktive olmasına neden olur. Bunun sebebi önceki bölümde bahsedilen, değerin büyüdükçe fonksiyonun 1'e yakın değerler üretmesidir. Ayrıca Sigmoid gibi fonksiyonlar doğal logaritmik taban ile işlem yaptığı için büyük sinir ağlarında öğrenme işlemi fazlaca yavaşlatmaktadır. Bu yüzden, hem öğrenme işlemi hızlandırmak hem de gradyan ölümünün önüne geçmek için bu fonksiyon daha çok tercih edilir. Relu fonksiyonu Şekil 3.19'da gösterilmiştir.



Şekil 3.19. ReLU aktivasyon fonksiyonu ve fonksiyonun türevi

3.2.4.3. Softmax fonksiyonu

Sigmoid fonksiyonu ile benzer bir yapıya sahiptir. Softmax regresyonu olarak da adlandırılan fonksiyon sınıflandırıcı olarak iyi bir performans vermektedir. Softmax regresyonu için lojistik regresyonun geliştirilmiş halidir denilebilir. Özellikle ikiden fazla sınıflandırmanın gerektiği durumlarda tercih edilmektedir. Ancak girdinin belirli bir sınıfa ait olma olasılığını 0-1 aralığında değerler üreterek belirlemektedir. Olasılıksal bir yorumlama ile aktivasyon işlemi gerçekleştirdiği için sadece çıkış katmanında kullanılması uygundur. Ara katmanlarda kullanılması olasılığı artıracak ve öğrenme işleminin kaotik veriler üzerinden devam etmesine neden olacaktır. Softmax fonksiyonu Şekil 3.20'de gösterilmiştir.



Şekil 3.20. Softmax fonksiyonunun değer üretme aralığı

3.2.4.4. İkili sınıflandırma dönüşümü (One hot encoding) yöntemi

İkili sınıflandırma yöntemi, kategorik değişkenlerin ikili olarak temsil edilmesi anlamına gelmektedir. Bu işlem ilk önce kategorik değerlerin tamsayı değerlerine eşlenmesini gerektirmektedir. Sonrasında her bir tamsayı değeri, 1 ile işaretlenmiş tamsayı indeksi dışındaki tüm değerleri sıfır olan bir ikili vektör olarak temsil eder. Örneğin Şekil 3.21’de katman sınıflandırmadan ikili sınıflandırmaya geçiş gösterilmiştir. Bu yöntemin tercih edilme sebebi, birçok makine öğrenmesi algoritmasının doğrudan kategorik veri ile çalışamamasıdır. Derin öğrenme algoritmalarında da çıkış katmanında sınıflamanın ikili sınıf dönüşümüne çevrilmesi, birbirine göre üstünlüğü bulunmayan değerlerde (örnek olarak sıcak, soğuktan sayısal olarak üstün olabilir ancak kiraz meyvesi elma meyvesine göre herhangi bir sayısal üstünlüğü yoktur.) daha başarılı sonuç vermektedir.

Label Encoding			One Hot Encoding			
Food Name	Categorical #	Calories	Apple	Chicken	Broccoli	Calories
Apple	1	95	1	0	0	95
Chicken	2	231	0	1	0	231
Broccoli	3	50	0	0	1	50

Şekil 3.21. İkili sınıflandırma yönteminin dönüşümünün gösterimi

3.3. Görüntü İşleme

Görüntü işleme ve bilgisayar görüşü üzerine çalışmalar, Optik Karakter Tanıma (Optical Character Recognition – OCR) teknolojisinin icadı ile gelişerek hız kazanmıştır. Günümüzde, otomobil, atm, cep telefonu, okul güvenlik sistemleri, savunma sistemleri, endüstri 4.0 uygulamaları ve pek çok alanda ESA mimarisi veya Haar Cascade yöntemi ile görüntü işleyebilen teknolojiler kullanılmaktadır. Tez çalışmasında obje yeri tanımlama ve segmentasyon işlemleri yapılmayacağı için görüntü işleme yöntemi olarak çok sınıflandırılmalı ESA seçilmiştir.

3.3.1. Veri Seti

Çalışmada iki farklı veri seti üzerinde sınıflandırma yapılacaktır. İlk olarak nesnenin hangi malzemeden yapıldığının tespiti için bir veri seti hazırlanmıştır. Veri seti Stanford Üniversitesi'nin açık kaynaklı olarak sunduğu ImageNET veri seti üzerinden indirilen resimlerden ve sanayide çokça kullanılan mekanizmaların resimlerinin çekilmesi ile elde edilen veriler harmanlanacaktır. Görüntü işleme yöntemlerinde, ESA'da veri setinin eğitilmesinden önce, OpenCV kütüphanesinden yardım alınarak resimler üzerinde ön hazırlık işlemleri yapılacaktır. OpenCV kütüphanesi web kamerası veya bir video üzerinde direkt olarak resimden matris dönüşümü işlemi gerçekleştirebilen bir araçtır. OpenCV ve Keras kütüphaneleri Anaconda Spyder açık kaynak kodlu derleyicide Python programlama dilinde çağırılacak ve bütün tez çalışması Python dilinde yazılacaktır.

Çizelge 3.5. Malzeme sınıflandırmada kullanılacak sınıflar, resim adedi ve etiket değerleri

Malzeme	Resim Adedi	Etiket Değeri
Çelik	200	0
Demir & Döküm	200	1
Alüminyum	200	2
Ahşap	200	3
Bakır	200	4
Pirinç	200	5
Bronz	200	6
ABS Plastik (Beyaz)	200	7
Şeffaf Plastik	200	8
Cam	200	9

Çizelge 3.5.'de gösterildiği üzere, malzeme bilgisini tanımlayacak veri seti, on farklı malzemeden 2000 adet resim içermektedir. Bu veri seti çalışmada artık Malzeme Veri Seti (MVS) olarak adlandırılacaktır.

Çizelge 3.6. Objeye sınıflandırmada kullanılacak veri setinin sınıfları, resim adedi ve etiket değerleri

Objeye Cinsi	Resim Adedi	Etiket Değeri
Araba	250	0
Sandalye	250	1
Dişli	250	2
Uçak	250	3
Tanımsız	250	4

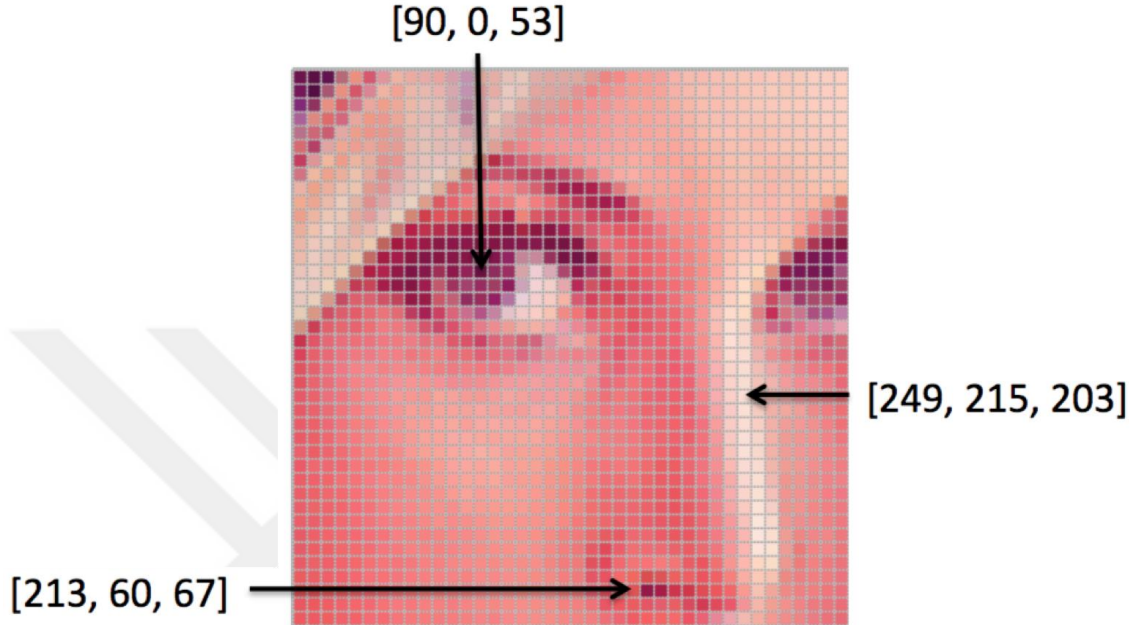
Çizelge 3.6.'da görüntü işleme ile objeye sınıflandırma için kullanılacak veri setinin detayları verilmiştir. Bu veri seti toplamda 1250 adet resim içermektedir ve 5 farklı sınıfa sahiptir. Bu veri seti çalışmada artık Objeye Veri Seti (OVS) olarak adlandırılacaktır.

The screenshot shows the ImageNet website interface. At the top, there is a search bar with the text 'glass solar' and a 'SEARCH' button. Below the search bar, it indicates '14,197,122 images, 21841 synsets indexed'. The main content area displays a search result for 'Solar dish, solar collector, solar furnace' with 1142 pictures and a 53.53% popularity percentile. A description states: 'A concave mirror that concentrates the rays of the sun; can produce high temperatures'. The page features a 'Treemap Visualization' section on the left, a grid of image thumbnails in the center, and a 'Downloads' section on the right. The grid shows various types of solar collectors and dishes. At the bottom, there is a pagination control showing 'Prev 1 2 3 4 5 6 7 8 9 10 ... 56 57 Next'.

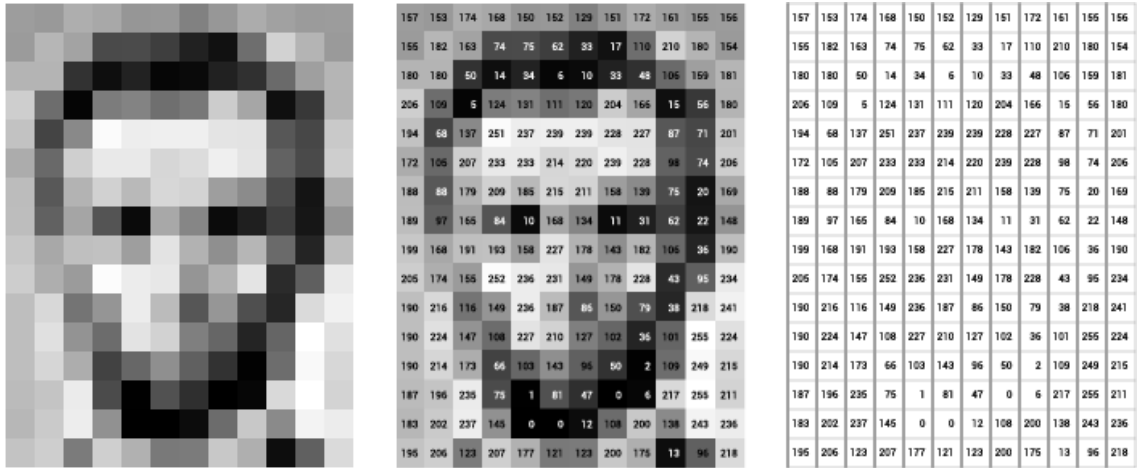
Çizelge 3.7. ImageNET web sitesinde anahtar kelime ile resim arama ekranının gösterimi

3.3.2. Veri setlerinin veri çerçevesine (data frame) dönüştürülmesi

Bilgisayarlı görme yöntemlerinde, resimler üzerinde piksel dönüştürme ve normalizasyon işlemleri sıklıkla yapılmaktadır. (Şeker,2012). Bir resim, bilgisayar ortamında kırmızı-yeşil-mavi (KYM) ana renklerinin 0 ile 255 arasındaki değerlerinden oluşan 3 katmanlı bir yapıdır. Değer 0'a yaklaştıkça piksel beyaza döner, değer 255'e yaklaştıkça ilgili renge yaklaşır.



Şekil 3.22. Renkli bir resmin üç renk katmanındaki piksel değerlerinin gösterimi



Şekil 3.23. Siyah – Beyaz bir resmin tek katmanındaki piksel değerlerinin gösterimi

<https://ai.stanford.edu/~syueung/cvweb/tutorial1.html>

Şekil 3.22.'de gösterildiği gibi renkli resimlerin her pikselinde, KYM renkleri için değer vektörü bulunmaktadır. Şekil 3.23'de gösterilen siyah beyaz bir resim tek katmanlıdır ve her pikselde tek bir değer bulunmaktadır. Bu çalışmada resimler siyah-beyaz resme dönüştürülmüştür.

Veri setinin hazırlanması için gerekli kütüphaneler belirli kod dizinleri ile çağırılmıştır. Şekil 3.24’de çağırılan kütüphanelerin ve yaptıkları görevler açıklaması ile verilmiştir.

```
import os
# Operating system import edilmesi
import cv2
# OpenCV görüntü işleme kütüphanesi import edilmesi
import numpy as np
# Matris dönüşümleri için Numpy kütüphanesi importu
import matplotlib.pyplot as plt
# Sonuçları çıktı alabilmekte için Mat Plot Library kütüphanesi importu

from sklearn.utils import shuffle
# Veri setini karıştırabilmek için random karıştırıcı
from sklearn.cross_validation import train_test_split
# Veri setini, eğitim ve test verisi olarak bölmek için kütüphane

from keras import backend as K
K.set_image_dim_ordering('th')
# Keras kütüphanesinin resim dizilim ayarları

from keras.utils import np_utils
# One Hot Encoding yöntemi için Utils çağırma işlemi
from keras.models import Sequential
# Bir keras modeli oluşturabilmek için çağırılan kütüphane
from keras.optimizers import SGD,RMSprop,adam,Adadelta
# Farklı optimizasyon algoritmalarının denenmesi için çağırılan optimizatörler
from keras.preprocessing import image
# Resim alabilmek için çağırılan işlem birimi
from keras.layers import Conv2D, MaxPooling2D
# Evrişim katmanlarının çağırılması
from keras.layers import Activation, Dropout, Flatten, Dense
# Aktivasyon fonksiyonu, dropout, düzleştirme ve YSA işlem birimlerinin çağırılması
```

Şekil 3.24. Python programlama dilinde görüntü işleme için gerekli kütüphanelerin çağırılması gösterimi

Kütüphanelerin çağırılması işlemi bitiminde resim klasörleri tek bir klasör altına tanışmış ve Ek-1’de verilen kodlar ile veri çerçevesi içine yüklenmiştir (Augspurger,2015). MVS ve OVS aynı yapay sinir ağı mimarisi ile eğitilecek fakat farklı eğitim model dosyaları oluşturulacaktır. Ek-1’de verilen YSA mimarisi modeli, MVS ve OVS için de geçerlidir.

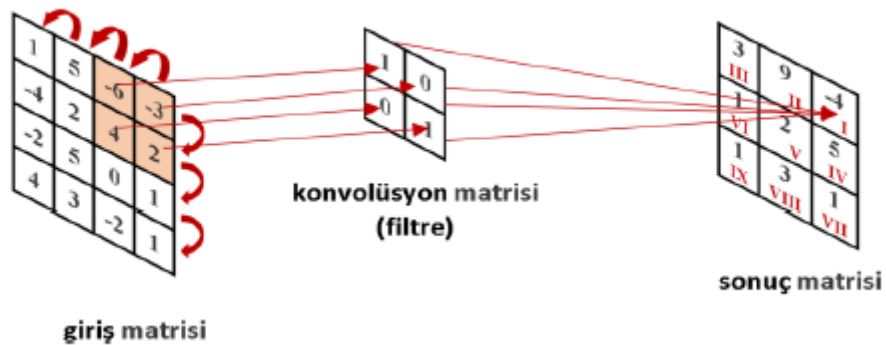
Şekil 3.25’de gösterildiği üzere, oluşturulan veri çerçevesi OVS için özetlenmiştir. “img_data” olarak belirtilen değişken 4 boyutlu bir tensördür. Resimler 128 x 128 piksel boyutuna indirgenerek veri çerçevesine yazılmıştır. Veri çerçevesinin 4 boyutlu olmasının sebebi, birinci boyutun resim adedi, ikinci boyutun kanal sayısı, üçüncü boyutun yatay piksel adedi ve dördüncü boyutun ise dikey piksel adedini içermesindedir. Veri çerçevesi eğitilirken, resimlerin YSA mimarisine uygun olarak alınabilmesi için fazladan bir boyut daha eklenecektir.

Name	Type	Size	Value
data_dir_list	list	5	['car', 'chair', 'gear', 'plane', 'undefined']
data_path	str	1	S:\fea4.0\cnn_our\data
dataset	str	1	undefined
img	str	1	undefined(99).jpg
img_cols	int	1	128
img_data	float32	(1500, 1, 128, 128)	[[[[[0.2901961 0.30980393 0.32941177 ... 0.5176471 0.5... 0.50 ...
img_data_list	list	1500	[Numpy array, Numpy array, Numpy array, Numpy array, Numpy array, Nump ...
img_list	list	300	['undefined(1).jpg', 'undefined(10).jpg', 'undefined(100).jpg', 'undef ...
img_rows	int	1	128
input_img	uint8	(480, 640)	[[59 58 58 ... 15 14 13] [57 57 57 ... 12 12 11]
input_img_resize	uint8	(128, 128)	[[57 57 60 ... 4 10 14] [58 59 60 ... 5 7 10]
num_channel	int	1	1
num_classes	int	1	5

Şekil 3.25. Ek-1’de verilen kodlar ile resimler veri çerçevesine yazdırılmıştır.

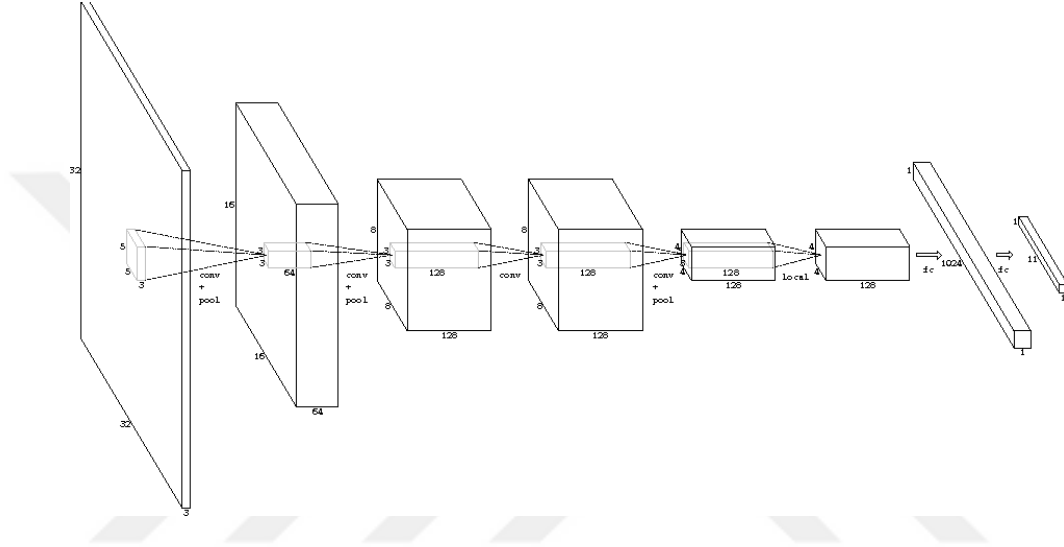
3.3.3. Evrişimli sinir ağı mimarisinin oluşturulması

ESA’larda filtre ve ortaklamalama operatörleri bulunmaktadır. Filtreleme işlemleri, resim verisi alındıktan sonra matrisler üzerinde yapılmaktadır. Ancak filtre işlemlerinden sonra resim piksellerini içeren kare matrisler, düzleştirilerek vektör haline getirildikten sonra YSA’nın giriş katmanına bağlanabilir. Bu sebeple şekil 3.24’de gösterilen “flatten” işlem birimi çağırılmıştır.



Şekil 3.26. Örnek bir konvolüsyon filtresi işlemi gösterimi

Evrişim filtreleri resmin baskın özelliklerini ortaya çıkardığı gibi, resim boyutunu da küçültür. Ancak bu küçültme işlemi filtreler sayesinde özellik kaybı yaşanmadan gerçekleşir. 128 x 128 boyutundaki bir piksel matrisi düzleştirildiğinde 16384 x 1 boyutlu bir vektöre dönüşür. Böyle bir büyüklükteki bir vektörü YSA'ya girdi olarak vermek öğrenme süresini fazlaca uzatacaktır. Çalışmada 128 x 128 piksellik resim matrisi filtreler ile 80 x 80 kare matrise dönüştürülecektir. OVS ve MVS için inşa edilecek ESA mimarisinin gösterimi Şekil 3.27'de verilmiştir.



Şekil 3.27. Görüntü işleme için oluşturulan ESA mimarisinin şematik gösterimi

Şekil 3.28'de gösterildiği üzere mimaride birbirine bağlı iki adet konvolüsyon filtresi bulunmaktadır. Konvolüsyon filtresi maksimum ortaklamalama filtresinden geçirildikten sonra düzleştirilerek bir vektör haline getirilmektedir. Düzleştirilen vektör 16 nöronlu bir gizli katmana bağlanmıştır. OVS'de 5 adet farklı sınıf olduğu için, gizli katman ReLU aktivasyonundan geçirildikten sonra 5 nöronlu çıkış katmanına girdi olarak verilmektedir. Çıkış katmanı karar katmanı olduğu için aktivasyon fonksiyonu olarak softmax seçilmiştir. Bu filtre ve katmanlar toplamda 122133 adet eğitilebilir parametre içermektedir. Parametrenin az oluşu öğrenme işleminin yetersiz kılacağı gibi parametrenin çok fazla olması da öğrenme işlemini yavaşlatacağı gibi ezberleme (overfitting) problemine sebep olacağı için kararlı bir sayıda tutulmalıdır. Ayrıca yine ezberleme probleminin önüne geçmek için ESA mimarisinin iki adet dropout katmanı eklenmiştir. Dropout birimi, öğrenme esnasındaki her döngüde, rastgele nöronların yarısını pasifleştirerek modelin genelleme yeteneği kazanmasını sağlamaktadır. Dropout katmanı sadece öğrenme yaparken kullanılmaktadır. Model test edilirken

kullanılmamaktadır. Şekil 3.29’da ESA’nın 80 devir eğitildikten sonraki doğruluk değerleri verilmiştir. Eğitim, bilgisayar hafızasının zorlanmaması için 20 devirlik bölümler ile gerçekleştirilmiştir.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 128, 128)	1184
activation_1 (Activation)	(None, 32, 128, 128)	0
conv2d_2 (Conv2D)	(None, 16, 123, 123)	18448
activation_2 (Activation)	(None, 16, 123, 123)	0
max_pooling2d_1 (MaxPooling2D)	(None, 16, 20, 20)	0
dropout_1 (Dropout)	(None, 16, 20, 20)	0
flatten_1 (Flatten)	(None, 6400)	0
dense_1 (Dense)	(None, 16)	102416
activation_3 (Activation)	(None, 16)	0
dropout_2 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 5)	85
activation_4 (Activation)	(None, 5)	0
Total params: 122,133		
Trainable params: 122,133		
Non-trainable params: 0		

Şekil 3.28. Keras kütüphanesi ile oluşturulan ESA mimarisi özeti

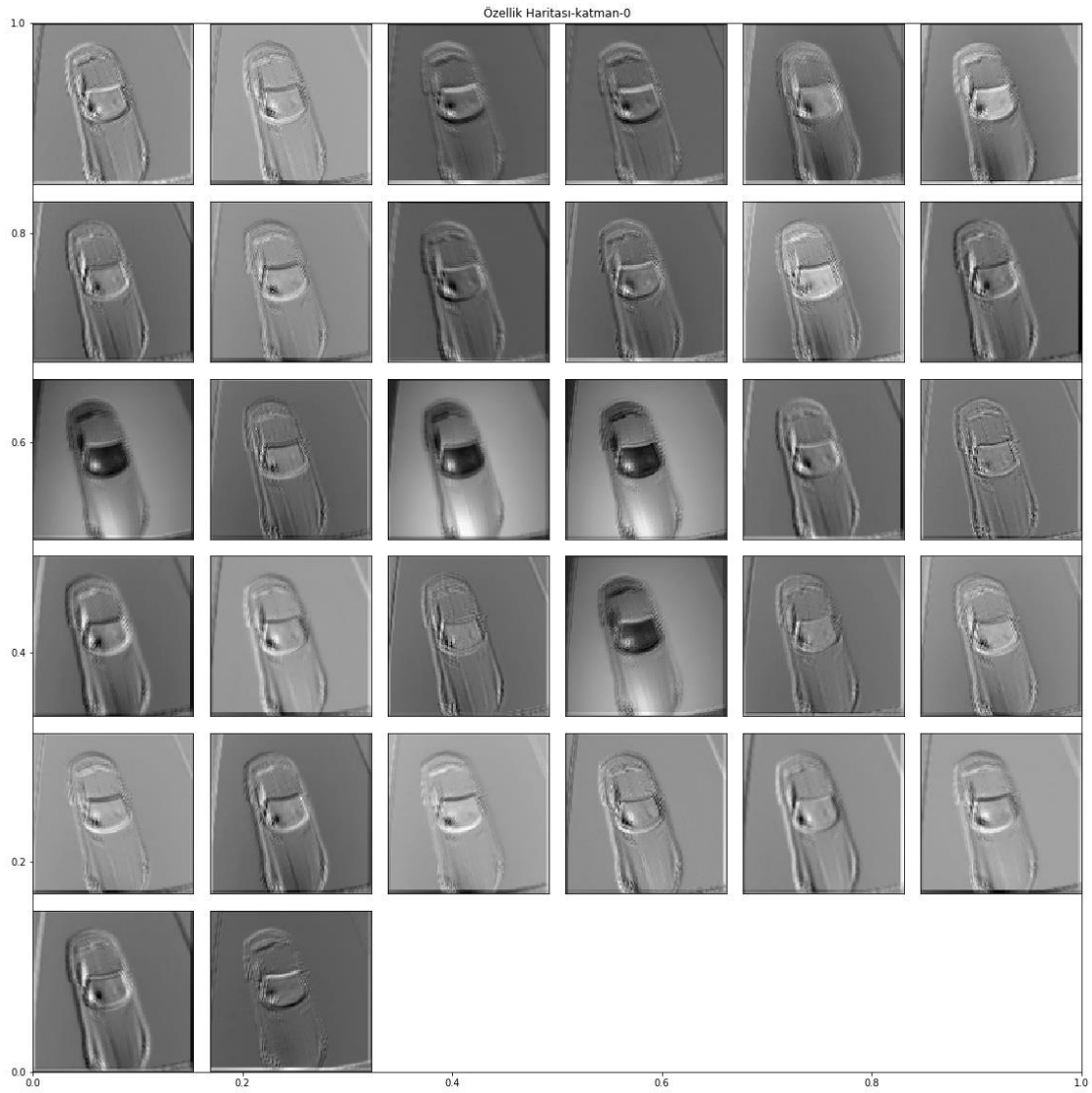
```
Epoch 18/20
1200/1200 [=====] - 8s 7ms/step - loss: 0.3270 - acc: 0.8767 - val_loss: 0.1081
- val_acc: 0.9733
Epoch 19/20
1200/1200 [=====] - 8s 7ms/step - loss: 0.3097 - acc: 0.8758 - val_loss: 0.0977
- val_acc: 0.9767
Epoch 20/20
1200/1200 [=====] - 8s 7ms/step - loss: 0.3296 - acc: 0.8775 - val_loss: 0.0993
- val_acc: 0.9833
Out[7]: <keras.callbacks.History at 0x1e602d79710>
```

Şekil 3.29. Keras kütüphanesi ile OVS'nin ESA'ya eğitilmesi, 80 devir (epochs) süren eğitimde validasyon doğruluğu %98.33 değerine ulaşmıştır.

Konvolüsyon filtrelerinin işlevinin daha iyi anlaşılabilmesi için filtrelerden çıkan sonuçlar görselleştirilebilir. Ek-1’de verilen kodlar ile evrişim filtreleri çıktılarını görüntülemek mümkündür. Katmanların birbirine çok fazla benzemesi, filtre sayısını azaltabileceğimiz anlamına gelmektedir. Şekil 3.30’da gösterilen veri setindeki bir resmin evrişim katmanlarından geçtikten sonra özelliklerinin belirginleştirilmesi Şekil 3.31’de gösterilmiştir.



Şekil 3.30. Konvolüsyon katmanlarının gösterileceği resmin orijinal hali

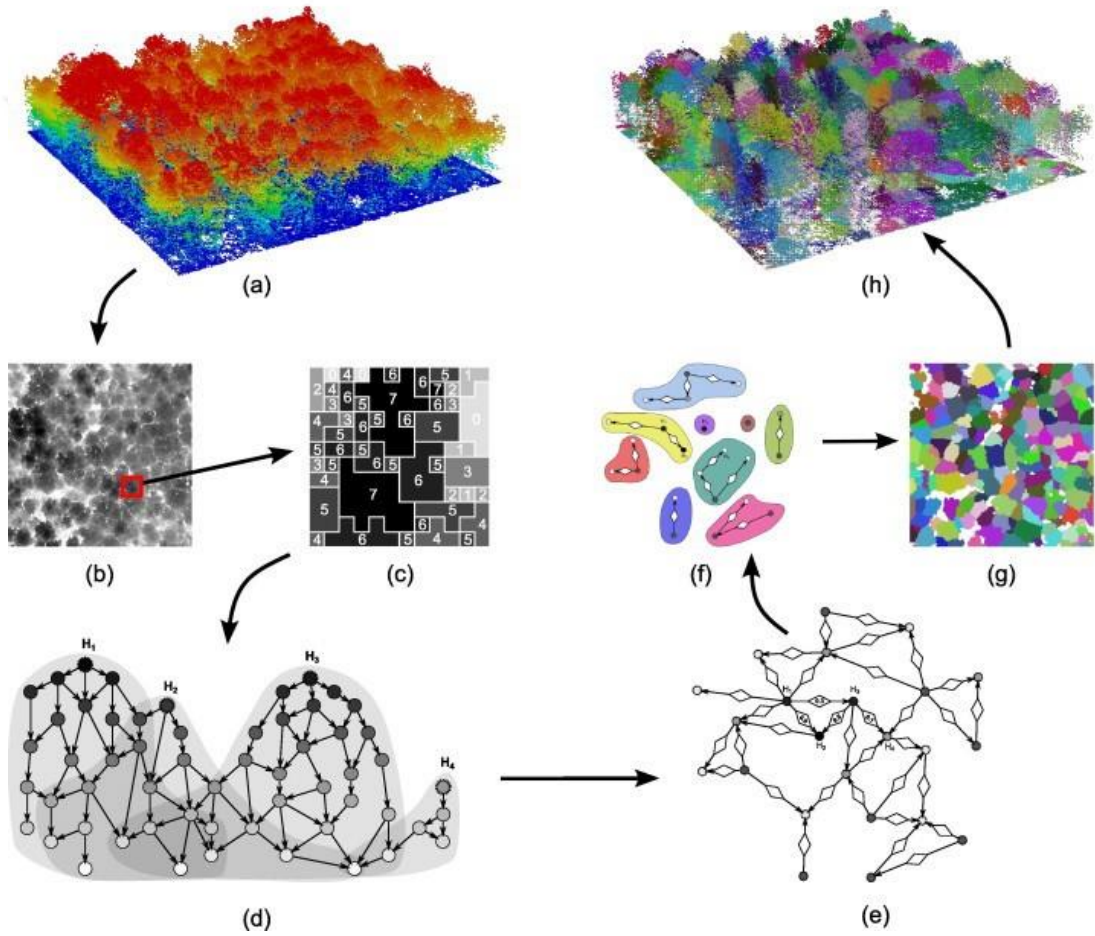


Şekil 3.31. Konvolüsyon katmanlarının görselleştirilmesi

Model başarı ile eğitilmiştir. Train esnasında model kendisini valide etmektedir. Ancak model daha önce hiç görmediği veriler ile bölüm 4’de tekrar test edilecektir.

3.4. Üç Boyutlu Cisim Sınıflandırma

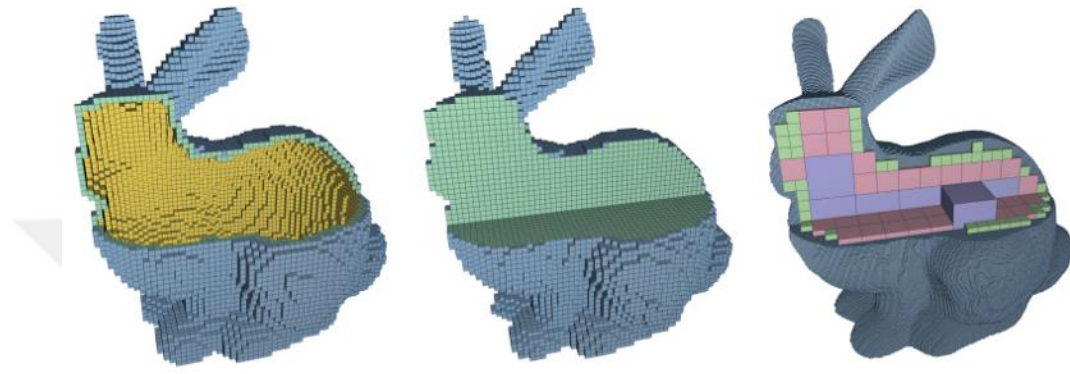
Tez çalışmasında üç boyutlu cisim sınıflandırma belirli dönüşümler ile gerçekleştirilecektir. Görüntü işleme yöntemlerinde resimler OpenCV kütüphaneleri ile matrislere dönüştürülebilmektedir. Ancak bilgisayar ortamındaki üç boyutlu geometrilerin matrislere dönüştürülebilmesi için önce hacimsel olarak piksellenebilmesi gerekmektedir. Şekil 3.32’de gösterilen çalışmada bir arazideki bitki örtüsü LIDAR ile taranmış, hacimsel pikselleri elde edilmiş ve piksel verisi hiyerarşik derin öğrenme algoritmaları ile segmente edilmiştir.



Şekil 3.32. Lidar verisinin segmente (bölütleme) edilmesi

3.4.1. Hacimsel pikselleme (Voxelization)

Hacimsel pikselleme katı geometriyi veya yüzeyi, uzayda işgal ettiği alana göre bölütleyerek temsil etmektedir. Voxelization; nesnelerin voxel (volümetrik piksel) kümeleri olarak sunulmasını sağlayan bir süreçtir (Şekil 3.33). Tam olarak geometrik nesnelerin nesnenin doğasına/yapısına en yakınsayacak şekilde, sürekli geometrik temsillerden bir voxel kümesi haline dönüştürülmesi safhası olarak tanımlanmıştır.



Şekil 3.33. Bir katı geometrinin farklı yöntemler ile hacimsel piksellenmesi

3.4.1.1. İkili hacimsel pikselleme dönüşümü

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 alınacaktır. Binvox.exe programı Windows komut satırı üzerinden çalışmaktadır ve kullanmak için yazılması gereken kodlar program içinde açıklanmaktadır.

3.4.2. Üç boyutlu obje veri dosyası temini

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. Üç boyutlu evrişimli sinir ağı bu çalışmada 3B-ESA olarak adlandırılacaktır. Diğer yandan Princeton üniversitesi modelnet web sitesinde, veri seti ile çalışan tez yazarlarının ve takımların elde ettikleri başarı liste olarak verilmiştir.

```

C:\Users\devel>binvox.exe
--- [binvox] mesh voxelizer, version 1.26, build #644 on 2017/10/22 15:19:22
--- written by Patrick Min, 2004-2017

Error: missing model filename

Usage: binvox [-d <voxel dimension>] [-t <voxel file type>] [-c] [-v] <model filespec>
  -license: show software license
  -d: specify voxel grid size (default 256, max 1024)(no max when using -e)
  -t: specify voxel file type (default binvox, also supported: hips, mira, vtk, raw, schematic, msh)
  -c: z-buffer based carving method only
  -dc: dilated carving, stop carving 1 voxel before intersection
  -v: z-buffer based parity voting method only (default is both -c and -v)
  -e: exact voxelization (any voxel with part of a triangle gets set)(does not use graphics card)
Additional parameters:
  -bb <minx> <miny> <minz> <maxx> <maxy> <maxz>: force a different input model bounding box
  -ri: remove internal voxels
  -cb: center model inside unit cube
  -rotx: rotate object 90 degrees ccw around x-axis before voxelizing
  -rotz: rotate object 90 degrees cw around z-axis before voxelizing
  both -rotx and -rotz can be used multiple times
  -aw: also render the model in wireframe (helps with thin parts)
  -fit: only write the voxels in the voxel bounding box
  -bi <id>: when converting to schematic, use block ID <id>
  -mb: when converting using -e from .obj to schematic, parse block ID from material spec 'usemtl blockid_<id>' (ids 1-255 only)
  -down: downsample voxels by a factor of 2 in each dimension (can be used multiple times)
  -dmin <nrx>: when downsampling, destination voxel is on if >= <nrx> source voxels are (default 4)
Supported 3D model file formats:
  VRML V2.0: almost fully supported
  UG, OBJ, OFF, DXF, XGL, POV, BREP, PLY, JOT: only polygons supported
Example:
binvox -c -d 200 -t mira plane.wrl

```

Şekil 3.34. Binvox.exe programında dosya dönüşümü işleminin açıklanması

Çizelge 3.8. ModelNET veri kütüphanesi ile yapılan çalışmalar ve doğruluk oranları

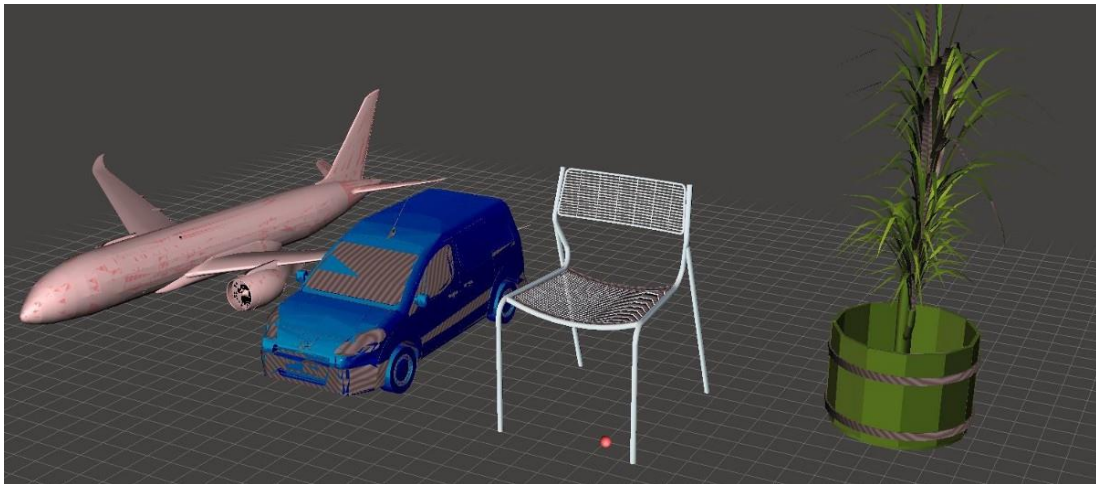
Algoritma	ModelNET40	ModelNET10
RS-CNN	93.60%	-
LP-3DCNN	92.10%	94.40%
LDGCNN	92.90%	-
Primitive-GAN	86.40%	92.20%
3DCapsule	92.70%	94.70%
3D2SeqViews	93.40%	94.71%
OrthographicNet	-	88.56%
Ma et al.	91.05%	95.29%
MLVCNN	94.16%	-
iMHL	97.16%	-
HGNN	96.60%	-
SPNet	92.63%	97.25%
MHBN	94.7	95
VIPGAN	91.98	94.05
Point2Sequence	92.6	95.3
Triplet-Center Loss	-	-
PVNet	93.20%	-
GVCNN	93.10%	-
MLH-MV	93.11%	94.80%
MVCNN-New	95.00%	

SeqViews2SeqLabels	93.40%	94.82%
G3DNet	91.13%	93.10%
VSL	84.50%	91.00%
3D-CapsNets	82.73%	93.08%
KCNet	91.00%	94.40%
FoldingNet	88.40%	94.40%
binVoxNetPlus	85.47%	92.32%
DeepSets	90.30%	
3D-DescriptorNet		92.40%
SO-Net	93.40%	95.70%
Minto et al.	89.30%	93.60%
RotationNet	97.37%	98.46%
LonchaNet		94.37
Achlioptas et al.	84.50%	95.40%
PANORAMA-ENN	95.56%	96.85%
3D-A-Nets	90.50%	
Soltani et al.	82.10%	
Arvind et al.	86.50%	
LonchaNet		94.37%
3DmFV-Net	91.60%	95.20%
Zanuttigh and Minto	87.80%	91.50%
Wang et al.	93.80%	
ECC	83.20%	90.00%
PANORAMA-NN	90.70%	91.10%
MVCNN-MultiRes	91.40%	
FPNN	88.40%	
PointNet	89.20%	
Klokov and Lempitsky	91.80%	94.00%
LightNet	88.93%	93.94%
Xu and Todorovic	81.26%	88.00%
Geometry Image	83.90%	88.40%
Set-convolution	90%	
PointNet		77.60%
3D-GAN	83.30%	91.00%

VRN Ensemble	95.54%	97.14%
ORION		93.80%
FusionNet	90.80%	93.11%
Pairwise	90.70%	92.80%
MVCNN	90.10%	
GIFT	83.10%	92.35%
VoxNet	83%	92%
DeepPano	77.63%	85.45%
3DShapeNets	77%	83.50%

3.4.2.1. OFF dosya formatı

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 dönüştürülmüştür. ModelNET sitesinden indirilen OFF dosyalarının uygunluğu Autodesk Meshmixer ücretsiz yazılım ile görüntülenebilmektedir (Şekil 3.35).



Şekil 3.35. Autodesk Meshmixer ile görselleştirilen OFF dosyaları.

3.4.3. Eğitimde kullanılacak veri dosyaları

3 boyutlu parçaların eğitiminde 4 farklı sınıf için 300'er adet dosyadan dönüştürülen veriler kullanılacaktır. Çizelge 3.9'da kullanılacak sınıflar ve sınıf numaraları belirtilmiştir. Üç boyutlu veri seti sınıfları ikili dönüştürme yöntemi ile sınıflandırılacaktır.

Çizelge 3.9. Eğitimde kullanılacak dosya çeşitleri ve sınıfları

Veri Tipi	Obje Sınıfı	Sınıf No	Veri Kaynağı	Veri Adedi	Test Oranı
OFF dosyası	Uçak çeşitleri	1	ModelNET	300	%20
OFF dosyası	Otomobil çeşitleri	2	ModelNET	300	%20
OFF dosyası	Tekli sandalye çeşitleri	3	ModelNET	300	%20
OFF dosyası	Saksılı ev bitkisi çeşitleri	4	ModelNET	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 kazanacağı düşünülmektedir. Tez çalışmasında kullanılacak olan veri seti Katı Veri Seti (KVS) olarak anılacaktır. Bilgisayarda bulunan diğer dosya türlerinden OFF formatına dönüşüm için gerekli kodları Ek-2'de verilmiştir.

3.4.4. OFF dosyalarının hacimsel piksel dosyalarına çevrilmesi

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, kod satırları arasındaki geçiş de dahil olmak üzere 102 dakika sürmüştür. Program algoritması Ek-3'de verilmiştir. Komut satırı, python çekirdeğinde her döngüde otomatik olarak çağırılmaktadır.

3.4.5. Bivox dosyalarının elde edilmesi ve görselleştirme

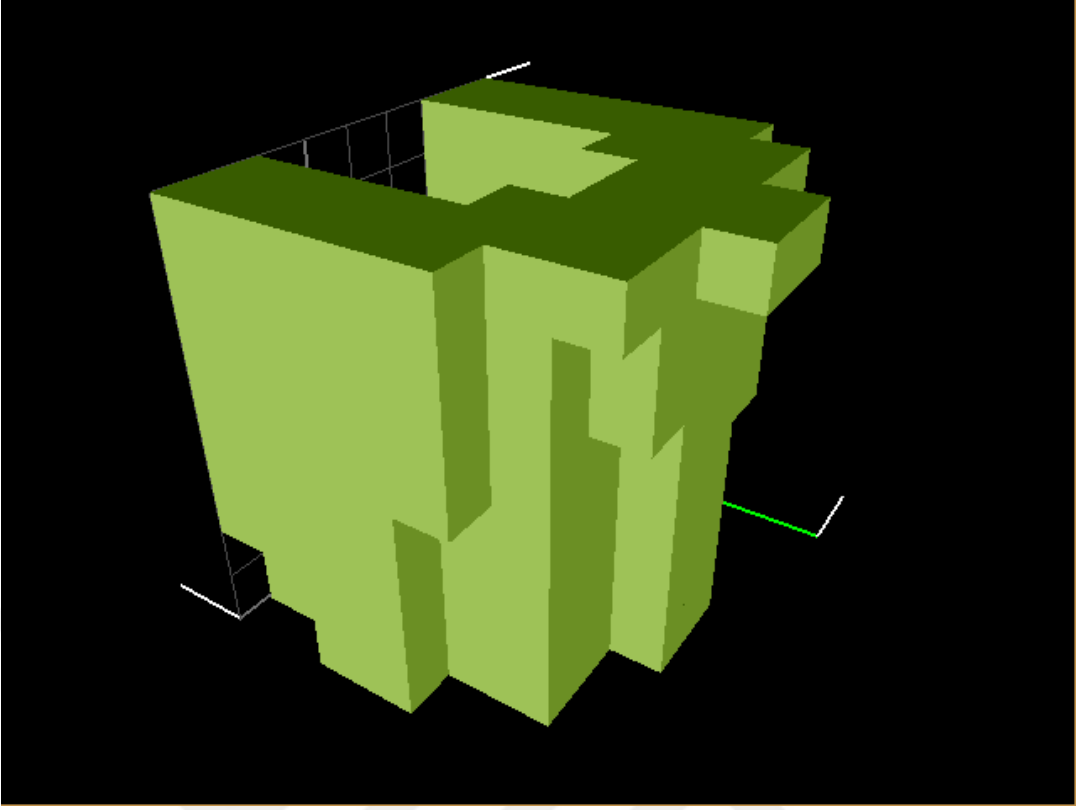
Bivox dosyaları, 3 boyutlu veri setindeki 1200 ayrı obje için tek tek oluşturulmuştur. Programında doğru çalışılığının kontrolü için bivox dosyalarından rastgele seçilen otuz adet dosya viewvox.exe yazılımı ile görselleştirilmiştir. Şekil 3.36.'da dosyaların görselleştirilmiş halleri gösterilmektedir.

Farklı kenar uzunluğu birimlerinin dosya pikselleme ve pikselleme sürelerinin değişimleri arasındaki farklar bulunmuştur.

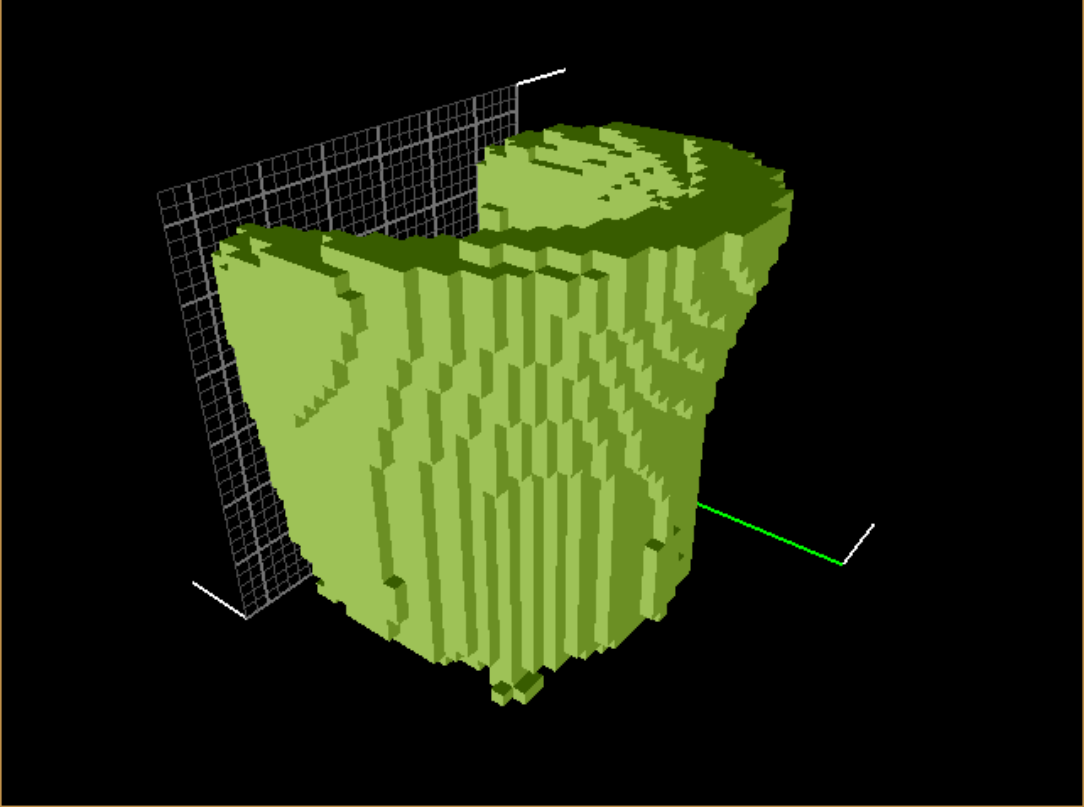
Çizelge 3.10. Farklı kübik değerlerde dosyaların karşılaştırılması

Pikselleme derecesi	Toplam piksel adedi	Piksel dosyası boyutu (kb)	Pikselleme süresi (sn)	Katı uyuşma 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

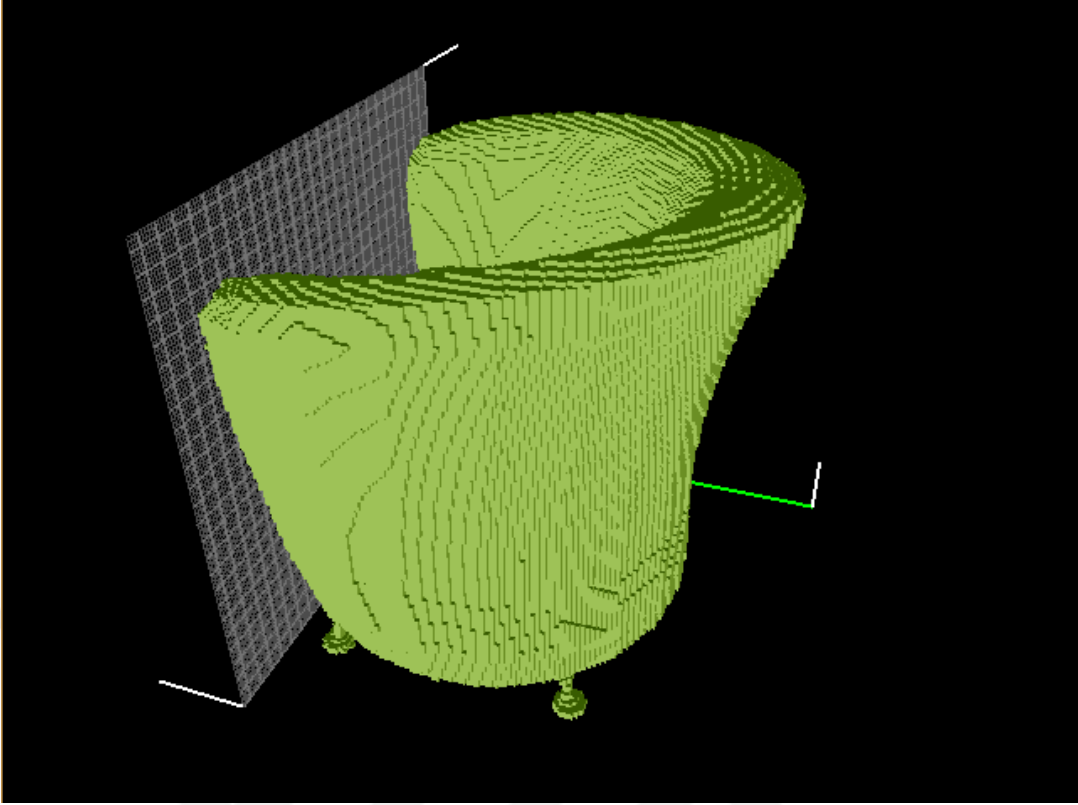
Çalışmada farklı boyutlarda hacimsel piksellenen dosyaların işlem süreleri ve katı uygunluk oranları saptanmıştır. Aynı boyutlarda piksellenen dosyalar sıra ile verilecektir. Dosyaların katı uygunluk oranları Autodesk Meshmixer programı ile analiz edilmiştir.



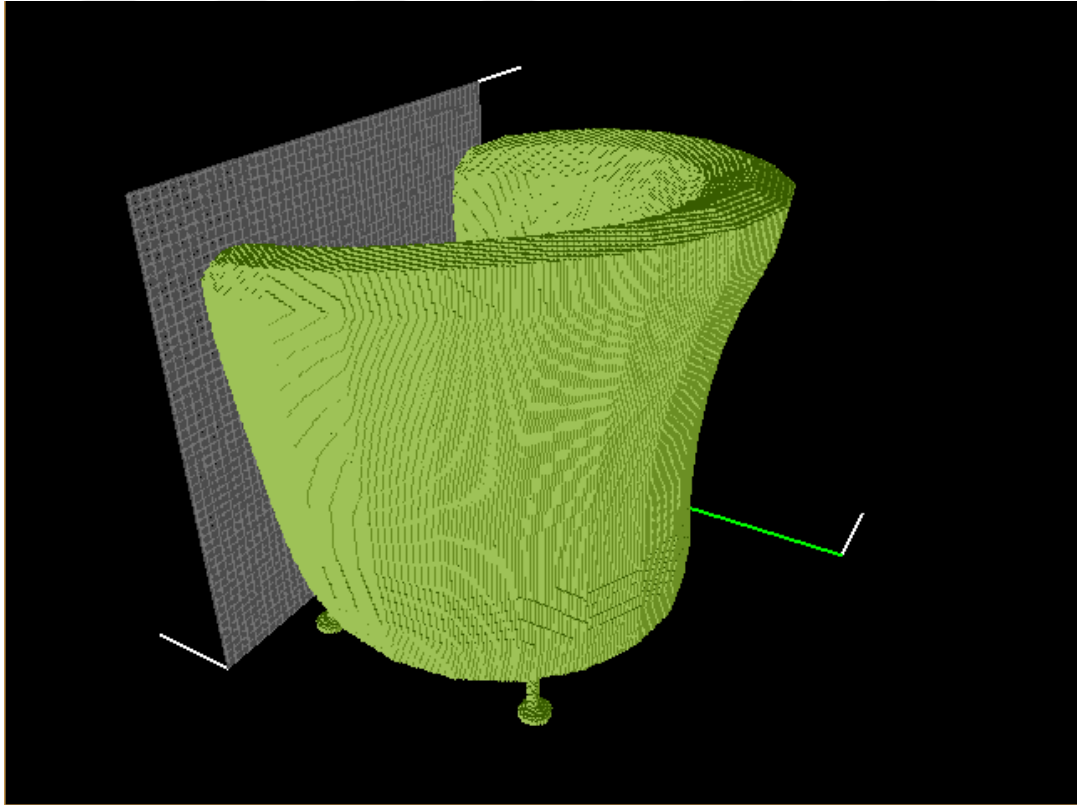
Şekil 3.36. 8 x 8 x 8 boyutlarında hacimsel piksellenen geometrinin gösterimi



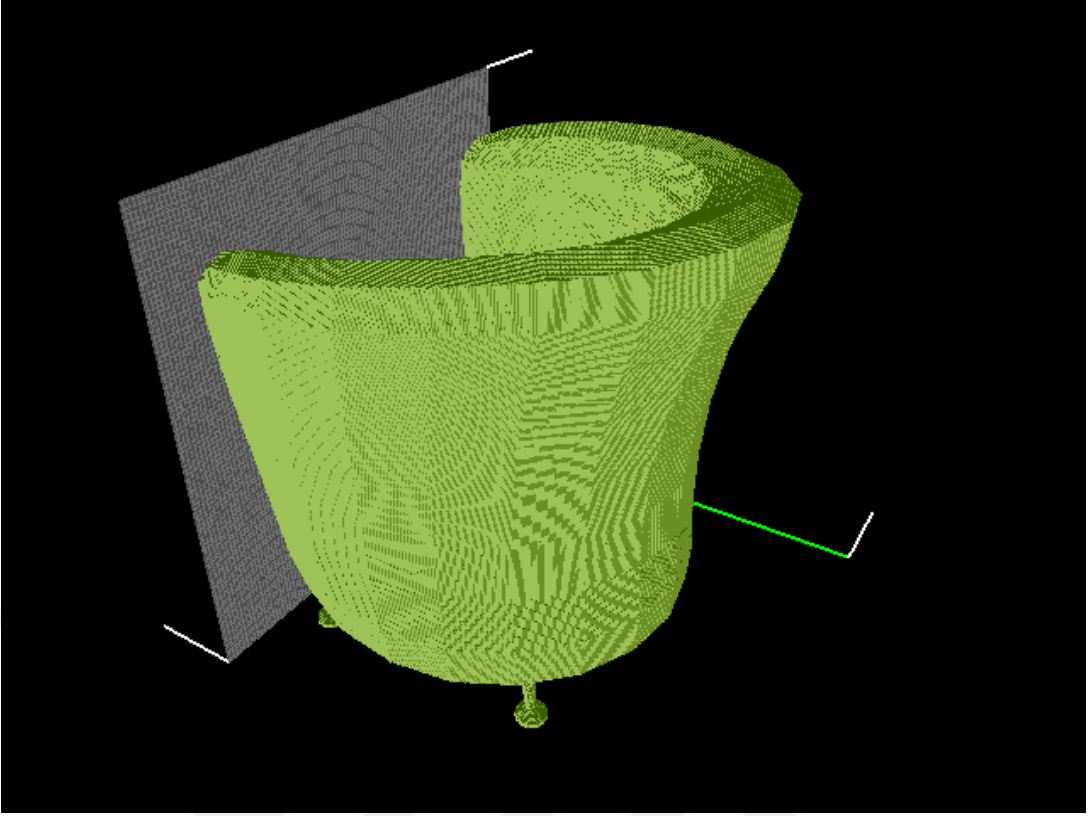
Şekil 3.37. 32 x 32 x 32 boyutlarında hacimsel piksellenen geometrinin gösterimi



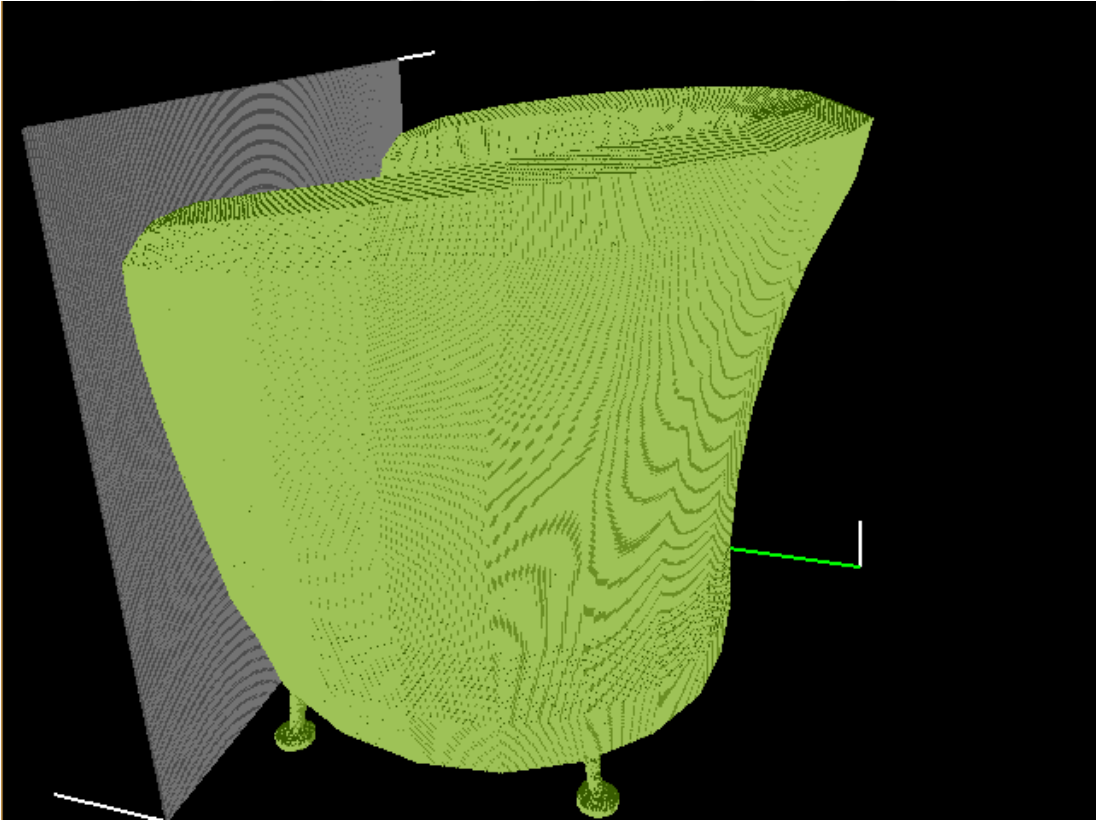
Şekil 3.38. 128 x 128 x 128 boyutlarında hacimsel piksellenen geometrinin gösterimi



Şekil 3.39. 256 x 256 x 256 boyutlarında hacimsel piksellenen geometrinin gösterimi

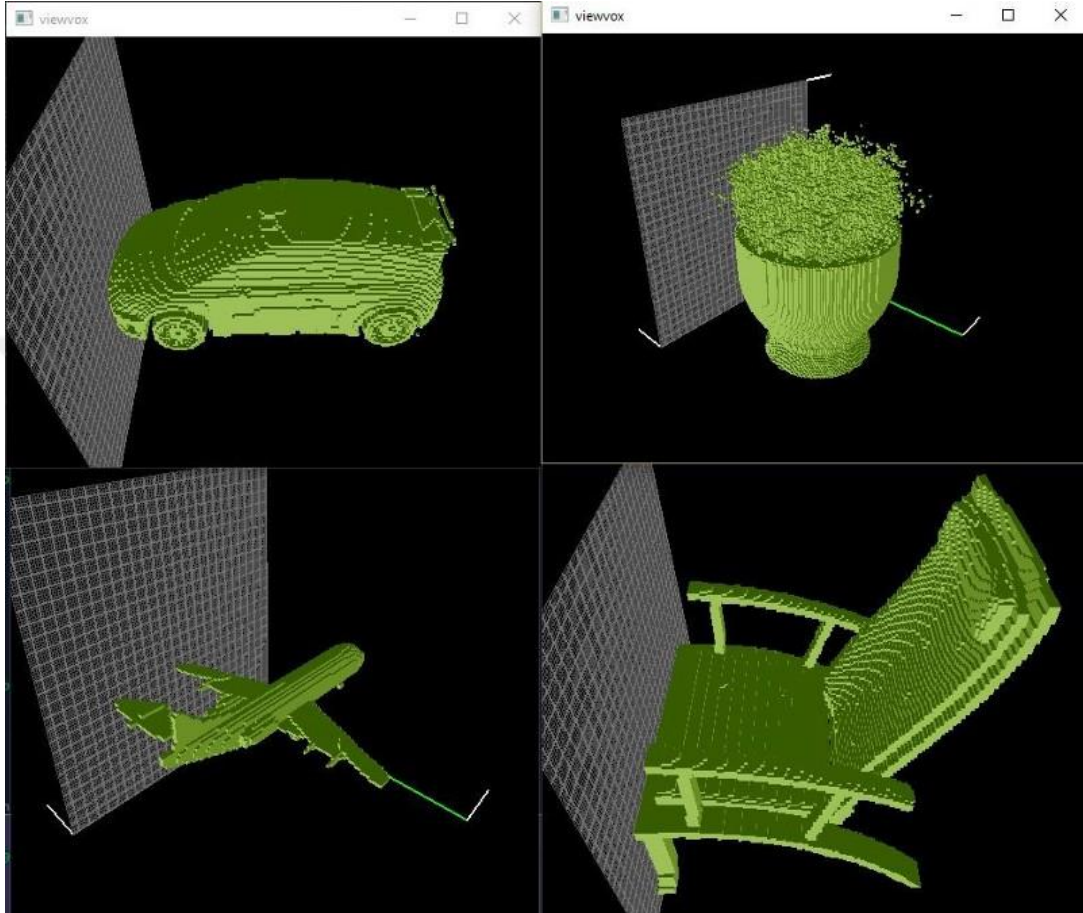


Şekil 3.40. 512 x 512 x 512 boyutlarında hacimsel piksellenen geometrinin gösterimi



Şekil 3.41. 1024 x 1024 x 1024 boyutlarında hacimsel piksellenen geometrinin gösterimi

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.



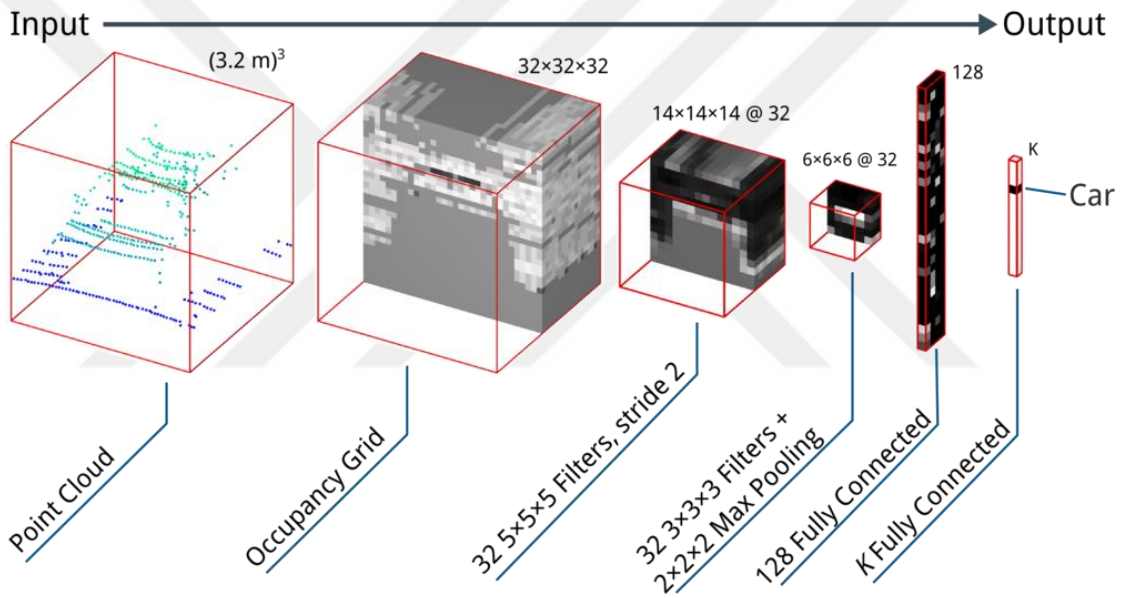
Şekil 3.42. Viewvox yazılımı ile OFF formatından 128 birim küplük binvox dosyasına çevirilen verilerin rastgele seçim yapılarak görselleştirilmesi ve doğrulanması

3.4.6. Binvox dosyalarının veri seti matrisi içine yazdırılması

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-4’de kodları verilen algoritmada, 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 çevirilmiştir.

3.4.7. Üç boyutlu evrişimli sinir ağı mimarisi oluşturulması

Bölüm 3.3.3’de inşa edilen sinir ağı mimarisi iki boyutlu resim dosyaları içindir. Tezin bu bölümünde, görüntü işleme yöntemine benzer şekilde bir Evrişimsel sinir ağı inşa edilmiştir. Ancak yapısı ve görevi gereği filtre ve kodlama farklılıkları bulunmaktadır. Şekil 3.43’de verilen VoxNET mimarisi, Theano kütüphanesi üzerine kurulmuştur. Theano kütüphanesi oldukça zor bir yapay sinir ağı mimarisi oluşturma kütüphanesidir. Ek-5’de mimari kodları verildiği üzere, üç boyutlu evrişimli yapay sinir ağı Keras kütüphanesinde kodlanmıştır ve kod bloklarının okunması oldukça kolay olduğu gibi mimaride değişiklik yapıp farklı varyansları denemek de oldukça elverişlidir.



Şekil 3.43. VoxNET mimarisinin şematik gösterimi. Voxnet mimarisi voxel veriyi 32 x 32 x 32 boyutlarında YSA’ya beslemektedir (Maturana,2015)

Şekil 3.43.’de gösterilen VoxNET mimarisi, veriyi 32 x 32 x 32 boyutlarında almakta ve iki konvolüsyon filtresi ve bir de maksimum ortaklama filtresi ile 128 x 1 boyutlarında vektöre dönüştürmektedir. Bu vektör tam bağlantılı sinir ağı ile sınıflandırma katmanına gönderilmekte ve bu katmanda bir tahmin yapılmaktadır.

Tez çalışmasında, üç boyutlu obje sınıflandırmada, konvolüsyon katmanlarının etkisi de incelenmiştir. Kurulan iki ayrı mimari, konvolüsyonlu ve konvolüsyonsuz olarak eğitilmiş ve doğruluk oranları saptanmıştır. Doğruluk oranları bölüm 4’de verilecektir.

3.4.7.1. Konvolüsyon filtreli sinir ağı mimarisi

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_l2 (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 3.44. Keras kütüphanesi ile oluşturulan ve özeti görüntülenen konvolüsyon filtreli YSA'nın gösterini

Şekil 3.44'de 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 ortaklama ve bir adet ortalama ortaklama 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.

3.4.7.2. Konvolüsyon filtresiz sinir ağı mimarisi

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 3.45. Konvolüsyon filtresi bulunmayan sinir ağı mimarisi özet gösterimi

Şekil 3.45'de özet görünümü gösterildiği üzere, konvolüsyon filtresi bulunmayan YSA, 128 x 128 x 128 boyutundaki tensörü, konvolüsyonlu modeldeki aynı ortalama filtrelerinden geçirmekte ve 343 x 1 boyutundaki vektör elde edilmektedir. Konvolüsyon filtreleri veriyi küçülttüğü için bu modelde giriş vektörü

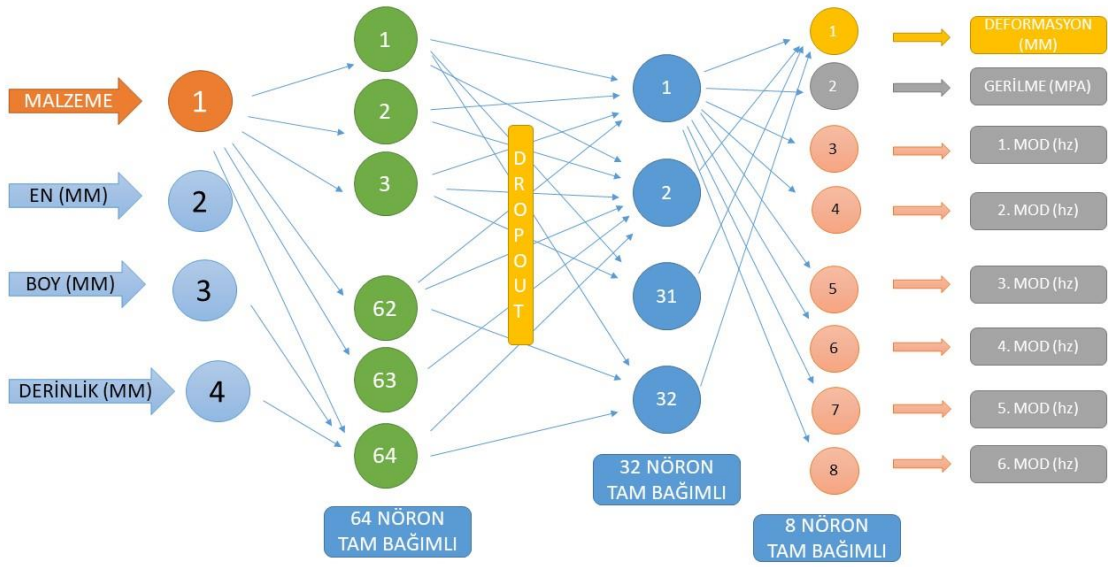
daha büyüktür. Konvolüsyon filtresi olmayan YSA'da gizli katmanlar aynı şekilde inşa edilmiş ve eğitilebilir parametre sayısı 11,604 olarak gözlemlenmiştir.

3.5. Sonlu Elemanlar Analizi Regresyon Sinir Ağı

Sonlu elemanlar analizi farklı malzeme ve farklı boyutlar için yapılan simülasyonların sonuçları, özelleştirilmiş bir yapay sinir ağında eğitilmiştir. YSA, görüntü işleme modellerinden farklı olarak çok çıkışlı ve çok nöron gruplu olarak tasarlanmıştır. Bu yönü ile sınıflandırma değil lojistik regresyon sonucu vermektedir. YSA'da gerilme ve deformasyon değerleri farklı bir çıkış katman grubunda, modelin altı adet ilk doğal frekansı ise farklı bir nöron grubunda eğitilecektir. Bu yöntemin izlenmesinin sebebi, farklı gruplar için farklı maliyet değerleri alınabilecek ve modelde olası değişiklikler daha rahat yapılabilecektir.

3.5.1. SEM için YSA modeli

Sonlu elemanlar analizi sonuçlarının öğrenilmesi için kurulan yapay sinir ağı Şekil 3.46.'da gösterildiği üzere, ilk katmanda 4 adet giriş verisi için nöron bulunmaktadır. Malzemeler, kendi arasında büyüklük skalasına sahip olacağı için bu sınıfların ikili sınıflandırma yöntemine göre kodlanmasına gerek görülmemiştir. Malzeme sıralaması çizelge 3.4'de verilen malzemelerin rijitlik sırası ile kodlanacaktır. Çelik için 1, alüminyum için 2 ve ABS plastik için 3 değer atanacaktır. Giriş katmanında da malzeme bilgisini alan nöron ile boyutsal verileri alan diğer 3 nöron farklı gruplar ile eğitilecek ve farklı maliyet değerleri hesaplatılacaktır. Malzeme değerleri ve boyutsal değerler kendi grupları içinde 0 ile 1 arasına normalizasyon yapılacak yapay sinir ağına beslenecektir. 4 adet nöron ikinci tam bağımlı katmanda 64 adet nörona bağlıdır ve aktivasyon fonksiyonu "ReLU" olarak belirlenmiştir. Üçüncü katmandan önce %50 nöron kapatma işlemi yapılacak "Dropout" fonksiyonu bulunmaktadır. Üçüncü katmanda aktivasyonu "ReLU" olan 32 adet nöron bulunmaktadır. Çıkış katmanı 8 adet nörondan oluşmaktadır ve aktivasyon fonksiyonu "Sigmoid" olarak belirlenmiştir.



Şekil 3.46. SEM analizi için inşa edilen YSA modelinin şematik gösterimi

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

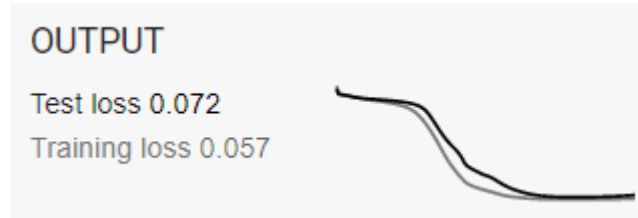
Tez çalışmasında, görüntü işleme, katı parça sınıflandırma ve statik, modal analizlerin sonuçlarının öğrenilmesi için çalışmalar yapılmıştır. Çalışmalar sonucunda, yapılan eğitimlerin eğitim süreleri, validasyon doğruluk oranları, maliyet değerleri, devir adedi gibi sonuçlar karşılaştırılacaktır.

Şekil 4.1’de MVS’nin, Şekil 4.2’de ise OVS’nin, Tensorboard Web kütüphanesi ile eğitim anında görselleştirilmiş eğitim ve test maliyet (kayıp) değerleri gösterilmektedir. MVS’nin eğitiminde kayıp değerlerinin süreksiz değişmesi, OVS’de ise eğitimin yumuşak bir ivme ile devam etmesi, MVS veri setinde malzeme yüzeylerinin birbirine benzemesi, çelik ve alüminyum gibi parçaların yansımalarının benzer olması sebebi ile yapay sinir ağının kayıp değerlerinin lineer bir doğru üzerine uzun sürmüştür. OVS veri setinde objelerin belirgin hatları ve belirgin özellikleri olduğu için eğitim daha yumuşak gerçekleşmiştir ve kısa sürede kabul edilebilir değerlere ulaşmıştır. OVS ve MVS’nin eğitim değerlerinin karşılaştırılması Çizelge 4.1’de verilmiştir.

4.1. Görüntü İşleme YSA’sının Eğitim Sonuçları



Şekil 4.1. Malzeme Veri Seti (MVS) için eğitim ve test maliyetleri grafiği gösterimi, devir (epoch) adedi = 1200



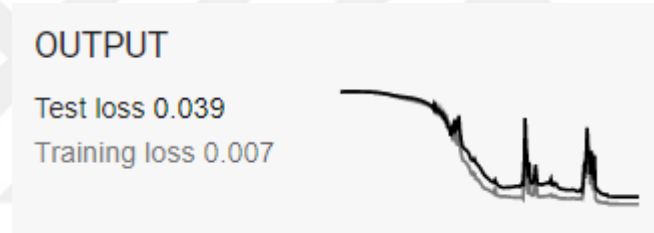
Şekil 4.2. Obje Veri Seti (OVS) için eğitim ve test maliyetleri grafiği, devir sayısı = 400

Çizelge 4.1. MVS ve OVS veri setleri eğitimleri için değerler tablosu

Veri Seti	Devir	Eğitilebilir Parameter Sayısı	Eğitim Maliyet Değeri	Eğitim Doğruluk Değeri	Validasyon Maliyet Değeri	Validasyon Doğruluk Değeri	Eğitim Süresi (sn)
MVS	1200	1537982	0.063	0.94	0.099	0.86	2359
OVS	400	122133	0.057	0.98	0.072	0.93	128

4.2. Katı Obje Sınıflandırma YSA'sının Eğitim Sonuçları

Katı obje sınıflandırma YSA'sı, konvolüsyonlu ve konvolüsyonsuz olarak iki farklı yapay sinir ağı ile eğitilmiştir ve çıkan değerler Çizelge 4.2'de karşılaştırılmıştır. Şekil 4.3'de Konvolüsyon filtresi bulunan, Şekil 4.4'de ise Konvolüsyon filtresi bulunmayan YSA'ların eğitim değerleri gösterilmektedir.



Şekil 4.3. Konvolüsyon filtresi bulunan YSA'nın eğitim maliyeti ve test maliyeti değerleri grafik gösterimi, devir sayısı = 10



Şekil 4.4. Konvolüsyon bulunmayan YSA'nın eğitim maliyeti ve test maliyeti değerleri grafik gösterimi, devir sayısı = 10

Çizelge 4.2. KVS için konvolüsyonlu ve konvolüsyonsuz YSA'ların eğitim değerleri karşılaştırılması

Tip	Devir	Eğitilebilir Parameter Sayısı	Eğitim Maliyet Değeri	Eğitim Doğruluk Değeri	Validasyon Maliyet Değeri	Validasyon Doğruluk Değeri	Eğitim Süresi (sn)
Konvolüsyonlu	100	2790	0.007	0.9812	0.039	0.9708	6260
Konvolüsyonsuz	100	11604	0.056	0.9667	0.078	0.9792	440

Çizelge 4.2’de gösterildiği üzere, evrişim filtreleri bulunan YSA’nın eğitim süresi 6260 saniye sürerken, evrişim filtresi bulunmayan YSA’nın eğitimi 440 saniyede gerçekleşmiştir. Doğruluk değerleri birbirine yakın olsa da, sınıf arttıkça evrişim filtrelerinin farkının daha çok ortaya çıkacağı düşünülmektedir.

4.3. Sonlu Elemanlar Analizleri Çözüm Sonuçları

Bu bölümde üç farklı malzeme için çözümü yapılan analiz çeşitleri, malzeme ve analiz sırası ile verilecektir. Çalışmada toplamda 648 farklı statik gerilme analizi ve 648 farklı doğal frekans tespiti analizi yapılmıştır. Sonuçlar tasarım numaralarına göre verilmiştir. Analiz süreleri her malzemenin statik veya doğal frekans analizlerine göre 216’şar analiz için verilmiştir. Analiz süreleri, boyutlar büyüdükçe mesh kalitesinin aynı kalması ve eleman boyutunun geometri değişse bile hep 0.5mm büyüklükte ayarlanmasından dolayı uzun sürmüştür. Böylece geometriler arasında mesh farkından dolayı oluşabilecek sonuç farklılıklarının önüne geçilmiştir. Geometrilere hexahedron ağ çeşidi ile eleman kalitesi “1” olacak şekilde mesh işlemi yapılmıştır.

4.3.1. Çelik malzemesi için statik analiz sonuçları

Çizelge 4.3. Çelik malzeme için Ansys ticari yazılımda çözümü yapılan analizlerin gerilme ve deplasman değerleri

Tasarım Numarası	Gerilme (MPa)	Deformasyon (mm)
1	46.960	0.002
2	88.110	0.013
3	199.470	0.195
4	717.510	12.470
5	1,760.000	195.170
6	6,800.000	12,400.000
7	23.270	0.001
8	41.550	0.004
9	90.500	0.058
10	363.920	3.700
11	838.890	58.310
12	3,120.000	3,720.000
13	9.250	0.000
14	15.360	0.001
15	32.780	0.013
16	120.680	0.800
17	290.140	12.500

18	1,170.000	807.250
19	3.180	0.000
20	4.850	0.000
21	9.220	0.002
22	31.190	0.101
23	76.790	1.570
24	302.930	102.080
25	0.955	0.000
26	1.120	0.000
27	1.990	0.000
28	6.480	0.007
29	13.530	0.101
30	49.980	6.550
31	0.208	0.000
32	0.249	0.000
33	0.255	0.000
34	0.575	0.000
35	0.995	0.002
36	3.270	0.101
37	30.840	0.001
38	56.870	0.008
39	127.290	0.129
40	511.780	8.320
41	1,160.000	130.590
42	4,570.000	8,230.000
43	14.720	0.000
44	26.840	0.003
45	57.120	0.039
46	214.970	2.460
47	525.740	38.590
48	2,050.000	2,470.000
49	6.240	0.000
50	10.640	0.001
51	23.220	0.009
52	78.930	0.532
53	194.410	8.330
54	761.620	536.650
55	2.240	0.000
56	3.220	0.000
57	6.400	0.001
58	20.730	0.067
59	48.740	1.040
60	198.000	67.350
61	0.652	0.000
62	0.747	0.000

63	1.300	0.000
64	3.750	0.005
65	8.160	0.067
66	31.940	4.330
67	0.135	0.000
68	0.147	0.000
69	0.186	0.000
70	0.346	0.000
71	0.669	0.001
72	2.100	0.068
73	16.360	0.001
74	29.330	0.005
75	69.880	0.077
76	274.080	4.960
77	679.060	77.600
78	2,550.000	4,930.000
79	8.470	0.000
80	15.750	0.002
81	34.870	0.023
82	127.560	1.470
83	311.820	23.080
84	1,240.000	1,470.000
85	3.660	0.000
86	6.930	0.000
87	14.100	0.005
88	49.020	0.319
89	114.800	4.990
90	450.840	320.120
91	1.170	0.000
92	1.980	0.000
93	3.910	0.001
94	13.830	0.040
95	33.210	0.625
96	122.080	40.150
97	0.379	0.000
98	0.462	0.000
99	0.776	0.000
100	2.430	0.003
101	5.270	0.040
102	21.460	2.570
103	0.079	0.000
104	0.095	0.000
105	0.109	0.000
106	0.213	0.000
107	0.405	0.001

108	1.400	0.041
109	7.500	0.000
110	14.470	0.002
111	33.930	0.038
112	139.470	2.470
113	335.970	38.670
114	1,190.000	2,430.000
115	4.180	0.000
116	6.830	0.001
117	15.940	0.011
118	60.240	0.731
119	150.000	11.450
120	567.830	731.920
121	1.680	0.000
122	3.290	0.000
123	6.570	0.003
124	26.820	0.160
125	59.040	2.490
126	216.390	159.410
127	0.565	0.000
128	0.914	0.000
129	1.880	0.000
130	6.520	0.020
131	14.940	0.311
132	59.530	20.010
133	0.174	0.000
134	0.220	0.000
135	0.370	0.000
136	1.170	0.001
137	2.820	0.020
138	9.410	1.280
139	0.021	0.000
140	0.043	0.000
141	0.055	0.000
142	0.098	0.000
143	0.196	0.000
144	0.729	0.020
145	2.710	0.000
146	4.940	0.001
147	13.540	0.015
148	53.150	0.969
149	130.490	15.110
150	400.650	898.790
151	1.350	0.000
152	2.230	0.000

153	6.180	0.004
154	23.940	0.290
155	60.070	4.520
156	220.200	286.230
157	0.587	0.000
158	1.000	0.000
159	2.320	0.001
160	9.130	0.063
161	22.320	0.987
162	83.320	63.150
163	0.200	0.000
164	0.318	0.000
165	0.654	0.000
166	2.380	0.008
167	6.380	0.124
168	21.840	7.960
169	0.062	0.000
170	0.078	0.000
171	0.146	0.000
172	0.507	0.001
173	1.040	0.008
174	3.700	0.511
175	0.013	0.000
176	0.016	0.000
177	0.020	0.000
178	0.040	0.000
179	0.078	0.000
180	0.288	0.008
181	0.610	0.000
182	1.190	0.000
183	2.970	0.003
184	11.980	0.229
185	24.720	3.370
186	71.140	192.170
187	0.293	0.000
188	0.552	0.000
189	1.310	0.001
190	5.510	0.069
191	13.620	1.080
192	31.440	57.320
193	0.124	0.000
194	0.227	0.000
195	0.513	0.000
196	2.050	0.015
197	5.330	0.237

198	18.690	14.940
199	0.042	0.000
200	0.067	0.000
201	0.150	0.000
202	0.548	0.002
203	1.400	0.030
204	5.320	1.960
205	0.013	0.000
206	0.016	0.000
207	0.031	0.000
208	0.107	0.000
209	0.231	0.002
210	0.874	0.127
211	0.003	0.000
212	0.003	0.000
213	0.004	0.000
214	0.009	0.000
215	0.019	0.000
216	0.065	0.002

Çelik malzeme için 216 adet analizin çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 9 saat sürmüştür.

4.3.2. Çelik malzeme için modal analiz sonuçları

Çizelge 4.4. Çelik malzeme için Ansys ticari yazılımında yapılan doğal frekans analizlerinin ilk 6 doğal frekans değeri tablosu

Tasarım Numarası	1. Doğal Frekans (Hz)	2. Doğal Frekans (Hz)	3. Doğal Frekans (Hz)	4. Doğal Frekans (Hz)	5. Doğal Frekans (Hz)	6. Doğal Frekans (Hz)
1	59,413.1	59,413.1	144,707.6	249,885.5	249,885.5	254,867.4
2	15,977.1	15,977.1	72,151.0	86,149.3	86,149.3	126,927.0
3	2,609.9	2,609.9	15,897.1	15,897.1	28,929.0	42,692.7
4	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0
7	59,695.7	80,234.8	137,098.9	250,138.0	255,217.2	276,698.6
8	16,029.3	23,158.1	67,648.9	86,344.1	110,184.8	127,054.9
9	2,614.0	3,891.0	15,920.5	22,932.2	26,857.9	42,759.5
10	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0
13	60,168.9	106,612.3	113,960.4	249,112.3	255,548.3	284,719.3
14	16,137.9	35,143.5	54,115.9	86,673.3	127,239.7	133,392.6
15	2,623.4	6,367.5	15,973.9	20,747.9	34,373.4	42,924.1

16	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0
19	60,813.8	83,248.9	132,022.0	144,737.4	245,402.7	245,744.1
20	16,342.2	35,904.9	53,145.3	86,233.3	109,728.9	117,047.8
21	2,655.5	11,786.2	12,738.4	16,167.3	39,849.4	43,607.1
22	164.7	811.5	1,030.3	2,878.9	2,980.1	4,870.4
23	0.0	0.0	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0
25	61,678.0	65,677.1	78,131.3	98,697.3	127,961.3	145,765.5
26	16,693.1	21,939.7	35,752.0	59,617.1	68,353.7	89,119.7
27	2,703.0	6,417.9	16,119.0	20,350.5	21,211.4	22,852.9
28	166.2	1,038.2	1,304.9	1,954.3	2,906.3	4,025.8
29	26.4	165.3	324.4	462.9	506.8	907.7
30	0.0	0.0	0.0	0.0	0.0	0.0
31	67,492.5	67,633.1	68,294.7	69,394.2	70,991.7	73,138.7
32	17,505.9	17,819.4	18,860.4	20,605.2	23,066.3	26,256.4
33	2,769.4	3,157.2	4,263.3	6,041.7	8,605.8	12,051.0
34	169.9	413.0	1,039.3	1,318.6	1,502.2	2,616.8
35	26.8	139.4	167.3	444.1	469.8	822.6
36	1.7	10.3	29.0	32.1	56.8	81.1
37	59,695.7	80,234.8	137,098.9	250,138.0	255,217.2	276,698.6
38	16,029.3	23,158.1	67,648.9	86,344.1	110,184.8	127,054.9
39	2,614.0	3,891.0	15,920.5	22,932.2	26,857.9	42,759.5
40	0.0	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0
43	80,539.1	80,539.1	145,131.2	255,619.0	276,694.4	276,694.4
44	23,220.7	23,220.7	72,245.5	110,320.1	110,320.1	127,193.8
45	3,896.1	3,896.1	22,958.2	22,958.2	28,843.4	50,669.9
46	0.0	0.0	0.0	0.0	0.0	0.0
47	0.0	0.0	0.0	0.0	0.0	0.0
48	0.0	0.0	0.0	0.0	0.0	0.0
49	81,076.9	106,910.6	134,715.9	256,031.5	275,814.0	284,591.8
50	23,351.0	35,222.3	65,967.1	110,577.7	127,396.6	133,429.6
51	3,907.6	6,375.1	23,017.9	25,877.6	34,401.6	50,717.5
52	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	0.0
54	0.0	0.0	0.0	0.0	0.0	0.0
55	81,815.7	106,397.9	132,204.3	179,433.4	245,648.5	271,543.8
56	23,630.5	48,173.5	53,238.4	110,164.6	127,641.1	142,387.3
57	3,940.2	11,798.1	17,517.4	23,160.5	49,877.3	50,800.5
58	246.7	811.9	1,540.5	4,189.4	4,290.0	4,872.5
59	0.0	0.0	0.0	0.0	0.0	0.0
60	0.0	0.0	0.0	0.0	0.0	0.0

61	82,801.3	87,200.0	101,343.4	125,570.2	145,858.3	160,454.3
62	23,943.3	30,702.8	48,684.7	68,397.2	79,654.4	110,326.1
63	4,007.3	9,294.4	21,227.7	23,082.7	29,191.7	32,182.3
64	249.0	1,552.5	1,922.2	1,955.2	4,330.5	5,921.7
65	39.6	247.7	324.5	693.5	747.4	1,358.8
66	0.0	0.0	0.0	0.0	0.0	0.0
67	90,190.0	90,288.8	90,952.9	92,032.8	93,659.4	95,927.1
68	25,319.1	25,715.7	27,081.7	29,386.3	32,662.0	36,934.2
69	4,127.8	4,689.0	6,297.7	8,887.8	12,618.6	17,603.8
70	254.7	617.1	1,554.3	1,970.3	2,240.4	3,892.5
71	40.2	208.4	250.8	663.9	703.9	1,175.4
72	2.5	15.5	43.4	48.1	81.1	85.2
73	60,168.9	106,612.3	113,960.4	249,112.3	255,548.3	284,719.3
74	16,137.9	35,143.5	54,115.9	86,673.3	127,239.7	133,392.6
75	2,623.4	6,367.5	15,973.9	20,747.9	34,373.4	42,924.1
76	0.0	0.0	0.0	0.0	0.0	0.0
77	0.0	0.0	0.0	0.0	0.0	0.0
78	0.0	0.0	0.0	0.0	0.0	0.0
79	81,076.9	106,910.6	134,715.9	256,031.5	275,814.0	284,591.8
80	23,351.0	35,222.3	65,967.1	110,577.7	127,396.6	133,429.6
81	3,907.6	6,375.1	23,017.9	25,877.6	34,401.6	50,717.5
82	0.0	0.0	0.0	0.0	0.0	0.0
83	0.0	0.0	0.0	0.0	0.0	0.0
84	0.0	0.0	0.0	0.0	0.0	0.0
85	107,387.3	107,387.3	145,898.0	256,535.6	283,882.7	283,882.7
86	35,375.9	35,375.9	72,459.7	127,633.2	133,499.5	133,499.5
87	6,390.8	6,390.8	28,860.4	34,459.7	34,459.7	50,770.8
88	0.0	0.0	0.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0
90	0.0	0.0	0.0	0.0	0.0	0.0
91	108,111.9	131,729.0	132,514.3	217,027.3	245,834.3	270,192.5
92	35,705.9	53,379.1	63,149.8	127,901.1	133,047.5	142,329.2
93	6,433.4	11,823.2	24,204.7	34,609.6	49,907.1	50,862.1
94	410.3	812.6	2,545.1	4,876.2	5,989.3	7,014.3
95	0.0	0.0	0.0	0.0	0.0	0.0
96	0.0	0.0	0.0	0.0	0.0	0.0
97	109,222.4	113,751.9	128,953.3	146,009.7	156,635.7	168,410.3
98	36,173.8	44,477.3	67,561.3	68,482.5	106,640.1	110,382.1
99	6,536.9	14,362.0	21,258.1	34,493.3	43,891.5	46,819.1
100	413.8	1,956.9	2,564.0	3,066.9	7,076.4	9,406.9
101	65.9	324.6	412.1	1,151.6	1,192.1	1,948.2
102	0.0	0.0	0.0	0.0	0.0	0.0
103	117,813.5	117,873.9	118,510.5	119,546.2	121,222.9	123,697.5
104	38,256.8	38,710.1	40,389.5	43,284.3	47,499.0	53,105.3
105	6,748.4	7,606.2	10,083.9	14,081.4	19,819.8	27,368.1

106	424.0	1,019.7	2,569.4	3,250.2	3,680.5	5,298.6
107	67.0	344.8	417.3	1,097.7	1,168.9	1,175.8
108	4.1	25.8	72.4	79.9	81.1	142.0
109	60,813.8	83,248.9	132,022.0	144,737.4	245,402.7	245,744.1
110	16,342.2	35,904.9	53,145.3	86,233.3	109,728.9	117,047.8
111	2,655.5	11,786.2	12,738.4	16,167.3	39,849.4	43,607.1
112	164.7	811.5	1,030.3	2,878.9	2,980.1	4,870.4
113	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0
115	81,815.7	106,397.9	132,204.3	179,433.4	245,648.5	271,543.8
116	23,630.5	48,173.5	53,238.4	110,164.6	127,641.1	142,387.3
117	3,940.2	11,798.1	17,517.4	23,160.5	49,877.3	50,800.5
118	246.7	811.9	1,540.5	4,189.4	4,290.0	4,872.5
119	0.0	0.0	0.0	0.0	0.0	0.0
120	0.0	0.0	0.0	0.0	0.0	0.0
121	108,111.9	131,729.0	132,514.3	217,027.3	245,834.3	270,192.5
122	35,705.9	53,379.1	63,149.8	127,901.1	133,047.5	142,329.2
123	6,433.4	11,823.2	24,204.7	34,609.6	49,907.1	50,862.1
124	410.3	812.6	2,545.1	4,876.2	5,989.3	7,014.3
125	0.0	0.0	0.0	0.0	0.0	0.0
126	0.0	0.0	0.0	0.0	0.0	0.0
127	133,032.2	133,032.2	146,991.3	217,413.6	235,870.0	245,699.4
128	53,693.7	53,693.7	72,949.0	128,267.8	141,941.4	141,941.4
129	11,882.6	11,882.6	28,941.5	49,977.1	49,977.1	50,973.5
130	814.1	814.1	4,884.5	4,884.5	7,232.8	12,662.4
131	0.0	0.0	0.0	0.0	0.0	0.0
132	0.0	0.0	0.0	0.0	0.0	0.0
133	133,689.7	138,453.7	146,251.8	155,025.2	168,170.0	181,091.3
134	54,194.1	62,722.9	68,654.7	91,102.6	110,247.4	126,844.0
135	12,033.8	21,322.5	22,792.1	49,822.5	51,109.7	56,943.9
136	819.1	1,960.3	4,911.1	5,225.9	9,944.4	12,682.2
137	131.5	324.9	818.3	1,949.8	2,077.1	2,268.4
138	0.0	0.0	0.0	0.0	0.0	0.0
139	139,051.5	139,183.9	139,911.1	141,271.2	143,481.5	146,720.2
140	55,602.1	56,040.8	57,803.1	61,046.2	66,049.9	72,942.9
141	12,299.2	13,568.1	17,314.7	23,507.4	28,723.6	32,331.8
142	841.3	1,978.7	4,947.7	5,305.0	6,244.2	6,965.2
143	133.6	674.8	828.7	1,176.8	2,139.8	2,302.0
144	8.2	51.6	81.1	144.6	157.1	283.4
145	61,678.0	65,677.1	78,131.3	98,697.3	127,961.3	145,765.5
146	16,693.1	21,939.7	35,752.0	59,617.1	68,353.7	89,119.7
147	2,703.0	6,417.9	16,119.0	20,350.5	21,211.4	22,852.9
148	166.2	1,038.2	1,304.9	1,954.3	2,906.3	4,025.8
149	26.4	165.3	324.4	462.9	506.8	907.7
150	0.0	0.0	0.0	0.0	0.0	0.0

151	82,801.3	87,200.0	101,343.4	125,570.2	145,858.3	160,454.3
152	23,943.3	30,702.8	48,684.7	68,397.2	79,654.4	110,326.1
153	4,007.3	9,294.4	21,227.7	23,082.7	29,191.7	32,182.3
154	249.0	1,552.5	1,922.2	1,955.2	4,330.5	5,921.7
155	39.6	247.7	324.5	693.5	747.4	1,358.8
156	0.0	0.0	0.0	0.0	0.0	0.0
157	109,222.4	113,751.9	128,953.3	146,009.7	156,635.7	168,410.3
158	36,173.8	44,477.3	67,561.3	68,482.5	106,640.1	110,382.1
159	6,536.9	14,362.0	21,258.1	34,493.3	43,891.5	46,819.1
160	413.8	1,956.9	2,564.0	3,066.9	7,076.4	9,406.9
161	65.9	324.6	412.1	1,151.6	1,192.1	1,948.2
162	0.0	0.0	0.0	0.0	0.0	0.0
163	133,689.7	138,453.7	146,251.8	155,025.2	168,170.0	181,091.3
164	54,194.1	62,722.9	68,654.7	91,102.6	110,247.4	126,844.0
165	12,033.8	21,322.5	22,792.1	49,822.5	51,109.7	56,943.9
166	819.1	1,960.3	4,911.1	5,225.9	9,944.4	12,682.2
167	131.5	324.9	818.3	1,949.8	2,077.1	2,268.4
168	0.0	0.0	0.0	0.0	0.0	0.0
169	146,642.6	146,642.6	149,066.7	162,734.9	162,744.1	168,097.7
170	68,930.0	68,930.0	73,653.7	97,933.6	103,346.8	108,751.6
171	21,477.5	21,477.5	29,179.6	51,307.1	56,776.5	56,776.5
172	1,968.7	1,968.7	7,219.6	9,968.8	9,968.8	12,706.5
173	325.7	325.7	1,953.8	1,953.8	2,893.1	5,065.0
174	0.0	0.0	0.0	0.0	0.0	0.0
175	149,138.2	149,424.4	150,463.0	152,374.2	153,334.3	153,733.2
176	69,638.2	70,226.6	72,282.7	75,222.1	76,195.1	77,208.2
177	21,779.0	23,229.0	27,917.8	28,778.1	35,914.3	36,247.1
178	2,007.7	4,250.7	5,318.7	9,961.0	12,754.0	12,784.1
179	329.9	1,179.3	1,521.7	1,973.8	4,712.6	4,987.1
180	20.6	81.2	128.6	358.8	361.0	487.1
181	67,492.5	67,633.1	68,294.7	69,394.2	70,991.7	73,138.7
182	17,505.9	17,819.4	18,860.4	20,605.2	23,066.3	26,256.4
183	2,769.4	3,157.2	4,263.3	6,041.7	8,605.8	12,051.0
184	169.9	413.0	1,039.3	1,318.6	1,502.2	2,616.8
185	26.8	139.4	167.3	444.1	469.8	822.6
186	1.7	10.3	29.0	32.1	56.8	81.1
187	90,190.0	90,288.8	90,952.9	92,032.8	93,659.4	95,927.1
188	25,319.1	25,715.7	27,081.7	29,386.3	32,662.0	36,934.2
189	4,127.8	4,689.0	6,297.7	8,887.8	12,618.6	17,603.8
190	254.7	617.1	1,554.3	1,970.3	2,240.4	3,892.5
191	40.2	208.4	250.8	663.9	703.9	1,175.4
192	2.5	15.5	43.4	48.1	81.1	85.2
193	117,813.5	117,873.9	118,510.5	119,546.2	121,222.9	123,697.5
194	38,256.8	38,710.1	40,389.5	43,284.3	47,499.0	53,105.3
195	6,748.4	7,606.2	10,083.9	14,081.4	19,819.8	27,368.1

196	424.0	1,019.7	2,569.4	3,250.2	3,680.5	5,298.6
197	67.0	344.8	417.3	1,097.7	1,168.9	1,175.8
198	4.1	25.8	72.4	79.9	81.1	142.0
199	139,051.5	139,183.9	139,911.1	141,271.2	143,481.5	146,720.2
200	55,602.1	56,040.8	57,803.1	61,046.2	66,049.9	72,942.9
201	12,299.2	13,568.1	17,314.7	23,507.4	28,723.6	32,331.8
202	841.3	1,978.7	4,947.7	5,305.0	6,244.2	6,965.2
203	133.6	674.8	828.7	1,176.8	2,139.8	2,302.0
204	8.2	51.6	81.1	144.6	157.1	283.4
205	149,138.2	149,424.4	150,463.0	152,374.2	153,334.3	153,733.2
206	69,638.2	70,226.6	72,282.7	75,222.1	76,195.1	77,208.2
207	21,779.0	23,229.0	27,917.8	28,778.1	35,914.3	36,247.1
208	2,007.7	4,250.7	5,318.7	9,961.0	12,754.0	12,784.1
209	329.9	1,179.3	1,521.7	1,973.8	4,712.6	4,987.1
210	20.6	81.2	128.6	358.8	361.0	487.1
211	153,145.4	153,150.3	153,150.3	153,212.5	154,440.0	154,440.0
212	76,019.8	76,019.8	76,173.6	77,279.5	78,080.0	78,263.6
213	28,895.9	28,895.9	29,669.9	33,843.4	34,252.9	35,646.2
214	5,369.4	5,369.4	7,294.9	12,826.8	14,194.1	14,194.1
215	1,188.3	1,188.3	2,894.2	4,997.7	4,997.7	5,097.3
216	81.4	81.4	488.4	488.4	723.3	1,266.2

Çelik malzemenin 216 farklı boyut parametresi analizleri gerçekleştirilmiş ve sonuçlar tayin edilmiştir. Çizelge 4.4’de görüldüğü üzere bazı değerler sıfıra eşit çıkmıştır. Çizelgelerde virgülden sonra 3 basamak verilmiştir. Yapay sınır ağlarına öğretilecek analiz sonuçlarında virgülden sonra 8 basamak alınacaktır.

Çelik malzeme için 216 adet analizin çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 14 saat sürmüştür.

4.3.3. Alüminyum malzeme için statik analiz sonuçları

Çizelge 4.5. Alüminyum malzeme için Ansys ticari yazılımında yapılan analizlerin sonuçları tablosu

Tasarım Numarası	Gerilme (MPa)	Deformasyon (mm)
1	48.369	0.005
2	90.753	0.038
3	205.454	0.585
4	739.035	37.410
5	1,812.800	585.510
6	7,004.000	37,200.000
7	23.968	0.002

8	42.797	0.012
9	93.215	0.174
10	374.838	11.100
11	864.057	174.930
12	3,213.600	11,160.000
13	9.528	0.001
14	15.821	0.003
15	33.763	0.038
16	124.300	2.400
17	298.844	37.500
18	1,205.100	2,421.750
19	3.275	0.000
20	4.996	0.001
21	9.497	0.005
22	32.126	0.303
23	79.094	4.710
24	312.018	306.240
25	0.984	0.000
26	1.154	0.000
27	2.050	0.001
28	6.674	0.020
29	13.936	0.303
30	51.479	19.650
31	0.214	0.000
32	0.256	0.000
33	0.263	0.000
34	0.592	0.001
35	1.025	0.005
36	3.368	0.303
37	31.765	0.004
38	58.576	0.025
39	131.109	0.387
40	527.133	24.960
41	1,194.800	391.770
42	4,707.100	24,690.000
43	15.162	0.001
44	27.645	0.008
45	58.834	0.116
46	221.419	7.380
47	541.512	115.770
48	2,111.500	7,410.000
49	6.427	0.000
50	10.959	0.002
51	23.917	0.026
52	81.298	1.596

53	200.242	24.990
54	784.469	1,609.950
55	2.307	0.000
56	3.317	0.000
57	6.592	0.004
58	21.352	0.200
59	50.202	3.120
60	203.940	202.050
61	0.672	0.000
62	0.769	0.000
63	1.339	0.000
64	3.863	0.014
65	8.405	0.201
66	32.898	12.990
67	0.139	0.000
68	0.151	0.000
69	0.192	0.000
70	0.356	0.000
71	0.689	0.004
72	2.163	0.203
73	16.851	0.002
74	30.210	0.015
75	71.976	0.230
76	282.302	14.880
77	699.432	232.800
78	2,626.500	14,790.000
79	8.724	0.001
80	16.223	0.005
81	35.916	0.070
82	131.387	4.410
83	321.175	69.240
84	1,277.200	4,410.000
85	3.770	0.000
86	7.138	0.001
87	14.523	0.015
88	50.491	0.957
89	118.244	14.970
90	464.365	960.360
91	1.205	0.000
92	2.039	0.000
93	4.027	0.002
94	14.245	0.121
95	34.206	1.875
96	125.742	120.450
97	0.390	0.000

98	0.476	0.000
99	0.799	0.000
100	2.503	0.008
101	5.428	0.121
102	22.104	7.710
103	0.082	0.000
104	0.000	0.000
105	0.112	0.000
106	0.219	0.000
107	0.417	0.002
108	1.442	0.122
109	7.725	0.001
110	14.904	0.007
111	34.948	0.113
112	143.654	7.410
113	346.049	116.010
114	1,225.700	7,290.000
115	4.305	0.000
116	7.035	0.002
117	16.418	0.034
118	62.047	2.193
119	154.500	34.350
120	584.865	2,195.760
121	1.730	0.000
122	3.389	0.001
123	6.767	0.008
124	27.625	0.480
125	60.811	7.470
126	222.882	478.230
127	0.582	0.000
128	0.941	0.000
129	1.936	0.001
130	6.716	0.060
131	15.388	0.933
132	61.316	60.030
133	0.179	0.000
134	0.227	0.000
135	0.381	0.000
136	1.205	0.004
137	2.905	0.060
138	9.692	3.840
139	0.000	0.000
140	0.044	0.000
141	0.057	0.000
142	0.101	0.000

143	0.202	0.001
144	0.751	0.061
145	2.791	0.000
146	5.088	0.003
147	13.946	0.044
148	54.745	2.907
149	134.405	45.330
150	412.670	2,696.370
151	1.391	0.000
152	2.297	0.001
153	6.365	0.013
154	24.658	0.870
155	61.872	13.560
156	226.806	858.690
157	0.605	0.000
158	1.030	0.000
159	2.390	0.003
160	9.404	0.188
161	22.990	2.961
162	85.820	189.450
163	0.206	0.000
164	0.328	0.000
165	0.674	0.000
166	2.451	0.024
167	6.571	0.372
168	22.495	23.880
169	0.064	0.000
170	0.080	0.000
171	0.150	0.000
172	0.522	0.002
173	1.071	0.024
174	3.811	1.533
175	0.013	0.000
176	0.016	0.000
177	0.020	0.000
178	0.042	0.000
179	0.081	0.000
180	0.297	0.024
181	0.628	0.000
182	1.226	0.001
183	3.059	0.010
184	12.339	0.687
185	25.462	10.110
186	73.274	576.510
187	0.302	0.000

188	0.569	0.000
189	1.349	0.003
190	5.675	0.207
191	14.029	3.240
192	32.383	171.960
193	0.128	0.000
194	0.234	0.000
195	0.528	0.001
196	2.112	0.045
197	5.490	0.711
198	19.251	44.820
199	0.043	0.000
200	0.069	0.000
201	0.155	0.000
202	0.564	0.006
203	1.442	0.090
204	5.480	5.880
205	0.013	0.000
206	0.017	0.000
207	0.031	0.000
208	0.110	0.000
209	0.238	0.006
210	0.900	0.381
211	0.003	0.000
212	0.003	0.000
213	0.004	0.000
214	0.009	0.000
215	0.019	0.000
216	0.067	0.006

Alümiyüm malzeme için 216 adet analizün çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 8 saat sürmüştür.

4.3.4. Alümiyüm malzeme için modal analiz sonuçları

Çizelge 4.6. Alümiyüm malzeme için Ansys ticari programında yapılan doğal frekans analizi sonuçları tablosu

Tasarım Numarası	1. Doğal Frekans (Hz)	2. Doğal Frekans (Hz)	3. Doğal Frekans (Hz)	4. Doğal Frekans (Hz)	5. Doğal Frekans (Hz)	6. Doğal Frekans (Hz)
1	59,829.3	59,829.3	143,643.8	250,102.7	250,102.7	256,438.1
2	16,053.5	16,053.5	71,556.4	86,343.7	86,343.7	127,485.8
3	2,616.9	2,616.9	15,927.8	15,927.8	28,558.0	42,734.9
4	163.7	163.7	1,024.1	1,024.1	2,858.9	2,858.9

5	26.2	26.2	164.0	164.0	459.1	459.1
6	1.6	1.6	10.3	10.3	28.7	28.7
7	60,171.8	80,646.5	136,152.2	250,397.1	256,853.9	276,215.7
8	16,116.2	23,258.1	67,109.3	86,573.7	110,212.5	127,640.6
9	2,621.6	3,902.6	15,954.9	22,972.9	26,574.1	42,811.9
10	163.8	245.5	1,024.6	1,532.1	2,860.3	4,261.2
11	26.2	39.3	164.1	246.0	459.2	687.9
12	1.6	2.5	10.3	15.4	28.7	43.1
13	60,729.2	106,789.1	113,396.9	249,076.2	257,182.0	284,172.2
14	16,248.2	35,249.5	53,749.4	86,968.7	127,860.4	133,100.8
15	2,632.8	6,385.3	16,018.2	20,555.1	34,387.7	43,006.3
16	164.0	408.8	1,025.8	2,532.1	2,864.0	5,022.6
17	26.2	65.5	164.1	409.4	459.4	899.6
18	1.6	4.1	10.3	25.6	28.7	56.3
19	61,429.3	83,175.3	131,570.9	144,348.2	244,879.6	245,624.9
20	16,487.7	35,743.0	53,117.7	86,336.8	110,411.3	116,706.4
21	2,662.8	11,808.7	12,539.0	16,152.3	39,199.9	43,387.8
22	164.6	813.3	1,029.5	2,875.8	2,910.8	4,875.2
23	26.2	130.8	164.4	460.2	813.7	901.3
24	1.6	8.2	10.3	28.7	51.2	56.3
25	61,946.6	65,698.3	77,771.4	98,029.8	127,170.2	144,740.0
26	16,715.6	21,731.1	35,263.6	58,939.1	67,977.1	87,764.8
27	2,712.9	6,353.4	16,045.8	20,432.5	21,175.5	22,642.2
28	26.4	165.2	325.1	462.6	494.9	907.4
29	1.6	10.3	20.4	28.8	56.4	93.2
30	12,213.2	34,420.0	45,666.6	65,513.3	65,686.2	66,599.2
31	10,035.4	16,615.0	17,522.4	17,582.4	18,995.5	24,974.8
32	2,762.0	3,158.5	4,244.6	7,038.9	9,862.8	10,081.2
33	170.3	412.4	1,033.0	1,367.1	1,504.5	2,578.3
34	26.8	138.8	167.2	443.3	469.0	823.1
35	1.6	10.3	28.9	31.8	56.8	81.3
36	62,815.8	83,554.5	143,087.0	254,409.3	297,078.2	316,058.3
37	16,388.4	23,552.6	68,592.8	92,041.9	116,408.1	127,600.0
38	2,639.6	3,923.8	16,096.2	23,153.3	26,847.0	43,627.9
39	164.1	245.7	1,026.4	1,533.1	2,865.5	4,263.7
40	26.2	39.3	164.2	246.0	459.5	688.1
41	1.6	2.5	10.3	15.4	28.7	43.1
42	84,318.5	84,318.5	151,134.4	254,830.4	315,583.0	315,583.0
43	23,706.3	23,706.3	73,322.0	116,737.5	116,737.5	127,936.3
44	3,933.6	3,933.6	23,203.7	23,203.7	28,875.1	50,886.6
45	245.8	246.0	1,533.9	1,535.1	4,266.0	4,270.0
46	39.3	39.3	246.1	246.1	688.2	688.5
47	2.5	2.5	15.4	15.4	43.1	43.1
48	85,668.5	111,479.1	139,338.3	256,272.5	305,223.6	311,581.3
49	23,961.9	35,959.0	66,770.1	117,204.7	128,342.9	139,421.7

50	3,953.1	6,435.2	23,306.3	25,891.6	34,742.4	50,959.9
51	246.3	409.5	1,537.2	2,537.2	4,276.1	6,373.4
52	39.3	65.5	246.3	409.7	688.9	1,143.8
53	2.5	4.1	15.4	25.6	43.1	71.7
54	86,959.4	109,965.9	136,163.1	189,968.9	248,226.7	273,864.0
55	24,375.7	48,853.1	54,287.0	116,227.8	128,812.2	145,018.3
56	3,998.4	11,911.0	17,578.3	23,515.0	50,238.2	51,094.2
57	247.2	814.9	1,542.5	4,123.4	4,292.9	4,886.7
58	39.4	130.9	246.6	689.8	814.2	1,349.9
59	2.5	8.2	15.4	43.1	51.2	84.4
60	87,655.8	91,142.3	107,062.9	128,341.6	147,982.4	163,192.6
61	24,714.7	31,436.8	50,199.2	68,988.8	87,449.2	110,422.8
62	4,073.0	9,411.0	21,368.1	23,340.5	30,562.1	32,652.7
63	249.7	1,555.7	1,925.0	1,963.2	4,336.0	5,932.0
64	39.6	247.7	325.4	693.3	739.3	1,358.3
65	2.5	15.4	20.4	43.1	84.5	127.8
66	17,487.5	34,781.1	45,668.7	71,797.8	74,917.0	80,877.1
67	14,619.2	23,449.3	25,233.4	25,274.3	27,122.1	30,275.4
68	4,120.2	4,682.3	6,273.0	9,750.6	14,353.2	14,779.4
69	255.4	617.0	1,545.5	2,044.4	2,244.0	3,842.9
70	40.2	207.9	250.7	663.6	702.9	1,178.7
71	2.5	15.5	43.4	47.7	81.3	85.1
72	63,892.7	110,868.9	118,572.6	255,793.3	290,014.1	305,247.7
73	16,591.0	35,763.1	54,682.7	92,618.6	127,956.3	139,323.0
74	2,654.9	6,421.3	16,185.0	20,746.2	34,690.9	43,898.7
75	164.3	409.3	1,027.9	2,535.8	2,870.2	5,051.2
76	26.2	65.5	164.3	409.6	459.8	900.4
77	26.2	65.5	164.3	409.6	459.8	900.4
78	1.6	4.1	10.3	25.6	28.7	56.3
79	85,668.5	111,479.1	139,338.3	256,272.5	305,223.6	311,581.3
80	23,961.9	35,959.0	66,770.1	117,204.7	128,342.9	139,421.7
81	3,953.1	6,435.2	23,306.3	25,891.6	34,742.4	50,959.9
82	246.3	409.5	1,537.2	2,537.2	4,276.1	6,373.4
83	39.3	65.5	246.3	409.7	688.9	1,143.8
84	2.5	4.1	15.4	25.6	43.1	71.7
85	112,848.4	112,848.4	150,537.9	257,977.1	304,188.6	304,188.6
86	36,266.5	36,266.5	73,354.4	128,813.2	139,551.3	139,551.3
87	6,461.9	6,461.9	28,879.2	34,842.5	34,842.5	51,043.3
88	410.0	410.0	2,539.9	2,539.9	6,986.0	6,986.0
89	65.5	65.5	409.9	409.9	1,144.3	1,144.3
90	4.1	4.1	25.6	25.6	71.8	71.8
91	114,430.4	135,273.0	137,093.3	226,902.5	249,689.3	274,748.9
92	36,810.0	54,590.6	63,890.4	129,355.4	138,777.2	144,974.4
93	6,525.3	11,953.6	24,220.8	35,075.2	50,289.6	51,193.2
94	411.2	815.7	2,547.2	4,891.4	5,840.9	7,008.0

95	65.6	131.0	410.3	814.5	1,145.6	2,235.4
96	4.1	8.2	25.6	51.2	71.8	140.6
97	115,286.6	118,482.9	135,545.5	148,204.2	158,171.6	168,461.8
98	37,347.7	45,462.6	69,122.7	69,700.5	110,610.2	114,575.9
99	6,649.1	14,505.8	21,419.0	34,903.4	45,978.4	47,372.4
100	415.1	1,965.3	2,568.7	3,052.5	7,077.4	9,361.3
101	65.9	325.5	412.1	1,151.2	1,176.7	1,952.2
102	4.1	20.5	25.7	71.9	127.8	140.8
103	25,887.0	35,751.7	45,681.5	71,389.9	74,465.9	79,014.6
104	22,410.2	33,268.7	34,981.9	37,704.5	37,897.9	38,204.6
105	6,740.7	7,581.4	10,039.9	14,820.3	21,867.1	23,139.2
106	425.2	1,020.8	2,554.6	3,376.0	3,684.2	5,306.4
107	67.1	344.9	417.3	1,099.3	1,167.6	1,179.2
108	4.1	25.8	72.3	79.1	81.3	141.8
109	64,809.3	87,020.1	135,815.8	153,944.5	247,723.6	269,634.2
110	16,873.9	36,511.1	54,080.2	91,504.5	116,015.8	124,522.1
111	2,686.9	11,887.9	12,718.6	16,334.0	39,836.1	44,310.8
112	165.0	814.4	1,031.8	2,882.3	2,943.8	4,884.2
113	26.3	130.9	164.5	460.5	814.0	902.0
114	1.6	8.2	10.3	28.7	51.2	56.3
115	86,959.4	109,965.9	136,163.1	189,968.9	248,226.7	273,864.0
116	24,375.7	48,853.1	54,287.0	116,227.8	128,812.2	145,018.3
117	3,998.4	11,911.0	17,578.3	23,515.0	50,238.2	51,094.2
118	247.2	814.9	1,542.5	4,123.4	4,292.9	4,886.7
119	39.4	130.9	246.6	689.8	814.1	1,349.9
120	2.5	8.2	15.4	43.1	51.2	84.4
121	114,430.4	135,273.0	137,093.3	226,902.5	249,689.3	274,748.9
122	36,810.0	54,590.6	63,890.4	129,355.4	138,777.2	144,974.4
123	6,525.3	11,953.6	24,220.8	35,075.2	50,289.6	51,193.2
124	411.2	815.7	2,547.2	4,891.4	5,840.9	7,008.0
125	65.6	130.9	410.3	814.4	1,145.6	2,235.4
126	4.1	8.2	25.6	51.2	71.8	140.6
127	138,491.4	138,491.4	149,188.9	220,890.3	238,965.9	249,421.0
128	55,130.3	55,130.3	73,554.5	130,064.2	144,534.6	144,534.6
129	12,046.5	12,046.5	28,907.9	50,400.8	50,400.8	51,377.7
130	817.7	817.7	4,902.2	4,902.2	7,172.7	12,710.8
131	131.1	131.1	815.2	815.4	2,255.7	2,256.5
132	8.2	8.2	51.3	51.3	143.4	143.4
133	139,086.7	142,629.0	148,666.8	158,710.7	168,915.3	182,242.8
134	55,743.6	63,676.5	69,377.5	93,292.1	110,617.3	129,407.2
135	12,239.1	21,526.9	22,868.3	50,228.6	51,553.4	56,977.6
136	823.8	1,970.0	4,935.4	5,206.2	9,970.8	12,737.1
137	131.5	325.9	818.2	1,954.0	2,034.1	2,265.2
138	8.2	20.5	51.3	127.8	143.6	281.0
139	37,041.9	38,646.3	45,786.1	70,945.2	75,625.3	76,770.9

140	33,703.3	37,180.1	41,463.1	43,485.2	45,234.7	53,249.9
141	12,434.5	13,664.2	17,453.8	24,375.1	28,474.1	31,866.9
142	845.1	1,981.4	4,918.5	5,314.2	6,490.3	6,952.6
143	133.9	675.9	829.0	1,180.4	2,143.7	2,298.9
144	8.2	51.6	81.3	144.4	155.4	283.1
145	65,245.7	69,050.4	82,582.5	105,926.4	134,669.0	147,903.2
146	17,074.5	22,253.8	36,387.5	66,482.5	68,897.1	94,337.4
147	2,733.5	6,463.3	16,184.2	21,195.1	21,340.8	23,036.0
148	166.6	1,040.2	1,307.8	1,962.0	2,910.1	4,037.2
149	26.4	165.3	325.3	462.8	501.7	907.5
150	1.6	10.3	20.4	28.8	56.4	93.2
151	87,655.8	91,142.3	107,062.9	128,341.6	147,982.4	163,192.6
152	24,714.7	31,436.8	50,199.2	68,988.8	87,449.2	110,422.8
153	4,073.0	9,411.0	21,368.1	23,340.5	30,562.1	32,652.7
154	249.7	1,555.7	1,925.0	1,963.2	4,336.0	5,932.0
155	39.6	247.7	325.4	693.3	739.3	1,358.3
156	2.5	15.4	20.4	43.1	84.5	127.8
157	115,286.6	118,482.9	135,545.5	148,204.2	158,171.6	168,461.8
158	37,347.7	45,462.6	69,122.7	69,700.5	110,610.2	114,575.9
159	6,649.1	14,505.8	21,419.0	34,903.4	45,978.4	47,372.4
160	415.1	1,965.3	2,568.7	3,052.5	7,077.4	9,361.3
161	65.9	325.5	412.1	1,151.2	1,176.7	1,952.2
162	4.1	20.5	25.7	71.9	127.8	140.8
163	139,086.7	142,629.0	148,666.8	158,710.7	168,915.3	182,242.8
164	55,743.6	63,676.5	69,377.5	93,292.1	110,617.3	129,407.2
165	12,239.1	21,526.9	22,868.3	50,228.6	51,553.4	56,977.6
166	823.8	1,970.0	4,935.4	5,206.2	9,970.8	12,737.1
167	131.5	325.9	818.2	1,954.0	2,034.1	2,265.2
168	8.2	20.5	51.3	127.8	143.6	281.0
169	149,235.5	149,235.5	151,294.6	162,396.9	164,348.4	168,262.4
170	69,766.1	69,766.1	73,693.0	98,613.1	102,659.8	109,361.2
171	21,732.4	21,732.4	29,103.2	51,823.6	56,770.1	56,770.1
172	1,980.9	1,980.9	7,186.3	10,002.3	10,002.3	12,770.0
173	326.7	326.7	1,958.3	1,958.4	2,867.5	5,082.5
174	20.5	20.5	127.9	127.9	356.3	356.5
175	44,130.1	44,371.2	46,197.4	72,188.2	75,615.5	77,870.5
176	40,452.9	41,075.7	41,914.9	53,635.0	55,977.6	58,415.4
177	21,978.1	23,344.3	27,961.0	28,529.1	34,984.9	36,900.1
178	2,025.6	4,263.7	5,331.6	9,982.1	12,808.2	13,357.6
179	331.6	1,183.1	1,521.6	1,981.4	4,715.3	4,986.2
180	20.6	81.4	128.5	355.5	358.3	487.9
181	12,213.2	34,420.0	45,666.6	65,513.3	65,686.2	66,599.2
182	10,035.4	16,615.0	17,522.4	17,582.4	18,995.5	24,974.8
183	2,762.0	3,158.5	4,244.6	7,038.9	9,862.8	10,081.2
184	170.3	412.4	1,033.0	1,367.1	1,504.5	2,578.3

185	26.8	138.8	167.2	443.3	469.0	823.1
186	1.6	10.3	28.9	31.8	56.8	81.3
187	17,487.5	34,781.1	45,668.7	71,797.8	74,917.0	80,877.1
188	14,619.2	23,449.3	25,233.4	25,274.3	27,122.1	30,275.4
189	4,120.2	4,682.3	6,273.0	9,750.6	14,353.2	14,779.4
190	255.4	617.0	1,545.5	2,044.4	2,244.0	3,842.9
191	40.2	207.9	250.7	663.6	702.9	1,178.7
192	2.5	15.5	43.4	47.7	81.3	85.1
193	25,887.0	35,751.7	45,681.5	71,389.9	74,465.9	79,014.6
194	22,410.2	33,268.7	34,981.9	37,704.5	37,897.9	38,204.6
195	6,740.7	7,581.4	10,039.9	14,820.3	21,867.1	23,139.2
196	425.2	1,020.8	2,554.6	3,376.0	3,684.2	5,306.4
197	67.1	344.9	417.3	1,099.3	1,167.6	1,179.2
198	4.1	25.8	72.3	79.1	81.3	141.8
199	37,041.9	38,646.3	45,786.1	70,945.2	75,625.3	76,770.9
200	33,703.3	37,180.1	41,463.1	43,485.2	45,234.7	53,249.9
201	12,434.5	13,664.2	17,453.8	24,375.1	28,474.1	31,866.9
202	845.1	1,981.4	4,918.5	5,314.2	6,490.3	6,952.6
203	133.9	675.9	829.0	1,180.4	2,143.7	2,298.9
204	8.2	51.6	81.3	144.4	155.4	283.1
205	44,130.1	44,371.2	46,197.4	72,188.2	75,615.5	77,870.5
206	40,452.9	41,075.7	41,914.9	53,635.0	55,977.6	58,415.4
207	21,978.1	23,344.3	27,961.0	28,529.1	34,984.9	36,900.1
208	2,025.6	4,263.7	5,331.6	9,982.1	12,808.2	13,357.6
209	331.6	1,183.1	1,521.6	1,981.4	4,715.3	4,986.2
210	20.6	81.4	128.5	355.5	358.3	487.9
211	42,900.0	44,650.3	46,611.5	46,611.5	50,626.1	51,620.5
212	39,489.9	40,007.2	42,255.4	42,255.4	43,810.0	46,026.9
213	28,661.3	28,661.3	29,362.3	32,938.7	33,401.8	34,825.9
214	5,388.3	5,388.3	7,260.6	12,876.2	14,163.0	14,163.0
215	1,193.2	1,193.3	2,880.1	4,997.4	4,997.6	5,118.0
216	81.6	81.6	489.1	489.1	717.2	1,270.0

Alüminyum malzeme için 216 adet analiz için çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 14 saat sürmüştür.

4.3.5. Abs plastik malzemesi için statik analiz sonuçları

Çizelge 4.7. Abs plastik malzemesi için Ansys ticari yazılımında yapılan analizlerin sonuçları tablosu

Tasarım Numarası	Gerilme (MPa)	Deformasyon (mm)
1	48.720	0.317
2	83.920	2.290
3	189.660	35.160
4	706.760	2,260.000

5	1,750.000	35,400.000
6	5,350.000	108,222.857
7	22.290	0.111
8	38.880	0.711
9	87.880	10.490
10	318.440	671.230
11	816.950	10,500.000
12	2,800.000	623,000.000
13	8.850	0.036
14	16.400	0.071
15	31.470	2.310
16	127.080	145.610
17	312.690	2,300.000
18	1,090.000	142,000.000
19	3.070	0.012
20	4.730	0.037
21	8.600	0.324
22	31.360	18.300
23	76.080	285.810
24	287.240	18,300.000
25	0.989	0.004
26	1.250	0.009
27	1.930	0.037
28	7.720	0.148
29	19.300	0.370
30	77.200	1.480
31	0.241	0.002
32	0.482	0.005
33	1.205	0.011
34	4.820	0.045
35	12.050	0.113
36	48.200	0.450
37	29.130	0.206
38	58.260	0.412
39	145.650	1.030
40	582.600	4.120
41	1,456.500	10.300
42	5,826.000	41.200
43	13.660	0.072
44	27.320	0.143
45	68.300	0.358
46	273.200	1.430
47	683.000	3.575
48	2,732.000	14.300
49	6.080	0.024

50	12.160	0.047
51	30.400	0.119
52	121.600	0.474
53	304.000	1.185
54	1,216.000	4.740
55	2.010	0.008
56	4.020	0.016
57	10.050	0.040
58	40.200	0.159
59	100.500	0.398
60	402.000	1.590
61	0.621	0.003
62	1.242	0.006
63	3.105	0.015
64	12.420	0.058
65	31.050	0.145
66	124.200	0.580
67	0.126	0.001
68	0.252	0.001
69	0.630	0.004
70	2.520	0.014
71	6.300	0.036
72	25.200	0.143
73	15.290	0.118
74	30.580	0.236
75	76.450	0.590
76	305.800	2.360
77	764.500	5.900
78	3,058.000	23.600
79	8.140	0.043
80	16.280	0.085
81	40.700	0.213
82	162.800	0.850
83	407.000	2.125
84	1,628.000	8.500
85	3.490	0.014
86	6.980	0.028
87	17.450	0.069
88	69.800	0.276
89	174.500	0.690
90	698.000	2.760
91	1.170	0.005
92	2.340	0.009
93	5.850	0.023
94	23.400	0.093

95	58.500	0.232
96	234.000	0.928
97	0.366	0.002
98	0.732	0.003
99	1.830	0.008
100	7.320	0.034
101	18.300	0.084
102	73.200	0.336
103	0.075	0.000
104	0.149	0.001
105	0.374	0.002
106	1.494	0.008
107	3.735	0.020
108	14.940	0.080
109	7.240	0.057
110	14.480	0.115
111	36.200	0.287
112	144.800	1.148
113	362.000	2.870
114	1,448.000	11.480
115	3.470	0.020
116	6.940	0.041
117	17.350	0.102
118	69.400	0.408
119	173.500	1.020
120	694.000	4.080
121	1.560	0.007
122	3.120	0.013
123	7.800	0.033
124	31.200	0.133
125	78.000	0.333
126	312.000	1.330
127	0.532	0.002
128	1.064	0.004
129	2.660	0.011
130	10.640	0.045
131	26.600	0.112
132	106.400	0.446
133	0.167	0.001
134	0.334	0.002
135	0.835	0.004
136	3.340	0.016
137	8.350	0.040
138	33.400	0.161
139	0.032	0.000

140	0.065	0.000
141	0.162	0.001
142	0.648	0.004
143	1.620	0.010
144	6.480	0.039
145	2.700	0.022
146	5.400	0.045
147	13.500	0.112
148	54.000	0.448
149	135.000	1.120
150	540.000	4.480
151	1.190	0.008
152	2.380	0.016
153	5.950	0.039
154	23.800	0.158
155	59.500	0.395
156	238.000	1.578
157	0.533	0.003
158	1.066	0.005
159	2.665	0.013
160	10.660	0.051
161	26.650	0.127
162	106.600	0.506
163	0.178	0.001
164	0.356	0.002
165	0.890	0.004
166	3.560	0.017
167	8.900	0.043
168	35.600	0.171
169	0.058	0.000
170	0.116	0.001
171	0.291	0.002
172	1.164	0.006
173	2.910	0.016
174	11.640	0.062
175	0.012	0.000
176	0.024	0.000
177	0.059	0.000
178	0.236	0.001
179	0.590	0.004
180	2.360	0.014
181	0.571	0.005
182	1.142	0.010
183	2.855	0.025
184	11.420	0.100

185	28.550	0.251
186	114.200	1.002
187	0.265	0.002
188	0.530	0.004
189	1.325	0.010
190	5.300	0.039
191	13.250	0.098
192	53.000	0.392
193	0.110	0.001
194	0.220	0.001
195	0.550	0.003
196	2.200	0.012
197	5.500	0.030
198	22.000	0.121
199	0.038	0.000
200	0.076	0.000
201	0.190	0.001
202	0.760	0.004
203	1.900	0.010
204	7.600	0.041
205	0.012	0.000
206	0.024	0.000
207	0.059	0.000
208	0.236	0.001
209	0.590	0.004
210	2.360	0.014
211	0.002	0.000
212	0.005	0.000
213	0.012	0.000
214	0.049	0.000
215	0.123	0.001
216	0.492	0.003

ABS plastik malzeme için 216 adet analizin çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 4 saat sürmüştür.

4.3.6. Abs plastik malzemesi için doğal frekans analizi sonuçları

Çizelge 4.8. ABS Plastik malzemesi için Ansys ticari yazılımda yapılan doğal frekans analizi sonuçları

Tasarım Numarası	1. Doğal Frekans (Hz)	2. Doğal Frekans (Hz)	3. Doğal Frekans (Hz)	4. Doğal Frekans (Hz)	5. Doğal Frekans (Hz)	6. Doğal Frekans (Hz)
1	13,474.963	13,474.963	31,136.526	53,981.083	62,069.265	62,069.265

2	3,530.914	3,530.914	15,081.569	19,577.799	19,577.799	27,156.606
3	564.883	564.883	3,440.264	3,440.264	5,939.427	9,305.041
4	34.930	34.930	218.482	218.482	609.883	609.883
5	5.571	5.575	34.895	34.924	97.649	97.740
6	0.348	0.348	2.179	2.180	6.102	6.104
7	13,808.155	17,960.589	29,497.004	54,092.774	61,888.374	65,810.603
8	3,584.084	5,095.163	14,132.064	19,773.312	24,633.161	27,259.633
9	568.049	841.759	3,459.628	4,950.500	5,526.153	9,359.930
10	34.978	52.321	218.780	326.395	610.750	907.480
11	5.577	8.357	34.938	52.330	97.780	146.350
12	0.348	0.522	2.180	3.269	6.104	9.153
13	14,318.585	23,531.882	24,660.325	54,496.595	59,990.107	63,998.797
14	3,673.817	7,694.294	11,353.436	20,044.795	27,385.823	29,168.198
15	574.185	1,376.384	3,497.301	4,283.746	7,378.169	9,474.226
16	35.081	87.204	219.441	539.972	612.714	1,039.982
17	5.583	13.932	34.981	87.141	97.903	191.726
18	0.348	0.870	2.181	5.448	6.106	11.964
19	14,784.800	18,662.645	28,426.013	32,440.019	52,444.488	56,387.369
20	3,805.514	7,739.968	11,501.184	19,704.104	25,190.347	26,315.879
21	586.987	2,539.961	2,649.094	3,560.737	8,331.401	9,645.317
22	35.315	173.504	220.882	607.267	617.241	1,038.479
23	5.596	27.841	35.062	98.145	173.111	192.261
24	0.348	1.739	2.182	6.110	10.893	11.973
25	15,013.803	15,521.120	18,101.908	22,170.423	28,414.325	30,764.641
26	3,901.033	4,888.361	7,774.001	14,085.787	14,432.746	20,673.860
27	605.895	1,367.834	3,500.651	4,509.589	4,628.196	4,937.849
28	35.924	224.005	270.878	417.654	627.244	838.823
29	5.643	35.335	69.225	98.985	103.469	194.225
30	0.349	2.187	4.347	6.123	12.000	19.841
31	2,565.922	7,088.457	9,507.752	14,784.592	15,136.289	15,153.686
32	2,135.752	3,482.413	4,003.884	4,005.090	4,299.021	5,390.894
33	617.055	694.792	905.537	1,488.599	2,099.381	2,144.021
34	37.307	86.720	220.961	298.211	320.361	546.247
35	5.803	28.819	36.026	92.596	101.131	173.502
36	0.352	2.206	6.182	6.559	12.139	17.283
37	13,808.155	17,960.589	29,497.004	54,092.774	61,888.374	65,810.603
38	3,584.084	5,095.163	14,132.064	19,773.312	24,633.161	27,259.633
39	568.049	841.759	3,459.628	4,950.500	5,526.153	9,359.930
40	34.978	52.321	218.780	326.395	610.750	907.480
41	5.576	8.357	34.936	52.329	97.773	146.349
42	0.348	0.522	2.180	3.269	6.104	9.153
43	18,266.038	18,266.038	31,095.999	54,224.240	65,671.244	65,671.244
44	5,159.417	5,159.417	15,083.726	24,770.476	24,770.476	27,384.162
45	845.686	845.686	4,971.454	4,971.454	5,939.781	10,855.462
46	52.374	52.428	326.724	327.131	908.396	909.680

47	8.360	8.363	52.351	52.376	146.410	146.490
48	0.522	0.522	3.270	3.270	9.154	9.156
49	18,853.308	23,754.787	28,790.744	54,647.222	64,013.580	64,502.988
50	5,269.554	7,774.479	13,787.720	24,972.872	27,534.901	29,206.461
51	853.469	1,381.999	5,014.203	5,334.786	7,399.440	10,882.105
52	52.563	87.296	327.985	540.528	912.163	1,311.526
53	8.372	13.938	52.431	87.178	146.648	243.339
54	0.522	0.870	3.271	5.449	9.157	15.254
55	19,465.765	23,305.979	28,540.694	39,831.487	52,602.294	58,065.272
56	5,456.281	10,267.702	11,583.175	24,707.986	27,685.435	30,341.166
57	871.350	2,549.265	3,650.724	5,101.452	10,575.222	10,929.090
58	52.881	173.686	329.981	850.077	918.430	1,039.482
59	8.390	27.853	52.541	146.979	173.181	287.645
60	0.522	1.739	3.273	9.163	10.895	17.954
61	19,808.405	20,127.410	23,175.103	26,889.141	30,790.107	34,170.406
62	5,616.336	6,864.939	10,662.795	14,466.433	18,475.742	23,474.705
63	901.584	1,987.743	4,520.276	5,035.702	6,651.784	6,965.996
64	53.803	334.846	398.564	418.109	933.940	1,231.734
65	8.459	52.956	69.257	148.255	152.451	290.627
66	0.523	3.279	4.348	9.183	17.995	27.162
67	3,670.660	7,170.634	9,508.624	14,773.793	15,414.495	16,640.096
68	3,107.400	4,878.718	5,754.809	5,759.410	6,112.096	6,592.833
69	920.380	1,029.827	1,337.746	2,071.347	3,049.655	3,138.489
70	55.946	129.705	330.583	445.859	477.628	813.998
71	8.704	43.161	54.018	138.585	151.537	250.166
72	0.528	3.309	9.272	9.825	17.286	18.205
73	14,318.585	23,531.882	24,660.325	54,496.595	59,990.107	63,998.797
74	3,673.817	7,694.294	11,353.436	20,044.795	27,385.823	29,168.198
75	574.185	1,376.384	3,497.301	4,283.746	7,378.169	9,474.226
76	35.081	87.204	219.441	539.972	612.714	1,039.982
77	5.583	13.931	34.978	87.137	97.895	191.710
78	0.348	0.870	2.181	5.448	6.106	11.964
79	18,853.308	23,754.787	28,790.744	54,647.222	64,013.580	64,502.988
80	5,269.554	7,774.479	13,787.720	24,972.872	27,534.901	29,206.461
81	853.469	1,381.999	5,014.203	5,334.786	7,399.440	10,882.105
82	52.563	87.296	327.985	540.528	912.163	1,311.526
83	8.372	13.938	52.431	87.178	146.648	243.339
84	0.522	0.870	3.271	5.449	9.157	15.254
85	24,283.643	24,283.643	30,981.511	55,162.405	63,748.408	63,748.408
86	7,903.899	7,903.899	15,094.115	27,717.869	29,258.048	29,258.048
87	1,392.664	1,392.664	5,941.166	7,440.682	7,440.682	10,913.533
88	87.472	87.472	541.606	541.606	1,475.044	1,488.446
89	13.949	13.949	87.249	87.249	243.540	243.540
90	0.870	0.870	5.450	5.451	15.257	15.259
91	24,953.402	28,237.892	28,853.781	47,303.364	53,066.889	58,395.300

92	8,138.911	11,705.856	13,276.151	27,901.409	29,034.823	30,332.224
93	1,417.386	2,566.198	5,006.382	7,535.256	10,595.861	10,968.014
94	87.910	174.022	544.337	1,041.320	1,202.889	1,496.619
95	13.973	27.873	87.399	173.309	243.983	476.051
96	0.870	1.740	5.453	10.897	15.266	29.907
97	25,321.151	25,587.950	28,773.821	30,861.105	32,954.984	35,493.057
98	8,382.480	9,815.026	14,515.974	14,666.370	23,557.482	24,015.677
99	1,466.709	3,050.423	4,539.970	7,479.805	9,935.408	10,007.942
100	89.334	418.957	552.225	631.505	1,521.519	1,941.008
101	14.078	69.316	88.045	242.570	246.000	414.818
102	0.872	4.349	5.464	15.296	27.168	29.971
103	5,423.573	7,386.421	9,513.873	14,692.615	15,321.424	16,258.562
104	4,748.351	6,900.834	7,231.549	7,899.403	8,417.629	8,581.892
105	1,503.872	1,664.280	2,137.551	3,147.353	4,624.147	4,897.230
106	93.125	214.502	546.274	735.716	783.452	1,117.154
107	14.500	71.592	89.904	229.514	250.370	251.675
108	0.880	5.513	15.449	16.305	17.290	30.328
109	14,784.800	18,662.645	28,426.013	32,440.019	52,444.488	56,387.369
110	3,805.514	7,739.968	11,501.184	19,704.104	25,190.347	26,315.879
111	586.987	2,539.961	2,649.094	3,560.737	8,331.401	9,645.317
112	35.315	173.504	220.882	607.267	617.241	1,038.479
113	5.596	27.841	35.061	98.142	173.100	192.256
114	0.348	1.739	2.182	6.110	10.893	11.973
115	19,465.765	23,305.979	28,540.694	39,831.487	52,602.294	58,065.272
116	5,456.281	10,267.702	11,583.175	24,707.986	27,685.435	30,341.166
117	871.350	2,549.265	3,650.724	5,101.452	10,575.222	10,929.090
118	52.881	173.686	329.981	850.077	918.430	1,039.482
119	8.389	27.852	52.540	146.975	173.170	287.639
120	0.522	1.739	3.273	9.163	10.894	17.954
121	24,953.402	28,237.892	28,853.781	47,303.364	53,066.889	58,395.300
122	8,138.911	11,705.856	13,276.151	27,901.409	29,034.823	30,332.224
123	1,417.386	2,566.198	5,006.382	7,535.256	10,595.861	10,968.014
124	87.910	174.022	544.337	1,041.320	1,202.889	1,496.619
125	13.973	27.873	87.397	173.298	243.978	476.041
126	0.870	1.740	5.453	10.896	15.266	29.907
127	29,359.783	29,359.783	30,720.808	46,640.379	50,403.694	52,235.712
128	11,922.288	11,922.288	15,157.252	28,157.577	30,195.301	30,195.301
129	2,602.314	2,602.314	5,950.382	10,639.731	10,639.731	11,039.041
130	174.760	174.760	1,045.403	1,045.403	1,475.636	2,709.968
131	27.918	27.920	173.577	173.617	480.042	480.230
132	1.740	1.740	10.901	10.902	30.497	30.501
133	29,582.852	30,003.623	31,000.932	32,961.255	35,703.400	37,759.552
134	12,173.603	13,457.081	14,606.889	19,442.542	23,598.677	26,847.780
135	2,676.775	4,581.082	4,763.547	10,592.110	11,098.105	11,946.420
136	176.938	420.772	1,057.418	1,074.795	2,109.079	2,719.494

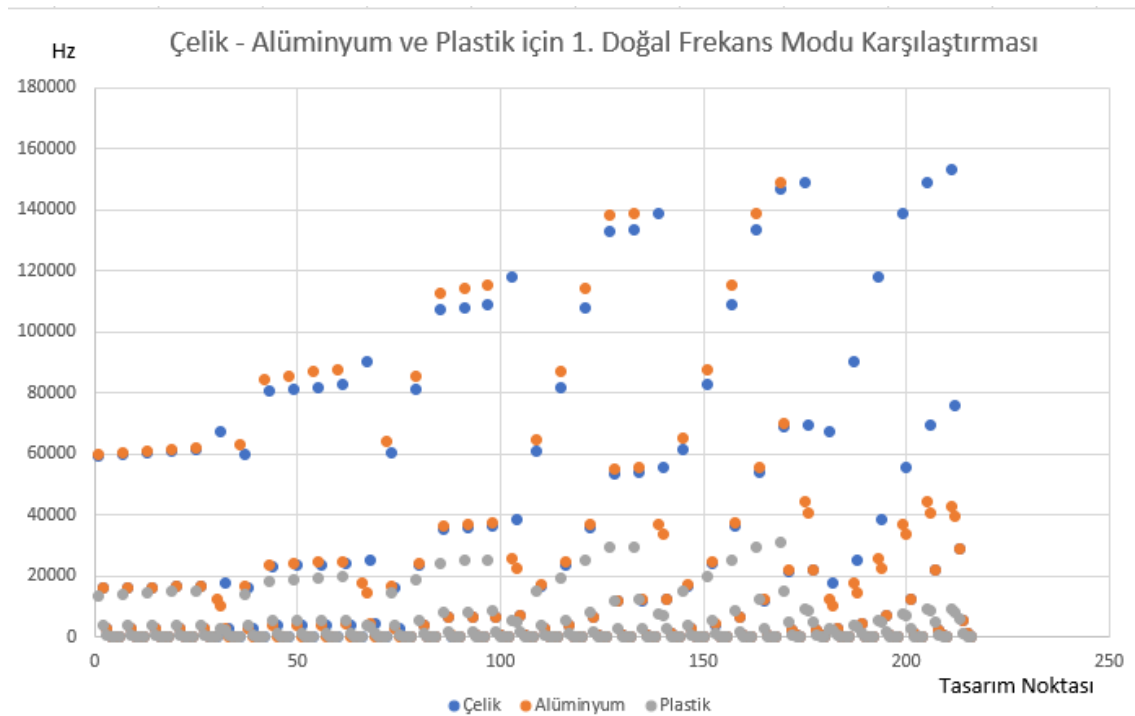
137	28.080	69.445	174.618	415.518	419.009	483.320
138	1.743	4.351	10.917	27.179	30.547	59.797
139	7,736.690	8,007.636	9,550.765	14,606.344	15,563.502	15,796.951
140	7,082.861	7,715.603	8,654.807	9,056.487	9,401.246	11,310.929
141	2,755.553	2,975.064	3,692.922	5,127.527	5,934.480	6,690.847
142	184.921	415.846	1,050.280	1,120.144	1,410.369	1,473.518
143	28.933	140.239	178.536	250.835	447.115	495.001
144	1.760	11.018	17.300	30.855	32.023	60.518
145	15,013.803	15,521.120	18,101.908	22,170.423	28,414.325	30,764.641
146	3,901.033	4,888.361	7,774.001	14,085.787	14,432.746	20,673.860
147	605.895	1,367.834	3,500.651	4,509.589	4,628.196	4,937.849
148	35.924	224.005	270.878	417.654	627.244	838.823
149	5.643	35.335	69.225	98.985	103.469	194.225
150	0.349	2.187	4.347	6.123	12.000	19.841
151	19,808.405	20,127.410	23,175.103	26,889.141	30,790.107	34,170.406
152	5,616.336	6,864.939	10,662.795	14,466.433	18,475.742	23,474.705
153	901.584	1,987.743	4,520.276	5,035.702	6,651.784	6,965.996
154	53.803	334.846	398.564	418.109	933.940	1,231.734
155	8.459	52.956	69.257	148.255	152.451	290.627
156	0.523	3.279	4.348	9.182	17.995	27.161
157	25,321.151	25,587.950	28,773.821	30,861.105	32,954.984	35,493.057
158	8,382.480	9,815.026	14,515.974	14,666.370	23,557.482	24,015.677
159	1,466.709	3,050.423	4,539.970	7,479.805	9,935.408	10,007.942
160	89.334	418.957	552.225	631.505	1,521.519	1,941.008
161	14.078	69.316	88.045	242.570	246.000	414.818
162	0.872	4.349	5.464	15.296	27.166	29.970
163	29,582.852	30,003.623	31,000.932	32,961.255	35,703.400	37,759.552
164	12,173.603	13,457.081	14,606.889	19,442.542	23,598.677	26,847.780
165	2,676.775	4,581.082	4,763.547	10,592.110	11,098.105	11,946.420
166	176.938	420.772	1,057.418	1,074.795	2,109.079	2,719.494
167	28.080	69.445	174.618	415.518	419.009	483.320
168	1.743	4.351	10.917	27.179	30.547	59.797
169	31,146.228	31,146.228	31,312.980	33,802.402	34,469.225	35,139.604
170	14,745.466	14,745.466	15,260.860	20,766.894	21,586.221	22,958.305
171	4,658.125	4,658.125	6,004.196	11,202.567	11,879.725	11,879.725
172	424.796	424.796	1,479.245	2,120.710	2,120.710	2,732.307
173	69.737	69.737	417.108	417.119	590.008	1,083.363
174	4.355	4.356	27.208	27.213	75.799	75.826
175	9,173.581	9,265.343	9,689.556	14,862.036	15,556.972	16,028.533
176	8,464.267	8,554.991	8,820.955	11,146.900	11,615.867	12,091.723
177	4,750.271	4,971.826	5,831.947	5,951.750	7,399.230	7,672.559
178	441.262	890.188	1,126.716	2,114.152	2,743.950	2,807.108
179	71.465	251.887	315.141	425.321	979.400	1,046.723
180	4.389	17.323	27.417	73.246	76.465	103.657
181	2,565.922	7,088.457	9,507.752	14,784.592	15,136.289	15,153.686

182	2,135.752	3,482.413	4,003.884	4,005.090	4,299.021	5,390.894
183	617.055	694.792	905.537	1,488.599	2,099.381	2,144.021
184	37.307	86.720	220.961	298.211	320.361	546.247
185	5.803	28.819	36.026	92.596	101.131	173.502
186	0.352	2.206	6.182	6.559	12.139	17.283
187	3,670.660	7,170.634	9,508.624	14,773.793	15,414.495	16,640.096
188	3,107.400	4,878.718	5,754.809	5,759.410	6,112.096	6,592.833
189	920.380	1,029.827	1,337.746	2,071.347	3,049.655	3,138.489
190	55.946	129.705	330.583	445.859	477.628	813.998
191	8.704	43.161	54.018	138.585	151.537	250.166
192	0.528	3.309	9.272	9.825	17.286	18.205
193	5,423.573	7,386.421	9,513.873	14,692.615	15,321.424	16,258.562
194	4,748.351	6,900.834	7,231.549	7,899.403	8,417.629	8,581.892
195	1,503.872	1,664.280	2,137.551	3,147.353	4,624.147	4,897.230
196	93.125	214.502	546.274	735.716	783.452	1,117.154
197	14.500	71.592	89.904	229.514	250.370	251.675
198	0.880	5.513	15.449	16.305	17.290	30.328
199	7,736.690	8,007.636	9,550.765	14,606.344	15,563.502	15,796.951
200	7,082.861	7,715.603	8,654.807	9,056.487	9,401.246	11,310.929
201	2,755.553	2,975.064	3,692.922	5,127.527	5,934.480	6,690.847
202	184.921	415.846	1,050.280	1,120.144	1,410.369	1,473.518
203	28.933	140.239	178.536	250.835	447.115	495.001
204	1.760	11.018	17.300	30.855	32.023	60.518
205	9,173.581	9,265.343	9,689.556	14,862.036	15,556.972	16,028.533
206	8,464.267	8,554.991	8,820.955	11,146.900	11,615.867	12,091.723
207	4,750.271	4,971.826	5,831.947	5,951.750	7,399.230	7,672.559
208	441.262	890.188	1,126.716	2,114.152	2,743.950	2,807.108
209	71.465	251.887	315.141	425.321	979.400	1,046.723
210	4.389	17.323	27.417	73.246	76.465	103.657
211	9,074.581	9,190.814	9,801.294	9,801.294	10,743.021	10,864.965
212	8,126.322	8,431.670	8,916.300	8,916.300	9,263.024	9,815.968
213	5,991.575	5,991.575	6,093.354	6,875.994	6,987.737	7,295.429
214	1,147.026	1,147.026	1,496.130	2,769.517	2,961.990	2,961.990
215	255.589	255.594	592.849	1,050.627	1,050.662	1,096.429
216	17.403	17.403	104.082	104.094	147.545	270.579

Çelik malzeme için 216 adet analiz için çözüm süresi Ansys ticari yazılımının 2019 R2 sürümünde APLD çözücü ile 22 saat sürmüştür.

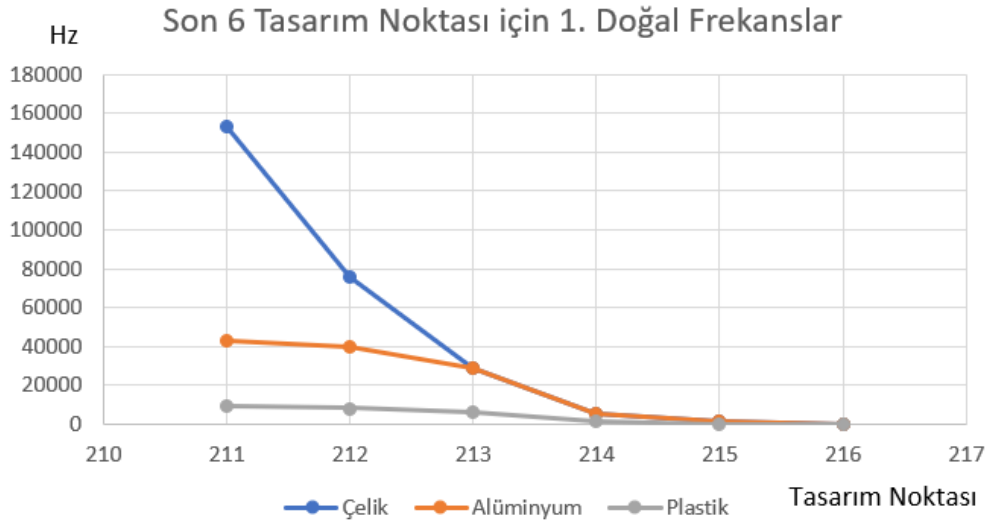
4.3.7. Doğal frekans analizi 1. Modların karşılaştırılması

Şekil 4.5.'de her bir tasarım noktası için malzemelerin ilk doğal frekanslarının karşılaştırılması verilmiştir. Şekilde, parçanın boyundaki değişimin, çelik malzemede sürekli olarak frekans ötelemesine yol açarken, plastik ve alüminyum malzeme için belirli bir aralıkta kalmıştır. Bu durumun daha iyi gözlemlenmesi için son 6 tasarım noktası için bir grafik daha oluşturulmuştur.



Şekil 4.5. Ansys ticari yazılımında çelik, alüminyum ve plastik malzemelerin doğal frekans analizlerinin 1. Modlarının karşılaştırılması

Şekil 4.6'da 211, 212, 213, 214, 215 ve 216'nci tasarım noktalarında parçaların ilk doğal frekans değerleri verilmiştir. Şekil 4.6.'da daha iyi gözlemleneceği üzere, parça boyutu büyüdükçe, çelik malzemelerin frekansları lineer olmayan şekilde azalmıştır. Alüminyum ve plastik malzemede boyut değişimi lineer olarak etki etmiştir. Buradan çıkarılabilecek sonuç çok büyük ve uzun yapılarda çelik malzeme kullanımı doğal frekans noktalarını doğrusal olmayan şekilde düşürdüğü için rüzgâr ve deprem yükü ile yapının rezonansa girmesi kolaylaşmaktadır. Diğer taraftan en boy ve uzunluk olarak birbirine yakın değerler ile üretilecek yapılarda rezonans durumu oluştuğunda yapının çelik tercih edilmesi daha tutarlı bir davranış olacaktır.



Şekil 4.6. Son 6 tasarım noktası için üç farklı malzemeye ait 1. Doğal frekans noktaları

4.3.8. Analiz sonuçlarının yapay sinir ağlarına öğretilmesi

Statik ve doğal frekans analiz sonuçları bir excel dosyasında birleştirilmiş ve veri çerçevesi oluşturularak yapay sinir ağlarında eğitilmiştir. Bölüm 3.5.1’de belirtildiği gibi, çıkış katmanı iki farklı grup nöronundan oluşmaktadır. Bu sebeple eğitimin maliyet değeri ve doğruluk değeri iki çıkış grubu için ayrı ayrı hesaplanmaktadır. YSA’nın eğitim sonuçları Çizelge 4.9’da iki farklı analiz grubu için verilmiştir.

Çizelge 4.9. Analiz sonuçları ile eğitimi yapılan YSA’nın maliyet ve doğruluk değerleri

Çıkış katmanı	Eğitim maliyet değeri	Eğitim doğruluk oranı (%)	Validasyon maliyet değeri	Validasyon doğruluk oranı (%)	Epoch (devir)	Eğitim Süresi (sn)
Statik grup	0.012	99.76	0.028	99.36	500	250
Modal grup	0.068	92.34	0.079	91.23	500	768

Eğitim süresi, Statik grup için 250 sn sürerken Modal grup için 768 saniye sürmüştür. Bu durum, statik grupta 2, modal grupta 6 değişken YSA’ya öğretildiği içindir. Yine de bu durumun öğrenme oranını büyük oranda etkilemediği gözlemlenmiştir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Bu tez çalışmasında, parça veya sistem tasarımlarının sonlu elemanlar analizi ile bilgisayar ortamında mukavemet ve mekanik davranışlarının simülasyonlarının daha kolay yapılabilmesi adına algoritmalar geliştirilmiştir. Çalışmada parçanın malzemesinin tanınabilmesi için bölüm 3.'de evrişimli sinir ağları ile derin öğrenme yapabilen bir algoritma tasarlanmıştır ve bölüm 4.'de test edildiği üzere %86 doğruluk ile çalıştığı gözlemlenmiştir.

Benzer şekilde görüntü işleme yöntemi ile objeyi sınıflandırmak için tasarlanan algoritma ise %93 doğruluk oranı ile çalışmaktadır.

Tez çalışmasında, optik tarama cihazları, derinlik algılayabilen kamera sistemleri ve LIDAR cihazları için üç boyutlu objeleri sınıflandırabilen bir algoritma yazılmıştır. 4 farklı obje için test edilen derin öğrenme algoritması %97 doğruluk oranı ile çalışmıştır. Bu kısımda objelerin üç boyutlu veriden nasıl yapay sinir ağlarına uygun dönüşümün yapılacağı da detaylı olarak anlatılmıştır.

Görüntü işleme ve üç boyutlu parça sınıflandırma yöntemlerinde, aktivasyon fonksiyonu olarak "softmax" fonksiyonunun daha başarılı sonuç verdiği ve öğrenme işleminin daha kısa sürdüğü tespit edilmiştir.

Çalışmada, bir cismin üç eksenindeki boyutları ve malzeme çeşidi ile özelleştirilmiş bir yapay sinir ağı eğitilmiştir. Bu çalışmada, farklı sonuçlar için, tek bir yapay sinir ağının, çıkış katmanları ayrıştırılarak eğitilebileceği tespit edilmiştir. Tasarlanan bu YSA, statik analiz sonuçları için %99.36 ve doğal frekans sonuçları için %91.23 doğruluk oranı ile çalışmaktadır.

5.2 Öneriler

Çalışmanın üç boyutlu kamerası bulunan bir sistemde bütün sınır koşulları tanımlanarak yapılması sistemin bütünleşmesi adına fayda sağlayacaktır. Diğer yandan, statik ve doğal frekans analizlerinin yanında, burkulma, termal genişleme, harmonik cevap, rastgele titreşim ve lineer olmayan davranış analizleri sonuçlarının da tek bir

YSA'da toplanması, gerçek zamanlı ve çözümlene yapmadan çalışan bir algoritmanın oluşmasına büyük katkı sağlayacaktır.

Çalışmada algoritma sisteminin davranışının ayrı ayrı görülmesi için, görüntü işleme, katı obje sınıflandırma ve analiz sonuçları YSA'ları ayrı olarak tasarlanmış ve eğitilmiştir. Analiz yapılacak cihazda gerekli donanımın bulunması halinde bu üç YSA birleştirilerek, gerekli ince ayarlar ve giriş çıkış katmanlarının bölüştürülmesi ile ortak bir sinir ağı oluşturulabilir. Böylece tez çalışması, mobil cihazlarda çalışabilen, her kullanıcının incelemek istediği parçayı göstermesi suretiyle çalışabilen bir yapay zeka sistemi geliştirilebilecektir.

Görüntü işleme ve malzeme tanımlama algoritmaları, AlexNet, VGG19 ve ResNet gibi modellerin üzerine tekrar öğrenim yaptırılarak doğruluk oranları artırılabilir. Bu hazır modellerde binlerce sınıf olduğu için, transfer learning (öğrenme transferi) yapılmadan önce, çalışma kümesinin hazır modele uygunluğu sorgulanmalıdır. Bu tez çalışmasında daha çok sanayi tabanlı üretilen parçalar inceleneceği için sıfırdan veri seti ve derin öğrenme algoritması hazırlanmıştır.

KAYNAKLAR

- Arslan, M., 2014, Hands on applied finite element analysis, 1. Baskı, Gebze Teknik Üniversitesi
- Augspurger, T., 2015, “Python Data Analyst Library [online], Açık kaynak, <https://pandas.pydata.org/about.html> [Ziyaret Tarihi: 01 Şubat 2018]
- Bier, H., Heule, M., 2007, Prototypes for automated architectural 3d-layout, *13th International Conference Virtual Systems and Multimedia, VSMM 2007*, Australia, 203-214.
- Fahlman, S.E., 1988, An Empirical Study of Learning Speed in Backpropagation Networks”, *Computer Science Technical Report, CMU-CS-88-162*, Carnegie Mellon University, Pittsburgh
- Goodfellow, I., Bengio, Y., and Courville, A., 2016, Deep Learning [online]. MIT Press, <http://www.deeplearningbook.org>. [Ziyaret Tarihi: 15 Ekim 2018]
- Harish, A., 2019, Why the Tacoma Narrows Bridge collapsed: An engineering analysis [online], blog, Simscale, <https://www.simscale.com/blog/2018/07/tacoma-narrows-bridge-collapse/> [Ziyaret Tarihi: 6 Mayıs 2019]
- Hedge, V. , 2016, FusionNet: 3D Object Classification Using Multiple Data Representations [online], Web kütüphanesi, Stanford University, https://www.researchgate.net/publication/305440736_FusionNet_3D_Object_Classification_Using_Multiple_Data_Representations [Ziyaret Tarihi: 5 Mart 2019]
- Hinton, G., 2013, “Deep learning,” *Nature*, 521, 436–444.
- Hinton, G., 2016, Layer normalization, *Computing Research Repository*,
- Hinton, G., 1990, A fast learning algorithm for deep belief nets, *Neural Computation*, 18, 1527–1554.
- Hochreiter, S., Schmidhuber, J., 1997, Long short-term memory. *Neural Comput.* 9(8), 1735-1780.
- Khalid, N., 2018, Off-line Arabic Hand-Writing Recognition Using Artificial Neural Network with Genetics Algorithm, *International Arab Journal of Information Technology (IAJIT)*, 1(1), 55-58.
- Krizhevsky, A., Sutskever, I., Hinton, G., 2012, ImageNet Classification with Deep Convolutional Neural Networks, *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1097-1105.
- Liang, L., Liu, M., Martin, C., Sun, W., 2018, A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis, *Journal of The Royal Society Interface*, 15(138), 1-10.

- Maturana, D., and Scherer, S., 2015, VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition, International Conference on Intelligent Robots and Systems, 922-928.
- Mutlu, H., E., 2018, "Classification of Hyperspectral and LiDAR Data with Deep Learning", Master's Thesis, Hacettepe University, Ankara, Turkey
- Otair, M., 2006, "Solving Xor Problem Using An Optical Backpropagation Neural Networks" *2nd Jordanian International Conference on Computer Science and Engineering (JICCSE2006)*, 298 – 301.
- Rosenblatt, F., 1958, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, 65(6), 386–408.
- Şeker, S. E., 2012, İstatiksel Normalleştirme [online], Blog, <http://bilgisayarkavramlari.sadievrenseker.com/2012/01/29/istatistiksel-normallestirme-statistical-normalisation/>, [Ziyaret Tarihi: 20 Eylül 2018]
- Tippabhotla, S. K., 2010, Introduction to Finite Element Analysis using Ansys, [online], https://static1.squarespace.com/static/57e8c28ff5e23123e5a12f92/t/58d49d3d4402432cd58b81e2/1490328915187/FEM_DManD_Class_SasiKumar.pdf, [Ziyaret Tarihi: 30 Ekim 2017]
- Wang, Z., Xiao, D., Fang, F., 2018, Model identification of reduced order fluid Dynamics systems using deep learning, *Int J Numer Meth Fluids*, 86, 255–268.

EKLER

EK-1 Görüntü işleme ve ESA algoritmaları

```

import os,cv2
import numpy as np
import matplotlib.pyplot as plt

from sklearn.utils import shuffle
from sklearn.cross_validation import train_test_split

from keras import backend as K
K.set_image_dim_ordering('th')

from keras.utils import np_utils
from keras.models import Sequential
from keras.optimizers import SGD,RMSprop,adam,Adadelta
from keras.preprocessing import image
import helper
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
###

PATH = os.getcwd()
data_path = PATH + '/data'
data_dir_list = os.listdir(data_path)

img_rows=128
img_cols=128
num_channel=1
num_epoch=2

num_classes = 5

img_data_list=[]

for dataset in data_dir_list:
    img_list=os.listdir(data_path+'/'+ dataset)
    print ('veriseti yüklendi-'+'{}\n'.format(dataset))
    for img in img_list:
        input_img=cv2.imread(data_path + '/' + dataset + '/' + img )
        input_img=cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
        input_img_resize=cv2.resize(input_img,(128,128))
        img_data_list.append(input_img_resize)

img_data = np.array(img_data_list)
img_data = img_data.astype('float32')
img_data /= 255
print (img_data.shape)

if num_channel==1:
    if K.image_dim_ordering()=='th':
        img_data= np.expand_dims(img_data, axis=1)

```

```

    print (img_data.shape)
else:
    img_data= np.expand_dims(img_data, axis=4)
    print (img_data.shape)

else:
    if K.image_dim_ordering()=='th':
        img_data=np.rollaxis(img_data,3,1)
        print (img_data.shape)

###

###

num_classes = 5

num_of_samples = img_data.shape[0]
labels = np.ones((num_of_samples,),dtype='int64')

labels[0:250]=0
labels[250:500]=1
labels[500:750]=2
labels[750:1000]=3
labels[1000:]=4

names = ['car','chair','gear','plane','undefined']

Y = np_utils.to_categorical(labels, num_classes)

x,y = shuffle(img_data,Y, random_state=2)
print(x)

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)

###

input_shape=img_data[0].shape

model = Sequential()

model.add(Conv2D(32, (6,6),input_shape=input_shape,padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(16, (6, 6)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))
"""
model.add(Conv2D(16, (10, 10)))
model.add(Activation('relu'))
#model.add(Convolution2D(64, 3, 3))
#model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(5, 5)))
model.add(Dropout(0.5))
"""

```

```

model.add(Flatten())
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dropout(0.5))
"""

model.add(Dense(8))
model.add(Activation('relu'))
model.add(Dropout(0.5))
"""

model.add(Dense(num_classes))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy', optimizer='Adadelta', metrics=["accuracy"])

#özet
model.summary()
###
model.fit(X_train, y_train, batch_size=32, epochs=30, verbose=1, validation_data=(X_test,
y_test))

###

score = model.evaluate(X_test, y_test, verbose=0)

print('Test Loss:', score[0])
print('Test accuracy:', score[1])

#test bölümü

test_image = X_test[0:1]
print (test_image.shape)

print(model.predict(test_image))
print(model.predict_classes(test_image))
print(y_test[0:1])

###
test_image = cv2.imread('data/chair/chair(1).jpg')
test_image=cv2.cvtColor(test_image, cv2.COLOR_BGR2GRAY)
test_image=cv2.resize(test_image,(128,128))
test_image = np.array(test_image)
test_image = test_image.astype('float32')
test_image /= 255
print (test_image.shape)

if num_channel==1:
if K.image_dim_ordering()=='th':
test_image= np.expand_dims(test_image, axis=0)
test_image= np.expand_dims(test_image, axis=0)
print (test_image.shape)
else:
test_image= np.expand_dims(test_image, axis=3)
test_image= np.expand_dims(test_image, axis=0)
print (test_image.shape)

else:

```

```

if K.image_dim_ordering()=='th':
    test_image=np.rollaxis(test_image,2,0)
    test_image= np.expand_dims(test_image, axis=0)
    print (test_image.shape)
else:
    test_image= np.expand_dims(test_image, axis=0)
    print (test_image.shape)

```

```

print((model.predict(test_image)))
print(model.predict_classes(test_image))

```

```

Y_pred = model.predict(X_test)
y_pred = np.argmax(Y_pred, axis=1)
helper.tahmin_matrisi(y_test,y_pred)

```

```

from keras.models import model_from_json
from keras.models import load_model

```

```

model_json = model.to_json()
with open("model.json", "w") as json_file:
    json_file.write(model_json)
model.save_weights("model.h5")

```

```

json_file = open('model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)

```

```

loaded_model.load_weights("model.h5")

```

```

model.save('model.hdf5')
loaded_model=load_model('model.hdf5')

```

EK-2 CAD dosya formatlarından OFF formatına geçiş python kodları

```
import os
import sys
print(sys.version)
os.chdir(r'S:\.....')

import time

file_num = 2

k = 0

while k <= file_num:
    DocumentOpen.Execute(r"S:\.....\train(" + String(k) + ").sldprt")
    time.sleep(10)

    DocumentSave.Execute(r"S:\.....\train(" + String(k) + ").wrl",FileSettings1)
    time.sleep(10)

    targets = Selection.SelectAll()
    Delete.Execute(targets)

k = k + 1
```

- Derleyici SpaceClaim ticari yazılımı uyumludur.
- ile belirtilen yere dosya konumu yazılmalıdır.

Ek-3 OFF dosya formatının Bivox dosyalarına dönüştürülmesi python kodları

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

k = 1
while k < 1201:

    print(k)
    # Open File

    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
```

Ek-4 Binox dosyalarının Numpy matrislerine alınması python kodları

```

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)

#%%
narray = narray.astype(int)
z = narray.reshape(512,512)
import matplotlib.pyplot as plt
plt.subplot(1, 2, 1)
plt.imshow(z.T)
plt.axis('off')

#%%
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)

```

Ek-5 Üç boyutlu evrişimli sinir ağının python dilinde kodlanması

```

import numpy as np

x_class = np.load("x_class.npy")

Y = np.load("y_class.npy")

###
#Expand Dim

x_train_e = np.expand_dims(x_class, axis=4)

###
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_train_e, Y, test_size=0.2, random_state=2)
###
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.layers import Input, Dense, Flatten
from keras.optimizers import SGD
from keras.layers.normalization import BatchNormalization
from keras.layers import Conv3D, MaxPooling3D, AveragePooling3D

input_shape = (128,128,128,1)
model = Sequential()

model.add(Conv3D(2, (6,6,6),input_shape=input_shape,padding='valid'))
model.add(Activation('relu'))
model.add(MaxPooling3D(pool_size=(6,6,6),padding='valid'))
model.add(MaxPooling3D(pool_size=(3,3,3),padding='valid'))
model.add(Activation('relu'))
model.add(AveragePooling3D(pool_size=(2,2,2),padding='valid'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dropout(0.3))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dropout(0.3))
model.add(Dense(4))
model.add(Activation('softmax'))

model.compile(optimizer='Adadelta', loss = 'categorical_crossentropy', metrics=['accuracy'])
model.summary()

model.fit(x_train, y_train, batch_size=2, epochs=1, verbose=1, validation_data=(x_test, y_test))

y_pred =model.predict(x_test,batch_size=2,verbose=1)

y_pred2 =model.predict_classes(x_test,batch_size=2,verbose=1)

```

- Konvolüsyon filtresiz kodlama:

```

import numpy as np

x_class = np.load("x_class.npy")

Y = np.load("y_class.npy")

###
#Expand Dim

x_train_e = np.expand_dims(x_class, axis=4)

###
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_train_e, Y, test_size=0.2, random_state=2)
###
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.layers import Input, Dense, Flatten
from keras.optimizers import SGD
from keras.layers.normalization import BatchNormalization
from keras.layers import Conv3D, MaxPooling3D, AveragePooling3D

input_shape = (128,128,128,1)
model = Sequential()

model.add(MaxPooling3D(pool_size=(3,3,3),input_shape=input_shape,padding='valid'))
model.add(Activation('relu'))

model.add(MaxPooling3D(pool_size=(3,3,3),padding='valid'))
model.add(Activation('relu'))
model.add(AveragePooling3D(pool_size=(2,2,2),padding='valid'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dropout(0.3))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dropout(0.3))
model.add(Dense(4))
model.add(Activation('softmax'))

model.compile(optimizer='rmsprop', loss = 'categorical_crossentropy', metrics=['accuracy'])
model.summary()
###
model.fit(x_train, y_train, batch_size=2, epochs=20, verbose=1, validation_data=(x_test,
y_test))

```

Ek-6 Sonlu elemanlar analizi için geliştirilen YSA'nın python dilinde kodlanması

```

import pandas as pd
import numpy as np
from keras.models import *
from keras.layers import Dense, Dropout, Activation
from keras.layers.normalization import BatchNormalization
from keras.engine.input_layer import Input
from keras.models import Model
import keras

#%%
df = pd.read_csv("train_def.csv",sep=';')
st = pd.read_csv("train_von.csv",sep=';')

#%%

dimesions = np.array([[10,20,1]],dtype=float)
forces = np.array([[1,2,5,10,20,30,40,50]],dtype=float)
vector = np.arange(1,21,dtype=float)

y_val = df.N1_DEF.values
y_all_df = df.values

y_all_st = st.values
#%%

import matplotlib.pyplot as plt

plt.plot(vector,y_all_df)

plt.xlabel("Node Number")

plt.ylabel("Deformation (mm)")

plt.show()

#####

plt.plot(vector,y_all_st)

plt.xlabel("Node Number")

plt.ylabel("Von-Mises Stress (MPa)")

plt.show()

#%%
# Normalization

df_norm = (y_all_df - np.min(y_all_df))/(np.max(y_all_df)-np.min(y_all_df))
st_norm = (y_all_st - np.min(y_all_st))/(np.max(y_all_st)-np.min(y_all_st))
forces_norm = (forces - np.min(forces))/(np.max(forces)-np.min(forces))

df_norm_t = df_norm.T
st_norm_t = st_norm.T
forces_norm_t = forces_norm.T

```

```

###

inp = Input(shape=(1,))

x = Dense(64, activation='relu')(inp)
x = Dropout(.5)(x)
x = Dense(32, activation='relu')(x)

deformation = Dense(20, activation='softmax')(x)
stress = Dense(20, activation='softmax')(x)
model = Model(inputs=inp, outputs=[deformation,stress])
model.compile(optimizer = "rmsprop", loss = 'mse')
model.summary()

model.fit(forces_norm_t, [df_norm_t,st_norm_t], nb_epoch=500, verbose=1)

tahmin = model.predict(test)

tahmin_df = tahmin[0].T
tahmin_st = tahmin[1].T

plt.plot(vector,tahmin_df)

plt.xlabel("Node Number")

plt.ylabel("Deformation (mm)")

plt.show()

#####

plt.plot(vector,tahmin_st)

plt.xlabel("Node Number")

plt.ylabel("Von-Mises Stress (MPa)")

plt.show()

###
#karsilastirma
con[0:20,0] = df_norm_t[0].T
con[0:20,1] = tahmin_df.reshape(20,)
plt.plot(vector,con)

plt.xlabel("Node Number")

plt.ylabel("Deformation (mm)")

plt.show()

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ahmet Okudan
Uyruğu : Türkiye Cumhuriyeti Vatandaşı
Doğum Yeri ve Tarihi : Selçuklu 1991
Telefon : +90 555 543 17 00
Faks : ---
E-Posta : ahmetokudan27@gmail.com

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: Büyükkoyuncu Anadolu Lisesi			2009
Üniversite	: Sakarya Üniversitesi Makine Mühendisliği Bölümü			2014
Yüksek Lisans	: Konya Teknik Üniversitesi Makine Mühendisliği Bölümü			

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2014-2015	FMC Hid. Sist. Oto. Mak. San. Tic. Ltd. Şti.	Ar-Ge Mühendisi
2015-2019	Yıldız Pul Oto. Mot. Parç. A.Ş	Ar-Ge Mühendisi
2019-Devam ediyor	Numesys İleri Mühendislik Hizmetleri A.Ş.	Kıdemli Uygulama Mühendisi

UZMANLIK ALANI

Sonlu Elemanlar Analizi, Bilgisayar Destekli Tasarım, Topoloji Optimizasyonu, Derin Öğrenme, Makine Öğrenmesi, Titreşim, Yorulma, ERP, MRP, Ürün Veri Yönetimi, 5S, Ürün Geliştirme, Kalite Kontrol

YAYINLAR

Okudan, A., Şahin, Ö., S., 2019, Yapay Sinir Ağları ile Üç Boyutlu Obje Sınıflandırmada Konvolüsyon Filtrelerinin Etkisinin İncelenmesi, *Selçuk Teknik Dergisi* (Yüksek Lisans tezinden yapılmıştır)