



# BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ

JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGIES

**HAZİRAN (JUNE) 2020**  
**Cilt (Volume) : 1**  
**Sayı(issue) : 1**

**EDİTÖR: DOÇ. DR. ERDİNÇ AVAROĞLU**  
**ISSN 2717-8579**





*Bilgisayar Bilimleri ve Teknolojileri Dergisi*

# **BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ**

**CİLT 1, SAYI 1**

**ISSN: 2717-8579**

**HAZİRAN 2020**



Bilgisayar Bilimleri ve Teknolojileri Dergisi

#### Dergi Hakkında

Bilgisayar Bilimleri ve Teknolojileri Dergisi bilim ve teknolojideki gelişmelere paralel olarak bilgisayar bilimleri ve teknolojileri alanında yeni gelişmelerle ilgili yapılan çalışmaları yayınlayan bir dergidir.

#### Amaç & Kapsam

BIBTED Dergisi,

✚ Bilgisayar Bilimleri ve Teknolojileri Dergisinin amacı bilgisayar alanında yapılan özgün çalışmaları yayınlamaktır. Yazım kurallarına uygun olarak hazırlanan eser, dergi editörlüğüne değerlendirme için hakemlere gönderilir. Bilgisayar Bilimleri ve Teknolojileri Dergisinde **KÖR HAKEMLİK** uygulaması mevcuttur. Yayımlanmasına, hakemlerin görüşü doğrultusunda Dergi Editör ve Yayın Kurulu karar verir. Gönderilen makaleler yayımlansın veya yayımlanmasın iade edilmez. Dergimizde yayınlanan yazıların her türlü sorumluluğu (bilimsel, mesleki, hukuki, etik vb.) yazarlara aittir. Yayınlanan yazıların telif hakkı dergiye aittir ve referans gösterilmeden aktarılamaz. Araştırmacılar arasındaki bilimsel iletişimi oluşturmak amacıyla aşağıda nitelikleri açıklanan, başka bir yerde yayımlanmamış makaleler Türkçe ve İngilizce olarak kabul edilmekte ancak Türkçe Kabul edilen makalenin özetinin İngilizce de basılması zorunluluğu vardır.

Aşağıdaki türlerdeki makaleler dergide yayına kabul edilmektedir:

- ✚ **Araştırma makalesi:** Özgün bir araştırmayı sonuçlarıyla birlikte sunan makale,
- ✚ **Derleme makale:** Bilgisayar Mühendisliği alanında belli bir konuda yeterli sayıda bilimsel makaleyi tarayıp, özetleyen, değerlendirme yapan ve bulguları yorumlayan makale,
- ✚ **Endüstriyel makale:** Bu alanda endüstride yapılan araştırma ve geliştirilen yeni ürün veya teknolojilerin açıklandığı makale,
- ✚ **Tez çalışması:** Lisansüstü düzeyde yapılan özgün bir tez çalışmasının genişletilmiş özetini içeren yazı,
- ✚ **Kitap yorumu:** Bilgisayar mühendisliği alanında yayınlanmış yeni bir kitabın tanıtılması ve değerlendirilmesi.
- ✚ **Kısa Bildiri:** Yapılan bir araştırmanın önemli bulgularını açıklayan yeni bir yöntem veya teknik tanımlayan yazılar.

Bütün yazıların Telif Hakkı Devri, yazarlarına bir form gönderilmek suretiyle alınır. Telif Hakkı Devir Formu göndermeyen yazarların yayınları işleme konmaz. Yayımlanmasına karar verilen yazılar üzerine yazarlarınca hiçbir eklenti yapılamaz.

Her yazı konusu ile ilgili en az iki hakeme gönderilerek şekil ve içerik bakımından incelenir. Dergide yayımlanabilecek nitelikteki yazılar dizgisi yapıldıktan sonra, yazarlarına gönderilerek baskı öncesi gözden istenir. Makale içinde, dergide basıldığı haliyle gözükken hataların sorumluluğu yazarlarına aittir. Hata, editörlük ofisinden kaynaklandığı takdirde düzeltme yayımlanabilir.

#### Derginin Kapsamı;

Bilgisayar Bilimleri ve Teknolojileri Dergisinin kapsamı, akıllı sistemler, algoritmalar, benzetim, bilgisayar ağları, bilgisayar grafiği, bilgisayarla görme, bilgisayar mimarisi, bilgiye erişim, bilimsel hesaplama, bilişim güvenliği, biyoenformatik, kriptografi, paralel işleme, doğal dil işleme donanım, görüntü işleme, hesaplama kuramı, işaret işleme, işletim sistemleri, makine öğrenmesi, mobil sistemler, modelleme, tıbbi bilişim, veri madenciliği, veri tabanı sistemleri, yazılım mühendisliği, siber güvenlik, yapay zeka dahil olmak üzere bilgisayar bilimleri ve teknolojilerin tüm alanları içerir.

#### Yayımlanma Sıklığı

Yılda 2 sayı

#### ISSN

2717-8579

#### WEB

<https://dergipark.org.tr/tr/pub/bibted>

#### İletişim

[eavaroglu@mersin.edu.tr](mailto:eavaroglu@mersin.edu.tr) / [ttuncer@firat.edu.tr](mailto:ttuncer@firat.edu.tr) / [kemaladem@gmail.com](mailto:kemaladem@gmail.com)



Bilgisayar Bilimleri ve Teknolojileri Dergisi

---

#### EDİTÖR

**Doç. Dr. Erdiç AVAROĞLU**

Mersin Üniversitesi, Mühendislik Fakültesi / Bilgisayar Mühendisliği, Mersin

---

#### EDİTÖR YARDIMCILARI

**Doç. Dr. Taner TUNCER**

Fırat Üniversitesi, Mühendislik Fakültesi / Bilgisayar Mühendisliği, Elâzığ

**Dr. Öğr. Üyesi. Kemal ADEM**

Aksaray Üniversitesi, İktisadi ve İdari Bilimler Fakültesi / Yönetim Bilişim Sistemleri, Aksaray

---

#### EDİTÖR KURULU

- **Prof. Dr. Zeki YETKİN, MERSİN ÜNİVERSİTESİ**
- **Doç. Dr. İsmail KOYUNCU, AYFON KOCATEPE ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Murat TUNA, KIRKLARELİ ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah ELEWİ, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah Erhan AKKAYA, İNÖNÜ ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Lutfiye KUŞAK, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Fatma Bünyal ÜNEL, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Çiğdem ACI, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Soner KIZILOLUK, TURGUT ÖZAL ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Selman YAKUT, TURGUT ÖZAL ÜNİVERSİTESİ**

---

#### DANIŞMA KURULU

- **Prof. Dr. Ahmet Bedri ÖZER, FIRAT ÜNİVERSİTESİ**
- **Prof. Dr. Murat YAKAR, MERSİN ÜNİVERSİTESİ**
- **Doç. Dr. Fatih ÖZKAYNAK, FIRAT ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Mehmet ACI, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Murat TUNA, KIRKLARELİ ÜNİVERSİTESİ**
- **Doç. Dr. İsmail KOYUNCU, AFYON KOCATEPE ÜNİVERSİTESİ**

---

#### DİL EDİTÖRLERİ

- **Dr. Öğr. Üyesi Abdullah ELEWİ, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah Erhan AKKAYA, İNÖNÜ ÜNİVERSİTESİ**
- **Arş. Gör. Dr. Dilek SABANCI, GAZİOSMANPAŞA ÜNİVERSİTESİ**

---

#### MİZANPAJ

- **Arş. Gör. Semih KAHVECİ, MERSİN ÜNİVERSİTESİ**



Bilgisayar Bilimleri ve Teknolojileri Dergisi

# İçindekiler

## Contents

### ARAŞTIRMA MAKALELERİ; RESEARCH ARTICLES;

#### S.No

- 
- |       |  |
|-------|--|
| 1-9   | <i>Bulanık Mantık Üyelik Fonksiyonlarının FPGA Üzerinde Gerçeklenmesi</i><br><i>Implementation of Fuzzy Logic Membership Functions on FPGA</i><br><b>Fatih KARATAŞ, İsmail KOYUNCU, Murat TUNA, Murat ALÇIN</b>  |
| 10-15 | <i>Nesne Sınıflandırma İşlemi için Tensor İşleme Birimi ve CPU Performansı</i><br><i>Comparing Tensor Processing Unit and CPU performance for Object Classification</i><br><b>Ayşe Ayyüce DEMİRBAŞ &amp; Ahmet ÇINAR</b>   |
| 16-21 | <i>Rastgele Seçim Tabanlı Yer Değiştirme Kutularının Performans İyileştirmesi için Son İşlem Algoritmaları</i><br><i>Post-Processing Algorithms for Performance Improvement of Substitution Boxes Based on Random Selection</i><br><b>Yaşar Selim BAHÇECİ &amp; Fatih ÖZKAYNAK</b> |
| 22-27 | <i>Ortaöğretimde Öğrenim Gören Öğrenci Performanslarının Veri Madenciliği Yöntemleri ile Tahmin Edilmesi</i><br><i>Predicting the Performance of Students Studying in Secondary Education Using Data Mining Methods</i><br><b>Senem Hazel BAŞER, Onur HÖKELEKLİ, Kemal ADEM</b>    |
| 28-41 | <i>Hibrit Evrişimli Sinir Ağı Modelleri ile Tıbbi Görüntü Sınıflandırması</i><br><i>Medical Image Classification with Hybrid Convolutional Neural Network Models</i><br><b>Osman Doğuş GÜLGÜN &amp; Hamze EROL</b>   |



Araştırma Makalesi

## Bulanık Mantık Üyelik Fonksiyonlarının Fpga Üzerinde Gerçeklenmesi

Fatih Karataş<sup>\*1</sup>, İsmail Koyuncu<sup>2</sup>, Murat Tuna<sup>3</sup>, Murat Alçın<sup>4</sup>

<sup>1</sup>Afyon Kocatepe University, Institute of Natural and Applied Sciences, Afyonkarahisar, Turkey

<sup>2</sup>Afyon Kocatepe University, Technology Faculty, Department of Electrical Electronics Engineering, Afyonkarahisar, Turkey

<sup>3</sup>Kırklareli University, Technical Sciences Vocational School, Department of Electric, Kırklareli, Turkey

<sup>4</sup>Afyon Kocatepe University, Technology Faculty, Department of Electrical Electronics Engineering, Afyonkarahisar, Turkey

### ÖZ

#### Anahtar Kelimeler:

Bulanık Mantık  
Bulanıklaştırma  
Üyelik Fonksiyonları  
FPGA  
VHDL

Bu çalışmada, gerçek zamanlı bulanık mantık uygulamaları için gauss, triangular (üçgen), trapezoidal (yamuk), generalized bell-shaped (genelleştirilmiş çan eğrisi) üyelik fonksiyonlarının matematiksel modeli referans alınarak 32-bit IEEE-754-1985 kayan noktalı sayı standardına uygun FPGA (Field Programmable Gate Array-Alan Programlanabilir Kapı Dizileri) tabanlı gauss, üçgen, yamuk ve çan eğrisi üyelik fonksiyonu üniteleri tasarlanmıştır. Tasarım VHDL (Very High Speed Integrated Circuit Hardware Description Language-Çok Yüksek Hızlı Tümlleşik Devre Donanımı Tanımlama Dili) ile kodlanmış ve Xilinx ISE Design Suite programı kullanılarak sentezlenmiştir. Tasarlanan ünitelerin test edilebilmesi amacıyla testbench dosyaları oluşturulmuştur. Bu test verilerinden elde edilen sonuçlar ile Matlab tabanlı nümerik sonuçlar karşılaştırılmış ve hata değerleri hesaplanmıştır. Place&Route işleminin ardından FPGA tabanlı Gauss Üçgen, Yamuk ve Çan eğrisi üyelik fonksiyonlarına ait ünitelerin maksimum çalışma frekansları elde edilmiştir. Place&Route işleminin ardından FPGA tabanlı Gauss üyelik fonksiyon ünitesinin maksimum çalışma frekansı 304.534 MHz, Üçgen üyelik fonksiyon ünitesinin maksimum çalışma frekansı 245.031 MHz, Yamuk üyelik fonksiyon ünitesinin maksimum çalışma frekansı 599.664 MHz ve Çan eğrisi üyelik fonksiyon ünitesinin maksimum çalışma frekansı 185.086 MHz olarak elde edilmiştir.

## Implementation Of Fuzzy Logic Membership Functions On FPGA

### ABSTRACT

#### Keywords:

Fuzzy Logic  
Fuzzification  
Membership Functions  
FPGA  
VHDL

In this study, FPGA (Field Programmable Gate Array) based gauss, triangular, trapezoidal and generalized bell-shaped membership function units are designed for real-time Fuzzy Logic applications in accordance with the 32-bit IEEE-754-1985 floating-point number standard by referring to the mathematical model of gauss, triangular, trapezoidal and generalized bell-shaped membership functions. The design was coded in VHDL (Very High Speed Integrated Circuit Hardware Description Language) and synthesized using Xilinx ISE Design Suite program. Testbench files were created in order to test the designed units. The results obtained from this test data were compared with Matlab-based numerical results and error values were calculated. Following the Place&Route operation, the maximum operating frequencies of the units belonging to FPGA based gauss, triangular, trapezoidal and generalized bell-shaped membership functions were obtained. After the Place&Route operation, the maximum operating frequency of FPGA based gauss membership function unit was got as 304.534 MHz, the maximum operating frequency of FPGA based triangular membership function unit was achieved as 245.031 MHz, the maximum operating frequency of FPGA based trapezoidal membership function unit was obtained as 599.664 MHz and the maximum operating frequency of FPGA based generalized bell-shaped membership function unit was got as 185.086 MHz.

#### \*Sorumlu Yazar

\*(fatihkaratas@usr.aku.edu.tr) ORCID ID 0000-0003-1877-5552  
(ismailkoyuncu@aku.edu.tr) ORCID ID 0000-0003-4725-4879  
(murat.tuna@klu.edu.tr) ORCID ID 0000-0003-3511-1336  
(muratalcin@aku.edu.tr) ORCID ID 0000-0002-2874-7048

## 1. GİRİŞ

Günümüzde, bulanık mantıkla ilgili araştırmaların odak noktası, kullanıcı konforunu en üst düzeye çıkarmak ve enerji tüketimini en aza indirmek için yenilikçi tahmin ve optimizasyon tekniklerinin sürekli olarak geliştirilmesidir. Bununla birlikte, kontrol doğruluğu ve kontrol hızı, sistem performansına katkıda bulunan ancak araştırmacılar tarafından büyük ölçüde göz ardı edilen en kritik kısımlardan biridir (Khokhar ve ark., 2020). Bulanık mantık; bilgi tabanı, bulanıklaştırma, karar verme ve durulaştırma birimleri olmak üzere dört temel bileşenden oluşmaktadır (Elmas, 2003). Bulanıklaştırma ünitesi sistemden alınan denetim giriş bilgilerini dilsel niteleyiciler olan sembolik değerlere dönüştürme işlemi yapmaktadır. Bu üniteye çeşitli üyelik fonksiyonları mevcut olup en yaygın kullanılan üçgen, trapezoidal (yamuk), generalized bell-shaped (genelleştirilmiş çan eğrisi), gauss ve sigmoidal üyelik fonksiyonlarıdır (Altaş, 1999). Kontrol doğruluğu ve kontrol hızını etkileyen parametrelerden birisi olan aktivasyon fonksiyonları bu çalışmada ele alınmış ve paralel çalışma, düşük güç tüketimi, hızlı ilk üretim, yüksek kapasite gibi özellikleriyle diğer sayısal platformlara göre ön plana çıkan FPGA çiplerinde (Koyuncu, 2018) uygulanmak üzere tasarımı gerçekleştirilmiştir.

Karar verme ve kontrol sistemi olarak tanımlanabilen bulanık mantık (Fuzzy Logic) kavramı ilk defa Lotfi A. Zadeh tarafından 1965 'te ortaya atılmıştır. Bulanık mantık, doğa ve insanın oluşturduğu yapay sistemlerin modellenmesi ve kontrolü için uygun bir yaklaşım olarak gelişmekte olan bir tekniktir. Klasik yöntemlerle modellenemeyen karmaşık sistemlerde, lineer olmayan sistemlerde ve belirsizliklerin çok olduğu sistemlerde ideal olarak kullanılmaktadır. Zadeh belirsiz kümeleri, belirsizliği ifade etme aracı olarak tanıttı ve o zamandan beri, bulanık küme teorisi farklı disiplinlerde önemli bir konu haline gelmiştir (Zadeh, 1965). Bulanık sistem teorisi, belirsiz düşünce ve karar süreçlerinin gelişen modellerine ait başlama noktası olarak kabul edilmekte ve bu nedenle; yönetim ve sosyal problemler için kullanılan insan modellerinin yapılması, otomasyon ve bilgi sistemlerinde kullanım için yüksek derecede insan yeteneklerinin taklidi, insan ve makineler arasındaki insan merkezli ara birimlerin oluşumu, risk analizi, tahmin ve fonksiyonel cihazların gelişimi gibi diğer sosyal ve yapay zeka uygulamalarında kullanılmaktadır (Şen, 2012). Bulanık mantık ilk kez 1973 yılında, Londra 'da Ebrahim H. Mamdani tarafından bir buhar makinasında uygulanmıştır. Ticari olarak ise ilk kez, 1980'de Danimarka'daki bir çimento fabrikasının fırınına kontrol etmek için kullanılmıştır (Mamdani ve Assilian, 1975). Bulanık mantık ile hazırlanan bir

sistem, bilgisayar desteğinde, sensörlerden gelen ısı ve maddelere ait bilgileri alarak ve "feed-back" (geri besleme) metoduyla değişkenleri kontrol ederek büyük oranda enerji tasarrufu sağlamıştır (Klir ve Yuan, 1995). Uluslararası Bulanık Sistemler Derneği'nin 1987'de Tokyo'da düzenlediği bir konferansta bir mühendis, bulanık mantıkla programladığı bir robota, bir çiçeği ince bir çubuğun üzerinde düşmeyecek şekilde bıraktırmayı başarmıştır. Robota ait bir devre çıkarılmasına rağmen robot beklenmedik bir şekilde yine aynı hassaslıkla çiçeği düşürmeden çubuğun üzerine bırakmıştır (Mukaidono ve ark., 1987). Kısacası bulanık mantık sistemleri, yetersiz bilgi temin edilse dahi tıpkı insanların yaptığı gibi bir tür "sağduyu" kullanarak işlemleri gerçekleştirebilmektedir.

Bulanık mantığın kullanıldığı bazı önemli uygulamalar; helikopterler için bulanık tabanlı uçuş desteği (Sugeno, 1985), raylı metro sistemlerinde bulanık mantık tabanlı sürüş rahatlığı, duruş mesafesinin kesinliğinin ve ekonomikliğinin geliştirilmesi (Yasunobu ve ark., 2002), el yazısı sembollerin çevrimiçi tanınmasında bulanık mantık uygulanması (Fitzgerald ve ark., 2004), hidroelektrik güç üniteleri için kullanılan baraj kapılarının otomatik kontrolü (Karaboğa ve ark., 2004), yapay sinir ağları ile birlikte bulanık mantık tabanlı ses tanıma sistemleri (Melin ve ark., 2006), sağlık alanında bulanık mantık tabanlı kanser teşhisi çalışmaları (Zadeh, 2011), depremlerin önceden bilinmesi için bulanık mantık tabanlı tahmin sistemi (Dutta ve ark., 2012), stok değerlendirmesi ve portföy inşası için bulanık kural tabanlı uzman sistemler (Avcı ve Selim, 2013), Bulanık Mantık tabanlı biyomedikal sistemler (Patel ve ark., 2013), otomobiller için bulanık mantık kullanarak "Cruise-control" uygulamaları (Cholis ve ark., 2014), çelik sanayiinde makina hızı ve ısısının bulanık mantık tabanlı kontrolü (Singhala ve ark., 2014), bulanık mantık kullanılarak üretilen fotoğraf makineleri ve kameralar (otomatik odaklama yapan makinelerden daha net bir görüntü vermekte ve sarsıntılardan doğan görüntü bozukluklarını asgariye indirmektedir) (Fayaz ve Kim, 2017) ve bulanık mantık tabanlı enerji yönetimi strateji ile elektrikli ve hibrit araçların yakıt ekonomisinde artış (Danapalasingam ve ark., 2018) uygulamaları gerçekleştirilmiştir.

FPGA tabanlı bazı bulanık mantık uygulamaları; robotik uygulamalar için gömülü bulanık denetleyicilerin FPGA tabanlı uygulaması (Sanchez-Solano ve ark., 2007), FPGA üzerinde bulanık mantık kullanılarak gerçek zamanlı DC motor hız kontrolü (Sulaiman ve ark., 2009), FPGA tabanlı gerçek zamanlı uyarlanabilir bulanık mantık denetleyicisi (Altaş, 1999), harici Analog-Dijital dönüştürücü yokken boost dönüştürücü için tek girişli Bulanık Mantık denetleyicisinin FPGA uygulaması (Taeed ve ark., 2012), FPGA ve bulanık mantık kullanarak gerçek zamanlı akıllı konum

izleme sistemi (Ferreira ve Santos, 2014), bulanık mantığa dayalı FPGA stereo eşleme uygulaması (Pérez-Patricio ve ark., 2016), çok yönlü robot sistemi için Bulanık-PI (Proportional Integral) denetleyicisinin tasarımı ve FPGA uygulaması (Masmoudi ve ark., 2017), FPGA ile bulanık mantık tabanlı objektif görüntü kalitesi değerlendirmesi (Tchendjou ve ark., 2018), gerçek zamanlı bulanık mantık uygulamaları için gauss üyelik fonksiyonunun FPGA tabanlı tasarımı ve iki eksenli güneş takibi için FPGA üzerinde bulanık mantık kontrolü (Cruz-Alejo ve ark., 2019) çalışmaları gerçekleştirilmiştir.

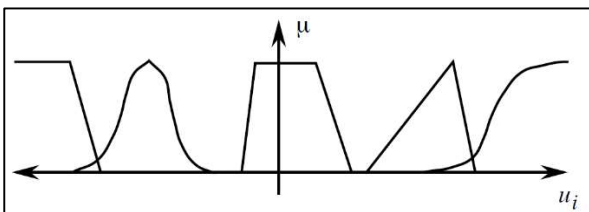
## 2. GENEL BİLGİLER

Bu çalışmada, gerçek zamanlı bulanık mantık uygulamalarında kullanılmak üzere üçgen, trapezoidal, genelleştirilmiş çan eğrisi üyelik fonksiyonlarının matematiksel modeli referans alınarak 32-bit IEEE-754-1985 kayan noktalı sayı standardına uygun FPGA-tabanlı tasarımları gerçekleştirilmiştir. Tasarım VHDL dilinde kodlanmış ve Xilinx ISE Design Suite programı kullanılarak sentezlenmiştir. Tasarlanan ünitelerin test edilebilmesi amacıyla testbench dosyaları oluşturulmuş ve performans analizleri yapılarak çip istatistikleri sunulmuştur. Bu test verilerinden elde edilen sonuçlar ile Matlab tabanlı nümerik sonuçlar karşılaştırılmış ve hata değerleri hesaplanmıştır. Place&Route işleminin ardından FPGA tabanlı üyelik fonksiyonlarına ait ünitelerin maksimum çalışma frekansları elde edilmiştir.

### 2.1. Bulanık Mantık ve Üyelik Fonksiyonları

Bilgilerin nitelik özelliklerinin kompleks, modellenmesi güç, belirsiz veya kesin olmadığı durumlarda dilsel ifadelerle desteklenerek netleştirilmesi kavramına bulanık mantık (Fuzzy Logic) denilmektedir (Klir ve Yuan, 1995). Bulanık denetleyici aşağıdaki dört öğeden oluşur (Zadeh, 1965):

*Bulanıklaştırma (Fuzzification)*, dışarıdan alınan gerçek sayısal bir değeri üyelik fonksiyonları tarafından dilsel ifadelerle dönüştürülmesi işlemidir. Uygulamalarda sıklıkla kullanılan üyelik fonksiyonları Şekil 1'de gösterildiği gibi Üçgen, Yamuk, Çan Eğrisi, Gauss ve Sigmoidal 'dir (Elmas, 2003).



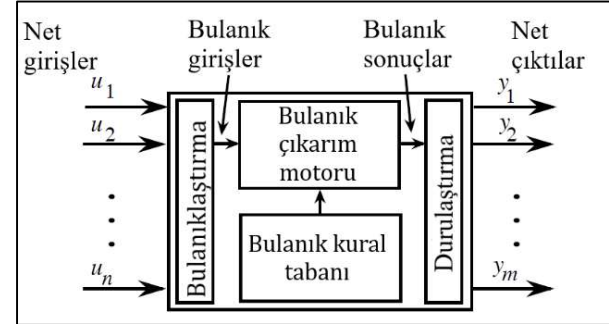
Şekil 1. Bulanık mantıkta kullanılan üyelik fonksiyon tipleri

*Durulaştırma (Defuzzification)*, bulanık çıkarım motoru ünitesinden aktarılan bulanık kümenin kesin bir değere dönüştürülmesi işlemidir. Sıklıkla kullanılan durulaştırma metotları ağırlık merkezi, en büyük üyelik ilkesi, ortalama en büyük üyelik, ağırlıklı ortalama, en büyüklerin en küçüğü ve en büyüklerin en büyüğü şeklindedir (Şen, 2012).

*Bulanık kural tabanı (Rule-base)*, uzman kişinin bilgi, beceri kontrol stratejisini karakterize eden kontrol kurallarının dilsel olarak ifade edildiği kısımdır. Veri tabanındaki girişlerin çıkış değişkenleri ile mantıksal bağ kurulduğu "Eğer-İse (If-Then)" türünde yazılabilen kurallar bütünüdür (Kaur ve Kaur, 2009).

Bulanık çıkarım motoru (Inference mechanism), kurallar üzerinde bulanık mantık yürütür ve bulanık kural tabanını kullanarak giriş ve çıkış uzayı arasında bir bağlantı kurar. Bu üniteye bilgi genellikle Mamdani ve Sugeno yöntemleri aracılığı ile modellenir (Lokman, 2011).

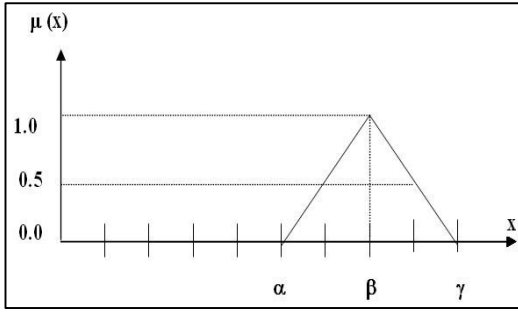
Bulanık bir kontrol sisteminin blok diyagramı Şekil 2'de gösterilmiştir.



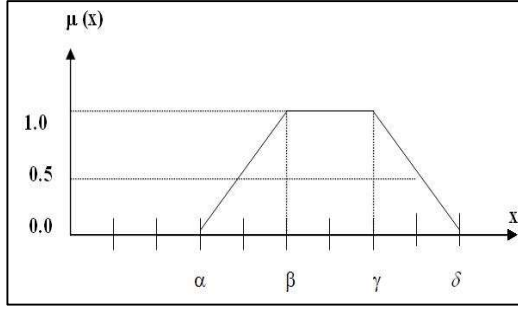
Şekil 2. Bulanık Mantık Sisteminin Şematik Yapısı

Bulanık kümeler üyelik fonksiyonları ile ifade edilir. Bu üyelik fonksiyonları her objenin bir kümedeki ağırlık derecesini vermektedir. Bu ağırlık derecesi 1 'den 0 'a kadar olabilmekte, yani tam üyelikten üye olmamaya kadar değişmektedir. Üyelik dereceleri 1' e eşit olan öğelerin toplandığı alt küme kısmına, o alt kümenin özü (Core) denir. Burada  $\mu(x)=1$ 'dir. Bunun aksine bir kümenin tüm öğelerini içeren aralığa o kümenin Dayanağı (Support) adı verilir. Dayanakta bulunan her öğenin az veya çok değerinde (0 ile 1 arasında) üyelik dereceleri vardır. Bunun matematiksel ifadesi  $0 < \mu(x) < 1$  şeklindedir. Üyelik dereceleri 1 'e veya 0 'a eşit olmayan öğelerin oluşturduğu kısımlara üyelik fonksiyonunun sınırları (Boundary) bölgeleri denir. Genel olarak, tüm üyelik fonksiyonlarında, biri sağda diğeri de solda olmak üzere iki adet geçiş bölgesi vardır. Bu çalışmada tasarım için ele alınan üyelik fonksiyonları üçgen, yamuk, genelleştirilmiş çan eğrisi ve Gauss 'tur. Bu üyelik fonksiyonlarına ait matematiksel denklemler sırası ile Denklem (1), Denklem (2), Denklem (3), Denklem (4)'te ve görseller sırası ile Şekil 3, Şekil 4, Şekil 5 ve Şekil 6'da sunulmuştur (Passino ve Yurkovich, 1998).

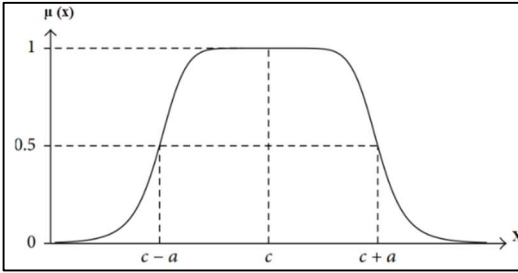




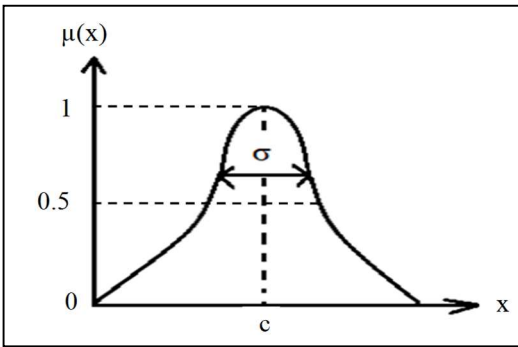
Şekil 3. Üçgen (Triangular) Üyelik Fonksiyonu



Şekil 4. Yamuk (Trapezoidal) Üyelik Fonksiyonu



Şekil 5. Genelleştirilmiş Çan Eğrisi (Generalized Bell-Shaped) Üyelik Fonksiyonu



Şekil 6. Gauss Üyelik Fonksiyonu

### Denklem (1): Üçgen (Triangular) Üyelik Fonksiyonu

Fonksiyon  $A: x \rightarrow [0,1]$   $x: X$  evreninde herhangi bir eleman

$$A(x; \alpha, \beta, \gamma) = \begin{cases} 0, x < \alpha \\ \frac{(x - \alpha)}{(\beta - \alpha)}, \alpha \leq x \leq \beta \\ \frac{(\gamma - x)}{(\gamma - \beta)}, \beta \leq x \leq \gamma \\ 0, x > \gamma \end{cases}$$

### Denklem (2): Yamuk (Trapezoidal) Üyelik Fonksiyonu

Fonksiyon  $\pi: x \rightarrow [0,1]$

$$\pi(x; \alpha, \beta, \gamma, \delta) = \begin{cases} 0, x < \alpha \\ \frac{(x - \alpha)}{(\beta - \alpha)}, \alpha \leq x \leq \beta \\ 1, \beta \leq x \leq \gamma \\ \frac{(\delta - x)}{(\delta - \gamma)}, \gamma \leq x \leq \delta \\ 0, x > \delta \end{cases}$$

### Denklem (3): Genelleştirilmiş Çan Eğrisi (Generalized Bell-Shaped) Üyelik Fonksiyonu

Fonksiyon  $\Omega: x \rightarrow [0,1]$

$b$ : fonksiyonun tavan genişliğini etkileyen parametredir.

$$\Omega(x; \alpha, b, c) = \left\{ \frac{1}{1 + \left| \frac{(x-c)}{a} \right|^{2b}} \right\}$$

### Denklem (4): Gauss Üyelik Fonksiyonu

Fonksiyon  $G: x \rightarrow [0,1]$

$$G(x; c, \sigma) = \left\{ e^{-\frac{(x-c)^2}{2\sigma^2}} \right\}$$

## 2.2. FPGA Çipleri

FPGA (Field Programmable Gate Array-Alan Programlanabilir Kapı Dizileri) çipleri ilk üretim aşamasından sonra kullanıcının istediği sistem tasarımına göre donanım yapısı kullanıcı/tasarımcı tarafından değiştirilebilen, tekrar tekrar programlanabilme özelliğine sahip dijital tüm-devrelerdir (Tuna ve ark., 2018; Tuna ve ark., 2019).

Bir FPGA, programlanabilir üç bileşenden oluşmaktadır:

*Konfigüre Edilebilir Lojik Bloklar (Configurable Logic Blocks (CLB))*: CLB kullanıcının oluşturmak istediği lojik devre için fonksiyonel elemanlar sağlar. Mantıksal fonksiyonların oluşturulabildiği Look-up Table (LUT), tek bitlik bilgilerin saklanabildiği Flip-Flop 'lar ve bilgi akışını yönlendiren Multiplexer 'lar gibi çeşitli elemanlardan oluşmaktadır.

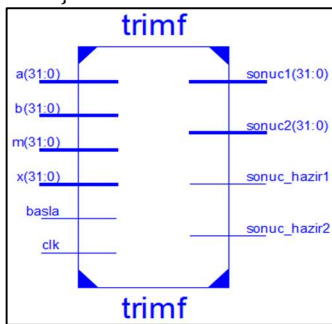
*Giriş-Çıkış Blokları (Input/Output Blocks (IOB))*: IOB 'ler FPGA çiplerinin programlanabilir giriş/çıkış terminalleridir. Bu bloktaki pinler tasarımcının isteğine göre giriş, çıkış veya hem giriş hem de çıkış olarak programlanabilir. I/O bloklarının temel görevi çipin iç sinyal hatları ile çipin pinleri arasında arabirimi sağlamaktır.

**Ara Bağlantılar (Interconnection Network):** Bu birimler hem mantıksal bloklar arasında hem de mantıksal bloklar ile IOB 'ler arasında bağlantıları konfigüre kullanırlar (Alçın ve ark., 2016; Alçın ve ark., 2019).

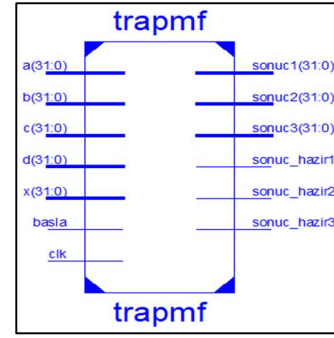
FPGA çiplerinde sıklıkla VHDL ve Verilog programlama dilleri ile tasarımlar gerçekleştirilmektedir. Bu çipler paralel çalışma, düşük güç tüketimi, hızlı ilk üretim ve yüksek kapasite gibi özellikleriyle diğer sayısal platformlara göre ön plana çıkmaktadır (Karataş ve ark., 2020). Günümüzde yapay sinir ağları, osilatör tasarımı, uzay, havacılık ve savunma sanayi, ses, otomotiv, video ve görüntü işleme, kaos teorisi ve uygulamaları, ASIC gibi amaca özel tasarlanmış entegre prototiplerinde ve daha birçok alanda kullanılmaktadır (Koyuncu ve ark., 2018; Yılmaz ve ark., 2019; Rajagopal ve ark., 2019; Güngör ve ark., 2019; Akgül ve ark., 2019).

### 3. FPGA-TABANLI ÜYELİK FONKSİYONLARI

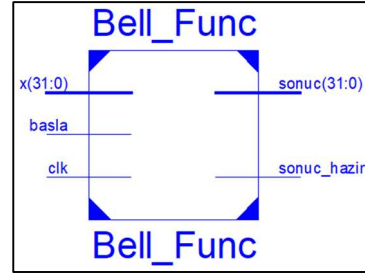
Bu çalışmada, gerçek zamanlı bulanık mantık uygulamaları için üçgen, yamuk, genelleştirilmiş çan eğrisi üyelik fonksiyonlarının matematiksel modeli referans alınarak 32-bit IEEE-754-1985 kayan noktalı sayı (Floating Point Number) standardına uygun FPGA-tabanlı üçgen, yamuk, çan eğrisi ve Gauss üyelik fonksiyonu üniteleri tasarlanmıştır. Tasarım VHDL dilinde kodlanmış ve Xilinx ISE Design Suite 14.7 programı kullanılarak sentezlenmiştir. Tasarlanan ünitelerin en üst seviye RTL şemaları ve ikinci seviye RTL şemaları çıkartılmıştır. FPGA-tabanlı üçgen, yamuk, çan eğrisi ve Gauss üyelik fonksiyonlarına ait birinci seviye RTL şemaları sırası ile Şekil 7, Şekil 8, Şekil 9 ve Şekil 10'da gösterilmiştir. İkinci seviye RTL şemaları sırası ile Şekil 11, Şekil 12, Şekil 13 ve Şekil 14'te gösterilmiştir.



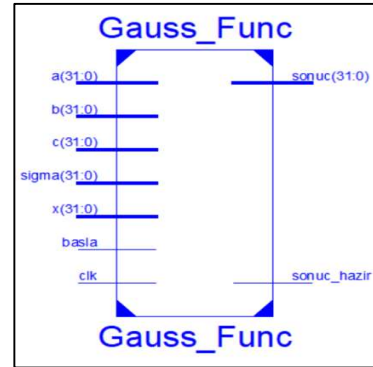
**Şekil 7.** Üçgen Üyelik Fonksiyonunun Birinci Seviye RTL Şeması



**Şekil 8.** Yamuk Üyelik Fonksiyonunun Birinci Seviye RTL Şeması

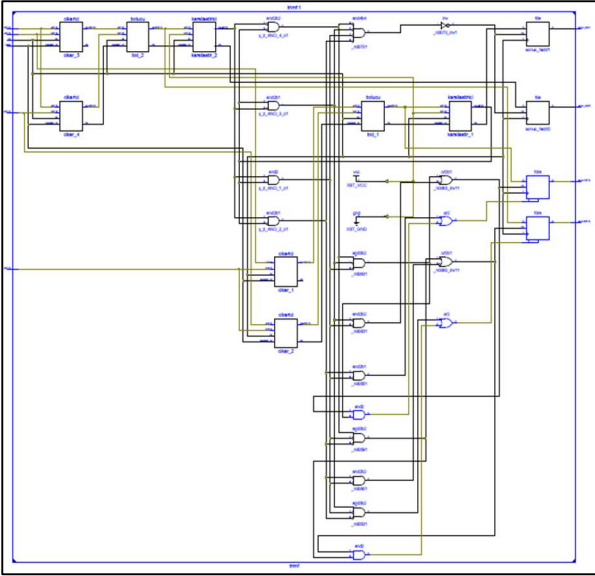


**Şekil 9.** Çan Eğrisi Üyelik Fonksiyonunun Birinci Seviye RTL Şeması

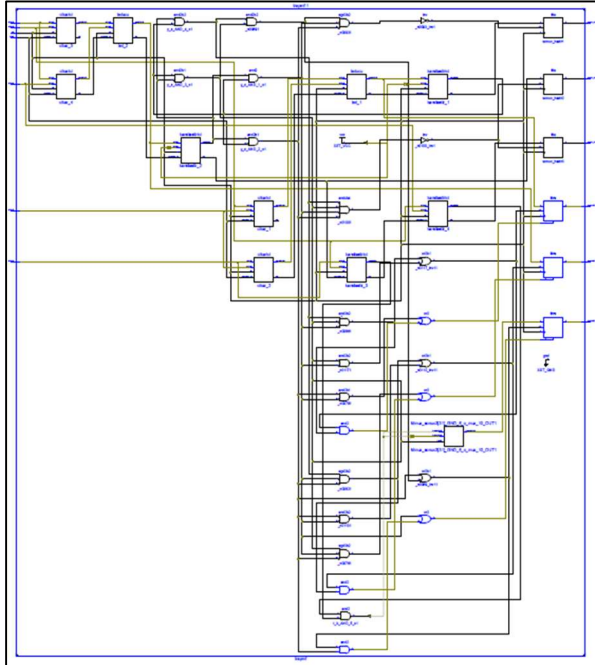


**Şekil 10.** Gauss Üyelik Fonksiyonunun Birinci Seviye RTL Şeması

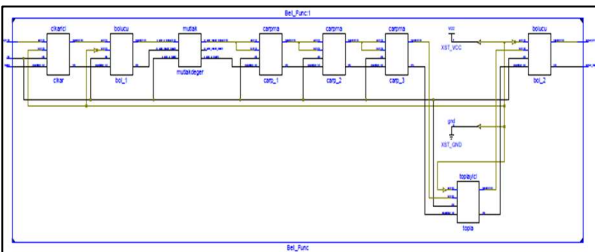
RTL şemalarında görüldüğü üzere üniteler 32-bit giriş sinyallerine ve 32-bit çıkış sinyallerine sahiptir. Ünitelerin çalışmaya başlayabilmesi için *basla* giriş sinyal değeri '1' olmalıdır. 1-bit *clk* sinyali ünite içerisindeki alt birimlerin senkron bir şekilde çalışabilmesi için kullanılmaktadır. Üniteler ilk sonucunu üretmeye başlamadan önce *sonuc\_hazir* sinyalleri '0' olmaktadır. Üniteler sonucu ürettiği durumlarda *sonuc\_hazir* sinyal değerleri '1' olmaktadır.



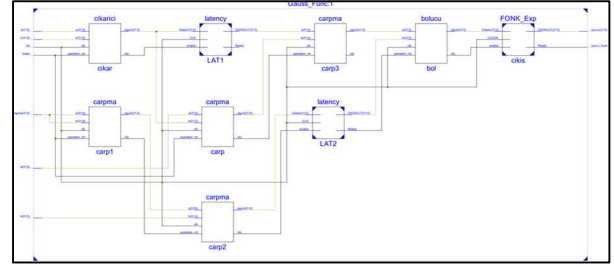
Şekil 11. Üçgen Üyelik Fonksiyonunun İkinci Seviye RTL Şeması



Şekil 12. Yamuk Üyelik Fonksiyonunun İkinci Seviye RTL Şeması



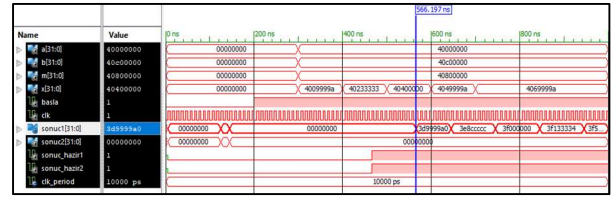
Şekil 13. Çan Eğrisi Üyelik Fonksiyonunun İkinci Seviye RTL Şeması



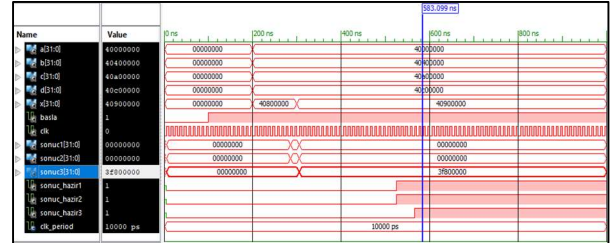
Şekil 14. Gauss Üyelik Fonksiyonunun İkinci Seviye RTL Şeması

#### 4. TEST SONUÇLARI VE ÇİP İSTATİSTİKLERİ

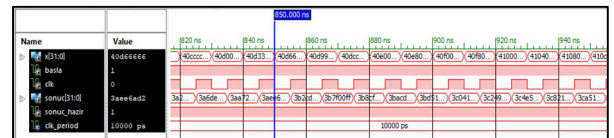
Yapılan FPGA-tabanlı üyelik fonksiyonlarının test edilebilmesi amacı ile toplamda 105 adet test verisi oluşturulmuştur. Bu test verileri öncelikle nümerik olarak işlenmiştir. Ardından Xilinx ISE Design Suite 14.7 programı kullanılarak testbench dosyaları oluşturulmuştur. Her bir üyelik fonksiyonu için Xilinx ISE Design Suite programından test sonuçları elde edilmiştir. Üçgen, Yamuk, Çan Eğrisi ve Gauss üyelik fonksiyonlarının test sonuçları sırası ile Şekil 15, Şekil 16, Şekil 17 ve Şekil 18'de verilmiştir.



Şekil 15. Üçgen Üyelik Fonksiyonunun Test Sonuçları



Şekil 16. Yamuk Üyelik Fonksiyonunun Test Sonuçları



Şekil 17. Çan Eğrisi Üyelik Fonksiyonunun Test Sonuçları



Şekil 18. Gauss Üyelik Fonksiyonunun Test Sonuçları

Xilinx ISE Design Suite 14.7 ortamında tüm tasarımların test işleminden beklenen sonuçlar alındıktan sonra tasarımlar Xilinx XC6VLX75T-

3FF784 FPGA çipi için sentezlenmiştir. Tablo 1'de tüm aktivasyon fonksiyonlarına ait FPGA çip istatistikleri gösterilmektedir.

**Tablo 1.** Üyelik Fonksiyonları ve Çip İstatistikleri

FPGA Çipi Kaynak Kullanımı	Üçgen Üyelik Fonksiyon	Yamuk Üyelik Fonksiyon	Çan Eğrisi Üyelik Fonksiyon	Gauss Üyelik Fonksiyon
IOBs Sayısı / Kullanım Oranı	196 / %54	261 / %72	67 / %18	194 / %53
BUFGCTLs Sayısı / Kullanım Oranı	1 / %3	1 / %3	1 / %3	1 / %3
Slice LUTs Sayısı / Kullanım Oranı	3534 / %7	3627 / %7	4399 / %9	6778 / %14
Slice Registers Sayısı / Kullanım Oranı	3220 / %3	4916 / %5	3544 / %3	5911 / %6
Maksimum Çalışma Frekans (MHz)	<b>245.031</b>	<b>599.664</b>	<b>185.086</b>	<b>304.534</b>

## 5. SONUÇLAR

Sunulan bu çalışmada, gerçek zamanlı bulanık mantık uygulamaları için gauss, üçgen, yamuk ve çan eğrisi üyelik fonksiyonlarının matematiksel modeli referans alınarak 32-bit IEEE-754-1985 kayan noktalı sayı standardına uygun FPGA-tabanlı gauss, üçgen, yamuk ve çan eğrisi üyelik fonksiyonu üniteleri tasarlanmıştır. Tasarım VHDL ile kodlanmış ve Xilinx ISE Design Suite programı kullanılarak sentezlenmiştir. Tasarlanan ünitelerin test edilebilmesi amacıyla toplam 105 adet test verisi ile 8 ayrı testbench dosyası oluşturulmuştur. Bu test verilerinden elde edilen sonuçlar ile Matlab tabanlı nümerik sonuçlar karşılaştırılmış ve hata değerleri maksimum  $6.4 \times 10^{-5}$  olarak elde edilmiştir. Bu test verileri kullanılarak nümerik tabanlı üyelik fonksiyonu ünitelerinin veri işleme hızı ile FPGA-tabanlı üyelik fonksiyonu ünitelerinin veri işleme hızı karşılaştırılmıştır. Place&Route işleminin ardından FPGA tabanlı Gauss üyelik fonksiyon ünitesinin maksimum çalışma frekansı 304.534 MHz, Üçgen üyelik fonksiyon ünitesinin maksimum çalışma frekansı 245.031 MHz, Yamuk üyelik fonksiyon ünitesinin maksimum çalışma frekansı 599.664 MHz ve Çan eğrisi üyelik fonksiyon ünitesinin maksimum çalışma frekansı 185.086 MHz olarak elde edilmiştir. Bu çalışmada Bulanık Mantık Kontrolcüsünde bulunan Bulanıklaştırıcı ünitesindeki üçgen, yamuk ve çan eğrisi üyelik fonksiyonları FPGA çiplerinde uygulanmak üzere modellenmiştir. İleriki çalışmalarda FPGA üzerinde gerçek zamanlı bulanık mantık uygulamaları gerçekleştirebilmek amacıyla bulanık çıkarım motoru ve durulaştırma üniteleri gerçekleştirilebilir.

## KAYNAKÇA

Abu-Khudhair, A., Muresan, R., Yang, S.X. (2010). FPGA based real-time adaptive fuzzy controller. IEEE International Conference on Automation and Logistics, Hong Kong and Macau, pp. 539-544.

- Akgül, A., Arslan, C., Arıcıoğlu, B. (2019). Design of an Interface for Random Number Generators based on Integer and Fractional Order Chaotic Systems. *Chaos Theory and Applications*, 1(1), 1-18.
- Altaş, İ.H. (1999). Bulanık Mantık: Bulanık Denetim. Enerji, Elektrik, Elektromekanik-3e, Bilesim Yayıncılık A.Ş., İstanbul, Sayı 64, Sayfalar:76-81
- Alçın, M., Pehlivan, İ., Koyuncu, İ. (2016). Hardware design and implementation of a novel ANN-based chaotic generator in FPGA. *Optik-International Journal for Light and Electron Optics*, 127(13), 5500-5505.
- Alçın, M., Koyuncu, İ., Tuna, M., Varan, M., & Pehlivan, İ. (2019). A novel high speed Artificial Neural Network-based chaotic True Random Number Generator on Field Programmable Gate Array. *International Journal of Circuit Theory and Applications*, 47(3), 365-378.
- Avcı, M.G., Selim, H. (2013). A fuzzy rule based expert system for stock evaluation and portfolio construction: An application to Istanbul Stock Exchange. *Expert Systems with Applications* 40(3):908-920.
- Cholis, N., Kuspriyanto, K., Saefudin, D., Nugraha, I.K. (2014). Developing Adaptive Cruise Control Based on Fuzzy Logic Using Hardware Simulation. *International Journal of Electrical and Computer Engineering* 4(6)
- Cruz-Alejo, J., Antonio-Méndez, R., Salazar-Pereyra, M. (2019). Fuzzy logic control on FPGA for two axes solar tracking. *Neural Comput & Applic* 31, 2469-2483.
- Danapalasingam, K.A., Sabri, M.F.M, Rahmat, M.F. (2018). Improved Fuel Economy of Through-the-Road Hybrid Electric Vehicle with Fuzzy Logic-Based Energy Management Strategy. *International Journal of Fuzzy Systems* 20(8)
- Dutta, P., Mishra, O.P., Naskar, M.K. (2012). Decision analysis for earthquake prediction methodologies: fuzzy inference algorithm for trust validation. *International Journal of Computer Applications* 45:13-20.

- Elmas, Ç. (2003). Bulanık mantık denetleyiciler. Seçkin Yayıncılık, Ankara.
- Fayaz, M., Kim, D. (2017). An Effective Control Method of IP Camera Based on Fuzzy Logic and Statistical Moments. *International Journal of Control and Automation* 10(4):97-108.
- Ferreira, J.A.F., Soares dos Santos, M.P. (2014). Novel intelligent real-time position tracking system using FPGA and fuzzy Logic. *ISA Transactions*, Volume 53, Issue 2, Pages 402-414
- Fitzgerald, J.A., Geiselbrechtinger, F., Kechadi, T. (2004). Application of fuzzy logic to online recognition of handwritten symbols. Conference: *Frontiers in Handwriting Recognition*, 2004. IWFHR-9 2004. Ninth International Workshop on.
- Güngör, E., Çavuş, E., Pehlivan, İ. (2019). A Logistic Map Runge Kutta-4 Solution for FPGA Using Fixed Point Representation. *Chaos Theory and Applications*, 1(1), 19-28.
- Karaboğa, D., Bağış, A., Haktanır, T. (2004). A new method for reservoir control of dams. *Hydrological Proces*, Vol. 18/13, pp 2485-2501.
- Karataş, F., Koyuncu, İ., Alçın, M., Tuna, M. (2020). FPGA based Implementation of membership functions for real time fuzzy logic applications, *International Asian congress on Contemporary Sciences-3*, Konya, Turkey, pp. 62-69.
- Kaur, D.A., Kaur, K. (2009). Fuzzy expert systems based on membership functions and fuzzy rules. *IEEE International Conference on Artificial Intelligence and Computational Intelligence*, AICI'09. Vol. 3, pp. 513-517.
- Khokhar, S., Peng, Q., Asif, A., Noor, M.Y., Inam, A. (2020). A Simple Tuning Algorithm of Augmented Fuzzy Membership Functions. *IEEE Access*, vol. 8, pp. 35805-35814.
- Klir, G.J., Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic. Theory and Applications*, Prentice Hall.
- Koyuncu, İ. (2018). Implementation of High Speed Tangent Sigmoid Transfer Function Approximations for Artificial Neural Network Applications on FPGA. *Advances in Electrical and Computer Engineering*, vol.18, no.3, pp.79-86.
- Koyuncu, İ., Tuna, M., Alçın, M. (2018). FPGA tabanlı farklı nümerik algoritmalar ile kaotik osilatör tasarımları. *International Eurasian Conference on Science, Engineering and Technology*. November 22-23, 2018 Ankara, Turkey.
- Lokman, G. (2011). Parçacık Sürü Optimizasyon Algoritması ile Bulanık Kontrolör Tasarımı. Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, pp.36-37.
- Mamdani, E.H., Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1), 1-13.
- Masmoudi, M.S., Krichen, N., Koesdwiady, A.B., Karray, F., Masmoudi, M. (2017). Design and FPGA Implementation of a Fuzzy-PI Controller for Omnidirectional Robot System. In: Kim JH, Karray F, Jo J, Sincak P, Myung H (eds). *Robot Intelligence Technology and Applications 4. Advances in Intelligent Systems and Computing*, vol 447. Springer, Cham
- Melin, P., Urías, J., Castillo, O. (2006). Voice Recognition with Neural Networks, Type2 Fuzzy Logic and Genetic Algorithms. *Engineering Letters* 13(2):108-116
- Mukaidono, M., Shen, Z., Ding, L. (1987). *Fuzzy Prolog*. International Fuzzy Systems Association Congress, Tokyo, Japan.
- Passino, K.M., Yurkovich, S. (1998). *Fuzzy Control*. Addison-Wesley; 1st edition, ISBN-10: 020118074X, Chapter 1, pp.1-110.
- Patel, A., Gupta, S.K., Rehman, Q., Verma, M.K. (2013). Application of Fuzzy Logic in Biomedical Informatics. *Journal of Emerging Trends in Computing & Information Sciences*. Vol 4, No 1.
- Pérez-Patricio, M., Aguilar-González, A., Arias-Estrada, M., Hernández-de León, H., Camas-Anzueto, J. (2016). An FPGA stereo matching unit based on fuzzy logic. *Microprocessors and Microsystems: Embedded Hardware Design (MICPRO)*, 42, pp.87-99.
- Rajagopal, K., Tuna, M., Karthikeyan, A., Koyuncu, İ., Duraisamy, P., & Akgul, A. (2019). Dynamical analysis, sliding mode synchronization of a fractional-order memristor Hopfield neural network with parameter uncertainties and its non-fractional-order FPGA implementation. *The European Physical Journal Special Topics*, 228(10), 2065-2080.
- Sanchez-Solano, S., Cabrera, A.J., Baturone, I., Moreno-Velo, F.J., Brox, M. (2007). FPGA Implementation of Embedded Fuzzy Controllers for Robotic Applications. *IEEE Transactions on Industrial Electronics*, vol. 54/4, pp. 1937-1945.
- Singhala, P., Shah, D.N., Patel, B. (2014). Temperature Control using Fuzzy Logic. *International Journal of Instrumentation and Control Systems (IJICS)* Vol.4, No.1
- Sugeno, M. (1985). An introductory survey of fuzzy control. *Information sciences*, 36(1-2), 59-83.
- Sulaiman, N., Obaid, Z.A., Marhaban, M.H., Hamido, M.N. (2009). FPGA-Based Fuzzy Logic: Design and Applications- a Review. *IACSIT International Journal of Engineering and Technology* Vol.1, No.5, ISSN: 1793-8236
- Şen, Z. (2012). Fuzzy Philosophy of Science. *Journal of Higher Education and Science*, 2, 20-24.
- Taeed, F., Salam, Z., Ayob, S. (2012). FPGA Implementation of a Single-Input Fuzzy Logic Controller for Boost Converter With the Absence of an External Analog-to-Digital Converter. *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1208-1217.
- Tchendjou, G.T., Simeu, E., Alhakim, R. (2018). Fuzzy logic based objective image quality assessment

- with FPGA implementation. *Journal of Systems Architecture*, Volume 82, Pages 24-36.
- Tuna, M., Koyuncu, İ., Alçın, M. (2018). Fixed and Floating point-Based High-Speed Chaotic Oscillator Design with Different Numerical Algorithms on FPGA. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*. Vol 7, Issue 7.
- Tuna, M., Alçın, M., Koyuncu, İ., Fidan, C. B., & Pehlivan, İ. (2019). High speed FPGA-based chaotic oscillator design. *Microprocessors and Microsystems*, 66, 72-80.
- Yasunobu, S., Miyamoto, S., Ihara, H. (2002). A Fuzzy Control for Train Automatic Stop Control. *Trans. of the Society of Instrument and Control Engineers*, Vol.E-2, No.1, 1/9.
- Yılmaz, C., Koyuncu, İ., Alçın, M., & Tuna, M. (2019). Artificial Neural Networks based thermodynamic and economic analysis of a hydrogen production system assisted by geothermal energy on Field Programmable Gate Array. *International Journal of Hydrogen Energy*, 44(33), 17443-17459.
- Zadeh, H.G. (2011). Diagnosing Breast Cancer with the Aid of Fuzzy Logic Based on Data Mining of a Genetic Algorithm in Infrared Images. *Middle East Journal of Cancer* 3(4):119-129.
- Zadeh, L.A. (1965). Fuzzy Algorithms. *Information and Control*, 8, 338-353. Australia, pp.1-6.
- Vapnik, V., and Chapelle, O. (2000). "Bounds on error expectation for support vector machines." *Neural Computation*, Vol. 12, No. 9, pp. 2013-2036.
- Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R. and Mittal, A. (2019). "Pneumonia detection using cnn based feature extraction." *Proc., IEEE Third International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, Tamil Nadu, India, pp. 1-7. IEEE.
- Zhang, J., Xie, Y., Wu, Q. and Xia, Y. (2019). "Medical image classification using synergic deep learning." *Medical Image Analysis*, Vol. 54, pp. 10-19



Araştırma Makalesi

## Nesne Sınıflandırma İşlemi İçin Tensor İşleme Birimi ve CPU Performans Karşılaştırması

Ayşe Ayyüce Demirbaş \*1, Ahmet Çınar<sup>1</sup>

<sup>1</sup>Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Elazığ, Türkiye

### Anahtar Kelimeler:

TPU  
CPU  
Derin Öğrenme  
Tensorflow  
MobileNet v1

### ÖZ

Tensor İşleme Birimi (TPU), Google tarafından derin öğrenme görevlerini hızlandırmak için özel olarak geliştirilmiş bir yongadır. Yakın zamana kadar TPU'lar sadece Google Cloud ve Google Colab Platformları üzerinden kullanılabilirdi. 2019 yılının başlarında Google firması bu ürünü Coral adı altında donanımsal olarak üretmiştir. Bu sayede, dizüstü bilgisayar ve Raspberry Pi 3 gibi düşük donanım özelliklerine sahip cihazlarda derin öğrenme uygulamaları daha hızlı bir şekilde gerçekleştirilebilmektedir. Bu makalede MobileNet v1 modeli kullanılarak ilk olarak TPU bulduran Google Coral USB Accelerator ile daha sonra ise CPU kullanılarak 5 kategori ve 4326 çiçek görüntülerinden oluşan bir Kaggle veri seti eğitilmiş ve eğitilen bu veri setine dayanarak görüntülerin sınıflandırılması sağlanmıştır. Bu yolla cihazın performansı incelenmiştir. CPU için başarı oranı %84.29252, TPU için başarı oranı ise %99.609 olarak elde edilmiştir.

## Comparing Tensor Processing Unit and CPU Performance for Image Classification

### Keywords:

TPU  
CPU  
Deep Learning  
Tensorflow  
MobileNet v1

### ABSTRACT

Tensor Processing Unit (TPU) is a processor to accelerate deep learning tasks developed by Google. Until recently, TPUs have been only used on Google Cloud and Google Colab platforms. In early 2019 Google produces this hardware as Coral. Thus, deep learning tasks can be performed more quickly on low-cost computers such as laptops and Raspberry Pi 3. In this paper, a Kaggle dataset that consists of 5 categories and 4326 flower images have been trained with the Google Coral USB Accelerator that contains TPU and a CPU using the MobileNet v1 model and the images are classified based on this dataset. In this way, we have investigated the performance of Google Coral USB Accelerator. The accuracy of the training with CPU is 84.29252%, for the TPU training, 99.609%.

### \*Sorumlu Yazar

( a.ayyuiced@gmail.com) ORCID ID 0000-0002-6731-9345  
( acinar@firat.edu.tr) ORCID ID 0000-0001-5528-2226

e-ISSN: 2717-8579

Geliş Tarihi: 01/04/2020; Kabul Tarihi: 29/04/2020

Bilgisayar Bilimleri ve Teknolojileri Dergisi

## 1. GİRİŞ

Makine öğrenmesi ile ilgili ilk çalışmalar 1943 yılında Walter Pitts ve Warren McCulloch tarafından yapılmıştır. Bu çalışma ile bir sinir ağı matematiksel olarak modellenmiştir. (McCulloch ve Pitts, 1943). 1950 yılında ise Alan Mathison Turing, bir bilgisayarın zeki olup olmadığını belirlenmesi için Turing Testi'ni ortaya koymuştur. Bu fikir yapay zekânın ilk kilometre taşı olarak tarihe geçmiştir (Turing, 1950). Modern yapay sinir ağlarının temelini atılması, 1958 yılında Frank Rosenblatt tarafından perceptron algoritmasının yazılmasıyla mümkün olmuştur (Rosenblat, 1958). David E. Rumelhart, Geoffrey E. Hinton ve Ronald J. Williams 1986'da yayımladıkları bir bildiri ile birçok gizli katmana sahip olan bir sinir ağının daha basit bir yolla daha etkili bir şekilde nasıl eğitilebileceğini göstermişlerdir (Rumelhart ve ark., 1988).

TensorFlow; günümüzde çok yaygın olarak kullanılan, Google tarafından geliştirilen açık kaynak kodlu bir makine öğrenmesi sistemidir. (Abadi ve ark., 2016) Çalışma biçimi basitçe açıklanacak olursa, Tensor olarak adlandırılan skaler, vektör veya n-boyutlu diziyi girdi olarak alır ve bir çıktı üretir. "Tensorflow'da tüm işlemler Graph adı verilen art arda gerçekleşen bir hesaplama kümesinde gerçekleştirilir." (URL-1) TPU'lar (Tensor processing unit- Tensör işleme üniteleri) sadece Tensorflow ile kullanılabilirler için bu çalışmada Tensorflow kullanılmıştır.

Derin öğrenme, bir çeşit makine öğrenmesidir. Görüntü tanıma, görüntü işleme, doğal dil işleme, tıbbi görüntü analizi gibi birçok alanda yaygın olarak kullanılır. İnsan beynindeki nöronların çalışma yapısına benzeyen yapay sinir ağları ile modellenen algoritmalarla çalışır. Bu yapay sinir ağları aynı nöronlar gibi, dış dünyadan girdi alır, bu girdiyi işler ve bir çıktı üretirler.

Derin öğrenme için CPU (Central Processing Unit-Merkezi işlem ünitesi), GPU (Graphics Processing Unit- Grafik işlem ünitesi ) ve TPU kullanılabilir. CPU bu iş için özelleştirilmiş bir donanım değildir. GPU ise CPU'ya kıyasla daha verimli çalışacaktır. Fakat Tensorflow ile yapılan çalışmalarda TPU kullanılması büyük bir verim sağlayacaktır. Bunun sebebi TPU'nun Tensorflow için özel olarak geliştirilmiş bir işlemci olmasıdır.

Google Colab platformu; CPU, GPU veya TPU kullanılarak derin öğrenme çalışmalarının yapılmasını mümkün kılmaktadır. Bunun yanında Google Cloud üzerinden de TPU kullanılabilir. Fakat bu kullanımlar sırasında veri seti ve yazılan kodlar sadece çalışma yapılan bilgisayarda kalmayacak, karşı taraftaki bilgisayarlarda da veri olarak depolanacaktır. Bu sebepten dolayı çalışmanın yapıldığı bilgisayarda TPU kullanılabilmesi sorunu çözecektir. Google Coral USB Accelerator bu amaç için geliştirilmiş bir modüldür. İçinde TPU bulunduran bu cihaz, USB portundan bilgisayara bağlanabilen bir yardımcı işlemcidir.

Bu yardımcı işlemci, kişisel bilgisayarlarda veya Raspberry Pi gibi tek kart bilgisayarlarda elbette ki Google'ın bilgisayarlarındaki kadar verim elde edemeyecektir. Fakat Tensorflow için CPU'dan daha hızlı çalışacağı gerçektir.

Bu çalışmanın amacı, kişisel bilgisayarlarda yürütülen derin öğrenme çalışmalarını süre açısından incelemektir. Detaylı bir literatür taraması sonucunda bu konuda yeterli sayıda çalışma olmadığı görülmüştür.

Bu çalışmanın yapısı aşağıdaki gibidir;

Giriş bölümünde bu alanda daha önce yapılan çalışmalar ele alınmıştır. Yöntem bölümünde TPU'nun nasıl çalıştığı, çalışmada kullanılan model ve veri seti incelenmiştir. Daha sonra veri seti, bu modelle eğitilmiş ve sınıflandırmalar yapılmıştır. Bulgular bölümünde bir önceki bölümde elde edilen deneysel sonuçlar kullanılarak eğitim süresine göre işlemlerin performansları karşılaştırılmıştır. Sonuç bölümünde ise bu konuda daha önce yapılan çalışmalara değinilmiş ve karşılaştırmalı sonuçlar verilmiştir. Yapılan çalışmada elde edilen verilere göre TPU'nun avantajlarından söz edilmiştir.

## 2. YÖNTEM

Derin öğrenme çalışmalarında işlemci olarak genellikle Graphical Processing Unit (GPU) kullanılır. Bu durum verimi artıracaktır. Bu çalışmada Google Coral USB Accelerator ve Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'nun verisetinin eğitilmesi sırasında gösterdikleri performans ele alınacaktır. Derin öğrenme ile çalışırken Tensorflow kütüphanesi kullanılacaksa işlemci olarak TPU kullanmanın avantajlı olacağı öngörülmektedir.

### 2.1. TPU Nedir?

Bilgisayarlarda genel amaçlı işlemler için kullanılan CPU'lar skaler verilerle çalışır. "CPU'nun en büyük yararı esnekliğidir. Von Neumann mimarisi ile milyonlarca farklı uygulama için her türlü yazılım yüklenebilir. Bir bilgisayarda kelime işleme, roket motorlarını kontrol etmek, banka işlemlerini gerçekleştirmek veya görüntüleri sinir ağları ile sınıflandırmak için CPU kullanılabilir." (URL-2) Fakat Von Neumann mimarisinin darboğazı CPU'nun bir dezavantajıdır.

Bunun yanında GPU'lar, CPU'dan farklı olarak çok fazla sayıda Arithmetic Logic Unit (ALU) bulundurlar. Vektörler üzerinde işlem yaptıkları için grafiksel görevleri daha hızlı bir şekilde çalıştırabilirler. CPU'dan daha verimli çalıştıkları için derin öğrenme uygulamalarında sıkça tercih edilirler.

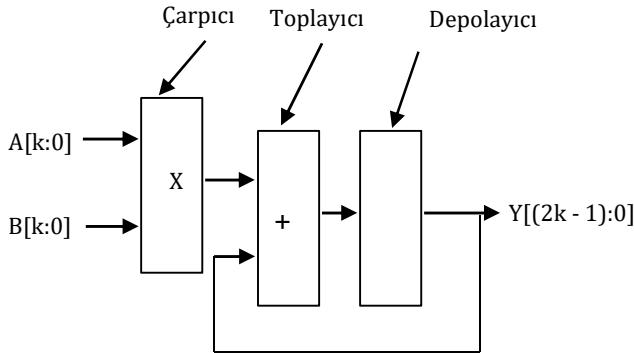
Yapay sinir ağları matrisler ile oluşturuldukları için bu iş için matrisler üzerinde işlem yapan TPU'ları kullanmak çok uygundur. TPU'lar Matris Çarpım Birimi ve Vektör İşleme Birimi



bulundururlar. Yapay sinir ağları için gereken matris işlemlerini Şekil 1’de gösterildiği gibi (çarpma ve toplama) hızlı bir şekilde, daha az güç tüketimi ile gerçekleştirirler.

Basit bir ifadeyle TPU, 8 bitlik donanım çarpanlarının 256x256 matrisini destekleyen donanımsal bir matris çapıcısıdır. (Olsen, 2017) Verinin donanım matrisine akmasını sağlayan sistolik kaydırma devresi sayesinde saniyede 65536 çarpma işlemi yapabilir. (Jouppi ve ark., 2017) TPU’lar binlerce toplama ve çarpma devresi içerirler. Bu devreler birbirine bağlanarak fiziksel bir matris oluşturulmuştur. Bu, sistolik dizi mimarisidir. (Jouppi ve ark., 2017) Bir sistolik dizi, yapay sinir ağı hesaplamalarını şu şekilde yürütür:

“İlk başta, TPU parametreleri bellekten çarpma ve toplama devrelerinden oluşan matrise yükler. Ardından, TPU verileri bellekten yükler. Her çarpma işleminde, toplama işlemi yapılırken sonuç bir sonraki çarpım devresine geçer. Böylece çıktı, veriler ve parametreler arasındaki tüm çarpma sonucunun toplamı olacaktır. Tüm büyük hesaplamalar ve verilerin aktarılması sürecinde, hiçbir hafıza erişimine ihtiyaç duyulmaz.” (URL-2)

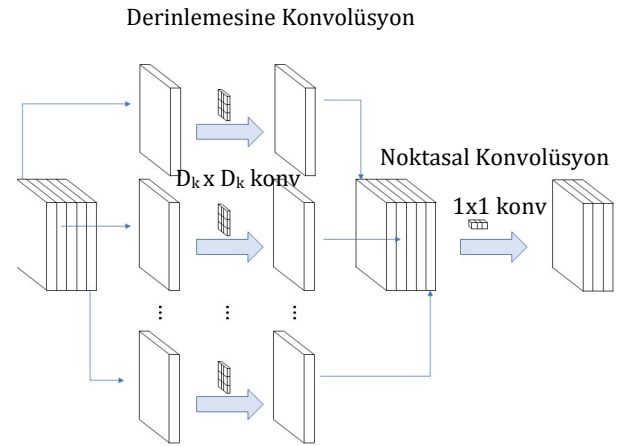


Şekil 1. Matris Çarpım Birimi (URL-3)

## 2.2. Model

ImageNet, 14 milyondan fazla görüntüden oluşan bir veri kümesidir. (Deng ve ark., 2009) Kullanılan model “ImageNet veri kümesinden 1.000 sınıfı anlamak için önceden eğitilmiş, değiştirilmiş bir MobileNet v1 modelidir.” (Retrain a classification model on-device with weight imprinting)

MobileNet v1 konvolüsyonel sinir ağları yapısını kullanır. (Sheng ve ark., 2018) “Konvolüsyon katmanının işi iki alt göreve ayrılır; girdiyi filtreleyen derinlemesine konvolüsyon katmanı ve yeni özellikler oluşturmak için bu filtrelenmiş değerleri birleştiren 1x1 konvolüsyon katmanı (noktasal konvolüsyon katmanı olarak da adlandırılır).” (URL-4) Bu iki konvolüsyon katmanı, derinlemesine ayrılabilir konvolüsyon bloğu oluştururlar. Şekil 2’de gösterilen derinlemesine ayrılabilir konvolüsyon, model boyutunu ve karmaşıklığını azaltmak için kullanılır.



Şekil 2. Derinlemesine Ayrılabilir Konvolüsyon Bloğu (URL-5)

MobileNet modeli aktivasyon fonksiyonu olarak ReLu6 kullanır. ReLu6, ReLu fonksiyonuna benzer. Ancak ReLu6 fonksiyonunda aktivasyon değeri 6’dan büyük olamamaktadır. Bu fonksiyon matematiksel olarak Denklem (1)’deki gibi tanımlanır,

### Denklem (1): ReLu Fonksiyonu

$$y = \min(\max(0, x), 6)$$

## 2.3 Veri setinin Eğitilmesi

Farklı sayıda kategori ve görüntülerden oluşan veri setleri kolaylıkla bulunabilir. Çalışma için uygun veri seti seçildikten sonra bu veri seti bir model kullanılarak eğitilir. Bu noktada kullanılan işlemci hız açısından büyük öneme sahiptir.

### 2.3.1 Veri seti

Bu çalışmada veri seti olarak Kaggle’den alınan Papatya, Karahindiba, Gül, Ayçiçeği ve Lale olmak üzere 5 kategoride toplam 4326 çiçek görüntüsü kullanılmıştır. (Kaggle Flowers Recognition) Görüntüler yüksek çözünürlüklü olmayıp, yaklaşık 320x240 boyutlarındadır. Her bir kategoride ortalama 864 görüntü bulunmaktadır.

### 2.3.2 İşlemci

Veri seti eğitilirken önce MobileNet v1 modeli ile içinde TPU bulduran Google Coral USB Accelerator kullanılmıştır. Daha sonra ise Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU ile aynı veri seti aynı model ile tekrar eğitilmiştir.

Veri setinde bulunan görüntüler ve önceden eğitilmiş model kullanılarak Google tarafından yazılan classification\_transfer\_learning.py (URL-6) ile eğitilmiştir. Bu işlem 23140 milisaniye sürmüştür.

Daha sonra ise aynı veri seti, aynı model ile Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU kullanılarak eğitilmiştir. Bu işlem için Google tarafından yazılmış retrain.py kodları kullanılmıştır (URL-7). Bu işlem 373800 milisaniyede tamamlanmıştır.

### 2.3.3 Görüntülerin Sınıflandırılması

Eğitilen model kullanılarak Şekil 3'te gösterilen papatya, karahindiba, gül, ayçiçeği ve lale olmak üzere 5 kategoride veri setinde bulunmayan 30 tane çiçek görüntüsü Google Coral USB Accelerator ile sınıflandırılmıştır. Bu işlem sırasında Google tarafından yazılmış classify\_image.py (URL-8) kodları kullanılmıştır. Bu 30 tane testin 23'ünde, eğitilen modelin doğru sonuç verdiği gözlemlenmiştir.



Şekil 3. Test aşamasında kullanılan görüntüler

Google Coral TPU USB Accelerator kullanılarak papatya.jpg dosyası için yapılan sınıflandırmada sonuç aşağıdaki gibi elde edilmiştir,

daisy: 0.99609

Aynı görüntü Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU ile sınıflandırıldığında aşağıdaki skor elde edilmiştir,

daisy  
Score : 0.8429252

Her iki işlemci için de elde edilen skorlar birbirine yakın ve çoğu zaman doğru olacaktır. Ancak sonuçlar bölümünde de görüleceği üzere eğitim sırasında Google Coral TPU USB Accelerator daha büyük performans gösterecektir.

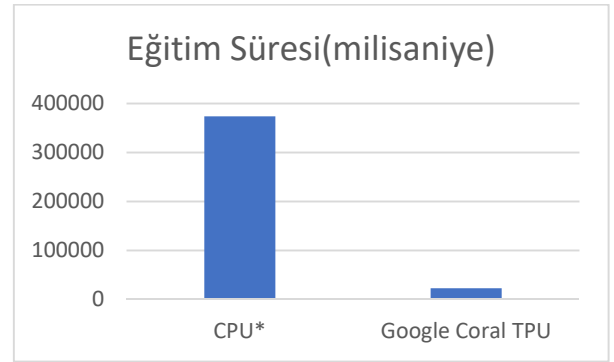
### 3. BULGULAR

Google Coral USB Accelerator ve Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'nun MobileNet v1 modeli ile aynı veri seti için başarımları araştırılmıştır.

Aşağıda Tablo 1'de, MobileNet v1 modelinin eğitilmesi sırasında milisaniye cinsinden CPU ve Google Coral USB Accelerator TPU'nun elde ettiği başarımlar gösterilmiştir.

Tablo 1. Verisetinin CPU ve TPU'daki Eğitim Süresi

Model	Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU	Google Coral USB Accelerator TPU
MobileNet v1	373800	23140



Şekil 4. Veri seti eğitim sürelerinin grafiği

\* Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU

Şekil 4'te Google Coral USB Accelerator 23,14 saniyede veri setinin eğitilmesi işlemini tamamlarken, CPU aynı işlemi 6.23 dakikada tamamlamıştır. Yani Google Coral USB Accelerator'ın Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'dan daha hızlı bir şekilde çalıştığı deneysel olarak ispatlanmıştır.

Bu değerlerden de görüldüğü üzere Google Coral USB Accelerator, derin öğrenme işlemlerinde verimi artırmaktadır.

### 4. SONUÇ

Bu çalışmaya benzer bir çalışma da Google tarafından yapılmıştır. Fakat bu çalışmadan farklı olarak görüntülerin tanınması sırasında ne kadar zaman harcadığı araştırılmıştır.

Tablo 2. Google Tarafından 64-bit Intel(R) Xeon(R) E5-1650 v4 @ 3.60GHz CPU ve Google Coral Accelerator Kullanılarak Elde Edilen Milisaniye Cinsinden Süreler (URL9)

Model Mimarisi	Masaüstü CPU (1)	Masaüstü CPU (1) + USB Accelerator (USB 3.0) Edge TPU ile	Gömülü CPU (2)
DeepLab V3 (513x513)	394	52	1139
DenseNet (*) (224x224)	380	20	1032
Inception v1 (224x224)	90	3.4	392
Inception v4 (299x299)	700	85	3157
Inception-ResNet V2 (299x299)	753	57	2852
MobileNet v1 (224x224)	53	2.4	164
MobileNet v2 (224x224)	51	2.6	122

MobileNet v1 SSD (224x224)	109	6.5	353
MobileNet v2 SSD (224x224)	106	7.2	282
ResNet-50 V1 (*) (299x299)	484	49	1763
ResNet-50 V2 (*) (299x299)	557	50	1875
ResNet-152 V2 (299x299)	1823	128	5499
SqueezeNet (224x224)	55	2.1	232
VGG16 (224x224)	867	296	4595
VGG19 (224x224)	1060	308	5538
EfficientNet-EdgeTpu-S(**)	5431	5.1	705
EfficientNet-EdgeTpu-M(**)	8469	8.7	1081
EfficientNet-EdgeTpu-L(**)	22258	25.3	2717

*“1 Desktop CPU: 64-bit Intel(R) Xeon(R) E5-1650 v4 @ 3.60GHz*

*2 Embedded CPU: Quad-core Cortex-A53 @ 1.5GHz*  
*3 Performansa engel olması nedeniyle bazı işlemlerin CPU üzerinde yürütülmesi gerekiyor*  
*\* Edge TPU çalışma zamanının 10. sürümü tarafından desteklenmiyor.”*

*\*\* CPU'daki gecikme, bu modeller için yüksektir, çünkü TensorFlow Lite çalışma zamanı, tüm platformlardaki eğitim sonrası niceleme modelleri için tam olarak optimize edilmemiştir. “*

Tablo 2’de çeşitli modeller Google tarafından eğitilmiştir ve daha sonra CPU, Google Coral USB Accelerator ve Gömülü CPU üzerinde görüntü sınıflandırma işlemi yapılmıştır. Burada, kullanılan işlemcinin ve modelin görüntü tanıma çalışmalarında zaman açısından önemi incelenmiştir. Görüleceği üzere Google Coral USB Accelerator yalnızca eğitim sürecinde değil, görüntülerin sınıflandırılması sırasında da önemli başarımlar sağlamaktadır. Burada bazı modeller için CPU ve Google Coral USB Accelerator ile yapılan görüntü tanıma süreleri arasında çok büyük fark vardır. Bu durum modelin yapısından kaynaklanmaktadır.

Yeni bir sınıflandırma yaklaşımı sunan bir çalışmada DenseNet, Dense1\_MobileNet, Dense2\_MobileNet ve MobileNet modelleri kullanılarak Caltech-101 ve Uebingen Animals veri setlerindeki görüntüler sınıflandırılmıştır. (Wang ve ark. ,2020) MobileNet için sınıflandırma doğrulukları aşağıdaki gibi elde edilmiştir

**Tablo 3.** Caltech-101 ve Uebingen Animals veri setleri için MobileNet modeli ile elde edilen doğruluk değerleri

Veri seti	Doğruluk
Caltech-101	0.766
Uebingen Animals	0.916

Bu çalışmada bir papatya görüntüsü kullanılarak TPU için 0.99609, CPU için ise 0.8429252 doğruluk değerleri elde edilmiştir. Tablo 3’de görüldüğü üzere Caltech-101 ve Uebingen Animals veri setleri için elde edilen doğruluk değerleri bu çalışmada TPU üzerinde elde edilen doğruluk değerinden düşüktür. CPU için doğruluk değeri değerlendirilecek olursa, Caltech-101 veri seti ile elde edilen değerden yüksek ancak Uebingen Animals veri seti ile elde edilen değerden düşüktür. Bu durum kullanılan veri setinden kaynaklanmaktadır.

Derin öğrenme çalışmalarında genellikle CPU yerine GPU tercih edilir. Fakat çalışmada Tensorflow kütüphanesi kullanılacaksa TPU’nun tercih edilmesi daha uygundur. TPU, Google Coral USB Accelerator cihazı yerine Google Cloud platformu üzerinden veya Google Colab ile kullanılabilir. Buradaki sakıncalardan bir tanesi yapılan çalışmada yazılan kodlar, kullanılan veri seti gibi verilerin karşı taraftaki bilgisayar üzerinde de depolanacak olmasıdır. Bu yolla istenmeden karşı tarafla veriler paylaşılmaktadır. Bunun önüne geçmek için Google Coral USB Accelerator yardımcı işlemcisi kullanılabilir. Böylece düşük donanımsal özelliklere sahip cihazlarda da derin öğrenme çalışmaları daha verimli hale getirilebilir.

## KAYNAKÇA

- A. M. Turing (1950). Computing Machinery and Intelligence. Mind 49: 433-460
- D. E. Rumelhart, G. E. Hinton, R. J. Williams (1988): Learning Representations by Back-Propagating Errors, Cognitive modeling
- Eric B. Olsen (2017) Proposal for a High Precision Tensor Processing Unit
- Frank Rosenblatt (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review Vol. 65, No. 6
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei (2009) Dept. of Computer Science, Princeton University, USA, ImageNet: A Large-Scale Hierarchical Image Database
- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay

Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, (2016) TensorFlow: A system for large-scale machine learning 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) November 2–4, • Savannah, GA, USA

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson ve diğerleri (2017) In-Datacenter Performance Analysis of a Tensor Processing Unit Google, Inc. , Mountain View, CA USA

Tao Sheng, Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, Mickey Aleksic (2018), A Quantization-Friendly Separable Convolution for MobileNets

URL-1:

<https://www.guru99.com/what-is-tensorflow.html> [Erişim Tarihi: 10.03.2020]

URL-2:

<https://cloud.google.com/blog/products/ai-machine-learning/what-makes-tpus-fine-tuned-for-deep-learning> [Erişim Tarihi: 10.02.2020]

URL-3:

<https://medium.com/@antonpaquin/whats-inside-a-tpu-c013eb51973e> [Erişim Tarihi: 10.03.2020]

URL-4:

<https://machinethink.net/blog/mobilenet-v2/> [Erişim Tarihi: 5.02.2020]

URL-5:

<https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolutionlight-weight-model-a382df364b69> [Erişim Tarihi: 10.03.2020]

URL-6:

<https://gist.github.com/ayyucedemirbas/6c2d6bd9324834432df02e8083be9031> [Erişim Tarihi: 23.03.2020]

URL-7:

<https://gist.github.com/ayyucedemirbas/37ce6f12deb9db99715ac398a309285c> [Erişim Tarihi: 5.03.2020]

URL-8:

<https://gist.github.com/ayyucedemirbas/2901b48a1b33eec1fd4794a522c7e204> [Erişim Tarihi: 23.03.2020]

URL-9:

<https://coral.withgoogle.com/docs/edgetpu/benchmarks/> [Erişim Tarihi: 15.03.2020]

Warren McCulloch, Walter Pitts (1943): A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, Vol. 5, p. 115, 133

Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo (2020) A Novel Image Classification Approach via Dense-MobileNet Models



Araştırma Makalesi

## Rasgele Seçim Tabanlı Yer Değiştirme Kutularının Performans İyileştirmesi için Son İşlem Algoritmaları

Yaşar Selim Bahceci<sup>1</sup>, Fatih Özkaynak<sup>\*1</sup>

<sup>1</sup>Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye

### ÖZ

#### Anahtar Kelimeler:

Kaos  
Kriptoloji  
Bilgi Güvenliği

Yer değiştirme kutuları (substitution-box / s-box) önemli bir kriptolojik yapı taşıdır. Bu kriptolojik yapıların tasarımında matematiksel dönüşümleri temel alan tasarım teknikleri özellikle Gelişkin Şifreleme Standardı (Advanced Encryption Standard / AES) olarak bilinen blok şifreleme algoritmasına damgasını vurmuştur. Çünkü AES s-box yapısı en iyi kriptolojik özelliklere sahiptir. Ancak uygulamaya yönelik saldırılar göz önüne alındığında çeşitli zafiyetler ortaya çıkmaktadır. Rasgele seçim tabanlı tasarım tekniklerini temel alan s-box yapılarının ise kriptolojik özellikleri AES s-box yapısından kötü olmasına rağmen uygulama saldırıları özelinde daha başarılıdır. Bu çalışmanın amacı rasgele seçim tabanlı yer değiştirme kutularının performans iyileştirmesini sağlayacak yöntemleri araştırmaktır. Çalışmada önerilen son işlem algoritması ile rasgele seçim tabanlı tasarımların performans iyileştirmelerinin sağlanabileceği gösterilmiştir. Elde edilen bu sonuçların ileride özellikle uygulamaya yönelik saldırıların engellenmesi için bir karşı önlem olarak önemli katkılar sunacağı düşünülmektedir.

## Post-processing Algorithms for Performance Improvement of Substitution Boxes Based on Random Selection

#### Keywords:

Chaos  
Cryptography  
Information Security

### ABSTRACT

Substitution boxes are an important cryptographic primitive. Design techniques based on mathematical transformations, a design technique of these cryptographic primitives, have played a particularly important role in the block encryption algorithm known as the Advanced Encryption Standard (AES). Because AES substitution box structure has the best cryptographic properties. However, various weaknesses arise when the implementation attacks are considered. Although the cryptographic properties of substitution box structures based on random selection techniques are worse than the AES substitution box structure, they are more successful to prevent implementation attacks. The aim of this study is to investigate the performance improvement algorithms of substitution boxes based on random selection. In this study, it has been shown that performance improvements of substitution boxes based on random selection can be achieved with the proposed post-processing algorithms. These results are thought to make significant contributions in the future, especially as a countermeasure for the prevention of implementation attacks.

\*Sorumlu Yazar

\*(ozkaynak@firat.edu.tr) ORCID ID 0000-0003-1292-8490  
(yasarselimbahceci@gmail.com) ORCID ID 0000-0002-7567-4961

e-ISSN: 2717-8579

Geliş Tarihi: 21/03/2020; Kabul Tarihi: 05/05/2020

Bilgisayar Bilimleri ve Teknolojileri Dergisi

## 1. GİRİŞ

Bir kriptolojik protokolün sağlaması gereken temel gereksinimler karıştırma ve yayılma özellikleridir. Blok şifreleme algoritmaları özelinde karıştırma özelliğini sağlamak için yer değiştirme kutuları (substitution boxes/s-boxes) olarak adlandırılan doğrusal olmayan dönüşümler kullanılmaktadır (Cusick ve Stanica, 2009; Wu ve Feng, 2016). En basit anlamda s-box yapısı doğrusal olmayan bir fonksiyondur. Literatürde bu doğrusal olmayan fonksiyonların tasarımı için matematiksel, sezgisel ve rasgele seçim tabanlı tasarım teknikleri vardır. En yaygın bilinen tasarım tekniği matematiksel dönüşümleri temel alan tasarım teknikleridir. Matematiksel dönüşümleri temel alan tasarım teknikleri özellikle Advanced Encryption Standart olarak bilinen blok şifreleme algoritmasına damgasını vurmuştur. Çünkü AES s-box yapısı en iyi kriptolojik özelliklere sahiptir (Cusick ve Stanica, 2009; Wu ve Feng, 2016). Ancak bu tasarım tekniklerinin uygulamaya yönelik saldırılar için çeşitli problemlere sebep olabileceği gösterilmiştir. Bu yüzden rasgele seçimleri temel alan tasarım teknikleri giderek popüler olmaya başlamıştır (Açıkkapı ve ark., 2019).

Rasgele seçim tabanlı tasarım teknikleri arasında kaos tabanlı s-box yapıları giderek popüler olmuştur. Bu popülerliğin en önemli sebeplerinden biri kaosu doğrusal olmayan ve tahmin edilemez yapısı olarak gösterilebilir (Kocarey ve Lioan, 2011; Sprott, 2010). Ancak kaos tabanlı tasarımların performans karakteristiklerinin matematiksel tabanlı tekniklere göre kötü olması ciddi bir problemdir (Özkaynak, 2017). Bu çalışmanın amacı bu problemin etkisi azaltacak son işlem algoritmalarını araştırmaktır. Çalışmada önerilen son işlem algoritmasının s-box performans ölçütlerinden biri olan doğrusal olmama özelliğini iyileştirebileceği gösterilmiştir.

Çalışmanın geri kalan kısmı aşağıdaki gibi organize edilmiştir. İkinci bölümde kaos tabanlı s-box literatürü için kısa bir tanıtım verilmiştir. Üçüncü bölümde performans iyileştirmesi için önerilen yöntemin detayları açıklanmıştır. Dördüncü bölümde analiz sonuçları verilmiştir. Son bölümde çalışma özetlenmiş ve ileride yapılabilecek çalışmalar için önerilerde bulunulmuştur.

## 2. KAOS TABANLI S-BOX YAPILARI

En genel ifade ile bir kriptolojik s-box yapısının tasarımı Denklem 1'de verildiği gibi doğrusal olmayan bir ve örten bir fonksiyon tasarımı ile ilişkilidir. Denklem 1'de ifade edildiği gibi s-box büyüklüğüne bağlı olarak 0 ile n arasındaki sayılar yine 0 ile n arasındaki sayılara dönüştürülerek karıştırma özelliğinin sağlanması garanti edilmeye çalışılmaktadır. S-box yapılarının kriptolojik özelliklerini değerlendirmek için fonksiyonun bijektive olmasının yanı sıra doğrusal olmama, giriş

çıkış bağımsızlığı, katı çığ kriteri ve diferansiyel kriptanalizle ilişkili olan XOR dağılım tablosu gibi kriterler bulunmaktadır (Cusick ve Stanica, 2009; Wu ve Feng, 2016; Özkaynak, 2019).

### Denklem 1: S-Box yapısı tasarımı

$$S(n): [0: n] \rightarrow [0: n]$$

S-box tasarımı blok şifreleme algoritmalarının tarihsel gelişiminde önemli bir rol oynamıştır. Data Encryption Standart (DES) algoritmasında kullanılan s-box yapısının tasarım prensiplerinin açıklanmamış olması, diferansiyel saldırılara karşı dirençli olmadığı gösterilmesi araştırmacıları yeni bir blok şifreleme algoritması tasarım sürecine yönlendirmiş ve 2000'li yılların başında AES algoritması ortaya çıkmıştır. AES algoritmasında kullanılan s-box yapısı Nyberg tarafından indirgenemez polinomlarda ters haritalama yöntemi kullanılarak tasarlanmıştır. DES s-box yapısının problemlerini gideren bu tasarım güçlü matematiksel fonksiyonlara dayandığı için kriptolojik özellikler bakımından herhangi bir zayıflık içermediği görülmüştür. AES s-box yapısının kriptolojik özellikleri Tablo 1'de verilmiştir (Cusick ve Stanica, 2009; Wu ve Feng, 2016).

**Tablo 1.** AES blok şifreleme algoritması kriptolojik özellikleri

Kriptolojik Özellik	Değer
Doğrusal Olmama (Nonlinearity)	112
Katı Çığ Kriteri (SAC)	0.5
Giriş/Çıkış Bitleri- Nonlinearity	112
Giriş/Çıkış Bitleri- SAC	0.5
XOR Dağılım Tablosu	4

Her ne kadar AES s-box yapısı kriptolojik özellikler bakımından zayıflıklar içermese de kriptanaliz tekniklerinin gelişmesi ve çeşitlenmesi ile çeşitli zayıflıkların olabileceği ortaya koyulmuştur. Bu saldırı tekniklerinden biri uygulamaya yönelik saldırılardır. Bir uygulamaya yönelik saldırı tekniği olan yan kanal saldırıları bu zafiyeti ve rasgele seçim tabanlı s-box tasarımlarının bu zafiyete karşı bir alternatif olabileceğini göstermiştir (Tanyıldızı ve Özkaynak, 2019; Solami ve ark., 2018). Literatürde en etkili rasgele seçim tabanlı s-box yaklaşımlarından biri entropi kaynağı olarak kaotik sistemleri temel alan tasarımlar olmuştur (Yi ve ark., 2019; Naseer ve ark., 2019; Özkaynak, 2020). Çok farklı kaotik sistemler kullanılarak birçok s-box tasarımı önerilmiştir. Her ne kadar kaos tabanlı tasarımlar bir alternatif olma niteliği taşımasına rağmen kriptolojik özelliklerinin matematiksel tabanlı s-box yapılarına göre daha kötü olması bir problem olarak bu konuda çalışan araştırmacıların bir problemi olarak ortaya çıkmıştır (Özkaynak, 2017). Üçüncü bölümde bu problemi gidermek için yeni bir son

işlem algoritması önerilmiştir. İleride bu son işlem algoritmalarının özellikle rasgele sayı üreticilerinin (Avaroğlu ve ark., 2015; Koyuncu ve ark., 2017) performans iyileştirmesi içinde bir alternatif olarak değerlendirilebileceği düşünülmektedir.

### 3. ÖNERİLEN YÖNTEM

Bu çalışmada önerilen son işlem algoritmasının s-box performans ölçütleri üzerindeki etkisini en iyi şekilde analiz edebilmek için AES s-box yapısına benzer olarak 16x16 boyutunda s-box yapıları temel alınmıştır. Önerilen son işlem algoritmasının temel prensibi kaotik sistem çıkışlarını s-box değerlerine dönüştürülmesinin ardından elde edilen tablonun satır ve sütun pozisyonlarının karıştırılması prensibine dayanmaktadır. Bu süreçte sekiz farklı olasılık incelenerek s-box performans ölçütlerine olan etkileri analiz edilmiştir. Bu olasılıklar Tablo 2'de verilmiştir.

**Tablo 2.** Önerilen Son İşlem Teknikleri

İşlem No	Açıklama
1	i numaralı satır (i-1) defa dairesel olarak sola kaydırılır.
2	i numaralı satır (i-1) defa dairesel olarak sağa kaydırılır.
3	i numaralı sütun (i-1) defa dairesel olarak yukarı kaydırılır.
4	i numaralı sütun (i-1) defa dairesel olarak aşağı kaydırılır.
5	i numaralı satır (i-1) defa dairesel olarak sola kaydırılır. Ardından i numaralı sütun (i-1) defa dairesel olarak yukarı kaydırılır.
6	i numaralı satır (i-1) defa dairesel olarak sağa kaydırılır. Ardından i numaralı sütun (i-1) defa dairesel olarak yukarı kaydırılır.
7	i numaralı sütun (i-1) defa dairesel olarak aşağı kaydırılır. Ardından i numaralı satır (i-1) defa dairesel olarak sola kaydırılır.
8	i numaralı sütun (i-1) defa dairesel olarak aşağı kaydırılır. Ardından i numaralı satır (i-1) defa dairesel olarak sağa kaydırılır.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Original s-box

1	2	3	4
6	7	8	5
11	12	9	10
16	13	14	15

(a)

1	2	3	4
8	5	6	7
11	12	9	10
14	15	16	13

(b)

1	6	11	16
5	10	15	4
9	14	3	8
13	2	7	12

(c)

1	14	11	8
5	2	15	12
9	6	3	16
13	10	7	4

(d)

1	7	9	15
6	12	14	4
11	13	3	5
16	2	8	10

(e)

1	5	9	13
8	12	16	4
11	15	3	7
14	2	6	10

(f)

1	14	11	8
2	15	12	5
3	16	9	6
4	13	10	7

(g)

1	14	11	8
12	5	2	15
3	16	9	6
10	7	4	13

(h)

**Şekil 1.** Çalışmada önerilen detayları Tablo 2'de verilen sekiz son işlem algoritmasının çıktıları

Tablo 2'de listelenen bu son işlemlerin mantığını etkili bir şekilde ifade edebilmek için 4x4 boyutunda küçük bir s-box yapısı için sonuçlar grafiksel olarak Şekil 1'de gösterilmiştir.

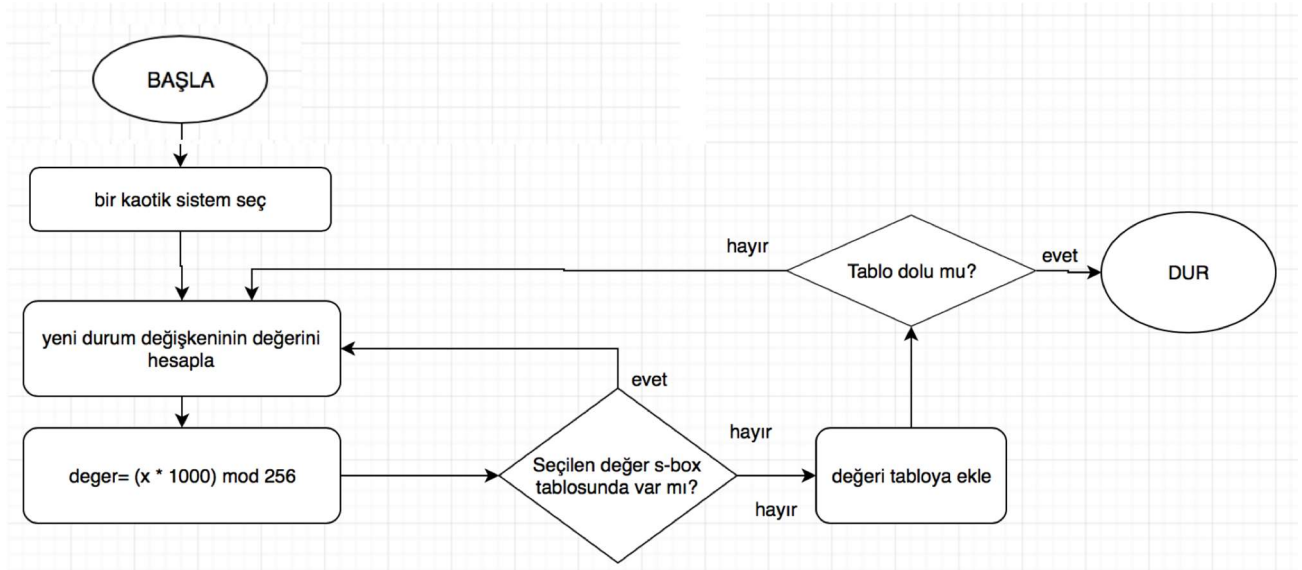
### 4. ANALİZ SONUÇLARI

S-box yapısı oluşturulurken son işlem tekniğinin başarısını ön plana çıkarmak için

mümkün olduğunca basit bir yaklaşım kullanılmaya çalışılmıştır bu yüzden çok güçlü bir tek yönlü fonksiyon olan mod fonksiyonu kullanılarak kaotik sistem çıkışları 0 ile 255 arasındaki değerlere dönüştürülmüştür. Kaotik sistemleri temel olarak s-box oluşturan algoritmanın akış şeması şekil 2’de verilmiştir. Bu algoritma ve s-box değerlendirme kriteri olarak kullanılacak ölçümler hakkında daha detaylı bilgiler için Kaynak (Özkaynak, 2019) incelenebilir. Kaynak (Özkaynak, 2017)’deki algoritma kullanılarak elde edilen biri ayrık zamanlı diğeri sürekli zamanlı kaotik sistem için üretilen iki farklı s-box yapısı sırasıyla Tablo 3 ve Tablo 4’de verilmiştir.

Her iki s-box yapısı için bölüm 3’de çalışma mantığı açıklanan sekiz farklı son işlem algoritması

için 16 farklı s-box elde edilmiştir. Son işlemler sonucunda üretilen s-box yapılarının performans üzerinde olumlu etkileri olduğu görülmüştür. Önerilen yöntemin başarısını genelleştirmek için her bir son işlem algoritması için beş farklı s-box yapısı kullanılarak toplam 40 s-box yapısı elde edilmiştir. Bu yeni s-box yapılarının 34’ünde performans iyileşmesi gözlemlenmiştir. Başka bir iade ile %85 oranında bir başarı elde edilmiştir. İlerideki çalışmalarda daha fazla deneme yapılarak yöntemin başarı yüzdesinin artabileceği düşünülmektedir. Yakın zamanda yayınlanan çalışmalar bu görüşü desteklemektedir (Artuğer ve Özkaynak, 2020).



Şekil 2. S-box oluşturmak için kullanılan algoritma

Tablo 3. Örnek s-box yapısı 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	100	169	155	98	235	49	152	73	19	81	36	43	9	216	62	195
1	18	247	147	105	29	137	75	232	57	237	173	16	224	163	82	165
2	35	162	97	21	225	193	63	4	74	54	68	61	218	177	146	71
3	77	66	238	83	122	240	151	139	133	25	205	161	220	183	254	39
4	33	154	243	17	55	191	182	106	255	78	96	58	233	171	221	253
5	40	129	46	28	87	67	119	53	174	244	234	101	142	89	126	229
6	190	24	236	166	104	145	23	136	56	198	90	197	181	201	217	204
7	65	143	22	84	91	131	251	214	112	207	231	48	252	248	228	102
8	37	189	69	60	168	20	245	116	10	76	227	156	188	44	196	113
9	95	246	42	215	167	223	175	128	7	14	230	239	209	250	5	88
A	202	51	164	120	158	206	213	38	0	72	138	186	176	125	3	2
B	222	184	70	212	47	121	144	134	117	132	123	85	50	199	13	208
C	118	80	111	109	200	45	178	6	30	93	160	64	241	79	11	194
D	12	210	130	110	108	185	107	124	226	41	242	1	192	114	149	148
E	187	115	99	94	31	203	157	179	86	52	141	26	59	15	180	140
F	127	159	34	135	249	8	92	211	27	153	170	172	103	150	32	219



**Tablo 4.** Örnek s-box yapısı 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	100	169	235	49	9	19	73	152	43	98	195	105	147	16	224	162
1	57	81	163	137	36	21	61	165	232	62	216	75	237	54	18	254
2	133	97	83	151	238	218	183	240	29	74	173	33	55	17	205	66
3	4	154	25	129	58	39	146	67	78	161	220	106	40	229	35	221
4	191	177	190	24	139	53	82	126	253	71	166	119	225	46	96	247
5	56	198	90	182	68	171	84	89	63	102	231	155	197	255	201	217
6	145	251	244	77	142	22	207	65	243	174	167	91	23	181	101	193
7	14	87	230	37	112	227	252	245	143	28	164	76	209	136	131	48
8	223	189	236	158	2	168	250	10	206	234	125	38	188	3	122	69
9	116	138	50	228	199	88	5	20	214	13	70	104	117	175	44	72
A	212	60	184	42	64	156	186	132	51	222	204	202	0	239	111	176
B	45	93	130	144	246	215	30	123	113	80	11	248	47	95	241	12
C	210	233	118	178	121	6	124	213	226	134	149	208	85	109	1	203
D	7	192	187	41	128	194	26	200	196	110	108	159	94	160	179	86
E	107	242	31	120	114	140	115	34	15	59	185	135	52	8	180	148
F	27	249	153	92	141	127	211	170	79	157	150	103	172	99	32	219

## 5. SONUÇLAR

Bu çalışmada kaotik s-box yapıların performans kriterlerinin işlem sonrası algoritma kullanılarak daha da iyileştirilip iyileştirilemeyeceği araştırılmıştır. Sekiz farklı son işlem algoritması önerilmiştir. Analiz sonuçlarının sonunda aşağıdaki çıkarımlar yapılabilir.

- Analiz sonuçları, önerilen son işlem algoritmalarının s-box yapılarının kriptolojik özelliklerine olumlu katkıda bulunabileceğini göstermiştir.
- Satır tabanlı son işlem algoritmalarının s-box performans ölçütlerinde sütun tabanlı son işlem algoritmasından daha etkili olabileceğini göstermiştir.
- Çoklu son işlem tekniklerinin kombinasyonlarının birlikte uygulanmasının performans üzerinde her zaman olumlu bir etkisi olmayabilir.
- Katı çıg kriteri için ideal değer olan 0,5'e çok yakın sonuçların elde edilebileceği gösterilmiştir.
- Katı çıg kriteri ile XOR dağılımı arasında genellikle doğrusal bir ilişki gözlemlenmiştir.
- Bazı son işlem teknikleri doğrusal olmama özelliğini etkilerken, bazıları XOR dağıtım özelliğini geliştirmiştir.

Analizde kullanılan her iki s-box yapısı, ortalama şifreleme özelliklerine sahip s-box'lardır. İşlem sonrası algoritmaların s-box performans kriterleri üzerinde olumlu bir etkiye sahip olmasına rağmen, ileri çalışmalar, sonraki işlemlerde post-processing algoritmaların daha büyük bir s-box seti

üzerindeki etkilerini genelleştirebilecektir. Ayrıca, bu etkinin uygulama (yan kanal) saldırılarına karşı saldırıları önlemek için yapacağı olumlu katkılarla analiz edilmesi planlanmaktadır.

## BİLGİLENDİRME/TEŞEKKÜR

Bu makale TEKF 19.18 numaralı proje kapsamında Fırat Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafında desteklenmiştir.

## KAYNAKÇA

- Açikkapi M. S., Özkaynak F., & Özer A. B., (2019). Side-channel analysis of chaos-based substitution box structures, IEEE Access, vol. 7, pp. 79030-79043, 2019. doi: 10.1109/ACCESS.2019.2921708.
- Artuğer F. & Özkaynak F., (2020). A Novel Method for Performance Improvement of Chaos-Based Substitution Boxes, Symmetry 12 (4), 571.
- Avaroğlu E., Koyuncu I., Özer A.B. & Türk M., (2015). Hybrid pseudo-random number generator for cryptographic systems. Nonlinear Dyn 82, 239-248. <https://doi.org/10.1007/s11071-015-2152-8>.
- Cusick T. & Stanica P. (2009). Cryptographic Boolean Functions and Applications. Amsterdam, The Netherlands: Elsevier.
- Kocarev L. & Lian S. (2011). Chaos Based Cryptography Theory Algorithms and Applications. Berlin, Germany: Springer-Verlag.

- Koyuncu İ. & Özcerit A. T. (2017). The design and realization of a new high speed FPGA-based chaotic true random number generator, *Computers & Electrical Engineering*, Volume 58, February 2017, Pages 203-214.
- Naseer Y., Shah T., Shah D., & Hussain S. (2019). A novel algorithm of constructing highly nonlinear s-p-boxes, *Cryptography*, vol. 3, no. 1, p. 6. doi: 10.3390/cryptography3010006.
- Özkaynak F. (2019). An analysis and generation toolbox for chaotic substitution boxes: A case study based on chaotic labyrinth rene thomas system, *Iranian J. Sci. Technol.-Trans. Elect. Eng.*, pp. 1-10. doi: 10.1007/s40998-019-00230-6.
- Özkaynak F. (2017). Construction of robust substitution boxes based on chaotic systems, *Neural Comput. Appl.*, pp. 1-10. doi: 10.1007/s00521-017-3287-y.
- Özkaynak F. (2020). On the effect of chaotic system in performance characteristics of chaos based s-box designs, *Physica A: Statistical Mechanics and its Applications*, 124072.
- Özkaynak F. (2017). Role of NPCR and UACI tests in security problems of chaos based image encryption algorithms and possible solution proposals, 2017 International Conference on Computer Science and Engineering (UBMK) DOI10.1109/UBMK.2017.8093481.
- Solami E. A., Ahmad M., Volos C., Doja M. N., & Beg M. M. S. (2018). "A new hyperchaotic system-based design for efficient bijective substitution boxes," *Entropy*, vol. 20, no. 7, p. 525. doi: 10.3390/e20070525.
- Sprott J. (2010). *Elegant Chaos Algebraically Simple Chaotic Flows*. World Scientific.
- Tanyıldızı E. & Özkaynak F. (2019). A New Chaotic S-Box Generation Method Using Parameter Optimization of One Dimensional Chaotic Maps, VOLUME 7. DOI 10.1109/ACCESS.2019.2936447.
- Tuna M., Alçın M., Koyuncu İ., Fidan C. B. & Pehlivan İ. (2019). High speed FPGA-based chaotic oscillator design, *Microprocessors and Microsystems*, Volume 66, April 2019, Pages 72-80.
- Wu C. & Feng D. (2016). *Boolean Functions and Their Applications in Cryptography*. Berlin, Germany: Springer.
- Yi L., Tong X., Wang Z., Zhang M., Zhu H., & Liu J. (2019). "A novel block encryption algorithm based on chaotic s-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 53079-53090. doi: 10.1109/ACCESS.2019.2911395.



Araştırma Makalesi

## Ortaöğretimde Öğrenim Gören Öğrenci Performanslarının Veri Madenciliği Yöntemleri İle Tahmin Edilmesi

Senem Hazel Başer\*<sup>1</sup>, Onur Hökelekli<sup>1</sup>, Kemal Adem<sup>1</sup>

<sup>1</sup>Aksaray Üniversitesi, Sosyal Bilimler Enstitüsü, Yönetim Bilişim Sistemleri, Aksaray, Türkiye

**Anahtar Kelimeler:**  
Öğrenci performansı  
Veri madenciliği  
Sınıflandırma  
OneR

### ÖZ

İnsanların günlük yaşantılarında sergiledikleri davranışlarının analizi sürecinin gerçekleştirilmesi sonrasında belli metodolojiler ve sınıflandırma yöntemleri kullanılarak, kendi başına öğrenme ve ölçümler yapabilmeye olanak tanıyan veri madenciliği yöntemleri ile insan beyninin öğrenmeye olan eğilimi ve sistematığına benzer paradigmlar üretilmeye çalışılmaktadır. Bu yöntemler kullanılarak eğitim, öğrenme, karar verme süreçlerinin bir makineye uygulanabilmesi amaçlanmaktadır. Bu doğrultuda analizi gerçekleştirilen veri kümesi UC Irvine Machine Learning Repository veri tabanından elde edilmiştir. Söz konusu içeriği oluşturan okul raporları ve anketlerden elde edilen veri üzerinde 31 öznitelikten oluşan veri kümesi kullanılmıştır. Veri içeriğindeki öznitelikler öğrencinin; notları, demografik öğeleri, sosyal ve okul faaliyetlerini kapsamaktadır. Bu kapsamda gerçekleştirilen çalışmada Iterative Classifier, OneR, LogitBoost ve Yapay Sinir Ağı veri madenciliği yöntemleri ile analizler gerçekleştirilmiştir. En başarılı yöntem OneR veri madenciliği yöntemi olmuştur ve bu yöntem ile 5, 7 ve 10 fold çapraz geçerlilik deneyleri yapılarak bu sonuçların ortalamalarına bakıldığında doğruluk değerinin %92.15; duyarlılık değerinin %96; özgüllük değerinin %86.6 olduğu görülmektedir.

## Predicting the Performance of Students Studying in Secondary Education Using Data Mining Methods

**Keywords:**  
Student performance  
Data mining  
Classification  
OneR

### ABSTRACT

After the process of analyzing the behaviors of people in their daily lives, by using certain methodologies and classification methods, it is tried to produce paradigms similar to the human brain's inclination and systematics with data mining methods that can make learning and measurements on its own. Using these methods, it is aimed to apply training, learning, and decision making processes to a machine. The data set analyzed accordingly was obtained from the UC Irvine Machine Learning Repository database. A data set consisting of 31 attributes was used on the data obtained from school reports and questionnaires that constitute the content in question. Attributes in the content of the student; notes, demographic items, social and school activities. In this study, analyzes were carried out with Iterative Classifier, OneR, LogitBoost and Artificial Neural Network data mining methods. The most successful method was the OneR data mining method, and with this method, 5, 7 and 10 fold cross-validation experiments were performed. As a result of the experiments, it is seen that 92.15% of the accuracy value; 96% of the sensitivity value; 86.6% of the specificity value.

\*Sorumlu Yazar

\*(senem.hazel.baser@asu.edu.tr) ORCID ID 0000-0002-1223-1901  
(onur.hokelekli@asu.edu.tr) ORCID ID 0000-0002-9376-9024  
(kemaladem@aksaray.edu.tr) ORCID ID 0000-0002-3752-7354

e-ISSN: 2717-8579

Geliş Tarihi: 13/03/2020; Kabul Tarihi:01/05/2020

Bilgisayar Bilimleri ve Teknolojileri Dergisi

## 1. GİRİŞ

Başarı kavramı en temel anlamıyla belli hedefler doğrultusunda gerçekleştirilen özverili çalışmalar sonucunda istenilen sonuçların elde edilmesi olarak tanımlanabilir. Öğrencilerin başarılarını ve performanslarını etkileyen faktörler vardır. Değişik yaşamları olan kişilerin ortak bir sınıf içerisinde eğitim alması, kişilerin farklılıklarına dikkat edilmemesine neden olmuştur. Fakat aynı sınıfta birbiriyle eşit olarak görülen öğrencilerin bilgi edinme ve öğrenme yolları birbirinden farklıdır. Bu durumu kanıtlayan durum da aynı sınıfta aynı eğitimi gören öğrencilerin başarılarının değişkenlik göstermesidir. Sınıf ortamında öğrencilerin başarısını ve öğrenmelerini etkileyen çeşitli faktörler mevcuttur (Arslan ve Babadoğan, 2005). Öğrenciler arasındaki farklılıklar ve performansları üzerine yapılan çalışmalar ve geliştirilen metodolojilerin artması, gelecekte eğitimdeki başarı düzeylerinde yukarı yönlü bir eğilim potansiyeli sunmaktadır.

Literatür taraması sonucunda öğrenci performansını etkileyen temel ölçütler; bireyin geçmişte aldığı eğitimin kalitesi, ebeveynlerin aldıkları eğitim düzeyleri, ailenin aylık ortalama gelir düzeyi, öğrencilerin devam etmekte oldukları bölüm, okul ortamında oluşan çevrenin verdiği memnuniyet duygusu ve bireyin mevcut psikolojik durumu olduğuna yönelik fikir birliği vurgulanmıştır. Literatür incelemesinde öğrenci performansını tahminleme üzerine ilk çalışma 1994 yılında Nagin Gorr ve Szcypula tarafından yapılmıştır. Bu çalışmada öğrencilerin aldıkları notların ortalamalarının tahminlenmesinde Doğrusal Regresyon ve Çoklu Regresyon Analizi ile Yapay Sinir Ağları (YSA) analizinin karşılaştırılması yapılmıştır. Elde edilen sonuçlar değerlendirildiğinde YSA yönteminin kullanılmasıyla yapılan tahminlemenin daha isabetli sonuçlar ürettiği tespit edilmiştir. Başka bir çalışmada ise Subbanarasimha, Arizeb ve Anandarajanb (2000), belirli bir grup öğrencinin elde ettikleri başarının akademik performans düzeylerinin tahmininin yapılmasında Regresyon yöntemleri ile YSA yöntemlerinin karşılaştırılması için iki farklı değişkenli veri kümesi kullanılmıştır. Buradan elde edilen sonuçlarda YSA tahminleme tekniklerinin bu veri kümesi üzerinde daha iyi değerler ürettiği sonucuna varılmıştır. Tosun (2007), öğrenci performansı üzerine yaptığı çalışmada Karar Ağaçları ve YSA yöntemlerini ele almıştır. Karar Ağaçları baz alındığında %86 oranında başarı elde ederken aynı veri kümesi üzerinde YSA kullanıldığında %92 oranında başarı sağlanmıştır. Literatürde öğrencilerin akademik performansını üzerine gerçekleştirilen güncel çalışmalar incelendiğinde; Aydemir, E. (2019), Türkiye’de bir üniversitede öğrenim gören öğrencilerin yabancı dil dersindeki geçer notlarının YSA ve bazı sınıflama metodları ile tahminleme

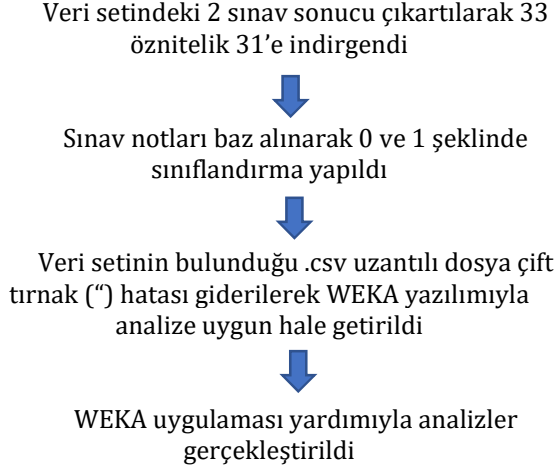
modelleri oluşturulmuş ve bu metodların karşılaştırılması yapılmıştır. Bu çalışmada veriler eğitim ve test olarak ayrıştırılarak sınımlar yapılmış ve sınımla metodları arasında Bagging yönteminde 1.22 ortalama mutlak hata ve 0.80 korelasyon katsayısı ile en başarılı tahminler üretildiği görülmüştür. Güncel diğer bir çalışma; Güre, Kayri ve Erdoğan (2020), matematik okuryazarlığını etkileyen unsurları tespit etmek için veri madenciliği metodlarından Rastgele Orman ve Çok Katmanlı Algılayıcı yöntemlerinin tahminleme becerilerini karşılaştırılmıştır. Yapılan testler ile öğrencilerin aldıkları puanların analizi neticesinde Rastgele Orman metodunun daha az hata ile tahminleme yaptığı görülmüştür. Tahminleme performansı yüksek olan metod çıktıları matematik okuryazarlığına etki eden unsurlar olarak kabul edilmiştir. Başka bir çalışma, Altun, Kayıkçı ve İrmak (2019), tarafından yapılan bir araştırmadır. Çalışmada sınıf öğretmenliği öğrencilerinin vize sınav puanlarının dikkate alınarak final notlarının tahminlenmesi amaçlanmıştır. Bu doğrultuda analizi gerçekleştirilen veriler, Çoklu Doğrusal Regresyon ve YSA yöntemleri karşılaştırılmıştır. Değerlendirme çıktılarında regresyon analizi %94.30 başarı sağlarken YSA ile %94.43 başarı elde edilmiş, öğrenci başarısını tahminlemede birbirine yakın sonuçlar verdiği görülmüştür.

Bu çalışmada ortaöğretimde öğrenci başarı ölçütünün belirlenmesinde bireysel ve demografik faktörlere bağlı sınıflandırmalar için pek çok veri madenciliği yöntemi ile deney yapılmıştır. Deneyler içerisinde en başarılı olan yöntemler Iterative Classifier Optimizer, OneR, LogitBoost ve YSA yöntemleri olmuştur. Analiz için UCI üzerinden edinilen Student Performance Dataset kullanılmıştır. Bu veri seti ile yapılan ilk veri madenciliği çalışmaları kümeleme algoritmaları ile gerçekleştirilmiştir. Literatür incelendiğinde ilerleyen çalışmalarda farklı algoritmalar kullanılarak sınıflandırma analizleri de yapılmıştır. Bu çalışmada veri seti sınıflama algoritmalarının kullanılmasına uygun hale getirilerek veri madenciliği çalışmaları yapılmıştır. Verilerin düzenlenmesi sonrasında yapılan analizlerde yüksek başarı oranlarına ulaşılmıştır. Yapılan deneyler sonucunda en başarılı yöntem diğer veri madenciliği algoritmalarından daha yüksek sonuçlar vererek OneR yöntemi olmuştur.

## 2. YÖNTEM

Öğrenci başarı düzeylerini veri madenciliği tekniklerini kullanarak sınıflandırmak için UC Irvine Machine Learning Repository (UCI) üzerinden elde edilen Student Performance Dataset içerisindeki veriler, 395 öğrenci (187’si Erkek, 208’i kız) ve 33 öznitelikten oluşmaktadır (Cortez, ve Silva, 2008). Bu 33 öznitelik içerisinde bulunan öğrencilerin matematik dersine ait 3 sınav sonucu yer almaktadır. Veri setinin edinildiği web sitesi

üzerinden edinilen bilgiye göre en anlamlı sınav sonucunun 3. sınav sonucu olduğu kanısına varılmıştır. Bu sebeple sonuçlardan yalnızca son sınav notları baz alınarak sınıflandırma gerçekleştirildi. 0-20 aralığında değerlendirilen sınav sonuçları için 0-10 aralığında puana sahip olan öğrenciler için "0" sınıfı ile başarısız, 11-20 aralığında sınav notu olan öğrenciler için ise "1" sınıfı ile başarılı olarak sınıflandırıldı.



Çalışmada kullanılan veri madenciliği yöntemlerinin ayrıntıları aşağıda verilmiştir.

#### a. Iterative Classifier Optimizer

Iterative Classifier Optimizer algoritması, ilk kaydın sınıflandırmasından elde edilen hataları geri besleyerek daha fazla yineleme için değiştirmektedir. Bu algoritma nöron ağı gibi çalışır ve bu nedenle beynin sinir yapısı ile karşılaştırılabilir. Bu yöntemde kayıtlar ağa dağıtılır. Tüm giriş örnekleri sunulduktan sonra süreç tekrarlanır ve bu nedenle bu algoritma YSA'nın kilit bir özelliğidir. Sinir ağını yapılandırarak belirli bir uygulama için eğitilebilir. Öğrenme başlangıç ağırlıklarını rastgele seçerek başlar (Manikandan ve ark., 2018).

#### b. OneR

Holte (1993), tarafından önerilen OneR yöntemi, karar ağacını öğrenir. Temelde bu yaklaşım hata tabanlı kural çıkarım anlayışına dayanır. Sınıflandırmanın doğru sonuçlara yaklaşması için iyileştirmeler yapar. Hata oranı en az olan ve sınıflandırmada farkı ortaya koyacak en iyi kriteri seçer. Yalnızca tek özelliğe odaklandığı için OneR yüzeysel bir yaklaşım olarak görülmektedir (Uzun, 2005). OneR algoritması (Holte, 1993) aşağıdaki gibi bir mantıkla çalışır: Her bir özellik için,

- 1.Özelliğin her bir değeri için, aşağıdaki gibi bir kural oluştur.
- 2.Her bir sınıfın ne kadar görüldüğünü say.
3. En sık görülen sınıfı bul.

- 4.Oluşturulan kuralı tanımla, bu özellik değeri ile sınıflandır.
- 5.Kuralların hata oranını hesapla.
- 6.Kuralları en küçük hata oranı ile seç.

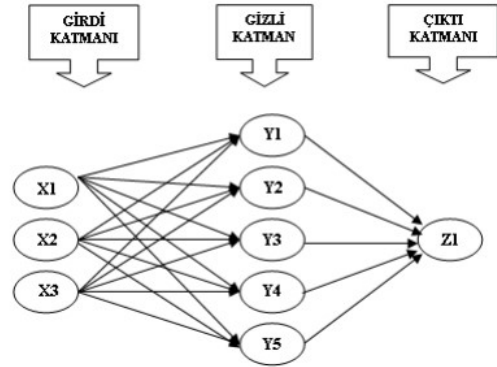
Yukarıda verilen algoritma adımları ile veri kümelerine OneR veri madenciliği yöntemi uygulanmaktadır.

### 2.3. LogitBoost

LogitBoost, boosting algoritmalarının içinde yer alır. AdaBoost'ta yaşanan sorunlara çözüm sunmak için geliştirilmiştir ki bu sorunlar yoğun verilerden ortaya çıkan aşırı öğrenme (overfitting) sorunlarıdır. Bu algoritma, bahsedilen problemi gidermek için doğrusal olarak eğitimdeki hataları indirmektedir. Böylelikle genelleme seviyesini yükseltmektedir. Neticede LogitBoost algoritması, lojistik kayıp fonksiyonunu kullanır ve aşırı öğrenme problemini ortaya çıkaran verilerin ağırlığını artırarak bu soruna çözüm üretmeye çalışmaktadır (Aydın ve Arslan, 2017).

### 2.4. Yapay Sinir Ağları

YSA, nöronlar veya düğümler olarak adlandırılan bir dizi birbirine bağlı basit işleme elemanlarından oluşan biyolojik sinir ağlarından esinlenen hesaplama modelidir. Düğümler, Şekil 1'deki gibi girdi katmanı, gizli katman ve çıktı katmanından oluşur. Her nöron(düğüm), önceki katman nöronlarından veya harici bir kaynaktan bir dizi veri alır, bir aktivasyon fonksiyonu kullanarak yerel olarak dönüştürür ve sonucu bir sonraki düğüme gönderir (Osborn ve ark., 2011).



Şekil 1. YSA çalışma yapısı

Bulduğu ortama uyum sağlayabilen, örnekleri kullanarak öğrenebilen, eksik verilere rağmen çalışabilen bir algoritmadır. Sınıflandırma, örüntü tanıma, sinyal filtreleme, veri sıkıştırma ve optimizasyon gibi çalışmalarda YSA en başarılı yöntemler arasında yer almaktadır. Veri madenciliği, yön belirleme, parmak izi tanıma, malzeme analizi, kalite kontrol, tıbbi analizler gibi

pek çok alanda günlük hayatta karşılaşılabilecek başarılı çalışmalar bulunmaktadır (Öztemel, 2003).

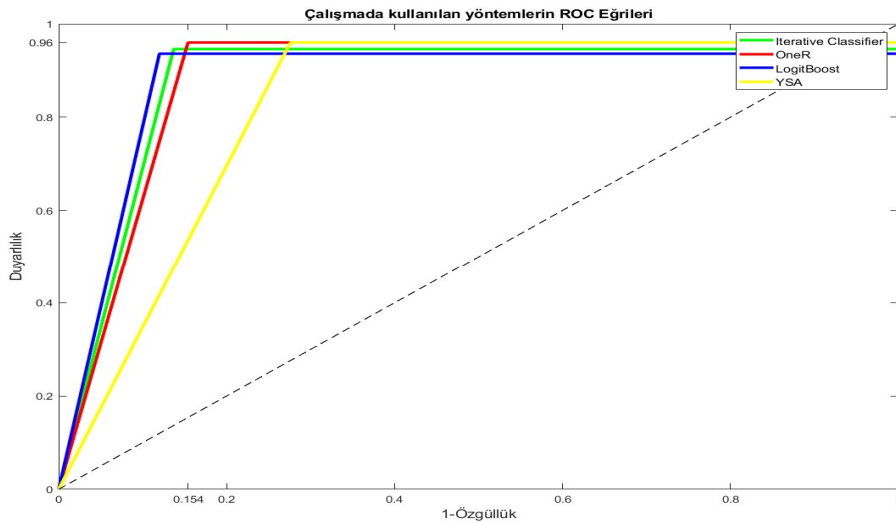
### 3. BULGULAR

Bu çalışmada kullanılan veri seti üzerinde Intel Core i7-4720 HQ işlemciye ve 12 GB belleğe sahip bilgisayar aracılığıyla Weka (URL-1) yazılımı kullanılarak çeşitli veri madenciliği yöntemleri denenmiştir. 3 farklı yöntem ve objektif olarak değerlendirme yapılabilmesi adına her yöntem için 3 ayrı fold değeri ile analizler gerçekleştirilmiştir. Veri madenciliği yöntemleri ile oluşturulan modellerin değerlendirme ölçütü olarak doğruluk, özgüllük ve duyarlılık değerleri kullanılmıştır. Ayrıntılar Tablo 1'de verilmiştir.

**Tablo 1.** Analiz sonuçları

Sınıflandırma Algoritmaları	Fold	Duyarlılık	Özgüllük	Doğruluk
Iterative Classifier Optimizer	5	%96	%86	%92.41
	7	%94	%87	%91.90
	10	%94	%86	%91.39
	Ort.	%94.6	%86.3	%91.9
OneR	5	%96	%84	%92.15
	7	%96	%86	%92.15
	10	%96	%84	%92.15
	Ort.	%96	%84.6	%92.15
LogitBoost	5	%94	%88	%92.15
	7	%93	%88	%91.90
	10	%94	%88	%91.89
	Ort.	%93.6	%88	%91.98
YSA	5	%96.4	%76.6	%89.11
	7	%95.6	%69.6	%86.07
	10	%96	%71	%86.8
	Ort.	%96	%72.4	%87,32

Gerçekleştirilen analizlerin ROC eğrisi Şekil 2'de verilmiştir.



**Şekil 2.** Kullanılan yöntemlerin ROC eğrileri

Veri madenciliği yöntemleri ile oluşturulan modellerin değerlendirme ölçütü olarak doğruluk, özgüllük ve duyarlılık değerleri kullanılmıştır. Veri kümesi üzerinde 5, 7 ve 10 fold değerleri kullanılarak ölçütler için en iyi sonuçlara ulaşılmaya çalışılmıştır. Kullanılan veri kümesi üzerinde yapılan ilk analizler neticesinde doğruluk değerine bakıldığında en iyi performans Iterative Classifier Optimizer (5 fold) ile sağlanmıştır.

Iterative Classifier Optimizer yöntemiyle 5, 7 ve 10 fold için elde edilen sonuçlara bakıldığında en yüksek değerler 5 fold ile elde edilmiştir. OneR yöntemi incelendiğinde ise değerlendirme ölçütlerinde en iyi sonuçları 7 fold vermiştir. LogitBoost yönteminde 5 fold ile yapılan deneyler sonucu en yüksek değerler elde edilmiştir. YSA algoritması ile yapılan analizler neticesinde en yüksek başarı oranı 5 fold ile yapılan deneylerle elde edilmiştir.

Deneylerin gerçekleştirildiği 3 ayrı fold değeri için ortaya çıkan değerlendirme ölçütlerinin her bir yöntem için ortalaması alınmıştır. Bu sonuçlara göre doğruluk değeri için en iyi ortalamanın OneR yönteminde olduğu görülmektedir. Duyarlılık için ortalamalar bazında değerlendirildiğinde en iyi sonucun yine OneR yönteminde olduğunu, özgüllük değerlerinde ortalamalara bakıldığında ise en iyi sonucu LogitBoost yöntemi vermektedir.

Tüm değerlerin ortalamaları baz alındığında özgüllük değeri diğer algoritmalara göre düşük performans gösterse de yaklaşık %2'lik bir farktan bahsedildiği için göz ardı edilebilir ve bu veri seti için en başarılı yöntem olarak OneR önerilmektedir.

#### 4. SONUÇLAR

Öğrenciler yaşamlarının belli dönemlerinde zamanlarının büyük bir çoğunluğunu okul ortamlarında harcamaktadır. Bu süreçte öğrenim hayatlarını etkileyen, karşılaştıkları pek çok olumsuz faktör bulunmaktadır; yeni ortamın getirdiği uyum sorunu, alınan derslerdeki başarısızlıklar, öğretmenler ile ilgili sorunlar, arkadaş çevresi ile ilgili sorunlar, ebeveyn ortamından kaynaklı sorunlar vb. durumlar günümüzde öğrencilerin başarı düzeylerinin belirlenmesinde ana etkenler arasında görülmektedir. Literatür araştırmalarında bu problemin tespit ve çözümlenmesi hususunda birçok YSA ve veri madenciliği çalışmalarının yapıldığı görülmektedir. Bu araştırmada, ortaöğretimdeki öğrencilerin karşılaştıkları sorunların, öğrencinin başarı performansına olumsuz etki eden ana faktörlerin belirlenmesi amaçlanmaktadır. Söz konusu araştırma ortaöğretimde öğrenim gören öğrencilerin başarı düzeylerini etkileyen etmenlerin neler olduğunu ortaya koyması bakımından önem teşkil etmektedir. Çalışmada literatüre kıyasla farklı sınıflandırma algoritmaları kullanılarak analizler gerçekleştirilmiştir. Öğrencinin başarı düzeyinin sınıflandırılması için Iterative Classifier Optimizer, OneR ve LogitBoost ile YSA yöntemleri uygulanmış ve OneR veri madenciliği yönteminin diğer yöntemlere kıyasla daha olumlu sonuçlar sergilediği tespit edilmiştir. Öğrencinin başarı düzeyinin arttırılmasında göz önünde bulundurulması gereken nokta, performans etki eden ana faktörler öncelikle tespit ve teşhis edilmeli, ardından gerekli iyileştirmelerin yapılması önerilmektedir. Bu sayede, performansın gerektirdiği salt faktörler belirlenebilir ve öğrenci başarı eğilimleri farklı konularda yoğunlaştırılabilir. Gelecek çalışmalarda, daha farklı veri madenciliği yöntemleriyle deneyler gerçekleştirilebileceği gibi öğrenci başarısını etkileyen faktörler arasından en etkililerini tespit etmek için analizler gerçekleştirilebilir.

#### KAYNAKÇA

Altun, M., KAYIKÇI, K., & Irmak, S. (2019). Sınıf Öğretmenliği Öğrencilerinin Mezuniyet Notlarının Regresyon Analizi ve Yapay Sinir Ağları Yöntemleriyle Tahmini/Estimation of Graduation Grades of Primary Education Students by Using Regression Analysis and Artificial Neural Networks. *e-Uluslararası Eğitim Araştırmaları Dergisi*, 10(3), 29-43.

Arslan B., & Babadoğan, C. (2005). İlköğretim 7. ve 8. sınıf öğrencilerinin öğrenme stillerinin akademik başarı düzeyi, cinsiyet ve yaş ile ilişkisi. *Eurasian Journal of Educational Research*, 31, 35-48.

Aydemir, E. (2019). Ders Geçme Notlarının Veri Madenciliği Yöntemleriyle Tahmin Edilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (15), 70-76.

Aydın, F., & Aslan, Z. (2017). Yapay Öğrenme Yöntemleri ve Dalgacık Dönüşümü Kullanılarak Nöro Dejeneratif Hastalıkların Teşhisi. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 32(3).

Alves, E. T. A., Frucht, L. C., Souza, M. N., & Nóbrega, A. F. A. (2012, March). Iterative optimization algorithm- An alternative clustering tool for biological analysis using flow cytometry data. *In 2012 Pan American Health Care Exchanges* (pp. 66-70). IEEE.

Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance. *In the Proceedings of 5th Annual Future Business Technology Conference*, Porto, Portugal, 5-12.

Daş, B., & Türkoğlu, İ. (2014). DNA dizilimlerinin sınıflandırılmasında karar ağacı algoritmalarının karşılaştırılması. *Elektrik-elektronik-bilgisayar ve Biyomedikal Mühendisliği Sempozyumu (ELECO 2014)*, 381-383.

Gorr, W. Lç, Nagin, D., & Szczyppula, J. (1994). Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, 10(1), 17-34.

Güre, Ö. B., Kayri, M., & Erdoğan, F. (2020). PISA 2015 Matematik Okuryazarlığını Etkileyen Faktörlerin Eğitsel Veri Madenciliği ile Çözülmesi. *Eğitim ve Bilim*.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1), 63-90.

İstenmeyen Elektronik Posta (Spam) tespitinde karar ağaç algoritmalarının performans kıyaslaması. *Journal of Internet Applications & Management/İnternet Uygulamaları ve Yönetim Dergisi*, 5(2).

Manikandan, G., Aravind, V., & Anitha, B. (2018). A Survey to Identify an Efficient Classification Algorithm for Heart Disease Prediction. *International Journal of Pure and Applied Mathematics*, 119(2), 13337-13345.

Osborn, J., Francisco Javier De, C. J., Guzman, D., Butterley, T., Myers, R., Guesalaga, A., & Laine, J. (2011). *Using artificial neural networks for open-loop tomography*. Ithaca: Cornell

University Library, arXiv.org.  
doi:<http://dx.doi.org/10.1364/OE.20.002420>

Öztemel, E. (2003). Yapay sinir ağları. *PapatyaYayincilik, İstanbul*.

Subba Narasimha, P. N., Arinze, B., & Anandarajan, M. (2000). The predictive accuracy of artificial neural network and multiple regression in the case of skewed data. *Exploration of some issues. Expert systems with Applications*, 19(2), 117-123.

Tosun, S. (2007). *Sınıflandırmada yapay sinir ağları ve karar ağaçları karşılaştırması: öğrenci başarıları üzerine bir uygulama*. (Yayımlanmamış yüksek lisans tezi). İstanbul Teknik Üniversitesi/Fen Bilimleri Enstitüsü, İstanbul.

Uzun, Y. (2005). *Tıbbi veriler üzerinde makine öğrenme algoritmaları ve bulanık mantık ile kurallar öğrenme*. (Yayımlanmamış yüksek lisans tezi). Selçuk Üniversitesi/Fen Bilimleri Enstitüsü, Konya.

URL-1:[https://waikato.github.io/weka-wiki/citing\\_weka/](https://waikato.github.io/weka-wiki/citing_weka/) [Erişim Tarihi: 23.04.2020]





Research Article

## Medical Image Classification with Hybrid Convolutional Neural Network Models

Osman Doğuş Gülgün<sup>\*1</sup>, Hamza Erol<sup>2</sup>

<sup>1</sup>Mersin University, Institute of Science, Computer Engineering Department, Mersin, Turkey

<sup>2</sup>Mersin University, Engineering Faculty, Computer Engineering Department, Mersin, Turkey

### ABSTRACT

#### Keywords:

Deep learning  
Medical diagnosis  
Convolutional neural network  
Data augmentation  
Classification

Despite important developments in medicine and technology today, many people die due to false or late diagnosis. It is very important to identify the small details in the images that can be overlooked in the examinations made on medical images in terms of early diagnosis of the disease. Therefore, it is vital in some cases to provide early diagnosis by detecting the details in the images automatically by computer systems. In the study carried out, it was aimed to diagnose the disease through medical images by classifying different types of images. For this purpose, convolutional neural networks, which are among deep learning techniques, were evaluated together with different classifier models. In the applied hybrid model approach, feature extraction was obtained from medical images with the convolutional neural network model. The extracted features are used to train different classification models. In the continuation of the study, the performance results obtained from the classifier models are compared. Two different datasets including brain MR images and lung x-ray images were used in the training and testing of hybrid models. In the study, images were classified into two categories as malignant and benign tumors in order to detect images containing malignant tumors in MR images. In order to identify images with pneumonia, the images are similarly classified into two categories, healthy and pneumonia. At the end of the study, the performance results obtained from the model approaches were compared and the performance evaluation of the models was performed.

## Hibrit evrişimli sinir ağı modelleri ile tıbbi görüntü sınıflandırması

#### Anahtar Kelimeler:

Derin öğrenme  
Tıbbi teşhis  
Evrişimli sinir ağı  
Veri artırma  
Sınıflandırma

### ÖZ

Günümüzde tıp ve teknolojiye önemli gelişmelere rağmen, birçok kişi yanlış veya geç tanı nedeniyle hayatını kaybetmektedir. Tıbbi görüntüler üzerinde yapılan muayenelerde hastalığın erken teşhisi açısından gözden kaçırılacak görüntülerdeki küçük detayların belirlenmesi çok önemlidir. Bu nedenle, bazı durumlarda görüntülerdeki detayları bilgisayar sistemleri tarafından otomatik olarak tespit ederek erken teşhis sağlamak hayati önem taşımaktadır. Yapılan çalışmada, farklı görüntü tiplerini sınıflandırarak hastalığın tıbbi görüntülerle teşhis edilmesi amaçlanmıştır. Bu amaçla, derin öğrenme teknikleri arasında yer alan evrişimli sinir ağları, farklı sınıflayıcı modellerle birlikte değerlendirilmiştir. Uygulanan hibrid model yaklaşımında, evrişimli sinir ağı modeli ile tıbbi görüntülerden özellik çıkarımı elde edilmiştir. Çıkarılan özellikler farklı sınıflandırma modellerini eğitmek için kullanılır. Çalışmanın devamında, sınıflandırıcı modellerinden elde edilen performans sonuçları karşılaştırılmıştır. Hibrid modellerin eğitim ve testinde beyin MR görüntüleri ve akciğer röntgeni görüntüleri dahil olmak üzere iki farklı veri seti kullanılmıştır. Çalışmada MR görüntülerinde malign tümör içeren görüntüleri saptamak için görüntüler malign ve benign tümörler olarak iki kategoriye ayrıldı. Akciğer iltihaplanmalı görüntüleri tanımlamak için görüntüler benzer şekilde sağlıklı ve akciğer iltihaplanması olmak üzere iki kategoriye ayrılır. Araştırma sonunda model yaklaşımlarından elde edilen performans sonuçları karşılaştırılmış ve modellerin performans değerlendirmesi yapılmıştır.

#### \*Correspond Author

(dogusgulgun@gmail.com) ORCID ID 0000-0003-1824-4401  
(herol@mersin.edu.tr) ORCID ID 0000-0001-8983-4797

e-ISSN: 2717-8579

Arrival Date: 24/04/2020; Acceptance Date: 11/05/2020

## 1. INTRODUCTION

In our study, various diagnostic models are explained that can be used for diagnosis of diseases by classification. Hybrid convolutional neural network models that can be used to increase the accuracy rate in the classification of medical images have been described. The hybrid models described in the study were obtained by changing the original convolutional neural network architecture. In these models, different classifiers were used instead of artificial neural networks. Classifiers such as random forest and support vector machines were used as classifiers. Therefore, machine learning and deep learning techniques were evaluated together. In the continuation of the study, some useful deep learning techniques used to improve the performance of the models are also explained. In this study, the medical images of 2 different cases were diagnosed by classification. In the first case, images with pneumonia were detected within the lung x-ray images. Images are classified as "people with pneumonia" and "healthy". In the second case, brain MRI images with benign and malignant tumors were classified. Images are classified as binary, "benign" and "malignant", similar to the first case.

Zhang et al. have proposed a synergistic deep learning model approach to eliminate deep learning techniques from performing poorly in some cases (Zhang et al., 2019). In the developed synergistic model, using two convolutional neural networks, these networks were jointly trained and learned through each other. ResNet-50 architecture was used in the design of convolutional neural networks. ResNet-50 is a 50-layer specific neural network architecture pre-trained and verified by testing its performance (He et al., 2016). The ResNet-50 model has learned high-level features for image classification because it was previously trained with the ImageNet data set, which is a fairly large data set containing millions of image data. If one of the convolutional neural networks used in the synergistic model developed is correctly classified and the other is misclassifying; The error made creates an extra effect for updating the parameters of the model that misclassifies. Therefore, this model is trained mainly on the mistakes made. Therefore, this model learns the classification errors made more effectively. This model developed in the study was evaluated using 4 different data sets and its performance was tested. According to the results obtained at the end of the study, the synergistic deep learning model created has managed to reveal the most successful performance results for each data set.

Oh et al. have proposed a computer-aided diagnostic system that can diagnose Parkinson's disease via EEG signals (Oh et al., 2018). EEG signal records of 20 Parkinson's patients and 20 healthy individuals were used to develop the diagnostic system. Noises in the received EEG signals were

filtered with amplitude and frequency filters to increase diagnostic performance. The researchers used the convolutional neural network architecture from deep learning techniques to classify signals, and proposed a 13-layer convolutional neural network model. The proposed model consists of 1 input layer, 4 (1 x 1) size convolution layers, 4 pooling layers and 1 fully connected layer. There is a 3-layer artificial neural network in the fully connected layer. In the model, 20 filters were used in the first convolution layer, 10 in the second and third convolution layers and 5 in the last convolution layer. In addition, dropout technique was used in the artificial neural network in the fully connected layer in order to avoid overfitting problem in the model. In this way, 50% of the neurons of the artificial neural network in the fully connected layer are disabled every iteration. The model proposed in the study achieved a good performance in diagnosing Parkinson's disease from EEG signal data, achieving 88.25% accuracy, 84.71% sensitivity and 91.77% specificity. The most important advantages of the study carried out compared to other studies in the literature are that the diagnosis of Parkinson is performed directly over EEG signals and there is no need for feature extraction.

Frid-Adar et al. proposed a deep learning-based diagnostic model approach for the classification of liver lesions via computed tomography images (Frid-Adar et al., 2018). They also proposed a data augmentation approach based on deep convolutional generative adversarial network (DCGAN) to improve the classification performance and reliability of their models. In the training of the proposed model, they used a data set containing 182 liver computed tomography images. The data set used contains images belonging to 3 different classes. In the first stage of the study, in order to increase the size of the data set, synthetic image data was produced with the proposed DCGAN model. Synthetic computed tomography images in size (64 x 64 x 1) were produced over random noises using the proposed DCGAN model. As a second data augmentation in addition to this data augmentation method, they also benefited from the classical data augmentation technique, which allows images to be displayed at different angles and at different distances to the model. After the data increase, the convolutional neural network model, another model proposed in the study, was classified with the images in the data set. The proposed convolutional neural network model consists of 3 convolution, 3 pooling and 1 fully connected layer. The artificial neural network in the fully connected layer consists of a total of 2 layers, the first with 256 neurons and the second with 3 neurons. In addition, in order to avoid overfitting problem, dropout technique is used in the fully connected layer. When using only classical data augmentation technique, the proposed model achieved 78.6% sensitivity and 88.4% specificity values. In addition to the classical

technique, using the additional data produced with the GAN model, the model has achieved 85.7% sensitivity and 92.4% specificity values.

Bejnordi et al. proposed a diagnostic model based on the convolutional neural network architecture from deep learning techniques to diagnose breast cancer disease through breast biopsy slide images (Bejnordi et al., 2017). In the training of the model proposed in the study, a data set containing 646 breast biopsy images was used. The model proposed in the study consists of two parts and accordingly, the diagnosis is carried out in two stages. The first convolutional neural network used in the first stage is classified as the epithelial, fat and stroma regions in the images. The model used in the first stage is based on the VGG-Net network architecture and consists of 11 layers. In this section, 3x3 size filters are used in the convolutional neural network. In addition, the ReLU activation function was used in the convolution layers, and (2x2) pooling layers were used after each convolution layer. In the second stage, a different second convolutional neural network model was used to classify the stroma regions as normal or cancer related stroma. The architecture of the second model is based on the VGG-Net architecture as in the first model and consists of 16 layers. Using this model, feature extraction is performed from the stroma regions. Attribute outputs from the second model are then classified into a normal or cancer-related stroma at the last stage using a random forest classifier. Accordingly, breast cancer occurrences are detected by the model in biopsy images. According to the results obtained at the end of the study, the first CNN model, which constitutes the first part of the proposed model, has managed to classify the tissue sections in the images as fat, epithelium and stroma with a 95.5% accuracy rate. In the second stage, the second model used to determine whether the stroma sections are cancer-related or not has managed to accurately predict the classes of stroma sections in the images with an accuracy rate of 92.0%.

Martinez-Murcia et al. proposed a deep learning-based model to diagnose Alzheimer's disease via MRI images (Martinez-Murcia et al., 2019). They proposed a convolutional autoencoder model in their studies to obtain high-level attributes to be used for classifying images. The performance of the autoencoder model proposed in the study was evaluated with the ADNI data set. ADNI dataset is an open source dataset developed for the diagnosis of Alzheimer's disease via MRI images. The proposed autoencoder model consists of encoder and decoder parts. While the encoder part of the model consists of 6 convolution layers and 1 pooling layer, the decoder part has 5 reverse convolution layers. They compared the different results obtained by changing the number of neurons in the convolution layers. According to the results obtained, the proposed model can determine the

images of people with Alzheimer's disease with 84% sensitivity rate. Therefore, the classification of the images in this way by obtaining important attributes with the convolutional autoencoder models produced promising results in terms of diagnosis.

Indraswari et al. proposed an advanced deep learning model in their studies to perform the segmentation of 3D images (Indraswari et al., 2019). Three different data sets were used to evaluate the proposed model. The first two of the data sets were used to perform brain tumor segmentation, and the third data set was used to perform dental segmentation over the jaw images of individuals. A convolutional autoencoder architecture consisting of encoder and decoder parts has been proposed to perform segmentation. First of all, images are divided into axial, coronal and sagittal components and 2D image component is obtained for each plane. Important attributes were obtained by applying convolution and deconvolution operations to the image components. Important features were obtained by applying convolution and deconvolution operations to the image components. Attributes extracted from axial, coronal, and sagittal slices were combined after the final pooling and size reduction phase to obtain the main input vector to be used in the model. By applying the convolution process to the combined input information, the attributes to be used in the classification of the images are extracted from the data. A special cost function is also proposed for this model used in the study. The proposed cost function adds a certain weight to the classes to take into account the probability of all classes in the classification. At the end of the study, despite the insufficient data sets, they managed to achieve good performance results thanks to the model they proposed and the cost function they used.

Shahzadi et al. proposed a deep learning model that performs the type detection of tumors in the brain via MRI images (Shahzadi et al., 2018). In the study, brain MRI images from individuals with glioma, a common type of brain tumor, were used. Glioma type tumors were classified as high grade (HG) and low grade (LG). In the model proposed for this purpose, deep convolutional neural network and long short-term memory (LSTM) network architectures are used together. In the model, significant features were extracted from the images using the VGG-16 convolutional neural network architecture and then these features were used to train the LSTM network. In this way, using the trained model, glioma cases were classified in volumes by high grade (HG) or low grade (LG). During the training phase, 80% of image data was used to train the proposed model and the remaining 20% was used to test the proposed model. According to the results, the model managed to accurately predict the class of 84% of the test data. In the study, feature extraction was also performed by using AlexNet and ResNet architectures as the

convolutional neural network, but the best result was obtained with VGG-16 architecture with 84% accuracy rate.

Afshar et al. proposed a classification model based on capsule networks architecture with a different deep learning approach to perform brain tumor type detection via MRI images (Afshar et al., 2018). In the study, a data set containing 3064 MRI images belonging to 233 patients with one of three different tumor types was used. It is aimed to reduce the processing time and increase the model performance by rescaling the images in the data set to the size (64 x 64). The scaled images are first processed in the convolution layer containing 56 filters and then transferred to the main capsule layer. In this section, 8 feature maps (24 x 24) are obtained and the obtained features are transferred to the last capsule layer. The last capsule layer contains 3 capsules in size 16 for 3 tumor classes. Different attempts were made by changing the layer and capsule numbers of the capsule network architecture used. According to the best results obtained at the end of the study, the proposed model estimated that 86.56% of the test data. The study is important in terms of proposing a different approach in the field of image classification with deep learning.

Korolev et al. suggested a model using deep learning techniques to perform the detection of Alzheimer's disease and different neurological diseases via MRI images (Korolev et al., 2017). In the study, binary classification of images belonging to 4 different categories was performed by using a part of ADNI data set that is open to public access. In the classification of MR images, two different model approaches were used, the first one is classical convolutional neural network and the other one is modern residual neural network model. Performance comparison of these two models was made. The first model used is a classical convolutional neural network architecture and consists of 21 layers. The second model used is the modern ResNet architecture that won the Imagenet competition in 2015. According to the results obtained at the end of the study, 79% accuracy rate was obtained with 21-layer convolutional neural network model in the classification of individuals with or without Alzheimer's from the categories. In the same task, with the ResNet model, 80% accuracy rate was achieved.

Khobragade et al. proposed a deep learning-based diagnostic model for the detection of different lung diseases (Khobragade et al., 2016). In the study, the classification of tuberculosis, pneumonia and lung cancer diseases, which are three important lung diseases, were dealt with on chest x-ray images. In order to classify the chest X-ray images into 3 different categories, a 4-step model approach has been proposed. In the first stage, the images are subjected to some preprocesses and the high-pass filter reduces the noise in the images. In the second stage, lung

segmentation is performed on the images and the limits of the lungs in the images are determined. For this purpose, density-based edge detection technique is used. In the third stage after lung segmentation, important geometrical features such as region circumference, equivalent diameter, irregularity index, and statistical features such as standard deviation and entropy are extracted for classification by image processing. In the fourth stage, an artificial neural network consisting of 3 hidden layers was trained and tested using the features. In this way, the model was evaluated. According to the results obtained after evaluating the model, the model successfully managed to estimate the class of 92% of the test data.

Varshni et al. proposed a diagnostic model approach based on deep learning and machine learning methods for the automatic determination of pneumonia in the lung over chest x-ray images (Varshni et al., 2019). The data set used in the study was created by making use of a large-scale data set containing 112,120 images. In the data set created, there are 1431 images for people with and without pneumonia. Therefore, a data set consisting of a total of 2862 images selected from 112,120 images was used in the study. In the study, feature extraction was performed with different convolutional neural network architectures such as Xception, VGG-16, ResNet-50, DenseNet-121, DenseNet-169. In the next step, the important features extracted were used to train and test different classification models such as artificial neural network, support vector machines, naive bayes, nearest neighborhood and random forest. Therefore, the study includes a comprehensive performance comparison from different combinations of convolutional neural network architectures and different classifier models. According to the results obtained in the study, the most successful result was obtained by classifying the features extracted with DenseNet-169 convolutional neural network model with the support vector machines classifier.

This study consists of "Introduction", "Materials", "Methods", "Results" and "Conclusions and Suggestions" sections. In the "Introduction" section, the literature review is included and the study is explained. In the "Materials" section, two different image datasets are described, which are used as materials in the study. In the "Methods" section, the deep learning models used in the classification of image data are explained in detail. In the "Results" section, all calculation and classification results obtained from the deep models used are included. In the "Conclusion and Suggestions" section, the results of two different cases were evaluated and the performance comparison of the models was made.

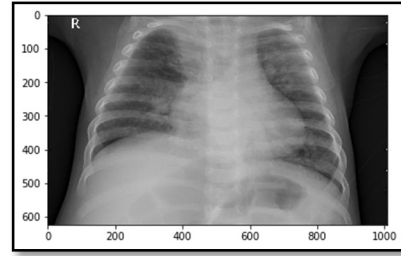
## 2. MATERIALS

In this part of the study, 2 different data sets that will be used as a material for testing the created models will be explained. Therefore, the classification models created in this study were used to diagnose two different diseases through medical images. The first data set used in the study was used to establish the pneumonia disease diagnosis model, and the second data set was used to create the brain tumor diagnosis model.

### a. Pneumonia Data Set

The first data set used in the study was created for the diagnosis of pneumonia on lung x-ray images, and the data set consists of a total of 5840 x-ray images (Kermany and Goldbaum, 2018). This data set consists of x-ray images of one and five-year-old pediatric patients taken from Guangzhou Women's and Children's Medical Center. All X-ray images taken were taken as part of routine clinical care of patients ("Chest X-Ray Images", 2019). In order to perform accurate analysis of chest x-ray images with computer systems, images with low quality or undetectable diagnosis were identified and removed from the data set, thereby ensuring quality control of the images. In order to determine the classification of the received images and to classify the classification models to be used in the classification of the images, the diagnosis of the data set images was first made by two specialist doctors. The data set was also checked by a third specialist to take into account any possible erroneous evaluations of the specialist physicians who evaluated the data.

While 4265 of the images in the data set are lung images of people with pneumonia, the remaining 1575 are images of healthy people without pneumonia. Therefore, the data set contains x-ray images of two different classes. 75% of the data set images were used to train the diagnostic models created and 25% were used to test the models. Therefore, 4380 images were used in the training of models, and 1460 images were used in testing the models. Before the image data was presented as input to the model, some pre-processes were applied to the image data in order to increase the classification performance of the models. The applied pre-processes are described in the following sections. An example image of the data set described is given in Figure 1.

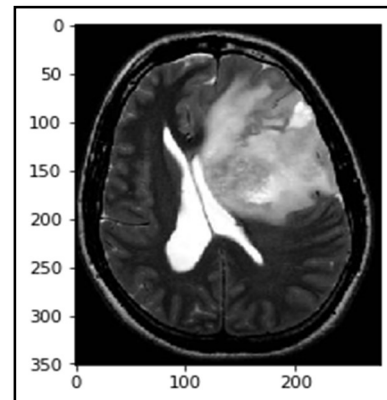


**Figure 1.** A sample chest x-ray image in the data set ("URL-1").

### b. Brain Tumor Data Set

The second data set used in the study was created for brain tumor detection and the data set consists of 253 brain MRI images (Chakrabarty, 2019). 153 of the MRI images included in the data set are images of patients with malignant (malignant) tumors, while 98 of them are images of benign (benign) tumors ("Brain MRI Images", 2019). Therefore, the data set consists of MRI images of two different classes.

189 images, which constitute 75% of the total images and randomly selected, were used to train the diagnostic models created, and the remaining 64 images were used to test the models. Therefore, 25% of the total images were used to test the model. In order for the created models to perform higher, a number of pre-processes such as normalization and resizing were applied to these images. Details of these pre-processes will be discussed in detail in the following sections. An example image of the data set described is shown in Figure 2.



**Figure 2.** A sample brain MRI image in the data ("URL-2").

### **3. METHODS**

In this section, deep learning models used to diagnose two different diseases are explained over two different medical image data sets. In addition, different deep learning techniques applied in models used to increase model performance are also discussed in this section. 3 different model designs were used in the classification of data sets. Classification procedures were carried out using deep learning models with deep convolutional neural network, convolutional neural network with random forest classifier, and convolutional neural network with support vector machine classifier.

#### **a. Deep Convolutional Neural Network Model**

A convolutional neural network model was first used to diagnose two different diseases based on image data. The keras library of the python programming language was used in the design of the convolutional neural network model. In order for the images to be perceived and classified by computer systems, they must first be converted into a digital format that the computer can perceive. For this reason, firstly, image matrices containing the pixel values of the images to be used must be obtained. In the model used, the images of the datasets previously described are resized and image matrices in the size of the images (64x64) are obtained.

After obtaining the image matrices, in addition to these matrices, normalization process is applied. In this way, matrix pixel values are provided to take a value between 0 and 1 (Goodfellow et al., 2016). Thanks to the normalization pre-process, the distorted pixel values in the image matrices are filtered out. In this way, the classification performance of the models used has been increased. When classifying images in convolutional neural networks, it is necessary to obtain classifying features that affect the class of images via image matrices. While the models used determine the class of the test images in the test phase, they determine the class of the images depending on the similarity and difference of these attributes. Therefore, the resulting image matrices are subjected to convolution with filters of a certain size in the convolution layer in order to obtain the features from the images, so that the features to be used in the classification are obtained. In the convolutional neural network model used in the study carried out, the image matrices are subjected to convolution with (5x5) filters.

As a result of the convolution process, the size of the original image matrix is reduced. The reduction of the image matrix size increases the processing speed of the models used and provides an advantage in a temporal sense (LeCun et al., 1989). However, this decrease in size also leads to a certain level of information loss (Aghdam and Heravi, 2017). Padding technique can be used in

such models to prevent loss of information due to the decrease in the size of the image matrix. When this technique is applied, the size of the image matrices presented as an input to the convolution process does not change and as a result of the process, a matrix of the same size is obtained as the output (Krizhevsky et al., 2012). In this way, the problem of any loss of information from the image data as a result of the convolution process is prevented (Qian et al., 2016). In the model used in the study, the same padding technique was used. The number of filters used in the convolution layer can vary depending on the designer and images.

After the convolution process, the image matrices in the convolution layer are included in an activation function. In this study, relu activation function was used at this stage. The main reason why relu activation function is used in the model is the complexity of this function. The use of this activation function performs better in detecting complex nonlinear shapes when convolutional neural networks detect distinctive features in images (Chollet, 2017). After this process, the image matrices are transferred to the pooling layer by the model. In this layer, the number of pixels is reduced by changing the image size. In this way, it is ensured that the performance and speed of the model is increased and the problem that the model called overfitting produces wrong results based on memorization during the learning phase is also solved (Goodfellow et al., 2016). In the model used in this section, the max pooling technique, one of the pooling methods, is used in the pooling layer.

As a result of passing the image matrices through the first convolution layer, the model learns low-level features in the first place through the data. In the model used, the image matrices are passed through convolution and pooling layers two more times. This allows the model to learn mid-level and high-level features in images. The model's learning of more complex high-level features also significantly increases classification performance. In the model created, 30 filters in the first convolution layer and 60 filters in the second and third convolution layers were used. As the layer depth increases, the model can learn more complex features in a better way. After these steps, the attributes of the processed image matrices are transferred to the fully connected layer by the model.

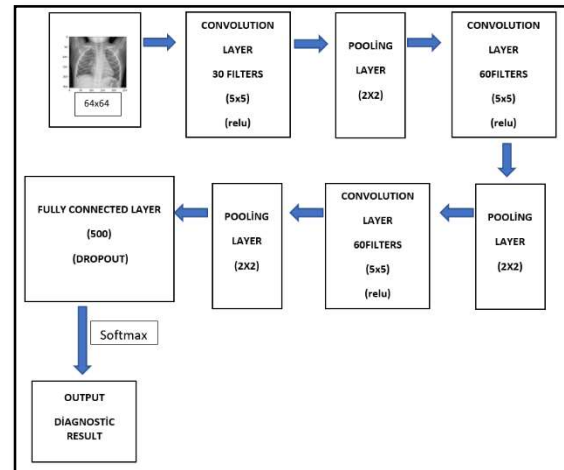
The fully connected layer is the section where the classification of the images is performed depending on the similarity and difference of the attributes. Therefore, the diagnosis of the test images in the data set will be made in this section. The model carries out the classification process and thus the diagnosis of the disease through an artificial neural network in this layer. However, since the artificial neural network in this layer cannot process the image data in matrix format, the image matrix data in question must be given to the artificial neural network in vector format.

Therefore, in this layer, the image matrices are converted into vectors by the flatten operation, and the data of the images are provided as input to the artificial neural network in this layer. Vectors carrying feature data are transferred to an artificial neural network with 500 neurons. Depending on the training data provided as input, the weights of the artificial neural network are cyclically rearranged, and as a result, the artificial neural network in this layer gains the ability to classify through features and thus diagnose diseases. The model trained on the training data is then tested on the image data allocated for the testing phase in the data set, and the classification accuracy percentage, total error and performance of the model in question are tested. Sigmoid function was used as activation function in the artificial neural network at this stage.

In this type of neural networks used in the fully connected layer, as described earlier, the problem of overfitting can be seen. In this problem, the model outputs erroneous classification results based on memorization based on the excessive learning of the data by the model. Therefore, dropout technique was used in this layer as an additional step to prevent this problem. Dropout technique is a regulation method used to train the artificial neural network in this layer (Srivastava et al., 2014). With this technique, half of the neurons are disabled in different iterations with different possibilities during data processing in the artificial neural network. Thus, in each iteration, a certain number of weights are changed instead of all the weights of the artificial neural network in this layer (Chen et al., 2015). Thus, the effect of any image given to the model on the neural network weights is limited. This prevents the network from performing incorrect classifications due to over learning. With the dropout technique applied in the model described in this section, 40% of neurons in the neural network are disabled in every iteration.

Another technique that is applied in order to provide better training on image data and to improve classification performance during the training of the model is the data augmentation technique. The data augmentation technique is applied to the images used to train the model in the data set. In this technique, the images in the image dataset are given to the model both in their original formats and by making some changes (Mikołajczyk et al., 2018). The changes are made by showing the images to the model at different angles and at different distances (Wong et al., 2016). In the model described in this section, a new image with a different structure was obtained from each image used to train the model in the data set by applying the data augmentation technique. Therefore, the number of images used in training the model has been doubled by using this technique. In addition, because the images are produced randomly and differently, the model learns the feature attributes in the images more comprehensively and more

efficiently. Thanks to the re-evaluation of the same images from different angles and different distances, it is provided to identify the features that will give better results in classification. In this way, the model can perform class separation of the data more successfully and accordingly, the classification accuracy percentage of the model increases. The model architecture described in this section is given in Figure 3.



**Figure 3.** CNN architecture described in this section.

### b. Convolutional Neural Network with Random Forest Classifier

Secondly, a hybrid convolutional neural network model was used to diagnose two different diseases based on image data. Keras and scikit-learn libraries belonging to the python programming language were used in the design of the classification model used. In the design of the hybrid model created in this section, the deep convolutional neural network model described in the previous section was used. The low, medium and high-level features of the image data were obtained using the deep convolutional neural network model described in the previous stage. After obtaining these features, random forest algorithm, one of the machine learning techniques, was used to classify these features in the model.

Random forest is a collective classification and regression algorithm that uses decision trees as a weak classifier (Breiman, 2001). Each of the decision trees used as weak learners in the algorithm is trained using a random data set derived from the original data set.

While classifying a sample with this algorithm, the classification of the sample is determined according to the majority decision by taking the decisions of weak classifiers for the sample in question. The sample evaluated according to the votes of weak classifiers is considered to belong to the class with the highest number of votes. Therefore, the most important difference of the

model described in this section from a conventional convolutional neural network is that after the features of the model are obtained, the random forest classifier is used as a different classifier instead of an artificial neural network. While classification operations are carried out with the random forest algorithm, the features that affect the classes the most are found in the tree's root and root nodes in decision trees, which are weak classifiers. Therefore, these features will be more effective when deciding on the class of an image instance. In this way, it will increase the classification performance of the model, since the features that have the most impact on determining classes will be more effective when deciding on the class of an image instance. Therefore, the strength of the model created in terms of performance is that the random forest algorithm is successful at the classification point. The hybrid model architecture described in this section is given in Figure 4.

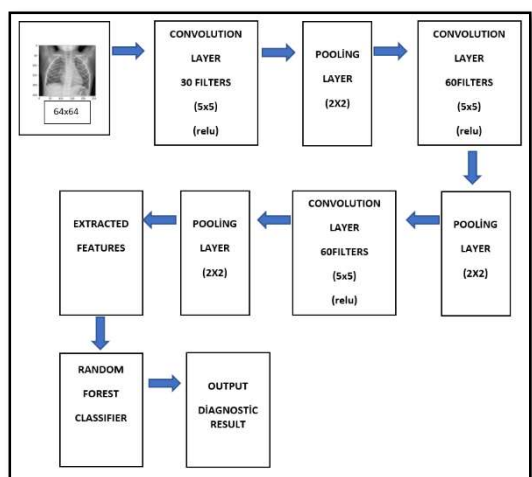


Figure 4. CNN with random forest classifier architecture described in this section.

**c. Convolutional Neural Network with Support Vector Machines**

Thirdly, a hybrid convolutional neural network model similar to the previous section was used to diagnose two different diseases based on image data. In the design of the classification model used, keras and scikit-learn libraries belonging to the python programming language were also used. The deep convolutional neural network model was used in the design of the hybrid model created in this section and the low, medium and high-level attributes of the image data were obtained using this model. Then, support vector machines algorithm, one of the machine learning techniques, was used to classify these attributes in the model.

Support vector machines are basically a classification algorithm that tries to find the plane that separates the classes with the widest margin in the sample space and in the most convenient way (Vapnik and Chapelle, 2000). Considering a linearly separable sample space, there are many plane

solutions, but the support vector machines algorithm calculates the widest margin of these solutions by derivative. Therefore, in the model created in this section, the support vector machines classifier was used to classify the features obtained from the images. The hybrid model architecture described in this section is given in Figure 5.

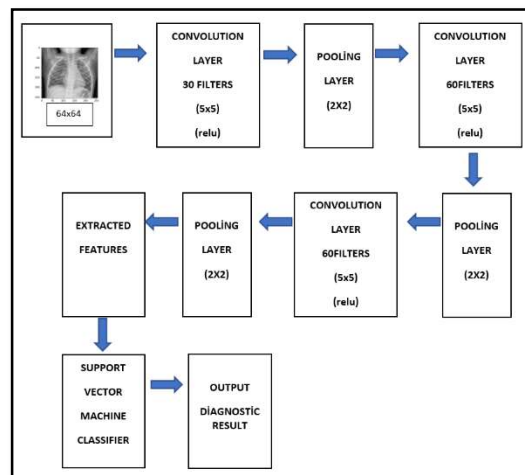


Figure 5. CNN with support vector machines classifier architecture described in this section.

**4. RESULTS**

In this section, the results obtained from three different classification models tested using two different medical image data sets are given and the performance of the models in the classification process is evaluated.

**a. Results from The Deep Convolutional Neural Network Model**

The deep convolutional neural network model was initially trained and tested using the pneumonia disease dataset. The number of images used in training of the model was increased by applying data augmentation technique to the part of the pneumonia disease data set reserved for education. The classification confusion matrix results obtained by classifying the test data of the model are given in Figure 6.

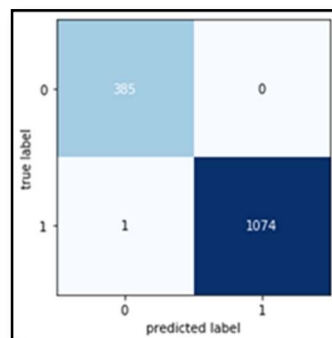


Figure 6. The result of the classification confusion matrix of the first model for the lung x-ray image data set.

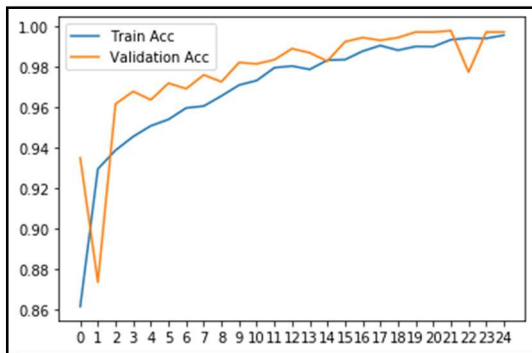


When the classification confusion matrix is analyzed, it is seen that the model made 1 incorrect classification in total. The model incorrectly estimated the class of 1 person with pneumonia and included this person in the class of healthy people. The classification report results of the model are given in Figure 7.

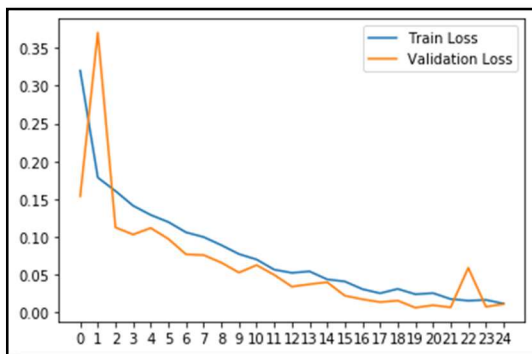
Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	385
1	1.00	1.00	1.00	1075
accuracy			1.00	1460
macro avg	1.00	1.00	1.00	1460
weighted avg	1.00	1.00	1.00	1460

**Figure 7.** The classification report results of the first model for the lung x-ray image data set.

Precision, recall, f1-score and accuracy values obtained from the classification process performed in the classification report results can be seen. The training of the model was carried out in a total of 25 iterations. The classification accuracy and loss graphs obtained during the training of the model are given in Figure 8 and Figure 9.



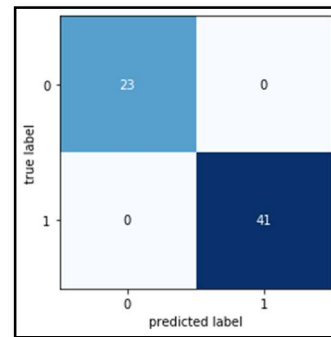
**Figure 8.** The graph of the classification accuracy values obtained by applying the first model for the lung x-ray image data set.



**Figure 9.** The graph of loss values obtained by applying the first model for the lung x-ray image data set.

According to the results, the deep convolutional neural network model, trained using pneumonia disease data, managed to accurately estimate the classes of 1460 test data by 99.8% in the post-training test process. The model has managed to classify 1459 of these images correctly, and it has incorrectly estimated the class of 1 test image.

The deep convolutional neural network model was evaluated in the second stage using the second dataset for brain tumor type detection. The number of images used in training the model was increased by applying data augmentation technique to the part of the brain tumor data set reserved for education, similar to the first data set. The classification confusion matrix results obtained by classifying the test data of the model are given in Figure 10.



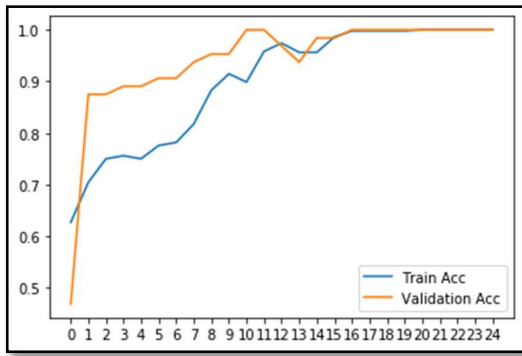
**Figure 10.** The result of the classification confusion matrix of the first model for the brain MRI image data set.

When the classification complexity matrix is analyzed, it is seen that the model correctly predicts the class of all samples and does not make any incorrect classification. The classification report results of the model are given in Figure 11.

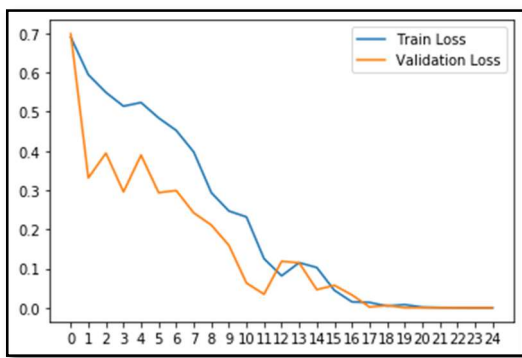
Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	41
accuracy			1.00	64
macro avg	1.00	1.00	1.00	64
weighted avg	1.00	1.00	1.00	64

**Figure 11.** The classification report results of the first model for the the brain MRI image data set.

Precision, recall, f1-score and accuracy values obtained from the classification process performed in the classification report results can be seen. The training of the model was carried out in a total of 25 iterations. The classification accuracy and loss graphs obtained during the training of the model are given in Figures 12 and Figure 13.



**Figure 12.** The graph of the classification accuracy values obtained by applying the first model for the brain MRI image data set.

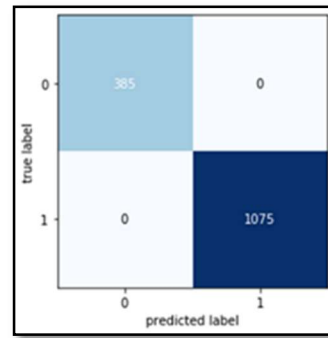


**Figure 13.** The graph of loss values obtained by applying the first model for the brain MRI image data set.

According to the results obtained, the deep convolutional neural network model, which was trained using brain MRI image data, succeeded to predict the classes of 64 test data 100% correctly in the post-training test process.

**b. Results from The Convolutional Neural Network with Random Forest Classifier Model**

The convolutional neural network model with a random forest classifier was initially trained and tested using the pneumonia disease dataset. The classification confusion matrix obtained by classifying the test data of the model is given in Figure 14.



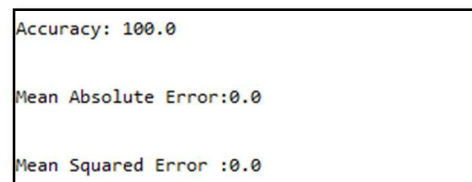
**Figure 14.** The result of the classification confusion matrix of the second model for the lung x-ray image data set.

When the classification confusion matrix result is examined, it is seen that the model correctly predicts the class of all the samples and does not make any incorrect classification. The classification report results of the model are given in Figure 15.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	385
1	1.00	1.00	1.00	1075
accuracy			1.00	1460
macro avg	1.00	1.00	1.00	1460
weighted avg	1.00	1.00	1.00	1460

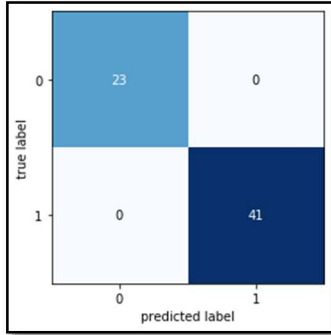
**Figure 15.** The classification report results of the second model for the the lung x-ray image data set.

Precision, recall, f1-score and accuracy values obtained from the classification process performed in the classification report results can be seen. According to the results obtained, the convolutional neural network model with random forest classifier, trained with the obtained features, managed to accurately predict the class of all 1460 test data at the test stage. Therefore, the model was able to achieve 100% accuracy in the testing phase. The calculated classification accuracy rate and total error value results of the model are given in Figure 16.



**Figure 16.** The classification accuracy rate and total error value result of the second model for the lung x-ray image data set.

The convolutional neural network model with a random forest classifier was evaluated in the second stage using the second dataset for brain tumor type detection. The classification confusion matrix results obtained by classifying the test data of the model are given in Figure 17.



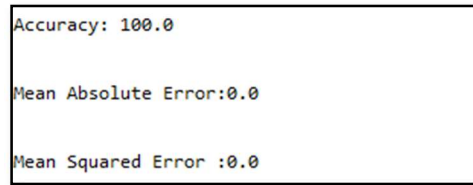
**Figure 17.** The result of the classification confusion matrix of the second model for the brain MRI image data set.

When the classification confusion matrix is analyzed, it is seen that the model correctly predicts the class of all samples and does not make any incorrect classification. The classification report results of the model are given in Figure 18.

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	41
accuracy			1.00	64
macro avg	1.00	1.00	1.00	64
weighted avg	1.00	1.00	1.00	64

**Figure 18.** The classification report results of the second model for the brain MRI image data set.

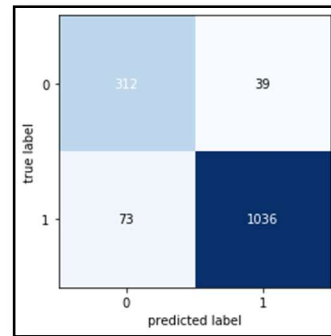
This model has been used in the deep convolutional neural network model explained in the first part in obtaining the features used in classification. The training of the deep convolutional neural network was carried out in 25 iterations. Then the attributes learned by the trained model were classified with the random forest classifier. It can be seen in the results that the model tested with the test data has reached 100% accuracy at this stage. The calculated classification accuracy rate and total error value results of the model are given in Figure 19.



**Figure 19.** The classification accuracy rate and total error value result of the second model for the brain MRI image data set.

**c. Results from The Convolutional Neural Network with Support Vector Machines Model**

The convolutional neural network model with a support vector machine classifier was initially trained and tested using the pneumonia disease dataset. The number of images used in training of the model was increased by applying data augmentation technique to the part of the pneumonia disease data set reserved for education. The classification confusion matrix results obtained by classifying the test data of the model are given in Figure 20.



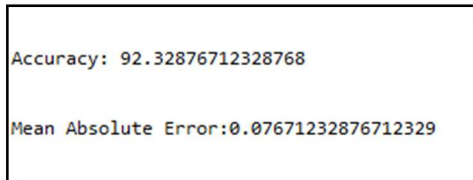
**Figure 20.** The result of the classification confusion matrix of the third model for the lung x-ray image data set.

When the classification confusion matrix result is analyzed, it is seen that the model correctly predicted the class of 1348 samples as well as 112 specimens incorrectly. The model estimated that 39 people who were actually patient were healthy and 73 people who were actually healthy were patients. The classification report results of the model are given in Figure 21.

Classification Report:				
	precision	recall	f1-score	support
0	0.81	0.89	0.85	351
1	0.96	0.93	0.95	1109
accuracy			0.92	1460
macro avg	0.89	0.91	0.90	1460
weighted avg	0.93	0.92	0.92	1460

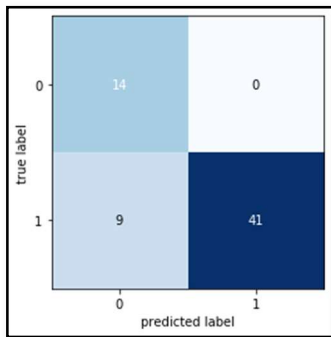
**Figure 21.** The classification report results of the third model for the lung x-ray image data set.

Precision, recall, f1-score and accuracy values obtained from the classification process performed in the classification report results can be seen. According to the results obtained, the convolutional neural network model with support vector machine classifier succeeded in obtaining 92.3% accuracy rate by succeeding in correctly predicting the class of 1348 test data in the test phase. The calculated classification accuracy rate and total error value results of the model are given in Figure 22.



**Figure 22.** The classification accuracy rate and total error value result of the third model for the lung x-ray image data set.

The convolutional neural network model with a support vector machine classifier was evaluated in the second stage using the second dataset for brain tumor type detection. The classification confusion matrix results obtained by the classification of the test data of the model are given in Figure 23.



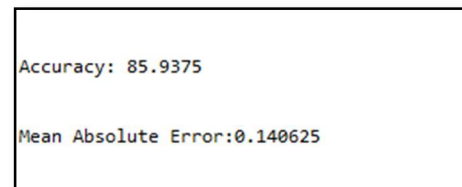
**Figure 23.** The result of the classification confusion matrix of the third model for the brain MRI image data set.

When the classification confusion matrix result is examined, it is seen that the model correctly predicted the class of 55 samples, but also incorrectly predicted the class of 9 samples. The model actually estimated the tumor type of 9 people with benign tumors incorrectly and identified the tumor type of these people as malignant. The classification report results of the model are given in Figure 24.

Classification Report:				
	precision	recall	f1-score	support
0	0.61	1.00	0.76	14
1	1.00	0.82	0.90	50
accuracy			0.86	64
macro avg	0.80	0.91	0.83	64
weighted avg	0.91	0.86	0.87	64

**Figure 24.** The classification report results of the third model for the brain MRI image data set.

The deep convolutional neural network model described in the first part was used to obtain the features used in this classification. Then the features learned by the trained model were classified with the support vector machine classifier. It can be seen from the results that the model tested with the test data reached 85.9% accuracy at this stage. The calculated mean absolute error value of the model was determined as 0.14. The calculated classification accuracy rate and total error value results of the model are given in Figure 25.



**Figure 25.** The classification accuracy rate and total error value result of the third model for the brain MRI image data set.

## 5. CONCLUSIONS AND SUGGESTIONS

When all model results are compared, it is seen that the most successful performance results for both data sets are obtained with the convolutional neural network with random forest classifier. The classification accuracy rates obtained for different data sets from the models are given in Table 1.

**Table 1.** Classification accuracy rates of all models.

Models	Accuracy rate for first dataset	Accuracy rate for second dataset
First Model (CNN)	%99,8	%100
Second Model (CNN+RF)	%100	%100
Third Model (CNN+SVM)	%92,3	%85,9

Since both data sets used in the study contain 2 classes, the models succeeded in classifying images with very high accuracy rates. Of the models used, the convolutional neural network model with a

random forest classifier has succeeded in accurately predicting all classes of test samples for both data sets. However, it is estimated that model performances may decrease below these levels if the number of classes increases. For example, although the hybrid model using the random forest classifier reached 100% accuracy in the study, it is estimated that this rate may decrease if the amount of data used increases. In addition, it is thought that the convolutional neural network with support vector machine classifier can exhibit higher performance in different types of problems with more class. As a result, it can be said that the hybrid model approaches used in the study give successful performance results in binary image classification problems.

Convolutional neural network-based models, which are used in the classification of medical images, require a high amount of data during the training phase. Even if the architectural classification used is good enough to perform, models cannot yield good and reliable results if there is not enough data. Therefore, in future studies to be carried out after this study, studies will be made to increase the samples in small size datasets with deep learning architectures such as generative adversarial networks or convolutional autoencoder. By using these methods, the number of data can be increased to make the study more reliable. Thus, evaluation of similar classification models can be studied by using sufficient and larger data for classification. The step of increasing data by producing synthetic image data, which is planned to be carried out in future studies, is expected to significantly increase the reliability of deep learning models.

## REFERENCES

- Aghdam, H. H. and Heravi, E. J. (2017). *Guide to Convolutional Neural Networks*, NY: Springer, New York, USA.
- Afshar, P., Mohammadi, A. and Plataniotis, K. N. (2018). "Brain tumor type classification via capsule networks." Proc., 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, pp. 3129-3133. IEEE.
- Bejnordi, B. E., Lin, J., Glass, B., Mullooly, M., Gierach, G. L., Sherman, M. E., Karssemeijer, N., van der Laak, J. and Beck, A. H. (2017). "Deep learning-based assessment of tumor-associated stroma for diagnosing breast cancer in histopathology images." Proc., IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, Australia, pp. 929-932. IEEE.
- Breiman, Leo. (2001). "Random forests." *Machine Learning*, Vol. 45, No. 1, pp. 5-32.
- Chakrabarty, N. (2019). *Brain MRI Images for Brain Tumor Detection*.
- Chen, X., Xu, Y., Wong, D. W. K., Wong, T. Y., and Liu, J. (2015). "Glaucoma detection based on deep convolutional neural network." Proc., 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, pp. 715-718.
- Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions." Proc., IEEE Conference on Computer Vision and Pattern Recognition, Hawai'i Convention Center, Honolulu, Hawaii, USA, pp. 1251-1258.
- Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification." *Neurocomputing*, Vol. 321, 321-331.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*, MIT press, Massachusetts, USA.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016). "Deep residual learning for image recognition." Proc., IEEE Conference On Computer Vision And Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778. IEEE.
- Indraswari, R., Kurita, T., Arifin, A. Z., Suciati, N. and Astuti, E. R. (2019). "Multi-projection deep learning network for segmentation of 3D medical images." *Pattern Recognition Letters*, Vol. 125, pp. 791-797.
- Kermany, D. K. and Goldbaum, M. (2018). *Labeled optical coherence tomography (OCT) and Chest X-Ray images for classification*. Mendeley Data, 2.
- Khobragade, S., Tiwari, A., Patil, C. Y. and Narke, V. (2016). "Automatic detection of major lung diseases using Chest Radiographs and classification by feed-forward artificial neural network." Proc., IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, pp. 1-5. IEEE.
- Korolev, S., Safiullin, A., Belyaev, M. and Dodonova, Y. (2017). "Residual and plain convolutional neural networks for 3D brain MRI classification." Proc., IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, Australia, pp. 835-838. IEEE.

- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks." Proc., Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, USA, pp. 1097-1105.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). "Backpropagation applied to handwritten zip code recognition." Neural computation, Vol. 1, No. 4, pp. 541-551.
- Martinez-Murcia, F. J., Ortiz, A., Gorriz, J. M., Ramirez, J. and Castillo-Barnes, D. (2019). "Studying the Manifold Structure of Alzheimer's Disease: A Deep Learning Approach Using Convolutional Autoencoders." IEEE Journal of Biomedical and Health Informatics.
- Mikołajczyk, A., and Grochowski, M. (2018). "Data augmentation for improving deep learning in image classification problem." Proc., International Interdisciplinary PhD Workshop (IIPhDW), Szczecin, Poland, pp. 117-122.
- Oh, S. L., Hagiwara, Y., Raghavendra, U., Yuvaraj, R., Arunkumar, N., Murugappan, M., and Acharya, U. R. (2018). "A deep learning approach for Parkinson's disease diagnosis from EEG signals." Neural Computing and Applications, 1-7.
- Qian, Y., Bi, M., Tan, T., and Yu, K. (2016). "Very deep convolutional neural networks for noise robust speech recognition." IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 24, No. 12, pp. 2263-2276.
- Shahzadi, I., Tang, T. B., Meriadeau, F. and Quyyum, A. (2018). "CNN-LSTM: Cascaded framework for brain Tumour classification." Proc., IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Sarawak, Malaysia, pp. 633-637. IEEE.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958.
- URL-1: <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection/metadata> [Date of Accessed: 09.09.2019]
- URL-2: <https://www.kaggle.com/paultimothymooney/chestxray-pneumonia> [Date of Accessed: 20.07.2019].
- Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). "Understanding data augmentation for classification: when to warp?" Proc., International conference on digital image computing: techniques and applications (DICTA), IEEE, Canberra, Australia, pp.1-6.
- Vapnik, V., and Chappelle, O. (2000). "Bounds on error expectation for support vector machines." Neural Computation, Vol. 12, No. 9, pp. 2013-2036.
- Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R. and Mittal, A. (2019). "Pneumonia detection using cnn based feature extraction." Proc., IEEE Third International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, Tamil Nadu, India, pp. 1-7. IEEE.
- Zhang, J., Xie, Y., Wu, Q. and Xia, Y. (2019). "Medical image classification using synergic deep learning." *Medical Image Analysis*, Vol. 54, pp. 10-19