



## DDL: A new deep learning based approach for multiple house numbers detection and clustering

Murat Taşyürek<sup>1\*</sup> , Celal Öztürk<sup>2</sup> 

<sup>1</sup>Department of Computer Engineering, Faculty of Engineering Architecture and Design, Kayseri University, Kayseri, 38280, Turkey

<sup>2</sup>Department of Computer Engineering, Faculty of Engineering, Erciyes University, Kayseri, 38039, Turkey

### Highlights:

- A new deep learning based approach (DDL) was proposed for multiple house numbers detection and clustering.
- The performance of the proposed DDL approach was compared with the performance of the existing deep learning (DL) method.
- Experimental evaluations conducted on real data showed that DDL approach outperformed DL method.

### Keywords:

- Convolutional neural network
- Deep learning
- Multiple house numbers detection and clustering

### Article Info:

Research Article  
Received: 02.04.2021  
Accepted: 21.08.2021

### DOI:

10.17341/gazimmfd.908332

### Acknowledgement:

The authors thank the Kayseri Metropolitan Municipality

### Correspondence:

Author: Murat Taşyürek  
e-mail: murattasyurek@kayseri.edu.tr  
phone: +90 352 504 3838 / 21351

### Graphical/Tabular Abstract

Character recognition in natural images is a very difficult problem due to many factors such as variability of light, background clutter, severe blur, inconsistent resolution and different scale depth. In addition to these features, distortions in characters and numbers are encountered in street view photographs with the effect of natural events. Detecting and reading house numbers from street views is a computer vision problem that falls under the category of natural scene text recognition. Convolutional neural network (CNN) model is one of the most commonly used deep learning (DL) methods in image analysis. In this study, firstly, CNN based DL method was applied to read characters from pictures that contain house numbers in their natural image. However, successful results could not be obtained, especially in cases where there are more than one house number in the image or when the depths are very variable. A new approach DDL (deep in deep learning) using two different CNN models was proposed to increase the accuracy of the DL method and also to reduce the data size created by natural images.

	DL	DDL
Detect Multi House Number	⊗	✓
Image Compression	⊗	✓
Size	113 gigabyte	17 gigabyte
Worst prediction rate	62%	93%
Best prediction rate	96%	100%

Figure A. Comparisons of approaches

**Purpose:** The purpose of this study is to develop an effective DL approach to detect multiple house numbers from natural images with clustering and decrease the memory space occupied by natural images.

### Theory and Methods:

The DL method cannot produce successful results in cases where there are multiple door numbers and the depths are very variable. In order to overcome the problem caused by the depths of the door number plate in the proposed DDL method, it is ensured that the door number plate is determined first. Due to the very variable depths, the dimensions of the detected plates differ from each other, but in fact the plates are actually the same size. Since the dimensions of the boards installed in the field are the same and to increase the door number accuracy, if the size of the plate is smaller than the input size of the image given for the CNN architecture (416x 416 is used in this study), the plate is resized. After this step, similar to DL architecture, the door number in the plate picture is estimated with the CNN-based DL method.

### Results:

The performance of the DDL approach was compared with the performance of the DL approach using real data consisting of 17,618 images with 113 GB size consisting of building street images with GPS location information taken from 35 neighborhoods of Kayseri Metropolitan Municipality (KBB) Yeşilhisar district for 2019. Experimental evaluations showed that proposed DDL approach can handle multiple house number from images with decreasing size of images and is stable, while the DL cannot handle them and is not stable.

### Conclusion:

Experimental results showed that the proposed DDL approach produced more accurate results and used less storage space than DL approach.



## DDL: Çoklu kapı numarası tespit etme ve kümeleme için derin öğrenme tabanlı yeni bir yaklaşım

Murat Taşyürek<sup>1\*</sup>, Celal Öztürk<sup>2</sup>

<sup>1</sup>Kayseri Üniversitesi, Mühendislik Mimarlık ve Tasarım Fakültesi, Bilgisayar Mühendisliği Bölümü, 38280, Talas Kayseri, Türkiye

<sup>2</sup>Erciyes Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 38039, Melikgazi Kayseri, Türkiye

### Ö N E Ç I K A N L A R

- Çoklu kapı numarası tespit etme ve kümeleme için yeni bir derin öğrenme yaklaşımı, DDL, önerilmiştir
- Önerilen DDL yaklaşımının performansı DL yaklaşımının performansı ile karşılaştırılmıştır
- Gerçek veriler üzerinde yapılan deneysel çalışmalarda DDL yaklaşımın daha iyi performans sergilemiştir

### Makale Bilgileri

Araştırma Makalesi

Geliş: 02.04.2021

Kabul: 21.08.2021

DOI:

10.17341/gazimmfd.908332

### Anahtar Kelimeler:

Evrişimsel sinir ağı,  
derin öğrenme,  
çoklu kapı numarası tespit  
etme ve kümeleme,  
YOLO

### ÖZ

Işığın değişkenliği, arka plan karmaşası, şiddetli bulanıklık, tutarsız çözünürlük ve farklı ölçekli derinlik gibi birçok faktörden dolayı doğal görüntülerde karakter tanıma oldukça zor problemdir. Bu özelliklerin yanı sıra sokak görünüm fotoğraflarında doğa olaylarının da etkisiyle karakterlerde ve sayılarda bozulmalara rastlanır. Sokak görüntülerinden kapı numaralarını tespit etmek ve okumak, doğal sahne metni tanıma kategorisine giren bir bilgisayar görme problemdir. Evrişimsel sinir ağı (convolutional neural network - CNN) modeli görüntü analizlerinde en sık kullanılan derin öğrenme (deep learning - DL) yöntemlerinden biridir. Bu çalışmada, ilk olarak doğal görüntülerinde kapı numarası bulunan görüntülerden karakter okumak için CNN temelli DL yöntemi uygulanmıştır. Ancak, özellikle görüntüde birden fazla kapı numarasının olduğu veya derinliklerin çok değişken olduğu durumlarda yeterince başarılı sonuçlar elde edilememiştir. DL yönteminin doğruluğunu artırmak aynı zamanda doğal görüntülerin oluşturduğu veri boyutunu azaltmak için farklı iki adet CNN modeli kullanan yeni bir yaklaşım DDL (deep in deep learning) önerilmiştir. Önerilen DDL yaklaşımının performansı, Kayseri Büyükşehir Belediyesi'ne (KBB) ait GPS konum bilgisiyile fotoğrafı çekilen 35 adet mahallenin bina görüntülerinden oluşan 113 GB (gigabayt) boyuta sahip 17.618 adet görüntü içeren gerçek veriler kullanılarak, DL yaklaşımının performansı ile karşılaştırılmıştır. Deneysel sonuçlar, önerilen DDL yaklaşımının DL yaklaşımına göre daha doğru sonuçlar ürettiğini ve daha az depolama alanı kullandığını göstermektedir.

## DDL: A new deep learning based approach for multiple house numbers detection and clustering

### H I G H L I G H T S

- A new deep learning approach, DDL, was proposed for multiple house numbers detection and clustering
- The performance of the proposed DDL approach was compared with the performance of the DL approach
- DDL approach outperformed DL approach in the experimental evaluations conducted on real data

### Article Info

Research Article

Received: 02.04.2021

Accepted: 21.08.2021

DOI:

10.17341/gazimmfd.908332

### Keywords:

Convolutional neural  
network,  
deep learning, multiple  
house numbers detection and  
clustering,  
YOLO

### ABSTRACT

Character recognition in natural images is a very difficult problem due to many factors such as variability of light, background clutter, severe blur, inconsistent resolution and different scale depth. In addition to these features, distortions in characters and numbers are encountered in street view photographs with the effect of natural events. Detecting and reading house numbers from street views is a computer vision problem that falls under the category of natural scene text recognition. Convolutional neural network (CNN) model is one of the most commonly used deep learning (DL) methods in image analysis. In this study, firstly, CNN based DL method was applied to read characters from pictures that contain house numbers in their natural image. However, successful results could not be obtained, especially in cases where there are more than one house number in the image or when the depths are very variable. A new approach DDL (deep in deep learning) using two different CNN models was proposed to increase the accuracy of the DL method and also to reduce the data size created by natural images. The performance of the proposed DDL approach was compared with the performance of the DL approach using real data consisting of 17,618 images with 113 GB (gigabyte) size consisting of building images with GPS location information taken from 35 neighborhoods of Kayseri Metropolitan Municipality (KBB). Experimental results showed that the proposed DDL approach produced more accurate results and used less storage space than DL approach.

\*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : \*murattasyurek@kayseri.edu.tr, celal@erciyes.edu.tr /

Tel: +90 352 504 3838 / 21351

844

## 1. GİRİŞ (INTRODUCTION)

Mekânsal verileri depolamak, analiz etmek ve görüntülemek için geliştirilen coğrafi bilgi sistemlerinin (CBS) kalitesi içerdiği verilerin doğruluğuna bağlıdır. Harita ve adres verilerini oluştururken kullanılan kapı numaraları görüntüleri, bu görüntülerdeki numaraların konumları ve içerdiği karakterlerin okunurluğu çok önemlidir [1]. Kapı numaralarının konum bilgisi içeren doğal görüntülerden tespit edilmesi [2] ve konumları ile işlenmesi adres altyapısını hızlandırmaktadır [3]. Kapı numaralarının görüntülerden elde edilmesinde kullanılan en yaygın kaynak 360° ile çekilen koordinatlı panoramik görüntülerden oluşan Google Sokak görüntüleridir [4]. Sokak görünümünden kapı numaralarını tespit etmek ve okumak [5], doğal sahne metni tanıma [6] kategorisine giren bir bilgisayar görme problemi [7]. Işığın değişkenliği, arka plan karmaşası, şiddetli bulanıklık, tutarsız çözünürlük vb. birçok faktörden dolayı doğal sahnelerdeki görüntülerde karakter tanıma oldukça zor problemidir. Bu özelliklerin yanı sıra sokak görünüm fotoğraflarında doğa olaylarının da etkisiyle karakterler ve sayılarda bozulmalar olmaktadır.

Son yıllarda, derin öğrenme (deep learning - DL) yöntemi görüntü sınıflandırma, nesne izleme, poz tahmini, metin algılama ve tanıma, görsel belirginlik algılama, eylem tanıma ve sahne etiketlemede yaygın olarak kullanılmaktadır [8]. Derin sinir ağları, derin inanç ağları, tekrarlayan sinir ağları ve evrişimli sinir ağları derin öğrenmede sıkça kullanılan yöntemlerdir [9]. Bu yöntemlerden evrişimli sinir ağlarının (convolutional neural network - CNN) görüntü sınıflandırmasında yüksek performans gösterdiği tespit edilmiştir [10]. CNN modeli adını evrişim adı verilen matrisler arasındaki matematiksel doğrusal işlemden alır [11]. CNN modeli evrişimli katman, doğrusal olmayan katman, havuz katmanı ve tam bağlantılı katmanı da içeren çoklu katman yapısından oluşur [12]. Doğal görüntülerden karakter ve/veya numara tespit edilmesi bilgisayar görme kategorisinde sınıflandırma problemlerindedir. Literatürde, görüntü sınıflandırmada çok yüksek performans gösteren CNN modeli ile oluşturulan DL yöntemi ile sokak görüntüsünden kapı numarasının tespit edilmesi konusunda çalışmalara rastlanmaktadır [2, 4]. Bu çalışmada, ilk olarak doğal görüntülerinde kapı numarası bulunan levhalardan karakter okumak için CNN temelli DL yöntemi uygulanmıştır. Ancak, özellikle görüntüde birden fazla kapı numarasının olduğu veya derinliklerin çok değişken olduğu durumlarda yeterince başarılı sonuçlar elde edilememiştir. DL yönteminin doğruluğunu artırmak aynı zamanda doğal görüntülerin oluşturduğu veri boyutunu azaltmak için farklı iki adet CNN modeli kullanan yeni bir yaklaşım DDL (deep in deep learning) önerilmiştir. Önerilen DDL yaklaşımının performansı, Kayseri Büyükşehir Belediyesi Yeşilhisar ilçesinin 2019 yılına ait GPS konum bilgisiyle fotoğrafı çekilen 35 adet mahallenin bina sokak görüntülerinden oluşan 113 GB (gigabayt) boyuta sahip 17.618 adet görüntü içeren gerçek veriler kullanılarak, DL yaklaşımının performansı ile karşılaştırılmıştır.

### 1.1. Katkılar (Contributions)

- Bu çalışmanın temel katkıları aşağıdaki gibi özetlenebilir:
- Çoklu kapı numarası tespit etme ve kümeleme için derin öğrenme tabanlı yeni bir yaklaşım (DDL) önerilmiştir.
  - Önerilen DDL yaklaşımının performansı derin öğrenme (DL) yaklaşımının performansı ile karşılaştırılmıştır.
  - Gerçek veriler üzerinde yapılan deneysel çalışmalarda DDL yaklaşımının daha iyi performans sergilediği gözlemlenmiştir.
  - Önerilen yaklaşım ile sonuçlar elde edilirken görüntülerin boyutları yaklaşık 7 kat küçültülerek kullanılan hafıza alanı azaltılmıştır.

Bu makalenin bundan sonraki bölümlerinde sırası ile literatür araştırması sunulmuş, CNN modeli ile oluşturulan DL yöntemi ile ilgili temel tanımlar verilmiş, önerilen iki farklı CNN modeli kullanan DDL yaklaşımı anlatılmış, deneysel değerlendirmeler sunulmuş, daha sonra sonuçlar tartışılmış ve gelecekteki planlanan çalışmalardan bahsedilmiştir.

## 2. LİTERATÜR ARAŞTIRMASI (LITERATURE RESEARCH)

Derin öğrenme yöntemlerinden birisi olan CNN yöntemi [13] son yıllarda bilgisayar ağları [14], görüntü algılama [15] ve hastalık sınıflandırma [16] gibi farklı alanlarda yaygın olarak kullanılmaktadır. Özcan vd. [17] CNN yöntemi ve hiper parametre optimizasyonu ile Türkçe işaret dilinin tanınması üzerine çalışmalar yapmıştır. Sonmez vd. [18] CNN yöntemi ile Türkçe dili için otomatik görüntü altyazısı oluşturan bir model üzerine çalışmalar yapmıştır. Kagaya vd. [19] gıda görüntülerinin tespit edilmesi ve tanınması için CNN yöntemini kullanmıştır. Hansen vd. [20] CNN yönteminin insan yüzü tanıma başarısından yola çıkarak çiftlikte hayvan yüzü tanıma için CNN yöntemini kullanmıştır. Pramerdorfer vd. [21] görüntü tabanlı yüz ifadesi tanıma için CNN yöntemini kullanmıştır. Gerke vd. [22] futbolcuların forma numaralarını tanıma için CNN yöntemini kullanmıştır. Lorencin vd. [23] deniz nesnelerini tanıma için CNN yöntemini kullanmıştır. Liv vd. [24] akciğer görüntü yamalarını interstisyel akciğer hastalığı ile sınıflandırmak için CNN yönteminden faydalanmıştır. Alwzawy vd. [25] el yazısı ile yazılan yazılarda bulunan rakamları tanıma için CNN yöntemini kullanmıştır.

Anguelov vd. [4] sokak görüntülerinden kapı numarasını tespit etmek için CNN modeli ile oluşturulan LeNet-5 yöntemini kullanmıştır. LeCun vd. [26] tarafından geliştirilen rakamları sınıflandıran 7 seviyeli evrişimli bir ağ olan LeNet-5, 32x32 piksel gri tonlamalı girdi görüntülerinde elle yazılmış sayıları tanıma için uygulanmıştır [27]. Ancak, LeNet-5 yöntemi ile daha yüksek çözünürlüklü görüntüler analiz edilmek istendiğinde evrişimsel ağı seviyesi yetersiz olmaktadır [28]. Diğer taraftan Goodfellow vd. [2] sokak görüntülerinden çoklu kapı numarasının tespiti için CNN modeli ile oluşturulan

ImageNet [29] kütüphanesini kullanmış ve başarılı sonuçlar elde etmiştir. Daha yüksek doğruluk ve sonuçları daha hızlı elde etmek için kullanılan CNN modelleri sürekli geliştirilmiştir [30]. Örneğin, 2012 yılında geliştirilen AlexNet daha önceki CNN yöntemlerinden daha iyi sonuçlar göstermiş [29], VGGNet 2014 yılında AlexNet'e benzer şekilde 3x3 evrişim ağından ve 16 katmandan oluşan mimarisi ile Simonyan vd. [31] tarafından geliştirilmiş, 2015 yılında 152 katmanlı yapay sinir ağını eğitebilen ResNet mimarisi He vd. tarafından önerilmiştir [32]. CNN modeli tek nesne içeren görüntü analizlerinde özellik çıkarma ve sınıflandırma açısından başarılı olmakta fakat çoklu nesne içeren görüntü analizlerinde yeterince başarılı olamamıştır. Girshick vd. CNN modelinde çoklu obje sorununu aşmak ve görüntüyü yaklaşık olarak 2000 bölgeye ayırıp bölge içerisinde CNN ile arama yapmak için R-CNN yöntemini önermiştir [33]. R-CNN yönteminin hesaplama maliyeti zamansal açıdan yüksektir. Ross Girshick R-CNN yavaş çalışma sorununu ortadan kaldırmak için daha hızlı çalışan Fast R-CNN yöntemini geliştirmiştir [34]. R-CNN algoritmaları, görüntü içindeki nesneyi yerelleştirmek için bölgeler kullanır. Bölge incelemek yerine nesneyi içermeye olasılığı yüksek olan görüntünün parçalarını inceleyen CNN tabanlı YOLO (You Only Look Once) yöntemi Redmon vd. tarafından geliştirilmiştir [35]. YOLO yöntemi [36], gerçek zamanlı nesne takibinde kullanılan Faster RCNN yöntemi [37] gibi pek çok nesne tespiti yöntemlerine göre daha başarılı sonuç üretmiştir [38, 39]. Bu çalışmada, ilk olarak doğal görüntülerinde kapı numarası bulunan levhalardan karakter okumak için son yıllarda gerçek hayat problemlerine çok fazla uyarlanan YOLO algoritması temelli CNN yöntemi uygulanmıştır. Ancak, bu yöntem ile yapılan çalışmalarda özellikle görüntüde birden fazla kapı numarasının olduğu veya derinliklerin çok değişken olduğu durumlarda yeterince başarılı sonuçlar elde edilememiştir. Doğruluğu artırmak aynı zamanda doğal görüntülerin oluşturduğu veri boyutunu azaltmak için YOLO algoritması temelli iki adet CNN yönteminden oluşan DDL olarak adlandırdığımız yeni bir yaklaşım önerilmektedir.

### 3. CNN TABANLI DERİN ÖĞRENME YÖNTEMİ İLE KAPI NUMARASININ TESPİT EDİLMESİ (DETECTION OF HOUSE NUMBER WITH CNN BASED DEEP LEARNING METHOD)

Bu çalışmada, CNN tabanlı derin öğrenme (DL) yöntemi olarak YOLO kullanılmıştır. Bu bölümde öncelikle YOLO yöntemi sunulmuş daha sonra ise DL yöntemi ile kapı numarası tespit etme sistemi tanıtılmıştır.

#### 3.1. YOLO (You Only Look Once)

YOLO yaklaşımı ismini "You Only Look Once" olarak ifade edilen ve yalnızca bir kez bakarsınız manasına gelen kelimelerden almaktadır [35]. YOLO yaklaşımı görüntüde bulunan nesnelerin ne olduğunu ve nerede olduğunu bir bakışta tahmin edebilmektedir [36]. Nesne tespiti, özellikle belirli bir görüntünün içinde hangi nesnelerin olduğunu ve ayrıca görüntünün neresinde olduklarını keşfetme kategorisine giren bir bilgisayar görme problemidir. Nesne

tespiti, nesnenin görüntünün içerisinde nerede olduğunu da bulduğundan dolayı görüntülerde sınıflandırma problemlerine göre daha kompleks ve zor bir problemidir. Ayrıca, görüntülerde birden fazla nesnenin olduğu durumlarda sınıflandırma yaklaşımları ile başarı elde edilmesi mümkün değildir.

YOLO yöntemi ile çoğu zaman yüksek doğruluk elde edilebilir, aynı zamanda gerçek zamanlı olarak çalışmaktadır ki bu yeteneklerinden dolayı son yıllarda sıklıkla tercih edilmektedir [37]. Algoritma, tahmin yapmak için sinir ağından yalnızca bir ileri yayılma geçişi gerektirmesi anlamında görüntüye "yalnızca bir kez bakar". Maksimum olmayan bastırmadan sonra (nesne algılama algoritmasının her nesneyi yalnızca bir kez algılamasını sağlar), ardından tanıyan nesnelere sınırlayıcı kutularla birlikte çıkarır. YOLO ile, tek bir CNN aynı anda birden fazla sınırlayıcı kutuyu ve bu kutular için sınıf olasılıklarını tahmin eder. YOLO, tam görüntüler üzerinde çalışma ve algılama performansını doğrudan optimize etme yeteneklerine sahiptir. Bu modelin diğer nesne algılama yöntemlerine avantajları şu şekilde sıralanabilir:

- Oldukça hızlıdır.
- Eğitim ve test süresi boyunca görüntünün tamamını görebilmesi sayesinde sınıflar ve görünüşleriyle ilgili bağlamsal bilgileri örtük olarak kodlayabilir.
- Nesnelerin genelleştirilebilir temsillerini öğrenebilir, böylece doğal görüntüler üzerinde eğitildiğinde ve sanat eserleri üzerinde test edildiğinde, iyi algılama performansı sergilemektedir.

YOLO yönteminin nasıl çalıştığı kapı numarası örneği üzerinde incelenecek olursa, öncelikle Şekil 1'de gösterildiği gibi bir girdi görüntüsü YOLO yöntemine sunulmaktadır. Şekilde görüldüğü üzere kapı numarası 2 ve 9 rakamlarından oluşmaktadır ve bu numaralar kırmızı çerçeve ile işaretlenmiştir.



Şekil 1 Giriş Görüntüsü (Input Image)

YOLO kütüphanesi giriş görüntüsünü ızgaralara bölmektedir. Şekil 2’de 3x3 ızgara örneği gösterilmiştir. YOLO uygulamalarında daha fazla sayıda ızgara kullanılmaktadır fakat Şekil 2’de örneğin daha net anlaşılması için 3x3 boyutunda bir yapı kullanılmıştır. Her bir ızgarada görüntüye sınıflandırma ve konumlandırma uygulanmaktadır. Bu adımdan sonra YOLO yöntemi eğer ızgara içerisinde aranan nesne var ise bu nesnelere karşılık gelen sınıf olasılıklarını ve nesnelere çevreleyen kutuları tahmin eder.



Şekil 2 3x3 ızgara (3x3 grid)

Görüntü 3x3 boyutunda bir ızgaraya bölündüğünde ve sadece rakam nesnelere sınıflandırılmak istendiğinde

0,1,2,3,4,5,6,7,8 ve 9 olmak üzere toplamda 10 sınıf olacaktır. Her bir ızgaranın oluşturacağı çıktı ( $y$ ) 15 boyutlu bir vektör olacaktır. ızgara çıktısı  $y$  Eş. 1’de gösterilmiştir.

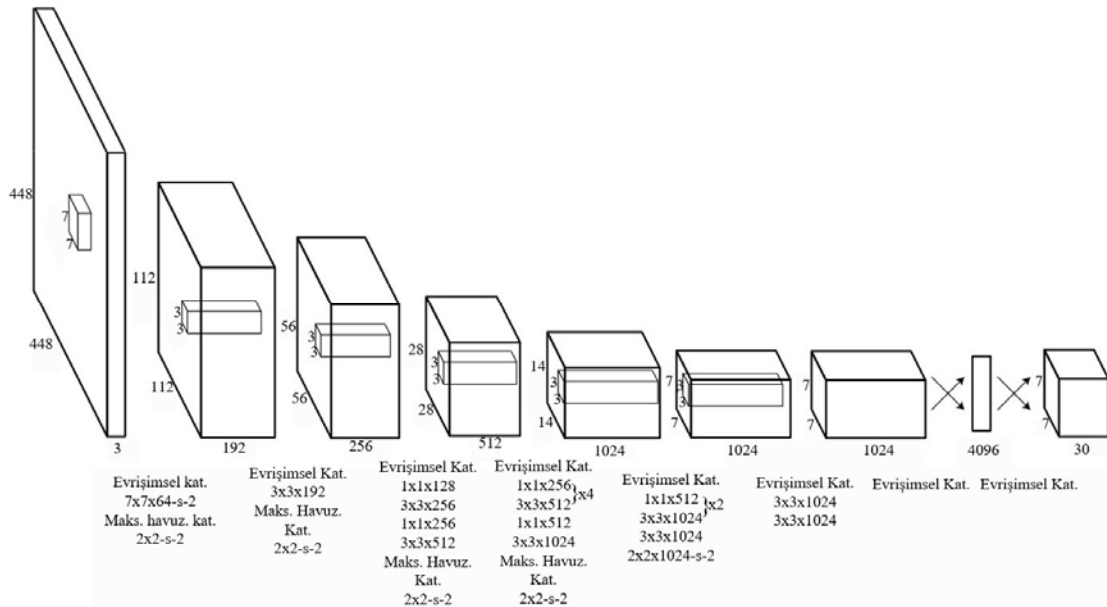
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_0 \\ c_1 \\ \vdots \\ c_9 \end{bmatrix} \quad (1)$$

Eş. 1’de bulunan  $p_c$  ızgarada bir nesnenin bulunup bulunmadığını göstermektedir.  $b_x$ , nesnenin başlangıç noktasının  $x$  eksenindeki konumunu,  $b_y$  nesnenin başlangıç noktasının  $y$  eksenindeki konumunu,  $b_h$  nesnenin yüksekliğini,  $b_w$  nesnenin genişliğini ve  $c_0, c_1, \dots, c_9$  sınıfları göstermektedir. ızgarada bulunan nesne hangi sınıfa ait ise ait olduğu sınıfın değeri 1 diğer sınıfların değeri sıfır olacaktır. YOLO algoritması Şekil 2’de bulunan 9 adet ızgaranın her biri için Eş. 1’de gösterildiği gibi bir vektör çıktısı oluşturacaktır.

YOLO yöntemi, eğitim kümesinde bulunan her bir görüntüyü Şekil 2’de gösterildiği gibi 3x3 ( $S \times S$ ) ızgaralara bölmektedir. Bir nesne birden fazla ızgaranın sınırları içerisinde bulunsa bile yalnızca orta noktasının bulunduğu tek bir ızgaraya atanacaktır. Her bir ızgara sınırlayıcı kutu ve bunların güven skorunu oluşturmaktadır.  $G_s$  güven skorunu göstermek üzere Eş. 2 ile hesaplanır.

$$G_s = p_c * IOU_{pred}^{truth} \quad (2)$$

Güven skoru ızgaranın ne kadar doğru tahmin ettiğini göstermektedir. ızgara içerisinde bir nesne var ise  $p_c$  değeri



Şekil 3 YOLO CNN Mimarisi (YOLO CNN Architecture)

1 diğer durumlarda 0 olmaktadır.  $IOU_{pred}^{truth}$  parametresi, tahmin edilen sınırlayıcı kutu ile olması gereken sınırlayıcı kutu arasındaki oranı göstermektedir [5d] [39]. Test sırasında, her bir ızgara tarafından tahmin edilen sınıf bilgileri ve sınırlayıcı kutu tarafından tahmin edilen güven bilgileri, her sınırlayıcı kutunun sınıfa özgü güven puanını elde etmek için Eş. 3 ile hesaplanır.

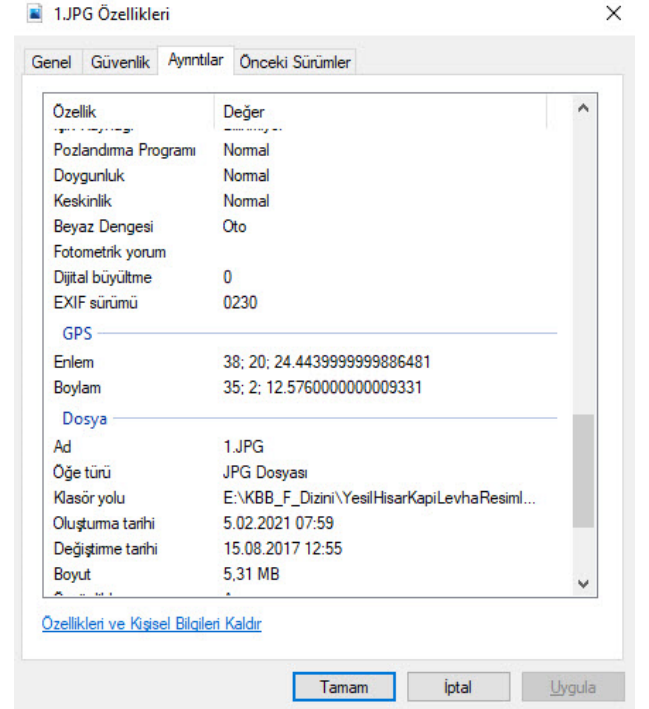
$$Pr(Class_i|Object) * p_c * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth} \quad (3)$$

Eş. 3'te bulunan  $Pr(Class_i|Object)$  terimi ızgara tarafından tahmin edilen kategori bilgisidir. Bu işlem, tahmin edilen kutunun belirli bir kategoriye ait olma olasılığını kodlar ve ayrıca kutunun doğruluğu hakkında bilgi içerir. Her kutu için sınıfa özgü güven puanını hesaplandıktan sonra, eşik değeri belirlenerek düşük puanlı kutular filtrelenir ve çakışan kutulardan doğru olanı seçmek için NMS (Non Maximum Suppression) tekniği kullanılır [41]. YOLO algoritması bu işlemleri CNN modeli kullanarak gerçekleştirmektedir. Redmon vd. [35] tarafından geliştirilen YOLO mimarisi Şekil 3'te gösterilmiştir. Şekil 3'te gösterilen mimari 24 adet evrişimsel katman ve ardından tamamen bağlı 2 katmandan oluşmaktadır. Alternatif 1x1 evrişimsel katmanlar önceki katmanlardan özellik alanı azalmaktadır. Mimari S=7 (7x7) ızgara yapısını kullanmaktadır. Giriş verisi olarak 448x448 boyutlarında görüntü almaktadır. 20 adet kategori bulunmaktadır. Mimari 7x7x30 boyutunda çıktı oluşturmaktadır. YOLO yaklaşımı sürekli geliştirilmiştir. Redmon vd. [35] tarafından 2016 yılında geliştirilen ilk sürümü olan YOLO V1 mimarisi çıktı katmanı tamamen bağlantılı bir katman olduğundan, test sırasında YOLO eğitim modeli yalnızca eğitim görüntüsü ile aynı giriş çözünürlüğünü desteklemektedir. YOLO V1 sürümünün eksiklerini ortadan kaldırmak ve başarısını devam ettirmek için 2017 yılında Redmon vd. [42] tarafından daha doğru, daha hızlı ve daha güçlü olan 9.000 adet nesne tanıyabilen YOLO v2 mimarisi tanıtılmıştır. 2018 yılında Redmon vd. [43] tarafından geliştirilen YOLO V3 modeli bir önceki modele göre daha karmaşıktır. YOLO V3 mimarisi modelin yapısının boyutunun değiştirilmesine izin vererek modelin hızının ve doğruluğunun değiştirilmesine olanak vermektedir. 2020 yılında Bochkovskiy vd. [44] tarafından optimum hız ve doğruluk ile nesne tanıma yöntemi olarak YOLO V4 sürümü tanıtılmıştır. YOLO V4 sürümünde etkili ve güçlü bir nesne tespit modeli önerilmektedir. YOLO V4 sürümünde giriş ağı çözünürlüğü, evrişimsel katman sayısı, parametre sayısı ve katman çıktılarının (filtreler) sayısı arasındaki en iyi dengeyi bulmak amaçlanmıştır. Deneysel değerlendirmelerde YOLO - V1, V2 ve V3 sürümlerine göre daha üstün olduğu gözlemlenen YOLO - V4 sürümü bu çalışmada kullanılmıştır.

### 3.2. DL Yöntemi ile Kapı Numarası Tespiti (Door Number Detection with DL Method)

Uzaysal verileri depolamak, analiz etmek ve görüntülemek için geliştirilen Coğrafi Bilgi Sistemlerinin kalitesi içerdiği verilerin doğruluğuna bağlıdır. Harita ve adres verilerini

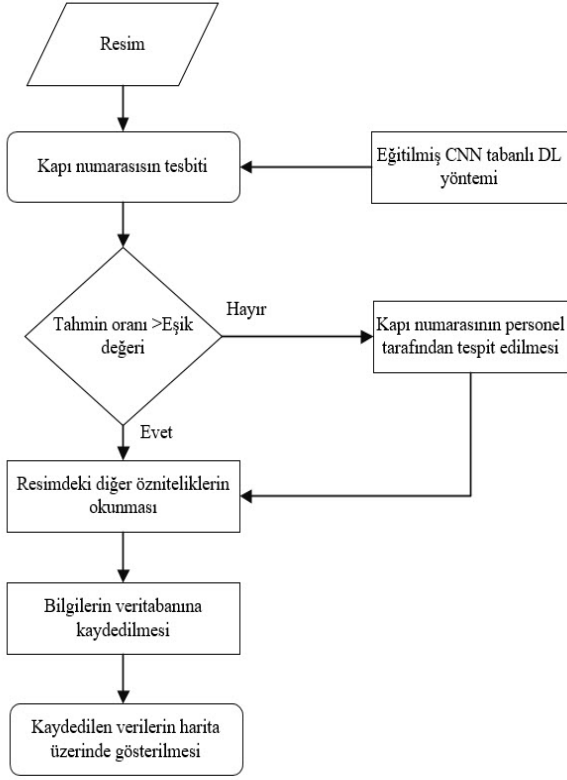
oluştururken kullanılan kapı numaraları görüntüleri, bu görüntülerdeki numaraların konumları ve içerdiği karakterlerin okunurluğu çok önemlidir [1]. Kapı numaralarının konum bilgisi içeren doğal görüntülerden tespit edilmesi [2] ve konumları ile işlenmesi adres altyapısını hızlandırmaktadır [3]. Bu çalışmada, ilk olarak doğal görüntülerinde kapı numarası bulunan levhalardan karakter okumak için CNN temelli DL yöntemi önerilmiştir.



Şekil 4 Fotoğraf Konum Bilgileri (Photo Location Information)

KBB ile yapılan görüşmelerde ve incelemelerde fotoğraf makinesi ile sahada koordinat bilgilerini içerecek şekilde çekilen görüntülerin daha sonra coğrafi bilgi sistemlerine (CBS) kaydedildiği tespit edilmiştir. KBB tarafından kullanılan CBS yazılımın çekilen fotoğrafın özellikler penceresinin ayrıntılı sekmesinde bulunan GPS başlığı altında yer alan enlem ve boylam bilgilerini kullanarak kapı numarasının konumunu tespit ettiği görülmüştür. Fakat kullanılan CBS yazılımında personeller kapı numaralarını fotoğrafa bakarak manuel olarak girmektedir. Kapı numaraları manuel olarak girildiğinden dolayı göz yorulması, klavyede yanlış tuşa dokunulması, farklı kişilerin kapı numarası girmesinden dolayı ortak bir standart oluşturulamaması (sıklıkla “/” yerine “.” veya “-” girilmesi) vb. gibi durumlarda kapı numarası sisteme yanlış girilebilmekte veya aynı numaralar farklı formatta yazılabilmektedir. Bu durumda CBS sisteminin en önemli bileşenlerinden olan adres harita alt yapısının yanlış veya düzensiz oluşmasına sebep olmaktadır. Bu çalışmada bu sorunların üstesinden gelmek için CNN tabanlı derin öğrenme (DL) yöntemi önerilmiş ve önerilen DL yönteminin sistem mimarisi Şekil 5'te gösterilmiştir. Şekil 5'de öncelikle CNN tabanlı derin öğrenme yönteminin kapı numarası eğitim verisi kullanarak eğitilmesi gerekmektedir.

Eğitim işlemi tamamlandıktan sonra önerilen sistem kapı numarası tespit amaçlı kullanılabilir. Kapı numarası tespiti için girdi olarak konum bilgileri ve görüntü içerisinde kapı numarasını içeren bir resim sunulmalıdır.



**Şekil 5** DL Yöntemi ile Kapı Numarası Tespiti  
(Door Number Detection with DL Method)

Resim sisteme sunulduktan sonra CNN tabanlı DL yöntemi ile resimde bulunan kapı numarası tahmin edilmektedir. DL yöntemi ile tahmin edilen kapı numarasının güven skoru belirlenen eşik değerinin üzerinde ise sistem tahmin edilen kapı numarası ve resimde bulunan konum bilgisini okuyarak bu bilgileri veri tabanına kaydetmekte ve daha sonra ise harita ortamında kaydedilen verilerin görüntülenmesini sağlamaktadır. DL yöntemi tarafından tahmin edilen kapı numarasının güven skoru eşik değerinin altında ise bu durumda sistem kapı numarasının kullanıcı tarafından girilmesini sağlamakta ve daha sonra resimden diğer özellik bilgilerinin okuyarak verilerin veri tabanına kaydedilmesi ve harita ortamında gösterilmesini sağlamaktadır.

DL yöntemi test aşamasında bazı görüntülerde çok başarılı sonuçlar elde etmiştir. Ancak, özellikle görüntüde birden fazla kapı numarasının olduğu veya derinliklerin çok değişken olduğu durumlarda yeterince başarılı sonuçlar elde edilememiştir. DL yönteminin doğruluğunu artırmak aynı zamanda doğal görüntülerin oluşturduğu veri boyutunu azaltmak için farklı iki adet CNN modeli kullanan yeni bir yaklaşım DDL (deep in deep learning) önerilmiştir.

#### 4. DDL: ÇOKLU KAPI NUMARASI TESPİT ETME VE KÜMELEME İÇİN ÖNERİLEN DERİN ÖĞRENME TABANLI YAKLAŞIM (DDL: APPROACH BASED ON DEEP LEARNING FOR MULTIPLE HOUSE NUMBERS DETECTION AND CLUSTERING)

CNN tabanlı DL yöntemleri [2, 4] sokak görüntülerinden kapı numarasının tespit edilmesinde başarılı sonuçlar elde etmesine rağmen KBB'den alınan gerçek veriler üzerinde test edildiğinde görüntü derinliklerinin çok değişken olduğu ve görüntülerde birden fazla birbirine benzeyen kapı numarası olduğu durumlarda yeterince başarılı sonuçlar elde edilememiştir. Şekil 6'da çoklu kapı numarasının örneği ve Şekil 7'de kapı numarasının bulunduğu derinliğin fazla olduğu resim gösterilmiştir. DL yöntemi Şekil 6'da kapı numarasını tespit ederken bütün kapı numaralarını okumakta ve hepsini bir kapı numarası olarak algılamaktadır fakat gerçekte birbirinden farklı 4 adet kapı numarası bulunmaktadır. Şekil 7 örneğinde ise fotoğraf çok uzak mesafeden çekildiğinden dolayı kapı numarası görüntü büyütülmeden anlaşılmamaktadır.



**Şekil 6** Çoklu Kapı Numarası (Multiple House Numbers)



**Şekil 7** Derinliğin Çok Fazla Olduğu Fotoğraf  
(Photo with extremely shallow field of depth)

DL yöntemi Şekil 6'da olduğu gibi çoklu kapı numarasının olduğu durumlarda ve Şekil 1, Şekil 6 ve Şekil 7 birlikte incelendiğinde derinliklerin çok değişken olduğu durumlarda başarı sonuçlar üretmemektedir. Ayrıca, Şekil 4'te görüldüğü üzere her bir resim hafızada en az 5 MB (mega byte) yer kaplamaktadır ve KBB tarafından kullanılan CBS yazılımında ve DL yönteminde de görüntülerin boyutundan dolayı hafıza karmaşıklığı yüksektir. Kapı numarası tahmin doğruluğunu artırmak amacıyla aynı zamanda doğal görüntülerin oluşturduğu veri boyutunu azaltmak için YOLO algoritması temelli iki adet farklı CNN modelinden oluşan DDL olarak adlandırdığımız yeni bir yaklaşım önerilmiştir. Önerilen sistemin mimarisi Şekil 8'de sunulmuştur. DDL yöntemi Şekil 8'de gösterildiği üzere iki adet birbirinden farklı olarak eğitilmiş CNN tabanlı DL yöntemi içermektedir. Önerilen DDL yönteminde kapı numarası levhasının derinliklerinin oluşturduğu sorunun üstesinden gelmek için öncelikle kapı numarası levhasının tespit edilmesi sağlanmıştır. Bir görüntüde birden fazla kapı numarası levhası bulunabilmektedir. Tespit edilen levhalar ayrı ayrı değerlendirilmektedir. DDL yöntemi tarafından tahmin edilen levhanın güven skoru belirlenen eşik değerinin üzerinde ise levha dikkate alınmaktadır. Derinliklerin çok değişken olmasından dolayı tespit edilen levhaların boyutları birbirinden farklı olmaktadır fakat aslında levhalar gerçekte aynı boyuttadır. Sahada montaj edilen levhaların boyutları aynı olduğundan dolayı ve kapı numarası doğruluğunun artırılması için eğer levhanın boyutu CNN mimarisi için verilen görüntünün giriş boyutundan daha küçük ise (bu çalışmada 416x 416 kullanılmıştır) levha yeniden boyutlandırılmaktadır. Bu adımdan sonra DL mimarisine benzer şekilde, CNN tabanlı DL yöntemi ile levha resminde bulunan kapı numarası tahmin edilmektedir. DL yöntemi ile tahmin edilen kapı numarasının güven skoru belirlenen eşik değerinin üzerinde ise sistem tahmin edilen kapı numarası ve resimde bulunan konum bilgisini okuduktan sonra hafıza karmaşıklığını azaltmak için görüntü sıkıştırılmaktadır. Bu adımdan sonra tespit edilen verileri veri tabanına kaydetmekte ve ardından harita ortamında kaydedilen verilerin görüntülenmesini sağlamaktadır. DL yöntemi tarafından tahmin edilen kapı numarasının güven skoru eşik değerinin altında ise bu durumda sistem kapı numarasının kullanıcı tarafından girilmesini sağlamakta ve daha sonra ise özniteliklerin okunması, görüntünün sıkıştırılması, bilgilerin veritabanına kaydedilmesi ve harita ortamında gösterileme adımları sırası ile yapılmaktadır.

## 5. DENEYSEL ÇALIŞMALAR (EXPERIMENTAL STUDIES)

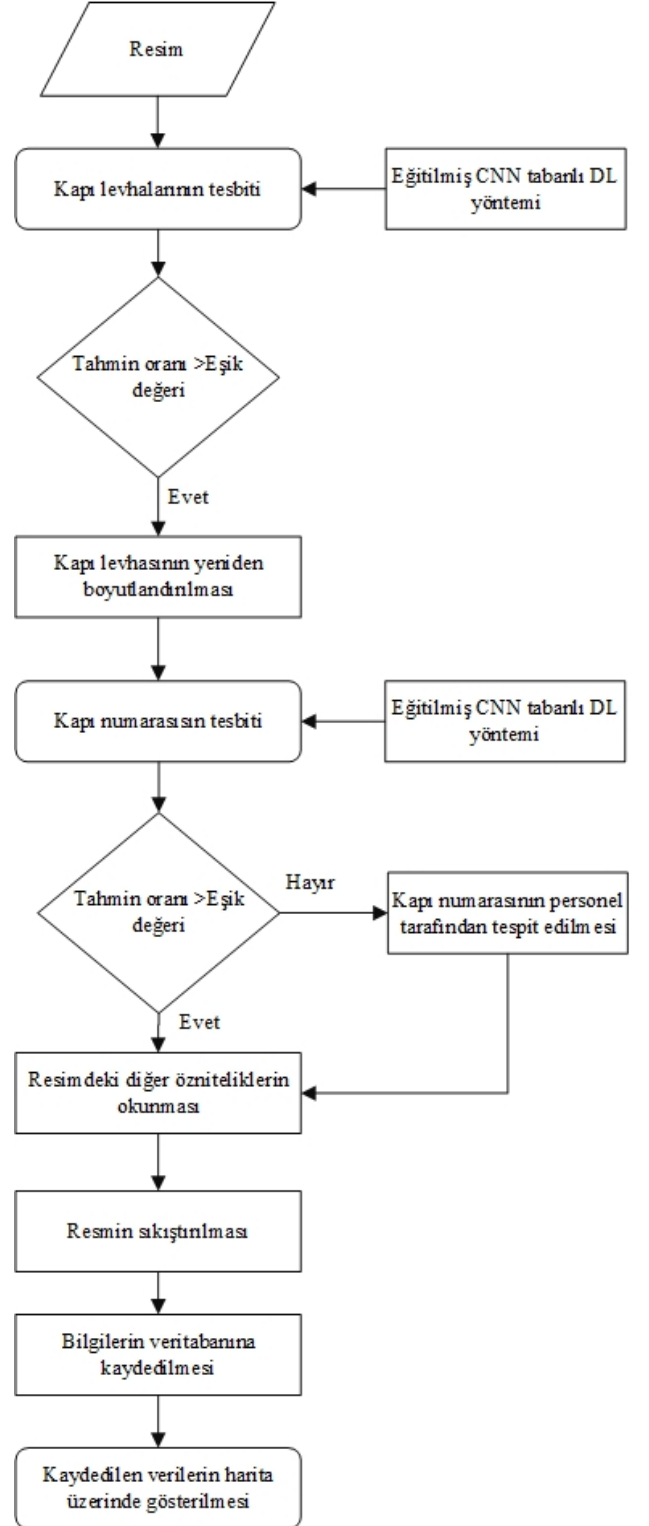
Bu bölümde DL ve DDL yöntemlerinin deneysel performansları karşılaştırılmıştır. Deneysel değerlendirmelerde aşağıdaki soruların cevapları irdelenmiştir.

- Yöntemlerin doğruluğu nedir?
- Yöntemlerin görüntü boyutu nedir?

### 5.1. Veri kümesi (Data Set)

Bu çalışmada KBB'den alınan Yeşilhisar ilçesinin 2019 yılına ait GPS konum bilgisiyle fotoğrafı çekilen 35 adet

mahallenin bina sokak görüntülerinden oluşan 113 GB boyuta sahip 17.618 adet görüntü içeren gerçek veriler kullanılmıştır. Tablo 1'de KBB'den alınan görüntülerin mahalle kodu, mahalle adı ve mahallede bulunan kapı numarası sayısı sunulmuştur.



Şekil 8 DDL Sistem Mimarisi (DDL System Architecture)



**Tablo 1.** Veri Kümesi Özet Bilgileri (Dataset Summary)

Mahalle Kodu	Mahalle Adı	Kapı Numarası Sayısı
181793	AKKÖY	150
181795	ARAPLI	193
181797	BAŞKÖY	424
42361	BEYLERYUKARI	393
181799	BÜGET	201
42362	CAMİKEBİR	291
42363	CUMHURİYET	471
181801	ÇADIRKAYA	475
181803	DERBENTBAŞI	571
181805	DOĞANLI	600
181807	ERDEMLİ	463
42365	FATİH	595
42366	FEVZİOĞLU	433
181809	GÜLBAYIR	1052
42367	GÜNEY	310
181811	GÜZELÖZ	593
181813	İÇMECE	273
42368	İDRİS	417
181815	KALE	182
42369	KARACABEY	1351
181819	KAVAK	479
181821	KAYADİBİ	769
42370	KELER	393
181823	KEŞLİK	261
181825	KOVALI	824
42371	KÖŞK	593
181827	KUŞÇU	566
42372	KUZEY	345
42373	MAHMUDIYE	733
181831	MUSAHACILI	1082
181833	OVAÇİFTLİK	491
181835	SOĞANLI	363
42374	YAVUZ SELİM	552
42375	YENİŞEHİR	300
181837	YEŞİLOVA	429

### 5.2. Model yapılandırması (Model configuration)

KBB'den alınan gerçek veriler YOLO V4 mimarisini kullanarak Google Colab sisteminde eğitilmiştir. Google Colab, Google tarafından geliştiricilere sunulan GPU üzerinde derin öğrenme işlemlerinin gerçekleştirilebildiği bulut sistemidir [45]. KBB'den alınan 17.618 gerçek veriden 1600 tanesi sistemi eğitmek için kullanılmış ve eğitim işleminden sonra sistemin doğruluğu diğer veriler üzerinde test edilmiştir. YOLO V4 mimarisinde 1600 adet verinin %80'i eğitim verisi %20'si ise test verisi olarak kullanılmıştır. Birbirinden farklı iki adet YOLO V4 modeli

eğitilmiştir. İlk olarak doğal resimlerdeki kapı numaralarının tespit edilmesi için "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, A, B, C" sınıflarından oluşan YOLO V4 modeli eğitilmiş ve bu model DL yöntemi olarak adlandırılmıştır. İkinci olarak öncelikle kapı levhalarının tespit edilmesi için sadece levha sınıfından oluşan YOLO V4 modeli eğitilmiştir. Bu yöntemde öncelikle doğal resimlerde bulunan kapı levhası eğitilen YOLO V4 modeli ile bulunmakta, bulunan levhanın boyutu ayarlanarak bu levha içerisinde bulunan kapı numaraları bir önceki eğitilen YOLO V4 modeli tarafından tespit edilmektedir ve bu yaklaşım DDL olarak adlandırılmaktadır.

Eğitim işlemi sonrası oluşan yitim (loss) fonksiyonu değeri ve ortalama hassasiyet (average precision - AP) değerlerinin ortalaması (mean average precision - mAP) eğitim işleminin başarısını göstermektedir [4x]. Eğitim aşamalarında çok fazla veri ile eğitim yapılması ve eğitim modelinin iterasyon sayısı çok fazla artırırsa sistemde aşırı öğrenme (over train) oluşabilmektedir [46]. İki adet YOLO modeli için de 2.000 iterasyon kullanılmış ve eğitim sonucu oluşan yitim ve mAP grafiği Şekil 9 ve Şekil 10'da gösterilmiştir. Şekil 9 kapı numarası için yitim ve mAP grafiğini göstermektedir. Şekil 9'da iterasyon sayısı arttıkça yitim değeri düşmekte mAP değeri artmaktadır ve bu işlem eğitim işleminin başarılı şekilde gerçekleştiğini göstermektedir. 1.800 iterasyondan sonra mAP ve yitim grafiği değişmemektedir ve bu grafikte 2.000 iterasyonun bu veri kümesi için yeterli olduğunu göstermektedir. Şekil 10'da ise Şekil 9'a benzer şekilde grafik eğitimin başarılı olduğunu göstermekte ve bununla birlikte YOLO modelinde tek bir sınıf olduğundan eğitim işlemini daha hızlı gerçekleştirebilmektedir. Deneysel çalışmalar için Intel Core i7-9700 3.0 GHz 16GB RAM özelliklerine sahip kişisel bilgisayar üzerinde Payton 3.7 ve OpenCv kütüphanesi kullanılmıştır.

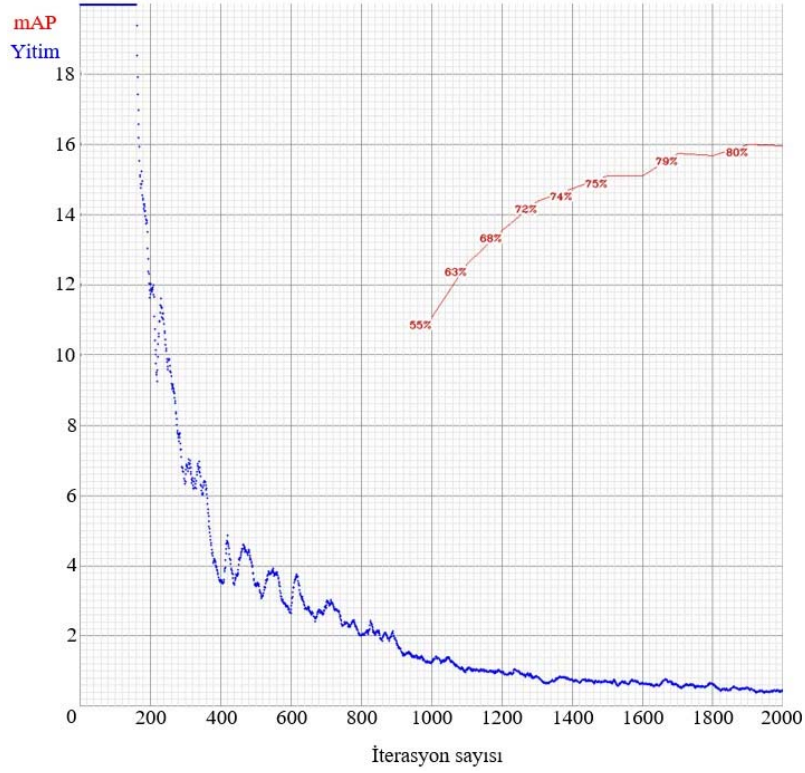
### 5.3. Deneyler (Experiments)

Bu bölümde DL ve önerilen DDL yöntemlerinin deneysel karşılaştırılmaları sunulmuştur. İlk olarak yöntemlerin doğruluğu karşılaştırılmış daha sonra ise görüntü boyutları incelenmiştir.

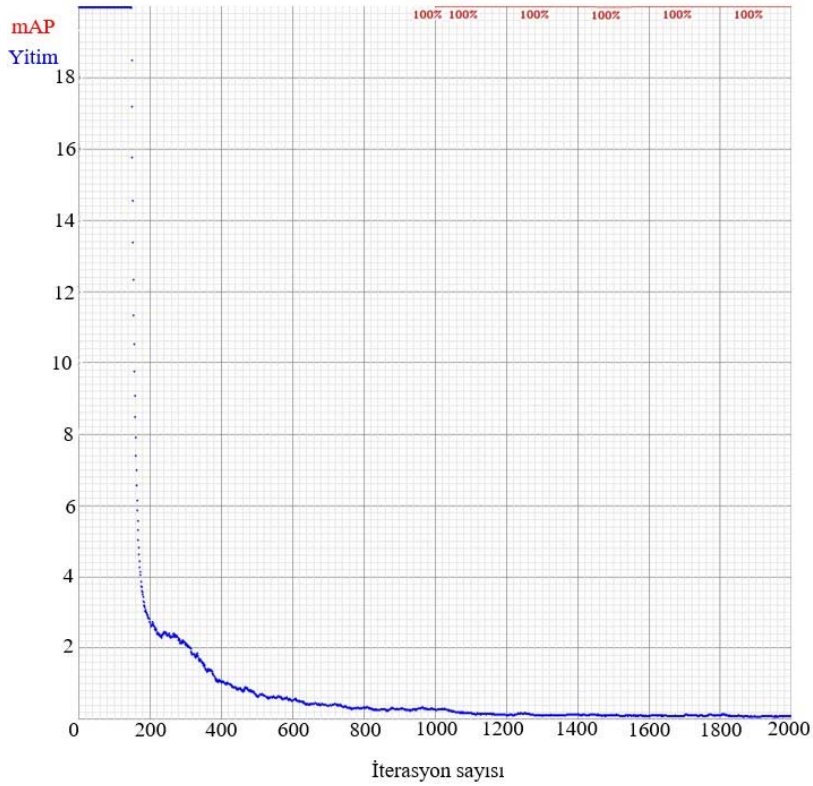
#### 5.3.1. Yöntemlerin doğruluğu (Accuracy of methods)

Yöntemler tarafından tespit edilen kapı numarası değeri ile görüntüler üzerinde bulunan gerçek kapı numarası değerlerinin karşılaştırması yapılmıştır. KBB'den alınan 17.618 adet kapı numarasından sadece toplam kapı numarasının %10'undan daha düşük olan 1.600 adet kapı numarası kullanılarak bütün kapı numaraları için CNN tabanlı DL ve DDL yöntemleri tarafından kapı numarası tahmin edilmiştir. 35 adet mahalle için mahalle bazında yöntemlerin doğruluk oranı doğru tahmin edilen kapı numarası sayısının mahallede bulunan toplam kapı numarasına oranı olarak yüzdelik dilimde Şekil 11'de gösterilmiştir.

DL ve DDL yöntemlerinin doğrulukları arasındaki fark, daha önce Bölüm 4'te bahsedildiği üzere KBB'den alınan gerçek verilerde kapı numarasının bulunduğu görüntü



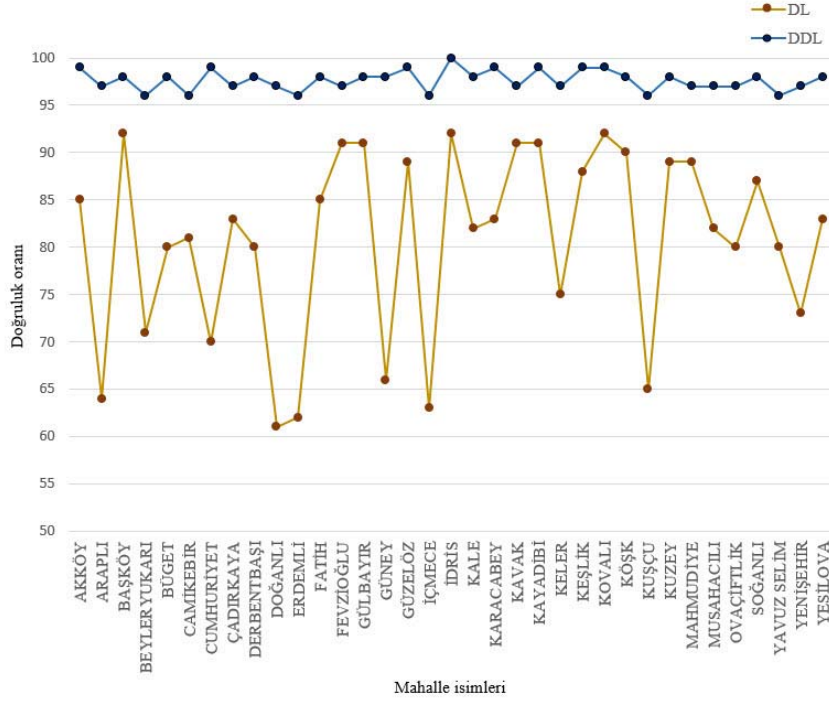
Şekil 9 Kapı Numarası Eğitim Grafiği (Door Number Training Chart)



Şekil 10 Kapı Levhası Eğitim Grafiği (Door Sign Training Chart)

derinliklerinin çok değişken olduğu ve/veya görüntülerde birden fazla birbirine benzeyen kapı numarası olduğu

durmalarından kaynaklanmaktadır. CNN tabanlı DL yöntemi Şekil 6'da gösterilen birden fazla kapı numarasının olduğu



Şekil 11 Yöntemlerin Doğruluğu (Accuracy of Methods)

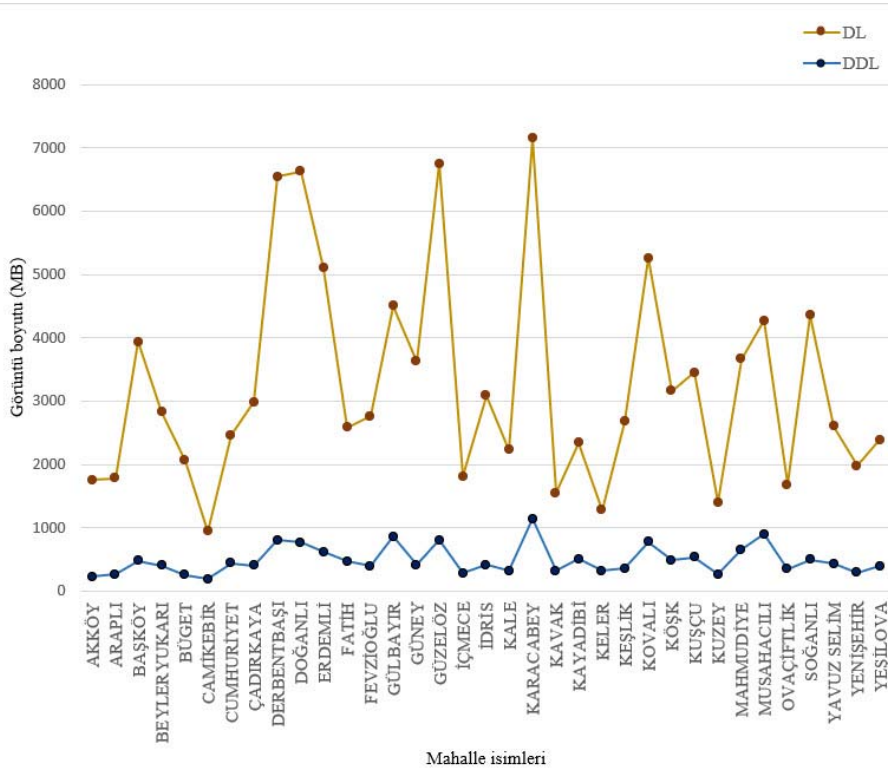
durumlarda görüntüden sadece tek bir kapı numarası tespit edebilmekte ve kapı numaraları arasında sınıflandırma yaparak birbirinden ayıramamaktadır. Ayrıca, DL yöntemi derinliklerin değişken olduğu bazı görüntülerde başarı sonuçlar elde etmesine rağmen bazı görüntülerde yanlış kapı numaraları tespit ettiği gözlemlenmiştir. Özetle, DL yöntemi en kötü durumda %62 en iyi durumda %93 başarı oranı göstererek kararsız bir durum sergilemiştir. DDL yöntemi ise öncelikle CNN tabanlı derin öğrenme yöntemi ile görüntüde bulunan kapı numarası levhalarını tespit etmekte daha sonra ise bu levhayı yeniden boyutlandırarak CNN tabanlı derin öğrenme yöntemi ile levha içerisinde bulunan kapı numaralarını tespit etmektedir. DDL yöntemi kapı numarasını tespit ederken aynı zamanda Şekil 6 örneğinde gösterilen durumlarda her bir kapı numarasını ayrı ayrı değerlendirerek kümeleme işlemi yapmaktadır. DDL yöntemi görüntüde bulunan levhayı tespit edip tespit edilen levhayı yeniden boyutlandırarak Şekil 1, Şekil 6 ve Şekil 7 birlikte incelendiğinde açıkça görülen farklı derinliklerin oluşma probleminin de üstesinden gelmektedir. DDL yöntemi en kötü durumda %96 en iyi durumda da %100 doğruluk oranı ile DL yöntemine göre çok daha kararlı performans sergilemiştir. Şekil 11'den de açıkça görüleceği üzere DDL yöntemi ile DL yöntemine göre daha başarılı ve tutarlı sonuçlar elde edilmiştir.

### 5.3.2. Yöntemlerin görüntü boyutu (Image memory of methods)

Bu deneyde, KBB'den alınan gerçek doğal sahne görüntüleri kullanılarak veri setinde bulunan her mahalle için DL ve DDL yöntemlerinin oluşturmuş olduğu görüntülerin diskte

da kapladığı alanlar incelenmiştir. KBB tarafından doğal sahne görüntüleri sadece kapı numaralarının doğruluğunu teyit etmek ve bina hakkında genel bilgi sunmak amacıyla veritabanında saklanmaktadır. Fakat doğal resimler veritabanında hiçbir işlem yapılmadan tutulduğunda çok fazla yer kaplamaktadır. Önerilen DDL yönteminde görüntü üzerinde ek işlem yapmadan öncelikle birbirinden farklı iki adet CNN tabanlı derin öğrenme yöntemi kullanılarak kapı numarası tespit edilmektedir. Kapı numarası tespit edildikten sonra ise görüntünün boyutunda herhangi bir değişiklik yapmadan açık kaynak kullanılabilen Pillow [47] kütüphanesi Payton ortamında kullanılarak görüntünün sadece kalitesi 100 değerinden 75 değerine düşürülerek kapladığı alan boyutundan kazanım sağlanmaktadır. Orijinal görüntü kalitesinin çok yüksek olmasından ve aynı zamanda bu işlem görüntüde bulunan kapı numarası tespit edildikten sonra yapıldığından dolayı herhangi bir kazanım kaybı söz konusu olmamaktadır. DL ve DDL yöntemlerinin oluşturmuş olduğu görüntülerin diskte kapladıkları alanlar Şekil 12'de gösterilmiştir.

CNN tabanlı DL yönteminde kapı numarası doğru tespit edilemediğinden dolayı görüntü sıkıştırma yapılmamıştır. DL yönteminin görüntü boyutu aynı zamanda KBB tarafından kullanılan veri setinin diskte kapladığı gerçek alanı da göstermektedir. Şekil 12'de gösterildiği üzere DDL yöntemi görüntü sıkıştırma işlemi ile kapı numarasının tespit edilmesinde doğruluktan herhangi bir taviz vermezken toplam hafıza boyutunu 113 GB boyuttan 17 GB boyutuna düşürülmüştür. Önerilen yaklaşım ile sonuçlar elde edilirken görüntülerin boyutları yaklaşık 7 kat küçültülerek kullanılan hafıza alanı azaltılmıştır.



Şekil 12 Yöntemlerin Hafıza Büyüklüğü (Memory Size of Methods)

## 6. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu çalışmada, doğal görüntülerinde kapı numarası bulunan görüntülerden karakter okumak için iki farklı CNN kullanan doğruluk ve hafıza boyutu açısından etkili DDL yöntemi önerilmiştir. Önerilen DDL yönteminin performansı KBB'den alınan gerçek veriler kullanılarak CNN tabanlı DL yönteminin performansı ile karşılaştırılmıştır. Deneysel sonuçlar, DDL yönteminin DL yöntemine göre doğal görüntülerde bulunan kapı numaralarını çok daha doğru tespit ederken aynı zamanda kümeleme yaptığını ve veri saklama boyutunda çok ciddi kazanım elde edildiğini göstermektedir.

DDL tabanlı yönteminin performansı insan eliyle yapılan işlemlere göre çok daha hızlı ve hatasız yapabileceğini göstermesiyle, gelecekte benzer işlemlerde ve süreçlerde önerilen yaklaşımının ve/veya türevlerinin tercih edilmesine yol açacağına inanmaktayız. Önerilen yöntemi farklı lokasyonlarda kullanıma sunmak ve kullanımını yaygınlaştırmanın yanında genel metin transkripsiyonu veya konuşma tanıma gibi farklı problemlere uygulamak ve geliştirmek gelecek çalışma planları arasındadır.

## TEŞEKKÜR (ACKNOWLEDGEMENT)

Kayseri Büyükşehir Belediyesi'ne Yeşilhisar ilçesinin 2019 yılına ait GPS konum bilgisiyle fotoğrafı çekilen 35 adet mahallenin bina görüntülerini paylaştığı için teşekkür ederiz.

## KAYNAKLAR (REFERENCES)

1. Batuk, F., Öztürk, D., Emem, O., Türkiye Ulusal Konumsal Veri Altyapısı İçin Temel Veriler. Jeodezi ve Jeoinformasyon Dergisi, (96), 3-12, 2007.
2. Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V., Multi-digit number recognition from street view imagery using deep convolutional neural networks, arXiv preprint arXiv:1312.6082, 2013.
3. Türk, T., Adres Kayıt Sistemi ile Kent Bilgi Sistemlerinin Bütünleştirilmesi, Jeodezi ve Jeoinformasyon Dergisi, (99), 13-22, 2008.
4. Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., ... Weaver, J., Google street view: Capturing the world at street level, Computer, 43 (6), 32-38, 2010.
5. Bayram, F., Derin öğrenme tabanlı otomatik plaka tanıma, Politeknik Dergisi, 23 (4), 955-960, 2020.
6. Zuo, L. Q., Sun, H. M., Mao, Q. C., Qi, R., Jia, R. S., Natural scene text recognition based on encoder-decoder framework, IEEE Access, 7, 62616-62623, 2019.
7. Aktaş A., Doğan B., Demir Ö., Tactile paving surface detection with deep learning methods, Journal of the Faculty of Engineering and Architecture of Gazi University, 35 (3), 1685-1700, 2020.
8. Perez, L., Wang, J., The effectiveness of data augmentation in image classification using deep learning, arXiv preprint arXiv:1712.04621, 2017.

9. Guo, T., Dong, J., Li, H., Gao, Y., Simple convolutional neural network on image classification, 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), IEEE, 721-724, 2017.
10. Lu, L., Zheng, Y., Carneiro, G., Yang, L., Deep learning and convolutional neural networks for medical image computing, *Advances in Computer Vision and Pattern Recognition*, 10, 978-3, 2017.
11. Albawi, S., Mohammed, T. A., Al-Zawi, S., Understanding of a convolutional neural network, 2017 International Conference on Engineering and Technology (ICET), IEEE, 1-6, 2017.
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P., A convolutional neural network for modelling sentences, *arXiv preprint arXiv:1404.2188*, 2014.
13. Li, Y. D., Hao, Z. B., Lei, H., Survey of convolutional neural network, *Journal of Computer Applications*, 36(9), 2508-2515, 2016.
14. Vinayakumar, R., Soman, K. P., Poornachandran, P., Applying convolutional neural network for network intrusion detection, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 1222-1228, 2017.
15. Chauhan, R., Ghanshala, K. K., Joshi, R. C., Convolutional neural network (CNN) for image detection and recognition, 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, 278-282, 2018.
16. Kido, S., Hirano, Y., & Hashimoto, N., Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN), In 2018 International workshop on advanced image technology (IWAIT), IEEE, 1-4, 2018.
17. Özcan T., Baştürk A., ERUSLR: A new Turkish sign language dataset and its recognition using hyperparameter optimization aided convolutional neural network, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 3 (1), 527-542, 2020.
18. Sonmez E.B., Yıldız T., Yılmaz B.D., Demir A.E., Image captioning in Turkish language: Database and model, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35 (4), 2089-2100, 2020.
19. Kagaya, H., Aizawa, K., Ogawa, M., Food detection and recognition using convolutional neural network, *Proceedings of the 22nd ACM international conference on Multimedia*, 1085-1088, 2014.
20. Hansen, M. F., Smith, M. L., Smith, L. N., Salter, M. G., Baxter, E. M., Farish, M., Grieve, B., Towards on-farm pig face recognition using convolutional neural networks, *Computers in Industry*, 98, 145-152, 2018.
21. Pramerdorfer, C., & Kampel, M., Facial expression recognition using convolutional neural networks: state of the art, *arXiv preprint arXiv:1612.02903*, 2016.
22. Gerke, S., Muller, K., & Schafer, R., Soccer jersey number recognition using convolutional neural networks, *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 17-24, 2015.
23. Lorencin, I., Anđelić, N., Mrzljak, V., Car, Z., Marine objects recognition using convolutional neural networks, *NAŠE MORE: znanstveni časopis za more i pomorstvo*, 66 (3), 112-119, 2019.
24. Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., Chen, M., Medical image classification with convolutional neural network, 2014 13th international conference on control automation robotics & vision (ICARCV), IEEE, 844-848, 2014.
25. Alwzway, H. A., Albehadili, H. M., Alwan, Y. S., Islam, N. E., Handwritten digit recognition using convolutional neural networks, *International Journal of Innovative Research in Computer and Communication Engineering*, 4 (2), 1101-1106, 2016.
26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86 (11), 2278-2324, 1998.
27. El-Sawy, A., Hazem, E. B., Loey, M., CNN for handwritten arabic digits recognition based on LeNet-5, In *International conference on advanced intelligent systems and informatics*, Springer, Cham, 566-575, 2016.
28. Silaparasetty, V., *Neural Network Collection, Deep Learning Projects Using TensorFlow 2*, Berkeley, CA, 249-347, Apress, 2020.
29. Krizhevsky, A., Sutskever, I., Hinton, G. E., Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 25, 1097-1105, 2012.
30. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... Murphy, K., Speed/accuracy trade-offs for modern convolutional object detectors, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7310-7311, 2017.
31. Simonyan, K., Zisserman, A., Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.
32. He, K., Zhang, X., Ren, S., & Sun, J., Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, 770-778, 2016.
33. Girshick, R., Donahue, J., Darrell, T., Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, 580-587, 2014.
34. Girshick, R., Fast r-cnn, *Proceedings of the IEEE international conference on computer vision*, IEEE, 1440-1448, 2015.
35. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, 779-788, 2016.
36. Huang, R., Pedoeem, J., Chen, C., YOLO-LITE: a real-time object detection algorithm optimized for non-GPU

- computers, 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2503-2510, 2018.
37. Du, J., Understanding of object detection based on CNN family and YOLO, Journal of Physics: Conference Series, IOP Publishing, 1004 (1), 012029, 2018.
  38. Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., Menotti, D., A robust real-time automatic license plate recognition based on the YOLO detector, 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 1-10, 2018.
  39. Tao, J., Wang, H., Zhang, X., Li, X., Yang, H., An object detection system based on YOLO in traffic scene, 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), IEEE, 315-319, 2017.
  40. Adarsh, P., Rathi, P., Kumar, M., YOLO v3-Tiny: Object Detection and Recognition using one stage improved model, 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 687-694, 2020.
  41. Wang, D., Li, C., Wen, S., Han, Q. L., Nepal, S., Zhang, X., Xiang, Y., Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples, IEEE Transactions on Cybernetics.
  42. Redmon, J., Farhadi, A., YOLO9000: better, faster, stronger, Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 7263-7271, 2017.
  43. Redmon, J., Farhadi, A., Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767, 2018.
  44. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M., Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.109, 2020.
  45. Bisong, E., Google colab, Building Machine Learning and Deep Learning Models on Google Cloud Platform, Apress, Berkeley, CA, 59-64, 2019.
  46. Wang, S., Niu, L., Li, N., Research on image recognition of insulators based on YOLO algorithm, 2018 international conference on power system technology (POWERCON), IEEE, 3871-3874, 2018.
  47. Clark, A., Pillow (PIL fork) documentation, <https://buildmedia.readthedocs.org/media/pdf/pillow/la-test/pillow.pdf>, Yayın tarihi Haziran 29, 2015. Erişim tarihi Kasım 23, 2021.