

## İnsansız Hava aracı (İHA) Görüntülerindeki Nesnelerin Kapsül Ağları Kullanılarak Sınıflandırılması

Atakan Körez<sup>1+</sup>, Necaattin Barışcı\*

<sup>1</sup>Bilgisayar Mühendisliği / Gazi Üniversitesi  
\*Corresponding Authors: atakan.korez@gazi.edu.tr  
+Speaker: nbarisci@gazi.edu.tr  
Presentation/Paper Type: Oral / Full Paper

**Özet**– Drone ya da İnsansız Hava Araçları (İHA), günümüzde birçok farklı sektör tarafından aktif şekilde kullanılmaktadır. Özellikle son yıllarda popülerlik kazanmayı başaran drone'lar; reklamcılıktan kargo taşımacılığına kadar çeşitli görevler üstlenebilmektedir. İHA'lar ile çekilen görüntülerdeki nesnelerin tespitindeki ilk aşama bu nesnelerin tanınmasıdır. Literatürde derin öğrenme ile nesnelerin sınıflandırılmasında çoğunlukla Evrişimsel Sinir Ağları kullanılmaktadır. Evrişimsel sinir ağları kullanılarak yapılan çalışmalarda nesne tanıma ve sınıflandırmada oldukça başarılı sonuçlar elde edilmiştir. Ancak bu sonuçlar evrişimsel sinir ağlarının sorunsuz bir yapıda olduğunu ortaya koymamaktadır. Bu çalışmada, evrişimsel sinir ağlarının dezavantajlarını ortadan kaldırmak için ortaya konulan Kapsül Ağları kullanarak İHA'dan çekilen görüntülerdeki nesnelerin sınıflandırılması amaçlanmaktadır.

**Anahtar Kelimeler**– İnsansız Hava Aracı, Nesne Sınıflandırma, Derin Öğrenme, Kapsül Ağları, Evrişimli Sinir Ağları

## Classification of Objects in Unmanned Aerial Vehicle (UAV) Images Using Capsule Networks

**Abstract** – Drone or Unmanned Aerial Vehicles (UAVs) are now actively used by many different sectors. In particular, the drone who have gained popularity in recent years; From advertising to cargo transportation, various tasks can be undertaken. The first step in detecting objects in images taken with UAVs is the recognition of these objects. In the literature, Convolutional Neural Networks are used mostly in the classification of objects with deep learning. Very successful results have been achieved in object recognition and classification in studies using convolutional neural networks. However, these results do not reveal that the convolutional neural networks have a good structure. In this study, it is aimed to classify the objects in the images taken from the UAV using the Capsule Networks, which are presented to eliminate the disadvantages of convolutional neural networks.

**Keywords**– Unmanned Air Vehicle, Object Classification, Deep Learning, Capsule Networks,, Convolutional Neural Networks

### I. GİRİŞ

İnsansız Hava Araçları (İHA), günümüzde birçok farklı sektör tarafından aktif şekilde kullanılmaktadır. Özellikle son yıllarda popülerlik kazanmayı başaran İHA'lar; reklamcılıktan kargo taşımacılığına kadar çeşitli görevler üstlenebilmektedir. İHA'lar ile çekilen görüntülerdeki nesnelerin tespitindeki ilk aşama nesnelerin tanınmasıdır. Literatürde derin öğrenme ile nesnelerin sınıflandırılması ilgili birçok çalışma bulunmaktadır. Bu çalışmaların tamamında Evrişimsel Sinir Ağları [1] kullanılmaktadır. Evrişimsel sinir ağları nesne tanıma ve sınıflandırma işlemlerinde oldukça başarılı sonuçlar elde etmektedir. Ancak bu sonuçlar evrişimsel sinir ağlarının sorunsuz bir yapıda olmadığını ortaya koymaktadır. Örneğin, eğitilmiş evrişimli sinir ağı, eğitilmiş olduğu nesneye farklı bir perspektiften baktığında aynı başarı ile tanıma/sınıflandırma işlemini

gerçekleştirememektedir. Özellikle yüz tanıma işlemlerinde yüzün tanınmasının yanında göz, burun, ağız gibi yüzü belirleyici parçalarının doğru yerlerinde olması gerekmektedir. Ancak evrişimsel sinir ağlarına verilen yüz resimlerinde göz ve burunun yeri değişmiş olsa bile ağ bunu yüz olarak algılayacaktır.

Bunun nedeni evrişimsel sinir ağlarında Ortaklama(Pooling) işlemi vasıtasıyla gerçekleştirilen boyut azaltma işlemi aynı zamanda bilgi kaybına neden olmaktadır. Ayrıca tanınmaya çalışılan nesnelerin örtüşüyor (overlapping) olması da evrişimli sinir ağı modellerinin sınıflandırma yaparken zorlanmasına neden olmaktadır [2]. Evrişimli sinir ağı modellerinin yetersiz kaldığı bu tip problemlere çözüm olarak Kapsül Ağları ve Dinamik Yönlendirme algoritması [2,3] önerilmiştir.

## II. MATERYAL VE YÖNTEM

Bu çalışmada, insansız hava aracından çekilmiş görüntülerdeki nesnelerin sınıflandırma işlemi Kapsül Ağları kullanılarak gerçekleştirilmiştir.

### A. Veri Seti

Çalışmada, Stanford Drone [4] veri seti kullanılmıştır. Veri seti, 8 farklı sahnedan çekilmiş toplam 60 videodan oluşmaktadır. 6 sınıf bulunmaktadır (insan, kaykay kullanan insan, bisiklet süren insan, araba, otobüs ve golf arabası). Veri seti kapsül ağları ile nesne sınıflandırması yapılması için bazı ön işlemlerden geçmiştir. Bunlar:

- Veri setindeki resimler 1300 x 800 piksel boyutlarında ve içerisinde birden fazla nesne barındırmaktadır. Bu nedenle etiket bilgisi kullanılarak resimler içerisindeki nesneler kırılmış ve her sınıfa ait resimler farklı klasörlere kaydedilmiştir (Şekil 3).
- Kırılan nesnelerin resim boyutları farklı olduğu ve kapsül ağlarının giriş boyutu 32 x 32 piksel olduğu için tüm sınıflarda ait nesnelerin boyutu 32x32 yapılmıştır.
- Her bir sınıfa ait nesne resim sayıları aşağıda

Sınıf	Yaya	Araba	Otobüs	Bisikletli	Kaykaycı	Cart
Sayı	35213	18054	5457	24768	15942	10246

verilmiştir. Oluşturulan bu veri setinin %70'i eğitim ve %30'u test olarak kullanılmıştır.

Tablo 1 Veri Seti Sınıf Sayıları (Toplam : 109679)

### B. Yöntem

Kapsül ağlarında; konum, yönelim, duruş ve açısal değerlerin değişmesi durumunda dahi bir grup nörondan oluşan kapsüller aracılığıyla nesnenin başarıyla tanınabilmesi için nesneyi temsil eden kalınlık, ölçek, kaydırma vb. özelliklerin anlaşılabilir yönlendirme (routing-by-agreement) ile öğrenilmesi önerilmiştir [5].

Kapsül modelinde standart bir sinir ağı modelinden farklı olarak kapsül katmanlarının çıkışları skaler değil vektörel olarak ifade edilmektedir. Derin sinir ağları modellerinde tek bir nöron için çoğunlukla doğrusal olmayan ReLU (Rectified Linear Unit) [6] aktivasyon fonksiyonu kullanılırken, kapsül ağlarında girdisinin ve çıktısının vektör olduğu denklem (1)'de gösterilen ezme (squash) fonksiyonu kullanılmaktadır.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (1)$$

$S_j$  : Ezme (squash) parametresi

$V_j$  : Çıkış Vektörü

Kapsül ağlarında evrişim katmanının çıkışı öncül kapsüle giriş olarak uygulanmaktadır. Çıkışlardan vektörler elde edebilmek için yeniden boyutlandırma yapılmaktadır. Bu vektörlerin değerlerini sınırlandırabilmek için ezme işlemi uygulanır. Algoritma aşağıda belirtildiği gibidir:

Kapsül ağında hesaplanan vektörler, nesnelere temsil eden konum, yönelim, kalınlık ve yönlendirme gibi özellikleri

#### Algoritma 1: Yönlendirme

```

1. algoritma Yönlendirme ( $\hat{u}_{j|l}, r, l$ )
2. tüm kapsüller için  $i$ . katman  $l$  ve kapsül  $j$ . katman
   ( $l + 1$ ):  $b_{ij} \leftarrow 0$ 
3. for  $r$  iterasyon do
4. tüm kapsüller için  $i$ . katman
    $l$ :  $c_i \leftarrow \text{softmax}(b_i) \Rightarrow$  softmax denklemi
5. tüm kapsüller için  $j$ . katman ( $l + 1$ ):  $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|l}$ 
6. tüm kapsüller için  $j$ . Katman
   ( $l + 1$ ):  $v_j \leftarrow \text{ezme}(s_j) \Rightarrow$  ezme denklemi
7. tüm kapsüller için  $i$ . katman  $l$  ve kapsül  $j$ . katman
   ( $l + 1$ ):  $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|l} \cdot v_j$ 
return  $v_j$ 

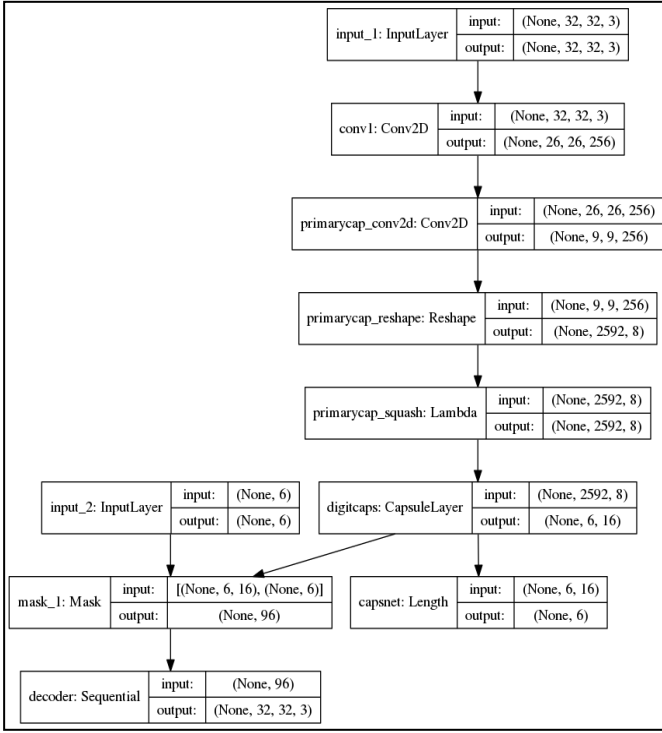
```

belirtmektedir. Bu kapsamda:

- Vektör uzunluğu: Nesnenin o noktadaki tahmini bulunma olasılığını temsil eder.
- Vektör yönü: Nesnenin poz, deformasyon, hız, yansıtılabilirlik (albedo), renk derecesi (hue) ve doku (texture) parametrelerini verir.

Çalışmada kullanılan veri setindeki görüntülerdeki nesnelerin tespiti, nesnelerin görünümünün çok küçük ve hep sabit açıdan olması nedeniyle, evrişimsel ağlar ile yapılan nesne tespiti başarı oranı insan gözünden bakış açısından yapılan nesne tespit başarı oranlarından düşük çıkmaktadır. Bu sorunu gidermek için çalışma kapsamında oluşturulan model Şekil 1'de görülmektedir. Modelde, ilk katmanda  $7 \times 7$  evrişim işlemi uygulanmaktadır. Bu işlem sonucunda  $26 \times 26 \times 256$  boyutlu tensör, öncül kapsül katmanına iletilmektedir. Kapsül katmanında yapılan işlemler ve elde edilen çıkış boyutları sırasıyla aşağıdaki gibidir;

- Evrişim:  $9 \times 9 \times 256$
- Yeniden boyutlandırma:  $2592 \times 8$
- Ezme:  $2592 \times 8$
- Kapsül katmanı çıkışı (DigitsCaps):  $10 \times 16$



Şekil 1 Kapsül Ağ Modeli

Çalışmada uygulanan derin öğrenme modellerin deneyleri (Şekil 4) **TESLA P100 GPU** kullanılarak gerçekleştirilmiştir. Modeller Keras [7] kütüphanesi kullanılarak Python [8] programlama dilinde oluşturulmuştur.

### III. SONUÇLAR

Önişleme tabi tutulmuş Stanford Drone veri setinin çeşitli veri artırma, eğitim süresinin artırılması, başlangıç ağırlığı kullanılması gibi deneyler ile modelin performansı değerlendirilmiştir.

Tablo 2 Modele Uygulanan Yöntemlerin Başarıma Etkisi

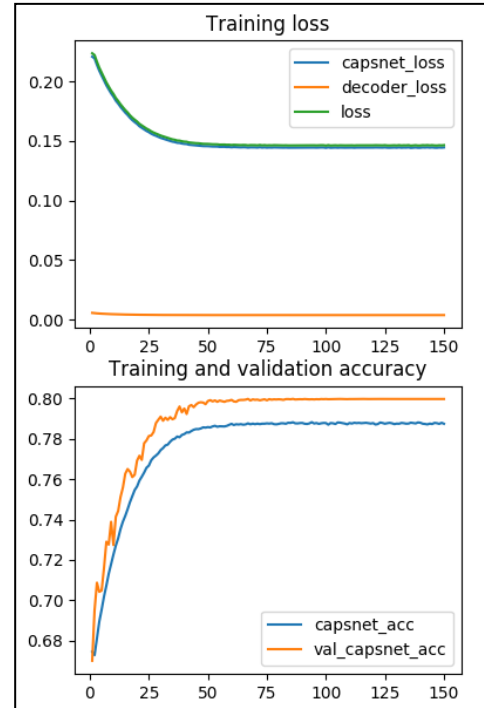
Uygulanan Yöntem		Başarım (%)
Veri Artırma	15 Derece Döndürme	78,01
	Dikey Döndürme	77,85
	Kaydırma	77,52
Eğitim Süresi (Epoch)	50	76,02
	100	78,54
	150	79,97
	200	79,98
Başlangıç Ağırlığı	Var	74,28
	Yok	78,69

Tablo 2 incelendiğinde;

- Başlangıç ağırlıksız başarımların daha iyi olduğu görülmektedir. Bu da hava görüntülerinin normal bakış açısından alınan eğitilmiş veriler ile uyumsuz olmasıdır,

- Eğitim serisinin artması belirli bir süre boyunca başarıyı arttırdığı ancak bir müddet sonra başarıma etkisinin olmadığı görülmektedir,
- Veri artırma tekniklerinin başarıya katkısının genellikle aynı olması görüntülerin dik bakış açısından çekilmesi nedeniyledir.

Oluşturulan model 10 kez çalıştırılıp sonuçların ortalaması alınarak elde edilen başarımların ortalaması ve eğitim kayıp eğrisi Şekil 2’de gösterilmektedir.



Şekil 2 Başarım Oranı Değişimi ve Eğitim Kayıp Eğrisi

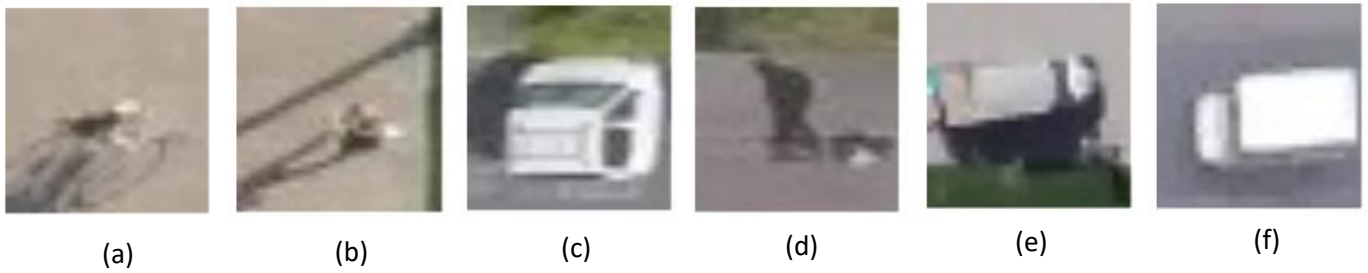
Sonuç olarak, literatürde evrimsel sınırlı ağırlıklara ciddi bir alternatif olan kapsül ağı modelinin daha önce hiç denenmemiş bir domain olan İHA ile çekilmiş görüntülerden oluşan veri seti üzerindeki başarısı incelenmiştir. Veri setini kapsül ağına uygulamak için yapılan ön işlemler ve çeşitli hiper parametrelerin değiştirilmesi ile oluşan en iyi ortalama başarımların %79,98 olarak tespit edilmiştir. Gelecekte modelin başarımlarının, farklı hiper parametreler ve mimariler ile de ölçülmesi hedeflenmektedir.

### TEŞEKKÜR

Çalışma kapsamında oluşturulan modelin deneylerini yapmış olduğumuz yüksek hesaplama sunucuları için TÜBİTAK ULAKBİM Yüksek Başarımlı ve Grid Hesaplama Merkezi’ne teşekkür ederiz.

## REFERENCES

- [1] Bengio, Y., Goodfellow, I. and Courville, (2016). A., An MIT Press Book, Chapter 9, Convolutional Networks .
- [2] Sabour, S., Frosst, N. ve Hinton, G.E., "Dynamic Routing Between Capsules", arXiv preprint arXiv:1710.09829, 2017.
- [3] Hinton, G. E., Krizhevsky A. ve Wang, S. D. "Transforming Autoencoders." International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2011.
- [4] Computational vision and Geometry lab, Stanford Drone Dataset. Available online: [http://cvgl.stanford.edu/projects/uav\\_data/](http://cvgl.stanford.edu/projects/uav_data/)
- [5] M. A. Kızrak, F. Beşer, B.Polat, T. Yildirim, "Kapsül Ağları ile İşaret Dili Tanıma", 26th Signal Processing and Communications Applications Conference (SIU), 2018.
- [6] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", 27th International Conference on International Conference on Machine Learning, 2010
- [7] <https://keras.io/> Erişim Tarihi : 01.12.2018
- [8] <https://www.python.org/> Erişim Tarihi : 14.08.2018



Şekil 3 Örnek Veri Seti Resimleri, (a) Bisikletli, (b) Yaya, (c)Araba, (d) Kaykaycı, (e) Otobüs, (f) Cart Aracı

```

BLURM_MODELIST akya2
NUMBER OF CORES 40
Namespace(batch_size=128, debug=0, epochs=150, is_training=0, lam_recon=0.512, lr=0.001, num_routing=3, save_dir='model_kayit', weights='model_kayit/trained_model.h5')
Layer (type)                Output Shape                Param #                    Connected to
-----
input_1 (InputLayer)        (None, 32, 32, 3)          0                          0
conv1 (Conv2D)              (None, 26, 26, 256)        37888                      input_1[0][0]
primarycap_conv2d (Conv2D)  (None, 9, 9, 256)          5308672                    conv1[0][0]
primarycap_reshape (Reshape) (None, 2592, 8)            0                          primarycap_conv2d[0][0]
primarycap_squash (Lambda)  (None, 2592, 8)            0                          primarycap_reshape[0][0]
digitcaps (CapsuleLayer)    (None, 6, 16)              1990656                    primarycap_squash[0][0]
input_2 (InputLayer)        (None, 6)                  0                          0
mask_1 (Mask)               (None, 96)                 0                          digitcaps[0][0]
                               input_2[0][0]
capsnet (Length)           (None, 6)                  0                          digitcaps[0][0]
decoder (Sequential)        (None, 32, 32, 3)          3723776                    mask_1[0][0]
-----
Total params: 11,060,992
Trainable params: 11,060,992
Non-trainable params: 0
-----
Test acc: 0.7997799646241468
Reconstructed images are saved to real_and_recon.png
-----

```

Şekil 4 Eğitim İşlemi Sonucu