



Kapsül Ağları ile Yüz Verilerinin Sınıflandırılması

Ayşe Çoban^{1*}, Fatih Özyurt²

^{1*} Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye, (ORCID: 0000-0002-9922-8616), aysecoban9603@gmail.com

² Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, (ORCID: 0000-0002-8154-6691), ozyurfatih@gmail.com

(İlk Geliş Tarihi 22 Eylül 2021 ve Kabul Tarihi 23 Ocak 2022)

(DOI: 10.31590/ejosat.999055)

ATIF/REFERENCE: Çoban, A. & Özyurt, F. (2022). Kapsül Ağları ile Yüz Verilerinin Sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, (33), 176-183.

Öz

Yüz verilerini sınıflandırma işlemi, üzerinde çok sayıda çalışma yapılan derin öğrenme konularından birisidir. Bu konuda evrişimli sinir ağları, kolay uygulanabilir olmaları ve başarılı sonuçlar vermeleri nedenlerinden ötürü tercih edilen derin öğrenme uygulamalarının başında gelmektedir. Buna karşın evrişimli sinir ağlarında bulunan havuzlama katmanı verilerde bilgi kaybına neden olmaktadır. Ayrıca evrişimli sinir ağları, verideki bileşenlerin birbirine göre durumlarını göz ardı ederek eğitim işlemini gerçekleştirmektedir. Bu duruma çözüm olarak kapsül ağları önerilen derin öğrenme yöntemlerindedir.

Bu çalışmada; 5 kişinin her biri için 200 veriden oluşan, toplam 1000 görüntü verisi içeren bir veri seti hazırlanmıştır. Hazırlanan veri seti ile sıkça kullanılan evrişimli sinir ağı yöntemlerinden olan AlexNet, Vgg16, Vgg19, ResNet50 algoritmalarının her biri ve önerilen kapsül ağlar yöntemi kullanılarak eğitim işlemi yapılmıştır. Yapılan çalışmanın sonucunda evrişimli sinir ağı yöntemlerinin başarımları ile kapsül ağlar yönteminin başarımları karşılaştırılmıştır.

Anahtar Kelimeler: Kapsül Ağlar, Yüz Tanıma, Örüntü Tanıma.

Classification of Facial Data with Capsule Networks

Abstract

The process of classifying facial data is one of the deep learning topics that has been studied extensively. In this regard, convolutional neural networks are among the most preferred deep learning applications because they are easy to implement and give successful results. On the other hand, the pooling layer in convolutional neural networks causes information loss in the data. In addition, convolutional neural networks perform the training process by ignoring the relative states of the components in the data. As a solution to this situation, capsule networks are one of the recommended deep learning methods.

In this study; A data set consisting of 200 data for each of the 5 people and a total of 1000 image data was prepared. With the prepared data set, each of the AlexNet, Vgg16, Vgg19, ResNet50 algorithms, which are frequently used convolutional neural network methods, and the proposed capsule network method were used for training. As a result of the study, the performances of the convolutional neural network methods and the performances of the capsule networks method were compared.

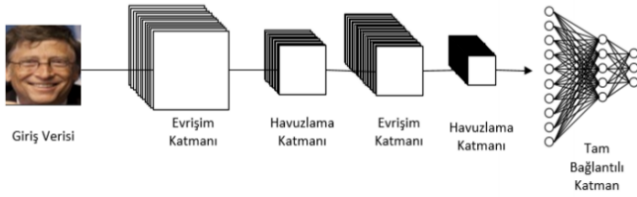
Keywords: Capsule Networks, Face Recognition, Pattern Recognition.

* Sorumlu Yazar: aysecoban9603@gmail.com

1. Giriş

Yüz verilerini sınıflandırma işlemi bir grup insanın yüz görüntülerinden oluşan verilerin belirlenen sınıflara göre sınıflandırılmasıdır. Yüz verilerini sınıflandırma işlemi örüntü tanımanın ilgilendiği önemli problemlerden birisidir. Yüz verileri günümüzde yoğun olarak evrişimli sinir ağları kullanılarak sınıflandırılmaktadır.

Evrişimli sinir ağları temel olarak Şekil 1'de gösterildiği gibi; evrişim katmanı ardından havuzlama katmanı ve tam bağlantılı katman dediğimiz katmanlardan oluşur (İnik ve Ülker, 2017).



Şekil 1. Evrişimli sinir ağı katmanları (Figure 1. Convolutional neural network layers)

Evrişim katmanı, giriş olarak alınan görüntülerin belirlenen boyutlarda ve sayılarda filtrelerden geçirildiği katmandır. Bu katman sayesinde görüntünün belirli sayıda filtreden geçirilmiş formları elde edilir. Bu katman bir veya birden fazla kez art arda uygulanabilir.

Evrişimli sinir ağlarının mimarisinde evrişim katmanından sonra havuzlama katmanı gelmektedir. Havuzlama katmanında evrişim katmanından alınan görüntülere farklı çeşitleri bulunan havuzlama işlemlerinden (maksimum, minimum, ortalama havuzlama) seçilen bir havuzlama yöntemi uygulanır. Burada amaç görüntüde istenmeyen öz niteliklerin boyut olarak sistemi verimsiz hale getirmemesi için en aza indirilmesidir. Bu katman sonucunda havuzlama katmanının parametrelerine göre boyutları azaltılan ve istenilen öz niteliklere sahip görüntüler elde edilir. Bu katman da modelin yapısına göre bir veya birden fazla kez tekrarlanabilir.

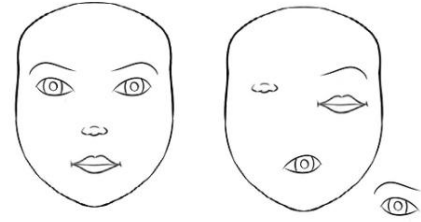
Son olarak tam bağlantılı katman gelmektedir. Bu katman basit anlamda bir yapay sinir ağıdır ve elde edilen görüntü matrisleri bir düzleştirme işleminden geçerek bu katmana iletilir. Bu katman sonunda bazı doğrusal olmayan aktivasyon fonksiyonları aracılığıyla sınıflandırma işlemi gerçekleştirilir.

Evrişimli sinir ağlarının kabul görmüş birçok modeli bulunmaktadır. İlk olarak 2012'de düzenlenen ImageNet yarışmasında önerilen AlexNet modeli sınıflandırma başarımı ile evrişimli sinir ağları literatüründe yerini almıştır (Krizhevsky vd., 2017). AlexNet modelinin sınıflandırma üzerinde evrişimli sinir ağlarına olan ilgiyi büyük ölçüde artırarak, bu alanda yeni modeller geliştirilmesine neden olmuştur. Düzenlenen ImageNet yarışmaları ile 2014 yılında birinci olan VGGNet (Simonyan ve Zisserman, 2014) ve 2016 yılında birinci olan ResNet (He vd., 2016) modelleri literatürde yerini alan diğer yüksek başarılı modellerdendir.

Evrişimli sinir ağlarının sınıflandırma üzerine olan bu başarımları ilgili verilerin farklı perspektiflerden örneklerinin alınması durumunda düşmektedir. Bu sorunun başlıca nedeni evrişimli sinir ağlarında kullanılan havuzlama katmanının

verilerde bilgi kaybına neden olmasıdır (Sabour vd., 2017). Bu kayıp, yüz verilerinin sınıflandırılmasında, sınıflandırma işlemine etkisi olabilecek bazı öz niteliklerin de kaybının olabileceği anlamına gelmektedir. Bu sorunun çözümü olarak kapsül ağları tercih edilen yöntemlerdendir. Kapsül ağlar verilerdeki bileşenlerin birbirine göre konum, açı, yönelim gibi özellikleri göz önüne alınarak eğitim işlemi yapılan ağlardır.

Ayrıca Evrişimli sinir ağları görüntüyü anlamlandıran nesnelerin birbiriyle olan durumlarını göz önüne almadan eğitilmektedir (Toraman, 2020). Örneğin bir yüz görüntü verisinde yüzü oluşturabilmesi için gözler, burun, ağız gibi bileşenlerin belirli bir konumu vardır. Bunların birbirlerine göre Şekil 2'de gösterildiği gibi yanlış konumlarda bulunması halinde söz konusu görüntü bir yüzü tanımlayamaz. Bu durumda bu bileşenlerin birbirine göre konum, durum, yönelim gibi özelliklerinin göz ardı edilmemesi gerekmektedir.

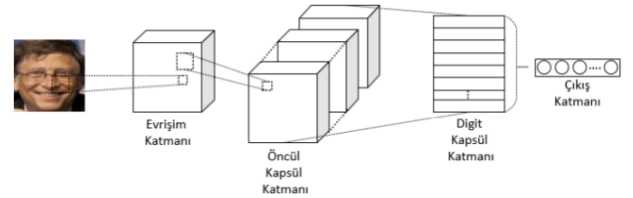


Şekil 2. Yüz bileşenlerinin birbirine göre durumları (Figure 2. The relative states of the facial components)

Bu problemlere bir çözüm olarak 2017 yılında Geoffrey Hinton, Sara Sabour ve diğerleri tarafından kapsül ağlar ve dinamik yönlendirme algoritması önerilmiştir (Sabour vd., 2017).

Kapsül ağlar mimarisinde evrişim katmanları kullanılması yönüyle evrişimli sinir ağlarına benzemektedir. Kapsül ağının ilk aşamasında kullanılan evrişim katmanı, evrişimli sinir ağlarındakiyle aynıdır. Fakat bu evrişim katmanından sonra gelen evrişim katmanında ve diğer katmanlarda dinamik yönlendirme algoritması uygulanmaktadır.

Evrişimli sinir ağlarındaki nöronların yerine kapsül ağlarda kapsüller kullanılır. Kapsül ağı katmanları temel olarak Şekil 3'de gösterildiği gibidir.



Şekil 3. Kapsül ağı katmanları (Figure 3. Capsule network layers)

Kapsül ağları giriş olarak aldığı verileri özelliklerine göre parametrelendirerek vektörel olarak ele alır, dinamik yönlendirme algoritması sayesinde de özelliklerine göre birine bağlı kapsüllerle eğitim gerçekleştirir. Kapsül ağlarının bu özellikleri sayesinde farklı perspektiflerden bakılan görüntüler aynı başarımlarla daha doğru sonuçlar üretebilmektedir.

Kapsül ağlar dolayısıyla evrişimli sinir ağlarına göre veriyi oluşturan parçalardan daha fazla bilgi alır (Toraman, 2020).

Dinamik yönlendirme algoritması ile de girişleri benzer kapsüllerin birbiri ile bağlantılı aktive olmalarını sağlar.

Bu çalışmada yüksek başarımları nedeniyle kabul gören AlexNet, Vgg16, Vgg19, ResNet50 evrişimli sinir ağı modelleri ve kapsül ağılar modeli ile yüz verileri üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Gerçekleştirilen uygulamaların sonuçlarına bakılarak kapsül ağılar ile evrişimli sinir ağlarının yüz verilerini sınıflandırma problemi üzerinde başarımları, verimliliği karşılaştırılmıştır.

2. Kapsül Ağlar

Evrişimli sinir ağları örüntü tanıma problemlerinde sıkça tercih edilen ve başarılı sonuçlar elde eden bir yöntemdir. Fakat eğitilmiş bir evrişimli sinir ağı görüntüdeki nesnelere birbirine olan durumunu, açısını, derinliğini tanımlamak için yeterince elverişli bir yöntem değildir. Bu sorun evrişimli sinir ağları mimarisinde kullanılan havuzlama katmanlarından kaynaklanmaktadır. Havuzlama katmanında görüntü, boyutları düşürülürken görüntüde bulunan bazı önemli öz nitelikleri kaybeder. Nesnelere göre açı, derinlik gibi olguların o nesneyi ayırt etmek için önemli olduğu durumlarda özelleştirilmiş bir evrişimli sinir ağı mimarisi olarak bilinen kapsül sinir ağları ve dinamik yönlendirme algoritması kullanılmaktadır.

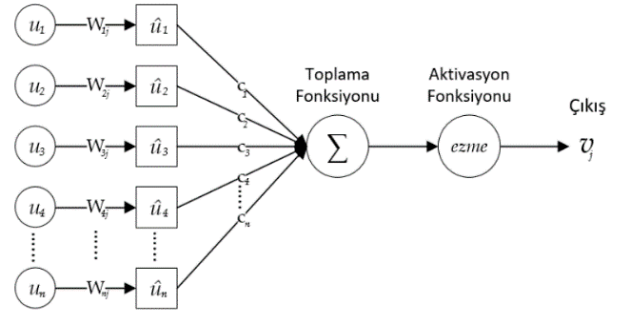
Kapsül ağlar, nöronlardan ziyade kapsüllerden oluşmaktadır. Bir kapsül ise, görüntünün belirli bir bölgesinde bulunan belirli bir nesneyi (örneğin göz) algılamayı öğrenen bir grup nörondan oluşur. Kapsül ağlarının mimarisi ilk olarak alınan verilere bir evrişim katmanı uygulanması ile başlamaktadır. Burada ilk kullanılan evrişim katmanı evrişimli sinir ağlarında kullanılan evrişim katmanlarının aynısıdır (Öztürk ve Özkaya, 2020). Daha sonra öncül kapsül katmanı adı verilen katmana geçilir. Bu katmanda da yine bir evrişim işlemi uygulanır. Uygulanan evrişim işleminin önceki evrişim işleminden farkı bu katmana gelen görüntüleri vektörel hale getirmek amacıyla bir yeniden boyutlandırma işlemi uygulanması ve ardından aktivasyon fonksiyonu olarak ezme aktivasyon fonksiyonunun kullanılmasıdır. Bu işlemlerin sonunda dinamik yönlendirme algoritmasının bulunduğu kapsül katmanı yer almaktadır. Bu katmanın sonucunda DigitCaps adı verilen vektörler elde edilmektedir. DigitCaps vektörlerinin sayısı sınıflandırma işlemi için belirlenen sınıf sayısı kadardır.

2.1. Dinamik Yönlendirme Algoritması

Dinamik yönlendirme algoritması, girişleri benzer alt seviyede bulunan kapsülleri, girişleri benzer üst seviyede bulunan kapsüller ile eşleştiren bir algoritmadır.

Evrişimli sinir ağlarında, evrişim katmanlarının çıkışlarında genel olarak doğrusal olmayan ReLU aktivasyon fonksiyonu kullanılmakta ve çıkışlar skaler olarak ifade edilmektedir (Özyurt vd., 2018). Kapsül ağlarda ise ReLU fonksiyonuna karşılık ezme fonksiyonu (squash) denilen bir aktivasyon fonksiyonu kullanılır ve çıkışlar skaler değil vektörel olarak ifade edilir. Buradaki vektörler alınan veri bileşeninin, hedef sınıfta bulunma olasılığını ve ayrıca durumunu başka bir deyişle parametrelerini ifade etmektedir. Görüntüde istenilen nesnenin olduğu alanlarda vektör uzunluğu büyük olmadığı alanlarda ise küçüktür (Beser vd., 2018). Ezme fonksiyonu vektörlerin uzunluğuna göre vektör uzunluklarının sıfır ile bir arasında bir

değer almalarını sağlar. Söz konusu vektörün yönü ise bileşenin durumunu (açı, yönelim, kalınlık vb.) ifade etmektedir. Dinamik yönlendirme algoritmasının uygulanması havuzlama yöntemlerinin uygulanmasına göre öznelilik çıkarımında daha etkili bir yöntemdir (Kınlı ve Kırac, 2020).



Şekil 4. Kapsül giriş ve çıkışları (Figure 4. Capsule inputs and outputs)

Kapsülün girişleri u_i vektörleri olmak üzere bu vektörlere bir dönüşüm matrisi W_{ij} uygulanır. Böylece tahmin vektörü olarak bilinen \hat{u}_{ij} elde edilir. Daha sonra tahmin vektörünün, kapsül i 'nin kapsül j 'yi ne kadar aktive edebileceğinin ölçümü olan c_{ij} ile çarpımlarının toplamı alınır, bu bize ağırlıklı bir toplam olan s_j 'yi verir.

$$s_j = \sum_i c_{ij} \hat{u}_{ij} \quad (1)$$

$$\hat{u}_{ij} = W_{ij} u_i \quad (2)$$

Hesaplanan bu değerlere ezme fonksiyonunun uygulanması ve kapsül çıkışı olan v_j 'nin elde edilmesi denklem 3'de gösterilmiştir.

$$v_j = \frac{\|s_j\|^2 s_j}{1 + \|s_j\|^2 \|s_j\|} \quad (3)$$

Tahmin vektörü \hat{u}_{ij} ile kapsül çıkışının v_j skaler çarpımlarının alınmasıyla benzerlik b_{ij} hesaplanır.

$$b_{ij} \leftarrow \hat{u}_{ij} v_j \quad (4)$$

Daha sonra b_{ij} 'ye softmax uygulanması ile c_{ij} 'nin yeni değeri denklem 5'deki gibi hesaplanır.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (5)$$

b_{ij} daha doğru bir hale getirilmek için yinelemeli olarak (genel olarak 3 adımda) güncellenir.

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{ij} v_j \quad (6)$$

Genel olarak dinamik yönlendirme algoritması Tablo 1’de gösterildiği gibi kaba kod yardımıyla özetlenebilir.

Tablo 1. Dinamik Yönlendirme Algoritması

<ol style="list-style-type: none"> 1. algoritma Yönlendirme (\hat{u}_{ij}, r, l) 2. tüm kapsüller i, katmanlar l ve bulunulan kapsül katmanı j olmak üzere $(l + 1)$: $b_{ij} \leftarrow 0$ 3. for r iterasyon do 4. tüm kapsüller için i. katman l: $c_i \leftarrow \text{softmax}(b_i)$ 5. tüm kapsüller için j. katman $(l + 1)$: $s_j \leftarrow \sum_i c_{ij} \hat{u}_{ij}$ 6. tüm kapsüller için j. katman $(l + 1)$: $v_j \leftarrow \text{ezme}(s_j)$ 7. tüm kapsüller için i. katman ve l kapsül j. katman $(l + 1)$: $b_{ij} \leftarrow b_{ij} + u_{ji} v_j$ <p>return v_j</p>

İlk satır, r defa algoritmayı yenilenerek, l seviyesindeki tüm kapsülleri ve bunların çıktılarını (\hat{u}_{ij}) aldığını ifade etmektedir. En son satır ise algoritmanın sonucunda daha yüksek seviyeli bir kapsül çıktısını (v_j) üreteceğini ifade eder.

b_{ij} katsayısı, yinelemeli olarak güncellenecek geçici bir değerdir ve prosedür bittikten sonra değeri c_{ij} ’de saklanacaktır. Eğitimin başlangıcında b_{ij} değeri sıfırdan başlatılır.

4. satırdaki adım, daha düşük seviyeli bir kapsül i için tüm yönlendirme ağırlıkları olan c_i vektörünün değerini hesaplar. Bunu yaparken ise b_i değerlerini softmax fonksiyonundan geçirir. Tüm düşük seviyeli kapsüller için bu işlem yapılır. İlk yinelemede, tüm c_{ij} katsayılarının değeri eşit olacaktır, çünkü ikinci satırda tüm b_{ij} değerleri sıfıra ayarlanmıştır. Süreç tekrarlandıkça bu tek tip dağılımlar değişecektir.

Tüm düşük seviyeli kapsüller için tüm ağırlıklar (c_{ij}) hesaplandıktan sonra, daha yüksek seviyeli kapsüllere bakılması için 5. satıra geçilir. Bu adımda, önceki adımda belirlenen c_{ij} yönlendirme katsayıları ile ağırlıklandırılan girdi vektörlerinin doğrusal bir kombinasyonunu hesaplar. Bu da s_j çıktı vektörünü üretir. Tüm yüksek seviyeli kapsüller için bu işlem yapılır.

Son adımdaki vektörler, vektörün yönünün korunmasını sağlar ve uzunluğunun 1’den fazla olmamasını sağlayan ezme doğrusal olmayan aktivasyon fonksiyonundan geçirilir. Bu adım, tüm yüksek seviyeli kapsüller için v_j çıktı vektörünü üretir. Bu adımlar, her bir yüksek seviyeli kapsül j ’ye bakar ve ardından her girişi inceler ve formüle göre karşılık gelen b_{ij} ağırlığını günceller. Böylece düşük seviyeli kapsül, çıktısını benzer olan daha yüksek seviyeli kapsüle gönderecektir. Bu benzerlik, iç çarpım tarafından yakalanmaktadır.

Bu algoritma, sonraki yinelemelerde 3. adımdan başlar ve işlemleri r kez tekrar eder. Böylece evrişimli sinir ağlarından farklı olarak nesnelerin birbirine göre olan durumları da hesaba katılmış olur.

2.1.1. Kayıp Hesaplama

Eğitim sırasında her bir eğitim örneği için kayıp, denklem 7’deki eşitliğe göre 5 vektörün her biri için hesaplanacak ve ardından son kaybı hesaplamak için bu hesaplanan 5 değer toplanacaktır. Kayıp fonksiyonu formülünde sınıf tahmini, T_c ’nin değerini belirler. Sınıf doğru tahmin edilmiş ise T_c ’nin değeri 1 aksi halde 0 olur.

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 + \lambda(1 - T_c) \max(0, \|v_c\| - m^-)^2 \quad (7)$$

Formüle göre, bir örneğin sınıfı doğru tahmin edildiğinde bir doğru sınıf ve dört yanlış sınıf oluşur. Doğru sınıfın T_c değeri 1 ve geri kalan sınıfların T_c değeri 0 olacaktır. Bu durumda doğru sınıfın kayıp değeri hesaplanmak istendiğinde eşitliğin birinci kısmı kullanılacaktır (Denklem 8).

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 \quad (8)$$

Doğru tahmin durumunda digit kapsülün kaybını hesaplamak için, digit kapsülün çıkış vektörü m^+ ’dan (0.9 sabit değer) çıkarılır. Ardından, elde edilen değerın karesi alınır. Bu işlem ile kapsül çıkışı 0.9 değerinden fazla olduğu durumlarda hata sıfır, diğer durumlarda sıfırdan büyük bir sayı elde edilir.

Doğru etiketle eşleşmeyen digit kapsüller için T_c değeri sıfır olacak ve bu nedenle eşitliğin ikinci kısmı kullanılacaktır (Denklem 9).

$$L_c = \lambda(1 - T_c) \max(0, \|v_c\| - m^-)^2 \quad (9)$$

Yanlış tahmin durumunda digit kapsülün kaybını hesaplamak için, digit kapsülün çıkış vektöründen m^- (0.1 sabit değer) değeri çıkarılır. Ardından, elde edilen değerın karesi alınır. Bu işlem ile kapsül çıkışı 0.1 değerinden az olduğu durumlarda hata sıfır, diğer durumlarda sıfırdan büyük bir sayı elde edilir.

Son olarak, formüle, eğitim sırasında sayısal kararlılık için lambda katsayısı (0.5 sabit değer) dâhil edilmiştir.

Kapsül ağırları evrişimli sinir ağlarının havuzlama katmanından kaynaklı oluşturduğu sorunlara, dinamik yönlendirme algoritmasıyla yeni bir yaklaşım öneren bir yöntemdir. Diğer bir deyişle evrişimli sinir ağlarında bulunan havuzlama katmanının yerini kapsül ağlarda dinamik yönlendirme algoritması almaktadır. Dinamik yönlendirme algoritması sayesinde verilerden ve söz konusu verileri oluşturan bileşenlerinden daha fazla bilgi öğrenilir. Bu özelliklerinden ötürü kapsül ağları öğrenilecek veride; derinlik, örtüşme, kalabalık gibi eğitim işlemini zorlaştıran durumlarda tercih edilen yöntemlerdendir.

3. Materyal ve Metot

Bu çalışmada, yüz tanıma ve sınıflandırma işlemi evrişimli sinir ağları algoritmalarından olan AlexNet, Vgg16, Vgg19, ResNet50 modelleri ve kapsül ağları kullanılarak Google Colaboratory çalışma ortamında Tesla K80 donanımı ile gerçekleştirilmiştir. Vgg16, Vgg19 ve ResNet50 modelleri Keras kütüphanesinin uygulamalarından alınarak Python programlama dilinde oluşturulmuştur. AlexNet ve Kapsül ağlar modeli ise Tensorflow ve Keras kütüphaneleri kullanılarak Python programlama dilinde oluşturulmuştur.

3.1. Veri Seti

Bu çalışma için 5 kişinin her biri için 200 tane olmak üzere kişilerin yüz verilerini içeren 64x64 boyutunda toplam 1000

renkli görüntüden oluşan bir veri seti, internet ortamından ünlü kişilerin yüz görüntüleri toplanarak hazırlanmıştır. Hazırlanan veri setinin bir örneği Şekil 5’de gösterildiği gibidir.



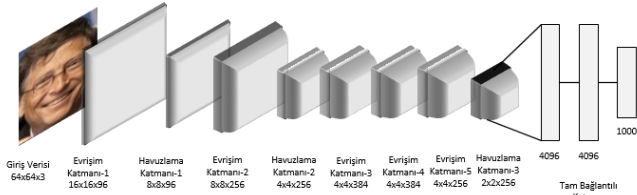
Şekil 5. Hazırlanan veri setine dair bir örnek (Figure 5. An example of the prepared data set)

Hazırlanan veri setinin %60’ı eğitim işlemi için, % 20’si doğrulama işlemi ve kalan %20’si ise test işleminde kullanılmak üzere ayrılmıştır.

3.2. Evrişimli Sinir Ağları Modelleri Yapısı

3.2.1. AlexNet Modeli

AlexNet modeli başarılarını kanıtlamış başlıca evrişimli sinir ağı algoritmalarından birisidir. Bu modelde 5 adet evrişim katmanı, 3 adet havuzlama katmanı ve 3 adet tam bağlantılı katman yer almaktadır (Toğaçar vd., 2020). Modelin yapısı Şekil 6’daki gibidir.



Şekil 6. AlexNet modeli yapısı (Figure 6. AlexNet model structure)

3.2.2. Vgg16 Modeli

Vgg16 modeli toplam 16 katmandan oluşan bir evrişimli sinir ağı yöntemidir. Modelin yapısı AlexNet modelinin yapısına oldukça benzemektedir. Bu model, yapısında 13 adet evrişim katmanı ve 3 adet tam bağlantılı katman bulundurmaktadır.

3.2.3. Vgg19 Modeli

Vgg19 modeli toplam 19 katmandan oluşan bir evrişimli sinir ağı yöntemidir. Yöntemin yapısı basit bir deyişle Vgg16 modelinin yapısının daha derinleştirilmiş bir halidir. Modelin yapısında 16 adet evrişim katmanı ve 3 adet tam bağlantılı katman yer almaktadır.

3.2.4. ResNet50 Modeli

ResNet50 modeli diğer evrişimli sinir ağları modellerinden farklı olarak ardışık ağı mimarisi yerine bünyesinde artık blok

yapısı bulundurmaktadır (Doğan, 2018). Bu yapının kullanılmasının nedeni, ağı yapısı derinleştikçe başarımın düşmesine engel olmak, başarımın düşeceği durumlarda aynı kalmasını sağlamaktır.

3.3. Kapsül Ağları Model Yapısı

Yüz verilerini sınıflandırma işleminde kullanılan model temel olarak altı katmandan oluşturulmuştur.

3.3.1. Giriş Katmanı

Giriş katmanı modelin ilk katmanıdır ve 64x64 boyutunda olan 3 kanallı (rgb) eğitim görüntü verilerinden oluşur. Giriş olarak alınan bu veriler evrişim katmanına gönderilir.

3.3.2. Evrişim Katmanı

Girişler evrişim katmanına iletilir. Bu katman evrişimli sinir ağları mimarisinde bulunan evrişim katmanlarının aynıdır. Evrişim katmanının parametreleri Tablo 2’de gösterildiği gibidir. Bu katmandan sonra özellik kapsül katmanı olarak adlandırdığımız öncül kapsül katmanı gelmektedir.

Tablo 2. Evrişim Katmanı Parametreleri

Evrişim Katmanı Parametreleri	
Filtre Sayısı	256
Çekirdek Boyutu	9
Kaydırma Adımı	1
Doldurma	Valid
Aktivasyon Fonksiyonu	ReLU

Görüntü verilerinin boyutu evrişim katmanından sonra (çekirdek boyutu - 1 = 8) 8 piksel kaybeder. Uygulanan evrişim katmanından sonra görüntü boyutları 56x56 piksele düşmüştür. Daha sonra elde edilen bu görüntülere özellik kapsülü katmanı uygulanmıştır.

3.3.3. Özellik Kapsülü Katmanı

Bu katman evrişim katmanı çıkışlarını giriş olarak almaktadır. Evrişim katmanından farklı olarak bu katmanda yeniden boyutlandırma işlemi ve ezme aktivasyon fonksiyonu uygulanmaktadır. Özellik kapsülünün parametreleri Tablo 3’de verildiği gibidir.

Tablo 3. Özellik Kapsülü Katmanı Parametreleri

Özellik Kapsülü Katmanı Parametreleri	
Kapsül Boyutu	16
Çekirdek Boyutu	5
Kaydırma Adımı	2
Doldurma	Valid

Evrişim katmanı ile 56x56 boyutlarına indirgenen görüntünün özellik kapsülü katmanının kaydırma adımı 2 olan parametresi ile boyutları 28x28 (2’ye bölümü) boyutlarına düşürülmüştür. Bununla birlikte çekirdek boyutu 5 alınarak (çekirdek boyutu - 1 = 4 / 2 = 2) görüntü 2 piksel daha kaybedip bu katman çıkışında 26x26 boyutuna gelmiştir. Ayrıca bu katmanda test edilerek model için en optimum değer olarak kabul edilen 0.3 değerinde Dropout işlemi uygulanmıştır.

Daha sonra bu görüntüleri 16 boyutlu vektörler haline getirmek için bir yeniden boyutlandırma işlemi uygulanarak görüntüler vektör boyutuna indirgenmiştir (26x26x16x16).

Özellik kapsül katmanı, digit kapsül katmanına tamamen bağlı olduğundan, çıkış vektörlerinin ezme (squash) aktivasyon fonksiyonundan geçirilmesi gerekir. Bu aşamada ezme fonksiyonuna, görüntüyü tekrar oluşturma kaybı olan küçük bir epsilon değeri eklenmelidir. Model için bu değer 0.005 olarak seçilmiştir. Ezme fonksiyonunun çıktısı digit kapsül katmanına iletilir.

3.3.4. Digit Kapsül Katmanı

Bu katmanındaki kapsül sayısı, tahmin edilmesi gereken sınıfların sayısı (5) ile belirlenir. Ayrıca bu katmanda bulunan kapsüller, özellik yönelimlerine dayalı olarak sınıflar hakkında tahminlerde bulunacaktır. Ancak bu tahminlerin öncül kapsüller tarafından yapılan tahminlere uyması gerekir ve bu yöntem, dinamik yönlendirme algoritması ile sağlanır.

Bu katmanda giriş, özellik kapsül katmanının çıktısı olan (26x26x16x16) 16 boyutlu vektörlerdir. Kapsülün iç işleyişine göre, giriş vektörlerinin her biri, 16 boyutlu girdi uzayını 32

boyutlu kapsül çıktı uzayına eşleyen kendi 16x32 boyutlu ağırlık matrisini alır. Dolayısıyla, bu katmanda (26x26x16x16x32x5) 27.688.960 parametre bulunmaktadır. Modelin katmanları ve parametreleri Tablo 4’de gösterildiği gibidir.

3.3.5. Kod Çözücü

Kod çözücü, 32 boyutlu bir vektörü doğru digit kapsülden alır ve onu etiketli bir görüntüye dönüştürmeyi öğrenir. Kod çözücü bir düzenleyici olarak kullanılır, doğru digit kapsülün çıktısını girdi olarak alır ve 64’e 64 piksellik bir görüntüyü yeniden oluşturmayı amaçlar.

Kayıp işlevi, yeniden yapılandırılmış görüntü ile giriş görüntüsü arasındaki Öklid mesafesidir. Kod çözücü; kapsülleri, orijinal görüntüyü yeniden oluşturmak için yararlı olan özellikleri öğrenmeye zorlar. Yeniden yapılandırılmış görüntü, giriş görüntüsüne ne kadar yakınsa o kadar iyidir. Bu katmanda da 66.879 parametre kullanılmaktadır.

Sonuç olarak bu aşamada kullanılan toplam parametre sayısı 29.456.959’dır. Toplanan yüz görüntüsü verileri oluşturulan modelde kullanılarak eğitim, doğrulama ve test işlemleri gerçekleştirilmiştir.

Tablo 4. Model katmanları ve parametreleri

Katman	Giriş	Parametre	Çıkış
Giriş Katmanı	64x64 resim	0	64x64x3
Evrişim Katmanı	64x64 resim	62464	56x56x256
Özellik Kapsülü Katmanı Evrişim	56x56x256 tensor	1638656	26x26x256
Özellik Kapsül Katmanı Yeniden Boyutlandırma	26x26x 256 tensor	0	10816x16
Digit Kapsül Katmanı	26x26x16x16 tensor	27688960	32x5
Kod Çözücü	32x5 matris	66879	64x64

4. Sonuçlar

4.1. Evrişimli Sinir Ağı Modelleri Sonuçları

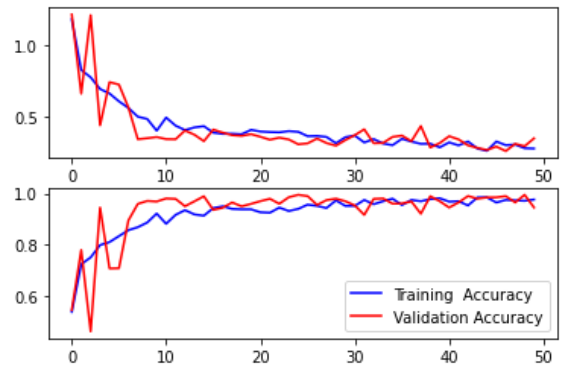
Hazırlanan yüz görüntüleri veri seti, 4 farklı evrişimli sinir ağı modeli ile eğitim işleminden geçirilerek, her biri için test işlemi başarımları sonuçları alınmıştır.

Modellerin her biri için eğitim işlemi 50 adımda (epochs) gerçekleştirilirken yalnızca ResNet50 modeli 100 adımda eğitim işleminden geçirilmiştir. Yapılan uygulamadan elde edilen sonuçlar Tablo 5’de gösterildiği gibidir.

Tablo 5. Evrişimli sinir ağı modelleri sonuçları

Model	Başarımları
AlexNet	%94
VGG16	%91
VGG19	%93.5
ResNet50	%75

Modeller arasında en iyi sonucu veren AlexNet modelinin eğitim ve doğrulama işlemleri üzerindeki kayıp ve başarımları grafiği Şekil 7’de gösterildiği gibidir.

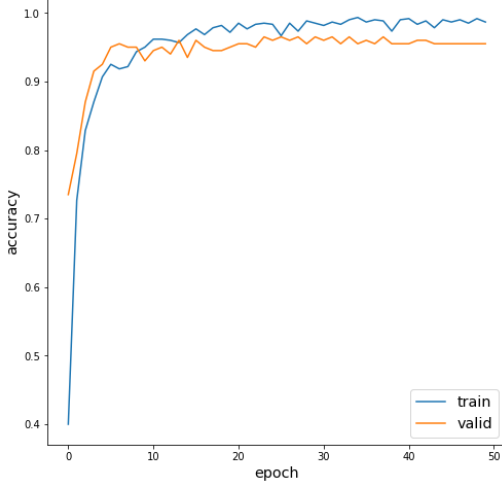


Şekil 7. AlexNet modeli eğitim ve doğrulama grafiği (Figure 7. AlexNet model training and validation graph)

Diğer modellere görece AlexNet modelinin hazırlanan veri kümesi üzerindeki yüksek sınıflandırma başarımı, ağın derinliği arttıkça başarımın her zaman artmadığını göstermektedir.

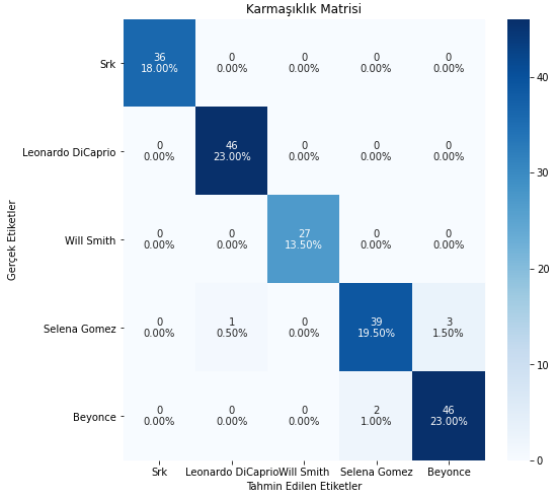
4.2. Kapsül Ağı Model Sonuçları

Kapsül ağlar ile hazırlanıp eğitilen bu modelde test işlemi %97 başarımla sonuçlanmıştır. Modelin eğitim ve doğrulama süreci grafiği Şekil 8'de gösterildiği gibidir.



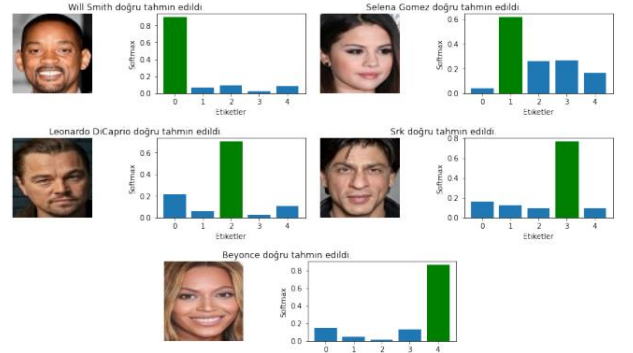
Şekil 8. Kapsül ağlar modeli eğitim ve doğrulama grafiği (Figure 8. Capsule networks model training and validation graph)

Bu başarımların değerleri modelin çok kez çalıştırılmasıyla elde edilen 5 en iyi sonucun ortalaması olan değerlerdir. Modelin sonucunda ortaya çıkan karmaşıklık matrisi Şekil 9'da gösterildiği gibidir.



Şekil 9. Kapsül ağlar modeli karmaşıklık matrisi (Figure 9. Capsule networks model complexity matrix)

Modelin, kullanılan yüz verilerinden bazılarının sınıflarını tahmin etmesine dair bir örnek Şekil 10'daki gibidir.



Şekil 10. Kapsül ağlar modeli tahminleme örnekleri (Figure 10. Capsule networks model estimation examples)

Modelin kullanılan veri seti üzerinde sınıflandırma işlemini başarılı bir şekilde gerçekleştirdiği gözlemlenmiştir.

5. Sonuç ve Öneriler

Yapılan uygulamalar sonucunda kapsül ağlar ile hazırlanan modelin test doğruluğu evrişimli sinir ağları modellerine kıyasla daha yüksek sonuçlanmıştır. Bunun nedeninin evrişimli sinir ağlarında bulunan havuzlama katmanının verilerde öznitelik kaybına neden olması olduğu tahmin edilmektedir. Buna bağlı olarak kapsül ağların yüz verilerini sınıflandırma problemi üzerinde başarılı sonuçlar verebileceği gözlemlenmiştir.

Tablo 6. Modellerin eğitim işlemi tamamlama süreleri

Model	Eğitim Süresi
Kapsül Ağlar	13 dakika 44 saniye
AlexNet	2 dakika 47 saniye
VGG16	1 dakika 25 saniye
VGG19	1 dakika 29 saniye
ResNet50	6 dakika 50 saniye

Buna karşılık Tablo 6'da verilen modellerin eğitim işlemini tamamlama sürelerine bakıldığında ise kapsül ağlar modelinin diğer modellere göre daha uzun sürede bu işlemi gerçekleştirdiği görülmektedir. Modellerin eğitim tamamlama sürelerinden yola çıkılarak daha büyük veri kümelerinde kapsül ağlar modeli ile yapılacak eğitim işlemlerinin daha maliyetli olacağı göz önünde bulundurulabilir.

Modelde evrişim katmanlarının sayısı veya hiperparametreleri değiştirilerek daha iyi sonuçlar elde edilebilir. Ayrıca modelde özniteliklerin çıkarılması aşamasında farklı teknikler kullanılarak sonuçların incelenmesi, kapsül ağların gelişimi için önemli bir adım olacaktır.

Çalışmada yüz verilerini sınıflandırma işlemi küçük bir veri seti üzerinde incelenmiştir. Söz konusu veri setini kullanarak dahi kapsül ağlar başarılı bir sınıflandırma işlemi gerçekleştirmiştir. Gelecekte farklı boyut ve çeşitlikte veri setleri üzerinde kapsül ağlar ile oluşturulmuş bir model deneyerek daha gelişmiş bir analiz yapılabilir.

6. Teşekkür

Bu çalışma, Fırat Üniversitesi MF.21.54 numaralı Bilimsel Araştırma Projesi kapsamında desteklenmiştir.

Kaynaklar

- Beser, F., Kizrak, M. A., Bolat, B., Yildirim, T. (2018). Kapsül Ağları ile İşaret Dili Tanıma Recognition of Sign Language using Capsule Networks. *26th Signal Processing and Communications Applications Conference (SIU)*, 1-4. doi:10.1109/SIU.2018.8404385.
- Doğan, F. ve Türkoğlu, İ. (2018). Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması. *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- İnik, Ö., Ülker, E. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- Kınlı, F., Kırac, F. (2020). FashionCapsNet: Clothing Classification with Capsule Networks. *Bilişim Teknolojileri Dergisi*, 13(1), 87-96. doi:10.17671/gazibtd.580222.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25.
- Özyurt, F., Tuncer, T., Avci, E., Koç, M., Serhatlioğlu, İ. (2019). A Novel Liver Image Classification Method Using Perceptual Hash-Based Convolutional Neural Network. *Arab J Sci Eng*, 44(4), 3173–3182. doi: 10.1007/s13369-018-3454-1.
- Öztürk, Ş., & Özkaya, U. (2020). Skin lesion segmentation with improved convolutional neural network. *Journal of digital imaging*, 33(4), 958-970. doi: 10.1007/s10278-020-00343-z.
- Sabour, S., Frosst, N. and Hinton, G. E. (2017). Dynamic Routing Between Capsules. *arXiv preprint*, arXiv:1710.09829.
- Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. arXiv:1409.1556.
- Toğaçar, M., Ergen, B., Cömert, Z., Özyurt, F. (2020). A Deep Feature Learning Model for Pneumonia Detection Applying a Combination of mRMR Feature Selection and Machine Learning Models. *IRBM*, 41(4), 212-222. doi: 10.1016/j.irbm.2019.10.006.
- Toraman, S. (2020). Kapsül Ağları Kullanılarak EEG Sinyallerinin Sınıflandırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 32(1), 203-209. doi: 10.35234/fumbd.661955.