

Bilgisayarlı Görü Yardımıyla Amerikan İşaret Dili Harflerinin Sınıflandırılması

Kubilay Karakayalı

Uluslararası Bilgisayar Enstitüsü

Ege Üniversitesi

Bornova, 35100, İzmir-Türkiye

karakayalikubilai@gmail.com

Özet

İşaret dili işitme engelli insanlarla iletişimde kullanılan en önemli yöntemdir. Ancak bu dilin bilinirliği çok fazla değildir. Sadece çevresinde ya da bir yakını işitme engelli olan insanlar bu dili öğrenmekte. Bu da insanlar arası iletişimde bazı sorunlara yol açabiliyor.

Bu projede, içinde her harfin görüntüsünün ve etiketinin bulunduğu veri seti yardımıyla eğittiğimiz evrimsel sinir ağıımızı kullanarak, kamera yardımıyla anlık aldığımız ve gerekli ön işleme işlemlerinden geçirdiğimiz görüntüyü sınıflandırıyoruz. Bu sınıflandırma sonucunu görüntü ekranının köşesine anlık olarak yazıp, hızlı bir şekilde sonucun gösterilmesini sağlıyoruz.

Anahtar Kelimeler – Bilgisayarlı Görü, Görüntü Sınıflandırma, Evrimsel Sinir Ağları, İşaret Dili.

1. GİRİŞ

İnsanlar arasından iletişimin büyük bir kısmı konuşma ile yapılmaktadır. Bu yüzden konuşmanın insan hayatında önemi çok büyüktür. Ancak her insan aynı dili konuşmadığı gibi, her insan aynı fiziksel özelliklere (konuşabilme, duyabilme vb.) sahip olmamaktadır. Bu da insanlar arasında iletişim konusunda zorluklar yaratmaktadır. Bu zorlukları yaşayanlardan olan işitme ve duyma engelliler, iletişim sağlamak için işaret dilini kullanmaktadır. Ancak işaret dili, herkes tarafından bilinen bir dil değildir. Bu yüzden bu engellere sahip insanlar, bu dili bilmeyen insanlarla iletişim kurmakta zorluk geçmektedir.

Bu zorluğu aşmak için, derin öğrenmenin önemli konularından biri olan evrimsel sinir ağını kullanan bir sınıflandırıcı kullandım. Evrimsel sinir ağları özellikle görüntü üzerinde sınıflandırma işlemleri yaparken çok iyi sonuçlar elde edebilmektedir. Ayrıca görüntünün açısı, ışığın şiddeti gibi değişkenlerden, diğer derin öğrenme modellerine kıyasla daha az etkilendiği için bu problemi çözmede diğer metotlara göre daha iyi sonuçlar çıkarmaktadır.

Ağın eğitim sonrası, kamera yardımı ile gönderilen görüntüler önce küçük bir görüntü işleme işlemlerinden geçmektedir. Bu işlemler görüntüyü ağın tanıma işlemine sokmak için gerekli morfolojik işlemlerdir. Ayrıca görüntüyü

gri tonlamalı görüntüye dönüştürüp ağı sınıflandırma işlemini hızlandırma gibi işlemleri de bu adımda gerçekleştirmekteyiz.

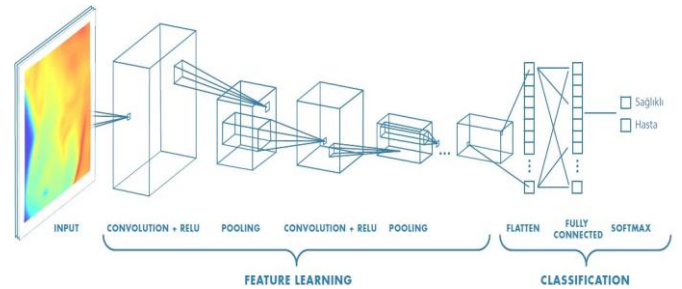
2. EVRİŞİMSSEL SİNİR AĞLARI

Son zamanlarda en popüler derin öğrenme modellerinden biri olan evrimsel sinir ağları, özellikle ayırım ve görüntü sınıflandırma problemlerinde diğer modellere göre üstünlük sağlamaktadır.

Evrimsel sinir ağları, minimum ön işlem gerektirecek şekilde tasarlanmış çok katmanlı algılayıcıların bir varyasyonunu kullanır[1]. Ayrıca ortak ağırlıkları mimarisi ve çeviri değişmezliği özelliklerine bağlı olarak yer değişmez veya alan değişmez yapay sinir ağları (SIANN) olarak da bilinir[2].

Evrimsel sinir ağları, nöronlar arasındaki bağlanma düzeninin hayvan görsel korteksinin organizasyonuna benzer olması nedeniyle biyolojik işlemlerden ilham almıştır. Bireysel kortikal nöronlar, uyarıcılara sadece alıcı alan olarak bilinen görsel alanın sınırlı bir bölgesinde yanıt verir. Farklı nöronların alıcı alanları kısmen tüm görsel alanı kaplayacak şekilde örtüşür [3].

Evrimsel sinir ağları, diğer görüntü sınıflandırma algoritmalarına kıyasla nispeten az ön işleme kullanır. Bu, ağı, geleneksel algoritmalarda el yapımı olan filtreleri öğrendiği anlamına gelir. Öncelikli bilgilerden ve özellik tasarımıındaki insani çabalardan bağımsızlık, diğer modellere kıyasla büyük bir avantajdır.

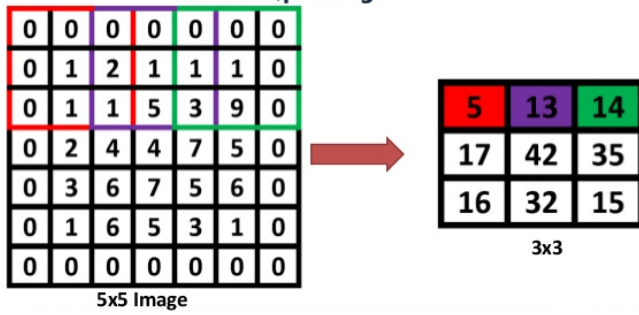


Şekil 1: Evrimsel Sinir Ağına Tasviri

Evrişimli sinir ağlarında, ağ görüntüyü çeşitli katmanlarda işler. Bu katmanlara genel bir bakış yapacak olursak:

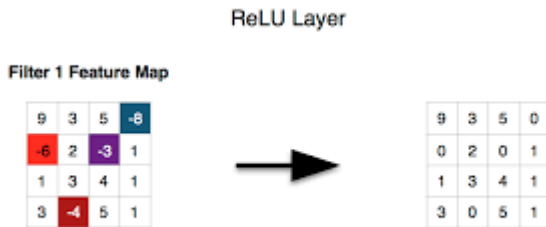
- 1- Convolutional Layer: Özellikleri saptamak için kullanılır.
- 2- Non-Linearty Layer: Sisteme doğrusal olmayanlığın işlenmesi
- 3- Pooling Layer: Ağırlık sayısını azaltır. Uygunluğu kontrol eder.
- 4- Flatten Layer: Klasik sinir ağı için verileri hazırlar.
- 5- Fully-Connected Layer: Sınıflandırmada kullanılan sinir ağı

Evrişim(Convolutional) Katmanı: Evrişimsel katmanın ana görevi, önceki katmandaki özelliklerin yerel bağlantılarını tespit etmek ve görünümünü bir özellik haritasına eşlemektir. Nöral ağlardaki evrişim sonucu, görüntü algılayıcılara bölünür, yerel alıcı alanlar yaratır ve son olarak algılayıcıları m2 x m3 boyutunda özellik haritalarına sıkıştırır. Bu nedenle, bu harita, özelliğin görüntüde nerede oluştuğunu ve filtreye ne kadar iyi karşılık geldiğini gösterir. Bu nedenle, her bir filtre, uygulandığı hacimdeki pozisyona ilişkin olarak uzaysal olarak eğitilir[4].



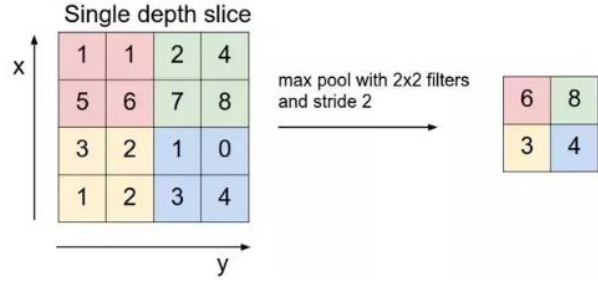
Şekil 2: Evrişim (Convolutional) Katmanı

Doğrusal Olmayanlık (Non-Linearty) Katmanı: Bir evrişimsel sinir ağındaki doğrusal olmayan bir kat, evrişim katmanı tarafından oluşturulan özellik haritasını alan ve aktivasyon haritasını çıktı olarak oluşturan bir aktivasyon fonksiyonundan oluşur. Aktivasyon fonksiyonu, giriş hacmi üzerinden eleman şeklinde bir işlemdir ve bu nedenle giriş ve çıkış boyutları aynıdır[6].



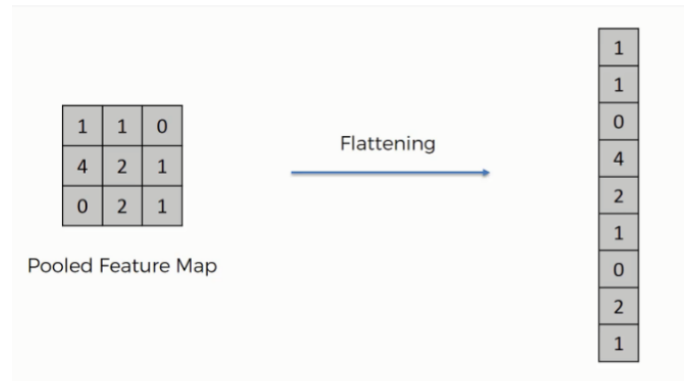
Şekil 3: Doğrusal Olmayanlık (Non-Linearty) Katmanı

Örnekleme (Pooling) Katmanı: Bu katman, evrişimsel sinir ağlarında ardışık evrişim katmanları arasında sıklıkla eklenen bir katmandır. Bu katmanın görevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını azaltmak içindir. Bu sayede ağıdaki uyumsuzluk kontrol edilmiş olur. Birçok Pooling işlemleri vardır, fakat en popülerleri max pooling'dir. Yine aynı prensipte çalışan average pooling, ve L2-norm pooling algoritmaları da vardır [5].



Şekil 4: Örnekleme (Pooling) Katmanı

Düzleme (Flatten) Katmanı: Bu katmanın görevi basitçe, son ve en önemli katman olan Fully Connected Layer'ın girişindeki verileri hazırlamaktır. Genel olarak, sinir ağları, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrixlerin tek boyutlu diziye çevrilmiş halidir.

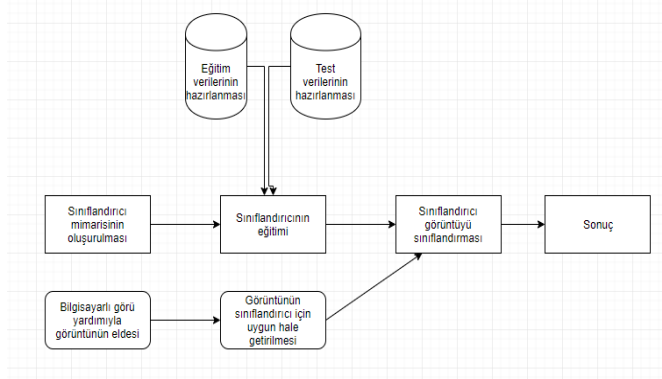


Şekil 5: Düzleme(Flatten) Katmanı

Tam Bağlı(Fully-Connected) Katman: Bu katman evrişimsel sinir ağı son ve en önemli katmanıdır. Verileri düzleştirme işleminden alır ve sinir ağı yoluyla öğrenme işlemini gerçekleştirir. Bu katmanda yapılan işlem geleneksel sinir ağları ile aynıdır. Önce gelen eğitim verileri ile ağı ağırlıklarının ayarlanamsı işlemi yapılır. Sonra test ve doğrulama işlemleri gerçekleştirilir. Bu işlemlerden sonra ağı sınıflandırma yapmaya hazır olur.

3. DENEYSEL ÇALIŞMA

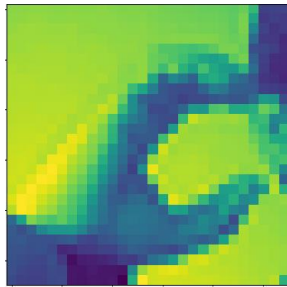
Deneyisel çalışma kısmını iki parçada inceleyebiliriz. Birinci adımda, gönderilen el işareti görüntülerini sınıflandıracak sınıflandırıcının tasarımı, eğitilmesi ve verilerinin hazırlanması, ikinci adımda ise bilgisayarlı görü yardımı ile elde ettiğimiz görüntülerin, sınıflandırıcının tahmin edebilmesi için gereken ön işlemlerden geçirilmesi şeklinde belirtebiliriz.



Şekil 6: Projenin Çalışma Sistemi

3.1. Veri Seti

Sınıflandırıcı eğitmek ve test etmek için kullanılan veri seti, her bir satırda etiket ve 784 adet piksel verisi olmak üzere 785 sütundan oluşmaktadır. Öncelikle alınan veriden etiket ve piksel verileri ayrılır. Sonra ayrılan piksel verileri kullanarak 28x28 piksel görüntüye dönüştürülmektedir.



Şekil 7: Veri setinden örnek bir görüntü

Bu görüntülerin piksel değerleri (0,255) arası değişmektedir. Ancak bu bizim ağımda tahminleme ve öğrenme işlemlerini yaparken bazı sorunlar çıkarmaktadır. Bu yüzden piksel değerlerinin normalizasyon işlemine sokarak (0,1) arasında değerler almasını sağladık. Bu sayede değer arasındaki farktan dolayı yaşanan sorunların önüne geçmiş oluyoruz.

Etiket verilerimiz her ne kadar sayısal değerler olsa da aslında kategorik verilerdir. 0-24 arası değer Amerikan işaret dilinin harflerine karşılık gelir(J ve Z harfleri hareket gerektirdiği için veri setinde bulunmamaktadır). Bu yüzden bu verileri sınıflandırıcıya uygun şekilde hazırlamak gerekmektedir. Bunun yapabilmek için “sci-kit learn” framework’ünün LabelBinarizer sınıfını kullandım. Bu sayede sayısal etiketimizi kategorik hale dönüştürdüm.

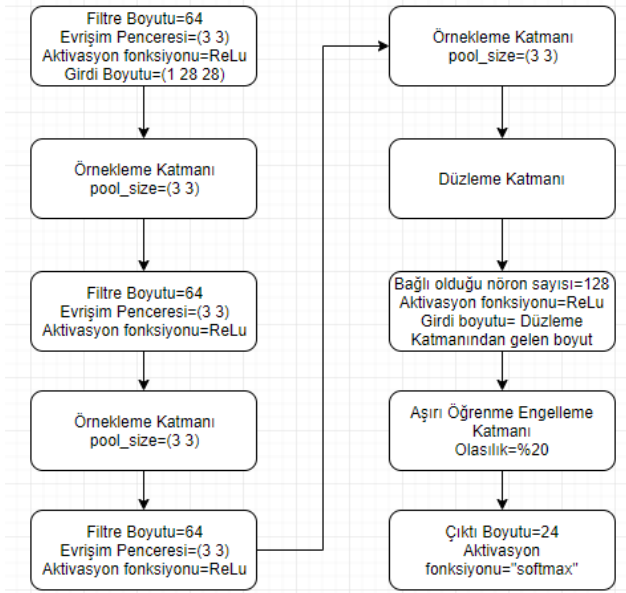
Veri seti	Veri Sayısı
Eğitim	27455
Test	7172

Tablo 1: Veri setlerinde bulunan veri sayısı

3.2. Sınıflandırıcı

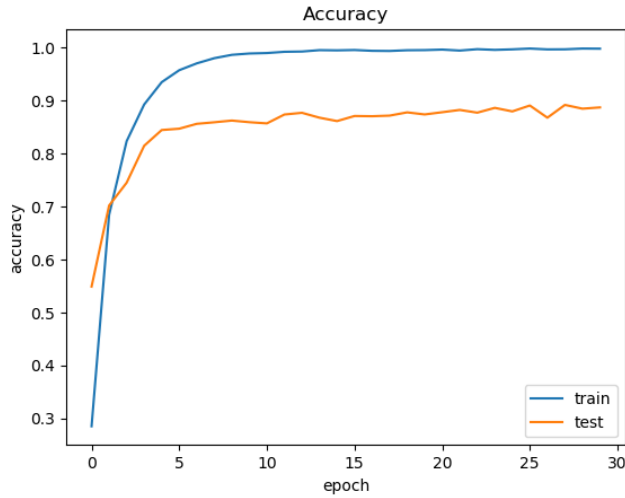
Bu projede sınıflandırıcı olarak evrişimsel sinir ağı kullanılmıştır. Evrişimsel sinir ağını oluştururken açık kaynaklı olan Keras framework kullanılmıştır. Oluştural evrişimsel sinir ağı mimarisinde, ilk evrişim katmanı, girdi boyutu olarak gönderilen görüntünün boyutunu (1,28,28) alır. Parametreleri ise Filtre boyutu 64, evrişim penceresinin boyutu (3x3) ve aktivasyon fonksiyonu olarak “ReLu” kullanır. İkinci ve üçüncü evrişim katmanları da evrişim penceresi boyutu (3x3) ve aktivasyon fonksiyonu olarak “ReLu” argümanları alır. Her evrişim katmanı arasında bir adet örnekleme katmanı bulunmaktadır. Bu örnekleme katmanları, argüman olarak pool_size=(2,2) alır.

Evrişim ve örnekleme katmanları sonunda düzleme katmanı işe veriler son sınıflandırıcı ağı için hazırlanır. Bu ağı girdisinin boyutu düzleme katmanından çıkan verinin boyutu ile aynıdır. İlk katman aktivasyon fonksiyonu olarak “ReLu” fonksiyonunu alır ve 128 adet nörona bağlı olarak verileri gönderir. Aşırı öğrenme olayının önüne geçmek için %20 şansla gelen girdilerin değerini sıfırlayan bir katman bulunmaktadır. En son çıktı katmanında aktivasyon fonksiyonu olarak “softmax” kullanılmıştır. Bu katmanda çıktı olarak 24 sınıfa ait olma yüzdeleri verilmektedir.



Şekil 8: Sınıflandırıcı Mimarisi

Eğitim işleminde, epoch sayısını 30, batch boyutunu 128 olarak belirlendi. Ölçme metriğini “accuracy” olarak belirlendi. Kayıp fonksiyonunu kategorik etiketleri ile çalıştığımız için “categorical_crossentropy” seçtim. Optimizer olarak Keras framework’unun “adam” optimizer’ını kullandım.

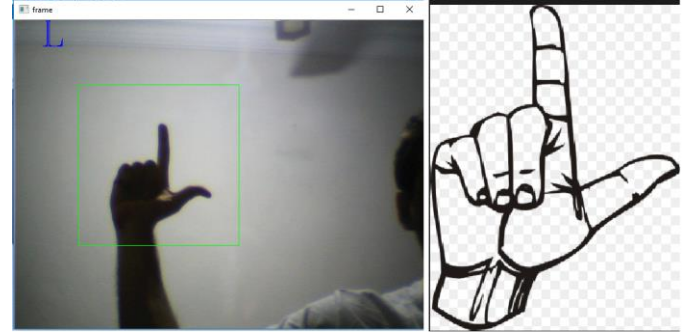


Şekil 9: Her epoch için eğitim ve test accuracy doğrusu

3.3. Bilgisayarlı Görü

Bu aşamada, kamera ve Python’un OpenCv kütüphanesi yardımı ile elde ettiğim görüntüyü, sınıflandırıcının tanıması için gerekli ön işlem adımlarından geçirdim. Canlı görüntüde (250 250) boyutlarında olan yeşil renkli karenin içinde yaptığımız el hareketi görüntüsü, öncelikle gri tonlamalı görüntüye dönüştürüldü. Bu sayede veri setindeki görüntüler gibi görüntümüzün piksellerini normalize ettim. Sonra OpenCv

yardımla “inter cubic” interpolasyon yöntemini kullanarak görüntüyü 28x28 piksel boyutuna getirdim. Sonra görüntüyü sınıflandırıcıya yolluyarak gelen sonucu anlık olarak canlı görüntünün üzerine yazdım.



Şekil 10: L harfi için gösterim

IV. SONUÇ

İletişim zorluğunun çözülmesi, insanların hayatında bir çok şeyi iyileştirmektedir. Bu yüzden insan hayatında oldukça önemli bir yere sahiptir. Bu nedenle sınıflandırıcıyı eğitmek için kullanılan görüntü veri setinin kalitesi ve görüntü almak kullanılan kameranın kalitesi gibi değişkenleri sınıflandırıcımızın gücünü artırıp, iyi sonuçlar elde etme konusunda büyük etkiye sahiptir.

Literatürde birçok yöntem bulunmasına karşın, araştırmacıların yeni yöntem ve yaklaşım arayışları devam etmektedir. Bu arayışlar neticesinde evrişimli sinir ağları, rastgele orman, destek vektör makineleri, bulanık mantık gibi bir çok yöntem öne sürülmüştür. Evrişimsel sinir ağları, son zamanlarda derin öğrenme furçasının öne çıkması ile diğer sınıflandırma yöntemlerine göre daha çok öne çıkmaktadır.

Bu çalışmada, sınıflandırma yöntemi olarak evrişimsel sinir ağları seçilmesinin iki nedeni bulunmaktadır. Birinci evrişimsel sinir ağlarının gücü, özellikle görüntü sınıflandırma işleminde evrişimli sinir ağlarının gücü diğer sınıflandırma yöntemlerine göre daha iyidir. Bunun en büyük nedeni ise evrişimsel sinir ağları görüntüdeki ufak farklılıklardan diğer sınıflandırıcı yöntemlerine kıyasla daha az etkilenmektedir. İkinci neden ise daha hızlı çalışması, özellikle görüntü sınıflandırma uygulamalarında, sınıflandırıcıya gönderilen görüntüler bir çok ön işleme adımında geçmek zorundadır. Çünkü görüntüde küçük değişiklikler bile sınıflandırıcının verdiği sonuçları değiştirmektedir. Ama evrişimsel sinir ağları bu durumdan daha az etkilendiği için görüntünün çok fazla ön işleme adımına girmesine gerek yoktur. Bu da sistemimizi oldukça hızlandırmaktadır.

KAYNAKLAR

- [1] LeCun, Yann. "LeNet-5, convolutional neural networks". 16 Ekim 2013.
- [2] Zhang, Wei, "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of annual conference of the Japan Society of Applied Physics, 1988
- [3] Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network" ,2003
- [4] <https://wiki.tum.de/display/lfdv/Convolutional+Neural+Networks> Eriřim Tarihi: 02.01.2019 21:30
- [5] <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>. Eriřim Tarihi: 02.01.2019 20:00
- [6] <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> Eriřim Tarihi: 02.01.2019 22:00