



**KARADENİZ TEKNİK ÜNİVERSİTESİ OF TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**2019 – 2020 BAHAR DÖNEMİ
YZM-4008 VERİ MADENCİLİĞİ DERSİ**

1. VERİ MADENCİLİĞİ

1.1. Genel Tanım

Bilgi teknolojilerinin hızlı gelişmesi sonucu veri hacmi, çeşidi ve karmaşıklığı hızlı büyümekte ve heterojen yapıli bilginin otomatik olarak analiz edilmesine daha fazla ihtiyaç duyulmaktadır. Bu nedenle, veri madenciliğı yöntemlerine olan ilgi artmakta ve gün geçtikçe daha fazla kullanılmaktadır. Bu yöntemlerin esas amacı, veri içindeki gizli bağıntıların ortaya çıkarılmasıdır. Veri madenciliğı, büyük veri kümelerinin içinden işe yarayacak bilgi keşfinin analiz sürecini içeren bilgisayar biliminin disiplinler arası bir alt alanını oluşturmaktadır. Bu gelişmiş analiz sürecinin amacı, bir veri kümesinden bilgi çıkarmak ve çıkardığı bu bilgiyi daha ileri bir kullanım için anlaşılabilir bir yapıya dönüştürmektir.

Son yıllarda bilgisayarların ve veri depolama sistemlerinin gelişmesiyle birlikte büyük miktardaki verilere kolaylıkla ulaşılabilir. Veri madenciliğı bu büyük miktarlardaki verilerin işlenip, analiz edilip bilgiye çevrilmesi olarak tanımlanabilir.

Veri madenciliğı; elde bulunan ham verinin tek başına sunamadığı bilgiyi ortaya çıkaran veri analizi sürecidir.

Veri madenciliğı; veri tabanlarında, veri ambarlarında veya diğer veri depolarındaki büyük miktarlardaki veri içindeki ilginç bilgileri keşfetme işidir.

1.2. Veri Madenciliğı Ön İşlemler

1.2.1. Veri temizleme; gürültülü ve tutarsız verileri çıkarmak

Veri madenciliğı yapılacak ham veri genellikle üzerinde analiz yapılabilecek kadar uygun değildir. Veri kümesi eksik veriler veya analize uygun olmayan tutarsız verileri içerebilmektedir. Bu tutarsız ve hatalı veriler gürültülü olarak değerlendirilmektedir. Böyle durumlarda analizin sağlıklı yapılabilmesi için veri kümesinin söz konusu sorunlardan temizlenmesi gerekecektir. Bu durumda; eksik veri içeren kayıtların silinmesi, değişkenin tüm verileri kullanarak ortalamasının hesaplanması ve eksik değer yerine bu değerlin konması veya eksik değerin tahmin edici algoritmalarla birinin kullanılmasıyla tahmin edilmesi gibi yöntemler kullanılabilir.

1.2.2. Veri bütünleştirme, birçok veri kaynağını birleştirmek

Veri madenciliğı süreci büyük veri kümeleri üzerine uygulanmaktadır. Ancak büyük veri kümeleri her zaman direkt ulaşılabilir değildir. Bazen veriler dağınık olabilmekte ve bu verilerin birleştirilmesi gerekebilir. Böyle veri tabanlarının birleştirilmesi sürecinde karşılaşılabilecek temel sorun, farklı türdeki verilerin tek türe dönüştürülmesi yani bütünleştirilmesi işlemidir. Aynı türde veriler farklı veri tabanlarında farklı türlerde tutulabilmektedir. Verilerin birleştirilmesi sırasında ise; eğer bütünleştirme yapılmadıysa, aynı veri kümesinde aynı şeyi ifade eden değişkenler sanki farklı değişkenlermiş gibi olabilmekte, bu durum da sağlıklı bir veri madenciliğı yapılmasını engellemektedir. Veri madenciliğı yapacağımız veri kümesini iyi tanımak bu tür büyük hataların yapılmasını engelleyecektir.

1.2.3. Veri İndirgeme Veri seçme, yapılacak olan analiz ile ilgili olan verileri belirlemek

Veri madenciliğinde bazen analiz işlemleri uzun sürebilir. Sonuçların değişmemesi kaydıyla veri sayısı ya da değişkenlerin sayısı azaltılabilir. Buna veri madenciliğinde veri indirgeme denmektedir. Ancak burada dikkat edilmesi gereken indirgeme işlemi sırasında analizde önemli sayılabilecek değişkenlerin veya verilerin veri tabanından çıkartılmamasıdır. Bu durumda amaçtan sapma söz konusu olacak ve sonuçlar istenildiğı kadar sağlıklı olmayacaktır. Veri indirgeme çeşitli biçimlerde yapılabilir. Bunlar; örnekleme, sıkıştırma, boyut indirgeme veya genelleme gibi biçimlerde yapılabilir.

1.2.4. Veri dönüşümü, verinin veri madenciliğı teknikleriyle kullanılabilir hale dönüşümünü gerçekleştirmek

Veri madenciliğı sürecinde her zaman veri, değişkenlerin aynı değerleri kullanılarak analize katılmaz. Bir dönüşüm yöntemi kullanarak değişkenlerin normalleştirilmesi veya standartlaştırılması gerekebilir. Bu durumun oluşmasına sebep olan durumlardan birkaçı aşağıda verilmiştir:

1.2.5. Veri madenciliğı, seçilen algoritmanın veri kümesi üzerine uygulanması

Veri seti ön işlem süreci ile hazır hale geldikten sonra yapılacak olan, probleme uygun yöntemin ve bu doğrultuda algoritmanın seçilerek veri kümesi üzerinde uygulanmasıdır. Veri dönüşümü sürecinden sonra kullanılacak veri madenciliğı modeline karar verilmektedir. Bu modeller sınıflandırma, regresyon, kümeleme, aykırı değer analizi ve evrimsel analiz gibi tekniklerdir.

Veri kümesi üzerine uygulanan her algoritma farklı sonuç üretmektedir. Burada önemli olan probleme en uygun algoritmanın belirlenebilmesidir. Bu algoritmalar; sınıflandırma, kümeleme ve birliktelik kuralları teknikleri altında toplanmaktadır.

1.2.6. Sonuçları sunma ve değerlendirme

Veri madenciliği algoritmaları veri kümesi üzerine uygulandıktan sonra yapılacak olan işlem, sonuçların değerlendirilmesi ve bu doğrultuda sunulmasıdır. Değerlendirme ve sunum aşamasında sonuçlar grafiklerle desteklenebilir, kullanılan veri madenciliği platformuna göre elde edilen çıktılar sunum için kullanılabilir. Örneğin hiyerarşik kümeleme yöntemi uygulanmış ise sonuçlar dendrogram grafiği ile sunulmaktadır.

1.3. Veri Madenciliğinde Karşılaşılan Problemler

Veri madenciliğinde büyük boyutlu ve gerçek hayat verileriyle çalışılmaktadır. Bu yüzden VM’de birçok problemle başa çıkmak gerekmektedir. Bu problemler şu şekilde özetlenebilir:

1.3.1. Veri tabanı boyutu:

Veri tabanı boyutları teknolojinin gelişmesiyle büyük bir hızla artmaktadır. Hem özellik sayısı olarak sütun bazında hem de örneklem sayısı olarak satır bazında artmaktadır. Büyük bir veri tabanıyla çalışmak, ondan anlamlı bilgiler elde etmek oldukça zordur. Bununla başa çıkmak için veri kümesinin indirgenmesi veya kullanılan algoritmaların büyük boyutlarla başa çıkabilecek kadar iyi seçilmesi gibi yollara başvurulabilir.

1.3.2. Gürültülü veri:

Veri girişi sırasında oluşan hatalar gürültülü veriye sebep olur. Bu hatalar sonucunda birçok değer yanlış olabilir. Özellikle gerçek hayat verileri düşünüldüğünde bu oldukça kötü bir durumdur. Bundan dolayı gürültülü verilerin saptanması veya etkilerinin en aza indirilmesi önemli bir problemdir.

1.3.3. Boş veriler:

Veri tabanında bulunan boş değerlerdir. Bilinmeyen bir değeri ifade eder. Bu bilinmeyen değerleri çalışmaya dâhil etmek için çeşitli yöntemler kullanılabilir. Boş değerleri dikkate alamamak, boş değerlerin yerine en çok rastlanan değer yazılması, boş değerlerin yerine ortalama bir değer yazılması, boş değer yerine varsayılan başka bir değer konulması gibi çözümler mevcuttur.

1.3.4. Eksik veri:

Veri tabanında bulunmayan verilerdir. Bu veriler sütun bazlı olabilir. Yani kayıt tutulması gereken önemli bir özellik tutulmamış olabilir. Ayrıca satır bazlı da olabilir. Belirli bir veri kümesinin veri tabanında bulunmaması sonucu oluşan eksik verilerdir.

1.3.5. Artık veri:

Veri kümesindeki gereksiz yere tutulan özellikler artık veri olarak adlandırılır. Bu verilerin elenmesi için özellik seçimi algoritmaları kullanılabilir.

1.3.6. Dinamik veri:

İçeriği sürekli değişen veri tabanları dinamik veri içerir. İçeriğin sürekli değişmesi veri madenciliği uygulamalarında problemlere neden olabilir. Veri tabanı ve veri madenciliği uygulamaları aynı anda çalışır ve bundan dolayı da bazı problemler çıkabilir.

1.4. Veri Madenciliği Modelleri

1.4.1. Birliktelik analizi

Veri madenciliğinde sıkça kullanılan yöntemlerden birisidir. Sepet analizi de denir. Alışverişlerde müşterilerin satın alma alışkanlıklarını analiz edip hangi ürünleri birlikte aldıklarını belirler. Bu sayede müşterilerin daha fazla ürün satın alması amaçlanmaktadır.

Birlikte olma kurallarını belirli olasılıklarla ortaya koyarak olayların birlikte gerçekleşme durumlarını analiz eden veri madenciliği modeline birliktelik kuralları denmektedir. Birliktelik kuralları yani ilişki analizi, veri kümesindeki bir kaydın diğer kayıtlarla olan bağlantısını açıklayan işlemler dizisidir. Bir kayıt varken bunun yanında başka bir kaydın da var olma olasılığı nedir? Ya da bu iki kayıt varken diğer bir üçüncü veya dördüncü kaydında var olma olasılığı nedir? Şeklindeki sorulara yanıt aramakta ve ortaya çıkardığı bağlantıları birliktelik kuralları olarak tanımlamaktadır.

1.4.2. Kümeleme

Veriyi daha önceden belirlenmeyen kümelerle ayırma işlemine denir. Aynı kümedeki elemanlar benzer özelliklere sahip oldukları gibi farklı kümelerdeki elemanlar birbirlerine pek benzemezler. Sınıflandırmadan farklıdır. Sınıflandırmada sınıflar daha önceden belirlenmiş iken kümelemede kümeler belli değildir.

Kümeleme verilerin kendi aralarındaki benzerliklerinden hareket edilerek, gruplandırmasıdır. Kümeleme modelinde, sınıflandırmada olduğu gibi sınıf etiketleri belli değildir. Yani bir verinin hangi kümeye ait olduğu belli değildir. Aynı zamanda sınıfların sayısı da belli değildir. Bu tür öğrenmeye danışmansız (gözetimsiz) öğrenme denmektedir. Kümeleme algoritmalarının uygulanmasından sonra sınıflar oluşturulmaktadır. Modelin uygulanmasıyla birbirine en çok benzeyen veriler aynı küme içine konmaktadır. Kümeleme modelindeki amaç, dağınık halde bulunan verileri gruplandırmak ve kullanıcıya anlaşılabilir bir özet veri sunmaktır. Bu modelde; farklı grup içindeki nesneler olabildiğince birbirinden farklı, aynı grup içindeki nesneler ise olabildiğince benzer olacak şekilde oluşturulmaktadır.

Kümeleme için pek çok algoritma vardır. Genel olarak şu şekilde sıralanabilirler:

- Bölme yöntemleri
- Hiyerarşik yöntemler
- Yoğunluk tabanlı yöntemler
- Izgara tabanlı yöntemler
- Model tabanlı yöntemler

Kullanım amaçlarına ve kullanım alanlarına göre kümeleme analizinin amaçları:

- Doğru tiplerin belirlenmesi
- Model oluşturmak
- Gruplara dayalı tahmin
- Hipotez testi
- Veri araştırma
- Hipotez oluşturma
- Veri indirgeme

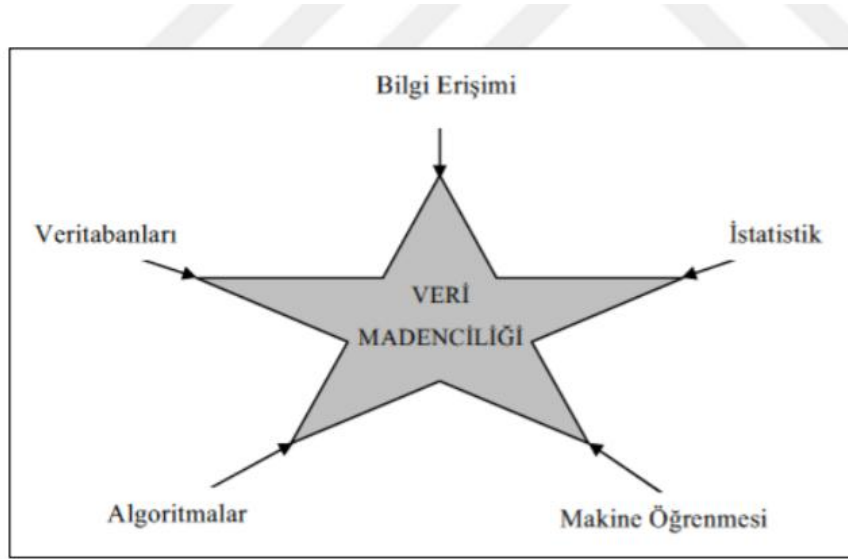
1.4.3. Sınıflandırma

Sınıflandırma modeli veri tabanındaki gizli örüntüleri ortaya çıkarmak için kullanılan bir modeldir. Bu model uygulanırken belli adımlar izlenmektedir. Sınıflandırmaya tabi tutulacak verilerin sınıf etiketleri ve sınıfların sayısı bellidir. Bu tür öğrenmeye danışmanlı (gözetimli) öğrenme denmektedir. Bu yöntemde; verilerin bir kısmı eğitim, bir kısmı ise test için kullanılmaktadır. Veri tabanındaki her sınıfın kendi içinde bir deseni vardır. Sınıflandırma modeli bu desenleri keşfederek, bir verinin hangi sınıfa ait olduğunu bulmaya çalışmaktadır. Eğitim kümesi bu desenleri öğrenmek için kullanılırken, test kümesi ise bu desenlerin ne kadar öğrenildiğini test etmek için kullanılmaktadır. Yapılan test sonucunda doğruluk oranı ne kadar yüksekse model doğrultusunda uygulanan algoritma o kadar başarılı olmuştur. Sınıflandırma modeli en çok bilinen veri madenciliği tekniklerinde birisidir. Resim, örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, kalite kontrol çalışmaları ve pazarlama konuları sınıflandırma modelinin kullanıldığı alanlardan bazılarıdır. Bu model, tahminleyicidir. Havanın sonraki gün durumunun ne olacağı ya da kutudaki mavi toplarının sayısının tahmin edilmesi, bir sınıflandırma işlemidir.

Sınıflandırma ve regresyon modellerinde kullanılan algoritmalar genellikle şunlardır:

- Karar ağaçları
- Yapay sinir ağları
- Evrimsel algoritmalar

- K en yakın komşu algoritması
- Bayes sınıflandırıcılar
- Doğrusal regresyon
- Doğrusal olmayan regresyon
- Sürü zekâsı teknikleri
- Kaba küme teorisi
- Bulanık küme



Şekil 1. Veri madenciliğinin ilişkili olduğu disiplinler

2. Genel Bilgilendirme

2.1. Evrişimsel Sinir Ağı

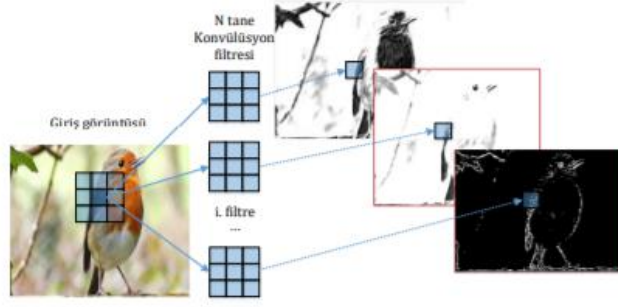
Derin öğrenme algoritmalarının başında çok katmanlı mimarisi ile Evrişimsel Sinir Ağı (ESA) gelmektedir. (2012 yılında yapılan ImageNet yarışmasında göstermiş olduğu başarımdan sonra birçok alanda ESA kullanılmıştır) ESA insanların algılama sistemini kullanarak öğrenen ve iyi bilinen bir derin öğrenme mimarisidir.

Geliştirilen derin evrişimsel ağ yapıları daha öncekilere göre daha fazla katmana sahip olmaktadır. Bu özellik giriş verisinin doğrusal olmayan özelliğini ortaya çıkarmakta ve ağ başarımını artırmaktadır. Fakat bu özellik aynı zamanda ağın karmaşıklığını arttırmakta ve eğitim sürecini uzatmaktadır.

Derin ESA'nın iyi başarımlar sağlaması için eğitim aşamasında büyük miktarda veri ile eğitilmesi gerekmektedir. Veri boyutunun büyümesi ve ESA'nın yapısındaki karmaşıklık eğitim sürecini klasik işlemcilerde yapılması önündeki en önemli engellerden birini oluşturmaktadır. Bu durum eğitim aşamasında paralel hesaplama yapan grafiksel işlemci kartlarının kullanılması gerekli kılınmıştır. Bu karta sahip bir bilgisayar üzerinde yapılan eğitim hem daha hızlı olmakta hem de gerçek zamanlı test işlemi kolaylıkla yapılabilmektedir.

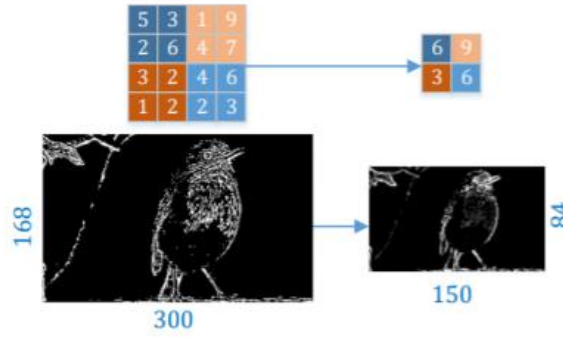
Bir ESA'nın temel bileşenleri evrişimler, havuzlama, aktivasyon fonksiyonu ve tam bağlı katmanıdır. Son bileşen olan tam bağlı ağ ileri beslemeli yapay sinir ağındaki gizli katmana benzer bir yapıya sahiptir.

- Evrişim katmanı görüntü üzerinde belirli filtrelerin kaydırılarak görüntünün yeni piksel değerlerinin komşu piksellere göre hesaplanması işlemidir. Şekil 2’de görülmekte



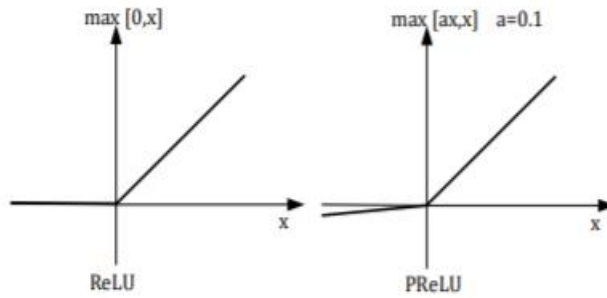
Şekil 2. Giriş görüntüsü üzerinde evrişim işlemi

- Havuzlama katmanı evrişim işleminden sonra görüntü boyutunu düşürmek için kullanılır. En çok kullanılan havuzlama işlemi maksimum havuzlamadır. Bu işlemin temel iki amacı veri boyutunu düşürmek ve görüntü boyutunu azaltmaktır. Şekil 4’te görülmektedir.



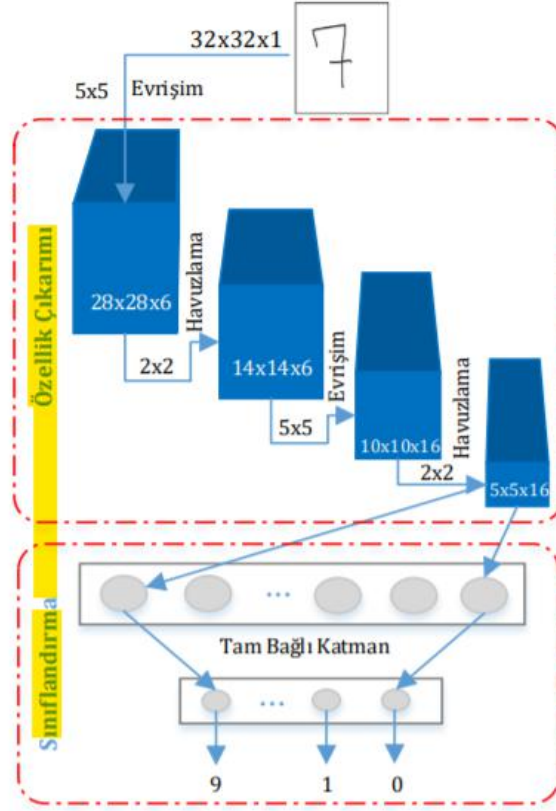
Şekil 3. Havuz katmanı

- İleri beslemeli yapay sinir ağlarında kullanılan tanjant, sigmoid ve doğrusal gibi aktivasyon fonksiyonları yerine ESA’da evrişim işleminden sonra düzeltilmiş bir doğrusal fonksiyon kullanılır. ReLU fonksiyonu görüntü işleme algoritmaları sonrasında elde edilen özellik haritasında negatif değerleri düzeltmek için kullanılır. PReLU fonksiyonu ise küçük negatif değerlere izin vermektedir. Şekil 3’te görülmektedir.



Şekil 4. ReLU ve PReLU aktivasyon fonksiyon gösterimi

Şekil 5’de verildiği gibi görüntü üzerinde gezdirilen evrişim filtreleri ile özellik haritaları oluşturulmaktadır. Havuzlama katmanında ise görüntü boyutu verilen alt örnekleme boyutuna göre düşürülmektedir. Tam bağlı katmanda ise en son aşamada vektör olarak elde edilen özelliklere göre görüntü sınıflandırması yapılmaktadır.



Şekil 5. Lenet-5 ağıının yapısı

2.2. Nvidia Jetson Tx2 GPU geliştirme

Nvidia Jetson Tx2 GPU geliştirme kartı Jetson TX2 gömülü sistemi NVIDIA tarafından üretilmiştir. Bu gömülü sistem önceden düzenlenip uyarlanmış olan bir Linux işletim sistemi kurulu bir şekilde gelmektedir. Yapay zekâ destekli sistemlerde düşük güç tüketimi ile yüksek hesaplama performansı gerektiren tüm işlemleri hızlı bir şekilde yerine getiren gömülü sistemdir. Geliştirme kartı NVIDIA’nın ücretsiz olarak sunduğu tüm geliştirme API’lerini (Application Programming Interface) ve yaygın olan diğer API’leri de desteklemektedir. Ayrıca geliştirme seti olduğu için çeşitli donanım ara yüzü sunarak, oldukça esnek ve genişletilebilir bir platforma olanak sağlamaktadır.

3. Araştırma Ödevi 1, Makale 1

3.1. İncelenen Tez Adı

Gömülü Derin Öğrenme ile Tehdit İçeren Nesnelerin Gerçek Zamanda Tespiti

3.2. Problem

X-ray cihazlarının kontrolü manuel bir şekilde ilgili noktadaki görevliler tarafından yapılmaktadır. Bu hem zaman alan bir süreç hem de görevlinin dikkatsizliği sonucunda kötü niyetli kişiler tarafından içeriye tehdit unsuru içeren bıçak, jilet, silah gibi nesnelerin koymasında bir güvenlik zafiyeti oluşturabilmektedir.

3.3. Neden veri madenciliği?

Birçok probleme çözüm getiren derin öğrenme algoritmalarının görüntülerde nesne tespit etme ve sınıflandırma problemlerinde yüksek performans sergilemesi.

3.4. Nasıl ve nerede kullanıldı?

Bu çalışmada X-ray cihazlarından alınan görüntülerde tehdit unsuru içeren nesneleri sınıflandırmak için transfer öğrenme yöntemi ile gömülü ve gerçek zamanlı çalışabilen bir sistem geliştirilmiştir. Bu sistem Nvidia Jetson TX2 geliştirme kartı üzerinde bir evrimsel sinir ağı olan Alexnet derin öğrenme modeli kullanılmaktadır. Bu amaçla NVIDIA Jetson TX2 geliştirme kartı üzerinde Caffe derin öğrenme platformu kurulmuştur. Her bir sınıf için 4.000 tane örnek olmak üzere toplam 12.000 tane görüntü eğitim aşamasında kullanılmıştır. Test aşamasında Nvidia Jetson TX2 üzerinde gömülü bulunan kameradan alınan gerçek zamanlı görüntüler kullanılmıştır.

3.5. Başarı Oranı

Deneyler sonucunda eğitim başarımı %99.6 olarak elde edilmiştir. Test aşamasında her bir görüntünün işlenmesi yaklaşık olarak 12 milisaniye sürmektedir. Dolayısıyla bir saniyede 80 görüntü işlenebilmektedir.

Jetson Tx2 geliştirme kartı üzerinde bulunan 5 mega piksel kamera ile alınan görüntülerden önerilen sistemin gerçek zamanlı testi yapılabilmektedir. Bu doğrultuda bıçak, silah, jilet ve Ninja yıldızı gibi görüntüler başarılı bir şekilde sınıflandırılmıştır. Her bir sınıf için nesnelerin, farklı açılardan ve arka plan gürültüleri eklenmiş görüntüler kullanılmıştır. Yapılan testlerde nesne sınıflandırmada başarı oranının (% 99) oldukça yüksek olduğu görülmüştür.

3.6. Kullanılan Kütüphane Özelliği

Problemin çözülmesinde Caffe derin öğrenme platformu kullanılmıştır. Bu platformun özelliği

- İleri beslemeli ağlar ve görüntü işleme konularında hızlıdır.
- Hassas ayarlama için önceden eğitilmiş modelleri vardır.
- Hiçbir kod yazmadan model eğitilebilir.
- Python arayüzü oldukça kullanışlıdır.
- GPU desteği vardır.

3.7. Kaynak

SALUR, M. U., AYDIN, İ., & KARAKÖSE, M. Gömülü Derin Öğrenme ile Tehdit İçeren Nesnelerin Gerçek Zamanda Tespiti. *DÜMF Mühendislik Dergisi*, 10(2), 497-509.

4. Araştırma Ödevi 1, Makale 2

4.1. İncelenen Tez Adı

Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi

4.2. Problem

Görüntü işleme yöntemleri kullanılarak durağan görüntülerin analizleri gerçekleştirilebilir ve söz konusu görüntülerden anlamlı bilgiler çıkarılabilir. Tespit ve tanıma sonrasında takip edilecek olan nesnenin değişken bir ortam içinde bulunması zorlaştırıcı unsurlardan birisidir.

4.3. Neden Veri Madenciliği?

Derin öğrenme algoritmaları yapay zekâ alanında bir devrim yaratmıştır. Derin öğrenme algoritmalarının görüntü işlemede kullanılması oldukça başarılı sonuçlar alınmasını ve karmaşık görüntü işleme problemlerinin kolaylıkla çözüme kavuşturulabilmesini sağlamaktadır

4.4. Nasıl ve Nerede Kullanıldı?

Bu çalışmada derin öğrenme ile hareketli nesne tanıma ve takibi için Google'ın açık kaynak kodlu makine öğrenmesi kütüphanesi olan TensorFlow kullanılmıştır. Nesne takibi için Region Based Convolutional Networks kütüphanesinden Faster R-CNN modeli ele alınmıştır. Bu kütüphaneler ile durağan görüntüler, video görüntüleri ve webcam görüntüleri üzerinde nesne tanıma işlemi gerçekleştirilmiş. Faster R-CNN'de bağımsız değişkenler ile bir bağımlı değişken arasındaki bağlantıyı modellemek için kullanılan doğrusal regresyon ile nesnenin sınırları belirlenmektedir. Doğrusal regresyon ile nesnenin bulunduğu yeri tam olarak tespit ederek nesnenin etrafında dikkörtgen çizilmektedir. Geliştirilen uygulama için “kedi”, “kopek”, “dokuz”, “on”, “uc”, “as” ve “resuldas” etiketleri oluşturuldu.

4.5. Başarı Oranı

Eğitim gerçekleştikten sonra Faster R-CNN modelinde %98 başarı oranı yakalanmıştır.

4.6. Kullanılan Kütüphane Özellikleri

Bu çalışmada derin öğrenme ile hareketli nesne tanıma ve takibi için Google'ın açık kaynak kodlu makine öğrenmesi kütüphanesi olan TensorFlow kullanılmıştır. Bu kütüphanenin özellikleri:

- Hızlı derleme yapabilmektedir.
- TensorBoard ile görselleştirme yapabilmektedir
- Veri ve model paralellliği sağlar
- GPU veya CPU'da paralel çalışabilmektedir

4.7. Kaynak

Daş, R., Polat, B., & Tuna, G. Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31(2), 571-581.

5. Araştırma Ödevi 1, Makale 3

5.1. İncelenen Tez Adı

Uzaktan algılama verilerinin derin öğrenme algoritmaları ile sınıflandırılması

5.2. Problem

Konumsal veri üretiminde hava ve uydu platformlarından elde edilen görüntüler kullanılmaktadır. Bu görüntüler bitki örtüsü tespitinden, doğal afetler ve şehir planlamasına kadar birçok alanda kullanılmaktadır. Uzaktan algılama verilerindeki yüksek çözünürlüklü görüntüler sayesinde insan yapımı yapay unsurların tespitinde kolaylaşmıştır. Ancak yüksek çözünürlüklü verilerinin sunduğu bu avantajların yanında yoğun ayrıntı içeriği nedeni ile sınıflandırmada işlemlerinde insan becerisi ve müdahalesine duyulan gereksinimin fazlalığı günümüzde önemli sorunlardan biri olarak karşımıza çıkmaktadır.

5.3. Neden Veri Madenciliği?

Daha önceden bilgisayar teknolojisinde yeterli olmayan donanımsal ve yazılımsal araçlardan dolayı, yapay zekanın temelleri sayılan makine öğrenmesi ve derin öğrenme gibi konular pek incelenmemiştir. Ancak günümüzde grafik işlemcilerde ve yazılımlardaki ilerlemeler sayesinde popülerliğini arttırmıştır. Son çalışmalardaki performansı ile derin öğrenme yaklaşımının problemleri çözümlemesindeki başarısından dolayı uzaktan algılama verilerinin otomatik sınıflandırılmasında derin öğrenme algoritmalarının iyi bir alternatif olduğu görülmektedir. Bilgisayar görüşü sistemleri artık birçok alanda insan performansının üstüne çıkmaktadır. Onlardan birisi olan derin öğrenme yapısı da mevcutta bulunan veri setlerinin sınıflandırılmasında insan faktörünün üzerine çıkmaktadır. Derin öğrenmenin bu özelliğinden dolayı çalışmada tercih edilmiştir.

5.4. Nasıl ve Nerede Kullanıldı?

Bu çalışmada derin öğrenme yapıları, hava ve uydu görüntülerinin sınıflandırılması sorununa alternatif bir çözüm yolu olarak sunulmuştur. Bu kapsamda Vaihingen veri seti kullanılarak bir şehirde havadan görüntülenebilecek detayların tespit edilip sınıflandırılması için derin öğrenme algoritmaları ile ne kadar süre aldığı ve ne kadar doğruluk elde edileceği tespit edilmeye çalışılmıştır. Görüntü sınıflandırması söz konusu olduğunda en iyi sonucu Evrişimsel Sinir Ağı(ESA) mimarisi vermektedir. UFUAT'ın Vaihingen veri setini kullanarak sınıflandırma yapılmıştır. Veri setinde 3 bantlı IRGB ortofotoların 12 tanesi eğitim, 4 tanesi test verisi olarak ayrılmıştır. Eğitim sırasında sınıflandırmanın doğruluğunu kontrol etmek için daha önceden ayrılmış gerçek zemin görüntüleri ile sonuç ürünler karşılaştırılmıştır. Objelerin sınırlarının daha keskin belirlenmesi içinde gerçek zemin görüntülerinde sınır belirginleştirme işlemi yapılarak ayrıca eğitim verilerine eklenmiştir. Veri seti zemin, bina, bitki örtüsü, ağaç ve araba olarak 5 sınıfa ayrılmıştır. Eğitim verilerinde sınıflandırma yaparken semantik segmentasyon yapabilmek için derin öğrenme yapısındaki SegNet mimarisini kullanılmıştır. Tüm bu eğitim ve test aşamaları yapılırken sistem GİB' in belleğinde büyük bir alan kullanmaktadır. Bu işlem için en az 12 GB' lik bir GİB gerekmektedir. Böyle donanımsal özelliğe sahip bir GİB maliyeti oldukça yükseltir. Bundan dolayı Google'ın ücretsiz GİB desteği verdiği Colaboratory (Colab) hizmeti kullanılarak eğitim ve test aşamaları gerçekleştirilmiştir. Colab' ta Nvidia' nın Tesla k40 GİB kullanılmaktadır.

5.5. Başarı Oranı

- Genel doğruluk 98.24.
- Zemin 98.33
- Bina 98.98
- Bitki Örtüsü 95.26
- Ağaç 98.16
- Araç 93.74

5.6. Kaynak

Döş, M. E., & Uysal, M. Uzaktan algılama verilerinin derin öğrenme algoritmaları ile sınıflandırılması. *Türkiye Uzaktan Algılama Dergisi*, 1(1), 28-34.

6. Araştırma Ödevi 1, Makale 4

6.1. İncelenen Tez Adı

Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması

6.2. Problem

Derin öğrenme yaklaşımı ile çözümlenecek problem için oluşturulacak modelin belirlenmesi.

6.3. Neden Veri Madenciliği?

Son zamanlarda bilgisayar teknolojisindeki gelişme hızı ile orantılı olarak yapay görüş ve yapay öğrenme alanlarında da gelişmeler olmuştur. Bu çalışmalarda popüler olarak derin öğrenme kullanılmaktadır. Derin öğrenme ile makine öğrenmesi arasında önemli bazı farklılıklar olsa da temelde aynı şeyleri ifade ederler. Derin öğrenme, makine öğrenmesinin bir alt kategorisidir. Makine öğrenmesi teknikleri ile ilk başlarda iyi sonuçlar elde edilmiştir. Ancak 2010 yılından itibaren artan veri seti boyutları nedeni ile klasik makine öğrenmesi kullanılarak yapılan çalışmalarda istenilen gelişmeler sağlanamamıştır. Derin öğrenme yapısı günümüzde oldukça popülerdir. Aslında derin öğrenme yapıları yeni gibi görünse de temeli eskiye dayanmaktadır. Genel olarak bakıldığında derin öğrenme ağ yapısı ile makine öğrenmesi ağ yapısı birbirine benzerler. Ancak temelde özellik çıkarımı makine öğrenmesinde kullanıcı tarafından belirlenmekte bu da özellik görüşünü kısıtlamaktadır. Derin öğrenme yapısı ile makine öğrenmesi yapısı arasındaki bu farklılıktan dolayı, derin öğrenmede veri miktarının artması ile performans aynı seviyede kalmayıp veri miktarı ile aynı oranda artmaktadır. Bundan dolayı artan verilerin analizinde derin öğrenme yaklaşımı sıklıkla kullanılmaktadır.

6.4. Nasıl ve Nerede Kullanıldı?

Bu çalışma, derin sinir ağlarında en iyi sınıflandırma modelini bulmak amacıyla gerçekleştirilmiştir. Bu amaçla, optimizasyon yöntemi (Sgd, Adagrad, Rmsprop, Adam ve Nadam), aktivasyon fonksiyonu (Tanh ve ReLU) ve nöron sayılarının kombinasyonları (32, 64) kullanılarak 20 farklı model oluşturulmuştur. Oluşturulan model kombinasyonlarının performansları karşılaştırılarak, sınıflandırma için en iyi model belirlenmiştir. Çalışmada Python kütüphanesi olan Keras kullanılmıştır.

Çalışmada kullanılan Mantar veri seti (Mushroom data set), UCI makine öğrenme veri deposundan alınmıştır. (UCI, 2018). Dua ve Graff (2018) tarafından 8124 mantar örneğinden oluşturulan veri setinde; şapkanın şekli, şapkanın yüzeyi, şapkanın rengi, çürüme durumu, kokunun tanımı, şapkanın alt kısmı, lamellerin aralığı, lamellerin ebadı, lamellerin rengi, sap şekli, kökün durumu, sap yüzeyinin üstündeki halkanın görünümü, sap yüzeyinin üstündeki halkanın rengi, sap yüzeyinin altındaki halkanın görünümü, sap yüzeyinin altındaki halkanın rengi, örtü tipi, örtü rengi, halka sayısı, halka tipi, spor rengi, popülasyon yoğunluğu ve yaşam alanı şeklinde 22 bağımsız değişken yer almaktadır. Bağımlı değişken ise mantarların “yenilebilir (edible=e)” veya zehirli (poisonous=p)” şeklinde sınıflandırılmasından oluşturulmuştur. Çalışmada, veri setinin %80’i eğitim ve %20’si test verisi olarak ayrılmıştır.

6.5. Başarı Oranı

Nöron Sayısı	Aktivasyon Fonksiyonu	Model No	Optimizasyon Yöntemi	Doğruluk	Hata	Test Doğruluk	Test Hata
32	ReLU	1	Sgd	0.98	0.02	0.70	0.30
		2	Adagrad	0.99	0.01	0.75	0.25
		3	Nadam	0.99	0.004	0.86	0.14
		4	Adam	0.99	0.003	0.86	0.14
		5	Rmsprop	0.99	0.004	0.84	0.16
	Tanh	6	Sgd	0.98	0.02	0.74	0.26
		7	Adagrad	0.99	0.009	0.73	0.27
		8	Nadam	0.99	0.003	0.87	0.13
		9	Adam	0.99	0.004	0.83	0.17
		10	Rmsprop	0.99	0.006	0.78	0.22
64	ReLU	11	Sgd	0.98	0.02	0.71	0.29
		12	Adagrad	0.99	0.005	0.80	0.20
		13	Nadam	0.99	0.002	0.87	0.13
		14	Adam	0.99	0.003	0.89	0.11
		15	Rmsprop	0.99	0.002	0.92	0.08
	Tanh	16	Sgd	0.98	0.02	0.75	0.25
		17	Adagrad	0.99	0.006	0.78	0.22
		18	Nadam	0.99	0.003	0.84	0.16
		19	Adam	0.99	0.004	0.82	0.18
		20	Rmsprop	0.99	0.007	0.78	0.22

6.6. Kullanılan Kütüphane Özellikleri

Bu çalışmada keras kullanılmıştır.

- Torch kütüphanesinden esinlenilmiş sezgisel bir API'dir.
- Theano, TensorFlow, DeepLearning4j ve CNTK arka planda kullanılmaktadır.
- Hızlı büyüyen bir yapısı vardır.
- GPU veya CPU'da paralel çalışabilmektedir.

6.7. Kaynak

Gazel, S. E. R., & BATI, C. T. Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması. *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, 29(3), 406-417.

7. Araştırma Ödevi 1, Makale 5

7.1. İncelenen Tez Adı

Derin Öğrenme Kullanarak Otonom Araçların İnsan Sürüşünden Öğrenmesi

7.2. Problem

Otonom araçlar, çevre koşullarını algılayarak kararlar alan ve aldıkları kararlar doğrultusunda hareket eden araçlardır. Otonom araçlara olan ilgi dünya çapında giderek artmaktadır. Bu teknolojiye gelişim potansiyeli çok açık bir şekilde görülmektedir. Yakın zamanda ulaşım ve taşımacılık alanında ciddi gelişmelerin yaşanmasının beklendiği bu alanda, otonom araçların algılama sistemlerinin mümkünse sıfır hata yapması gerekmektedir.

7.3. Neden Veri Madenciliği?

Son yıllarda, derin öğrenme yöntemleri nesne algılama ve tanıma için güçlü makine öğrenmesi yöntemleri olarak ortaya çıkmıştır. Derin öğrenme yöntemleri, özellikleri ham piksellerden doğrudan öğrenir. Otonom yer aracının insan sürüşünden alınan veriler yardımıyla otonom sürüşü öğrenmesi amaçlandığından derin öğrenme tercih edilmiştir.

7.4. Nasıl ve Nerede Kullanıldı?

Küçük bir yer aracı ile yapay öğrenme yöntemlerini kullanan otonom bir araç tasarlanmıştır. Bu amaçla, yer aracı üzerine çeşitli sensörler, kamera ve Nvidia TX2 kartı yerleştirilmiştir. Otonom yer aracının insan sürüşünden öğrenmesi için, Evrişimsel Sinir Ağları (ESA'lar) ve Uzun Kısa-Vade Hafıza Ağları (UKVHA'ları) birlikte kullanılan bir model önerilmiştir. Geliştirilen modelleri kullanan otonom araç, tasarlanan parkur üzerinde test edilmiştir.

Otonom aracın eğitiminde kullanılmak üzere kamera ve sensörlerden eğitim verileri toplayarak eğitim kümesi oluşturmaktır. El ile sürüş sonucunda, kameradan 1860 tane görüntüye ek olarak, araca ait hız ve direksiyon açılarının değerleri de kayıt altına alınmıştır. Elde edilen görüntülerin %80'i eğitim kümesi, %20'si test kümesi olacak şekilde ayrılmıştır.

Eğitim, Intel i7 8700K 3.70 GHz 12M 1151p işlemci ve NVIDIA GTX Titan X Pascal ekran kartı içeren bir bilgisayar üzerinde gerçekleştirilmiştir. Eğitilen model daha sonra Jetson TX2 kartına gömülmüştür ve 1860 görüntü toplanmıştır.

Katman	Filtre Boyutu	Öznitelik Uzayı Boyutu	Kaydırma/Azaltma Boyutu	Etkinleştirme İşlevi	Çıkış Boyutu	Parametre Sayısı
2B Kırpma katmanı (cropping2d 1)	-	-	-	-	156, 672, 3	0
Düzgeleme Katmanı (lambda 1)	-	-	-	-	156, 672, 3	0
2B Evrişim katmanı (conv2d 1)	5x5	24	2x2	DDB (ReLU)	76, 334, 24	1824
2B Evrişim katmanı (conv2d 2)	5x5	36	2x2	DDB (ReLU)	36, 165, 36	21636
Öznitelik Seçici Katman (Max pooling)	-	-	2x2	-	18, 82, 36	0
2B Evrişim katmanı (conv2d 3)	5x5	48	2x2	DDB (ReLU)	7, 39, 48	43248
Düğüm düşürme katmanı- (Dropout)	-	-	0.2	-	7, 39, 48	0
Vektöre dönüştürme Katmanı (Time-distributed flatten 1)	-	-	-	-	7, 1872	0
UKVHA katmanı	-	36	-	-	36	274896
Tam bağlı katman (dense 1)	-	100	-	-	100	3700
Tam bağlı katman (dense 2)	-	50	-	-	50	5050
Tam bağlı katman (dense 3)	-	10	-	-	10	510
Tam bağlı katman (dense 4)	-	1	-	-	1	11

7.5. Başarı Oranı

Otonom sürüşte hedef, yüksek hızda aracın kararlı bir şekilde sürüşünün gerçekleştirilmesi ve direksiyon açısı üzerindeki hatanın düşük olmasıdır. Bu da UKVHA katmanı ile başarılmıştır. Özellikle yüksek hızlarda virajlarda kararlı dönüşler elde edilmiştir.

7.6. Kaynak

BİNGÖL, M. S., KAYMAK, Ç., & UÇAR, A. (2019). Derin Öğrenme Kullanarak Otonom Araçların İnsan Sürüşünden Öğrenmesi. *Firat University Journal of Engineering*, 31(1).

8. Araştırma Ödevi 1, Makale 6

8.1. İncelenen Tez Adı

Metin Madenciliği, Makine ve Derin Öğrenme Algoritmaları ile Web Sayfalarının Sınıflandırılması

8.2. Problem

Bireylerin ve kurumların internet erişim imkanları arttıkça, web üzerinden paylaşılan bilgiler ve web sitelerinin sayısı orantılı olarak hızlı bir şekilde artmaktadır. 2019 yılı itibarı ile dünya çapında 1,7 milyar civarında Web sitesi bulunmakta, fakat bunlardan sadece 200 milyona yakını aktiftir (Internet Live Stats, 2019; Netcraft, 2019). Web sitelerinin ve bu sitelerde barındırılan sayfaların sayısının artması ile, kişilerin istedikleri bilgilere hızlı ve kolay erişimleri ya da zararlı olabilecek içeriklerin engellemek için web sayfalarını sınıflandırma ihtiyacı

doğmuştur. Ancak bu yönteminin insan gözüyle işlenmesi çok fazla efor ve zaman harcanmasına neden olmaktadır. Aynı zamanda bu sınıflandırma sürecinde hata yapılması ihtimali oldukça yüksektir.

8.3. Neden Metin Madenciliği?

Yapısal veriler üzerinden çıkarım yapmak ve bilgi edinmek için Veri Madenciliği teknikleri kullanılır. Veri tipinin analiz işlemine uygun olmayan bir metin yapısında olması durumunda Metin madenciliğine ihtiyaç duyulur. Metin madenciliği tipik olarak kümeleme, sınıflandırma, ilişkilendirme kuralları ve tahmine dayalı modelleme gibi makine öğrenme tekniklerini uygular. Bu tekniklerle metinlerdeki anlamları ve ilişkileri ortaya çıkartabilir.

8.4. Nasıl ve Nerede Kullanıldı?

Mevcut çalışmada, Web sitesi sınıflandırma problemi bir metin sınıflandırma problemi olarak kabul edilmiş olup metin sınıflandırmasında kullanılan yöntemler uygulanmıştır. Diğer bir deyişle ele alınmakta olan bir metne, bu metnin içeriğine göre ve önceden belirlenmiş yöntemler ile etiket veya kategori atanmıştır.

Amazon'un bir hizmeti Alexa Top Sites'tan alınmış olan 63 tane web sitesi sınıfı ve 45.543 Web site verisinden 37.814 tanesi (%83,03) eğitim, 7.729 tane Web site verisi (%16,97) ise test sırasında kullanılmıştır. Web sitelerinden toplanmış olan metin verisinin temizlenmesi ve ön işlenmesi sürecinde Python tabanlı bir doğal dil işleme kütüphanesi olan NLTK4 ve NLTK'da tanımlanmış stop word listesi5 kullanılmıştır. Seçilmiş olan makine ve derin öğrenme yöntemlerinin uygulanmasından önce Web sitelerinden elde edilmiş olan veriler metin tabanlı olduğundan, kelime vektörleştirme ile makine ve derin öğrenme algoritmaları tarafından eğitilebilecek ve analiz edilebilecek sayısal değerlere dönüştürülmüştür. Kelime vektörleştirme işlemi için yaygın olarak kullanılan Kelime Torbası Terim Sıklığı - Ters Metin Sıklığı ve Word2Vec yöntemleri her bir algoritmada ayrı ayrı kullanılmıştır.

Sınıflandırma sürecinde İkili Sınıflandırma ve Çoklu Sınıflandırma temelli iki farklı Web sitesi sınıflandırma yöntemi geliştirilmiş, her yönteme uygun makine öğrenmesi ve derin öğrenme algoritmaları ilişkilendirilerek bir Web sitesi sınıflandırma metodolojisi oluşturulmuştur.

İkili (Binom) Sınıflandırma'da, tanımlanmış sınıf sayısı kadar (63 tane) sınıflandırıcı oluşturulmuştur. Bu sınıflandırma yöntemi için ayrı ayrı Bernoulli Naive Bayes, Lojistik Regresyon ve Tam Bağlantılı Yapay Sinir Ağları algoritmaları sırası ile kullanılmıştır. Çoklu sınıflandırma yaklaşımında ise İkili Sınıflandırmadan farklı olarak; 63 sınıftan her biri için ayrı ayrı sınıflandırıcılar oluşturmak yerine tek bir sınıflandırıcı oluşturulmuş ve tüm testler bu tek model üzerinden yapılmıştır. Multinomial Naive Bayes, Rastgele Orman ve SVM sınıflandırıcılar Çok Sınıflı Sınıflandırma modelinin oluşturulmasında kullanılan yöntemlerdir.

8.5. Başarı Oranı

Başarı oranları değerlendirildiğinde, İkili Sınıflandırmanın istenilen bir Web site sınıfının filtrelenmesi görevi için kullanıldığında, Çoklu Sınıflandırmaya göre daha başarılı sonuçlar ürettiği tespit edilmiştir. İkili Sınıflandırmada uygulanan yöntemlerin mevcut çalışmanın veri seti, çalışmada kullanılan betikler ve donanım özellikleri kapsamındaki işlemsel performansları süre olarak incelendiğinde, Lojistik Regresyon ve Bernoulli Naive Bayes sınıflandırıcılarının, Tam Bağlantılı Yapay Sinir Ağlarına göre 150 kat daha hızlı çalıştığı ve sonuç ürettiği gözlemlenmiştir. Tam Bağlantılı Yapay Sinir Ağlarının diğer sınıflandırıcılara göre daha düşük başarı ile çalışmasının sebebi, verinin yeterli büyükte olmaması, her sınıfta yeterli sayıda örnek olmaması ya da dengeli şekilde dağılmaması olabileceği düşünülmektedir. Gelecek çalışmalarda, toplanan veri setindeki her sınıf için örnek sayısı artırılarak Yapay Sinir Ağlarının performansı arttırılabilir.

8.6. Kullanılan Kütüphane Özelliği

Modeller Python 3 betikleri ile kodlanmış olup kodlama sürecinde Python'un scikit-learn, SciPy, NLTK, Keras, PyPI regex, Pandas ve NumPy kütüphaneleri kullanılmıştır.

Keras kütüphanesi;

- Derin öğrenme modellerinin prototiplemesinin hızlıca yapılmasına olanak sağlayan ve Torch kütüphanesinden esinlenilmiş kullanıcı dostu, sezgisel bir API'dir
- Evrişimli ağlar, yinelemeli ağlar ve her ikisinin beraber kullanımı için önceden tanımlı desteğe sahiptir.
- Aynı kodu değiştirmeden hem CPU'da hem de GPU' da çalışmasını sağlar.

8.7. Kaynak

CHOUSEİNOGLOU, O , ŞAHİN, İ . (2019). Metin Madenciliği, Makine ve Derin Öğrenme Algoritmaları ile Web Sayfalarının Sınıflandırılması. Yönetim Bilişim Sistemleri Dergisi , 5 (2) , 29-43. Retrieved from <https://dergipark.org.tr/tr/pub/ybs/issue/51328/643892>

9. Araştırma Ödevi, Makale 7

9.1. İncelenen Tez Adı

Apache Spark ve Makine Öğrenmesi Algoritmaları ile Ağ Saldırısı Tespiti

9.2. Problem

İnternet tabanlı hizmetlerin sürekli olarak artış göstermesi, ağ trafik verilerini günden güne daha büyük ve karmaşık hale getirmektedir. Bu durum, ağ saldırılarının tespitini ve ağ güvenliğinin sağlanmasını giderek zorlaştırmaktadır.

9.3. Neden Büyük Veri?

Büyük veri düzenli veya düzensiz olmak üzere, her gün yapılan çalışmaların işleyişini zorlaştıran büyük miktardaki veriyi tanımlar. Büyük verilerin analiz edilip sınıflandırılarak, anlamlı ve işlenebilir hale dönüştürülmesi daha iyi strateji ve karar verme olanağı sağlar. Normalde görülmeyen kalıpları ve bağlantıları ortaya çıkarır, değerli bilgiler sunar.

9.4. Nasıl ve Nerede Kullanıldı?

Çalışmada; büyük veri teknolojisi Apache Spark kullanılarak KDD Cup'99 verilerinin tamamı makine öğrenmesi algoritmalarından Lojistik Regresyon, Destek Vektör Makineleri, Naive Bayes ve Rastgele Orman üzerinde koşturulmuş; sonuçlar karşılaştırmalı olarak analiz edilmiştir. Deneysel veri seti olarak, ağ saldırısı tespit sistemlerinde çokça kullanılan KDD Cup'99 veri seti kullanılmıştır. KDD Cup'99, askeri ağ ortamlarındaki çeşitli sahtekârlık saldırılarını geniş bir yelpazede kapsar, ağ üzerinde denetlenecek verilerin standart bir kümesini içerir. Her biri 41 özelliğinden oluşan normal veya özel bir saldırı tipi olarak etiketlenen yaklaşık 5 milyon bağlantı verisi içermektedir. Veri setinde yer alan bağlantı verileri, %60-%40 oranında ayrılarak eğitim ve tahminleme verisi olarak kullanılmıştır. Veri ön işleme aşamasında ilgili problemi tanımlamada etkili olacak özellikler çıkarılarak oluşturulan özellik vektörü ve yine veriden elde edilen sınıf etiketi şeklinde iki gruba ayrılır. Kategorik özellikler sayısallaştırılır. Verideki özellikler tümüyle sayısal forma dönüştürülür. Eğitim aşamasında sınıf etiketi olmayan bir veri makineye verildiğinde, verinin sınıfında tahminlemede bulunabileceği nitelikteki veri tipiyle ilgili öğrendiği bilgilerin çıktısını verebilmesi ve algoritma bazında modeller oluşturulması beklenir. Makine, sınıf etiketi olmayan bir veri ile karşılaştığında; bu veri tipiyle ilgili belirlenen algoritma kurallarını kullanarak eğitim aşamasında oluşturduğu model üzerinde çalışmaya başlar ve önceden öğrendiği bilgilerle yeni verinin sınıfını tahminlemeye çalışır. Tahminleme aşamasının çıktısı, makinenin yeni verinin sınıfı konusunda yaptığı tahminlemedir; yani tahmini sınıf etiketidir.

9.5. Başarı Oranı

En yüksek doğru tahminleme oranlarının sırasıyla Lojistik Regresyon, Rastgele Orman ve Destek Vektör Makineleri algoritmalarına aittir. Eğitim-tahminleme süreleri arasındaki farklar sonuçlar üzerinden değerlendirilecek olursa, diğer algoritmalarla karşılaştırıldığında Lojistik Regresyon algoritmasının en yüksek süre farkına sahip olduğu görülmektedir. Büyük ağ verileri üzerindeki saldırıları tespit etmede Apache Spark'ın çıkan yeni sürümleriyle giderek daha etkili bir hale gelmiştir. Naive Bayes algoritması haricindeki bahsi geçen makine öğrenmesi algoritmaları ile, kıyaslama yapılan önceki sistemden daha doğru tahminlemelerde bulunulmuştur. Naive Bayes algoritmasının doğru tahminleme oranı düşüşünün, algoritma yapısından kaynaklıdır.

9.6. Kullanılan Kütüphane Özelliği

Çalışmada Apache Spark Makine Öğrenmesi Kütüphanesi (Mllib)'ne ait DataFrame tabanlı API kullanılmıştır. Apache Spark büyük veri üzerinde paralel işlem yapmaya yarayan bir kütüphanedir.

- Açık kaynak kodludur.
- Scala dili ile geliştirilmiştir.
- Hadoop'tan farkı hızlı olmasıdır.

- Kullanımı kolaydır.
- Makine öğrenmesi, SQL, akan veri üzerinde çalışma(streaming) ve grafik işleme modülleri vardır.

MLlib(Machine Learning): Aksiyon almaya yarayan öngörüler elde etmek için, makine öğrenmesi, büyük verinin işlenmesinde kritik bir aşama olarak ortaya çıkmıştır. Ölçeklenebilir makine öğrenmesi kütüphanesi olan MLLib, hem yüksek kalitede algoritmalar (başarıyı artırmak için birden fazla iterasyon yapan) hem de hız (MapReduce'den 100 kat fazla) sunar. Kütüphane Java, Scala ve Python'da Spark uygulamalarının bir parçası olarak kullanılabilir.

9.7. Kaynak

Kurt, E. M. (2019). Apache spark ve makine öğrenmesi algoritmaları ile ağ saldırısı tespiti(Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü).

10. Araştırma Ödevi, Makale 8

10.1. İncelenen Tez Adı

Derin Öğrenme ile İki Boyutlu Optik Karakter Tanıma

10.2. Problem

Görüntülerdeki optik karakterlerin tanımlanması işlemi, kağıt üzerindeki verilerin saklanması veya görüntü dosyalarındaki verilerde arama yapılması amacıyla verilerin dönüştürülmesi üzerine yapılmış bir çalışmadır. Günümüzde elektronik belge yönetim sistemlerinin (EBYS) tümü yasalara göre OCR barındırması gerekir. Bu nedenle OCR, hem iş dünyasında hem de devletin birçok kurumunda kullanılması zorunlu bir araç haline gelmiştir.

10.3. Neden Derin Öğrenme?

Makine öğrenmesinin yeni bir alanı olan Derin Öğrenme, son dönemde geliştirilen yapay zekâ uygulamalarının başarısını oldukça yüksek düzeylere çıkarmıştır. Veri miktarının ve ekran kartlarının işlem gücünün artması nedeniyle derin öğrenme en çok talep gören yöntemlerden biridir.

10.4. Nasıl ve Nerede Kullanıldı?

Optik karakter tanıma işlemi birkaç aşamadan oluşmaktadır. Bu aşamalar görüntü ön işleme, karakter ayıklama ve karakter sınıflandırma işlemidir. Karakter ayıklama işlemi için matlab ortamında bwlable fonksiyonu kullanılmıştır. Karakter sınıflandırma; ayıklanan karakterlerin hangi ASCII kodu olduğunu bulma işlemidir. Bu işlemde Evrişimli Sinir Ağları kullanılmıştır. Sinir ağlarının doğruluk oranının eğitim işlemindeki veri sayısı ile doğru orantılı olması nedeniyle ve bu tanımlama işleminin başarısını arttırabilmek için 2.1600.000 karakter örneği ile sistem eğitilmiştir. Evrişimsel sinir ağını eğitmek için kullanılan eğitim verilerinin hepsinin aynı boyutta ve aynı derinliğe sahip olması gereğiyle karakterler 28x28 boyutunda tek katmanlı gri görüntü sayısına çevrilmiştir. Ağ eğitimi GPU ile yapıp eğitim 24 saat sürmüştür.

10.5. Başarı Oranı

Karakter tanıma işlemi; karakterlerin birbirlerine olan benzerliklerinden dolayı sınıflandırma işlemini olumsuz etkilemektedir. Bu nedenle karakter tanıma işleminin doğruluğunu artırmak için kullanılan verinin fazla olması önerilmektedir. Ayrıca karakter tanıma işlemi için kullanılan veri setinin gri renkte yani tek katmanlı olması; ağı doğruluk oranını artırmakla birlikte daha az veri seti ile ağı daha iyi doğruluk oranlarına ulaştırmıştır

10.6. Kullanılan Kütüphane Özelliği

MATLAB programlama dili ile genel olarak mantıksal ve matematiksel kavramları içinde barındıran projeler geliştirilir. Buna bağlı olarak bulanık mantık, sezgisel optimizasyon algoritmaları, yapay sinir ağları, görüntü işleme, genetik algoritma, veritabanı projeleri ortaya çıkar. Çünkü bu projelerin çoğu matematiksel uygulamalara dayanarak oluşturulur. Kullanılacak araçlar birbirleriyle çalışacak şekilde tasarlanmıştır ve paralel bilgi işlem ortamları, GPU'lar ve otomatik C kodu ile bütünleşir. Birlikte güncellenir, bu yüzden uyumsuz kütüphane sürümleriyle asla uğraşmazsınız.

10.7. Kaynak

Koyun, A , Afşin, E . (2017). Derin Öğrenme ile İki Boyutlu Optik Karakter Tanıma. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi , 10 (1) , 11-14 . Retrieved from <https://dergipark.org.tr/tr/pub/tbbmd/issue/31320/314138>

11. Araştırma Ödevi 1, Makale 9

11.1. İncelenen Tez Adı

Veri Madenciliği ile Yazılım Hata Tespiti

11.2. Problem

Yazılım projesi geliştirmede en önemli safhalar çözümleme, tasarım, kodlama, sınama ve bakımdır. Bu aşamaların bir kısmında yapılan hatalar, eksiklikler projenin tamamına olumsuz yönde etki eder. Hatalı ürün çıktıları aynı iş ürünü üzerinde tekrar çalışmalara sebep olmaktadır. Hatalı üretilen ürün bileşenleri üzerinde yazılım uzmanları tekrar düzenleme ve ek geliştirme yapmakta olduğu gibi test uzmanları da aynı iş ürünü üzerinde tekrar test gerçekleştirmektedir. Bu ise hem zaman hem müşteri nezdinde güven ve hem de ticari işletmenin görünür görünmez birçok maliyetleri olarak ortaya çıkmaktadır.

11.3. Neden Veri Madenciliği?

Veri madenciliği büyük veri setlerinden önemli bilgileri elde ettiğimiz bir süreçtir. Bankacılıkta risk analizlerinde; sigortacılıkta usulsüzlüklerin önlenmesinde; tıp alanında teşhiste; endüstride kalite kontrole kadar oldukça geniş bir uygulama alanı vardır. Verilerin çok hızlı artması, bilgisayarların güçlenmesi, algoritmalarındaki yeni gelişmeler ile veri bilimi birçok farklı alanda zor problemlerin çözümünde yeni bir fırsat sunmuştur. Kısaca veri madenciliğinin çözdüğü problemler hayatı kolaylaştırmakta ve verimliliği arttırmaktadır.

11.4. Nasıl ve Nerede Kullanıldı?

Bu bildiride sunulan çalışma kapsamında geliştirilen yazılım projelerinden toplanan veriler veri madenciliği yöntemleri kullanılarak hataların tespiti, yazılımcı-hata ilişkilerini içeren kurallar çıkartılması, sınıflandırmalar ve kümeleme yöntemleri uygulanmıştır. Bu yöntemler Karar Ağacı, Apriori Algoritması, Naive Bayes ve Yapay Sinir Ağlarıdır.

Karar Ağaçları yönteminde entropi hesaplayarak her bir özellik için kazanç hesabı yapıldıktan sonra elde edilecek ağacın ilk düğümü belirlenip diğer özellikler için veri tabanı güncellenerek sınıflar ortaya çıkana kadar devam edilip karar ağacı oluşturulur. Naive Bayes Sınıflandırma yöntemlerinde ise ihtimal hesapları kullanılarak ortaya çıkabilecek her bir olasılık için bir oran belirlenir. Yeni bir veri ortaya çıktığında her bir sınıf için ihtimal yüzdeleri hesaplanır ve verinin en yüksek sınıfa ait olduğu sonucu çıkarılır.

11.5. Başarı Oranı

Veri madenciliği yöntemi ile elde edilen başarı oranları;

- Karar Ağacı: %82,4
- Yapay Sinir Ağları: %89,6
- Naive Bayes Sınıflandırma: %91,3
- Apriori: 67,5

Bu çalışmada Naive Bayes Sınıflandırma yönteminin diğer yöntemlere göre daha etkili olduğu sonucuna varılmıştır.

11.6. Kullanılan Kütüphane Özelliği

Bildiride kullanılan dil ve kütüphaneden bahsedilmemiştir.

11.7. Kaynak

AKGÜN, Z . (2016). Veri Madenciliği İle Yazılım Hata Tespiti. El-Cezeri Journal of Science and Engineering , 3 (2) , 0-0 . DOI: 10.31202/ecjse.264197

12. Araştırma Ödevi 1, Makale 10

12.1. İncelenen Tez Adı

Video Görüntülerinde Şiddet İçeren Aktivitelerin Lstm Ağı ile Tespiti

12.2. Problem

Gelişmekte olan teknoloji ve internet ağı sayesinde, büyük miktarlarda video verisine ve bu serilerin şiddet içerikli olanlarına kolayca erişilebilmektedir. Şiddet içeren sahnelere sahip videoların etiketlenmesi, güvenlik ve içerik tabanlı video arama sistemleri için önemlidir. Güvenlik kamera sistemleri genellikle şiddeti ve uygunsuz hareketleri tespit etmek için elverişsizdir. Büyük ölçekli bir güvenlik kamera sistemi için, bir operatörün tüm kameraları aynı anda izlemesi imkânsızdır. Öte yandan, video akışı sitelerine yüklenen videoları kontrol edebilen otomatik video değerlendirme ve etiketleme sistemleri için de giderek artan bir ihtiyaç bulunmaktadır.

12.3. Neden Veri Madenciliği?

Geleneksel sinir ağları önceki gözlemlere dayanarak tahmin yapamamaktadır. Bu tür gözlem serilerinden tahmin ve sınıflama yapmak için RNN (Recurrent Neural Network) modelleri geliştirilmiştir. LSTM ağları da bir tür RNN'dir. RNN'lerin asıl amacı ardışık bilgileri kullanmaktır. Geleneksel bir sinir ağına tüm girişlerin (ve çıktıların) birbirinden bağımsız olduğu varsayılmaktadır. RNN'ler bir serinin her ögesi için aynı görevi gerçekleştirdiğinden tekrarlı (recurrent) olarak adlandırılır ve çıktı önceki hesaplamalara bağlıdır. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleğini girdilerin keyfi sıralarını işlemek için kullanabilirler. RNN'leri dizinin gözlemlene değerleri ile ilgili bilgi toplayıp “bellek” taşırlar. RNN'ler teorik olarak çok uzun diziler için kullanılabilirler, ancak pratikte yalnızca birkaç adım geriye dönmekle sınırlıdır.

Gradyan yok olması problemi, ileri beslemeli ağlarda görüldüğü gibi katman eklenmesi durumunda ağı eğitilemez olması durumudur. Bu sorunun çözülebilmesi için LSTM (Long Short-Term Memory), diğer bir deyişle Uzun Kısa Dönem Belleği algoritması 1997 yılında Hochreiter ve Schmidhuber tarafından geliştirilmiştir.

12.4. Nasıl ve Nerede Kullanıldı?

Video serilerinden elde edilen RGB çerçeveleri ile hız ve ivmelenme çerçevelerinden öznitelik çıkarma amacıyla, önceden eğitilmiş GoogleNet derin öğrenme ağı kullanılmıştır. GoogleNet, ImageNet veri kümesinde bulunan milyonlarca görüntü kullanılarak eğitilmiştir. Transfer öğrenme yani önceden eğitilmiş bir model ve LSTM ağları birleştirilerek şiddet içeren videoların sınıflandırılmasında kullanılabilecek bir yaklaşım önerilmiştir. Video serisinin zaman ekseninde hesaplanan 1. ve 2. türevleri ve video çerçeveleri önceden eğitilmiş bir CNN ağına girdi olarak verilmiş ve yüksek düzeyli öznitelikler elde edilmiştir. Video serileri üç kanallı bir derin öznitelik serisine dönüştürülmüştür. Bu derin nitelik serilerinden bir LSTM modeli eğitilmiştir. Önerilen yöntemin test edilmesinde Violent Flow ve Hockey Fight veri kümelerinden faydalanılmıştır. Hockey Fight veri kümesi buz hokeyi karşılaşmaları sırasında kaydedilmiş 1000 farklı videodan oluşmaktadır. Bu videoların yarısı şiddet içeren videolar iken diğer yarısı şiddet içermemektedir. Violent Flow veri kümesinde ise 246 adet kalabalık aktiviteleri içeren videodan oluşmaktadır. Bu veri kümesinde de yine şiddet içeren ve içermeyen örneklerin oranı yarı yarıyadır.

12.5. Başarı Oranı

Violent Flow veri kümesi daha karmaşık görüntüler içermesinden dolayı bu veri kümesi ile yapılan deneylerde daha düşük başarımlar elde edilmiştir. Ayrıca öznitelik çıkarımında önceden eğitilmiş bir ağı kullanmanın sınıflama üzerinde olumlu bir etkisi olduğu sonucuna varılmış.

12.6. Kullanılan Kütüphane Özelliği

GoogleNet'in literatürde yaygın olarak kullanılan diğer (AlexNet VGG-16, VGG-19) modellerinden farkı, inception modüllerine sahip olmasıdır. Inception modülleri kendi içerisinde convolution ve pooling katmanları içermektedir. Konvolüsyonel katmanlar özellikleri saptamak için kullanılır. Pooling Downsampling)/Havuzlama katmanı ise ağırlık sayısını azaltır ve uygunluğu kontrol eder. Inception modülleri içerisinde bulunan convolution katmanları farklı ölçeklerde evrişim ve birleştirme işlemleri yapabilmektedir. GoogleNet'in diğer bir farklılığı ise koşut convolution katmanlarına sahip olmasıdır.

12.7. Kaynak

KEÇELİ, A , KAYA, A . "Video Görüntülerinde Şiddet İçeren Aktivitelerin Lstm Ağı İle Tespiti". Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi 21 (2019): 933-939

13. Araştırma Ödevi 1, Makale 11

13.1. İncelenen Tezin Adı

Zatürre Hastalığının Derin Öğrenme ile Tespiti

13.2. Problem

Enfeksiyon hastalıklarının başında yer alan zatürre hastalığı en sık çocuklarda görülen bir hastalıktır. Teşhis konulabilmesi için akciğer röntgenleri doktorlar tarafından incelenir. Erken teşhis yapılmadığı sürece ölümle sonuçlanabilmektedir. Daha hızlı ve doğru sonuçların alınabilmesi gerekmektedir.

13.3. Neden Veri Madenciliği?

Veri kümesi ESA modellerinden biri olan AlexNet mimarisi kullanılarak öznetelik çıkarımı yapılmıştır. Karar ağacı, doğrusal ayırmacılık analizi, lineer regresyon, destek vektör makineleri, en yakın komşu ve softmax yöntemleri kullanılmıştır. Bu yöntemlerle yapılan karşılaştırmalar sonucu mevcut yöntemlere kıyasla daha iyi sonuç vermiştir.

13.4. Nasıl ve Nerede Kullanıldı?

Bu alandaki ölüm oranı her yıl yaklaşık dünyanın %7'sini etkilemektedir ve bu etki ölümle sonuçlanmaktadır. Zatürre hastalığı, çocuklarda daha fazla görüldüğü için veri kümesi oluşturulurken, bu yaş aralığı tercih edilmiştir. Veri kümesi, "normal ve pnömoni" adında iki kategoriden oluşmuştur. 1583 adet normal, 4266 adet pnömoni görüntü verisi olmak üzere toplam 5849 adet görüntü kümesi kullanıldı. Görüntülerin dosya uzantısı jpeg, derinliği 8 bit ve çözünürlüğü 1448×1056' dır. Veri kümesi, ESA modellerinden biri olan AlexNet mimarisi kullanılarak öznetelik çıkarımı elde edildi.

13.5. Başarı Oranı

Yapılan çalışmada kümenin %30' u test, %70'i eğitim kümesi olarak ayarlanmıştır. Çapraz doğrulama kullanılmamıştır. Transfer öğrenme kullanılmamıştır. Sınıflandırıcı olarak KA, DAA, LR, DVM, EYK ve softmax yöntemleri kullanılmış, sınıflandırma ve öznetelik çıkarımı için ortalama 371,45 saniye zaman harcanmıştır. En başarılı sınıflandırmayı DVM, yani doğrusal ayırmacılık analizi sağlamıştır. %95,8 başarı oranı sağlamıştır.

Tablo 2. TBA kullanılmadan sınıflandırıcı sonuçları.

Sınıflandırıcı	Doğruluk (%)	Duyarlılık (%)	Özgünlük (%)	Hassasiyet (%)	F1 Skor (%)
KA	92,80	91,74	93,91	94,06	92,89
DAA	73,50	96,28	65,72	48,96	64,91
LR	60,71	70,60	57,23	36,70	48,30
DVM	95,83	95,72	95,95	95,96	95,84
eYK	93,59	92,65	94,57	94,69	93,66
SOFTMAX	92,95	88,78	98,12	98,32	93,31

Tablo 3. TBA kullanılarak sınıflandırıcı sonuçları.

Sınıflandırıcı	Doğruluk (%)	Duyarlılık (%)	Özgünlük (%)	Hassasiyet (%)	F1 Skor (%)
KA	92,61	91,66	93,60	93,75	92,69
DAA	72,80	94,79	65,30	48,26	63,96
LR	60,90	70,81	57,38	37,08	48,67
DVM	95,40	95,68	95,10	95,07	95,37
eYK	92,10	91,27	92,97	93,11	92,18
SOFTMAX	92,95	88,78	98,12	98,32	93,31

13.6. Kullanılan Kütüphane Özelliği

AlexNet mimarisi, 64 bitlik Windows 10 işletim üzerinde kurulan Matlab R2017b görüntü işleme yazılımı kullanılarak derlenmiş, kullanılan bilgisayarın özellikleri; Nvidia GForce 2 GB grafik kartı, Intel i5 çekirdekli 2.5 GHz işlemci ve 8 GB ram bellektir. Kütüphane ve özelliği verilmemiştir.

13.7. Kaynak

TOĞAÇAR, M., ERGEN, B., & SERTKAYA, M. E. (2019). Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti. *Firat University Journal of Engineering*, 31(1).

14. Araştırma Ödevi 1, Makale 12

14.1. İncelenen Tezin Adı

Dış Ortam Görüntülerindeki İnsan Hareketlerinin Hibrit Derin Öğrenme Yöntemleri Kullanarak Sınıflandırılması

14.2. Problem

Görüntü verileri içerisindeki kişilerin tespit edilmesi ve bu kişilerin gerçekleştirdiği hareketlerin belirlenmesi.

14.3. Neden Veri Madenciliği?

Güvenlik sistemleri, sürücüsüz araçlar, robotik sistemler vs. Önerilen modelin yeterli derecede eğitim ve parametre düzenlemesinden sonra hareket tespit sistemleri, insan takip sistemleri, otonom araçlar için sürücü destek sistemleri gibi birçok alanda kullanılması sağlanabilir. İnsan hareketi tanımlı yapılarırken çeşitli seviyelerde soyutlamalar kullanılmaktadır. Örneğin sağ ayağın önde olması gibi atomik eylemlerin bir döngü içerisinde gerçekleşmesi ile bir hareket meydana gelmektedir. Genel tanımlı ile insan vücudunun ardışık manevralarının sonucunda ortaya çıkan durum olarak ifade edilebilir. Hareket tespiti için öncelikle kişilerin tespitinin gerçekleştirilmesi ve tüm görüntüden ayrılması gerekmektedir. Tespit edilen kişi görüntüleri üzerinde özellik çıkarımı ve sınıflandırma yöntemleri ile bu kişilerin hareketlerine dair anlamlı çıkarımlar yapılabilir.

14.4. Nasıl ve Nerede Kullanıldı?

K-nn yakın komşu araması, KSA Sınıflandırma Modeli (CNN) kullanılmıştır. Diğer görüntü boyutları ve veri setlerinde önerilen KSA modeli, diğer modellere göre daha yüksek sonuç vermiştir. Araçlara ya da benzer donanımlara yerleştirilebilecek bir görüntüleme sistemi ile insan hareketlerinin otomatik tespit edilmesi sağlanabilir. Kara yollarında araç hareket halindeyken gerçek görüntülerden insanların tespiti ve buna ilaveten hareketlerinin gerçek zamanlı algılanması, araçlarda erken uyarı sistemlerinin oluşturulması gibi birçok uygulama alanı için büyük önem arz etmektedir. Google Street view uygulamasından toplam 1000 adet görüntü alınarak veritabanı oluşturulmuştur. Yolo ile tespit edilerek 2000 kişi görüntüsü elde edilmiştir. Yükseklik, genişlik ve renk olarak 3 farklı boyut için çalışmalar yapılmıştır. Çalışmada, 234 tane ayakta duran, 198 tane oturan, 249 tane sola yönelen ve 228 tane sağa yönelen olmak üzere toplamda 909 tane insan hareket verisi bulunmaktadır.

14.5. Başarı Oranı

Bölümleme yapıları eğitim, doğrulama ve test olmak üzere; %80-%10-%10, %70-%15-%15 ve 50-%25-%25 şeklinde seçilmiştir. Veri seti %80 eğitim, %10 doğrulama ve %10 test şeklinde bölünmüştür. KSA modelinin eğitimi 100 adımda yapılmış olup 30. Adımda eğitim tamamlanmış ancak doğrulama tam olarak başarıya ulaşamayıp 80. Adımda model başarısı %85.71 olarak ölçülmüştür.

14.6. Kullanılan Kütüphane Özelliği

Yolo, Konvolüsyonel Sinir Ağları ve KSA'ya ait yöntemler kullanılmıştır. Platform olarak Google Street view platformu kullanılmıştır.

- Verilen modeli birden fazla konum ve ölçekte bir görüntüye uygular.
- Test zamanında tüm görüntüye bakar ve tahminleri görüntüdeki küresel bağlam tarafından bilgilendirir.
- R-CNN gibi tek bir görüntü için binlerce sistem gerektiren sistemlerin aksine tek bir değerlendirilmesiyle tahminlerde bulunur. Bu onu son derece hızlı yapar.
- R-CNN'den 1000 kat, Fast R-CNN'den 100 kat hızlı yapar.

14.7. Kaynak

ALGUR, Ö., TÜMEN, V., & YILDIRIM, Ö. (2018). Dış Ortam Görüntülerindeki İnsan Hareketlerinin Hibrit Derin Öğrenme Yöntemleri Kullanarak Sınıflandırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 30(3), 121-129.

15. Araştırma Ödevi 1, Makale 13

15.1. İncelenen Tezin Adı

Parkinson Hastalığını Yüksek Doğrulukla Tespit Etmek İçin Derin Öğrenme Algoritmasının Kullanımı

15.2. Problem

Parkinson hastalığının erken teşhisi için kullanılacak yöntem ve çalışmalar.

15.3. Neden Veri Madenciliği?

Çalışma, örnekleme ve önleme olarak ön işleme adımlarıyla YSA kullanarak %95' in üzerinde başarıya ulaşmakta ve %70 oranında tren olarak bölünerek ve %30 oranında test etmektedir. Bu nedenle, yapay sinir ağı algoritması veya herhangi bir ön işleme aşaması olmadan diğer makine öğrenme algoritmalarının herhangi bir algoritması kullanılarak Parkinson hastalığı teşhisi için %95'in üzerinde sonuçlar elde etmek zordur. Bu nedenle, sinir ağı sınıflandırıcısının yapısal değişikliklere ihtiyacı vardır.

15.4. Nasıl ve Nerede Kullanıldı?

Softmax, destek vektör makinesi, karar ağacı, MLP, RBF, ANN, FKNN, PCA, YSA yöntemleri kullanılmıştır. Keras tarafından Python'un Tensorflow kütüphanesinde inşa edilen bir FFNN'dir. Çalışmadaki mimari çıktı katmanı olarak ReLu ile bir girdi katmanı, iki gizli katman ve softmax fonksiyonundan oluşur. Çalışmadaki derin öğrenme mimarisi 1 giriş katmanı, 2 gizli katman ve 1 çıkış katmanı, aktivasyon fonksiyonu olarak ReLu, sınıflandırma fonksiyonu olarak softmax'dan oluşmaktadır. Dört katmana ek olarak, adam işlevi bir optimizasyon algoritması olarak kullanılır.

15.5. Başarı Oranı

Veri seti, 188 hastanın PD hastası ve geri kalanının sağlıklı olduğu 252 kişiden toplam 756 örnek toplanmıştır. Denenen yöntemlerin hepsinde başarı oranı doğal olarak farklı çıkmıştır. Parkinson Hastalığı veri seti, Python programlama dilinin Tensorflow derin öğrenme kütüphanesinde Keras API kullanılarak %80 doğruluk oranında %80-20 tren testi veri bölümü ve 30 çağ sayısı ile sınıflandırılmıştır.

15.6. Kullanılan Kütüphane Özelliği

Python'ın TensorFlow kütüphanesi kullanılmıştır.

- TensorFlow hızlı derleme yapabilir.
- Esnek yapısı sayesinde, tek bir API ile platform farketmeksizin hesaplamaları, bir veya birden fazla CPU, GPU kullanarak deploy etmemize olanak sağlar.
- TensorBoard ile görselleştirme yapabilmektedir
- Veri ve model paralelliği sağlar.

15.7. Kaynak

GEMCI, F., & IBRIKCI, T. (2019). USING DEEP LEARNING ALGORITHM TO DIAGNOSE PARKINSON DISEASE WITH HIGH ACCURACY. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 22, 19-25.

16. Araştırma Ödevi 1, Makale 14

16.1. İncelenen Tezin Adı

Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması

16.2. Problem

Türkçe doğal dil işleme çalışmalarının azlığı ve bu alandaki özellik çıkarma yöntemlerinin limitli olması sebebiyle, kelimelerin semantik değerlerinin önceden eğitilmiş Word2Vec ağı ile sınıflandırmaya katılabilmesi KSA modellerinin doğruluk değerlerini arttırmıştır.

16.3. Neden Veri Madenciliği?

Doğal Dil İşleme, Bilgi Getirimi ve Makine Öğrenmesi gibi alanlarda kullanılırken Türkçe metinlerin sınıflandırılmasında veri kümesinin oldukça kısıtlı olması, haber veri kümelerinin sıklıkla kullanılması, haberlerin konularına göre kategorize edilmesi, anlık olarak üretilen haber metinlerini yönetmek gerekli bir hale gelmiştir.

16.4. Nasıl ve Nerede Kullanıldı?

Çoğunlukla görüntü ve ses işleme çalışmalarında kullanılan KSA, metin sınıflama uygulamalarında makine öğrenmesi ve veri madenciliği algoritmalarına kıyasla daha iyi performans göstermiştir. Bazı modellerde yüksek performans göstermesine karşın aşırı uyum(overfitting) oluşmuştur. Birinci KSA modelinde gövdelenmiş veride %93,3, ham veride ise %90,1 doğruluk değeri alınmıştır. İkinci KSA modeli ise gövdelenmiş veri üzerinde %90,8, ham veri üzerinde %90,2 doğruluk performansı göstermiştir. İki KSA modeli de gövdelenmiş veri için ham veriye kıyasla daha yüksek doğruluk yüzdeleri üretmiştir.

16.5. Başarı Oranı

En iyi sonuç %93,3 ile birinci KSA modelinin Zemberek ile gövdelenmiş veri kümesinde elde edilmiştir.

16.6. Kullanılan Kütüphane Özelliği

Her iki KSA modeli de Python dili kullanılarak, Tensorflow arayüzünün Keras Kütüphanesi yardımıyla kodlanmıştır. Tensorflow paralel ve dağıtık olarak makine öğrenmesi alanında hesaplama arayüzü ve çerçevesi sağlarken, Keras bu katman üzerinde modellerin kolayca oluşmasını sağlamaktadır.

16.7. Kaynak

Çiğdem, A. C. I., & ÇIRAK, A. Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması. *Bilişim Teknolojileri Dergisi*, 12(3), 219-228.

17. Araştırma Ödevi, Makale 15

17.1. İncelenen Tezin Adı

Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini

17.2. Problem

Derin öğrenme ile cinsiyet tahmini yapılarak birçok farklı alanda farklı amaçlarla kullanılabilir.

17.3. Neden Veri Madenciliği?

Derin öğrenme algoritmalarını kullanarak, mağazalarda ürünlerle ilgilenen insanların, bir mağazada alışveriş yapmayı tercih eden insanların, alışveriş merkezlerindeki müşterilerin, sosyal medya platformlarındaki kullanıcıların, siyasilerin mitinglerindeki kişilerin başta olmak üzere, görüntüleri üzerinden cinsiyet tahmini demografisini tespit etmeye yardımcı olacak bilgiler ve yöntemler sunmaktır. Aynı zamanda veri sayısı artırmanın başarıma olan etkisi vurgulanmaktadır.

17.4. Nasıl ve Nerede Kullanıldı?

Fotoğraflardan derin öğrenme ile cinsiyet tahmini yapabilmek için veri seti olarak Vikipedi fotoğraflarından derlenen 3170 görüntülük dosya oluşturulmuştur. Veri seti içindeki resimler cinsiyetlerine ve yaşlarına göre etiketlenmiştir. Kadın ve erkek olmak üzere 2 sınıf oluşturulmuştur. Her biri veri görüntüsü yaş ve cinsiyet bilgisine göre etiketlenmiştir. Eğitim veri seti için 3170 görüntü, test veri seti için ise toplam görüntü içinden rasgele seçilen 318 görüntü kullanılmıştır. Uygulama, Ubuntu işletim sistemi üzerinde Python diliyle yazılmıştır. Sınıflandırma için derin öğrenme modellerinden Alex Net ve optimize edilmiş VGG-16 gibi derin öğrenme algoritmaları kullanılmıştır. Bütün algoritmalar 'PyCharm' Python geliştirme ortamında çalıştırılmıştır. Yeni geliştirilen bir derin öğrenme modeli ile Alex Net ve optimize edilmiş VGG-16 parametrelerinin kıyaslaması yapılmıştır. Yeni modelin VGG-16'dan farklı olarak, Alex Net gibi DroupOut katmanına sahip olmadığı da görülmektedir. DropOut katmanı, ağ içindeki bazı bağlantıların kaldırılmasıyla eğitim performansını artırmaktadır. Öğrenememe ve ezberleme problemlerine çözüm için kullanılmaktadır. Relu aktivasyon fonksiyonu da aynı problemlere çözüm için modellerde bulunmaktadır. Uygulama bölümündeki bütün modeller için keras, tensorflow ve matplotlib kütüphaneleri kullanılmıştır.

17.5. Başarı Oranı

Hazırlanan verisetine göre en iyi performansı VGG-16 modeli vermiştir. Geliştirilen model ise AlexNet'ten daha iyi VGG-16'dan daha kötü sonuç vermiştir. AlexNet ile başarı %65.62, geliştirilen model ile başarı %72.20'dir. VGG-16 modeli ile uyumlu olduğu görülen veri setine göre, 150 iterasyon ile çalışan modelin %99,41 başarı elde edilmiştir.

17.6. Kullanılan Kütüphane Özelliği

Ubuntu Linux işletim sistemi üzerinde, Python dilinde kodlanmıştır. Python seçilmesinin sebebi hızlı olması, topluluk desteğinin olması ve Linux işletim sisteminde Python'ın kurulu gelmesidir. Python' da kullanılan kütüphaneler Keras ve Tensorflow'dur. Kullanılan platform PyCharm'dır.

17.7. Kaynak

Gündüz, G., & Cedimoğlu, İ. H. (2019). Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini. *Sakarya University Journal of Computer and Information Sciences*, 2(1), 9-17.