

T.C
KARADENİZ TEKNİK ÜNİVERSİTESİ İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ
YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

TEZ 4000 BİTİRME ÇALIŞMASI

DERİN ÖĞRENME DESTEKLİ OTOMATİK ÇÖP SINIFLANDIRAN GUI

Hazırlayan: Mehmet Anıl TÜRKMEN

Trabzon-2024

T.C
KARADENİZ TEKNİK ÜNİVERSİTESİ İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ
YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

TEZ 4000 BİTİRME ÇALIŞMASI

DERİN ÖĞRENME DESTEKLİ OTOMATİK ÇÖP SINIFLANDIRAN GUI

Hazırlayan: 404060 Mehmet Anıl TÜRKMEN

Danışman: Prof. Dr. Bünyamin ER

Trabzon-2024

İÇİNDEKİLER

İÇİNDEKİLER	II
ŞEKİLLER DİZİNİ	IV
GİRİŞ	1

BİRİNCİ BÖLÜM

1. ATIK YÖNETİMİ	2
1.1. Önemi ve gerekçesi	3
1.2. Çevresel Etkileri	3

İKİNCİ BÖLÜM

2. ÖRNEK ÇALIŞMALAR	5
2.1. “Önerilen Derin Öğrenme ve Makine Öğrenmesi Tabanlı Hibrit Model ile Çevresel Atıkların Sınıflandırılması”	5
2.2. “Evsel Atıkların Derin Öğrenme Teknikleri ile Ayırıştırılması”	6
2.3. “Derin Öğrenme Yaklaşımlarıyla Çevresel İzlemeye Yönelik Çok-Sınıflı Sınıflandırma” ...	6

ÜÇÜNCÜ BÖLÜM

3. VERİ SETİ VE YÖNTEM	7
3.1. Veri Seti	7
3.2. Programlama Dili	7
3.3. Kullanılan Kütüphaneler	7
3.4. Proje Yapım Aşamaları	9
3.4.1. Veri Setinin Yükleme ve Hazırlama	9
3.4.2. Veri Setindeki Görsel Dosyaların Türlerini ve Boyutlarını Analizi	10
3.4.3. Veri Setindeki Sınıfa Ait Görsel Sayısını İnceleme	10
3.4.4. Veri Setinin Sınıflarının Görsel Dağılımını Görselleştirme	11
3.4.5. CNN Model Oluşturma	12
3.4.6. CNN Modelini Eğitim ve Test Etme	13
3.4.7. Eğitilen Modelin Test Grafiklerinin İncelenmesi	15

DÖRDÜNCÜ BÖLÜM

4. KULLANICI ARAYÜZÜ OLUŞTURMA	17
4.1. Kullanılan Kütüphaneler	17

4.2. Arayüzün Kodlanması.....	18
4.2.2. Kullanıcı Arayüzü (GUI)	18
4.2.3. Ana Uygulama Yapısı.....	18
SONUÇ.....	20
YARARLANILAN KAYNAKLAR.....	21

ŞEKİLLER DİZİNİ

Şekil 1:	Atık Yönetimi Diyagramı	2
Şekil 2:	Atık yönetimi Hiyerarşisi	3
Şekil 3:	Sürdürülebilir gelecek için atık yönetiminin önemi	4
Şekil 4:	ESA mimarisine ait karmaşık matrisler	5
Şekil 5:	DenseNet-121 mimarisi (Li, Shen, Zhou, Wang, & Li, 2020).....	6
Şekil 6:	Çöp Türlerine ait görseller	7
Şekil 7:	VGG-16 ağ mimarisi (Ferguson, Ak, Lee, & Law, 2017)	9
Şekil 8:	Veri Setinin Yüklenmesi ve Dosyaların listenmesi.....	10
Şekil 9:	Veri setindeki görsellerin sayılarının incelenmesi	11
Şekil 10:	Görsellerin sayılarının grafiksel dağılımları	11
Şekil 11:	VGG16 modeli ve katmanlarının görünümü.....	13
Şekil 12:	Eğitim, Test ve Geçerleme yollarının belirlenmesi.....	14
Şekil 13:	Eğitilen Modelin Son epoch değerleri.....	14
Şekil 14:	Modelin Test değerlendirmesi sonucu	15
Şekil 15:	Modelin Doğruluk grafiği	15
Şekil 16:	Modelin Loss grafiği.....	16
Şekil 17:	GUI Açılış ekran görüntüsü	17
Şekil 18:	Görüntü Seçim Ekranı.....	18
Şekil 19:	Seçilen Görüntünün Türünün Çıktı Sonucu	19

GİRİŞ

Bu tez, bireyin çevreyi korumaya yardımcı olmak ve atık yönetimi süreçlerine önemli bir katkı sağlamak amacıyla geliştirilen otomatik çöp sınıflandırma sistemi ile atıkların verimli bir şekilde yönetilmesini hedeflemektedir. Projenin temelinde, Python yazılım dili kullanılarak veri bilimi, derin öğrenme ve makine öğrenmesi teknikleriyle çöp sınıflandırma modelinin oluşturulması yer almaktadır. Çeşitli çöp türlerini 6 farklı sınıfa (karton, cam, metal, kağıt, plastik ve genel çöp) içeren görsel veri setleri toplanacak ve bu veriler kullanarak bir sınıflandırma modeli eğitilecektir. Eğitilen model, bir görüntüyü analiz ederek kağıt görseli olduğunu tespit ettiğinde, bu görselin hangi çöp türüne ait olduğunu tahmin edebilecektir. Bunu yaparken Pythonun Tkinter kütüphanesi kullanılarak kullanıcı ara yüzü oluşturulması hedeflenmiştir.

Hemen hemen her gün bir sürü farklı türde çöp atılıyor. Farklı çöp türleri farklı işlemlere tabi tutulur. Ancak, önemli miktarda atık uygun bir şekilde işlenmemekte ve bertaraf edilmemektedir. ABD Çevre Koruma Ajansı'na (EPA) göre, ABD'deki elektronik atıkların sadece %12,5'i geri dönüştürülmektedir. Çöpler genellikle yığılmakta veya gömülmekte, bu da olumsuz çevresel etkilere neden olmaktadır; örneğin, kötü koku yayılması, toprağın kirlenmesi, toksik maddelerin yeraltı sularına sızması ve atmosfere zarar vermesi gibi. Yanlış sınıflandırılan çöplerin getirdiği zararları önlemek adına, farklı çöp türlerini tanımlamaya yönelik yüksek düzeyde görüntü sınıflandırma yeteneklerine sahip bir makine öğrenimi modeli oluşturmaya karar verdik. Amacımız, kullanıcının çöp görüntüsünü yükleyebileceği bir uygulama geliştirmek ve modeli, görüntü işleme yöntemleriyle çöpün türünü tanımlamak.

Bu sorunu çözmek için öncelikle 6 farklı sınıfa ait 5840 görüntü içeren Kaggle'dan bir veri seti topladık. Ardından, görüntüleri analiz etmek için Veri Analizi teknikleriyle işlemlere devam ediyoruz bunlar: Görsellerin tür isimleri, sayıları, görüntü boyutları, özelliklerinin boyutları, etiketlerinin boyutları, sınıflandırılması vb. gibi bakılıyor. Sonrasında derin öğrenme teknikleriyle sınıflandırma işlemi gerçekleştiriliyor. Daha iyi bir CNN modeli oluşturmak ve modeli kullanıcı ara yüzü için tahmin yöntemi olarak entegre etmek için daha fazla zaman harcamayı amaçlıyoruz.

BİRİNCİ BÖLÜM

1. ATIK YÖNETİMİ

Atık yönetimi, atıkların toplanması, taşınması, işlenmesi, bertaraf edilmesi ve geri dönüşümü gibi süreçleri içeren bir disiplindir. Bu süreçlerin etkin bir şekilde yürütülmesi çevre koruma, insan sağlığı ve ekonomik açıdan önemlidir. Atık yönetimi, atıkların çevreye zarar vermeden kontrol altına alınmasını ve kaynakların verimli kullanılmasını hedefler.

Atık yönetimi, atıkların türlerine göre farklı yöntemlerle ele alınır. Geri dönüşüm, atıkların tekrar kullanılabilir hale getirilmesini sağlayarak kaynakların korunmasına katkıda bulunur. Bertaraf, atıkların güvenli bir şekilde yok edilmesini ve çevreye zarar vermeden bertaraf edilmesini sağlar.

Atık yönetimi süreçlerinde atıkların sınıflandırılması, geri dönüşüm potansiyellerinin belirlenmesi, atıkların toplanması ve taşınması gibi adımlar önemlidir. Bu süreçlerin etkin bir şekilde yürütülmesi için toplumun bilinçlendirilmesi, geri dönüşüm altyapısının güçlendirilmesi ve yasal düzenlemelerin uygulanması gerekmektedir.

Atık yönetimi, sürdürülebilir bir çevre politikasının temel unsurlarından biridir ve atıkların doğru şekilde yönetilmesi gelecek nesillere temiz bir çevre bırakılmasını sağlar. Bu nedenle atık yönetimi, toplumun ve endüstrinin ortak sorumluluğu olarak görülmektedir.

Şekil 1: Atık Yönetimi Diyagramı



Kaynak: (Arçev çevre ve danışmanlık, 2019, <https://www.arcev.com.tr/atik-yonetimi-nedir/>)

1.1. Önemi ve gerekçesi

Atık yönetiminin önemi ve gerekçesi, çevre mühendisliği eğitimi, çevre dostu konaklama işletmeleri, tıbbi atık yönetimi konularında incelenmektedir. Çevre mühendisliği, sağlıklı içme suyu temini ve atık uzaklaştırma gibi alanlarda çalışarak çevresel sorunlara çözüm getirmektedir. Konaklama işletmelerinde çevreye duyarlı uygulamaların önemi vurgulanmakta; otellerin çevreye duyarlı şekilde inşa edilmesi, atık yönetiminin iyi yapılması ve geri dönüşüme önem verilmesi gerektiği belirtilmektedir. Tıbbi atık yönetimi ise sağlık tesislerinde çalışan stajyer öğrencilerin tıbbi atıkların potansiyel tehlikeleri konusunda bilgi düzeylerinin değerlendirilmesiyle ele alınmaktadır. Bu çalışmalar, çevrenin korunması, atık yönetiminin etkinliği ve çevreye duyarlı uygulamaların yaygınlaştırılmasının önemini vurgulamaktadır.

Şekil 2: Atık yönetimi Hiyerarşisi



Kaynak: Model Çevre, Atık Yönetimi Hiyerarşisi, <https://www.modelcevre.com/atik-yonetimi-hiyerarşisi.html>

1.2. Çevresel Etkileri

Atık yönetimi, atıkların insan ve çevre sağlığını riske atmadan işlenmesi ve bertaraf edilmesi amacıyla gerçekleştirilen faaliyetleri kapsar. Atıkların yeniden kullanılması, geri dönüştürülmesi ve kazanılması gibi uygulamalar, çevresel unsurlar üzerinde önemli etkilere sahiptir. Sıfır atık yönetimi kapsamındaki uygulamaların çevre ve insan sağlığı açısından değerlendirilmesi, çevre etiği perspektifinden ele alınabilir.

Endüstriyel işletmeler, üretim prosesi, kapasite, ekipman ve hammadde özelliklerine göre değişen türde ve miktarda atık ortaya çıkarmaktadır. Ancak endüstriyel atık yönetiminin sağlıklı bir şekilde uygulanması konusunda sorunlar yaşanmaktadır. Denetim mekanizmalarının ve mevzuatın

yaptırım gücünün yeniden sorgulanması, endüstriyel atık yönetiminin daha sağlıklı uygulanabilmesi için önemlidir.

Çevre mühendisliği eğitiminde, faaliyetlerin ekosistem üzerindeki etkilerini anlamak için Çevresel Etki Değerlendirmesi (ÇED), Çevresel Risk Değerlendirmesi (ÇRD) ve Yaşam Döngüsü Analizi (YDA) gibi araçların kullanılması önemlidir. Böylece mühendislik uygulamalarının ekosisteme eküresel ölçekte etkilerinin bütüncül olarak değerlendirilmesi mümkün olabilir.

Ambalaj atıklarının ayrı toplanmaması, hem ekonomik hem de çevresel açıdan ciddi zararlara yol açmaktadır. Katı atık depolama alanları, sera gazlarının önemli kaynaklarından biri olduğu için küresel ısınmanın sebepleri arasında gösterilmektedir. Entegre Atık Yönetimi kapsamında ambalaj atıklarının ayrı toplanması, çevresel etkilerin azaltılmasına katkı sağlayabilir.

Sonuç olarak, atık yönetiminin çevresel etkileri, üretim ve tüketim faaliyetlerinin türüne göre değişmekle birlikte, genel olarak atıkların yönetimindeki eksiklikler nedeniyle ortaya çıkmaktadır. Sıfır atık yönetimi, temiz üretim, entegre atık yönetimi gibi yaklaşımlar ve ÇED, ÇRD, YDA gibi araçlar kullanılarak atık yönetiminin çevresel etkilerinin azaltılması mümkündür.

Şekil 3: Sürdürülebilir gelecek için atık yönetiminin önemi



Kaynak: Cadem Digital, Sürdürülebilir Gelecek, <https://cademdigital.com.tr/surdurulebilir-gelecek-icin-elektronik-atik-yonetiminin-onemi/>

İKİNCİ BÖLÜM

2. ÖRNEK ÇALIŞMALAR

2.1. “Önerilen Derin Öğrenme ve Makine Öğrenmesi Tabanlı Hibrit Model ile Çevresel Atıkların Sınıflandırılması”

Bu çalışmada yapay zeka teknikleri ile atık ayrıştırma tespitinde AlexNet, GoogLeNet, ResNet50, DenseNet201, ShuffleNet, SqueezeNet mimarileri ile sonuçlar elde edilmiştir. Sonraki aşama da derin mimarilerden elde edilen özellik haritaları destek vektör makineleri (DVM), k-en yakın komşu (KNN) ve karar ağaçlarında (KA) sınıflandırılmıştır. Mimarilerin elde ettikleri sonuçlar karşılaştırılarak bu problem için en iyi mimari seçilmiştir. Deneyler sonucunda önermiş olduğumuz hibrit model %85.94 doğruluk değeri elde etmiştir.

Şekil 4: ESA mimarisine ait karmaşık matrisler

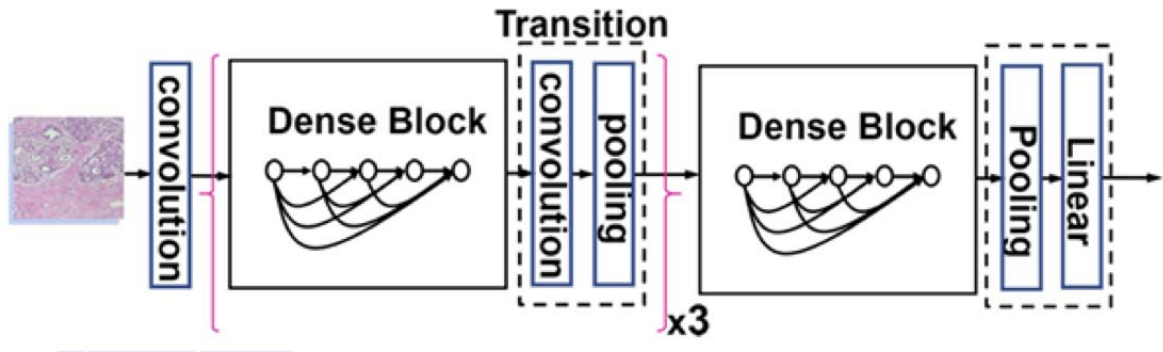
AlexNet							GoogLeNet						
1	73	2		6			1	78			3		
2		86	8		5	1	2		77	13	1	9	
3		10	69	1	1	1	3	1	4	70	3	4	
4	4	4	1	103	1	6	4	6	1	1	103	1	7
5	2	15	5	9	61	4	5	1	8	6	13	64	4
6	1	1	1	7	2	15	6	1	1	2	10	1	12
	1	2	3	4	5	6		1	2	3	4	5	6
ResNet50							DenseNet201						
1	75		1	5			1	78	1		2		
2		86	6		8		2		91	3		5	1
3		2	73		6	1	3		3	78			1
4	5	1	1	106		6	4	3		1	110		5
5	1	10	8	11	60	6	5	2	9	3	10	62	10
6	3	1	3	7	1	12	6	1		3	8		15
	1	2	3	4	5	6		1	2	3	4	5	6
ShuffleNet							SqueezeNet						
1	72			6	3		1	77			3	1	
2		83	6	2	9		2	1	77	13	1	8	
3		7	66	2	4	3	3	1	6	72	2		1
4	4	3	1	105	2	4	4	7	2	3	102		5
5	3	14	7	10	51	11	5	3	20	13	9	48	3
6	5		5	9		8	6	1		4	9	1	12
	1	2	3	4	5	6		1	2	3	4	5	6

Kaynak: Hibrit Model ile Çevresel Atıkların Sınıflandırılması,
<https://dergipark.org.tr/tr/download/article-file/2878435>

2.2. “Evsel Atıkların Derin Öğrenme Teknikleri ile Ayırıştırılması”

Bu çalışmada ise canlı yaşamı ve doğanın korunması açısından büyük öneme sahip olan atık ayırıştırılmasının derin öğrenme ile tespit edilip ayırıştırılması için CNN kullanıldı. Bu kapsamda çalışmamızda Kaggle’den alınan bir veri kümesi kullanılarak evsel atıklar olan karton, cam, kâğıt ve plastikten oluşan 3773 adet görüntü kullanıldı. Alınan veri setinde eğitim sınıfı için toplam görüntülerin %70’i, test sınıfı için de toplam görüntülerin %30u ayrılmıştır. Akabinde çalışmamızda, CNN kapsamında ResNet-50, DenseNet-121, Inception-V3, VGG-16 mimarileri evsel atıkların ayırıştırılmasının tespitinde uygulanmıştır.

Şekil 5: DenseNet-121 mimarisi (Li, Shen, Zhou, Wang, & Li, 2020)



Kaynak: Evsel Atıkların Derin Öğrenme Teknikleri ile Ayırıştırılması,

2.3. “Derin Öğrenme Yaklaşımlarıyla Çevresel İzlemeye Yönelik Çok-Sınıflı Sınıflandırma”

Bu çalışmada, UC Merced ve WHU-RS19 veri setlerinden uzaktan algılama sahne sınıflandırma üzerine AlexNet, DenseNet, ResNet, VGG derin öğrenme modellerinin performansı analiz edilmiştir. Test aşamasında kullanılan modellerin sınıflandırma sonuçları değerlendirme ölçütleri ile incelenmiştir. DenseNet201 modelinin her iki veri setinde de diğer modellere göre daha yüksek performans gösterdiği tespit edilmiştir. Elde edilen sonuçlar literatürdeki diğer çalışmalarla karşılaştırılmış, özellikle UC Merced veri setinde %98.81 genel doğruluk oranıyla DenseNet201 modelinin öne çıktığı belirlenmiştir. Benzer arazi kullanım sınıfları için en iyi performans gösteren algoritmanın sınıflandırma doğruluğu incelenmiş ve sınıflandırma sonuçları kesinlik, duyarlılık ve F1 skoru ölçütleriyle değerlendirilmiştir. Ayrıca, WHU-RS 19 veri setinde eğitilen modellerin UC Merced veri setindeki testleri ve tam tersi yapılarak model transfer edilebilirliği araştırılmıştır.

ÜÇÜNCÜ BÖLÜM

3. VERİ SETİ VE YÖNTEM

3.1. Veri Seti

Bu çalışma gerçekleştirilirken kaggle.com üzerinden görsel veri görselleri temin edilmiştir. Kullanılan Veri Setinin ismi “Garbage Classification” ait görseller kullanmıştır. Çöp Sınıflandırma Veri Kümesi 6 sınıflandırma içerir: Karton (892), cam (1004), metal (801), kağıt(1051), plastik (1347) ve çöp (745).

Şekil 6: Çöp Türlerine ait görseller



Bu veri setini seçmemizin nedeni, veri setinin yeterince büyük veri içermesidir. Ve büyük veri seti ile eğitim sırasında yetersiz ve fazla öğrenme gibi sorunları önleyecektir. Ayrıca görüntü, Makinenin görüntüdeki özellikleri kolayca bulabilmesi için temizdir, Yakınsama süresini azaltacaktır.

3.2. Programlama Dili

Python, zengin kaynak kitaplıklarına, çerçevelere, esnekliğe, platform bağımsızlığına ve geniş bir topluluğa sahip genel amaçlı bir dildir. Güçlü genişleme yetenekleridir. Python ayrıca özlü ve okunabilir bir kod sunar.

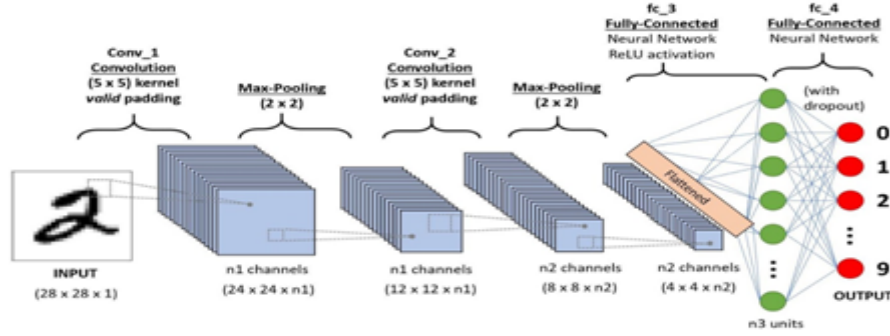
3.3. Kullanılan Kütüphaneler

- Numpy : Programlamada matris hesaplaması yapmak için kullanılır. "numpy", sayısal hesaplamalar ve veri işleme için oldukça yaygın olarak kullanılan bir kütüphanedir. Bu projede "numpy" kütüphanesi, veri işleme ve dönüşümleri için kullanılmıştır.

Özellikle, görüntü verilerinin sayısal temsilini işlemek, matematiksel işlemler yapmak veya veri manipülasyonu için "numpy" kütüphanesi sıkça kullanılır.

- Matplotlib: Veri görselleştirmesi için kullanılan bir Python kütüphanesidir. Özellikle derin öğrenme veya makine öğrenimi projelerinde, eğitim sürecinin doğruluk, kayıp gibi metriklerle izlenmesi ve sonuçların görsel olarak değerlendirilmesi önemlidir. Bu tür görselleştirmeler, model performansının anlaşılmasına ve iyileştirilmesine yardımcı olabilir. Bu projede verilerin görsel olarak temsil edilmesi, sonuçların grafiklerle gösterilmesi ve eğitim sürecinin izlenmesi amacıyla kullanılmıştır.
- Tensorflow ve Keras: Google tarafından geliştirilen açık kaynaklı bir makine öğrenimi kütüphanesidir. Derin öğrenme ve genel olarak makine öğrenimi alanında kullanılır. Grafiksel hesaplama yapısı sayesinde karmaşık matematiksel işlemleri optimize ederek büyük veri setlerinde yüksek performans sağlar. TensorFlow, hem araştırmacılar hem de endüstriyel uygulamalar için yaygın olarak kullanılan bir araçtır. Bu projede derin öğrenme modelinin oluşturulması, eğitilmesi, değerlendirilmesi ve görüntü verilerinin işlenmesi için kullanılmıştır. Özellikle, önceden eğitilmiş VGG16 modeli ile transfer öğrenme uygulanmış ve veri artırma teknikleri kullanılarak modelin performansı artırılmaya çalışılmıştır.
- PIL: görüntü işleme için oldukça yaygın bir Python kütüphanesidir. Bu projede muhtemelen görüntü verilerini yüklemek, işlemek, dönüştürmek veya görselleştirmek için "PIL" veya "Pillow" kullanılmış olabilir.
- CNN: CNN (Evrişimli Sinir Ağı), bu projede çöp sınıflandırma görevini yüksek doğrulukla gerçekleştirmek için kullanılmıştır. CNN'in görüntü özelliklerini otomatik olarak çıkarma yeteneği ve yüksek performansı, bu tür bilgisayarla görme problemleri için ideal bir seçim olmasını sağlar. Projede VGG16 gibi önceden eğitilmiş bir model kullanılarak transfer öğrenme uygulanmış ve çöp sınıflandırma görevi başarılı bir şekilde gerçekleştirilmiştir.

Şekil 7. VGG-16 ağ mimarisi (Ferguson, Ak, Lee, & Law, 2017)



Kaynak: Evsel Atıkların Derin Öğrenme Teknikleri ile Ayrıştırılması

- Torch ve Torchvision: Kütüphaneleri, derin öğrenme modelleri ile çalışmak ve görüntü verilerini işlemek için kullanılır. Projenin temel işlevselliği, bu kütüphaneler aracılığıyla sağlanmaktadır.
- os : kütüphanesi bu projede dosya işlemleri yapmak için kullanılmıştır. Özellikle veri setinin dizin yapısını belirlemek, dosya ve klasörleri listelemek, dosya yollarını birleştirmek gibi işlemler için "os" kütüphanesi tercih edilmiştir. Bu tür dosya işlemleri, veri setinin yapısını inşa etmek, veriye erişmek ve işlemek için sıkça kullanılan işlemdir. Bu nedenle, projenin dosya ve dizin düzeni üzerinde kontrol sağlamak için "os" kütüphanesi tercih edilmiştir.

3.4. Proje Yapım Aşamaları

3.4.1. Veri Setinin Yükleme ve Hazırlama.

- Veri Setinin Dizin Yolunu Belirtme: dataset_path değişkeni, kullanıcının veri setinin bulunduğu dizinin yolunu içerir. Bu, işlem yapılacak veri setinin konumunu belirtir.
- Dosyaları Listeleme: os.listdir(dataset_path) işlevi, belirtilen dizindeki tüm dosyaları ve dizinleri listeler. Bu, veri setinde bulunan tüm öğelerin adlarını içeren bir liste döndürür.
- Dosya İsimlerini Yazdırma: Döngü, her bir dosyanın adını (file değişkeni) alır ve ekrana yazdırır. Bu adım, belirli bir veri setinin içeriğini kontrol etmek veya dosya adlarını incelemek için kullanılabilir.

Şekil 8: Veri Setinin Yüklenmesi ve Dosyaların listlenmesi

```
# Veri setinin dizin yolunu belirt
dataset_path = "/Users/anilturkmen/Desktop/Bitirme_tezi/Garbage_classification2"

# Veri setindeki tüm dosyaları listele
files = os.listdir(dataset_path)

# Tüm dosya isimlerini yazdır
print("Tüm Dosyalar:")
for file in files:
    print(file)
```

Tüm Dosyalar:
paper
.DS_Store
metal
cardboard
trash
glass
plastic

3.4.2. Veri Setindeki Görsel Dosyaların Türlerini ve Boyutlarını Analizi:

- Veri Setindeki Klasörleri Listeleme: `os.listdir(dataset_path)` işlevi kullanılarak veri setindeki tüm klasörlerin listesi elde edilir. Bu klasörler, farklı sınıfları veya kategorileri temsil edebilir.
- Görsel Türlerini ve Boyutlarını Analiz Etme: Her klasör içinde, o klasörde bulunan dosyalar listelenir. Ardından, her dosyanın dosya uzantısı kontrol edilir ve eğer dosya bir görsel dosyasıysa (.jpg, .jpeg, .png, .bmp, .gif gibi), dosyanın türü ve boyutları belirlenir.
- Sonuçları Kaydetme ve Yazdırma: Elde edilen sonuçlar, görsel türlerinin bir listesi ve her görselin dosya adıyla birlikte boyutlarının bir sözlüğü olarak kaydedilir. Son olarak, bu sonuçlar ekrana yazdırılır.

3.4.3. Veri Setindeki Sınıfa Ait Görsel Sayısını İnceleme

- Görsel Sayısını Hesaplama: Her klasördeki dosyalar listelenir ve her dosya bir görsel dosyasıysa sayacı artırır. Bu sayede her sınıfa ait görsel dosyaların toplam sayısı belirlenir.
- Sonuçları Kaydetme ve Yazdırma: Elde edilen sonuçlar, her sınıfın adıyla birlikte görsel sayısını içeren bir sözlük olarak kaydedilir. Son olarak, bu sonuçlar ekrana yazdırılır.

Veri setinin sınıfları arasındaki görsel dağılımını analiz etmek için. Özellikle, eğitim verisi dengesizse veya her sınıf için yeterli görsel veriye sahip olup olmadığını kontrol etmek için kullanışlıdır. Bu tür bir analiz, modelin eğitimi ve performansı üzerinde önemli etkilere sahip olabilir.

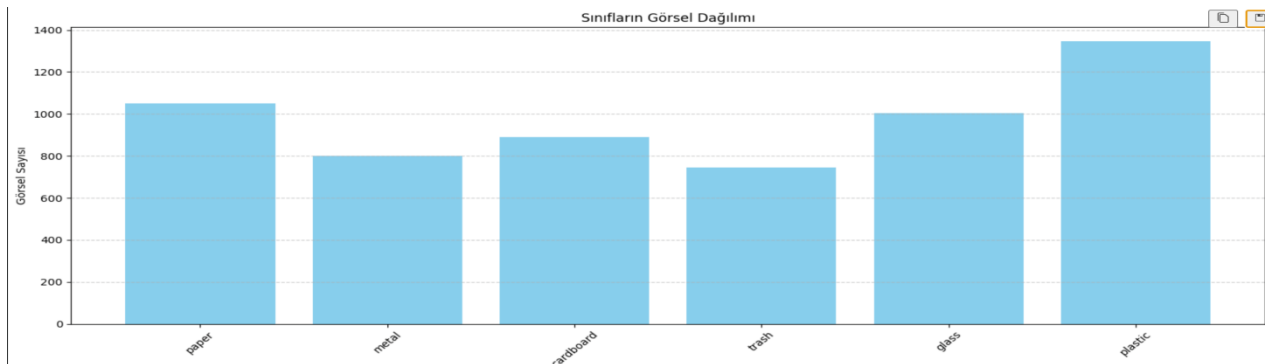
Şekil 9: Veri setindeki görsellerin sayılarının incelenmesi

```
Sınıf Adı: Görsel Sayısı  
paper: 1051  
metal: 801  
cardboard: 892  
trash: 745  
glass: 1004  
plastic: 1347
```

3.4.4. Veri Setinin Sınıflarının Görsel Dağılımını Görselleştirme

- Görsel Sayısını ve Örnek Görselleri Hesaplama: Her klasördeki dosyalar listelenir ve her dosya bir görsel dosyasıysa, görsel sayısı hesaplanır ve bu sınıfa ait örnek görseller alınır. Örnek görseller, her sınıftan sadece ilk 5 örneği içerecek şekilde sınırlıdır.
- Görsel Dağılımını Görselleştirme: matplotlib kütüphanesi kullanılarak, sınıfların görsel sayıları çubuk grafik ile görselleştirilir. Bu, her sınıfın veri setindeki görsel dağılımını gösterir.
- Örnek Görselleri Görselleştirme: Her bir sınıfa ait örnek görseller, bir alt grafik olarak gösterilir. Her sınıftan sadece ilk 5 örnek gösterilir ve bu görsellerin yanında sınıf adı belirtilir.

Şekil 10: Görsellerin sayılarının grafiksel dağılımları



3.4.5. CNN Model Oluřturma

Önceden eğitilmiş bir VGG16 modelini transfer öğrenme amacıyla kullanarak bir evriřimli sinir ağı (CNN) oluřturulur.

- VGG16 Modelinin Yüklmesi: tensorflow.keras.applications.VGG16 iřlevi kullanarak, önceden ImageNet veri kümesinde eğitilmiş olan VGG16 modeli yüklenir. Bu model, karmařık görsel özellikleri çıkarmak için evriřimli sinir ağı katmanlarından oluřur.
- Model Mimarisi ve Katmanlarının Belirlenmesi: Bir Sequential model oluřturulur. VGG16 modeli, modelin temel evriřimli sinir ağı katmanı olarak eklenir. Daha sonra, katmanları vektör formuna dönüřtürmek için düzleřtirme (Flatten) katmanı eklenir.
- Yeni Katmanların Eklenmesi: Daha sonra, modelin üzerine tam baėlantılı (fully connected) bir katman ve bir çıkıř katmanı eklenir. Tam baėlantılı katman, özellik vektörlerini daha yüksek seviyeli özniteliklere dönüřtürmek için kullanılır. Çıkıř katmanı ise, veri setindeki sınıfların sayısı kadar nörona sahip softmax aktivasyonlu bir katmandır.
- Transfer Öğrenme ve Katmanların Dondurulması: VGG16 modeli içindeki bazı katmanlar dondurulur ve eğitilmez. Bu, modelin daha alt seviyeli özelliklerini koruyarak, yalnızca daha yüksek seviyeli özellikleri öğrenmesini saėlar. Bu sayede, modelin eğitim süreci hızlanır ve daha iyi sonuçlar elde edilir.

Şekil 11: VGG16 modeli ve katmanlarının görünümü

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Total params: 14,714,688 (56.13 MB)

3.4.6. CNN Modelini Eğitme ve Test Etme

- Veri Yollarının Belirlenmesi: EGITIM, GECERLEME_YOLU ve TEST gibi değişkenler, eğitim, doğrulama ve test veri setlerinin bulunduğu dizin yollarını belirtir.

Şekil 12: Eğitim, Test ve Geçerleme yollarının belirlenmesi

```
# For validating the training process, augmented images are not utilized.
validation_datagen = tensorflow.keras.preprocessing.image.ImageDataGenerator(
    rescale=1. / 255
)

validation_generator = validation_datagen.flow_from_directory(
    GEÇERLEME_YOLU,
    target_size=(224, 224),
    batch_size=16,
)

2]
Found 4500 images belonging to 6 classes.
Found 300 images belonging to 6 classes.
```

- Veri Artırma: tensorflow.keras.preprocessing.image.ImageDataGenerator kullanılarak, eğitim veri seti için veri artırma teknikleri belirlenir. Bu teknikler, veri setinin çeşitliliğini artırmak ve aşırı uyum (overfitting) riskini azaltmak için kullanılır. Örneğin, döndürme, kaydırma, büyütme gibi işlemler gerçekleştirilir.
- Veri Akışlarının Oluşturulması: flow_from_directory işlevi kullanılarak, belirtilen dizinlerden veri akışları oluşturulur. Bu akışlar, belirtilen boyuta yeniden boyutlandırılır ve belirtilen batch boyutuna bölünür.
- Modelin Eğitimi: Oluşturulan veri akışları, modelin fit işlevine verilerek eğitilir. Bu işlemde, belirtilen adımlar (steps) ve epoch sayısı kullanılır. Eğitim sırasında, modelin doğrulama seti üzerindeki performansı da takip edilir.

Şekil 13: Eğitilen Modelin Son epoch değerleri

```
100/100 ————— 130s 1s/step - acc: 0.9095 - loss: 0.0918 - val_acc: 0.9167 - val_loss: 0.0706
Epoch 49/50
100/100 ————— 159s 2s/step - acc: 0.9102 - loss: 0.0774 - val_acc: 0.9625 - val_loss: 0.0514
Epoch 50/50
100/100 ————— 155s 2s/step - acc: 0.9176 - loss: 0.0815 - val_acc: 0.9750 - val_loss: 0.0392
```

- Modelin Kaydedilmesi: Eğitim tamamlandıktan sonra, model belirtilen dosya adıyla kaydedilir.
- Modelin Test Edilmesi: Modelin performansını değerlendirmek için test veri seti kullanılır. Test verileri üzerinde modelin kayıp (loss) ve doğruluk (accuracy) metrikleri hesaplanır ve ekrana yazdırılır.

Şekil 14: Modelin Test değerlendirmesi sonucu

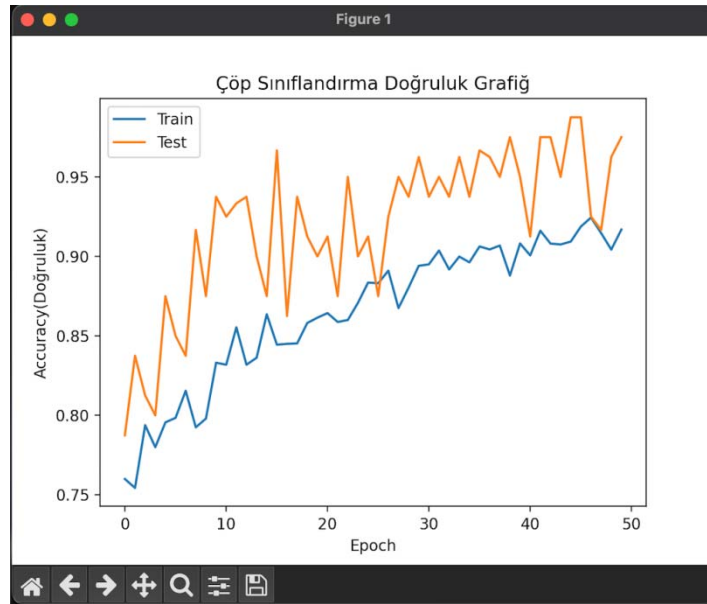
```
/opt/anaconda3/envs/my_env/lib/python3.9/site-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:14: self._warn_if_super_not_called()
25/25 — 38s 2s/step - acc: 0.9124 - loss: 0.0898
test acc: 0.9160000085830688

In [38]:
```

3.4.7. Eğitilen Modelin Test Grafiklerinin İncelenmesi

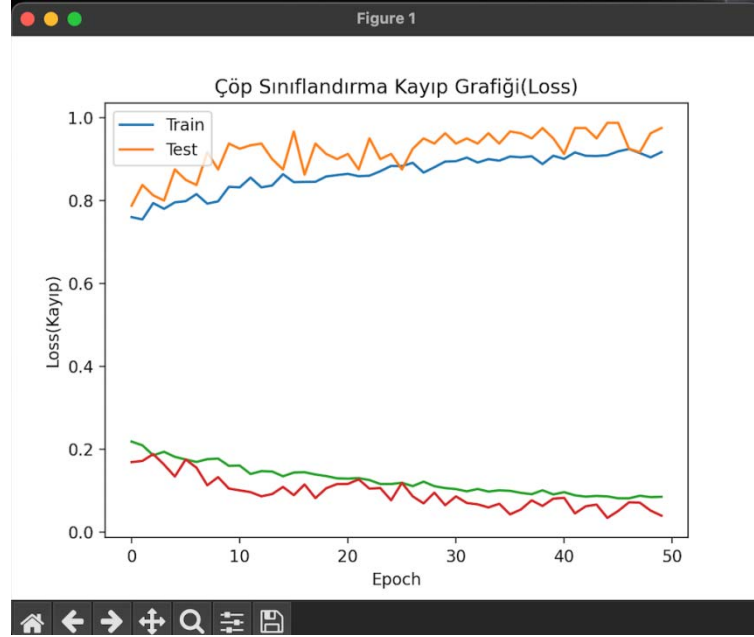
- Doğruluk Grafiği Oluşturma: İlk olarak, eğitim ve doğrulama (validation) veri setleri üzerinde elde edilen doğruluk değerleri, eğitim sürecinin her epoch'unda kaydedilir. Bu kod parçası, eğitim (train) ve doğrulama (validation) doğruluk değerlerini bir grafikte görselleştirir. Grafikte, eğitim ve doğrulama doğruluk değerleri epoch sayısına göre çizilir.

Şekil 15: Modelin Doğruluk grafiği



- Kayıp Grafiği Oluşturma: Ardından, eğitim ve doğrulama veri setleri üzerindeki kayıp değerleri de her epoch'ta kaydedilir. Eğitim ve doğrulama kayıp değerlerini bir grafikte görselleştirir. Grafikte, eğitim ve doğrulama kayıp değerleri epoch sayısına göre çizilir.

Şekil 16: Modelin Loss grafiği



Bu grafikler, modelin eğitim sürecinin ilerlemesini izlemek ve aşırı uyum (overfitting) gibi problemleri belirlemek için kullanılır. İdeal bir durumda, eğitim ve doğrulama doğruluk değerleri artarken, eğitim ve doğrulama kayıp değerleri azalır. Eğer eğitim doğruluk değeri yükselirken doğrulama doğruluk değeri düşerse veya eğitim kayıp değeri azalırken doğrulama kayıp değeri artarsa, model aşırı uyum gösterebilir.

DÖRDÜNCÜ BÖLÜM

4. KULLANICI ARAYÜZÜ OLUŞTURMA

Şekil 17: GUI Açılış ekran görüntüsü



4.1. Kullanılan Kütüphaneler

- TensorFlow ve Keras: Derin öğrenme modelinin eğitimi ve yüklenmesi için kullanıldı.
- NumPy: Sayısal işlemler ve veri manipülasyonu için kullanıldı.
- Tkinter: GUI oluşturmak için kullanılan Python kütüphanesi.
- PIL (Python Imaging Library): Görüntü işleme ve gösterme işlemleri için kullanıldı.

4.2. Arayüzün Kodlanması

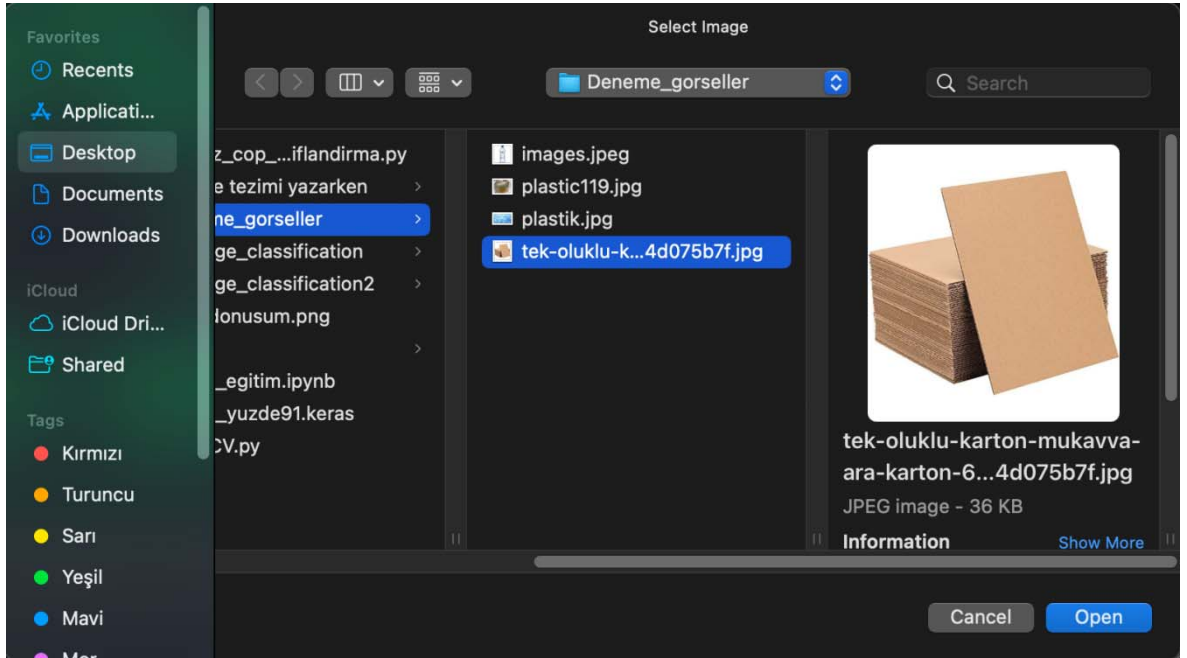
4.2.1. Görsel İşleme ve Sınıflandırma Fonksiyonu

- Görüntünün Yüklenmesi ve İşlenmesi: Kullanıcı tarafından seçilen görüntü, belirli bir boyuta (224x224 piksel) yeniden boyutlandırılır ve bir numpy dizisine dönüştürülür. Görüntü verileri normalleştirilir (0-255 aralığındaki piksel değerleri 0-1 aralığına çekilir).
- Modelin Yüklenmesi: Daha önceden eğitilmiş model yüklenir.
- Tahmin: Görüntü, model tarafından sınıflandırılır ve en yüksek olasılıkla hangi sınıfa ait olduğu belirlenir.

4.2.2. Kullanıcı Arayüzü (GUI)

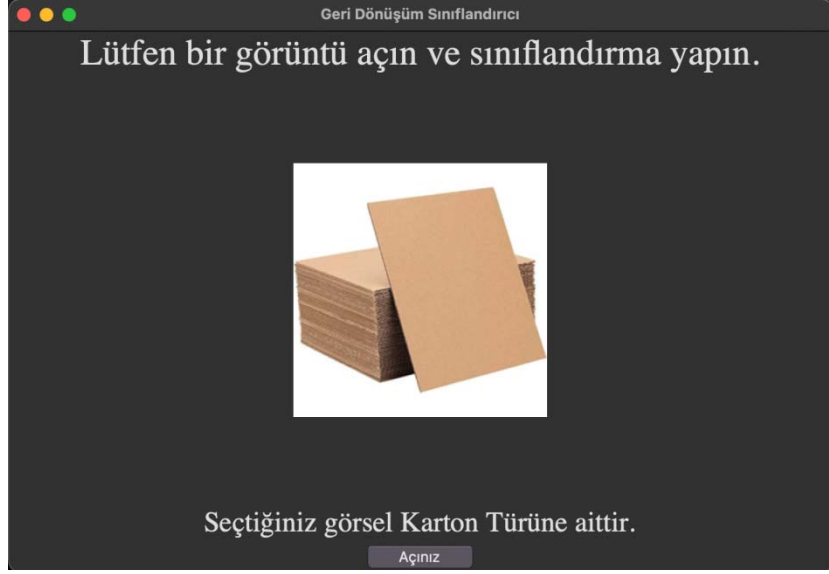
- Görüntü Seçimi: Kullanıcı, dosya diyalogu kullanarak bir görüntü seçer.

Şekil 18: Görüntü Seçim Ekranı



- Görselin Gösterilmesi: Seçilen görüntü, GUI üzerinde gösterilir.
- Sınıflandırma Sonucunun Gösterilmesi: Modelin tahmin ettiği sınıf, kullanıcıya metin olarak gösterilir.

Şekil 19: Seçilen Görüntünün Türünün Çıktı Sonucu



4.2.3. Ana Uygulama Yapısı

- Ana Pencere: Tkinter kullanılarak ana pencere oluşturulur. Bu pencere, gerekli bileşenler (talimat metni, görüntü gösterme alanı, sonuç metni, dosya açma butonu) ile donatılır.
- Etkinlik Döngüsü: Tkinter'ın mainloop fonksiyonu, GUI'nin sürekli olarak açık kalmasını ve kullanıcı etkileşimlerine yanıt vermesini sağlar.

SONUÇ

Bu çalışma, atık yönetiminde otomatik sınıflandırma sistemlerinin potansiyelini göstermiş ve geri dönüşüm süreçlerine katkı sağlayabilecek bir model geliştirmiştir. Geliştirilen model, geniş bir veri seti kullanılarak eğitilmiş ve test verileri üzerinde yüksek doğruluk oranı göstererek farklı çöp türlerini doğru bir şekilde sınıflandırabilmiştir. Bu başarı, hem veri setinin kalitesine hem de modelin eğitimi sırasında uygulanan optimizasyon tekniklerine bağlıdır.

Ayrıca, çalışma kapsamında geliştirilen kullanıcı arayüzü, sistemin pratik kullanımını kolaylaştırmış ve kullanıcıların çöpleri etkin bir şekilde sınıflandırmasına olanak tanımıştır. Kullanıcıların, çöp görsellerini yükleyip sınıflandırma sonuçlarını kolayca görebilmeleri, sistemin gerçek dünya uygulamalarında kullanımını artırabilir ve atık yönetimini daha etkin hale getirebilir.

Gelecekte, daha geniş veri setleri ve gelişmiş model mimarileri kullanılarak sistemin doğruluğu ve performansı artırılabilir. Ayrıca, gerçek dünya uygulamalarına entegrasyonu ile çevresel etkilerin azaltılması ve geri dönüşüm süreçlerinin iyileştirilmesine önemli katkılar sağlanabilir. Bu çalışma, atık yönetimi alanında otomatik sınıflandırma sistemlerinin önemli bir adımı olmuş ve gelecekteki araştırmalara ışık tutabilecek bir temel oluşturmuştur.

YARARLANILAN KAYNAKLAR

- Yıldız, E. N., Bingöl, H., & Yıldırım, M. (2023). Önerilen Derin Öğrenme ve Makine Öğrenmesi Tabanlı Hibrit Model ile Çevresel Atıkların Sınıflandırılması. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 35(1), 353-361.
- Avşar, M., & Karakaş Tandoğan, G. (2023). Çevre Dostu Konaklama İşletmelerine Fazla Ücret Ödeme Eğiliminin Kuşaklara Göre Değişikliğinin İncelenmesi. İnsan Ve Toplum Bilimleri Araştırmaları Dergisi, 12(2), 976-994.
- Yıldız, E. N., Bingöl, H., & Yıldırım, M. (2023). Önerilen Derin Öğrenme ve Makine Öğrenmesi Tabanlı Hibrit Model ile Çevresel Atıkların Sınıflandırılması. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 35(1), 353-361.
- Karaca, Y. E., Aslan, S., & Hark, C. (2022). EVSEL ATIKLARIN DERİN ÖĞRENME TEKNİKLERİ İLE AYRIŞTIRILMASI. Computer Science, Vol: 7(Issue: 1), 10-19.
- Atik, S. O. (2022). Derin Öğrenme Yaklaşımlarıyla Çevresel İzlemeye Yönelik Çok Sınıflı Sahne Sınıflandırma. Avrupa Bilim ve Teknoloji Dergisi, (41), 307-314.
- Arçev Çevre. (2023). Atık Yönetimi Nedir? Retrieved from <https://www.arcev.com.tr/atik-yonetimi-nedir/>
- Cadem Digital. (2023). Sürdürülebilir Gelecek İçin Elektronik Atık Yönetiminin Önemi. Retrieved from <https://cademdigital.com.tr/surdurulebilir-gelecek-icin-elektronik-atik-yonetiminin-onemi/>
- Model Çevre. (2022). Atık Yönetimi Hiyerarşisi. <https://www.modelcevre.com/atik-yonetimi-hiyerarşisi.html>
- URL, “Çevre Mühendisliği Eğitimi”, https://pdfs.semanticscholar.org/73a9/03088b9eb365ac30b5540213b1b75f503a91.pdf?_gl=1*pgosum*_ga*Njg0Njc3NjYuMTcxNTk1NDM4OQ.*_ga_H7P4ZT52H5*MTcxNjc3ODc3MS41LjEuMTcxNjc3OTExOC41Mi4wLjA. (30.01.2018)

- URL, “Çevre Dostu Konaklama İşletmelerine Fazla Ücret Ödeme Eğiliminin Kuşaklara Göre Değişikliğinin İncelenmesi”, <https://doi.org/10.15869/itobiad.1244982> (30.06.2023)
- URL, “Fırat Üniversitesi Mühendislik Bilimleri”, <https://doi.org/10.35234/fumbd.1230982> (28.03.2023)
- URL, “Evsel Atıkların Derin Öğrenme Teknikleriyle Ayrıştırılması”, <https://doi.org/10.53070/bbd.1071536> (06.06.2022)
- URL, “Avrupa Bilim ve Teknoloji”, <https://dergipark.org.tr/en/download/article-file/2191275>(20.11.2022)