

C- Bugüne kadar yapılanlar

1. EPA verileri tek tek Türkçeye çevrildi ve atıkların nereye atılması gerektiğindeki bilgiler kesinleştirildi. Atıklar hakkında kısa istatistik bilgileri de buna dahil.
 - a. <https://atik-tespit-sistemi.netlify.app/plugins/atik-verileri.pdf>
2. Web sitesinin Template'i hazırlandı
 - a. <https://atik-tespit-sistemi.netlify.app/index.html>
3. Javascript ile Web üzerinde kamera açma ve fotoğraf alma işlemleri gerçekleştirildi.
4. En yakın geri dönüşüm kutusunu bildirecek olan harita sisteminin çok basit tasarımı hazırlandı.
5. MsSql `de veritabanı oluşturuldu.
6. ImageNet araştırması yapıldı ve ImageNet üzerinden fotoğrafların alınması için gerekenler yapıldı
7. Python OpenCv öğreniminde aldığımız eğitim ile basit bir proje yaptık ve eğitime devam etmekteyiz.
8. ImageNet araştırması ile ilgili bilgi:

ImageNet projesi, görsel nesne tanıma yazılım araştırmalarında kullanılmak üzere tasarlanmış büyük bir görsel veri tabanıdır. ImageNet, "balon" veya "çilek" gibi tipik bir kategoride, birkaç yüz resimden oluşan 20.000'den fazla kategori içerir. Gerçek görüntüler ImageNet'e ait olmasa da üçüncü taraf resim URL'lerinin ek açıklamalarının veritabanı doğrudan ImageNet'ten ücretsiz olarak edinilebilir.

Veri Kümesi

ImageNet, açıklama sürecini kitle kaynaklı olarak kullanır. Görüntü düzeyinde ek açıklamalar, bir görüntüde "bu görüntüde kaplan var" veya "bu görüntüde kaplan yok" gibi bir nesne sınıfının varlığını veya yokluğunu belirtir. Nesne düzeyinde ek açıklamalar, belirtilen nesnenin (görünür kısmı) çevresinde bir sınırlayıcı kutu sağlar. ImageNet, nesneleri kategorilere ayırmak için geniş WordNet şemasının bir türevini kullanır

2012'de ImageNet, Mechanical Turk'ün dünyanın en büyük akademik kullanıcısıydı. Ortalama bir çalışan dakikada 50 görüntü belirledi.

Derin Öğrenmenin Önemi

30 Eylül 2012'de, AlexNet_ adlı bir evrimsel sinir ağı (CNN), ImageNet 2012 Challenge'da ilk 5'te %15,3'lük bir hata elde etti ve bu, ikinciden yüzde 10,8 puan daha düşüktü. Bu, derin öğrenme devriminin temel bir bileşeni olan eğitim sırasında grafik işleme birimlerinin (GPU'lar) kullanılması nedeniyle mümkün olmuştur. *The Economist*'e göre, "Birden insanlar, yalnızca yapay zekâ topluluğu içinde değil, bir bütün olarak teknoloji endüstrisi genelinde dikkat etmeye başladı.

2015'te AlexNet, ImageNet 2015 yarışmasını kazanan 100'den fazla katmanla Microsoft'un çok derin CNN'sinden daha iyi performans gösterdi

Veritabanının Geçmişi

AI araştırmacısı Fei-Fei Li , 2006 yılında ImageNet fikri üzerinde çalışmaya başladı. Çoğu AI araştırmasının modellere ve algoritmalara odaklandığı bir zamanda, Li, AI algoritmalarını eğitmek için mevcut verileri genişletmek ve iyileştirmek istedi. 2007 yılında, Li Princeton profesörü ile bir araya Christiane Fellbaum , yaratıcıları biri WordNet projesini görüşmek üzere.buluştular. Bu toplantının sonucunda Li, WordNet'in kelime veri tabanından başlayarak ve birçok özelliğini kullanarak ImageNet'i oluşturmaya devam etti.

Princeton'da yardımcı doçent olarak Li, ImageNet projesinde çalışmak üzere bir araştırma ekibi kurdu. Görüntülerin sınıflandırılmasına yardımcı olması için Amazon Mechanical Turk'ü kullandılar

Veritabanılarını ilk kez Florida'daki 2009 Bilgisayarlı Görme ve Örüntü Tanıma Konferansı'nda (CVPR) poster olarak sundular.

9. Atık Listesi ile ilgili araştırma ve verilerin tespit edilip Türkçeye çevrilmesi:

**** EPA -- U.S. Environmental Protection Agency -- ABD Çevre Koruma Ajansı** // 33 atık örneği**

1. İçecek kartonu -- Kâğıt İçecek kartonları sadece kartondan yapılmaz- plastik ve alüminyum katmanlar vardır, bu nedenle aynı şekilde geri dönüştürülemez. İçecek kartonlarının geri dönüştürülebilmesi için özel ekipman ve işleme tabi tutulması gerekir. EPA, aseptik ve katlanır karton, kâğıt torba, ambalaj kâğıdı ve diğer kâğıt ve karton ambalaj üretiminin 2017 yılında toplam 8,5 milyon tona ulaştığını ve bunun 2/3'ünden fazlasının çöpe atıldığını tahmin ediyor. Yeniden Kullanın Süt veya meyve suyu kartonları ile el yapımı oyuncaklardan kuş kutularına kadar yeniden kullanmanın birçok yolu vardır. Şimdi fikirleri keşfetmeye başlayın! <https://www.reddedart.com/milk-carton-crafts-juice-tetra-pack-crafts/> "Geri dönüşüm" kutusu (muhtemel) "Geri Dönüşüm" kutusuna atmadan önce hava boşlukları olan kutuları ve diğer kâğıt öğeleri düzleştirin. Geri dönüştürülebilir ürünlerinizin artık yiyecek kalıntısı içermediğinden emin olun. Gerekirse yiyecek ve içecek kaplarını hafifçe yıkayın. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.

2. Diğer karton ürünler- Kâğıt "Geri Dönüşüm" kutusu Nakliye, paketleme ve diğer amaçlar için kullanılan kâğıt, kartonlar, diğer malzemelerle karıştırılmadığı takdirde geri dönüştürülebilir. Geri Dönüşüm kutunuza koymadan önce hava boşlukları olan kutuları ve diğer kâğıt öğeleri düzleştirin. Islak veya kirli kâğıt geri dönüşüm için uygun değildir, ancak bu seçenek size sunuluyorsa gübrelenebilir. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.

3. Waxed karton -- Kâğıt (smooth-düz) kaplama olduğu için. İçecek kartonları, kâğıt dondurma kapları ve yapışkan karton kutular gibi yiyecek kapları tipik olarak plastik gibi kâğıttan başka ince bir malzeme tabakasıyla kaplanır. Bu kaplama geri dönüşümü daha zor hale getirir. Kâğıt geri dönüştürüldüğünde, "bulamaç" veya "kâğıt hamuru" oluşturmak için suyla karıştırılır ve bu ekstra malzemeler su ile çözülemez. "Geri dönüşüm" kutusu (muhtemel) Geri dönüşüm programınız ıslak veya donmuş gıda karton kutularını kabul ediyorsa, büyük olasılıkla mumlu karton kutuları da kabul edecektir. Gıda kontaminasyonu nedeniyle birçok geri dönüştürülebilir atık çöplüklere gidiyor. Geri dönüştürülebilir ürünlerinizin artık yiyecek kalıntısı içermediğinden emin olun. Gerekirse yiyecek ve içecek kaplarını hafifçe yıkayın. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.

4. Karton nakliye (kargo) kutusu -- Kâğıt Yeniden Kullanın Karton nakliye kutuları yeniden kullanılabilir, düzleştirilebilir ve daha sonra kullanılmak üzere saklanabilir veya herhangi bir yere bağışlanabilir. "Geri Dönüşüm" kutusu Nakliye, paketleme ve diğer amaçlarla kullanılan kâğıt, kartonlar, diğer malzemelerle karıştırılmadığı takdirde geri dönüştürülebilir. Geri dönüşüm kutunuza atmadan önce hava boşlukları olan kutuları ve diğer kâğıt öğeleri düzleştirin. Islak veya kirli kâğıt geri dönüşüm için kabul edilemez, ancak bu seçenek size sunuluyorsa gübrelenebilir. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.

5. Biyolojik olarak parçalanabilen kap, çanta, içecek şişesi, bulaşık takımı, gıda kutusu, kapak, çatal bıçak takımı -- Diğer Atıklar Nesne açıkça "Biyobozunur" veya "---" olarak etiketlenmiştir. Bazı öğeler de "Ben plastik değilim" olarak etiketlenmiştir. Biyobozunur plastikler, canlı organizmaların, genellikle mikropların etkisiyle suya, karbondioksit ve biyokütleyle ayrılan plastiklerdir. Bir çöplükte bozulmak üzere tasarlanmıştır ve bu nedenle siyah geri dönüşüm kutunuza atılmalıdır. Azaltmak Biyolojik olarak parçalanabilen plastik, normal plastikten çevre için daha iyi olmasına rağmen, tek kullanımlık plastik parçaların kullanımını azaltmak her zaman en iyi çözümdür. "Geri Dönüşüm" kutusu Normal plastikten daha iyi olsa da biyolojik olarak parçalanabilen plastik, ayrışması için en iyi koşulların olduğu bir çöp sahasına gider. Çöp sahasında çürümesi sadece 3 ila 6 ay sürebilir. Yeniden kullanılamayan, geri dönüştürülemeyen veya gübrelenemeyen ve tehlikeli veya toksik maddeler içermeyen malzemeleri siyah geri dönüşüm kutunuza yerleştirin.

6. Gübrelenebilir bardak, kapak, bulaşık takımı, gıda kutusu, çatal bıçak takımı -- Diğer Atıklar Nesne açıkça gübrelenebilir olarak etiketlenmelidir. Bazı öğeler ayrıca "Ben plastik değilim" olarak etiketlenmiştir. Gübrelenebilir plastikler, gübreleme yoluyla biyolojik olarak parçalanabilen yeni nesil plastiklerdir. Bazı gübrelenebilir plastiklerin gübrelenmesi, yüksek sıcaklıklar, basınç ve besin konsantrasyonunun yanı sıra belirli kimyasal oranlar dahil olmak üzere çevresel faktörlerin sıkı kontrolünü gerektirir. Bu koşullar yalnızca endüstriyel gübreleme tesislerinde yeniden oluşturulabilir, bu nedenle gübrelenebilir plastikler, el yapımı gübreleme yöntemleri ile işlenmez. Bölgenizde yeşil "Gübreleme" kutusu yoksa, gübrelenebilir plastikler geri dönüştürülemediğinden tek seçenek siyah "Geri dönüşüm" kutusudur. Azaltmak Gübrelenebilir plastiğin çevre için normal plastiğe göre daha iyi olmasına rağmen, tek kullanımlık plastik maddelerin kullanımını azaltmak her zaman en iyi çözümdür. İşe, hatta kahve dükkanına giderken kendi fincanınızı veya kupanızı yanınıza alın. "Bazı yerler kendi kupanızı kullanmak için indirim sunar." Neden Gübreleme? Gübreleme, çöp sahasına gönderdiğiniz atık miktarını azaltır, bahçelerinizdeki toprağı iyileştirir, bitkilerin sağlıklı büyümesini sağlar, toprağın su tutmasına yardımcı olur ve yabancı otları öldürür, çöp sahalarına gönderilen atıkların %30'u gübrelenebilir.

7. Köpük bardak, bulaşık takımı, gıda kutusu, küçük "fıstık" köpükler -- Diğer Atıklar Köpükler, bir plastik türüdür, bu nedenle çöp sahasında ayrışması 500 yıl kadar sürebilir. Azaltmak İşe, hatta kafeye giderken kendi fincanınızı veya kupanızı yanınıza alın. "Bazı yerler kendi kupanızı kullanmak için indirim sunar." Ek satın alımlardan kaçınmalıyız. Yeniden Kullanın Köpük 'den yapılmış öğeleri yeniden kullanmak için birçok seçenek vardır, 3D Printer için kapsamlı kullanım kılavuzları bulabilirsiniz. "Geri Dönüşüm" kutusu

(genellikle) Çoğu durumda Kopuk, geri dönüştürülemez. Bertaraf için, depolama sahasına giden çöp kutusuna atılmalıdır. Büyük parçaları kırdığınızdan emin olun, böylece çöp kutunuza sığar. Yeniden kullanılamayan, geri dönüştürilemeyen veya "gübrelenemeyen" ve tehlikeli veya toksik maddeler içermeyen malzemeleri siyah "Geri dönüşüm" kutunuza yerleştirin. Küçük parçaları bir torbaya koyun, böylece hayvanlar yiyecek olarak karıştırıp tüketebilecekleri parçalara ayıramazlar.

8. Zarf -- Kâğıt İş yazışmalarından doğum günü kartlarına ve önemsiz postalara kadar, kâğıt zarflar sürekli olarak kullanılır ve her yıl çöplüklere ekleniyor. EPA, posta ve ofis kâğıdı üretiminin 2017'de yaklaşık 10 milyon tona ulaştığını tahmin ediyor. Azaltmak Faturaları çevrimiçi olarak veya e-posta ile postalamak dışında herhangi bir kağıtsız faturalamayı seçin. Önemsiz posta almayı durdurmak için adımlar atın. Kataloglardan, siyasi postalara, kredi kartı ve sigorta tekliflerine kadar, size gönderilen istenmeyen postaları iletişim kurarak durdurun. "Geri Dönüşüm" kutusu Islak kâğıt lifleri toplanamayacağından zarfın kuru olduğundan emin olun. Şeffaf plastik adres etiketini kaldırmamız gerekmez, ancak önce onu kaldırmanın da bir zararı olmaz. Ardından geri dönüşüm kutusuna atmanız yeterlidir. Kâğıt öğelerinizin temiz, sağlam ve kuru olduğundan emin olun. Islak veya kirli kâğıt geri dönüşüm için kabul edilemez, ancak bu seçeneğin size sunulması halinde gübrelenebilir. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır

9. Cam İçecek Şişesi, Cam Şişe, Cam Gıda Şişesi -- Cam EPA, bira ve meşrubat şişeleri, şarap ve likör şişeleri gibi cam kapların yanı sıra kavanoz gıda, meyve suları, kozmetik ve diğer ürünlerin üretiminin 2017'de yaklaşık 9 milyon ton olduğu ancak bunun sadece 1/3'ünü geri dönüştürüldüğünü tahmin ediyor. Azaltmak Tek kullanımlık şişeler veya teneke kutular satın almak yerine kendi doldurulabilir şişenizi getirin. "Geri Dönüşüm" Kutusu Tek kullanımlık cam kapların çoğu genellikle geri dönüştürülebilir. Birçok geri dönüştürülebilir malzeme, çöplüklere gidiyor. Geri dönüştürülebilir ürünlerinizin artık yiyecek kalıntısı içermediğinden emin olun. Gerekirse yiyecek ve içecek kaplarını hafifçe yıkayın. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır

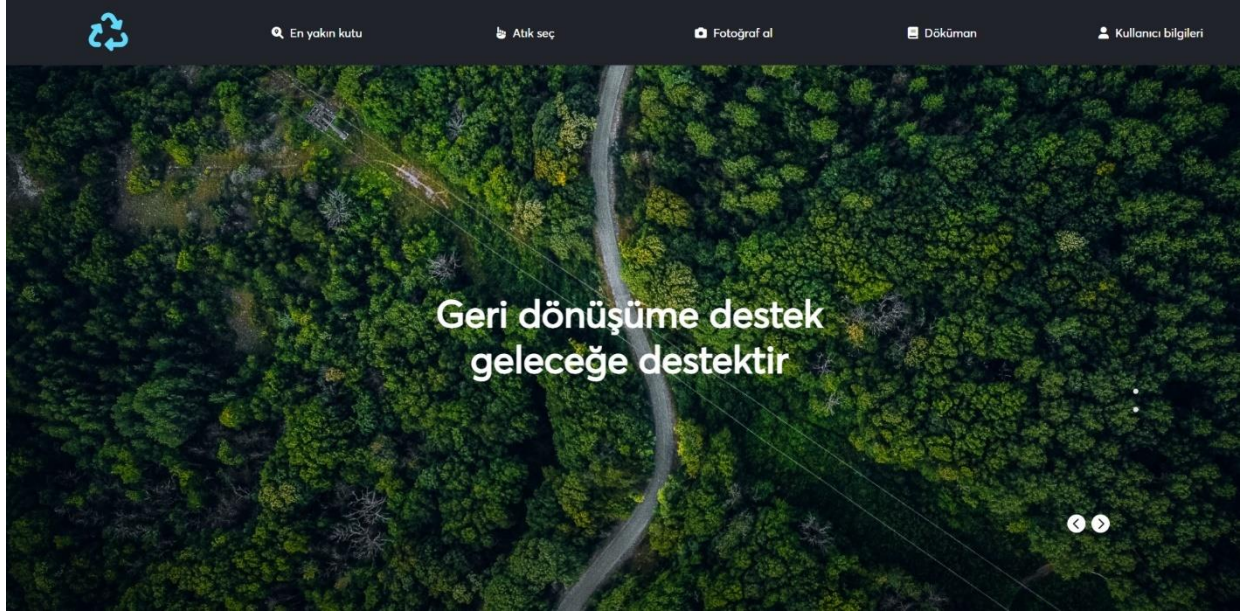
10. Sigara İzmariti -- Diğer Atıklar "Geri Dönüşüm" Kutusu Yeniden kullanılamayan, geri dönüştürilemeyen veya gübrelenemeyen tehlikeli veya toksik maddeler içermeyen malzemeleri siyah çöp kutunuza koyun.

11. Tek Kullanımlık Pil, Şarj Edilebilir Pil -- Piller için özel kutu Kullanılmış piller belediye çöplüklerinde bulunan tüm atıkların %1'inden daha azını oluştururken, bu pillerin %1'i çöplükte bulunan tüm zehirli ağır metallerin %88'inden sorumludur. Ek olarak, düğme pillerde az miktarda cıva bulunabilir, bu da bu küçük, yuvarlak cihazları tehlikeli atık haline getirir ve uygun yerel kaynaklar aracılığıyla geri dönüştürülmelidir. Bu tür piller tek kullanımlık olma eğilimindedir ve saatler, işitme cihazları, kameralar ve hatta oyuncaklar gibi öğelerde bulunur. Elektronik cihazlar, piller, floresan ampuller, böcek ilaçları ve diğer tehlikeli evsel atık kimyasalları gibi maddeleri geri dönüşüm kutularınıza atmak yasa dışıdır. Bu maddeler geri dönüştürülemez, gübreye dönüştürülemez veya çöplüklerde kullanılmasına izin verilmez. Azaltmak Aynı pilden 1000'e yakın kullanım sağlayan şarj edilebilir piller ve pil şarj cihazı satın alın. Yeniden şarj edilebilir piller başlangıçta tek kullanımlık pillerden daha pahalıdır, ancak değiştirilmeleri gerekmeden birçok kez yeniden şarj edilebildiklerinden toplam sahip olma maliyeti ve çevresel etkisi çok daha düşüktür. "Geri Dönüşüm" Kutusu Piller için özel olarak ayrılmış kutulara bırakabilirsiniz.

12. Çakmak -- Diğer Atıklar "Geri Dönüşüm" Kutusu Yeniden kullanılamayan, geri dönüştürilemeyen veya gübrelenemeyen tehlikeli veya toksik maddeler içermeyen malzemeleri siyah çöp kutunuza yerleştirin.

13. Alüminyum İçecek Şişesi, Alüminyum İçecek Kutusu -- Metal EPA, 2017 yılında bira, meşrubat kutuları, genel amaçlı kutular, folyo ve kapaklar dahil olmak üzere alüminyum kaplar ve ambalajların üretimini yaklaşık 2 milyon ton olarak tahmin etti, ancak bu malzemenin yalnızca 1/2'si geri dönüştürüldü. Yeniden kullanın Bazı alüminyum içecek şişeleri yeniden kullanılmak üzere tasarlanmıştır. "Geri dönüşüm" kutusu Alüminyum içecek şişeleri geri dönüştürülebilir. Gıda kirliliği nedeniyle birçok geri dönüştürülebilir atık çöplüklere gidiyor. Geri dönüştürülebilir ürünlerinizin artık yiyecek kalıntısı içermediğinden emin olun. Gerekirse yiyecek ve içecek kaplarını hafifçe durulayın. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.,

10.Web Sitesi Görselleri

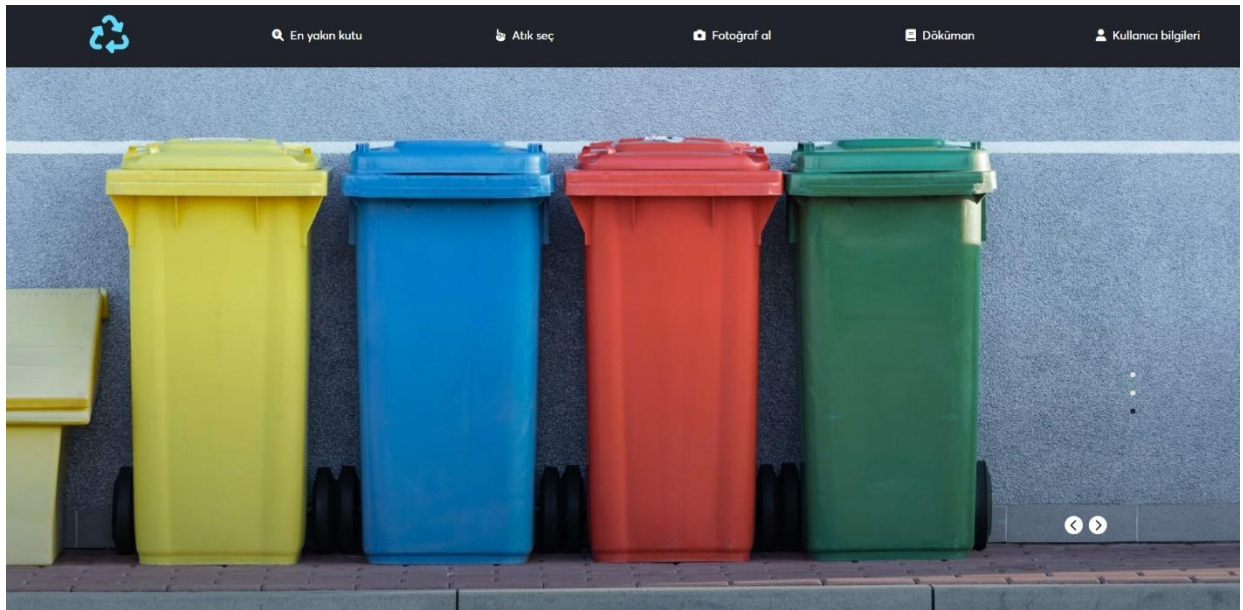


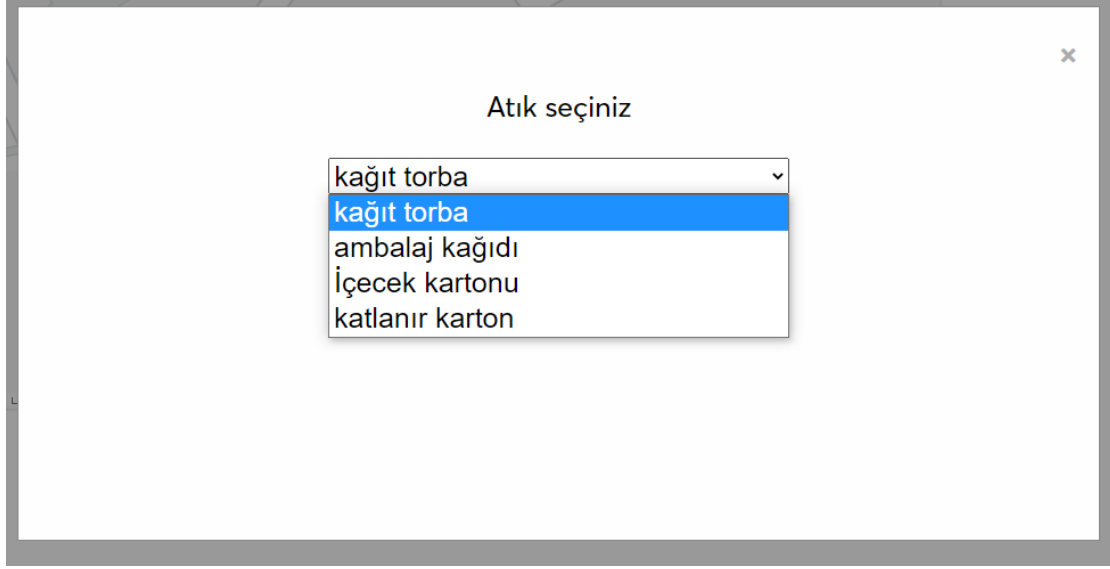
T. C. Muğla Sıtkı Koçman Üniversitesi-Atık Yönetimi Koordinatörlüğü

Geri dönüşebilir atıkların diğer atıklarla birlikte karışık olarak tek bir poşet içine konup atmanız. Önce geri dönüşebilir atıkların ayrı ayrı doğru atık kutusuna atarak ardından kalan atıkları siyah renkli "Diğer Atıklar" kutusuna atınız.

| CAM GLASS | METAL METAL | PLASTİK PLASTIC | KAĞIT PAPER | DiĞER ATIKLAR OTHER WASTE |
|---|---|---|---|---|
| DÖNERSE YİNE SENİNDİR! | DÖNERSE YİNE SENİNDİR! | DÖNERSE YİNE SENİNDİR! | DÖNERSE YİNE SENİNDİR! | DÖNERSE YİNE SENİNDİR! |
| <ul style="list-style-type: none">• CAM SU ŞİŞELERİ• CAM SODA ŞİŞELERİ• CAM MEŞRUBAT ŞİŞELERİ• CAM BARDAK, SÜRAHİ, KAVANÖZ GİBİ MALZEMELER | <ul style="list-style-type: none">• METAL MEŞRUBAT KUTULARI• KONSERVE TENEKELERİ• METAL GIDA KAPLARI• ALÜMİNYUM KAPLAR | <ul style="list-style-type: none">• PET ŞİŞELER, PİPET• PLASTİK AYRAN KUTUSU• PLASTİK MEŞRUBAT SİSESİ• PLASTİK KUTULAR VE KAPLAR• PLASTİK POSETLER• PLASTİK GIDA AMBALAJLARI• PLASTİK ÇATAL, KAŞIK VE BİÇAK | <ul style="list-style-type: none">• KAĞIT-KARTON MALZEMEDEN AMBALAJLAR• KARTON MEŞRUBAT KUTULARI• KARGO KUTULARI• KULLANILMIŞ KAĞITLAR | <ul style="list-style-type: none">• PARLAK - JELATİN PAKETLER (ÇİPS, GOFRET, ÜRÜNLERİN PARLAK DİŞ PAKETLERİ)• İSLAK MENÜLÜ - PEÇETE - TEK KULLANIMLIK• POSETİ - AĞSAP KANİSTİRMA• STRAFOZ KÖPÜK - SAKIZ• PORSELEN - SERAMİK - SUPRİNTİ• ORGANİK VE YİYECEK ATIKLARI• SİĞIRTA PAKETLERİ - İSLAK MENÜLÜ• AĞSAP ATIKLARI - ÇAY VE KAHVE POS• TEKSTİL ATIKLARI - PLASTİK EL DİVEN |

KUTUYA ATILMAYACAK ATIKLAR: •Yazıcı ve fotokopi kartuş ve tonerleri •Floresan ve tasarruflu lambalar •Elektronik atıklar (kablo, bilgisayar parçası, su ısıtıcı vb.) •Piller •Yağlar •Bazılar ve çözünenler •Tıbbi atıklar (maske, eldiven, enjeksiyon iğnesi vb.)





Önemli linkler

[Geri donusum kutularina hangi atiklar atilabilir ve hangileri atilamaz](#)

[Atilmamasi gerekenler](#)

[BSO \(bau & service oberursel\) Almanyanin atik yönetimi, şehir temizliği gibi hizmetler veren bir sirketdir ve bu sirket tarafından yayınlanmis, turkceye cevrilmis tam geri donusum bilgi kitapçigi](#)

[Tam detayli kodlar ile atiklar](#)

[Sirketlerin atik icin basvurmasi gereken site bilgisi](#)

Nedir

Atik tespit sistemi atiklarin mobil uygulama ile fotograf alinarak hangi geridonusum kutusuna atilmasi gerektiği hakkında bilgiler veren, atilacak nesnenin özelliklerini gösteren sistemdir. Bu sistemin tasarlanmasında kullanılacak olan teknolojiler aşağıdaki gibidir:

Mobil uygulama

- Mobil uygulama yapiminda **React Native** kullanılacaktır
- Mobil uygulama basit bir arayuz ile kullanicinin sonuca en hizli yol ile ulasmasini saglamayi hedeflemektedir. Kullanici ana ekrandaki buton ile hemen fotograf alabilecek ve bilgileri gorebilecek.
- Web destekli calisacak

Web uygulaması

- Web uygulamasinin yapiminda **Python Django** framework`u kullanılacak
- Nesne tanima **Python OpenCv** kutuphanesi ile yapilacak
- Veri tabani islemleri **MsSql** ile saglanacak
- Kullanicinin cektigi fotoğraflar makine öğrenmesine katkıda bulunarak doğruluk oranini artiracak

Nasil Calisir

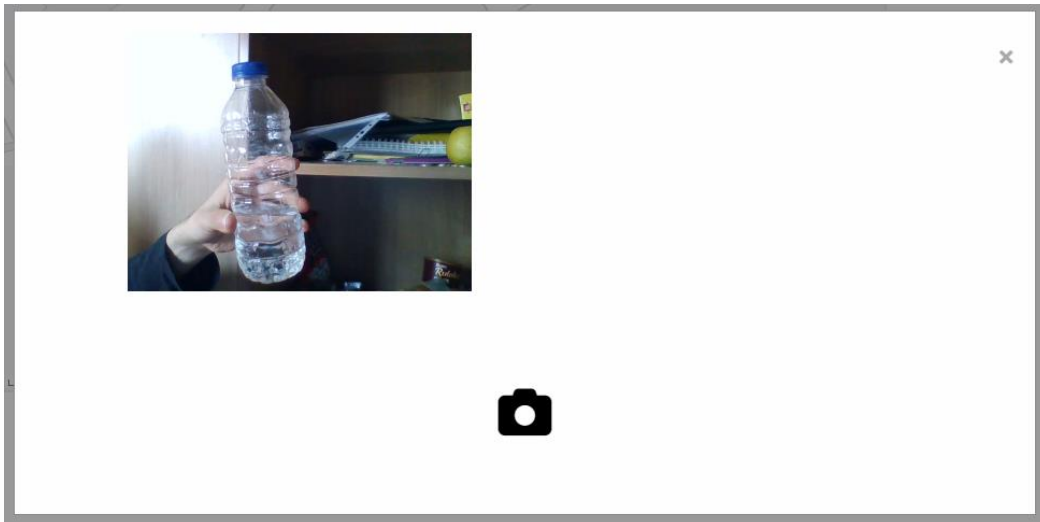
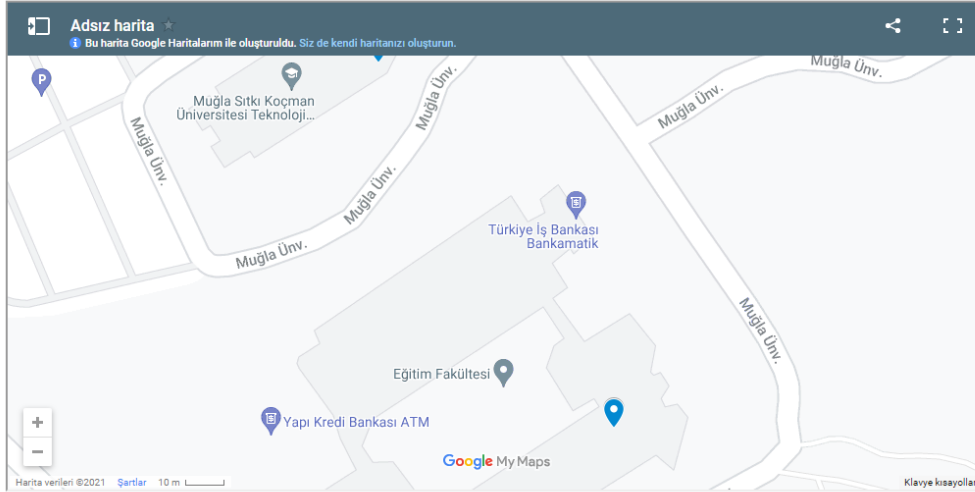
Ana sayfada bulunan 'fotograf al' butonuna tiklayarak kameraniz ile atmak istediginiz nesnenin fotografini cekiyorsunuz. Fotograf veritabaninda bulunan ayni nesneyi tanir, eslesir ve o nesnenin bilgilerini bize yazı tipinde gosterir.

Veriler

Veriler suan icin okul icinde bulunabilecek nesneleri kapsamaktadir. Veri ornegi asagidaki gibi olup, pdf link verilmiştir.

1. İçecek kartonu --Kağıtlı içecek kartonları sadece kartondan yapılmaz -plastik ve alüminyum katmanlar vardır, bu nedenle aynı şekilde geri dönüştürülemez. İçecek kartonlarının geri dönüştürülebilmesi için özel ekipman ve işleme tabi tutulması gerekir. EPA, aseptik ve katlanır karton, kağıt torba, ambalaj kağıdı ve diğer kağıt ve karton ambalaj üretiminin 2017 yılında toplam 8,5 milyon tona ulaştığını ve bunun 2/3'ünden fazlasının çöpe atıldığını tahmin ediyor. Yeniden Kullanın Süt veya meyve suyu kartonları ile el yapımı oyuncaklardan kuş kutularına kadar yeniden kullanmanın birçok yolu vardır. Şimdi fikirleri keşfetmeye başlayın! <https://www.redtedart.com/milk-carton-crafts-juice-tetra-pack-crafts/> "Geri dönüşüm" kutusu (muhtemel) "Geri Dönüşüm" kutusuna atmadan önce hava boşlukları olan kutuları ve diğer kağıt öğeleri düzleştirin. Geri dönüştürülebilir ürünlerinizin artık yiyecek kalıntısı içermediğinden emin olun. Gerekirse yiyecek ve içecek kaplarını hafifçe yıkayın. Geri dönüşüm plastik poşetsiz, temiz, boş ve kuru olmalıdır.

[Tum veriler](#)



11. Kamera açma ve fotoğraf alma Model'inin javascript ile yazılmış hali:

```
let camera_button = document.querySelector("#camera-btn"); let video =
document.querySelector("#video");
let click_button = document.querySelector("#click-photo"); let canvas =
document.querySelector("#canvas"); camera_button.addEventListener('click', async
function() {
    let stream = await navigator.mediaDevices.getUserMedia({ video: true, audio: false }); video.srcObject = stream;
});
click_button.addEventListener('click', function() { canvas.getContext('2d').drawImage(video, 0, 0, canvas.width,
    canvas.height); let image_data_url = canvas.toDataURL('image/jpeg');
    // data url of the image
    console.log(image_data_url);
});
var photo_modal = document.getElementById("photoModal"); var photo_btn =
document.getElementById("camera-btn");
var photo_span = document.getElementsByClassName("close")[0]; photo_btn.onclick = function() {
    photo_modal.style.display = "block";
}
photo_span.onclick = function() {
    photo_modal.style.display = "none";
}

var pick_modal = document.getElementById("pickModal"); var pick_btn =
document.getElementById("pick-btn");
var pick_span = document.getElementsByClassName("close")[1]; pick_btn.onclick = function () {
    pick_modal.style.display = "block";
}
pick_span.onclick = function () {
    pick_modal.style.display = "none";
}

window.onclick = function(event) {
    if (event.target == photo_modal || event.target == pick_modal) { photo_modal.style.display =
        "none";
        pick_modal.style.display = "none";
    }
}
```

Python ile yaptigimiz deneme projesinin kodlari:

```
import cv2
backsub = cv2.createBackgroundSubtractorMOG2() kamera =
cv2.VideoCapture(0) kamera.set(cv2.CAP_PROP_BUFFERSIZE, 1024)
genislik = 704
yukseklk = 288
konumSayac = 0
girenSayisi = 0
simdiki_veri = []
gecmis_veri = [] while
True:
    ret, frame = kamera.read() if kamera:
        fgmask = backsub.apply(frame, None, 0.018)
        cv2.line(frame, (0, yukseklik), (genislik, yukseklik), (0, 255, 0), 2) contours, hierarchy = cv2.findContours(
            fgmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
        for contour in contours:
            (x, y, w, h) = cv2.boundingRect(contour) if w > 65 and h >
            65:
                simdiki_veri.append([x, y])
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2) yer_listesi = []
        try:
            for i in range(len(simdiki_veri)): mini = 10000
            for k in range(len(gecmis_veri)):
                diff_x = simdiki_veri[i][0]-gecmis_veri[k][0] diff_y =
                simdiki_veri[i][1]-gecmis_veri[k][1] distance =
                (diff_x*diff_x)+(diff_y * diff_y) if(distance < mini):
                    mini = distance
                    konumSayac = k
            yer_listesi.append(konumSayac) except
            IndexError:
                continue
        try:
```



```

        for i in yer_listesi:
            for k in range(1, len(yer_listesi)): if i == yer_listesi[k]:
                yer_listesi.pop(k) yer_listesi.insert(k,
                    "pass")
    except IndexError:
        continue
    for i in range(len(simdiki_veri)): try:
        if yer_listesi[i] == "pass": pass
        else:
            y_previous = gecmis_veri[yer_listesi[i]][1] if(simdiki_veri[i][1] < yukseklik and y_previous >
                yukseklik):
                girenSayisi = girenSayisi+1

    except IndexError:
        continue
    gecmis_veri = simdiki_veri simdiki_veri
    = []

    cv2.putText(frame, "say: "+str(girenSayisi), (0, 20),
        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
    cv2.imshow("Python OpenCV", frame)

    key = cv2.waitKey(60) if key ==
    ord('q'):
        kamera.release()
        cv2.destroyAllWindows() break

```

Node js ile python calistirma denemesi:

```

const express = require('express')
const {spawn} = require('child_process'); const app =
express()
const port = 3000 app.get('/', (req, res) =>
{

    var dataToSend;
    // spawn new child process to call the python script const python =
    spawn('python', ['deneme.py']);
    // collect data from script python.stdout.on('data', function
    (data) {
        console.log('Pipe data from python script ...'); dataToSend =
        data.toString();
    });
    // in close event we are sure that stream from child process is closed python.on('close', (code) => {
    console.log('child process close all stdio with code ${code}');
    // send data to browser
    res.send(dataToSend)
    });

})
app.listen(port, () => console.log('Example app listening on port
${port}!'))

```

Sinifta yaptigimiz flask ile api calismasi

```

from flask import Flask, jsonify
# vsdl olarak yap. parametreyi url üzerinden yonlendirme security acik olusabilir app = Flask(__name__)

wastes = [{ 'id': "1 ",
    'atik-foto': "FOTOGRAF",
    'atik-isim': "Icecek kartonu",
    'atik-detay': "EPA, aseptik ve katlanir karton, kagit torba, ambalaj kagidi ve diger kagit ve karton ambalaj uretiminin 201
    }, {
    'id': "2",
    'atik-foto': "fotograf", 'atik-isim': "Plastik
    sise",
    'atik-detay': " Gerekirse yiyecek ve icecek kaplarini hafifce yikayin.Geri donusum plastik posetsiz, temiz, bos ve kuru olm
    }]

@app.route("/")

```

```

def index():
    return "Welcome tt api"

@app.route("/wastes", methods=['GET']) def get():
    return jsonify({'Wastes':wastes })

@app.route("/wastes/<int:id>", methods=["GET"]) def
get_waste(id):
    return jsonify({'waste':wastes[id]})

@app.route("/wastes", methods=['POST']) def create():
    waste = { 'id':"2",
              'atik-foto':"fotograf", 'atik-isim':"Plastik
              sise",
              'atik-detay':" Gerekirse yiyecek ve icecek kaplarini hafifce yıkayin.Geri donusum plastik posetsiz, temiz, bos ve kuruolm
              }
    wastes.append(waste)
    return jsonify({'Created': waste})

@app.route("/wastes/<int:id>", methods = ['PUT']) def
waste_update(id):
    wastes[id]['atik-detay'] = "XYZ" return
    jsonify({'waste':wastes[id]})

@app.route("/wastes/<int:id>", methods = ['DELETE']) def delete(id):
    wastes.remove(wastes[id]) return
    jsonify({'result':True})

if __name__ == "__main__": app.run(debug=True)

```

ImageNet`den verilerin filtrelenerek alınması için gereken kodlar

```

#!/usr/bin/env python3 import
os
import numpy as np
import requests import
argparse import json
import time import
logging import csv

from multiprocessing import Pool, Process, Value, Lock

from requests.exceptions import ConnectionError, ReadTimeout, TooManyRedirects, MissingSchema, InvalidURL parser =

argparse.ArgumentParser(description='ImageNet image scraper')
parser.add_argument('-scrape_only_flickr', default=True, type=lambda x: (str(x).lower() == 'true')) parser.add_argument('-number_of_classes',
default = 10, type=int)
parser.add_argument('-images_per_class', default = 10, type=int) parser.add_argument('-
data_root', default="", type=str)
parser.add_argument('-use_class_list', default=False, type=lambda x: (str(x).lower() == 'true')) parser.add_argument('-class_list', default=[],
nargs="*")
parser.add_argument('-debug', default=False, type=lambda x: (str(x).lower() == 'true')) parser.add_argument('-
multiprocessing_workers', default = 8, type=int)

args, args_other = parser.parse_known_args()

if args.debug:
    logging.basicConfig(filename='imagenet_scraper.log', level=logging.DEBUG)

if len(args.data_root) == 0:
    logging.error("-data_root is required to run downloader!") exit()

if not os.path.isdir(args.data_root):
    logging.error(f"folder {args.data_root} does not exist! please provide existing folder in -data_root arg!") exit()

IMAGENET_API_WNID_TO_URLS = lambda wnid: f'http://www.image-net.org/api/imagenet.synset.geturls?wnid={wnid}' current_folder =

os.path.dirname(os.path.realpath(__file__))

class_info_json_filename = 'imagenet_class_info.json'
class_info_json_filename = os.path.join(current_folder, class_info_json_filename)

```

```

class_info_dict = dict()

with open(class_info_json_filepath) as class_info_json_f: class_info_dict =
    json.load(class_info_json_f)

classes_to_scrape = []

if args.use_class_list == True: for item in
    args.class_list:
        classes_to_scrape.append(item) if item not in
        class_info_dict:
            logging.error(f'Class {item} not found in ImageNette') exit()

elif args.use_class_list == False:
    potential_class_pool = []
    for key, val in class_info_dict.items():

        if args.scrape_only_flickr:
            if int(val['flickr_img_url_count']) * 0.9 > args.images_per_class: potential_class_pool.append(key)
        else:
            if int(val['img_url_count']) * 0.8 > args.images_per_class: potential_class_pool.append(key)

    if (len(potential_class_pool) < args.number_of_classes):
        logging.error(f'With {args.images_per_class} images per class there are {len(potential_class_pool)} to choose from.') logging.error(f'Decrease number of classes or decrease images per
        class.')
        exit()

    picked_classes_idxes = np.random.choice(len(potential_class_pool), args.number_of_classes, replace = False) for idx in picked_classes_idxes:
        classes_to_scrape.append(potential_class_pool[idx])

print("Picked the following clases:")
print([ class_info_dict[class_wnid]['class_name'] for class_wnid in classes_to_scrape ])

imagenet_images_folder = os.path.join(args.data_root, 'imagenet_images') if not
os.path.isdir(imagenet_images_folder):
    os.mkdir(imagenet_images_folder)

scraping_stats = dict( all=dict(
    tried=0, success=0,
    time_spent=0,
),
    is_flickr=dict( tried=0,
        success=0,
        time_spent=0,
    ),
    not_flickr=dict( tried=0,
        success=0,
        time_spent=0,
    )
)

def add_debug_csv_row(row):
    with open('stats.csv', "a") as csv_f:
        csv_writer = csv.writer(csv_f, delimiter=",")
        csv_writer.writerow(row)

class MultiStats(): def __init__(
    self):

        self.lock = Lock()

        self.stats = dict( all=dict(
            tried=Value('d', 0),
            success=Value('d', 0),
            time_spent=Value('d', 0),
        ),
            is_flickr=dict( tried=Value('d', 0),
                success=Value('d', 0),
                time_spent=Value('d', 0),
            ),
            not_flickr=dict( tried=Value('d', 0),
                success=Value('d', 0),
                time_spent=Value('d', 0),
            )
        )

```

```

    )
    )
    def inc(self, cls, stat, val): with self.lock:
        self.stats[cls][stat].value += val

    def get(self, cls, stat): with self.lock:
        ret = self.stats[cls][stat].value return ret

multi_stats = MultiStats()

if args.debug: row = [
    "all_tried", "all_success",
    "all_time_spent", "is_flickr_tried",
    "is_flickr_success",
    "is_flickr_time_spent",
    "not_flickr_tried",
    "not_flickr_success",
    "not_flickr_time_spent"
]
add_debug_csv_row(row)

def add_stats_to_debug_csv(): row = [
    multi_stats.get('all', 'tried'), multi_stats.get('all', 'success'),
    multi_stats.get('all', 'time_spent'), multi_stats.get('is_flickr',
    'tried'), multi_stats.get('is_flickr', 'success'),
    multi_stats.get('is_flickr', 'time_spent'),
    multi_stats.get('not_flickr', 'tried'), multi_stats.get('not_flickr',
    'success'), multi_stats.get('not_flickr', 'time_spent'),
]
add_debug_csv_row(row)

def print_stats(cls, print_func):

    actual_all_time_spent = time.time() - scraping_t_start.value processes_all_time_spent =
    multi_stats.get('all', 'time_spent')

    if processes_all_time_spent == 0:
        actual_processes_ratio = 1.0
    else:
        actual_processes_ratio = actual_all_time_spent / processes_all_time_spent #print(f"actual all time: {actual_all_time_spent} proc

    all time{processes_all_time_spent}")

    print_func(f"STATS For class {cls}:")
    print_func(f"tried {multi_stats.get(cls, 'tried')} urls with 'f' {multi_stats.get(cls, 'success')}
        successes'")

    if multi_stats.get(cls, 'tried') > 0:
        print_func(f"{100.0 * multi_stats.get(cls, 'success') / multi_stats.get(cls, 'tried')}% success rate for {cls} urls ") if multi_stats.get(cls, 'success') > 0:
        print_func(f"{multi_stats.get(cls, 'time_spent') * actual_processes_ratio / multi_stats.get(cls, 'success')} seconds spent per {

lock = Lock()
url_tries = Value('d', 0) scraping_t_start = Value('d',
time.time()) class_folder = "
class_images = Value('d', 0) def

get_image(img_url):

    #print(f"Processing {img_url}") #time.sleep(3)

    if len(img_url) <= 1: return

    cls_imgs = 0 with
    lock:
        cls_imgs = class_images.value

    if cls_imgs >= args.images_per_class: return

```

```

logging.debug(img_url) cls = "

if 'flickr' in img_url: cls = 'is_flickr'
else:
    cls = 'not_flickr'
    if args.scrape_only_flickr: return

t_start = time.time()

def finish(status):
    t_spent = time.time() - t_start multi_stats.inc(cls, 'time_spent',
    t_spent) multi_stats.inc('all', 'time_spent', t_spent)

    multi_stats.inc(cls, 'tried', 1)
    multi_stats.inc('all', 'tried', 1)

    if status == 'success': multi_stats.inc(cls, 'success', 1)
        multi_stats.inc('all', 'success', 1)

    elif status == 'failure': pass
    else:
        logging.error(f'No such status {status}!!!') exit()
    return

with lock:
    url_tries.value += 1
    if url_tries.value % 250 == 0: print(f'\nScraping
    stats:') print_stats('is_flickr', print)
        print_stats('not_flickr', print) print_stats('all',
        print)
        if args.debug: add_stats_to_debug_csv()

try:
    img_resp = requests.get(img_url, timeout = 1) except
ConnectionError:
    logging.debug(f'Connection Error for url {img_url}') return finish('failure')
except ReadTimeout:
    logging.debug(f'Read Timeout for url {img_url}') return finish('failure')
except TooManyRedirects:
    logging.debug(f'Too many redirects {img_url}') return
    finish('failure')
except MissingSchema: return
    finish('failure')
except InvalidURL:
    return finish('failure')

if not 'content-type' in img_resp.headers: return
    finish('failure')

if not 'image' in img_resp.headers['content-type']: logging.debug("Not an image")
    return finish('failure')

if (len(img_resp.content) < 1000): return
    finish('failure')

logging.debug(img_resp.headers['content-type']) logging.debug(f'Image size
{len(img_resp.content)}')

img_name = img_url.split('/')[1] img_name =
img_name.split("?")[0]

if (len(img_name) <= 1): return
    finish('failure')

img_file_path = os.path.join(class_folder, img_name) logging.debug(f'Saving image in
{img_file_path}')

with open(img_file_path, 'wb') as img_f:
    img_f.write(img_resp.content)

with lock:
    class_images.value += 1

```



```

logging.debug(f'Scraping stats') print_stats('is_flickr',
logging.debug) print_stats('not_flickr', logging.debug)
print_stats('all', logging.debug)

return finish('success')

for class_wnid in classes_to_scrape:

    class_name = class_info_dict[class_wnid]["class_name"] print(f'Scraping images for
class "{class_name}"') url_urls = IMAGENET_API_WNID_TO_URLS(class_wnid)

    time.sleep(0.05)
    resp = requests.get(url_urls)

    class_folder = os.path.join(imagenet_images_folder, class_name) if not
os.path.exists(class_folder):
        os.mkdir(class_folder)

    class_images.value = 0

    urls = [url.decode('utf-8') for url in resp.content.splitlines()]

    #for url in urls:
    #    get_image(url)

    print(f"Multiprocessing workers: {args.multiprocessing_workers}") with
Pool(processes=args.multiprocessing_workers) as p:
        p.map(get_image,urls)

```

ImageNet `den alacağımız birkaç örnek kod:

```

n02876657,bottle,1228
,737
n02877266,bottle,1384
,1375
n02946921,can,1203,625



```

Bu kodlar kullanılarak 160gb veri içinden sadece bu verileri indirebileceğiz

Verileri araştırırken kullandığımız kaynaklar

- <https://www.epa.gov/recycle>
- http://www.cevresehirkutuphanesi.com/assets/files/slider_pdf/7NiB09QPiroJ.pdf
- https://tr.wikipedia.org/wiki/Geri_dönüşüm
- <https://www.bso-oberursel.de/de/downloads-formulare/abfall/bso-grosswohnanl-tuerkisch.pdf?cid=bkk>
- <https://www.yesilist.com/karton-bardak-kagit-pecete-pizza-kutusu-geri-donustum-kutularina-atilmamasi-gereken-12-sey/>
- <https://www.yesilist.com/geri-donusume-giris-hangi-cop-hangi-kutuya-atilmali/>

12. Veritabanı Verileri

| Column Name | Data Type | Allow Nulls | Column Name | Data Type | Allow Nulls |
|--|--------------|-------------------------------------|--|--------------|-------------------------------------|
|  Id | int | <input type="checkbox"/> |  Id | int | <input type="checkbox"/> |
| uye_ad | nvarchar(50) | <input checked="" type="checkbox"/> | atik_foto | image | <input checked="" type="checkbox"/> |
| uye_soyad | nvarchar(50) | <input checked="" type="checkbox"/> | atik_isim | nvarchar(50) | <input checked="" type="checkbox"/> |
| uye_no | int | <input checked="" type="checkbox"/> | atik_no | int | <input checked="" type="checkbox"/> |
| uye_cinsiyet | nvarchar(50) | <input checked="" type="checkbox"/> | tespit_edilen_id | nvarchar(50) | <input checked="" type="checkbox"/> |
| uye_dogum_tarihi | int | <input checked="" type="checkbox"/> | tespit_edilen_no | int | <input checked="" type="checkbox"/> |
| | | <input type="checkbox"/> | | | <input type="checkbox"/> |

| | Column Name | Data Type | Allow Nulls |
|---|----------------|-----------|-------------------------------------|
| 🔍 | Id | int | <input type="checkbox"/> |
| | kutu_id | int | <input checked="" type="checkbox"/> |
| | kutu_no | int | <input checked="" type="checkbox"/> |
| | kutu_koordinat | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| | Column Name | Data Type | Allow Nulls |
|---|-------------|--------------|-------------------------------------|
| 🔍 | Id | int | <input type="checkbox"/> |
| | atik_foto | image | <input checked="" type="checkbox"/> |
| | atik_isim | nvarchar(50) | <input checked="" type="checkbox"/> |
| | atik_detay | nvarchar(50) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| | Column Name | Data Type | Allow Nulls |
|---|--------------|-----------|-------------------------------------|
| 🔍 | Id | int | <input type="checkbox"/> |
| | duyuru_id | int | <input checked="" type="checkbox"/> |
| | duyuru_no | int | <input checked="" type="checkbox"/> |
| | duyuru_tarih | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

| | Column Name | Data Type | Allow Nulls |
|---|------------------|--------------|-------------------------------------|
| 🔑 | Id | int | <input type="checkbox"/> |
| | isim | nvarchar(50) | <input checked="" type="checkbox"/> |
| | soyisim | nvarchar(50) | <input checked="" type="checkbox"/> |
| | [mac-adres] | int | <input checked="" type="checkbox"/> |
| | toplamatiksayisi | int | <input checked="" type="checkbox"/> |
| | atik_foto | image | <input checked="" type="checkbox"/> |
| | atik_isim | nvarchar(50) | <input checked="" type="checkbox"/> |
| | atik_detay | nvarchar(50) | <input checked="" type="checkbox"/> |
| | atik_no | int | <input checked="" type="checkbox"/> |
| | tesipt_edilen_id | int | <input checked="" type="checkbox"/> |
| | tespit_edilen_no | int | <input checked="" type="checkbox"/> |
| | uye_ad | nvarchar(50) | <input checked="" type="checkbox"/> |
| | uye_soyad | nvarchar(50) | <input checked="" type="checkbox"/> |
| | uye_cinsiyet | nvarchar(50) | <input checked="" type="checkbox"/> |
| | uye_dogum_tarihi | int | <input checked="" type="checkbox"/> |
| | uye_no | int | <input checked="" type="checkbox"/> |
| | kutu_id | int | <input checked="" type="checkbox"/> |
| | kutu_no | int | <input checked="" type="checkbox"/> |
| | kutu_koordinat | int | <input checked="" type="checkbox"/> |
| | duyuru_id | int | <input checked="" type="checkbox"/> |
| | duyuru_no | int | <input checked="" type="checkbox"/> |
| | duyuru_tarih | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

13. Veritabani Diyagramı

