

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ


Bilgisayar Mühendisliği Bölümü

**DÜŞEN OBJELERİN
SINIFLANDIRILMASI**

Ahmet Yuşa TELLİ

Danışman
Doç. Dr. Habil KALKAN

Ocak, 2021
Gebze, KOCAELİ

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**


Bilgisayar Mühendisliği Bölümü

**DÜŞEN OBJELERİN
SINIFLANDIRILMASI**

Ahmet Yuşa TELLİ

**Danışman
Doç. Dr. Habil KALKAN**

**Ocak, 2021
Gebze, KOCAELİ**

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


Bu çalışma 28/01/2021 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Doç. Dr. Habil KALKAN	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Dr. Gökhan KAYA	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı		
Üniversite		
Fakülte		

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


ÖNSÖZ

Bu kılavuzun ilk taslaklarının hazırlanmasında emeği geçenlere, kılavuzun son halini almasında yol gösterici olan Sayın Doç. Dr. Habil KALKAN hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.


Ocak,2021

Ahmet Yuşa TELLİ

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


İÇİNDEKİLER

ÖNSÖZ.....	IV
İÇİNDEKİLER	V
ŞEKİL LİSTESİ.....	VI
TABLO LİSTESİ.....	VII
KISALTMA LİSTESİ	VIII
SEMBOL LİSTESİ	IX
ÖZET	X
SUMMARY	XI
1. GİRİŞ	1
2. PROJE TASARIMI	2
3. NESNE TESPİTİ VE GÖRÜNTÜ İŞLEME.....	4
3.1. GÖRÜNTÜ İŞLEME	4
3.1. NESNE TESPİTİ	5
3.3. ÇOKLU NESNE TESPİTİ	6
4. ARA YÜZ.....	7
5. VERİLER NASIL TUTULUYOR.....	10
6. ALGORİTMALAR.....	11
7. SINIFLANDIRMA.....	11
8. SONUÇ.....	12
9. KAYNAKLAR	13


	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

ŞEKİL LİSTESİ


Proje -1	2
Proje -2.....	3
Görüntü -1	4
Görüntü -2.....	4
Tespit -1	5
Tespit -2	5
Tespit -3	5
Çoklu -1	6
Çoklu -2	6
Arayüz -1	7
Arayüz -2	8
Arayüz -3	8
Arayüz -4	9
Arayüz -5	9

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


TABLO LİSTESİ

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

KISALTMA LİSTESİ

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

SEMBOL LİSTESİ

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


ÖZET

Gebze Teknik Üniversitesi Bilgisayar mühendisliği son sınıf öğrencisi olarak yaptığım bitirme projemde düşen gıda nesnelerinin görüntülerinden sınıflandırmayı amaçladım. Burada bir adet kameramız ve 60 cm ilerisinde beyaz bir arka planımız var.

Bu kamera ile gıda nesnelerinin görüntülerini alıp Python programla dili ile yazılmış olan algoritmamıza veriyoruz. Daha sonra bu görüntülerden belirli özellikler çıkartıp, belirlediğimiz sınıflandırma algoritmaları ile model eğitimi yapıyoruz.

Daha sonra bu modeli kullanarak sonraki nesnelerin hangi sınıfa ait olduğunu ekranda gösteriyoruz.

Kullanıcı bunların yönetimi programın ara yüzünden yapabilir. İstenirse yeni model kullanılabilir veyahut hazır modellerden seçilip onlar ile sınıflandırma yapılabilir.


	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

SUMMARY

In my graduation project as a senior year student at Gebze Technical University Computer Engineering, I aimed to classify them from the images of falling food objects. We have a camera here and a white background 60 cm ahead.

With this camera, we take images of food objects and give them to our algorithm written in Python programming language. Later, we extract certain features from these images and do model training with the classification algorithms we have determined. Then, using this model, we show on the screen which class the next objects belong to.

The user can manage them from the program interface. If desired, a new model can be used or it can be selected from ready-made models and classified with them.

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


1. GİRİŞ

İlk olarak projede bir tasarım yaptım. Bir adet Logitech C310 marka bir kamera var. Bunu yerden yaklaşık 30cm yukarı kaldırdım. Bu kameranın 60 cm karşısında beyaz bir arka plan hazırladım. Bu arka plan 45cm eninde 75 cm uzunluğunda yatay olarak duruyor. Programlama dili olarak Python 3.6, görüntü işleme için OpenCV 4.4 kullanılmıştır.

Projede bu beyaz arka planın önünden geçerek aşağı düşen gıda nesnelerinin sınıflandırılması amaçlanmıştır. Burada ilk olarak bu nesnelerin tespiti yapılmıştır. Her bir nesneyi ayrı ayrı tespit ettikten sonra bu nesneleri kayıt altına alıyoruz.

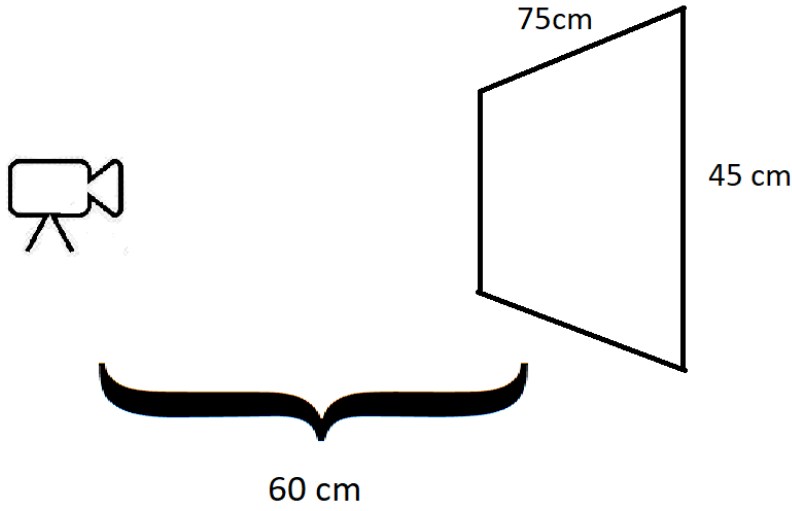
Daha sonra kameradan aldığımız görüntüleri bir görüntü işleme sürecinden geçirip belirli özellikler elde ediyoruz. Bu özellikler gelen görüntünün kırmızı, yeşil ve mavi değerlerinden oluşuyor.

Bu özelliklerle belirli sınıflandırma algoritmalarını kullanarak bir makine öğrenmesi modeli oluşturuyoruz. Bu modeli kullanıcı istediği şekilde eğitebiliyor. Daha sonra bu modeli kullanarak elindeki diğer nesnelerin sınıflarını öğrenebiliyor.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


2. PROJE TASARIMI

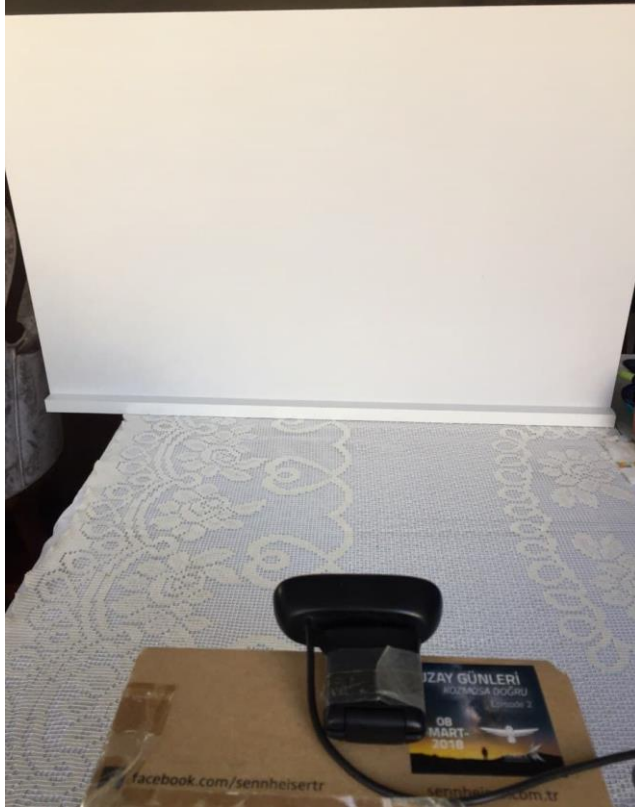
Projemizde aşağıdaki resimdeki gibi bir adet kamera ve karşısında beyaz bir arka plan bulunmaktadır.



Proje -1


Bu projede Logitech C310 marka kamera kullanılmıştır. Arka plan olarak 45 cm eninde 75 cm uzunluğunda 5mm kalınlığında MDF kullanılmıştır. Bu arka planın düz durması için 15 cm eninde 75cm uzunluğunda MDF ayak kullanılmıştır.

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Proje -2

Yukarıda benim tarafımdan kurulan düzeneği görmektesiniz.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

3. NESNE TESPİTİ VE GÖRÜNTÜ İŞLEME

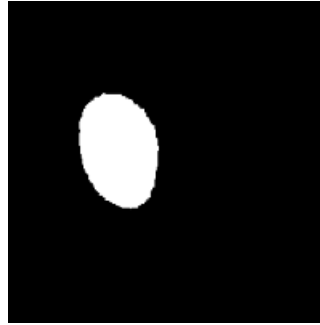
3.1. GÖRÜNTÜ İŞLEME

İlk olarak nesne tespiti yapmakla projeye başladım. Burada öncelikle kameradan görüntüler almaya başladım. Bu görüntüler ile hareket eden nesneleri tespiti için farklı yöntemler vardı. İlk olarak ben aldığım görüntüde nesneyi bulmak için neler yapmam gerektiğini araştırdım. OpenCV de gerekli olan görüntü işleme metodlarını kullandım. İlk görüntüyü alıp onu kaydettikten sonra diğer gelen sürekli görüntüler ile bu ilk görüntü arasındaki fark ile nesneyi bulmaya çalıştım. OpenCV içinde absdiff fonksiyonunu kullandım. Fakat gölgeler problem yaratıyordu. O nedenle daha sonra bu fonksiyon yerine OpenCV ‘nin içerisinde yer alan arka plan çıkartma algoritmalarından MOG isimli algoritmayı kullandım. Bu bize gölgelerin olmadığı ve ikili bir resmi çok rahat bir şekilde gösterebiliyordu.


Örnek olarak sol taraftaki kameradan aldığımız görüntü ve sağ tarafta arka planı çıkartılmış görüntü:



Görüntü -1



Görüntü -2

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

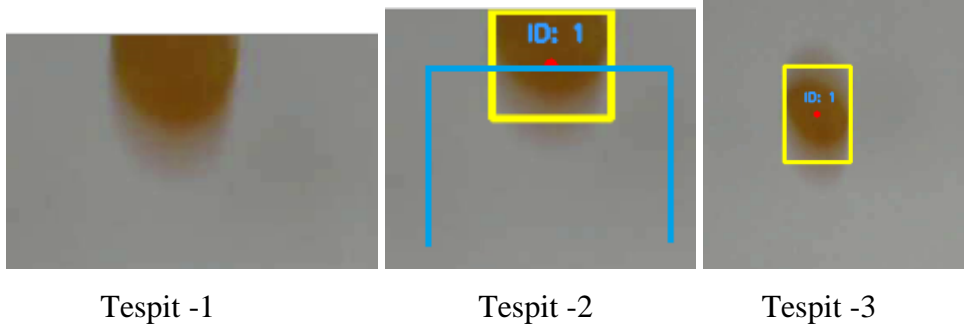
3.1. NESNE TESPİTİ


Daha sonra nesneleri saymak için görüntünün ortasından geçen nesneleri saymaya çalıştım. Yani bizim elimize 1920x720 piksel boyutunda görüntüler geliyor kameradan buradaki görüntüyü yatayda 3 kısıma ayırdım ve ortadan geçen nesneleri bir kere saymaya ve belli bir yerden sonra da tekrar saymamaya çalıştım. Çünkü yukarıda gördüğüm bir nesneyi aşağıda tekrar saymamam gerekiyordu. Yani her nesne bir defa sayılmalıydı.

Burada nesnelerin farklı hızlarda geçmesinden dolayı ortadaki saydığım alanın çok dikkatli bir şekilde ayarlanması gerekiyordu. Fakat bu yöntem tüm objeler için çalışmayabilirdi. Bu nedenle farklı bir yol denemeye karar verdim.

Bir adet nesne sınıfı oluşturdum. Bu nesne sınıfı içerisine nesneyi ilk tespit ettiğim andaki koordinatlarını kaydettim. Daha sonraki görüntüde bu nesnenin altında bir alanda mı geldiğini araştırdım. Eğer sonraki görüntüde gelen nesne önceki nesnenin altındaysa yeni bir nesne oluşturmam ve bu nesnenin devamına kaydettim. Böylece birkaç görüntüden gelen nesneleri tek bir nesne olarak alabildim.

Nesneler yukarıdan atılırken dönebiliyor ve hafifce sağa sola kayabiliyor bu benim yönetemim için biraz problem yarattı. Belli bir sınırın altında kaymayı göz ardı ederek aynı nesne olarak aldım. Aşağıdaki görüntüde kameradan gelen görüntü ve benim nereleri aramam gerektiğini gösteriyorum. Sol tarafta ilk kez nesneyi tespit ettik. Daha sonra sağ taraftaki gibi mavi alanın içerisinde bir nesne olmasını bekliyoruz. Sonraki gelen görüntüde bulunan nesne bu mavi alanın içerisindeyse bu iki nesne aynı nesnedir diyebiliriz.



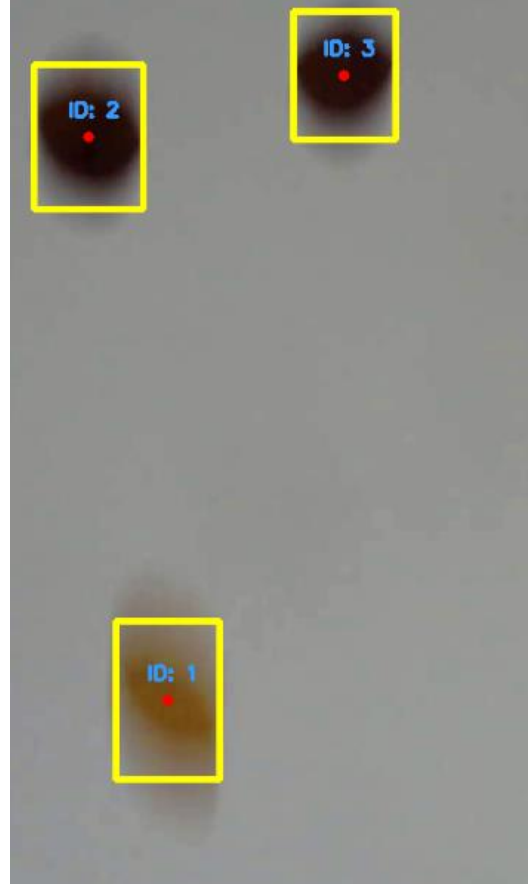
	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

3.3. ÇOKLU NESNE TESPİTİ

Yukarıda anlattığım algoritmamı çoklu nesne tespitiinde de doğru sonuçlar verdiğini gördüm.




Çoklu -1



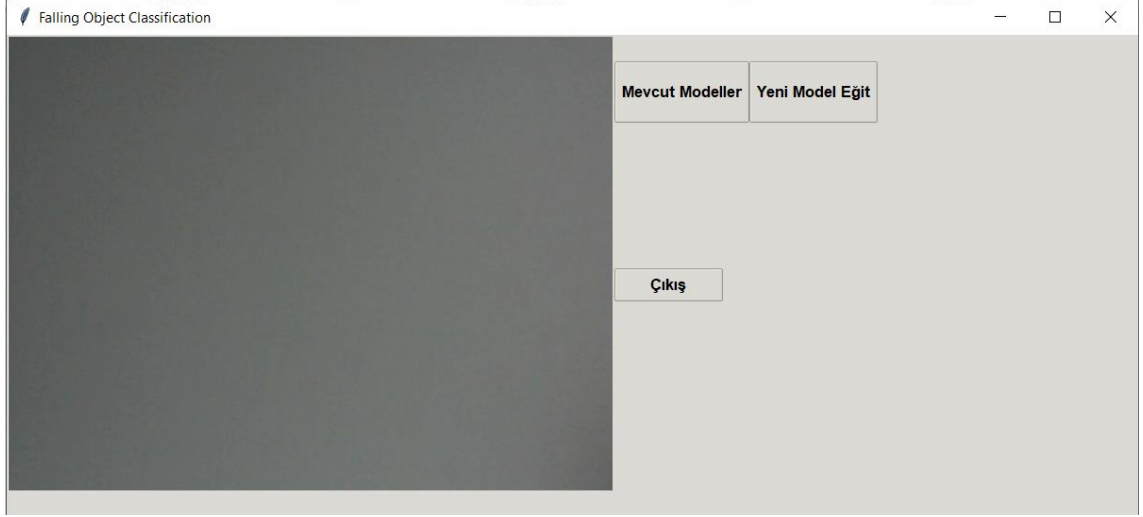
Çoklu -2

Her bir nesneyi ayrı ayrı olarak ele alıyoruz. Ve farklı numaralar veriyoruz.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


4. ARA YÜZ

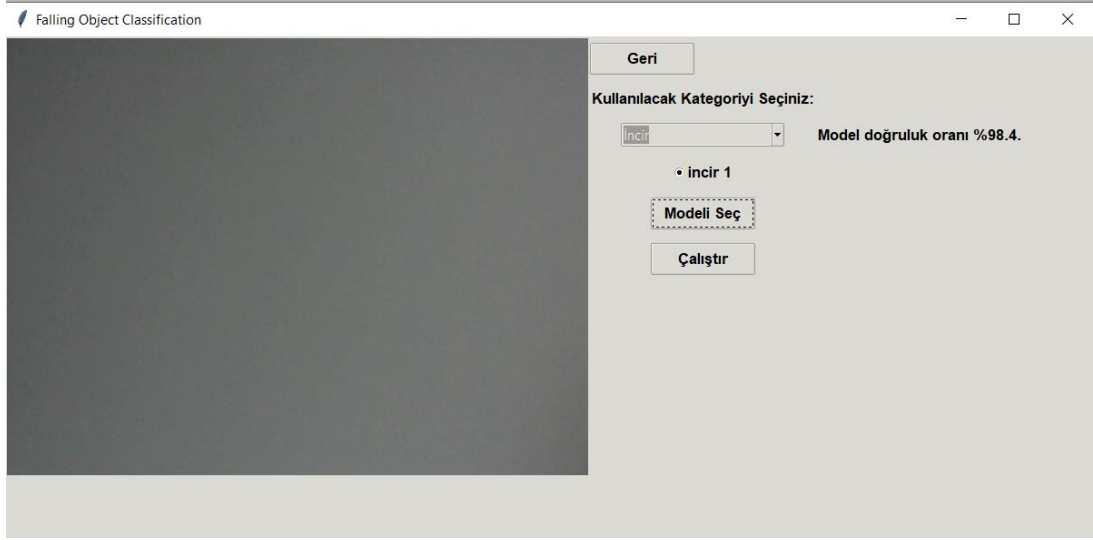
Programın arayüzünü Pyhton kütüphanesi olan Tkinter ile yaptım. Arayüzün giriş kısmı araştırdaki şekildedir. Sol tarafta kamera görüntüsü ve sağ tarafta kullanıcının model eğitip kullanabilmesi için iki adet buton yer almakta.



Arayüz -1

Soldaki “Mevcut Modeller” butonundan başlayalım. Bu buton ile kullanıcı önceden eğitmiş olduğu modellerin listesini görebilecek. İlk olarak türlerin olduğu bir liste geliyor kullanıcının karşısına. Daha sonra bu türe ait kayıtlı modeller listeleniyor. Aşağıdaki örnekte “İncir” türüne ait daha önceden kayıt edilmiş 1 adet eğitilmiş model bulunuyor. Bu modeli seçtikten sonra sağ tarafta modelin başarı yüzdesi gösteriliyor. Eğer kullanıcı bu modeli beğenmez veya istemezse yeni bir model ekleyebilir.

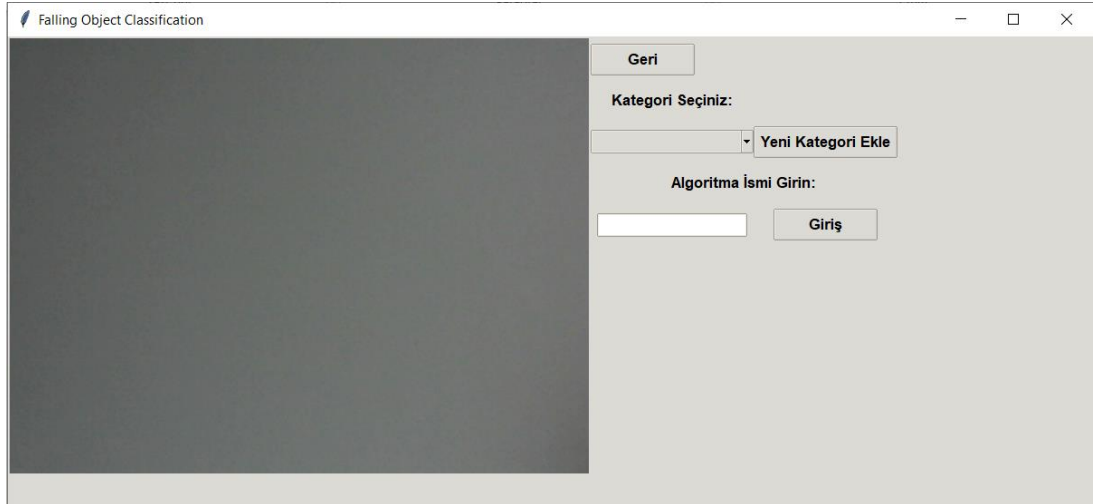
	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-




Arayüz -2

Daha sonra çalıştır diyerek anlık olarak sınıflandırma yapmaya başlıyor.

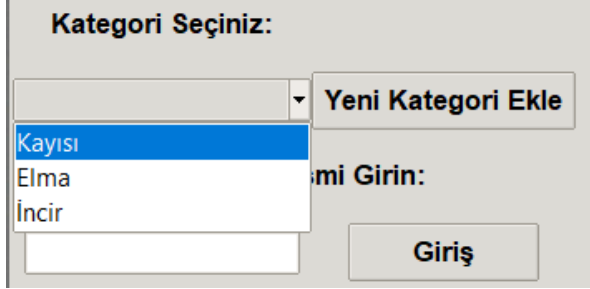
Giriş ekranında “Yeni Model Ekle” denilirse. Kategori ismi seçimi ve algoritma ismini kullanıcının girmesi bekleniyor.



Arayüz -3

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

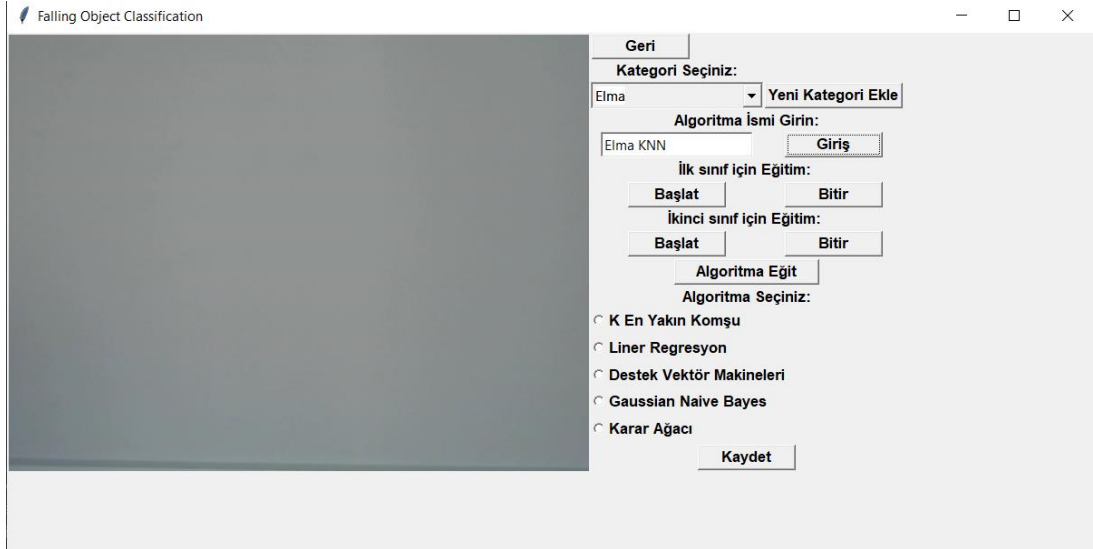
Bazı kategoriler aşağıdaki gibi listeleniyor.



Arayüz -4


Eğer kullanıcı bu türlerden farklı bir tür eklemek isterse “Yeni Kategori Ekle” butonu ile o listeye yeni bir kategori ekleyebilir.

Kategori seçimi ve algoritma ismi girildikten sonra 2 sınıf için ayrı ayrı eğitim verisi girilme işlemi kısmı açılıyor.



Arayüz -5

Burada kullanıcı model eğitime başlıyor. İlk olarak “İlk sınıf için Eğitim:” kısmındaki “Başlat” butonuna bastıktan sonra ilk sınıfa ait nesneleri yukarıdan bırakmaya başlamalı. Elindeki nesneler bittikten sonra “Bitir” butonu ile eğitimi durdurmalı. İlk sınıfın eğitimi bittikten sonra ikinci sınıf için de aynı şekilde “Başlat” ve “Bitir” butonlarıyla ikinci sınıfın eğitimini tamamlamalı.


 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

Daha sonra “Algoritma Eđit” butonuyla oradaki listelerde bulunan algoritmalar ile bu verileri eđitimi yapmalı. Bu 5 algoritma ile eđitim yaptıktan sonra kullanıcı istediđi bir algoritmayı seip “Kaydet” butonu ile eđitilmiş modeli kaydetmeli. Model eđitimi tamamlandıktan sonra “Geri” butonu ile giriş sayfasına gelip. Kayıtlı modeller içinden en son eđittiđi modeli seip sınıflandırmaya başlayabilir.

5. VERİLER NASIL TUTULUYOR

Tespit edilen nesneler bir Food sınıfı objesi olarak depolanıyor. Bu sınıfın içerisinde tespit edilen koordinatlar ve ilk görüntüdeki 10 adet ortalama kırmızı, yeşil, mavi değeri var.

Bu değeri histogram olarak buluyoruz ve ortalamalarını alıyoruz. Örnek olarak kırmızı değeri için; görüntüdeki en büyük ve en küçük değeri buluyoruz. Daha sonra bu sayılar arasında 10 farklı grup oluşturacak şekilde bir aralık belirliyoruz. Bu aralıktaki değeri farklı dizilerde saklıyoruz. En son her bir dizinin ortalama kırmızı değeri o Food objesinin içerisine kayıt ediyoruz. Bu işlemler mavi ve yeşil için ve tüm tespit edilen nesneler için yapılıyor. Yani onar adet kırmızı, mavi, yeşil verileri elimizde oluyor. Her bir nesne için 30 adet veri tutmuş oluyoruz.

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

6. ALGORİTMALAR

Projemizde 5 adet sınıflandırma algoritması var. Bunlar:


- K En yakın komşu
- Lineer Regresyon
- Destek Vektör Makineleri
- Karar Ağacı
- Gaussian Naive Bayes

İki sınıf içinde eğitim tamamlandıktan sonra tespit edilen nesnelerin kırmızı, mavi, yeşil değerleri tek tek ayrılır. Toplam 30 adet veri olduğunu söylemiştik. Bu veriler içerisinde sınıflandırma için en uygun, en etkili olan 2 tane veriyi seçiyoruz. Yani bu 30 adet veriden öyle bir iki tanesini seçmeliyiz ki bu iki veri tüm nesneler için ayırt edici bir özellik olsun. Algoritmalar daha doğru ve daha hızlı çalışması için bu 30 tane veriyi 2 veriye düşürüyoruz.

Bu veriler tespit edilen nesnelerin sayısına göre işaretlenir. Daha sonra %50 olarak ikiye ayırıyoruz. İlk olarak bir yarısı ile bu modelleri eğitiyoruz daha sonra diğer yarısı ile bu modelleri eğitiyoruz. Bu işleme cross validation denir.

7. SINIFLANDIRMA

Sınıflandırma için, model seçimi yaptıktan sonra program gelen görüntüleri işler, ve belirlenmiş olan algoritmaya gönderir. Bu algoritmadan çıkan sonuca göre, tespit edilen nesnenin etrafına çizilen dikdörtgen kırmızı ve sarı olmak üzere iki sınıf için farklı renkte gösterilir.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

8. SONUÇ

Bu projenin sonucunda yüksek bir başarı ile düşen nesneleri doğru sınıflandırdım. Elimdeki veriler ile yaptığım sınıflandırma yüzdem aşağıdaki gibidir. Elimde 40 adet veri vardı. Daha fazla olması daha iyi olurdu. Bu sonuç biraz yanıltıcı olabilir. Ne kadar fazla nesne ile model eğitirsek o kadar daha doğru sonuçlar alabiliriz.

```
kf = KFold(n_splits=2, random_state=5, shuffle=True)
model = LogisticRegression(solver='liblinear')
result = cross_val_score(model, x, y, cv=kf)
print("Avg accuracy: {}".format(result.mean()))
```

Avg accuracy: 1.0

Sonuç -1

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

9. KAYNAKLAR

[OpenCV: cv::bgsegm::BackgroundSubtractorMOG Class Reference](#)

[Object Detection and Tracking using OpenCV | by Devansh Varshney | Medium](#)

[opencv background subtraction - Stack Overflow](#)

[OpenCV: How to Use Background Subtraction Methods](#)

[Object detection with deep learning and OpenCV - PyImageSearch](#)