**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ



MÜHENDİSLİK TASARIMI

Görüntü İşleme İle Kişileri Algılama Ve Algıladığı Kişileri Anlık Olarak Adı Soyadı Tarih Saat Bilgilerinin Veri Tabanına Yazdırılması

HAZIRLAYAN

Mert İslam UĞURLU

DANIŞMAN

Prof. Dr. İbrahim ÇİL

|  |  |  |
| --- | --- | --- |
|  |  |  |

OCAK 2023

İÇİNDEKİLER

[İÇİNDEKİLER 3](#_Toc454189892)

[ÖZET 3](#_Toc454189893)

[BÖLÜM 1. GİRİŞ 4](#_Toc454189894)

[BÖLÜM 2. PROBLEMİN TANIMI 5](#_Toc454189895)

[BÖLÜM 3. PROBLEMİN ÇÖZÜMÜ İÇİN ÖNERİLEN YÖNTEM 6](#_Toc454189896)

[BÖLÜM 4. PROBLEMİN ÇÖZÜMÜ 8](#_Toc454189897)

BÖLÜM 5. GERÇEKCİ KISITLAR VE KOŞULLAR ALTINDA DEĞERLENDİRME 9

BÖLÜM 6. SONUÇ 10

[KAYNAKLAR 1](#_Toc454189899)1

ÖZET

Buradaki amacımız okul , sınıf , hastane , fabrika veya gibi yerlerde çalışan işçiler veya örneğin öğrenciler sınıftan içeriye girdiklerinde yüzlerinin tanınıp sınıf yoklamasının otomatik olarak görüntü işleme algoritmasıyla alınması hedeflenmiştir. Python opencv , dlib , face\_recognition , datetime , sqlite3, kütüphaneleri kullanılarak kameradan veya fotoğraf makinesinden yüklenen görüntülerin tanımlanabilmesi için bazı işlemlerin yapılması gerekmektedir . Bu durumda yazılım programları ve bir çok programlama dilleri bulunmaktadır. İnsan yüzlerinin algılanması , kişilerin tanınması ve bu tanınan kişilerin bir veritabanına Ad – Soyadı-Tarih-Saat Şeklinde Aktarılması. Sonuca vardığımız bu çalışmada %99 oranında başarıya varılmıştır.

**Anahtar kelimeler:** OpenCv , Yüz Algılama , Nesne Tespiti , Veritaban

# GİRİŞ

OpenCV, bir resim ya da video içindeki anlamlı bilgileri çıkarıp işleyebilmek için INTEL tarafından C ve C++ dilleri kullanılarak geliştirilmiş, açık kaynak kodlu bir “Bilgisayarla Görme” kütüphanesidir. (Erişti, Ezgi 2010 )

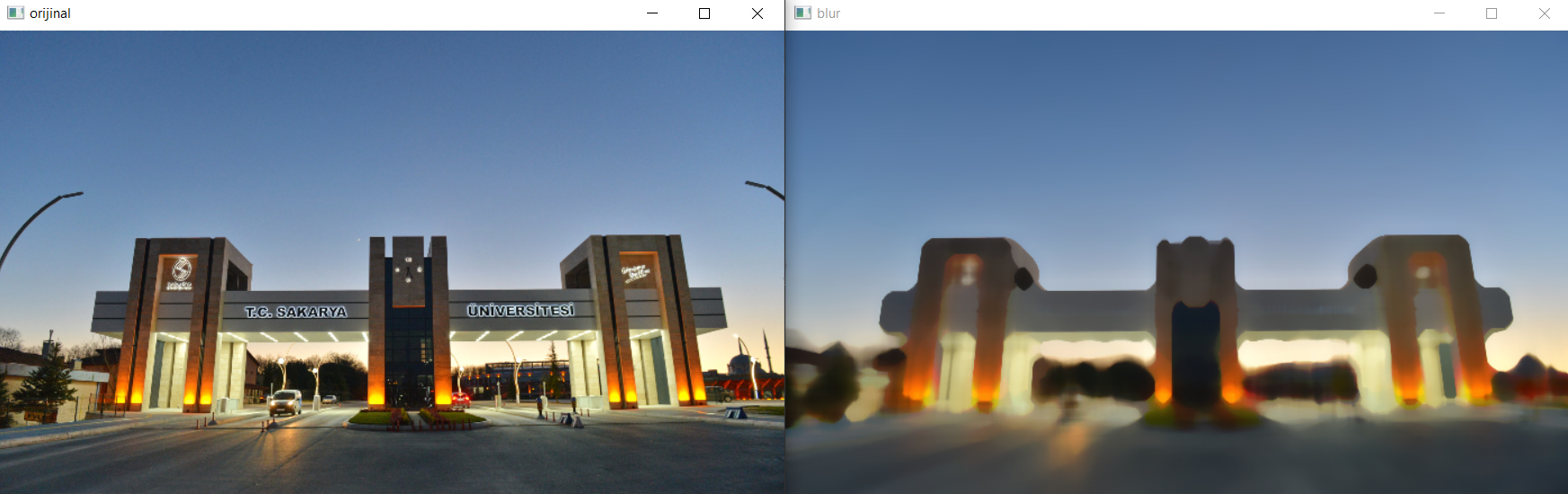
Bilgisayarla görme, ürünlerde yüksek kalite ve güvenlik gibi bazı gereksinimleri karşılamak için otomatik, zararsız ve düşük maliyetli alternatif bir teknik sunmaktadır. Mevcut teknolojide görüntü işlemeyle insan faktörünü en aza indirgeyerek verimlilik arttırılması ve maliyetin azaltılması mümkün olabilmektedir. HAYIT, T., Erbay, H., & HAYIT, F. (2018).

Bankalar, hastaneler, okullar, internet siteleri gibi hayatın her alanında insan ve bilgisayar etkileşimi alanından bir uygulama görmek mümkündür. Tüfekçi, M., & Karpat, F. (2019).

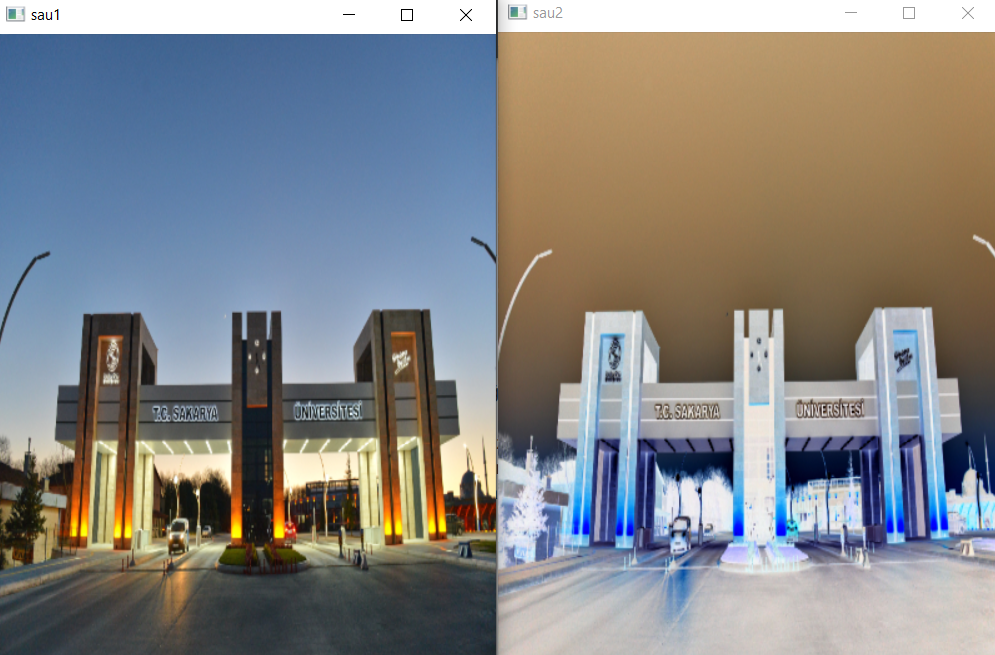
Önemi her geçen gün gittikçe artan bir yer olan yüz tanıma modele tanıtılan insan yüzlerinin yazılım tarafından tanınmasını ve bu tanınan kişilerin veritabanına kaydedilmesi amaçlanmıştır.

Tıp, Askeri, Endüstriyel ve Coğrafi Sistemler gibi birçok alanda kullanılan görüntü işleme teknikleri, güvenlik sistemleri alanında da yaygın olarak kullanılmaktadır. Parmak izi, iris ve yüz tanıma gibi uygulamalar güvenlik alanında görüntü işleme teknikleri kullanılarak yapılabilmektedir. (Eldem, A., Eldem, H., & Palali, A. 2017).

Görüntü işleme ile veritabanından veya herhangi bir yerden yüklenen fotoğraf veya video gibi görüntüler üstünde işlem yapabilmemizi sağlar. Görüntü işleme herhangi bir fotoğrafın veya videonun piksellerini bulanıklaştırma , netleştirme , gri tona çevirme gibi bir çok işlemi yapmamızı sağlar.

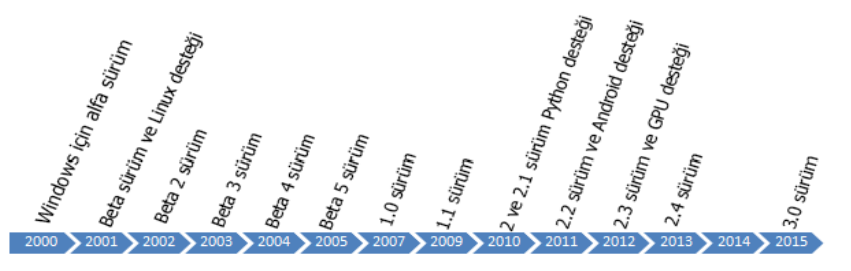


Herhangi bir fotoğrafta videoda veya anlık olarak alınan bir görüntü üzerinde sayısal işlemlerinin yapılması gerekmektedir. Sayısal İşlemler; Fotoğrafta olan mevcut renklerin sayısal olarak tanımlanması ve bunların üzerinde değişiklikler yaparak yorumlanabilir hale getirilmesidir.



Bu projede bilgisayar kamerasından alınan anlık görüntü ile kişilerin yüz bölgesi algılanmıştır. Modele tanıtılan fotoğraflar diğer fotoğraflar ile karşılaştırılarak doğru sonuçlar elde edilmesi beklenmiştir. Görüntü işlemenin kullanılabilmesi için OpenCV ve face\_recognition kütüphaneleri kullanılmıştır.

OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500’den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir. Pişkin, M. E. S. U. T. (2016)



Yıllara Göre OpenCV Sürümleri

Zayas ve ark. (1989), araştırmalarında buğday çeşitlerinin tanınmasında görüntü işleme tekniğinden yararlanmışlardır. Zayas ve ark. (1996); 17 farklı buğday çeşidine ait tanelerin görüntü işleme tekniğiyle şekil özelliklerinin belirlenmesi ve sınıflanması konusu üzerinde durmuşlardır. DEMİRBAŞ, H. Y., & DURSUN, İ. (2007)

Günümüz teknolojisinde bilgisayar destekli tespit uygulamaları tıp alanına büyük katkılar sağlamayı başarmıştır. Bilgisayar Destekli Tespit (BDT) yazılımları, ileri örüntü tanıma ve görüntü işleme yöntemlerini kullanarak radyoloji uzmanlarına medikal görüntülerdeki anormalliklerin tespitinde yardımcı olan yazılımlardır. Bu yazılımlar radyoloji uzmanlarına hem zaman kazandırır, hem de karar verme aşamasında yapma ihtimalleri olan hataları en aza indirir. Bu tezde, beyin manyetik rezonans görüntülerindeki tümörlerin tespit edilmesini amaçlayan yöntemler ortaya konmuştur. Kazdal, S. (2013)

Manyetik Rezonans (MR), canlıların içyapısını görüntüleme amacıyla daha çok tıpta kullanılan bir yöntemdir. Osmanoğlu, U. Ö., Mutlu, F., Gürsoy, H., & ŞANLISOY, S. (2016).

İnsanlara zarar verenin depremden çok depremde yıkılan binaların olduğu gerçeği düşünüldüğünde taşıyıcı sistem tasarımının mimarların ve mühendislerin ortak sorumluluğunda olduğu açıktır. Hesaplamalar doğru yapılsa bile taşıyıcı sistem kurgusunda yapılan hatalar nedeniyle yapılar depremlerden olumsuz yönde etkilenirler. Bu hataları en aza indirgeyebilmek için derin öğrenme ve görüntü işleme metotları kullanılarak taşıyıcı sistem kurgusunun düzenli veya düzensiz olduğuna dair DK asistanından yorum almak mimarlara ilk tasarım aşamasında yardımcı olacaktır. Bingöl, K., Aslı, E. R., Örmecioğlu, H. T., & Arzu, E. R. (2020).

Son çeyrek yüzyılda araç sayısının artması ve trafikte oluşan sorunlar, otomatik araç tanıma ve trafik akış kontrolü sistemlerine duyulan ihtiyacı arttırmıştır. Bu amaca yönelik çalışmalar temel olarak araçları özel bir noktadan geçerken tanımlamak, aracın konumunu belirlemek, davranışlarını gözlemlemek ve bu verileri kullanarak trafik denetimi sağlamaya yöneliktir. Bilgisayar tabanlı görsel sistemlerle araç plakası tanıma yeni geliştirilen bir sistemdir. Çevik, K. K., & Abdülkadir, Ç. A. K. I. (2010).

# PROBLEMİN TANIMI

Sınıfa giren öğrencilerin yoklamasının öğretmen tarafından alınıp zaman kaybedilmesi yerine yoklamayı kameranın alması üzerinde durulan bir problemdir. Günümüzde bir çok yere uygulanabilir hale getirilebilir örneğin fabrika , güvenlik kurumları , okul , hastane fabrikada işçilerin fabrikaya girdiklerinde kart okutmak yerine kamerayla otomatik olarak tanınması ve fabrikanın veritabanına işçi giriş çıkış saatlerinin tanımlanması , hasta kayıtta hastalar ID vererek kaçıncı hastanın girdiğini kaçıncı hastanın çıkış işlemi yaptığını ne kadar süre kaldığını bu gibi bilgileri bir çok yerlerde yararlı hale getirerek kullanabiliriz.

# PROBLEMİN ÇÖZÜMÜ İÇİN ÖNERİLEN YÖNTEM

Bu bölümde bahsedilen problemin nasıl çözüleceği adım adım gösterilmedir. Eğer önerilen yöntem belirli adımları kapsıyor ise bu adımlar sırasıyla (Aşağıdaki örnek1, örnek 2 ve örnek 3’te gösterildiği gibi) verilebilir. Aşağıdaki formatları kullanabilirsiniz, ekleyebilir, kendi çözüm adımlarınıza göre oluşturabilirsiniz.

Örnek 1

|  |  |  |
| --- | --- | --- |
| **Çözüm Aşamaları** | **Yapılan İş** | **Çözüm Aracı, Yöntem, Teknik, Program vs.** |
| 1 | Program Seçimi | PyCharm |
| 2 | Kişi Tanıma | Python-OpenCV |
| 3 | Tanınan Kişilerin DB Aktarılması | SQL |
|  |  |  |

* PyCharm Python geliştirme ortamıdır. Kod analizleri, grafiksel hata ayıklamacısı, versiyon kontrol sistemi ile entegre ve Django ile Python web geliştirmeleri yapılmasını sağlamaktadır.
* Python, nesne yönelimli, yorumlamalı, birimsel ve etkileşimli yüksek seviyeli bir programlama dilidir. Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır.
* SQL, verileri yönetmek ve tasarlamak için kullanılan bir dildir. SQL, kendisi bir programlama dili olmamasına rağmen birçok kişi tarafından programlama dili olarak bilinir. SQL herhangi bir veri tabanı ortamında kullanılan bir alt dildir.

# PROBLEMİN ÇÖZÜMÜ

Bu bölümde tanımlanan problemin önerilen çözüm yöntemi takip edilerek nasıl çözüleceği ile ilgili küçük bir örnek sunulmalıdır. Bu bölümde teorik ve tanımlayıcı bilgiler olmamalıdır.

Bölüm 3 belirlemiş olduğunuz, adımlara sadık kalınarak çözüm anlatılmalıdır. Örneğin örnek 2 göre problemimizi çözeceğimizi varsayarsak, bu bölümü anlatmaya şu şekilde devam edeceğiz.

**1.Aşama: Kod Derleyicisinin Seçilmesi**

Bu problemi çözebilmek için öncelikle kod derleyicimizin olması ve bilgisayarımıza kurulması gerekiyor

**2. Aşama: DB Browser for SQLite**

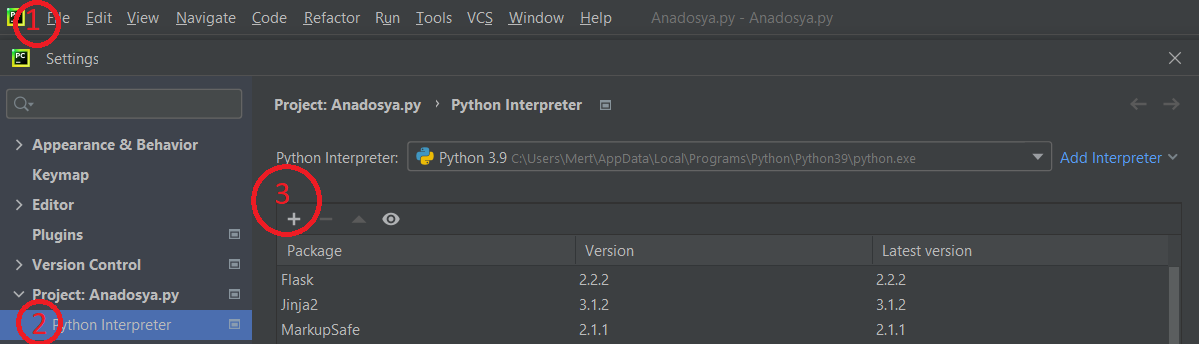
Kişilerin algılanma işleminden sonra veritabanına kaydedilebilmesi için bir veritabanına ihtiyacımız var. Biz burada DB Browser for SQLite’ı bilgisayarımıza kurduk.

**3. Aşama: Uygun Çözüm Tekniğinin Seçimi**

Çözüm tekniği belirlerken Python-OpenCV kütüphanesini kullandık

**4.Aşama: Problemin çözümü ve elde edilen sonuçlar**

Problemin çözümü için öncelikle gerekli kütüphanelerin PyCharm’a eklenmesi gerekmektedir.



Kütüphaneler sırasıyla gerekli adımlardan eklenmelidir.

import cv2 ## Kütüphaneleri ekledikten sonra import ile  
import dlib ## içeriye çağırmamız gerekmektedir.  
import face\_recognition  
from datetime import datetime ##Modelimizin tanıdığı kişiyi o anki tarih bilgisi ile kayıt  
import sqlite3 ##edebilmesi için datetime modülü eklenmektedir.

con=sqlite3.connect("taninanlariyaz.db") Kayıt edeceğimiz kişileri veritabanına yazabilmesi için  
cursor=con.cursor() Database dosyamızı oluşturuldu.

def tablo():##burada fonksiyon tanımlanarak  
 cursor.execute("create table if not exists KISILER(ad TEXT, zaman DATETIME)")

## tablonun içindeki verilerin hangi formatta tutulacağı belirtildi   
 con.commit()  
tablo() ##tabloyu sürekli olarak çağırma işlemi yapıldı

def ekle(isim,tarih): ## tablonun içindeki verileri eklemek için bir fonksiyon oluşturuldu  
 cursor.execute("Insert into KISILER Values(?,?)",(isim,tarih)) ## veriler aşağıdaki kodlardan gelicektir.  
 con.commit() ##Verilerin eklenme işlemi yapıldı.

detector=dlib.get\_frontal\_face\_detector()

##bize “detector” adında yüz bilgilerini kullanmak için gerekli olan bir fonksiyon döndürmektedir.   
Her yüz, görüntünün bulabileceği noktaları içeren bir nesnedir. Yukarıdaki kod, görüntüdeki tüm yüzleri alır ve her yüzün üzerinde bir dikdörtgen oluşturur.

mert=face\_recognition.load\_image\_file("mert.jpeg")

## Tanılacak olan yüzü modelimize dışarıdan fotoğraf formatında alarak modele yükleme işlemi yapıldı.

mert\_kodlama=face\_recognition.face\_encodings(mert)[0]

##Yüzü kodlayabilmesi için bir değişken ile belirtildi.

cap=cv2.VideoCapture(0) ##Bilgisayar kamerasından anlık olarak görüntü almak için değişken  
while True: ##oluşturup sonsuz döngü içerisine alıp read ile okuma işlemi gerçekleşti  
 ret,frame=cap.read()  
  
 yuz\_lokasyonu=[] ##Boş bir liste tanımlandı.  
 faces=detector(frame)  
 for face in faces:  
 x=face.left()  
 y=face.top()  
 w=face.right()  
 h=face.bottom()  
 yuz\_lokasyonu.append((y,w,h,x)) #Boş listenin içerisine yüz koordinatlarını sürekli ekleme işlemi yapıldı  
 face\_kodlama1=face\_recognition.face\_encodings(frame,yuz\_lokasyonu)  
  
 i=0

for face in face\_kodlama1: ##üzerinde gezinebilmek için bir for döngüsü daha açıldı.  
 y, w, h, x=yuz\_lokasyonu[i] ## Eğerki bu yüz lokasyonundaki i. Listedeki koordinatlara uyuyorsa  
  
  
 sonuc=face\_recognition.compare\_faces([mert\_kodlama],face)  
  
 if sonuc[0]==True: ## Evet doğru yakalandı bu kişi [Ad]  
 cv2.rectangle(frame,(x,y),(w,h),(255,0,0),2) ##Çerçeve rengi ve kalınlığı  
 cv2.putText(frame,"Mert",(x,h+17),cv2.FONT\_HERSHEY\_PLAIN,2,(0,0,255),4) ##Yazı stili rengi ve kalınlığı ##Belirtilen koordinattaki yere [Ad] yazdır  
 ekle("Mert",datetime.now()) ##Veritabanına [Ad]’ın görüldüğü saati ve tarihi eklenildi.

else: ##Eğer tanınan kişi [Ad] değilse kamera karşısında [Ad] yok ise  
 cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 2)  
 cv2.putText(frame, "Tanimlanamadi", (x, h + 17), cv2.FONT\_HERSHEY\_PLAIN, 2, (0, 0, 255), 4) ##Tanımlanamadı yazdırılması  
 ekle("Tanimlanamadi", datetime.now()) ##Veritabanınada tanımlanamadı yazdırılması işlemini ekle  
  
 cv2.imshow("Pencere",frame) ##Anlık olarak görülen frameleri oku ve pencerede açma işlemi.  
 if cv2.waitKey(10) & 0xFF==ord("q"): ##Klavyeden q tuşuna basıldığında programı sonlandır  
 break ##Ve döngüyü sonlandırma işlemi  
  
cap.release()  
cv2.destroyAllWindows() ##Açık olan pencere varsa hepsini kapat kodlarda herhangi bir hata almamak için cv2.destroyAllWindows() işlemi yapıldı.  
  
con.close() ##Veritabanında herhangi bir hata almamak için kapatma işlemi yapıldı.

BÖLÜM 5. GERÇEKCİ KISITLAR VE KOŞULLAR ALTINDA DEĞERLENDİRME

Bu çalışmada çözümüne ulaşılan problem ile çevre sorunları , güvenlik sistemleri , sosyal ve politik sorunlar , çevre sorunları gibi bir çok yere uyarlanılabilir ve sürdürülebilir olduğunu farkettik. Amacımız burada zamandan tasarruf sağlamaktı ve başardık.100 kişilik bir sınıfta sınıf yoklamasının 5 dakika olduğunu varsayalım bu gibi bir zaman kaybının önüne geçmiş olduk veya fabrikalardaki işçilerin kartlarını güvenlik görevlisine gösterirken harcadıkları zamanın önüne geçmiş olduk bir çok yerde uygulanabilir olacağını anlamış olduk.

Örneğin güvenlik sistemleri anlık görüntü veren mobese kameralara şüpheli olan kişilerin fotoğrafları tanıtılarak son görüldüğü yer saat tarih gibi bilgileri veritabanımızda depolayabiliriz.

BÖLÜM 6. SONUÇ

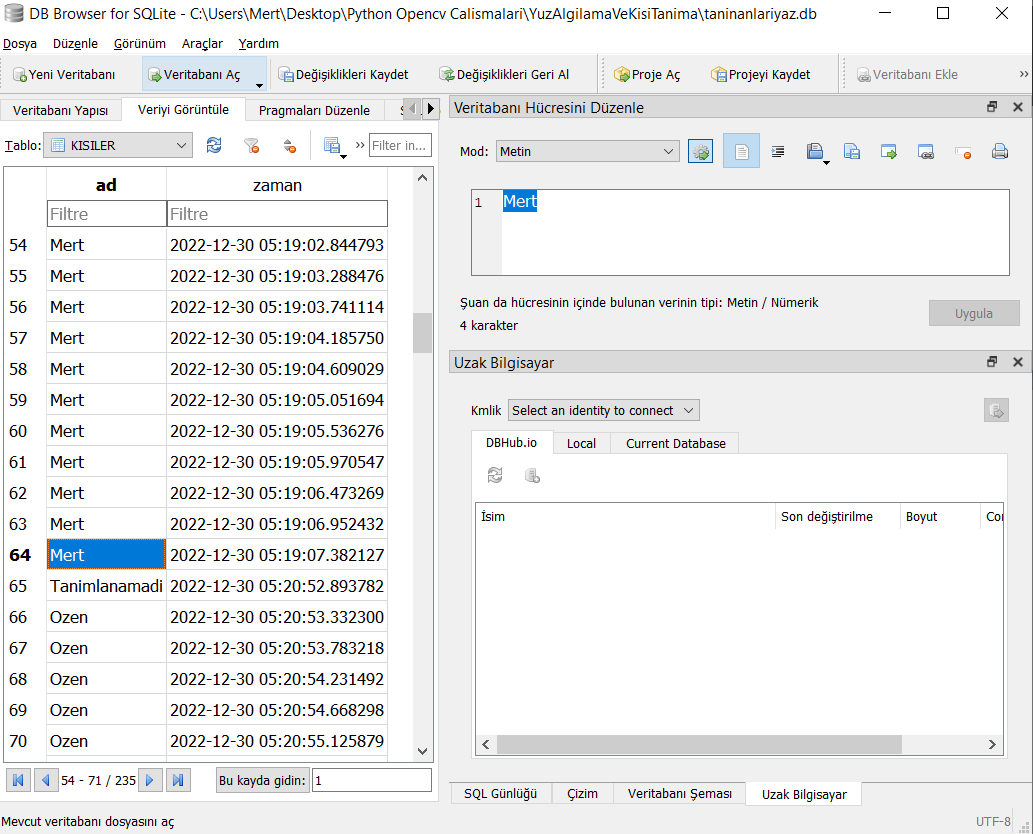
Modele tanıttığımız bir yüzü kodlar çalıştığında yukarıdaki görüntü ile karşılaşılacaktır.

Model tanımadığı bir yüz ile karşılaştığında ise aşağıdaki görüntü ile karşılaşılacaktır.



**Tablo 1** Veritabanı Çıktıları:

|  |  |
| --- | --- |
| Mert | 2022-12-30 05:19:06.952432 |
| Mert | 2022-12-30 05:19:07.382127 |
| Tanimlanamadi | 2022-12-30 05:20:52.893782 |
| Ozen | 2022-12-30 05:20:53.332300 |



Tanımladığı bir yüzü anlık olarak algılayıp veritabanına kayıt etme işlemi gerçekleştirilmiştir.

KAYNAKLAR

HAYIT, T., Erbay, H., & HAYIT, F. (2018). Gıda sektöründe uygulanan görüntü işleme çalışmalarının ve tekniklerinin incelenmesi. *Electronic Letters on Science and Engineering*, *14*(1), 12-22.

Osmanoğlu, U. Ö., Mutlu, F., Gürsoy, H., & ŞANLISOY, S. (2016). Görüntü işleme ve analizinin tıpta kullanımı ve bir uygulama. *Osmangazi Tıp Dergisi*, *41*(1), 6-16.

Tüfekçi, M., & Karpat, F. (2019). Derin Öğrenme Mimarilerinden Konvolüsyonel Sinir Ağları (CNN) Üzerinde Görüntü İşleme-Sınıflandırma Kabiliyetininin Arttırılmasına Yönelik Yapılan Çalışmaların İncelenmesi. In *International Conference on Human-Computer Interaction, Optimization and Robotic Applications* (pp. 28-31).

Erişti, E. (2010). Görüntü Đşlemede Yeni Bir Soluk, OPENCV.

PIŞKIN, MESUT. Opencv ile görüntü işleme. 2016.

Kazdal, S. (2013). *Beyin tümörlerinin ileri görüntü işleme ve örüntü tanıma teknikleri kullanılarak bilgisayar destekli tespiti* (Doctoral dissertation, Marmara Universitesi (Turkey)).

Eldem, A., Eldem, H., & Palali, A. (2017). Görüntü işleme teknikleriyle yüz algılama sistemi geliştirme. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *6*(2), 44-48.

Bingöl, K., Aslı, E. R., Örmecioğlu, H. T., & Arzu, E. R. (2020). Depreme dayanıklı mimari tasarımda yapay zeka uygulamaları: Derin öğrenme ve görüntü işleme yöntemi ile düzensiz taşıyıcı sistem tespiti. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, *35*(4), 2197-2210.

DEMİRBAŞ, H. Y., & DURSUN, İ. (2007). Buğday tanelerinin bazı fiziksel özelliklerinin görüntü işleme tekniğiyle belirlenmesi. *Journal of Agricultural Sciences*, *13*(03), 176-185.

Çevik, K. K., & Abdülkadir, Ç. A. K. I. (2010). Görüntü İşleme Yöntemleriyle Araç Plakalarının Tanınarak Kapı Kontrolünün Gerçekleştirilmesi. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, *10*(1), 31-38.