

Pardus için Yüz Tanıma Uygulaması

Yüz Tanıma Sistemleri Nedir?

Yüz tanıma sistemleri, kişinin yüz verileri ile birlikte doğrulama işlemi yapılmasıdır. Bir insan yüzüne ait özelliklerin belirlenmesi ve doğrulama yapılmak istenen yüz verisinin özellikleri ile karşılaştırılması üzerine sistem güvenliği sağlanmaktadır. Günümüzde yüz tanıma sistemleri üç adımda çalışmaktadır. Bu adımlardan ilk adım olan algılama, bir görüntüde yer alan bir veya birden fazla yüzün tespit edilmesidir. Bilgisayarlı görü teknoloji ile görüntüde yer alan yüz veya yüzler tespit edilmektedir. İkinci adım olan analiz, görüntüden tespit edilen yüz veya yüzlerin özelliklerinin çıkarılmasıdır. Bir insan yüzüne ait özellikler, insan yüzüne ait organların(ağız, burun, gözler vb.) yapıların şekil yapısı ya da birbirleri arasındaki uzaklıklar olmaktadır. Üçüncü ve son adım olan tanıma, sisteme kayıtlı olan bir görüntü ile kıyaslamak için elde edilen görüntü üzerindeki yüzlerin özelliklerinin karşılaştırılmasıdır. . Başka bir yöntem olarak ise daha önceden sisteme kayıtlı bir yüze ait eğitilmiş bir yapay zeka modeli üzerinden karşılaştırma işlemi yapılmasıdır.

Uygulama Açıklaması

Pardus için Yüz Tanıma uygulaması, günümüzde kullanımı artan yapay zeka teknolojilerinden olan yüz tanıma ve yüz doğrulama sistemleri esas alınarak geliştirilmiştir. Yüz tanıma ve yüz doğrulama sistemleri ile biyometrik olarak yüz verilerinin güvenlik amacıyla bilgisayarlı görü teknolojisi temel alarak günümüzde geliştirilmiştir. Pardus için Yüz Tanıma uygulaması, kişilerin biyometrik yüz verileri elde edilerek, her bilgisayar açılışında bilgisayar dahili veya harici kamerasından elde edilen görüntüdeki insan yüz verileri karşılaştırılıp güvenli bir giriş işlemi gerçekleştirilmektedir.

Pardus işletim sistemi için uluslararası kabul edilmiş olan yapay zeka yazılımlarının ve yüz tanıma yapay zeka yöntemlerinin yerli ve milli imkanlar ile kullanılabilir bir masaüstü uygulama yapılmıştır.

Kullanılan yazılımlar arasında yapay zeka alanından yoğun bir şekilde kullanılan Python programlama dili ile hem grafik arayüz ve yüz doğrulama yöntemi programlanmıştır. Grafik arayüz için PyQt5 kütüphanesi kullanılmıştır. Google tarafından geliştirilen TensorFlow ve TensorFlow'u esas alan Keras kütüphaneleri bu uygulamada kullanılmıştır. Ayrıca açık kaynak olarak yayınlanan RetinaFace ve ArcFace tabanlı DeepFace yüz tanıma teknolojisi kullanılmıştır. Yüz tanıma sistemleri açısından başarılı sonuçlar elde eden ve hakkında başarılı olduğuna dair bilimsel makale yazılan ArcFace yöntemi Pardus için Yüz Tanıma uygulamasında kullanılmıştır.

Uygulama geliştirilmesinde kullanılan bağımlılıklar ise Pardus 21 depoları ve Python'da kütüphane kurulumunda kullanılan pip install özelliği kullanılmıştır.

Kullanıcının giriş hareketlerinin takip edilmesi amacı ile giriş yapılan zaman bilgisi için SQLite3 tabanlı veritabanı dosyası geliştirilmiştir. Eğer hatalı giriş yapılmış ise giriş sırasında elde edilen kamera görüntüsünün depolanması(dosya ismi zaman bilgisi olarak ayarlanmıştır.) ve veritabanına kaydedilmesi özelliği aktiftir.

Uygulamanın konsol ekranı ve grafik arayüz ile çalışan kısımları birbiri ile uyumludur ve kullanıcının bilgisayarı üzerindeki işlemleri dikkate alınarak kullanılmaktadır.

Geliştirilme Amacı

Pardus için Yüz Tanıma Uygulaması, Pardus kullanıcılarının kişisel bilgisayarlarının güvenliği biyometrik verileri doğrulama yaparak açmak amacıyla geliştirilmiştir. Pardus işletim sistemi için yerli ve milli imkanlar ile uluslararası kabul yüz tanıma yöntemlerinin birleştirilmesi, Pardus işletim sistemine uygun yapıda yazılım geliştirilmesi hedeflenmiştir, Pardus kullanıcılarının kullanımı kolay olması için hem konsol ekranı hem de grafik arayüzü ile çalışabilecek bir düzeyde hazırlanmasının amacı ise uygulamaya kullanıcı tarafından erişilmesinin kolay olması hedeflenmiştir. İçerisinde yer alan ArcFace yapay zeka modeli ile birlikte yüksek düzeyde doğruluk seviyesi hedeflenmiştir. Ayrıca yapay zeka çalışma ortamının hem CPU hem de GPU desteği ile birlikte kullanıcıların kullanımına uygun olması hedeflenmiştir.

Ayrıca bu geliştirilen yazılım ile birlikte Pardus giriş ekranı, konsol ekranları veya ileri düzeyde kullanılan uygulamaların kullanımı sırasında güvenlik açığını en aza indirmek hedeflenmiştir. Böylece bilgisayarda yer alan kullanıcı şifresine ek olarak bilgisayarlı görüş teknolojisine sahip yüz tanıma sistemi geliştirilmiştir. Bu uygulamanın bir diğer amacı ise Pardus işletim sistemini üzerinde çalışacak ve açık kaynak topluluğu tarafından geliştirilecek yapay zeka uygulamalarına katkıda bulunmaktadır.

Kullanılan Paket Bağımlılıkları ve Kütüphaneler

Pardus için Yüz Tanıma uygulaması; yapay zeka, bilgisayarlı görüş, grafik arayüz, veri tabanı ve işletim sistemi kontrolü için bağımlı olduğu paketler ve Python kütüphaneleri aşağıdaki tabloda açıklanmıştır.

Bu paketler Linux işletim sistemleri için kaynak sitelerinden kurulabileceği gibi Pardus için Yüz Tanıma uygulamasının daha sağlıklı çalışması için Pardus işletim sisteminin resmi yazılım yükleme uygulaması Pardus 21 deposundaki paketler kullanılmıştır. Bazı paketler ise Python için kütüphane kurma aracı olan pip install özelliği kullanılmıştır.

Paket/Kütüphane Adı	Açıklama	Bulunduğu Kaynak
python3-pip	Gerekli olan Python kütüphanelerinin PyPi ile yüklenmesi için kullanılmıştır.	Pardus 21 işletim sistemi
python3-opencv	Bilgisayar ile görüş işlemleri için kullanılmıştır.	Pardus 21 deposu
python3-pyqt5	Grafik arayüz ile ilgili işlemlerin gerçekleştirilmesi için kullanılmıştır.	Pardus 21 deposu
TensorFlow	Derin öğrenme(yapay zeka) işlemleri için kullanılmıştır.	PyPi
Deepface	Yüz verilerinin elde edilmesi ve yüz doğrulama işlemi için	PyPi

	kullanılmıştır. ResNet50 ve ArcFace uyumludur.	
SQLite3 (Python kütüphanesi)	Kullanımı kolay olması ile veritabanı işlemleri için kullanılmıştır.	Pardus 21 işletim sistemi
OS (Python kütüphanesi)	Sistem üzerinde konsol işlemlerini gerçekleştirmek için kullanılmıştır.	Pardus 21 işletim sistemi
Time (Python kütüphanesi)	Veritabanı işlemlerinde zaman bilgisini eklemek için kullanılmıştır.	Pardus 21 işletim sistemi

python3-pip paketi ile TensorFlow ve DeepFace isimli derin öğrenme ve yüz tanıma uygulaması için Python kütüphaneleri kurulmuştur. Bu kütüphaneler elde edilen kamera görüntüsündeki insan yüzünü ve özelliklerini tespit etmek için derin öğrenme mantığı ile karşılaştırma yöntemi kullanmıştır. python3-opencv, bilinen adı ile OpenCV kütüphanesi, ile bilgisayar ile görü işlemleri kullanılmıştır. Bilgisayardan kamera görüntüsü alma ve DeepFace için görüntüyü kaydetmek için kullanılmıştır. python3-pyqt5, bilinen adı PyQt5, ile grafik arayüz oluşturulmuştur. Grafik arayüz işlemleri ile birlikte kullanıcıyı kolay bir şekilde yapacağı işlemleri gerçekleştirmesi amaçlanmıştır. SQLite3 ile birlikte kullanıcının yaptığı doğru girişler ve giriş zamanı, başarısız giriş denemeler ve giriş zamanları ile hatalı giriş görüntüsünün dosya yolu veritabanı işlemleri kullanılmıştır. Veritabanı içinde yer alan tabloya ilgili işlemler için veri girişi yapılmıştır. OS kütüphanesi ile Python'da konsol işlemleri kolayca yapılmaktadır. Yüz tanıma için yüz verisi kaydı yapmak amacıyla klasör oluşturulması, giriş denemeleri için klasör oluşturulması, hatalı girişler için klasör oluşturulması ve veritabanı oluşturulması işlemleri gerçekleştirilmiştir. Ayrıca hatalı giriş denemesi görselinin giriş denemesi klasöründen ilgili hatalı giriş klasörüne taşınması ve hatalı giriş zamanının bu görsele isim olarak atanması için kullanılmıştır. Ayrıca giriş zamanları ile ilgili bilgi almak için Time kütüphanesi kullanılmıştır.

Yüz Doğrulama için Deepface

GitHub üzerinde açık kaynak kodlu olarak paylaşılan bir projedir. VGG-Face, Google FaceNet, OpenFace, ArcFace, Facebook DeepFace ve Dlib ile çalışabilen bir projedir. İki insan yüzü arasındaki farkları bulmak amacıyla geliştirilmektedir. İlk önce görseller üzerinden yüz tespiti yapılır ve bu yüzler arasında karşılaştırma yaparak doğrulama işlemi yapılır. Varsayılan olarak yüz tespiti için OpenCV kullanılır ve tespit edilen yüzleri karşılaştırmak için VGG-Face kullanılır.

Pardus için Yüz Tanıma uygulaması yüz tespiti için RetinaFace yöntemini kullanmaktadır. Tespit edilen yüzler ise ArcFace yardımıyla karşılaştırılmaktadır.

Deepface için RetinaNet – Yüz Tespiti

Yüz tespiti işlemi ile bir kamera görüntüsü veya bir önceden kaydedilmiş görüntü üzerindeki yüzlerin koordinatlarının tespit edilmesidir. RetinaFace ise açık kaynak kodlu olarak GitHub üzerinden paylaşılan ve yüz tespiti için kullanılan bir projedir.

Deepface için ArcFace Fonksiyonu – Yüz Verileri Karşılaştırma

ArcFace mimarisi, iki yüz görüntüsünü girdi olarak alıp karşılaştırma yapan bir makine öğrenimi modelidir. Yüz tanıma ve yüz doğrulama işlemleri için kullanılmaktadır. İki yüz arasındaki mesafe farklarını bulmak için softmax yöntemi yerine kullanılmak amacıyla geliştirilmiştir.

Çıkarım sürecinde iki yüzün özellikleri normalleştirilir ve her iki yüzün aynı kişiye ait olup olmadığını belirlemek için gerekli hesaplamalar yapılmaktadır.

Doğruluk açısından ise LFW Veri Kümesi üzerinde en başarılı kayıp yöntemidir.

Method	#Image	LFW	YTF
DeepID [32]	0.2M	99.47	93.20
Deep Face [33]	4.4M	97.35	91.4
VGG Face [24]	2.6M	98.95	97.30
FaceNet [29]	200M	99.63	95.10
Baidu [16]	1.3M	99.13	-
Center Loss [38]	0.7M	99.28	94.9
Range Loss [46]	5M	99.52	93.70
Marginal Loss [9]	3.8M	99.48	95.98
SphereFace [18]	0.5M	99.42	95.0
SphereFace+ [17]	0.5M	99.47	-
CosFace [37]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	99.83	98.02

Table 4. Verification performance (%) of different methods on LFW and YTF.

Program Algoritması ve Dizin Yapısı





Pardus için Yüz Tanıma uygulamasının algoritması temel olarak kullanıcının yüz verisinin elde edilmesi, her giriş denemesinin yüz verisinin alınması ve karşılaştırma yaptıktan sonra sistem üzerinde kullanıcıya yetki vermesi esas alınarak tasarlanmıştır.

Aşağıdaki görselde programın algoritması basit düzeyde belirtilmiştir.



Program kurulumu için Debian paketi ve bash dosyası oluşturulmuştur. Ayrıca kullanıcı program kurulumundan sonra oluşturulmuş veritabanı ve kaydedilmiş kamera görüntülerini kolayca silebilmesi için ayrı bir bash dosyası oluşturulmuştur.

Kurulum ile ilgili dosyaların görevleri aşağıdaki tabloda belirtilmiştir.

			
Kurulum.txt	pardus-yuz-tanima.deb	pardus-yuz-tanima-kurulum	pardus-yuz-tanima-silme

Dosya Adı	Görev
Kurulum.txt	Kurulum ile ilgili açıklama ve bilgileri yer aldığı dosyadır. Kullanıcıyı bilgilendirmek amaçlı hazırlanmıştır.
pardus-yuz-tanima.deb	Yüz doğrulama işlemlerinin yapıldığı uygulamadır. Bu Debian paketi kurulduğunda sisteme pardus-yuz-tanima ve pardus-yuz-tanima-giris olmak üzere iki adet özellik eklenmiş olacaktır.
pardus-yuz-tanima-kurulum	pardus-yuz-tanima.deb paketinin kurulması ve ilgili diğer bağımlılıkların ve Python kütüphanelerinin kurulması için oluşturulmuştur.
pardus-yuz-tanima-silme	Bu dosyanın amacı ise sadece oluşturulan veritabanını ve sistemde kayıtlı olan bilgisayar sahibinin yüz verisini, hatalı girişler ile ilgili görüntülerin ve başarılı olmuş en son giriş ile ilgili kamera görüntüsünü silmektir. Pardus için Yüz Tanıma Uygulaması'nı sistem üzerinden kaldırmak amaçlı değildir.

Yukarıda belirtilmiş olan dosyalar arasında yer alan ve sisteme kurulacak olan pardus-yuz-tanima.deb paketinin dizin ve dosya yapısı aşağıdaki görselde belirtilmiştir. Kırmızı olarak belirtilmiş olan kısımlar klasörleri belirtmektedir. Siyah olarak belirtilmiş olan kısımlar ise klasörler ile birlikte aynı dizinde bulunan veya klasörlerin içinde bulunan dosyaları belirtmektedir.

```

DEBIAN      usr
  control   local
            bin
            pardus-yuz-tanima
            pardus-yuz-tanima-giris
            icons
            pardus_yuz.svg
            pencere_icon.png
            python
            PardusYuzTanima
            kurulum.py
            Kurulum-1-Bilgi
            kurulum_bilgi.py
            Kurulum-2-Kamera
            kamera.py
            kurulum_kamera.py
            Kurulum-3-Son
            kurulum_son.py
            share
            applications
            pardus-yuz-tanima.desktop
            Giris-1-Kamera
            kamera.py
            Giris-2-Karsilastirma
            giris_karsilastir.py
            Giris-3-Sonuc
            giris_basarili.py
            giris_basarisiz.py

```

Aşağıdaki tabloda ise yukarıda belirtilmiş olan Python dosyaları, bash dosyaları ve diğer dosyaların görevleri açıklanmıştır.

Dosya Adı	Görevi
control	Uygulama ile ilgili temel bilgileri içermektedir.
pardus-yuz-tanima	Bash dosyasıdır. Kullanıcının yüz verisinin kayıt edilmesi amaçlı kurulum işlemlerini başlatır, kurulum.py isimli Python dosyasını başlatır.
pardus-yuz-tanima-giris	Bash dosyasıdır. Kullanıcının bilgisayarı açıldığı zaman sistemde kayıtlı olan yüz verisi ile kameradan görüntü ile karşılaştırma yapmak için gerekli adımları başlatır. Giris-1-Kamera klasörü içindeki kamera.py isimli Python dosyasını çalıştırır.
pardus-yuz-tanima.desktop	Pardus için Yüz Tanıma Uygulaması'nın masaüstü uygulamaları menüsünde yer alacak simgesidir.
pardus_yuz.svg	Pardus için Yüz Tanıma Uygulaması'nın masaüstü uygulamaları menüsünde yer alacak simgesinin vektörel kaynak görüntüsüdür.
pencere_icon.png	PyQt5 ile oluşturulan Python dosyalarında kullanılmıştır ve bu dosyaların pencere simgesidir.
kurulum.py	Pardus-yuz-tanima tarafından çalıştırılır ve sahip olduğu buton ile kullanıcıyı bilgilendirme penceresi olan kurulum_bilgi.py dosyasına yönlendirir.
kurulum_bilgi.py	Kullanıcıyı kendisine ait yüz görüntüsü alımı hakkında bilgilendirir ve önemli bilgilendirmelerde bulunur. Kamera ile işlem yapmak için sahip olduğu buton ile kurulum_kamera.py dosyasına yönlendirir.
kurulum_kamera.py	Kamera ile yüz görüntüsü alımından önce son defa bilgilendirme amacıyladır. Bunun nedeni OpenCV ile kamera

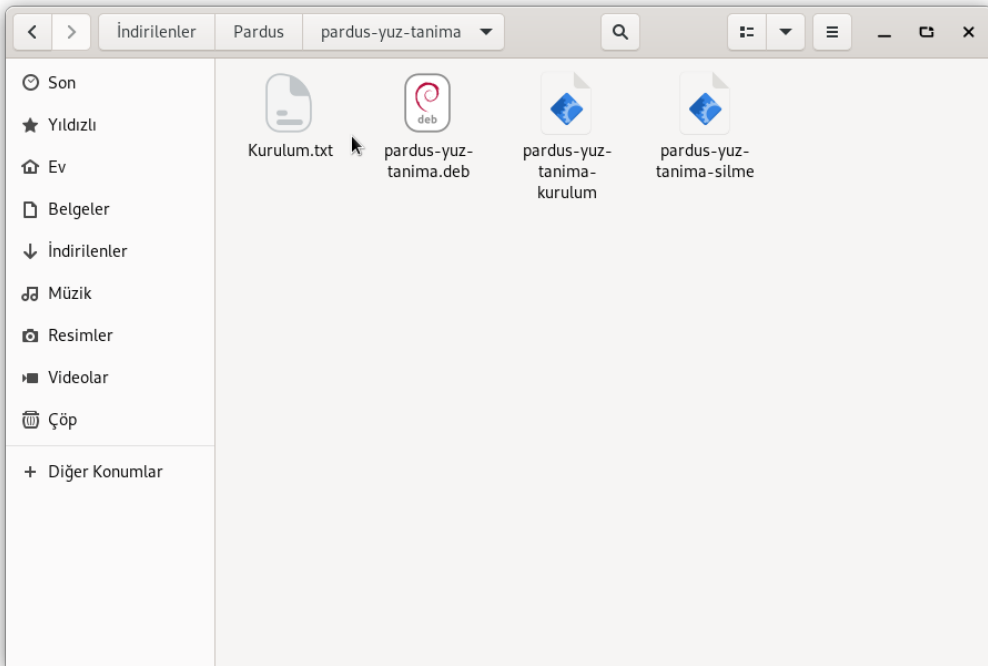
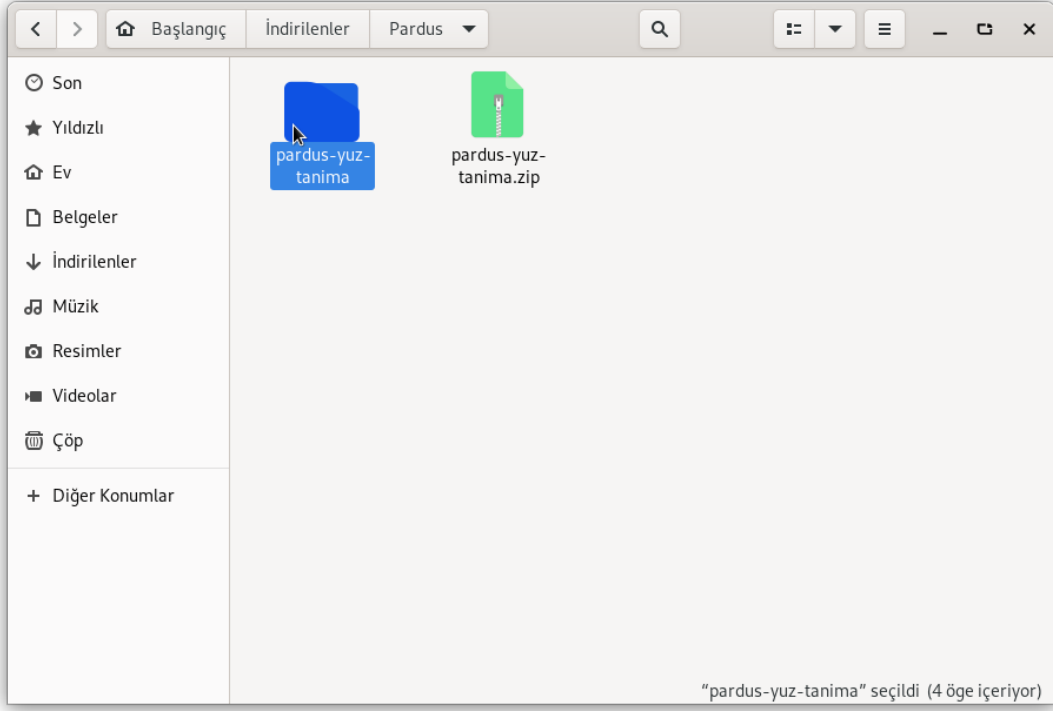
	görüntüsü bu aşamdan sonra hemen alınacaktır ve kaydedilecektir. Kullanıcının hazır olması bildirilmektedir. Sahip olduğu buton ile Kurulum-2-Kamera klasörü içinde yer alan kamera.py çalıştırılır.
kamera.py	Kurulum-2-Kamera klasörü içinde yer alır. Kamera ile kullanıcıdan yüz görüntüsü alınır ve kaydedilir. En son olarak Kurulum-3-son klasörü içinde yer alan kurulum_son.py çalıştırılır.
kurulum_son.py	Kullanıcıdan kamera ile yüz görüntüsü alındığı bildirilir. Tüm pencereler bu adımdan sonra kapatılabilir.
kamera.py	Giris-1-Kamera klasörü içinde yer alır ve her yapılan bilgisayar açılışı sırasında kamera ile görüntü alır ve kaydeder. Bu adımdan sonra Giris-2-Karsilastirma klasörü içinde yer alan giris_karsilastir.py çalıştırılır.
giris_karsilastir.py	Bilgisayar sahibinin bilgisayarda kayıtlı olan görüntüsü ile giriş denemesi sırasında elde edilen görüntüyü karşılaştırma yapmaktadır. Bu adımda TensorFlow, DeepFace ve ArcFace kullanılmaktadır. Olumlu veya olumsuz sonuca göre Giris-3-Sonuc içinde yer alan giris_basarili.py veya giris_basarisiz.py çalıştırılır.
giris_basarili.py	Başarılı girişleri bildirir.
giris_basarisiz.py	Başarısız girişleri bildirir.

Kurulum.py Python dosyası ayrıca sistem ana dizinine gizli klasörler ve gizli SQLite3 veritabanı dosyası oluşturmaktadır. Sırası ile .kamera_kayit, .kamera_giris ve klasör içinde de .hatali_giris klasörü oluşturulmaktadır. .pardus_yuz_tanima.sqlite veritabanı dosyası ise giris_kaydi tablosu olacak şekilde oluşturulmaktadır. Bu tablo içinde zaman,giris_ayrintisi, hatali_dosya_adi olmak üzere sütunlar yer almaktadır. Bilgisayar kullanıcısına ait kamera görüntüsü .kamera_kayit içine kaydedilir. Bu görüntü karşılaştırma işlemleri için kullanılır. Kullanıcı her defa giriş yaptığında kamera görüntüsü .kamera_giris içine kaydedilir. giris_karsilastir.py ise sırası ile veritabanı dosyasına zaman,başarılı giriş ise True değeri veya başarısız deneme ise False değeri, başarılı ise dosya adı yerine boşluk karakteri veya

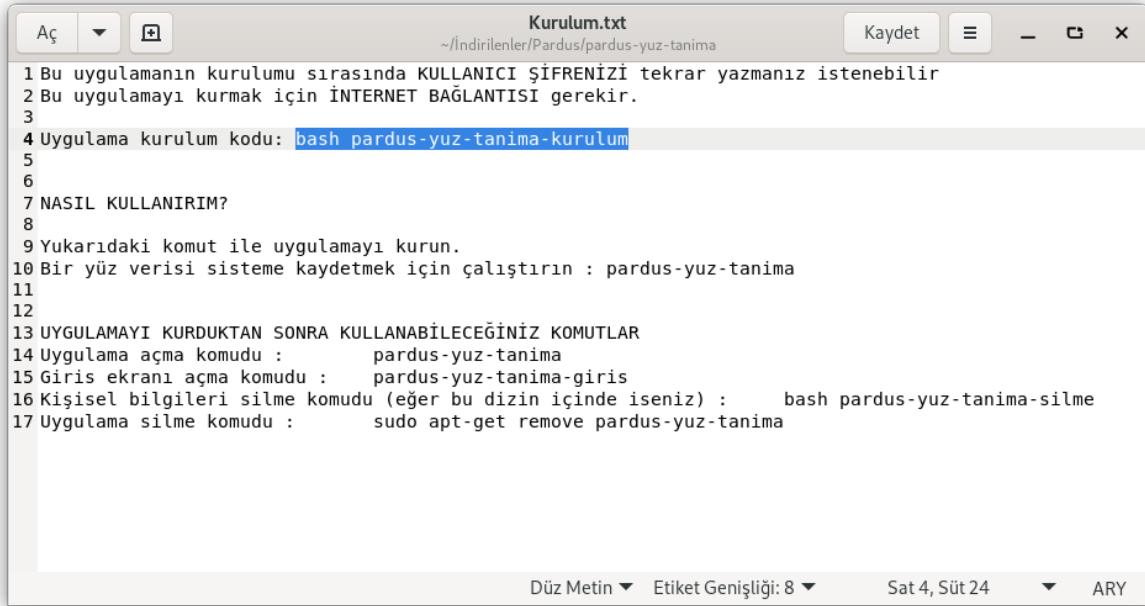
başarısız ise zaman sütununda yer alan ifade ile ismi değiştirilmiş dosya adı kaydedilir. Ayrıca başarısız denemey ait dosya yeniden adlandırılmış olunacaktır ve .hatali_giris içinde taşınır.

Program Kurulumu

Pardus için Yüz Tanıma Uygulaması için kurulum adımları aşağıda görsellerle açıklanmıştır. 1.adım olarak ilk önce arşivlenmiş olan dosya bulunulan dizine çıkartılır ve dışarı çıkartılan klasör açılır.



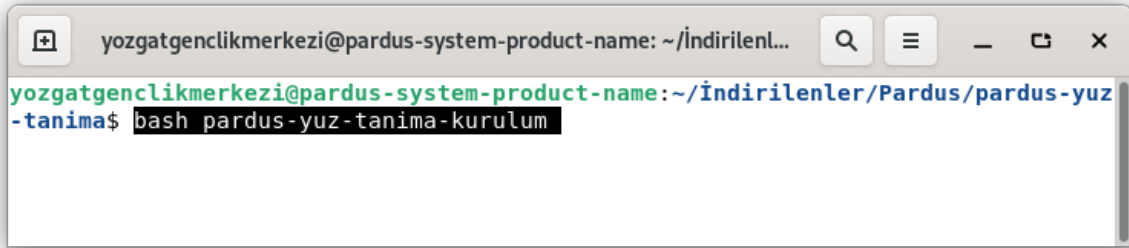
Kurulum.txt dosyası içinde yer alan program kurulum kodu çalıştırılır.



```
Kurulum.txt
~/İndirilenler/Pardus/pardus-yuz-tanima

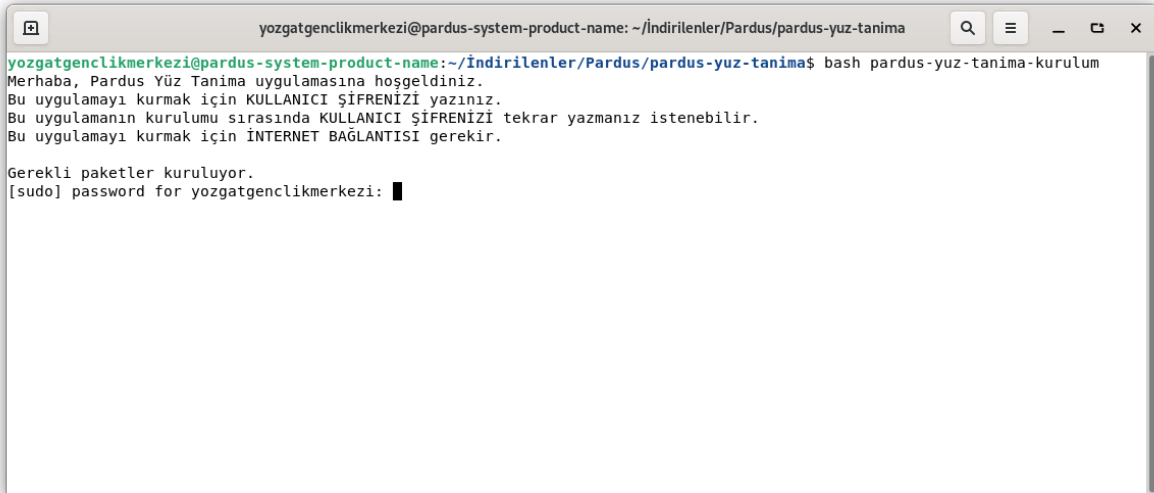
1 Bu uygulamanın kurulumu sırasında KULLANICI ŞİFRENİZİ tekrar yazmanız istenebilir
2 Bu uygulamayı kurmak için İNTERNET BAĞLANTISI gerekir.
3
4 Uygulama kurulum kodu: bash pardus-yuz-tanima-kurulum
5
6
7 NASIL KULLANIRIM?
8
9 Yukarıdaki komut ile uygulamayı kurun.
10 Bir yüz verisi sisteme kaydetmek için çalıştırın : pardus-yuz-tanima
11
12
13 UYGULAMAYI KURDUKTAN SONRA KULLANABİLECEĞİNİZ KOMUTLAR
14 Uygulama açma komutu :      pardus-yuz-tanima
15 Giriş ekranı açma komutu :  pardus-yuz-tanima-giris
16 Kişisel bilgileri silme komutu (eğer bu dizin içinde iseniz) :      bash pardus-yuz-tanima-silme
17 Uygulama silme komutu :      sudo apt-get remove pardus-yuz-tanima

Düz Metin Etiket Geniřlięi: 8 Sat 4, Süt 24 ARY
```



```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima$ bash pardus-yuz-tanima-kurulum
```

Bu aşamada gerekli olan paketler ve kütüphaneler kurulması için internet bağlantısı gerekir. Ayrıca eğer bu paketler sistemde kurulu ise veya internet bağlantısı sağlıklı değilse ilgili bilgilendirme yazıları ekrana gelecektir. Bu aşamada kullanıcıdan şifresi istenmektedir.



```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima$ bash pardus-yuz-tanima-kurulum
Merhaba, Pardus Yüz Tanıma uygulamasına hoşgeldiniz.
Bu uygulamayı kurmak için KULLANICI ŞİFRENİZİ yazınız.
Bu uygulamanın kurulumu sırasında KULLANICI ŞİFRENİZİ tekrar yazmanız istenebilir.
Bu uygulamayı kurmak için İNTERNET BAĞLANTISI gerekir.

Gerekli paketler kuruluyor.
[sudo] password for yozgatgenclikmerkezi:
```

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ bash pardus-yuz-tanima-kurulum
Merhaba, Pardus Yüz Tanıma uygulamasına hoşgeldiniz.
Bu uygulamayı kurmak için KULLANICI ŞİFRENİZİ yazınız.
Bu uygulamanın kurulumu sırasında KULLANICI ŞİFRENİZİ tekrar yazmanız istenebilir.
Bu uygulamayı kurmak için İNTERNET BAĞLANTISI gerekir.

Gerekli paketler kuruluyor.
[sudo] password for yozgatgenclikmerkezi:
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-pip zaten en yeni sürümde (20.3.4-4+deb11u1).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-pyqt5 zaten en yeni sürümde (5.15.2+dfsg-3).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-opencv zaten en yeni sürümde (4.5.1+dfsg-5).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.
```

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
Bu uygulamanın kurulumu sırasında KULLANICI ŞİFRENİZİ tekrar yazmanız istenebilir.
Bu uygulamayı kurmak için İNTERNET BAĞLANTISI gerekir.

Gerekli paketler kuruluyor.
[sudo] password for yozgatgenclikmerkezi:
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-pip zaten en yeni sürümde (20.3.4-4+deb11u1).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-pyqt5 zaten en yeni sürümde (5.15.2+dfsg-3).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.
Paket listeleri okunuyor... Bitti
Bağımlılık ağacı oluşturuluyor... Bitti
Durum bilgisi okunuyor... Bitti
python3-opencv zaten en yeni sürümde (4.5.1+dfsg-5).
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 0 paket yükseltilmeyecek.

Gerekli kütüphaneler kuruluyor.
```

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
Gerekli kütüphaneler kuruluyor.
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (23.0)
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError (SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (ssl.c:1123)))': /simple/pip/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError (SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (ssl.c:1123)))': /simple/pip/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError (SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (ssl.c:1123)))': /simple/pip/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError (SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (ssl.c:1123)))': /simple/pip/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError (SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (ssl.c:1123)))': /simple/pip/
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: tensorflow in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (2.11.0)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (2.11.0)
```

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
Requirement already satisfied: numpy>=1.20 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (1.23.4)
Requirement already satisfied: opt-einsum>=2.3.2 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: wrapt>=1.11.0 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: h5py>=2.9.0 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (3.7.0)
Requirement already satisfied: google-pasta>=0.1.1 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (0.27.0)
Requirement already satisfied: libclang>=13.0.0 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (14.0.6)
Requirement already satisfied: tensorboard<2.12,>=2.11 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (2.11.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (2.11.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (1.50.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /home/yozgatgenclikmerkezi/.local/lib/python3.9/site-packages (from tensorflow) (4.4.0)
```

Collecting gdown>=3.10.1

Downloading gdown-4.6.2-py3-none-any.whl (14 kB)

Requirement already satisfied: numpy>=1.14.0 in /usr/lib/python3/dist-packages (from retina-face) (1.19.5)

Requirement already satisfied: Pillow>=5.2.0 in /usr/lib/python3/dist-packages (from retina-face) (8.1.2) Collecting retina-face

Downloading retina_face-0.0.13-py3-none-any.whl (16 kB)

Bu adımda Pardus 21 deposunda bulunan OpenCV ile DeepFace arasında uyumluluk sorunu bulunmamaktadır. Aşağıdaki uyarı ile yüz tanıma için bir sorun oluşmamaktadır.

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1123)'))': /simple/deepface/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1123)'))': /simple/deepface/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1123)'))': /simple/deepface/
Could not fetch URL https://pypi.org/simple/deepface/: There was a problem confirming the ssl certificate: HTTPSConnectionPool(host='pypi.org', port=443): Max retries exceeded with url: /simple/deepface/ (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1123)')))) - skipping
ERROR: Could not find a version that satisfies the requirement opencv-python>=4.5.5.64 (from deepface) (from versions: none)
ERROR: No matching distribution found for opencv-python>=4.5.5.64

Pardus Yüz Tanıma kuruluyor.
Daha önce seçili olmayan pardus-yuz-tanima paketi seçiliyor.
(Veritabanı okunuyor ... 221787 dosya veya dizin kurulu durumda.)
Paket açılacak: pardus-yuz-tanima.deb ...
Paket açılıyor: pardus-yuz-tanima (1.1) ...
Ayarlanıyor: pardus-yuz-tanima (1.1) ...
```

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
Pardus Yüz Tanıma kuruluyor.
Daha önce seçili olmayan pardus-yuz-tanima paketi seçiliyor.
(Veritabanı okunuyor ... 221787 dosya veya dizin kurulu durumda.)
Paket açılacak: pardus-yuz-tanima.deb ...
Paket açılıyor: pardus-yuz-tanima (1.1) ...
Ayarlanıyor: pardus-yuz-tanima (1.1) ...
Tetikleyiciler işleniyor: gnome-menus (3.36.0-1) ...
Tetikleyiciler işleniyor: desktop-file-utils (0.26-1) ...
Tetikleyiciler işleniyor: mailcap (3.69) ...

Uygulama açma komudu : pardus-yuz-tanima
Giris ekranı açma komudu : pardus-yuz-tanima-giris
Kişisel bilgileri silme komudu (eğer bu dizin içinde iseniz) : bash pardus-yuz-tanima-silme
Uygulama silme komudu : sudo apt-get remove pardus-yuz-tanima

yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$
```

Pardus için Yüz Tanıma Uygulaması artık başarılı bir şekilde kurulmuş durumdadır.

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ pardus-
pardus-about          pardus-mycomputer      pardus-software        pardus-yuz-tanima-giris
pardus-configure      pardus-package-installer  pardus-usb-formatter   pardus-welcome
pardus-image-writer    pardus-power-manager     pardus-yuz-tanima
pardus-java-installer  pardus-restart-shell     pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ pardus-
```

Kullanıcı Kayıt Aşaması

Uçbirim ekranı ile kullanıcı kayıt işlemi için pardus-yuz-tanima komutu yazılır. Karşımıza kurulum ekranı gelir.

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ pardus-yuz-tanima
```



Bir sonraki ekranda yer alan Nvidia GPU hizmetinin gerekli olmasının nedeni ise yüz doğrulama işlemi sırasında çok kısa süre harcanmasıdır. Eğer Nvidia GPU desteği yok ise yüz doğrulama işlemi uzun sürebilmektedir. Bunun nedeni ise TensorFlow'un Nvidia CUDA çekirdeklerine ihtiyaç duymasındır. Ayrıntı için :

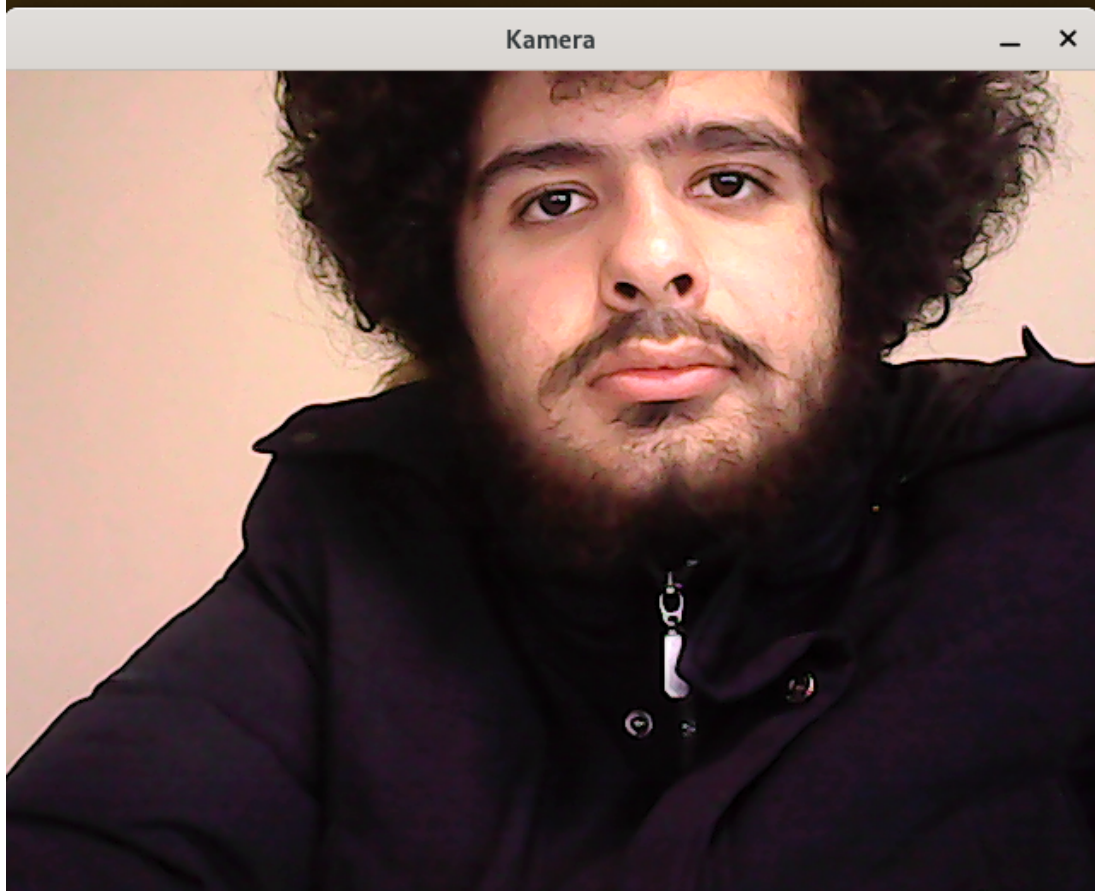
<https://www.tensorflow.org/install/pip?hl=tr>



Kamera ekranı gelmeden kullanıcıdan kameraya yüzü belirgin bir şekilde bakması istenir. Bunun nedeni kameranın açıldığında hemen yüz görüntüsünü alacak olmasıdır.



Kamera ekranı açılır ve kullanıcının yüz görüntüsü alınır.



Daha sonra aşağıdaki bilgilendirme ekranı gelir ve kullanıcı isterse tüm pencereleri kapatabilir.



Program Test Edilmesi

Aşağıdaki görseldeki gibi kullanıcı, Uçbirim ekranına `pardus-yuz-tanima-giris` komutunu yazıp programın sağlıklı bir şekilde çalışıp çalışmadığını test edebilir.

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ pardus-yuz-tanima-giris
```

Yukarıdaki komuttan sonra kamera görüntüsü açılır.



Bu adımdan sonra DeepFace açık kaynak yapay zeka yazılımı yüz görüntüleri üzerinde karşılaştırma işlemi yapacaktır.

```
yozgatgenclikmerkezi@pardus-system-product-name: ~/İndirilenler/Pardus/pardus-yuz-tanima
yozgatgenclikmerkezi@pardus-system-product-name:~/İndirilenler/Pardus/pardus-yuz-tanima$ pardus-yuz-tanima-giris
[ WARN:0] global ../modules/videoio/src/cap_gstreamer.cpp (961) open OpenCV | GStreamer warning: Cannot query video position: status=0, value=-1, duration=-1
/home/yozgatgenclikmerkezi
1/1 [=====] - 1s 1s/step
1/1 [=====] - 0s 28ms/step
True
```


Giriş başarılı ise aşağıdaki ekran karşımıza çıkacaktır.



Başarısız giriş denemesi yapılmış ise programın değerlendirme kriterleri aşağıdaki gibidir.

- Görüntüde yüz yok ise giriş başarısızdır.
- Görüntüdeki yüz net değil veya eğer çok uzakta ise giriş başarısızdır.
- Görüntüdeki yüz başka birine ait ise giriş başarısızdır.
- Görüntüdeki yüz kameraya doğrudan bakmıyorsa, başka bir yöne bakıyorsa giriş başarısızdır.
- Görüntüdeki yüzün sistem tarafından doğruluk oranı düşük ise giriş başarısızdır.

Yukarıdaki başarılı veya başarısız giriş denemeleri ayrıca program tarafından veritabanı dosyasına ilgili veriler işlenmektedir ve gerekli dosya kaydetme, isimlendirme ve taşıma işlemleri yapılmaktadır.

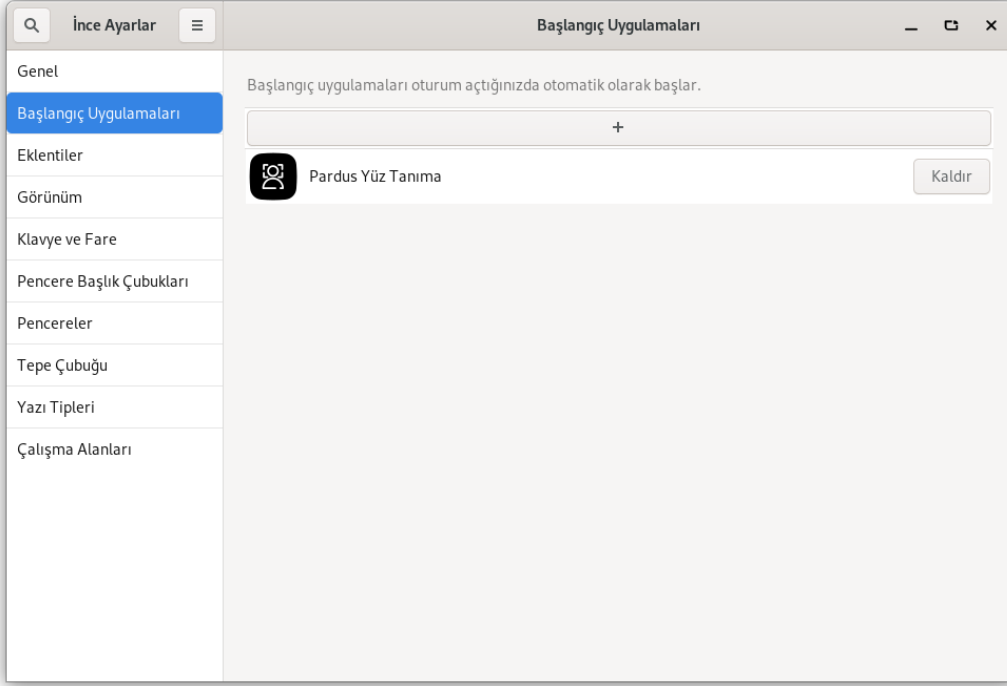
Başarısız giriş denemesi aşağıdaki açıklanmıştır. Aşağıdaki insan yüzü, kameraya yan hizada duracak şekildedir.



Başlangıç Uygulaması Olarak Ayarlama

GNOME ve XFCE sürümlerinin kendilerine ait başlangıç uygulaması olarak uygulama başlatma seçenekleri vardır. Böylece kullanıcı, bilgisayarını her açtığında veya oturumunu yeniden giriş yaptığında bu uygulama çalışmış olacaktır.

GNOME için İnce Ayarlar → Başlangıç Uygulamaları üzerinden gerekli ayarlamalar yapılabilmektedir :



XFCE için Ayar Yöneticisi → Oturum ve Başlangıç → Kendiliğinden Uygulama Başlatımı → + işareti ile ekleme menüsü açılır.

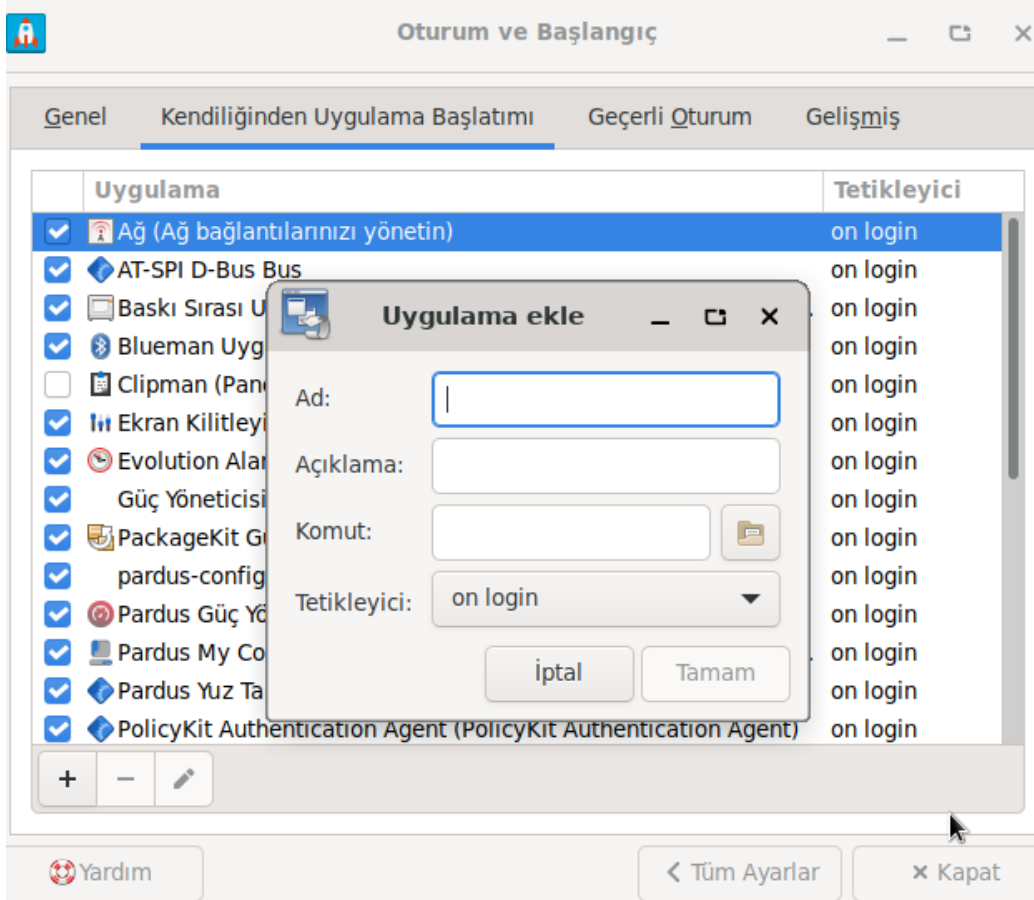
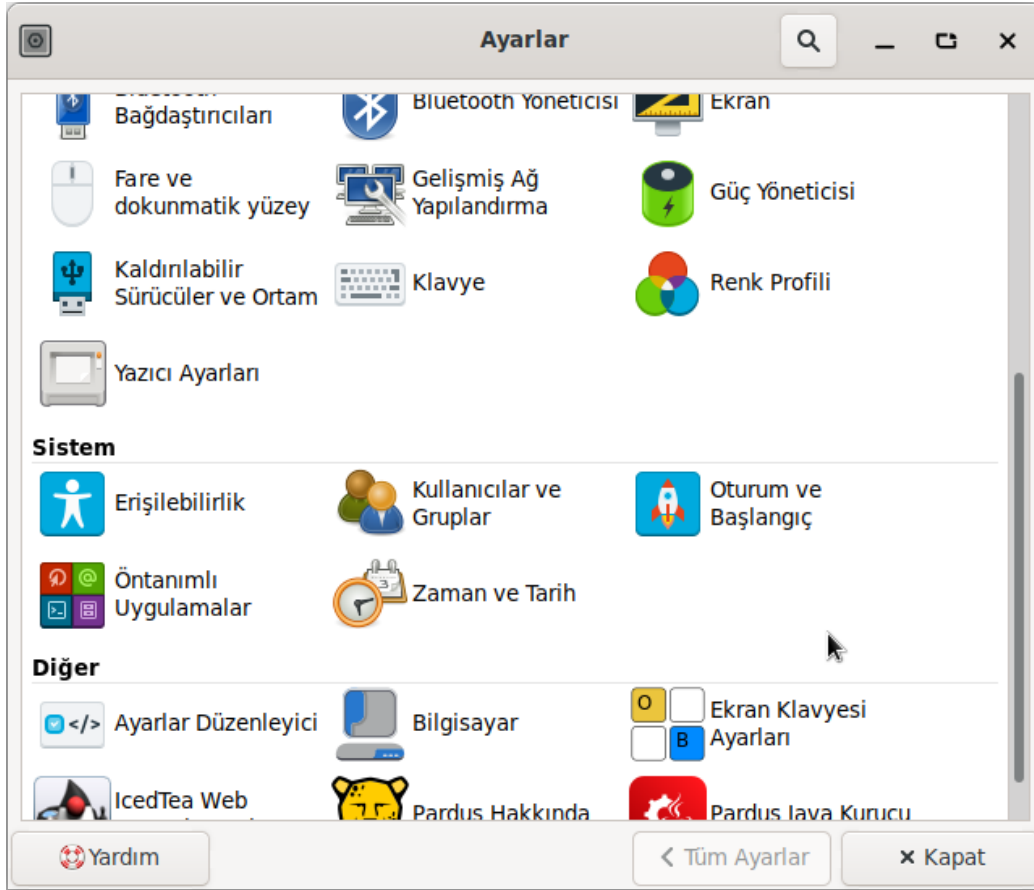
Bu ekranda tetikleyiciyi “on login” olarak belirlemek gerekmektedir. Bu işlem ile Pardus 21.4 XFCE sürümünde her oturum açılışında otomatik başlatma seçeneği aktif olacaktır.

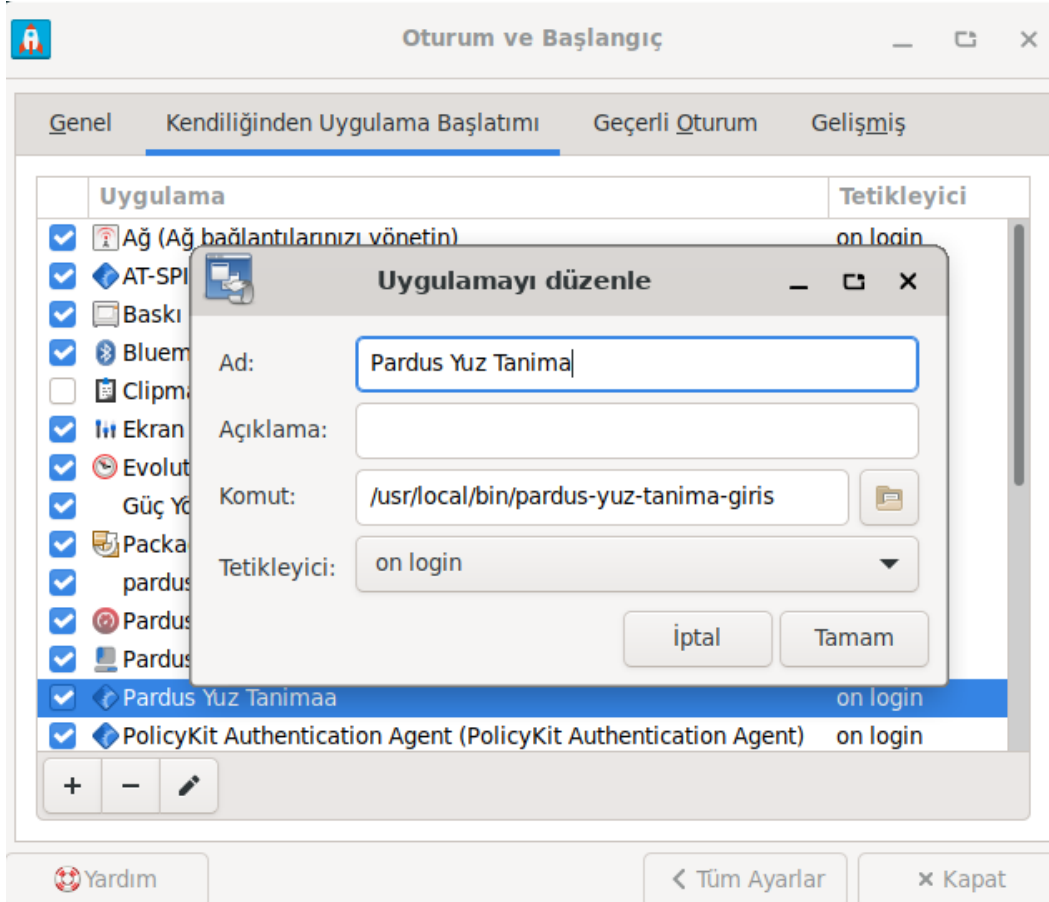
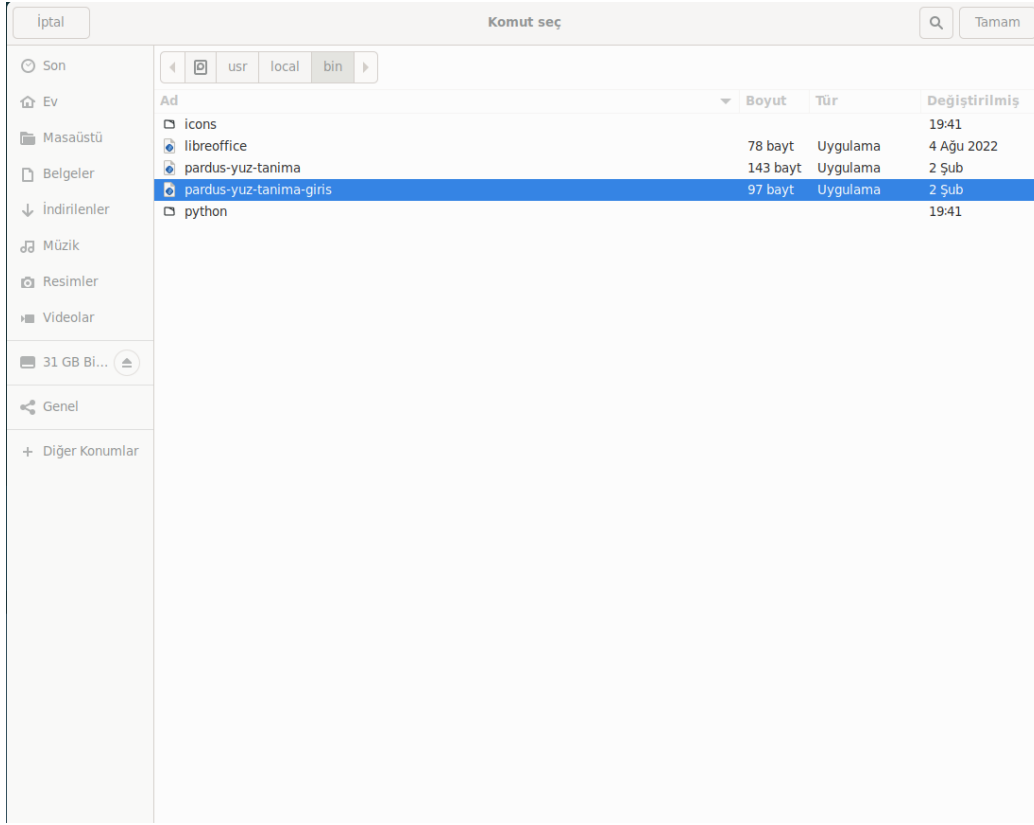
Ad kısmına ise “Pardus Yüz Tanıma” yazarak devam edilir. Açıklama kısmı ise isteğe bağlı olarak kullanıcı tarafından doldurulabilir.

Komut kısmı, otomatik olarak başlatılacak olan dosyayı çalıştırma imkanı vermektedir. Bu kısımda yer alan klasör simgesi ile aşağıdaki dizinde yer alan dosya seçilir. Ayrıca “Komut” kısmına doğrudan da yazılabilir.

Çalıştırılacak dizin:

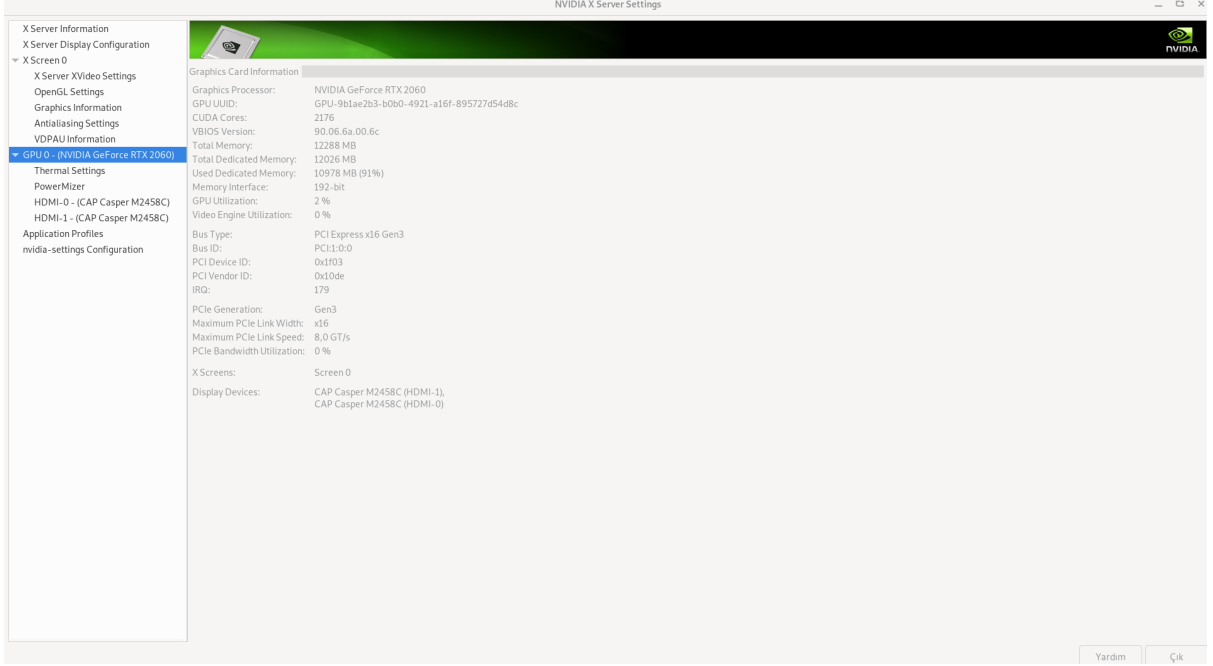
/usr/local/bin/pardus-yuz-tanima-giris





İstatistikler

Pardus için Yüz Tanıma Uygulamasının testleri Nvidia GPU üzerinden mevcut CPU'ya göre daha hızlı sonuçlar elde etmek amacıyla gerçekleştirilmiştir. Ayrıca TensorFlow tarafından tüm yapay zeka işlemleri otomatik olarak Nvidia GPU üzerinden gerçekleştirilmiştir. Aşağıdaki Nvidia X-Server uygulaması üzerinden kullanım istatistikleri belirtilmiştir. Bu istatistikler çeşitli Nvidia GPU'lara göre değişiklik göstermektedir. Bu durumun nedeni TensorFlow yapısından kaynaklanmaktadır. Ayrıca Nvidia GPU veya herhangi bir GPU modeli desteği olmayan masaüstü ve dizüstü bilgisayarlardaki işlemler CPU ve sistem RAM'i üzerinden gerçekleşmektedir.



Ayrıca uçbirim ekranı üzerinden nvidia-smi komutu yazıldıktan sonra aşağıdaki ekran ile karşılaşılır.

```
yozyatgenclikmerkezi@pardus-system-product-name: ~  
+-----+  
| NVIDIA-SMI 470.161.03   Driver Version: 470.161.03   CUDA Version: 11.4   |  
+-----+  
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |  
| Fan   Temp   Perf         Pwr:Usage/Cap|            Memory-Usage | GPU-Util  Compute M. |  
|=====  
| 0     NVIDIA GeForce ...   On         | 00000000:01:00.0 On   |           N/A        |  
| 0%    36C    P2         45W / 184W | 10977MiB / 12026MiB |      1%    Default   |  
|=====  
+-----+  
+-----+  
| Processes: |  
| GPU   GI    CI          PID    Type   Process name                      GPU Memory |  
|       ID    ID              |              Usage      |              |  
+-----+  
| 0     N/A    N/A        1848     G   /usr/lib/xorg/Xorg                 153MiB |  
| 0     N/A    N/A        1942     G   /usr/bin/gnome-shell               172MiB |  
| 0     N/A    N/A        2725     G   nvidia-settings                    2MiB |  
| 0     N/A    N/A        3292     G   ...830123874412951021,131072     38MiB |  
| 0     N/A    N/A        3850     C   python3                           10605MiB |  
+-----+  
yozyatgenclikmerkezi@pardus-system-product-name:~$
```

Video Linkleri

Kurulum vide linki:

<https://youtu.be/XQOjdCGAOCY>

Program silme video linki :

https://youtu.be/hgeTqY6O_Bs

Ayrıca bu belgenin bulunduğu klasör içinde videolar da yüklüdür.

Kaynakça

<https://suhedacilek.medium.com/resnet-residual-network-nedir-49105e642566>

<https://medium.com/yapi-kredi-teknoloji/kimsenin-size-s%C3%B6ylemeyece%C4%9Fi-y%C3%BCz-tan%C4%B1ma-i%C3%87pu%C3%A7lar%C4%B1-1f99ec6cca9f>

<https://datagen.tech/guides/computer-vision/resnet-50/>

<https://iq.opengenus.org/resnet50-architecture/>

<https://arxiv.org/abs/1801.07698>

<https://paperswithcode.com/method/arcface>

<https://medium.com/axinc-ai/arcface-a-machine-learning-model-for-face-recognition-5f743cdac6fa>

<https://github.com/serengil/deepface>

<https://sefiks.com/2021/04/27/deep-face-detection-with-retinaface-in-python/>

<https://github.com/serengil/retinaface>

<https://pypi.org/project/retina-face/>

<https://www.tensorflow.org/install/pip?hl=tr>