



**Düzce Üniversitesi**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

## **BM 399 Staj Raporu**

**Teslim Eden:**

**İsim-Soyisim : Meliha DEMİRCİ**

**Numara : 211001301**

**Kuruluşun İsmi ve Yeri : Arvis Teknoloji  
İstanbul, Türkiye**

**Staj Tarihleri : 07.08.2023- 15.09.2023**

**Ekim 2023**

## İÇİNDEKİLER

1. GİRİŞ.....	1
2. ŞİRKET HAKKINDA BİLGİ.....	1
3. YAPILACAK OLAN VEYA YAPILAN İŞ .....	2
3.1 Amaç .....	2
3.2 Tanım .....	2
3.3 Süreç ve İşleyiş .....	3
4. PROJE.....	3
4.1 Amaç .....	3
4.2 Tanım .....	3
4.3 Gereksinimler ve İhtiyaçlar.....	4
4.4 Analiz .....	4
4.5 Gelişim Süreci.....	6
4.6 Test ve Uygulama .....	7
4.6.1 Yüz Analizi.....	7
4.6.2 Hasar Tespiti .....	11
4.6.3 Mobil Uygulama.....	17
4.7 Proje Sonucu .....	21
5. SONUÇ.....	21
6. KAYNAKLAR.....	21
7. EKLER .....	21

## 1. GİRİŞ

Arvis Teknoloji şirketinde gerçekleştirdiğimiz altı haftalık staj, otonom araçlar için nesne tanıma ve davranışsal biyometri-insan davranışları analizi projesi üzerine yoğunlaşmıştır. Bu staj süresince, hem mevcut teknolojileri test etme hem de yeni çözümler geliştirme fırsatı bulduk. Bu projenin temel amacı, çeşitli araçların ve olası hasarların tespit ve tanıma yöntemlerini araştırmak ve öğrenmektir. Ayrıca, belirlediğimiz veri kümesi için en iyi çalışan tespit ve tanıma modellerini belirlemek, bu modülleri geliştirmek ve aynı zamanda davranışsal biyometri yöntemleri ve biyometrik verilerin temel prensipleri hakkında çalışmalar yapmak projenin hedeflerini oluşturmaktadır. Arvis Teknoloji, araştırma ve geliştirme alanında uzmanlaşmış bir şirket olduğundan, proje süresince ihtiyacımız olan GPU desteği firma tarafından sağlanmıştır. Bu destek, projenin başarıyla tamamlanmasına olanak tanımıştır. Proje başlangıcında, mevcut literatürü inceledik ve projenin çözümü için mümkün olan en iyi yaklaşımları araştırdık. Ayrıca, gerekli veri ve kaynaklarını topladık ve hazırladık. Şirket tarafından temin edilen veri kümesi, Roboflow veri etiketleme sitesi kullanılarak etiketlenmiştir. Etiketlenen veriler, farklı modellerle eğitilerek en uygun sonuçları elde etmek amacıyla kullanılmıştır. Proje kapsamında yüz tespiti için veriler toplandı ve farklı modeller kullanılarak sürücünün uyku veya uykusuz olma durumları başarıyla tespit edildi. Ayrıca mobil arayüz için gerekli tasarım çalışmaları yapılarak eğitilen model bu arayüze entegre edildi. Proje kapsamında yüz tespiti için veriler toplandı ve farklı modeller kullanılarak sürücünün uyku veya uykusuz olma durumları başarıyla tespit edildi. Ayrıca, mobil arayüz için gerekli tasarım çalışmaları yapılarak eğitilen model bu arayüze entegre edildi. Bu staj raporu projenin ayrıntılı açıklamalarını ve sonuçlarını içermektedir. Projemizin yöntemleri, uygulamaları ve elde edilen sonuçlar raporun ilerleyen bölümlerinde ayrıntılı olarak ele alınacaktır.

## 2. ŞİRKET HAKKINDA BİLGİ

Arvis Teknoloji İstanbul Teknopark'ta Cube Incubation binasında yer almaktadır. Görüntü işleme, büyük veri analizi, biyometri, yüz tanıma ile kimlik doğrulama, uzaktan müşteri edinimi, anomali, hata, hasar tespiti konularında yapay zeka teknikleri çalışan bir "Artificial Vision-Yapay Görü" Ar-Ge ve teknoloji şirkettir.

Arvis Teknolojinin başında yer alan Tansel Akgül ve Gülhan Ertürk Akgül şirketin kurucularındandır aynı zamanda girişimcilerdir.

Firmada 10 tane mühendis çalışmaktadır.

Firmada yazılım dilleri olarak C++, Python, QT olmak üzere çeşitli diller, Yapay Zeka araçları olarak Tensorflow, Pytorch, vb. ML, DL, CNN, vb. araçlar kullanılmaktadır. Firmada yer alan stajyerlere YZ, API, SDK, Mobil-IOS, Android desteği sunulmaktadır.

Sunucu, Bilgisayar, Uzak sunucu(bulut sistemleri), notebook, gerekirse IOT Cihazları, gömülü cihaz-donanımları, sensörler, ofis ortamındaki çeşitli donanım: kamera sistemleri, network sistemleri, ofis altyapısı vb. araçlar stajyerlerin kullanımına sunulmuştur.

Firmada yaz dönemi stajı için 72 öğrenci seçilmiştir. Bütün öğrenciler farklı üniversite ve bölümlerden oluşmaktadır.

Arvis Teknoloji ile iletişime geçilmek istenirse şirketin Kurucu/CTO-Teknoloji Direktörü olan Tansel Akgül'e ulaşılabilir. Tansel Bey'den bilgi edinmek istenirse 0530 122 78 43 nolu numara aranabilir, tansel.akgul@arvis.com.tr adresine mail yazılabilir veya şirketin web adresine ulaşılabilir ([www.arvis.com.tr](http://www.arvis.com.tr)).

### **3. YAPILACAK OLAN VEYA YAPILAN İŞ**

#### **3.1 Amaç**

Ülkemizdeki trafik kazalarının çoğu sürücülerden kaynaklanmaktadır. Bu kazalar, sürücülerin uykusuz olmaları, emniyet kemerlerini takmamaları, telefonla konuşmaları, dikkatsizlik gibi nedenlerle meydana gelmektedir. Buradaki amaç kazaları önlemek amacıyla sürücülerin dikkatini toplamasını sağlamak aynı zamanda hem sigorta şirketlerine maddi açıdan tasarruf sağlamak ve sürücülere kaza yapmamalarına yardımcı olmak ve kaza yapıldığı takdirde araçlarındaki hasarları görebilmelerini sağlamaktır.

#### **3.2 Tanım**

Jupyter Notebook'ta Python ile yazılmış, yapay zeka araçlarıyla geliştirilmiş ve Android Studio üzerinden Flutter ile yazılmış mobil uygulamaya entegre edilen Otonom Araçlar için Nesne Tanıma ve Davranışsal Biyometri-İnsan Davranışları Analizi adında bir projedir.

### 3.3 Süreç ve İşleyiş

Arvis Teknoloji’de Yapay Zeka Ar-Ge Stajyeri olarak görev alınmaktadır. Stajda 10 tane proje yapılmakta olup her projede 5-6 kişi bulunmaktadır. 10. Grup olan Otonom sistemler ve Yapay Zeka takımı olarak 8 kişiden oluşmaktadır. Her bir grup üyesinin farklı görevleri bulunmaktadır. Stajın ilk iki haftasında projeye ilgili literatür araştırması yapılmış araştırma sonucunda nihai çözüme ulaşılmıştır. Stajın üçüncü haftasında model araştırması yapılmış şirket tarafından verilen veri seti doğru şekilde etiketlenmiştir. Etiketlenen veriler belirlenen modellerle eğitilmiş en uygun çıktı alınmaya çalışılmıştır. Stajın dördüncü haftasında model geliştirmeleri devam ederken yüz tespit çalışmalarına başlanmıştır. Stajın beşinci haftasında mobil arayüz tasarlanmaya başlanmış hasar tespit kısmında model geliştirmelerine devam edilmiştir. Stajın altıncı haftasında projenin son hali ortaya çıkmaktadır. Mobil uygulamaya eğitilen model entegre edilmiş çekilen veya seçilen araç hasar görüntüsünün çıktısı alınmış proje başarıyla sonuçlanmıştır.

## 4. PROJE

### 4.1 Amaç

Sürücülerde sürüş sırasında uykululuk hali bulunabilir. Sürücüler uykulu hallerinde kaza yaptıklarında kaza sebebinin uykululuk olmasını ispatlamak oldukça zordur. Bu durumda ispat edilemediğinden sigorta şirketleri oluşan maliyetleri karşılamaktadır. Bu nedenle araç hem sigorta şirketlerine maddi açıdan yarar sağlamak hem de sürücünün canını korumak amacıyla araçlarda bulunan kameraların yardımıyla oluşan kazalarda sürücünün uykulu olup olmadığı belirlenmekte ve sürücünün uykulu olması takdirde sürücüye uyarı yapılmaktadır. Araç içindeki kameralar, sürücünün davranışlarını izleyerek uykulu olduğunu tespit eder ve gerekirse sürücüyü uyarmak için önlemler alır. Kaza yapan sürücü aynı zamanda ekspertise gitmeden aracın hasarlarını mobil uygulama üzerinden görebilir, böylece aracındaki hasarın türünü anlayabilir ve kaza yaptığı yerin konumuna da erişebilir.

### 4.2 Tanım

Jupyter notebook ve flutterla yapılmış bir projedir. Proje yüz analizi, hasar tespiti ve mobil uygulama kısımları bulunmaktadır. Yüz analizi kısmında media pipe kütüphanesi kullanılarak sürücülerin uykululuk durumları ölçülmüş ve gerektiğinde

sürücüye uyarılar verilmiştir. Hasar tespiti kısmında elde edilen veriler VGG Image Annotator ile etiketlenip birden fazla modelle eğitilmiş ve en uygun model seçilmiştir. Mobil uygulama kısmında Android Studio üzerinden flutterla arayüz tasarlanmış yapay zeka modeli arayüze entegre edilmiştir. Mobil uygulama aracılığıyla araçlardaki hasarlar tespit edilmiş ve anlık konuma erişilmiştir. Kaza tutanağı dijitalleştirilerek sürücülere kolaylık sağlanmıştır.

#### **4.3 Gereksinimler ve İhtiyaçlar**

Proje için jupyter notebook ve Android Studio gereklidir. Projede yüz analizi kısmında Pytorch, Media Pipe, Open CV, hasar tespiti kısmında Pytorch, mobil uygulama kısmında Pytorch Lite, Lottie, image\_picker, Google\_maps\_flutter, path\_provider ve flutter\_pdfview kütüphaneleri kullanılmıştır. Model eğitimi yapılırken GPU 'ya ihtiyaç duyulmuştur. Android Studio'da mobil uygulama geliştirildiğinden yüksek RAM'li bilgisayar tercih edilmiştir.

#### **4.4 Analiz**

İnsan için uyku fizyolojik bir ihtiyaçtır. İnsan günün üçte birini uykuyla geçirmektedir. Uyku ile yeterli dinlenme sağlanması halinde gün içerisinde yapılan işler daha verimli olmaktadır. Sürücü yeterli bir şekilde dinlenemediği zaman sürüş sırasında uykusu gelmeye başlar hem kendini hem de diğer sürücüler ve yayaları tehlikeye atar. Bu durum uykululuk haline bir örnektir. Bu sebeple sürücülerdeki uykululuk hali trafik kazalarının başlıca nedeni olup ciddi yaralanmalara, ölümlere ve maddi kayıplara neden olmaktadır.

TUİK verilerine göre Türkiye'de 2020 yılında toplam 983 bin 808 adet trafik kazası meydana gelmiş ve bu kazaların 833 bin 533 adedi maddi hasarlı, 150 bin 275 adedi ise ölümlü yaralanmalı trafik kazasıdır. Meydana gelen 150 bin 275 adet ölümlü yaralanmalı trafik kazası sonucunda 2 bin 197 kişi kaza yerinde, 2 bin 669 kişi ise yaralanıp sağlık kuruluşlarına sevk edildikten sonra kazanın sebep ve tesiriyle 30 gün içinde hayatını kaybetmiştir [1].

Tablo 4.1 2009-2020 yılları arasındaki Türkiye’de meydana gelen trafik kazaları [1]

Yıl	Toplam Kaza Sayısı	Ölümlü Yaralanma Kaza Sayısı	Maddi Hasarlı Kaza Sayısı	Ölü Sayısı			Yaralı Sayısı
				Toplam	Kaza Yerinde	Kaza Sonrası	
2009	1.053.346	111.121	942.225	4.324	4.324	-	201.380
2010	1.106.201	116.804	989.397	4.045	4.045	-	211.496
2011	1.228.928	131.845	1.097.083	3.835	3.835	-	238.074
2012	1.296.634	153.552	1.143.082	3.750	3.750	-	268.079
2013	1.207.354	161.306	1.046.048	3.685	3.685	-	274.829
2014	1.199.010	168.512	1.030.498	3.524	3.524	-	285.059
2015	1.313.359	183.011	1.130.348	7.530	3.831	3.699	304.421
2016	1.182.491	185.128	997.363	7.300	3.493	3.807	303.812
2017	1.202.716	182.669	1.020.047	7.427	3.534	3.893	300.383
2018	1.229.364	186.532	1.042.832	6.675	3.368	3.307	307.071
2019	1.168.144	174.896	993.248	5.473	2.524	2.949	283.234
2020	983.808	150.275	833.533	4.866	2.197	2.669	226.266

TUİK verilerine göre ülkemizde 2020 yılında gerçekleşen trafik kazalarında ölen kişilerin %49,4’ü sürücü, %30,7’si yolcu, %19,9’u ise yayadır. Ölümlü yaralanmalı trafik kazaları incelendiğinde bu kazalara %88,3 sürücü, %7,0 yaya, %2,7 taşıt, %1,4 yolcu ve %0,5 yol kaynaklı sebebiyet vermektedir. Meydana gelen 150 bin 275 ölümlü yaralanmalı kazanın %66,9’u gündüz, %30,4’ü gece ve %2,7’si alacakaranlıkta olmuştur [1].

Gerçekleşen bu kazalarda birisi hayatta kalmadığı veya sürücünün durumuna tanıklık etmediği sürece kazanın uyku hali halinden dolayı meydana geldiğini belirlemek ve ispat etmek çok zor olmaktadır. Bu nedenle, araç içindeki kameraların kullanımı, bu tür durumlarda büyük bir öneme sahiptir. Bu kameralar, kazanın nasıl meydana geldiğini ve sürücünün uykulu olup olmadığını belirlemeye yardımcı olabilir. Böylece, sürücülerin uykulu hallerini tespit ederek kazaların önlenmesine katkı sağlanabilir.

Uykusuzluk halinin önlenmesi için alınacak önlemlerden biri de davranışsal önlemlerdir. Davranışsal önlemler, sürücünün esnemesini, baş pozisyonunu, ağız ve göz açıklığı gibi parametreleri ölçerek uykusuzluk halini ölçer ve uyarı verir. Otomotiv sektöründe, uykulu sürüşü tespit etmek ve sürücülerini uyarmak için sistemler geliştirilmeye başlanmıştır [1].

#### 4.5 Gelişim Süreci

Bu projede yukarıda belirtilen parametreler incelenerek sürücülerin durumları kontrol edilmiş ve uykusuzluk halini tespit etmek için belirli aralıklarla sürücülere uyarılar verilmiştir. Ayrıca, hasar tespiti için Arvis Teknoloji tarafından yaklaşık 5000 hasarlı araç verisi toplanmıştır. Bu veriler titizlikle incelenerek kırık/kayıp, airbag, astar, cam çatlağı, çizik, far hasarı, göçük, kaput hasarı olmak üzere 8 farklı sınıfa ayrılmıştır. Etiketlenmiş veriler, 8 farklı modelde eğitilmiş ve en iyi doğruluk oranına sahip olan model seçilerek geliştirilmiştir.

Mobil uygulama, Android Studio ortamında Flutter kullanılarak geliştirilmiştir ve kullanıcılar için giriş, kayıt ol, anasayfa, hakkımızda, iletişim gibi sayfalar içerir. Anasayfada, kullanıcılar kaza yaptıklarında uygulamayı açarak araçlarının hasarlarını fotoğraflayabilir veya galeriden fotoğraf seçebilirler. Ayrıca, kaza yaptıkları yerin konum bilgisine de erişebilirler. Bu konum bilgisi, can kaybını önlemek amacıyla ilgili yetkililere paylaşılır. Sürücüler ayrıca kaza sonrası kaza tutanaklarını doldurmak için zaman kaybetmek zorunda kalmazlar; bu kaza tutanağına "Careless" adını verdiğiniz mobil uygulama üzerinden erişebilirler ve istedikleri yetkililerle paylaşabilirler.

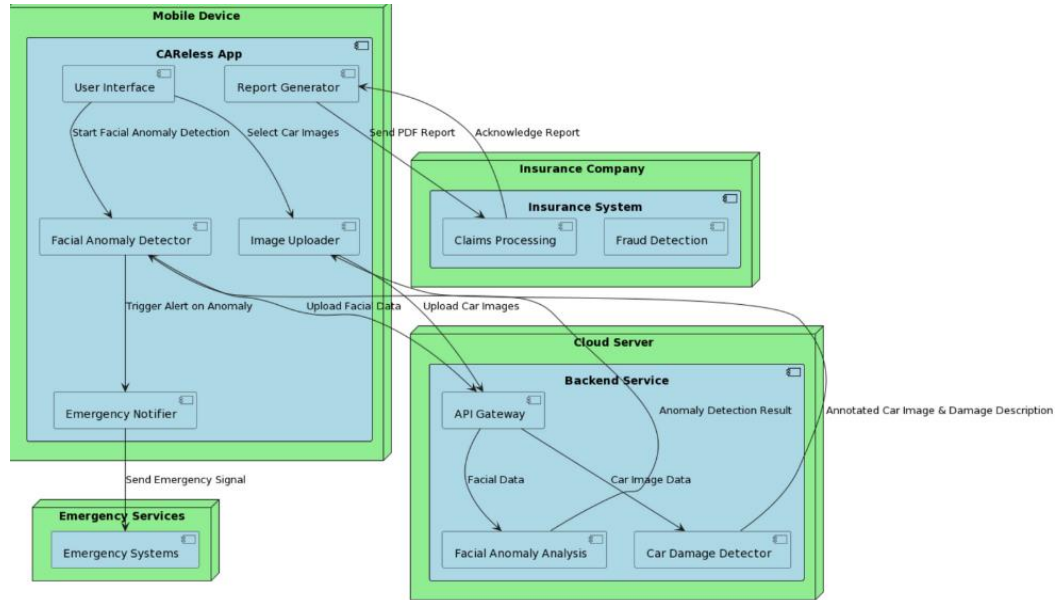


Şekil 4.2 Uygulamanın Kullanışı





Şekil 4.3 Kaza Yapan Kullanıcıların Konumlarının İlgili Yetkililere İletilmesi

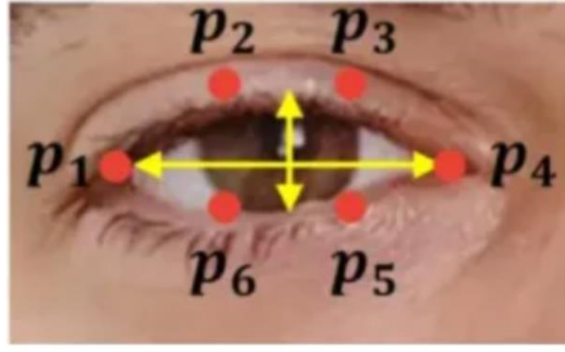


Şekil 4.4 Projenin Nasıl Kullanıldığını Gösteren Teknik Anlatım

## 4.6 Test ve Uygulama

### 4.6.1 Yüz Analizi

Yüz analizinde sürücülerdeki uykululuk hallerinin belirlenebilmesi için bazı parametreler incelenmiştir. Bu parametreler; göz açıklığı, göz kırpma sayısı, ortalama göz kırpma sayısı, 500 ms'den uzun süren mikro uykuların sayısı, esneme sayısı ve esneme süresidir.



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

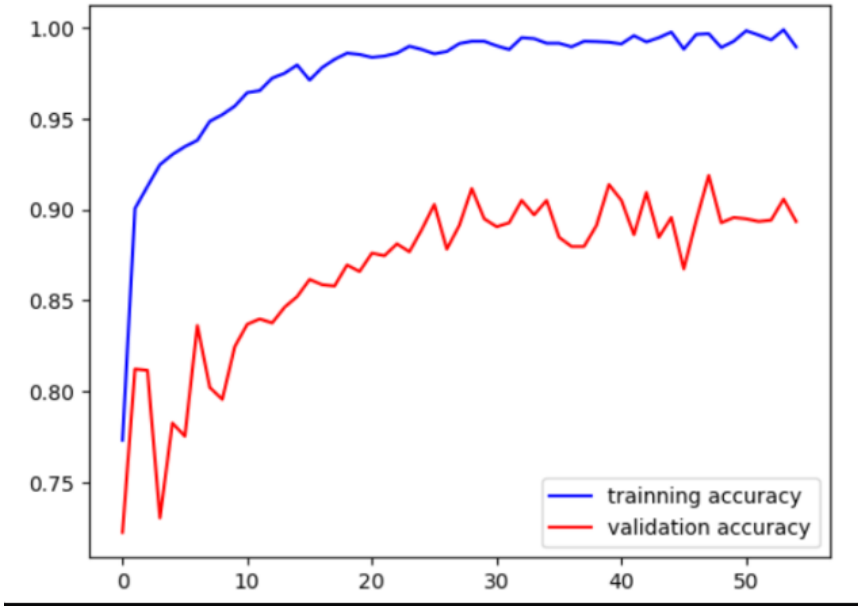
Şekil 4.1.1 Göz Açıklığının Hesaplanması [2]

6900 görüntüden oluşan veri açık göz, kapalı göz, esneme ve esneme yok olmak üzere 4 farklı sınıfa ayrıldı. Bu veriler Google'ın sağlamış olduğu Media Pipe modellerinden face\_landmarker\_v2\_with\_blendshapes modeliyle eğitildi ve %98 oranında bir başarı elde edildi.

Tablo 4.1.2 face\_landmarker\_v2\_with\_blendshapes Modelinin Doğruluk Sonuçları

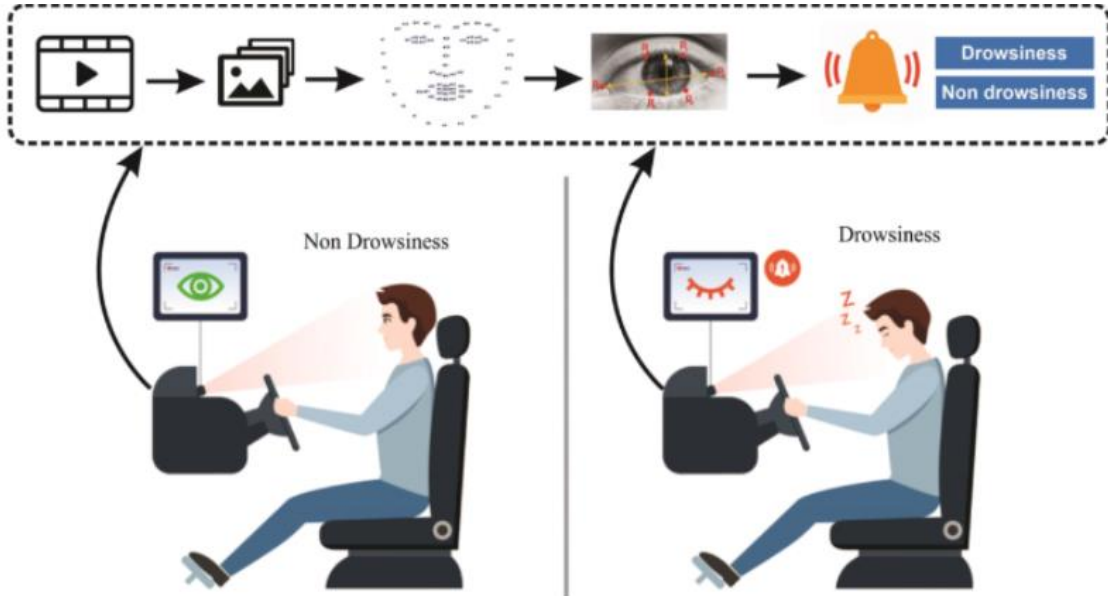
y: 0.8927	
Epoch 50/55	
691/691 [=====]	- 137s 199ms/step - loss: 0.0208 - accuracy: 0.9926 - val_loss: 0.4902 - val_accu
y: 0.8956	
Epoch 51/55	
691/691 [=====]	- 136s 196ms/step - loss: 0.0070 - accuracy: 0.9984 - val_loss: 0.5653 - val_accu
y: 0.8949	
Epoch 52/55	
691/691 [=====]	- 137s 198ms/step - loss: 0.0139 - accuracy: 0.9960 - val_loss: 0.6594 - val_accu
y: 0.8934	
Epoch 53/55	
691/691 [=====]	- 136s 196ms/step - loss: 0.0231 - accuracy: 0.9933 - val_loss: 0.6794 - val_accu
y: 0.8941	
Epoch 54/55	
691/691 [=====]	- 136s 197ms/step - loss: 0.0036 - accuracy: 0.9989 - val_loss: 0.7263 - val_accu
y: 0.9057	
Epoch 55/55	
691/691 [=====]	- 144s 208ms/step - loss: 0.0399 - accuracy: 0.9895 - val_loss: 0.7317 - val_accu
v: 0.8934	

```
plt.plot(epochs, accuracy, "b", label="training accuracy")
plt.plot(epochs, val_accuracy, "r", label="validation accuracy")
plt.legend()
plt.show()
```



Şekil 4.1.3 Eğitilen Modelin Test ve Train Değerlerinin Karşılaştırılması

Model değerlendirilmesi yapıldıktan sonra sürücülerin göz açıklığı parametreleri ölçülerek uykululuk halleri incelendi. Uykululuk hali durumunda sürücüye haber verilerek uyarıldı.

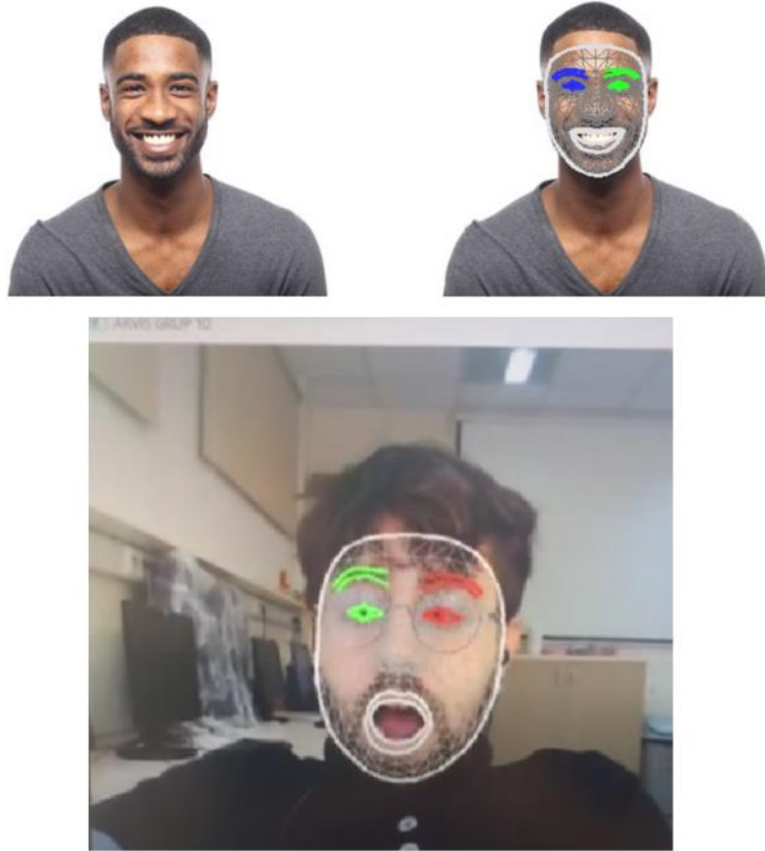


Şekil 4.1.4 Sürücülerin Uykululuk Hallerinde Uyarı Verilmesini Anlatan Görsel [3]

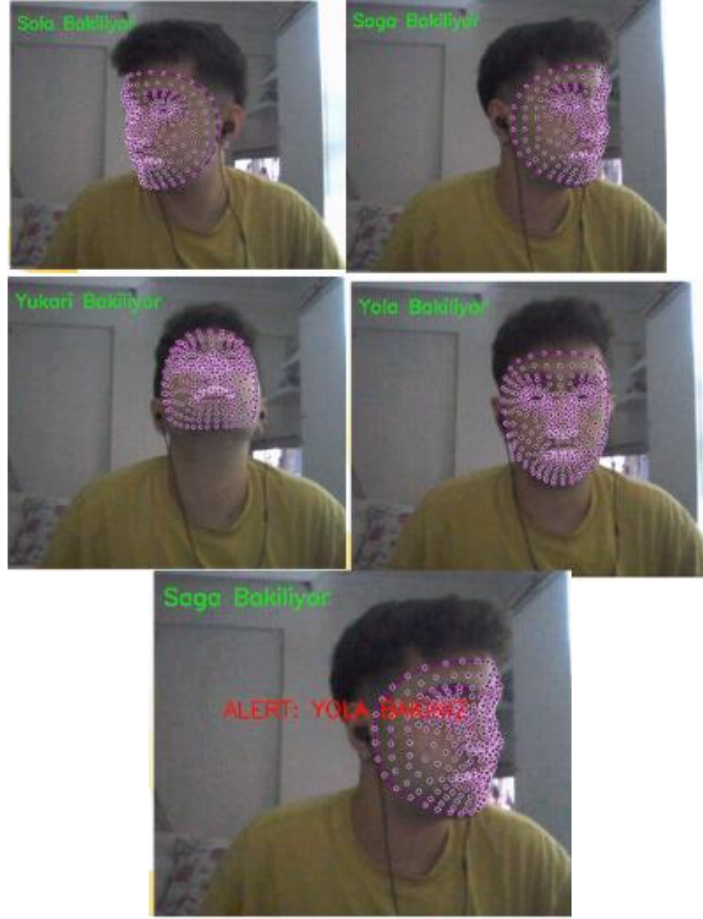


Şekil 4.1.5 Sürücün Uykululuk Halinin Ölçülmesi Ve Uyarılması

MediaPipe kütüphanesinde eğitilen modeli entegre ederek sürücülerin yüzlerini analiz edildi ve farklı yüz bölgelerini işaretlenildi. Bu işaretlemeler, kişilerin her hareketinde yüzlerinde belirli noktaları ve çizgileri takip etmekte ve bu hareketleri algılamaktadır.



Şekil 4.1.6 Sürücülerin Yüz Hareketlerinin Takip Edilmesi

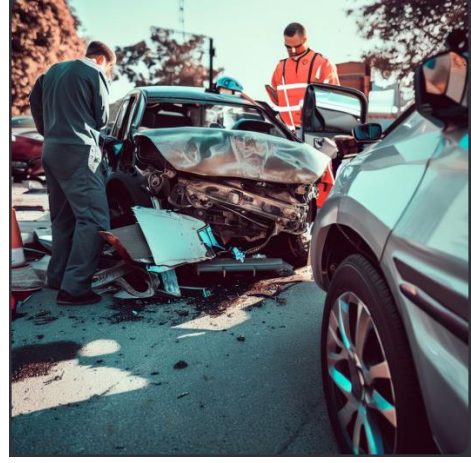
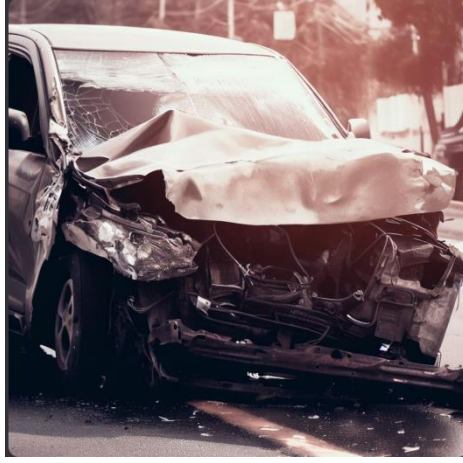


Şekil 4.1.7 Sürücülerin Yüz Hareketleri İncelenerek Uyarı Verilmesi

Çalışma sırasında, sürücülerin kafa hareketleri de yakından izlenmiştir. Sürücülerin direksiyon dışındaki bölgelere (sağa, sola veya aynalara) ne kadar süreyle baktığı kaydedilmiş ve bu süre belirli bir eşik değerini aştığında, sürücülere "yola bakınız" uyarısı otomatik olarak verilmiştir. Bu uyarı, sürücülerin dikkatlerini tekrar yola yoğunlaştırmalarına yardımcı olmuştur.

#### 4.6.2 Hasar Tespiti

Hasar tespiti için Arvis Teknoloji'den yaklaşık 5000 tane hasarlı araç verisi alınmıştır. Hasarlı araç verilerini 3. Tarafla paylaşımı yasak olduğundan örnek olarak hasarlı araç fotoğrafı ChatGpt tarafından oluşturulmuştur. Bu veriler titizlikle incelenerek gerekli görünmeyen bazı fotoğraflar silinerek yaklaşık 4800 veriye düşürülmüştür. VGG Image Annotator ve Roboflow kullanılarak veriler etiketlenmiştir.



Şekil 4.2.1 Chatgpt Tarafından Oluşturulan Hasarlı Araç Verileri

Veriler ilk başta çizik, kırık, göçük, kayıp, airbag, cam çatlağı, ön far hasar, arka far hasar, lastik, jant hasar, kaput hasar ve astar olmak üzere 12 farklı sınıfa ayrılarak etiketlenmiştir. Etiketlenip sınıflara ayrılan veriler arasındaki tutarsızlık sonucu yapay zeka hasar tespitinde başarılı olamamıştır. Böylelikle benzer olan kırık ve kayıp, ön far hasar ve araka far hasar verileri birleştirilmiş, lastik, jant hasar verileri de çıkarılmıştır. Etiketlenen verilerin son hali çizik, kırık-kayıp, göçük, airbag, cam çatlağı, far hasar, astar ve kaput hasar olmak üzere 8 sınıfa ayrılmıştır.

cizik	1981	
kirik	1070	
gocuk	889	
kayip	514	
airbag	62	
cam_catlagi		169
on_far_hasar		204
arka_far_hasar		65
lastik	36	
jant_hasar		34
kaput_hasar		179
astar	294	
]: toplam		
]: 5497		

Şekil 4.2.2 Sınıflara Ayrılan Verilerin Tutarsızlığı

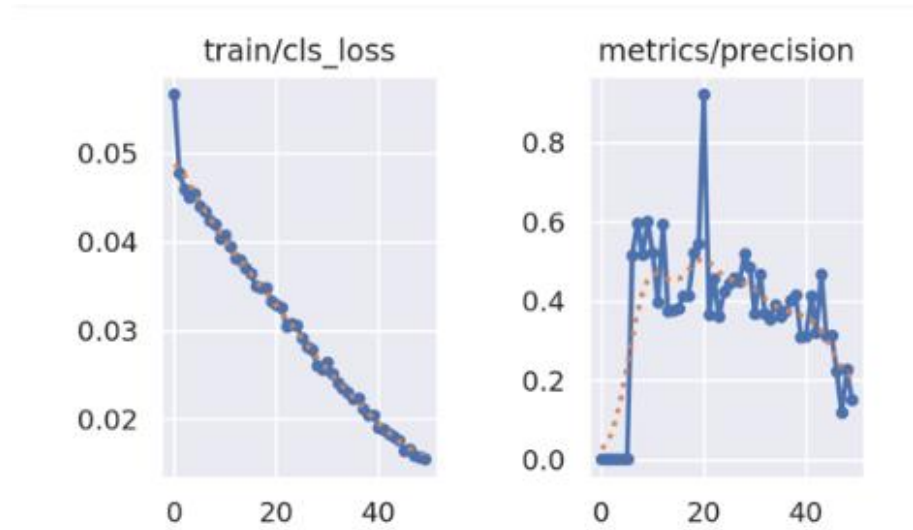




Şekil 4.2.3 VGG İmage Annotator ile Etiketlenen Hasarlı Araç Fotoğrafları

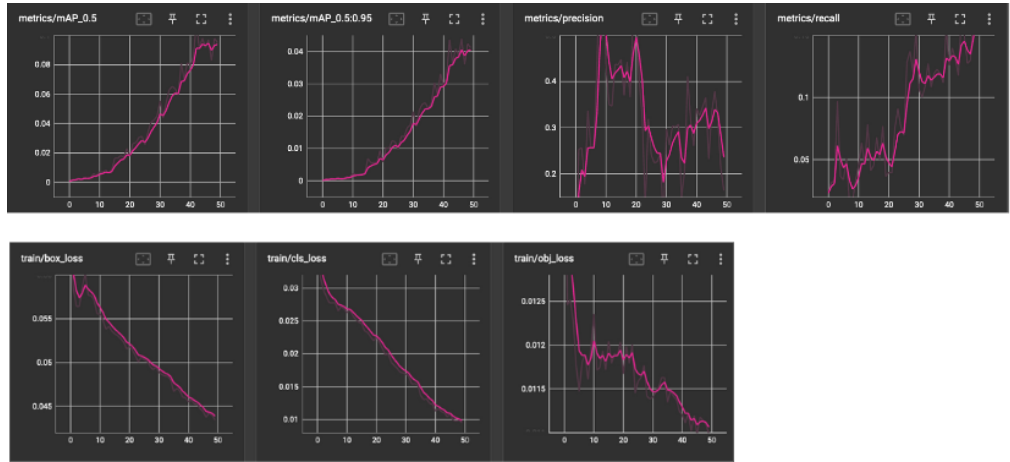
Etiketlenen veriler Yolov3, Yolov7, Yolov8, Mask-RCNN, Fast-RCNN ve Detectron2 gibi modellerle eğitilerek her bir modelin tahmin değerleri incelenerek en uygun model seçilmiştir. Tutarsız verilerle eğitildiğinde tüm modellerde başarısızlık görülmüştür. Veriler düzeltildikten sonra tüm modellerle eğitildiğinde Yolov3, Yolov5 ve Yolov5 modellerinde istenilen sonuca ulaşılammıştır. Mask-RCNN ve Fast-RCNN modellerinde sonuçlar iyi olmasına rağmen Detectron2 modeli en iyi sonucu vermiştir. Modellere ait çıktılar veri gizliliğinden dolayı verilmemektedir.

### Yolov3



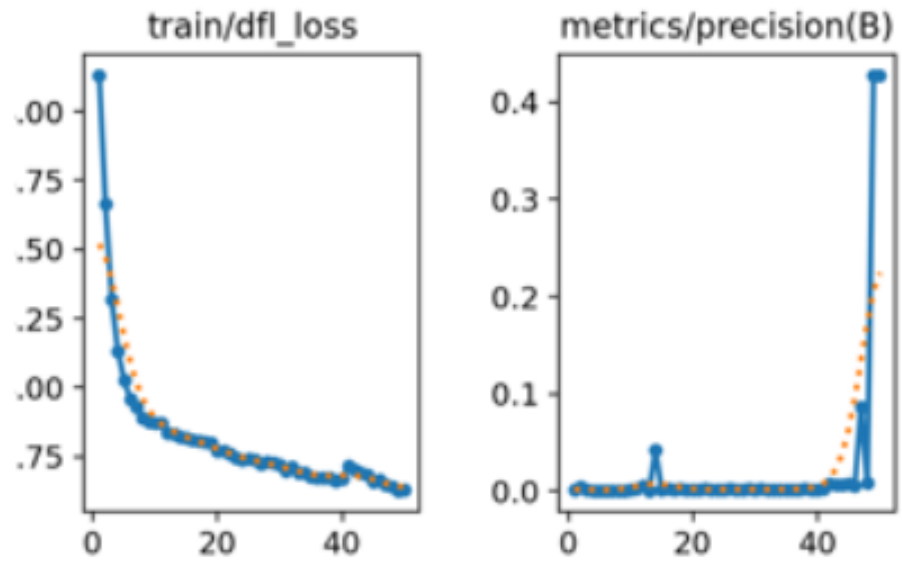
Şekil 4.2.4 Yolov3 Modelinin Tahmin ve Kayıp Sonuçları

## Yolov7



Şekil 4.2.5 Yolov7 Modelinin Tahmin ve Kayıp Sonuçları

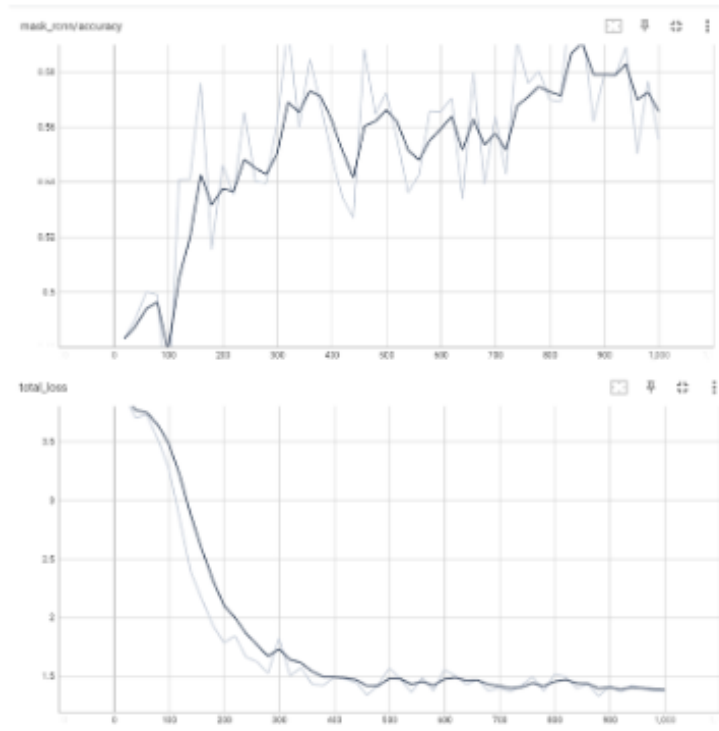
## Yolov8



Şekil 4.2.6 Yolov8 Modelinin Tahmin ve Kayıp Sonuçları

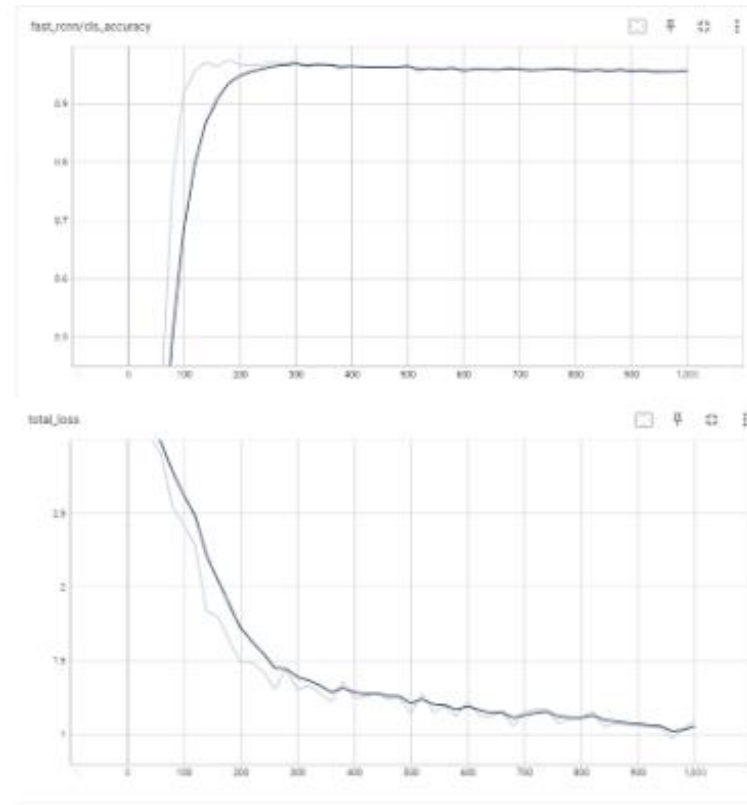


## Mask-RCNN



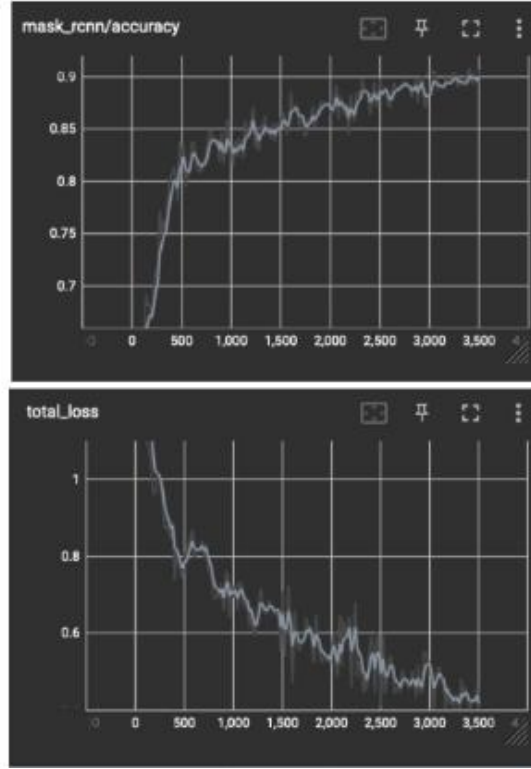
Şekil 4.2.7 Mask-RCNN Modelinin Tahmin ve Kayıp Sonuçları

## Fast-RCNN



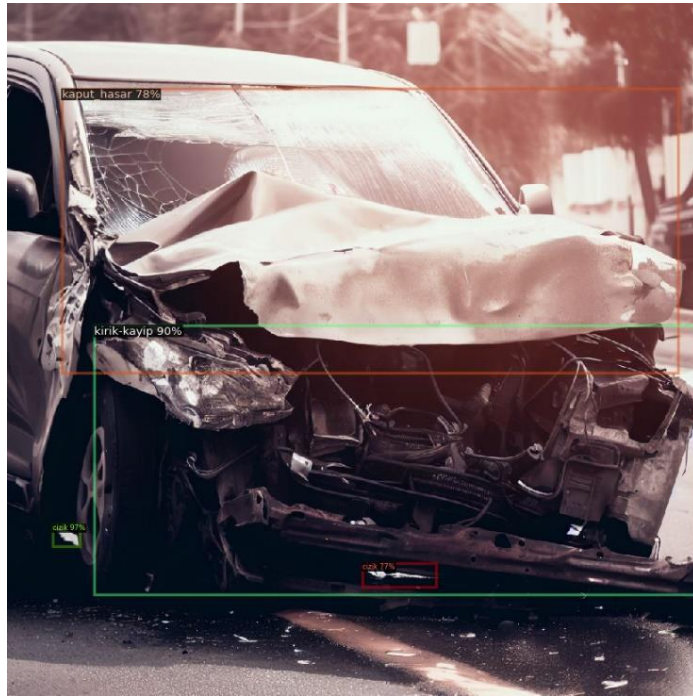
Şekil 4.2.8 Fast-RCNN Modelinin Tahmin ve Kayıp Sonuçları

### Mask R-CNN X-101



Şekil 4.2.9 Mask R-CNN X-101 Modelinin Tahmin ve Kayıp Sonuçları

Aşağıda yukarıdaki modellerden biriyle eğitilmiş olan hasarlı araç verisi bulunmaktadır. Her ne kadar istenilen sonuca ulaşılamasa da yüksek bir doğruluk oranına sahiptir.



Şekil 4.2.10 Eğitilen Veri Sonrası Çıktı

Yukarıdaki sonuçlara bakıldığında Detectron2 'ye ait Mask R-CNN X-101 modeli en iyi sonuç veren model olmuştur. Hasar tespit çalışmasında bu model kullanılmıştır.

#### 4.6.3 Mobil Uygulama

Android Studio kullanılarak Flutter ile mobil bir arayüz geliştirilmiştir. Uygulama, Lottie, image\_picker, Google\_maps\_flutter, path\_provider ve flutter\_pdfview gibi Flutter'a ait kütüphaneleri içermektedir. Kullanımı kolay bir arayüze sahip olan uygulama, her yaştan kullanıcı için tasarlanmıştır ve şu ana ekranlardan oluşmaktadır:

**Giriş Ekranı:** Kullanıcı uygulamayı açtığında karşısına giriş ekranı gelir. Eğer daha önce kaydolmadıysa, buradan kayıt olabilir ve ardından giriş yapabilir.

**İletişim:** İletişim bilgilerini içeren bir ekran.

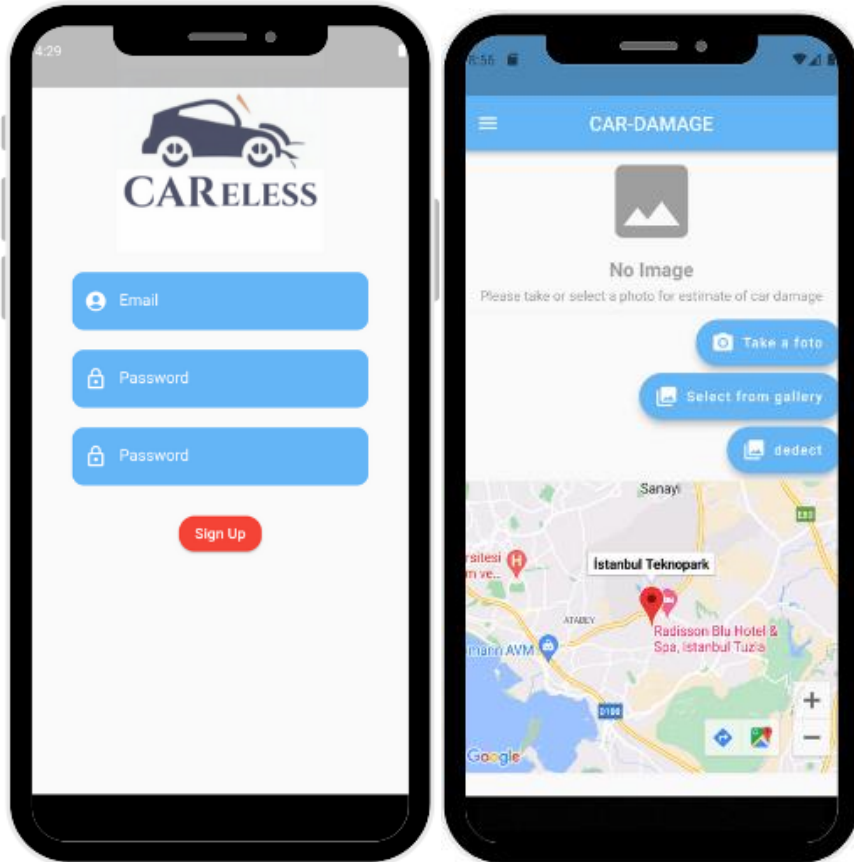
**Hakkımızda:** Uygulama hakkında bilgi veren bir sayfa.

**Anasayfa:** Kullanıcı, aracının fotoğrafını çekebilir veya galeriden fotoğraf seçebilir. Ardından "Detect" butonuna basarak aracındaki hasarları görebilir ve kaza yaptığı yerin konumuna erişebilir. Kaza durumunda, can kaybını önlemek amacıyla kaza yapan sürücünün konumu ilgili yetkililere iletilir. Kaza yapan kullanıcılar, mobil uygulama aracılığıyla kaza tespit tutanağını doldurabilirler, böylece kağıt tutanakla uğraşma zorunluluğu ortadan kalkar.

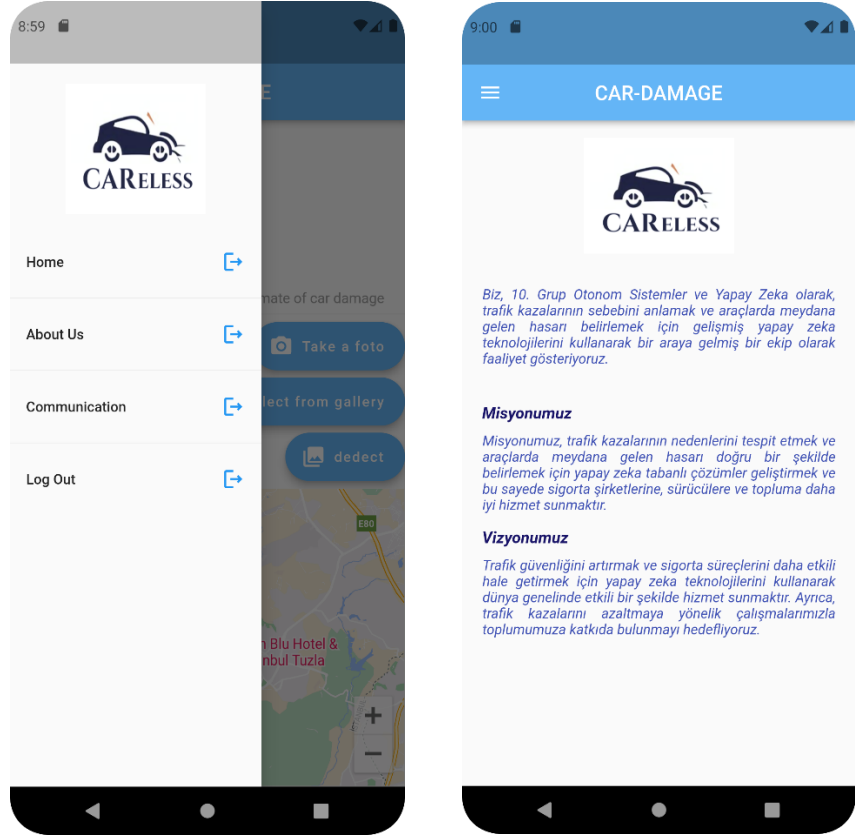
Mobil uygulamada kullanılan bazı önemli hususlar vardır. Hasar tespitinde kullanılan model mobil uygulamaya pytorch-lite kütüphanesiyle entegre edilmiştir. Kullanıcının anlık konumuna erişmek için Google maps API Key kullanılmıştır. Kaza tutanağına erişim sağlamak için flutter\_pdfview kütüphanesi kullanılmıştır.



Şekil 4.3.1 Açılış Ekranı ve Giriş Ekranı



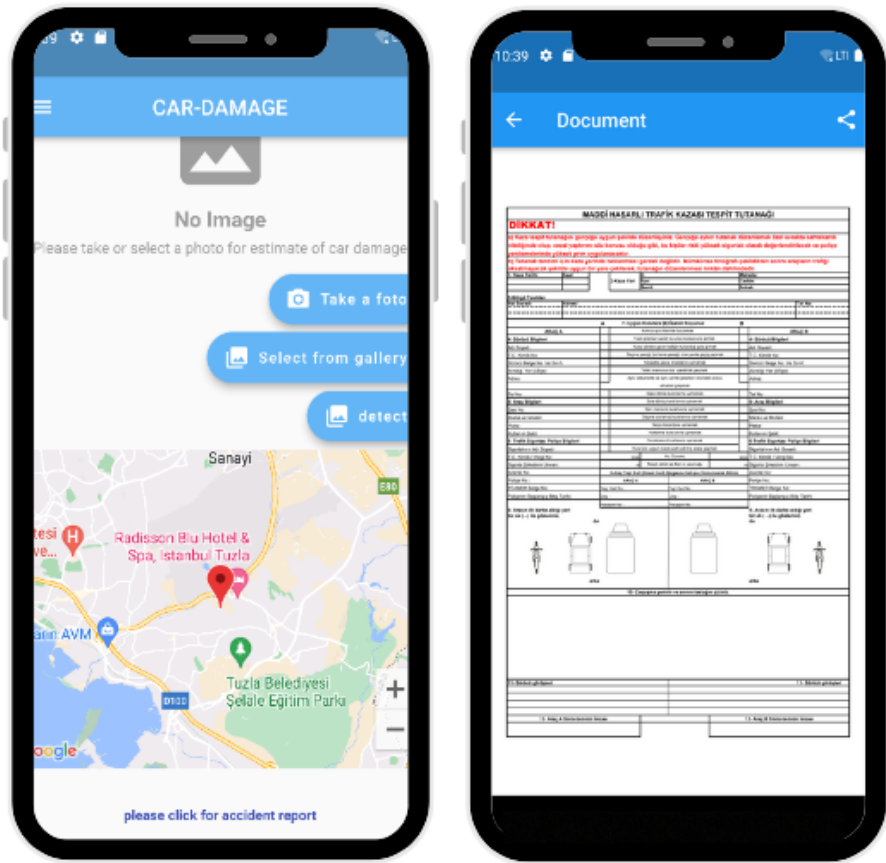
Şekil 4.3.2 Kaydol Ekranı ve Anasayfa



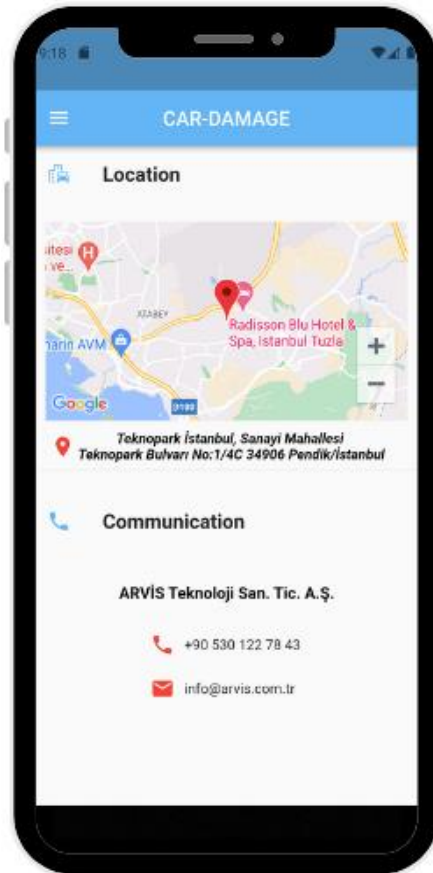
Şekil 4.3.3 Menü Ekranı ve Hakkımızda Ekranı



Şekil 4.3.4 Hasarlı Araç Fotoğrafı Görüntüleme ve Hasar Görme Ekranı



Şekil 4.3.5 Kaza Tutanağına Erişme ve Görüntüleme Ekranı



Şekil 4.3.6 İletişim ekranı

## 4.7 Proje Sonucu

Proje sürecinde birçok farklı model denendi ve bu modeller karşılaştırıldı. Bu çalışma, grafiklerin doğru bir şekilde okunmasının önemli olduğu sonucuna vardı. Proje, özellikle sigorta şirketlerine maddi fayda sağlarken aynı zamanda sürücülerin can güvenliğine de katkıda bulunuyor. Yüz analizi kullanarak yolcuların dikkatini toplamalarına yardımcı olacak uyarı mesajları da sunuluyor. Aynı zamanda, kaza yapan sürücülerin araçlarındaki hasarları önceden görmelerine ve kaza yerinin konumunu elde etmelerine olanak tanıyan bir sistem geliştiriliyor. Bu sayede kaza tutanağı doldurmak gibi zaman alıcı işlemlerle uğraşmaktan kaçınılabılır.

Proje süresince bir dizi zorlukla karşılaşıldı. Bunlar, yanlış etiketleme, doğru modelin seçimi ve veri dengesizliği gibi konuları içermekteydi. Veri etiketlemesi sırasında grup üyelerinin her biri farklı etiketlemeler yapması, veri tutarsızlıklarına yol açtı. Bu nedenle, verileri daha tutarlı hale getirmek için 12 farklı sınıf yerine 8 sınıfa ayrıldı.

## 5. SONUÇ

Proje süresince 6 haftalık bir staj gerçekleştirildi ve haftalık sunumlar, sunum hazırlama becerilerimi ve topluluk önünde konuşma yeteneğimi geliştirmeye yardımcı oldu. Ayrıca, grup içi ve gruplar arası etkileşimlerle aktif öğrenme deneyimi yaşadım ve ekip çalışmaları sayesinde organizasyon becerilerimi geliştirme fırsatı buldum.

## 6. KAYNAKLAR

[1] Peker, Nur Yasin. "Sürücülerde uykululuk hali tespit sistemi tasarımı ve uygulaması= Design and implementation of drowsiness detection system for drivers." (2022).

[2] Pandey, D., "Eye Aspect Ratio(EAR) and Drowsiness detector using dlib," Nisan 2021. <https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706>

[3] Motoroto Ekibi, "Sürücü Uykusu Tespit Sistemi Nedir? ," Kasım 2022. <https://www.motoroto.com.tr/surucu-uykusu-tespit-sistemi-nedir/>

## 7. EKLER

```
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'Login.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  //this widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: splash()
    );
  }
}
```

```

class splash extends StatefulWidget {
  const splash({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(debugShowCheckedModeBanner: false);
  }
  State<splash> createState() => _splashState();
}
class _splashState extends State<splash> {

  @override
  void initState() {
    super.initState();
    Future.delayed(Duration(seconds: 7), () {
      Navigator.push(
        context, MaterialPageRoute(
          pageBuilder: (_, __, ___) =>Login(),
          transitionsBuilder: (_, animation, __, child) {
            return FadeTransition(
              opacity: animation,
              child: child,
            );
          },
        ),
      );
    });
  }
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Padding(
            padding: const EdgeInsets.only(top:150.0, left: 20.0),
            child: Container(
              width: 300.0,
              height: 300.0,
              child:
                Image.asset("Resimler/cardamage1.png"),
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: Lottie.asset("Resimler/cardamage.json", repeat: true),
          ),
        ],
      ),
    );
  }
}

```

```

import 'package:cardamage/SignUp.dart';
import 'package:flutter/material.dart';

import 'Anasayfa.dart';

class Login extends StatefulWidget {
  @override
  LoginState createState() => _LoginState();
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false
    );
  }
}

class _LoginState extends State<Login> {
  @override
  Widget build(BuildContext context) {
    var ekranBilgisi=MediaQuery.of(context);
    final double ekranYuksekligi=ekranBilgisi.size.height;
    //final double ekranGenisligi=ekranBilgisi.size.width;
    //final screenSize = MediaQuery.of(context).size;

    return Scaffold(
      backgroundColor: Colors.blue[25],
      body: SingleChildScrollView(
        child: Column(
          children: [
            Padding(
              padding: const EdgeInsets.only(top: 20.0),
              child: Container(

```



23

[illegible]

```
import 'package:cardamage/Login.dart';
import 'package:flutter/material.dart';

class signUp extends StatefulWidget {
  @override
  State<signUp> createState() => _signUpState();
}

class _signUpState extends State<signUp> {
  @override
  Widget build(BuildContext context) {
    var ekranBilgisi=MediaQuery.of(context);
    final double ekranYuksekligi=ekranBilgisi.size.height;

    return Scaffold(
      backgroundColor: Colors.blue[25],
      //Color(0xf8f0e1),
      body: Padding(
        padding: const EdgeInsets.only(top: 20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children:[
            Container(
              width: 180,
              height: 180,
              decoration: BoxDecoration(
                //color: WColor.WGR_BLUE,
                image: DecorationImage(
                  colorFilter: ColorFilter.mode(
                    Colors.black.withOpacity(0.8), BlendMode.dstATop),
                  image: AssetImage('Resimler/cardamage1.png'),
                  fit: BoxFit.cover,
                ),
              ),
            ),
            Padding(
              padding: EdgeInsets.only(top: 20.0,
                left: ekranYuksekligi/17,
                right: ekranYuksekligi/17),
              child: TextField(
                decoration: InputDecoration(
                  prefixIcon: Icon(Icons.account_circle,color: Colors.white),
                  hintText: "Email",
                  hintStyle: TextStyle(color: Colors.white),
                  filled: true,
                  fillColor: Colors.blue[300],
                  enabledBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.all(Radius.circular(12.0)),
                    borderSide: BorderSide(color: Colors.transparent),
                  ),
                ),
              ),
            ),
            Padding(
              padding: EdgeInsets.only(top: 20.0,
                left: ekranYuksekligi/17,
                right: ekranYuksekligi/17),
              child: TextField(
                obscureText: true,
```

```

        decoration: InputDecoration(
          prefixIcon: Icon(Icons.lock_outline, color: Colors.white,),
          hintText: "Password",
          hintStyle: TextStyle(color: Colors.white),
          filled: true,
          fillColor: Colors.blue[300],
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.all(Radius.circular(12.0)),
            borderSide: BorderSide(color: Colors.transparent),
          ),
        ),
      ),
    ),
    Padding(
      padding: EdgeInsets.only(top: 20.0,
        left: ekranYuksekligi/17,
        right: ekranYuksekligi/17),
      child: TextField(
        obscureText: true,
        decoration: InputDecoration(
          prefixIcon: Icon(Icons.lock_outline, color: Colors.white,),
          hintText: "Password",
          hintStyle: TextStyle(color: Colors.white),
          filled: true,
          fillColor: Colors.blue[300],
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.all(Radius.circular(12.0)),
            borderSide: BorderSide(color: Colors.transparent),
          ),
        ),
      ),
    ),
    Padding(
      padding: const EdgeInsets.only(top:25.0),
      child: ElevatedButton(
        child: Text("Sign Up", style: TextStyle(
          fontSize: 15.0,
        )),
        onPressed: ()
        {
          print("giriş yapıldı");
          Navigator.push(context, MaterialPageRoute(builder: (context) => Login()));
        },
        style: ButtonStyle(
          backgroundColor: MaterialStateProperty.all<Color>( Colors.red),
          shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
              // Change your radius here
              borderRadius: BorderRadius.circular(16),
              side: BorderSide(
                color: Colors.transparent,
              ),
            ),
          ),
        ),
      ),
    ),
  ],
),
); // This trailing co

import 'package:flutter/material.dart';

import 'Anasayfa.dart';
import 'Communication.dart';
import 'Login.dart';

class About extends StatefulWidget {
  @override
  _AboutState createState() => _AboutState();
}

class _AboutState extends State<About> {
  @override
  Widget build(BuildContext context) {

    return Scaffold(
      backgroundColor: Colors.blue[25],
      drawer: new Drawer(
        width: 250,
        child: new ListView(
          children: [DrawerHeader(
            child: Align(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Container(
                    width: 140,
                    height: 130,

```



```

        "geliştirmek ve bu sayede sigorta şirketlerine, sürücülere ve "
        "topluma daha iyi hizmet sunmaktır.",
        textAlign: TextAlign.justify,
        style: TextStyle(
            color: Colors.indigo,
            fontStyle: FontStyle.italic,
            fontSize: 14),),
    ),
    Padding(
        padding: const EdgeInsets.only(left: 20, top:15),
        child: Row(
            children: [
                Text("Vizyonumuz",
                    textAlign: TextAlign.left,
                    style: TextStyle(
                        color: Color(0xFF181368),
                        fontWeight: FontWeight.bold,
                        fontStyle: FontStyle.italic,
                        fontSize: 16)
                ),
            ],
        ),
    ),
    Padding(
        padding: const EdgeInsets.only(left:20, right:20, top:10),
        child: Text("Trafik güvenliğini artırmak ve sigorta süreçlerini daha etkili "
            "hale getirmek için yapay zeka teknolojilerini kullanarak dünya "
            "genelinde etkili bir şekilde hizmet sunmaktır. Ayrıca, trafik "
            "kazalarını azaltmaya yönelik çalışmalarımızla "
            "toplumumuza katkıda bulunmayı hedefliyoruz.",
            textAlign: TextAlign.justify,
            style: TextStyle(
                color: Colors.indigo,
                fontStyle: FontStyle.italic,
                fontSize: 14),),
    ),
],
);
}

```

```

import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'About.dart';
import 'Anasayfa.dart';
import 'Login.dart';

class Communication extends StatefulWidget {
  @override
  _CommunicationState createState() => _CommunicationState();
}

class _CommunicationState extends State<Communication> {
  @override

  LatLng initialLocation = LatLng(40.8739, 29.3102);
  late GoogleMapController mapController;

  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[25],
      drawer: new Drawer(
        width: 250,
        child: new ListView(
          children: [DrawerHeader(
            child: Align(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Container(
                    width: 140,
                    height: 130,
                    child:
                      Image.asset("Resimler/cardamagel.png",
                        fit: BoxFit.fill),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
      new ListTile(
        title: Text("Home"),
        trailing: new Icon(Icons.logout_outlined,color:Colors.blue),
        onTap: () {
          Navigator.of(context).push(
            MaterialPageRoute(builder: (context) => Anasayfa()));
        },
      ),
    ),
  ),

```

```

        new Divider(),
        new ListTile(
            title: Text("About Us"),
            trailing: new Icon(Icons.logout_outlined,color:Colors.blue),
            onTap: () {
                Navigator.of(context).push(
                    MaterialPageRoute(builder: (context) => About()));
            },
        ),
        new Divider(),
        new ListTile(
            title: Text("Communication"),
            trailing: new Icon(Icons.logout_outlined,color:Colors.blue),
            onTap: () {
                Navigator.of(context).push(
                    MaterialPageRoute(builder: (context) => Communication()));
            },
        ),
        new Divider(),
        new ListTile(
            title: Text("Log Out"),
            trailing: new Icon(Icons.logout_outlined,color:Colors.blue),
            onTap: () {
                Navigator.of(context).push(
                    MaterialPageRoute(builder: (context) => Login()));
            },
        ),
    ],
),
),
appBar: AppBar(
    centerTitle: true,
    backgroundColor: Colors.blue[300],
    title: Text("CAR-DAMAGE"),
),
body: Column(
    children: [
        ListTile(
            title: Text("Location",
                style: TextStyle(
                    fontSize: 20,
                    color: Colors.blue[950],
                    fontWeight: FontWeight.bold
                ),
            ),
            leading: Icon(Icons.emoji_transportation, color: Colors.blue[300],),
        ),
        SizedBox(height: 20),
        Padding(
            padding: const EdgeInsets.only( left:15.0, right: 15.0 ),
            child: Container(
                height: 200,
                child: GoogleMap(
                    onMapCreated: (GoogleMapController controller) {
                        mapController = controller;
                    },
                    myLocationButtonEnabled: true,
                    mapType: MapType.normal,
                    tiltGesturesEnabled: true,
                    compassEnabled: true,
                    scrollGesturesEnabled: true,
                    zoomGesturesEnabled: true,
                    initialCameraPosition: CameraPosition(
                        target: initialLocation,
                        zoom: 12,
                    ),
                    markers:_createMarker()
                ),
            ),
        ),
        Padding(
            padding: const EdgeInsets.only(top:10.0, left: 20.0),
            child: Row(
                children: [
                    Icon(Icons.place, color: Colors.red, weight: 15,),
                    Text("Teknopark İstanbul, Sanayi Mahallesi \n Teknopark Bulvarı No:1/4C "
                        "34906 Pendik/İstanbul",
                        textAlign: TextAlign.center,
                        style: TextStyle(
                            color: Colors.black,
                            fontStyle: FontStyle.italic,
                            fontWeight: FontWeight.bold,
                            fontSize: 14
                        ),
                    ),
                ],
            ),
        ),
        Divider(),
        SizedBox(height:15),
        ListTile(

```

```

        title: Text("Communication",
          style: TextStyle(
            fontSize: 20,
            color: Colors.blue[950],
            fontWeight: FontWeight.bold
          ),
        ),
        leading: Icon(Icons.phone, color: Colors.blue[300]),
      ),
      SizedBox(height: 20),
      Padding(
        padding: const EdgeInsets.only(top: 15.0),
        child: Text("ARVIS Teknoloji San. Tic. A.Ş.",
          textAlign: TextAlign.center,
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
            color: Colors.black
          ),
        ),
      ),
      SizedBox(height: 20),
      Padding(
        padding: const EdgeInsets.only(top: 10, left: 120),
        child: Row(
          //mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(Icons.phone, color: Colors.red, weight: 15),
            SizedBox(width: 10),
            Text("+90 530 122 78 43"),
          ],
        ),
      ),
      SizedBox(height: 20),
      Padding(
        padding: const EdgeInsets.only(left: 120),
        child: Row(
          children: [
            Icon(Icons.mail, color: Colors.red, weight: 15),
            SizedBox(width: 10),
            Text("info@arvis.com.tr"),
          ],
        ),
      ),
    ],
  ),
);
}
Set<Marker> _createMarker() {
  return <Marker>[
    Marker(infoWindow: InfoWindow(title: "İstanbul Teknopark"),
      markerId: MarkerId("location-2"),
      position: LatLng(40.8739, 29.3102),
      icon: BitmapDescriptor.defaultMarkerWithHue(BitmapDescriptor.hueRed)
    ).toSet();
  ]
}

import 'dart:ui';
import 'package:cardamage/Communication.dart';
import 'package:flutter_pdfview/flutter_pdfview.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'dart:io';
import 'package:image_picker/image_picker.dart';
import 'Login.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:location/location.dart';
import 'package:path_provider/path_provider.dart';
import 'package:flutter/services.dart';
//import 'package:flutter_pytorch/flutter_pytorch.dart';
import 'dart:async';
import 'dart:ui';

class Anasayfa extends StatefulWidget {
  @override
  _AnasayfaState createState() => _AnasayfaState();
}

class _AnasayfaState extends State<Anasayfa> with TickerProviderStateMixin {
  //double get height => MediaQuery.of(context).size.height;
  File? _image;
  ImagePicker _picker = ImagePicker();
  LatLng initialLocation = LatLng(40.8739, 29.3102);
  late GoogleMapController mapController;
  Location location = Location();

  //late ModelObjectDetection _objectModel;
  //String? _imagePrediction;
  //List? prediction;

```

```

//bool objectDetection = false;
//List<ResultObjectDetection?> objDetect = [];
//bool firststate = false;
//bool message = true;

String pathPDF = "";

@override
void initState() {
  super.initState();
  fromAsset('Resimler/kazatutanak.pdf', 'kazatutanak.pdf').then((f) {
    setState(() {
      pathPDF = f.path;
    });
  });
}

Future<File> fromAsset(String asset, String filename) async {
  Completer<File> completer = Completer();
  try {
    var dir = await getApplicationDocumentsDirectory();
    File file = File("${dir.path}/${filename}");
    var data = await rootBundle.load(asset);
    var bytes = data.buffer.asUint8List();
    await file.writeAsBytes(bytes, flush: true);
    completer.complete(file);
  } catch (e) {
    throw Exception('Error parsing asset file!');
  }

  return completer.future;
}

// @override
//void initState() {
//  //super.initState();
//  //loadModel();
//}
Future<File>getFileFromAsset(String asset) async{
  try{
    var data = await rootBundle.load(asset);
    var bytes = data.buffer.asUint8List();
    var dir = await getApplicationDocumentsDirectory();
    File file = File("${dir.path}/kazatutanak.pdf");

    File assetFile = await file.writeAsBytes(bytes);
    return assetFile;
  }catch(e){
    throw Exception("Error loading pdf");
  }
}

//Future loadModel() async {
//  // String pathObjectDetectionModel = "Models/yolov5s.torchscript";
//  //try {
//    // _objectModel = await FlutterPytorch.loadObjectDetectionModel(
//    //   pathObjectDetectionModel, 80, 640, 640,
//    //   labelPath: "Labels/labels.txt");
//  // } catch (e) {
//    // if (e is PlatformException) {
//    //   print("only supported for android, Error is $e");
//    // } else {
//    //   print("Error is $e");
//    // }
//  // }
//}
//void handleTimeout() {
//  // call back function
//  // Do some work.
//  //setState(() {
//    //firststate = true;
//  // });
//}
//Timer.scheduleTimeout([int milliseconds = 10000]) =>
//  // Timer(Duration(milliseconds: milliseconds), handleTimeout);

// Future runObjectDetection(_image) async {
//  // setState(() {
//    // firststate = false;
//    //message = false;
//  //});
//  //pick an image
//  //final XFile? image = await _picker.pickImage(source: ImageSource.camera);
//  //objDetect = await _objectModel.getImagePrediction(
//    // await File(_image!.path).readAsBytes(),
//    //minimumScore: 0.1,
//    //IOUThershold: 0.3);
//  //objDetect.forEach((element) {
//    //print({
//    //  "score": element?.score,

```



```

        // "className": element?.className,
        // "class": element?.classIndex,
        // "rect": {
        //   "left": element?.rect.left,
        //   "top": element?.rect.top,
        //   "width": element?.rect.width,
        //   "height": element?.rect.height,
        //   "right": element?.rect.right,
        //   "bottom": element?.rect.bottom,
        // },
        // });
        // });
        // scheduleTimeout(5 * 1000);
        // setState(() {
        //   _image = File(_image.path);
        // });
        // }

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.blue[25],
    drawer: new Drawer(
      width: 250,
      child: new ListView(
        children: [DrawerHeader(
          child: Align(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Container(
                  width: 140,
                  height: 130,
                  child:
                    Image.asset("Resimler/cardamagel.png",
                      fit: BoxFit.fill),
                ],
              ),
            ),
          ),
        ),
        new ListTile(
          title: Text("Home"),
          trailing: new Icon(Icons.logout_outlined, color: Colors.blue),
          onTap: () {
            Navigator.of(context).push(
              MaterialPageRoute(builder: (context) => Anasayfa()));
          },
        ),
        new Divider(),
        new ListTile(
          title: Text("About Us"),
          trailing: new Icon(Icons.logout_outlined, color: Colors.blue),
          onTap: () {
            Navigator.of(context).push(
              MaterialPageRoute(builder: (context) => About()));
          },
        ),
        new Divider(),
        new ListTile(
          title: Text("Communication"),
          trailing: new Icon(Icons.logout_outlined, color: Colors.blue),
          onTap: () {
            Navigator.of(context).push(
              MaterialPageRoute(builder: (context) => Communication()));
          },
        ),
        new Divider(),
        new ListTile(
          title: Text("Log Out"),
          trailing: new Icon(Icons.logout_outlined, color: Colors.blue),
          onTap: () {
            Navigator.of(context).push(
              MaterialPageRoute(builder: (context) => Login()));
          },
        ),
      ],
    ),
    appBar: AppBar(
      centerTitle: true,
      backgroundColor: Colors.blue[300],
      title: Text("CAR-DAMAGE"),
    ),
    body: SingleChildScrollView(
      child: Column(
        children: [
          SizedBox(height: 15),
          Align(alignment: Alignment.center,
            child: FloatingActionButton.extended(
              label: Text(

```



```

        Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) => PDFScreen(path: pathPDF),
            ),
        );
    }
}

],
),
);
}

Set<Marker> _createMarker() {
    return <Marker>[
        Marker(infoWindow: InfoWindow(title: "İstanbul Teknopark"),
            markerId: MarkerId("asdasd"),
            position: LatLng(40.8739, 29.3102),
            icon: BitmapDescriptor.defaultMarkerWithHue(BitmapDescriptor.hueRed))
    ].toSet();
}

_getfromcamera() async {
    final image = await _picker.pickImage(source: ImageSource.camera);
    if (image == null) return;
    setState(() {
        _image = File(image.path);
        //_selectedImage=_image;
    });
}

getImageFileFromAssets() async {
    final byteData = await rootBundle.load("Resimler/29.jpg");
    final fileName = "Resimler/29.jpg".split('/').last;
    final file = File('${await getTemporaryDirectory().path}/${fileName}');
    await file.writeAsBytes(byteData.buffer
        .asUint8List(byteData.offsetInBytes, byteData.lengthInBytes));

    return _image=file;
}

getImageFileFromAssets1() async {
    final byteData = await rootBundle.load("Resimler/output29.jpg");
    final fileName = "Resimler/output29.jpg".split('/').last;
    final file = File('${await getTemporaryDirectory().path}/${fileName}');
    await file.writeAsBytes(byteData.buffer
        .asUint8List(byteData.offsetInBytes, byteData.lengthInBytes));

    return _image=file;
}

_getfromgallery() async {
    final image = await _picker.pickImage(source: ImageSource.gallery);
    if (image == null) return;
    setState(() {
        _image = File(image.path);
        //_selectedImage=_image;
    });
}

_content(File? _image) {
    if (_image == null) {
        return Column(
            //mainAxisAlignment: MainAxisAlignment.center,
            //crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
                Padding(
                    padding: const EdgeInsets.only(top: 30.0),
                    child: Icon(Icons.image, size: 100.0, color: Colors.grey),
                ),
                Center(
                    child: Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: Text('No Image',
                            style: TextStyle(
                                fontSize: 20.0,
                                fontWeight: FontWeight.bold,
                                color: Colors.grey),
                        ),
                    ),
                Center(
                    child: Text(
                        'Please take or select a photo for estimate of car damage',
                        style: TextStyle(color: Colors.grey),
                    ),
                ),
            ],
        );
    } else {
        //return _objectModel.renderBoxesOnImage(_image!, objDetect);
        return Image.file(_image, width: 400, height: 300);
    }
}
}

```

```

    // _detect(File? _image) {
    //   if (_image != null) {
    //     return Column(
    //       mainAxisAlignment: MainAxisAlignment.center,
    //       crossAxisAlignment: CrossAxisAlignment.stretch,
    //       children: [
    //         Padding(
    //           padding: const EdgeInsets.only(top: 10.0),
    //           child: Icon(Icons.image, size: 100.0, color: Colors.grey),
    //         ),
    //         Container(
    //           child:
    //             _objectModel.renderBoxesOnImage(_image!, objDetect),
    //         ),
    //       ],
    //     );
    //   }
    //   return _objectModel.renderBoxesOnImage(_image!, objDetect);
    // }
}

class PDFScreen extends StatefulWidget {
  final String? path;

  const PDFScreen({Key? key, this.path}) : super(key: key);

  @override
  _PDFScreenState createState() => _PDFScreenState();
}

class _PDFScreenState extends State<PDFScreen> with WidgetsBindingObserver {
  final Completer<PDFViewController> _controller =
    Completer<PDFViewController>();
  int? pages = 0;
  int? currentPage = 0;
  bool isReady = false;
  String errorMessage = '';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Document"),
        actions: <Widget>[
          IconButton(
            icon: const Icon(Icons.share),
            onPressed: () {},
          ),
        ],
      ),
      body: Stack(
        children: <Widget>[
          PDFView(
            filePath: widget.path,
            onRender: (_pages) {
              setState(() {
                pages = _pages;
                isReady = true;
              });
            },
            onError: (error) {
              setState(() {
                errorMessage = error.toString();
              });
              print(error.toString());
            },
            onPageError: (page, error) {
              setState(() {
                errorMessage = '$page: ${error.toString()}';
              });
              print('$page: ${error.toString()}');
            },
            onViewCreated: (PDFViewController pdfViewController) {
              _controller.complete(pdfViewController);
            },
          ),
          errorMessage.isEmpty
            ? !isReady
            ? const Center(
                child: CircularProgressIndicator(),
              )
            : Container()
            : Center(
                child: Text(errorMessage),
              ),
        ],
      ),
    );
  }
}

```