

Yolov5 ile Formula-1 Araç Model Marka Tanınması

**İSİM-SOYİSİM: AHMET ALPEREN ALTUNDAL
MUHAMMET TARIK YILMAZ**

15.12.2022

Seçilen konuya giriş paragrafı.....	3
Neden seçildiği ve önemi.....	3
Projenin Amacı ve Hedefleri	4
Kaynak Araştırması.....	5
Materyal ve yöntem.....	5
Kod Yapısı.....	6
Sonuçlar.....	7
Kaynak Listesi.....	25

Seçilen konuya giriş paragrafı

Büyük verilerin büyük hızlarla işlendiği çağımızda milyarlarca veriden farklı parametreler çıkararak çeşitli problemlerin çözümüne kolaylık getirmek için yapay zeka algoritmaları kullanılmaktadır. Gelişen teknolojiye en önemli hazine verilerin doğru işlenip uygun alanlarında kullanılmasıdır. Hayata geçireceğimiz projede veri ve verilerin işlenmesi büyük önem taşımaktadır. Verinin bir sonraki aşaması uygun hale getirilip faydalı bir çıktıya hazırlanmasıdır. Amaçlanan sonuç bu veriler ışığında yapay zeka algoritmaları kullanılarak en az hata ile öğrenimi gerçekleştirilmesidir. Bu sayede kullanım alanına göre yapay zeka yardımı ile daha hızlı, minimum hata ile hedeflenen alanda kolaylık sağlanması amaçlanır. Yukarıda bahsettiğimiz gibi hayatımızı kolaylaştıran yapay zeka temelli algoritmaların hayatımızdaki yeri giderek artmaktadır. Örnek verecek olursak yüz tanıma, araç tanıma, plaka tanıma, ses tanıma, sürücüsüz araç kullanma, savunma sanayii, güvenlik sanayii, sağlık sektörü ve birçok alanda kullanılmaktadır.

Günümüzde yapay zekanın yeri azımsanamayacak ölçüde fazladır. Örnek verecek olursak sosyal medya platformlarındaki yüz tanımadan kişi tespiti yanı sıra oyunlarda da yapay zeka temelli oyun içi etkenler bulunur. Projemizde Yolov5 algoritması kullanılarak araç marka tespiti yapılması hedeflenmektedir. Bu projede belirlenen yol güzergahı üzerinden kamera aracılığıyla görüntülerinin alınıp, oluşturulan algoritma sayesinde marka tespiti yapılması sağlanacaktır. Marka tespiti yapılan aracın doğru bir şekilde sınıflandırılması için roboflow üzerinden birçok araç verisinin etiketlenmesi gerekmektedir. Bunun sebebi ise yüklenen verilerin çokluğu ve farklı açılardan yüklenmiş olan veriler doğruluk oranını arttıracaktır. Bu amaçla yüklenen verilerin %50 oranından fazlası sentetik verileri içermektedir.

Araç marka tahmini projesinde kullanılması öngörülen algoritma Yolov5 algoritmasıdır. Pytorch ve opencv kütüphanelerinin kullanımı planlanmıştır. Yolov5 algoritması derin öğrenme tabanlı ve nesne takibi yapılması adına kullanabileceğimiz en optimum yöntemlerden biridir.

Biz projemizde ele aldığımız konu kapsamında gerçek zamanlı olarak belirlediğimiz 4 farklı modeldeki Formula 1 markalarını -Mercedes, Ferrari, Redbull, McLaren- tanımlarını ve bu araçları yüksek doğruluk oranlarında markalarının adları eşliğinde sürekli olarak takipte kalınmasını isteriz.

Neden secildiđi ve önemi

Bu projede eğitim amaçlı olduğundan dolayı bu projenin günlük trafik hayatımızda da kullanılması basittir fakat uzun işlemler gerektirir. Uzun işlemler tabirini açıklayacak olursak günlük yaşantımızda trafiđe çıkabilme izni olan araçların hepsinin yüzlerce hatta binlerce görüntüsünden oluşan bir veri seti hazırladığımızda yüksek doğruluk oranlı bir tanıma sistemi oluşacaktır. Oluşacak bu sistemi otonom araçlarda birçok kısmında yardımcı olacaktır.

Projenin amacı ve hedefi

Yapmış olduğumuz projemizi amaçlarına değinecek olursak eğitim ve veri edinme açısından kullanılabilirlik oranı yüksektir, trafikte seyreden araçların markalarını tespit ederek tespiti yapan otonom aracımızın olduğu yerleşim bölgelerinde hangi marka araçların kullanıldığı ve hangi dönemlerde ne sıklıkla kullanıldığı saptaması yapılabilir.

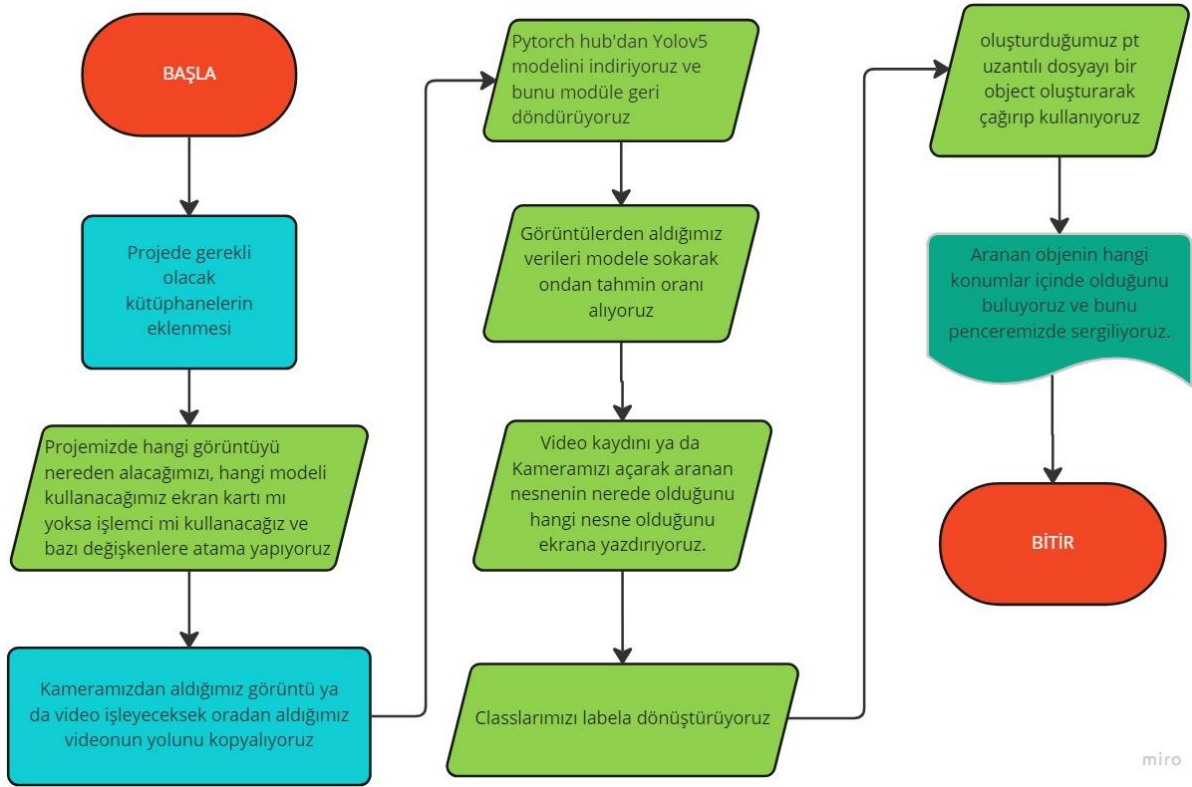
Normalde bu işlemin benzeri işlem araç satım bayilerinde mevcuttur fakat aracı alan kişiler sadece kendi bölgelerinde sabit kalmayacakları için böyle bir projede araç markalarının çoğunluk olduğu bölgeleri bilirsek o bölgelere özel olarak veri güncellemesi yapılabilir ve bu güncelleme sayesinde otonom aracımızın yapay zekâsı trafikte bölgesel yüksek tanıma oranlarının yanı sıra güvenliđi de beraberinde getirir. Biz projemizde bunu eğitim amaçlı olarak Formula 1 araçlarına uyguladık bileneceđi üzere formula araçlarının iskelet yapıları birbirlerine çok benzediđi üzere doğruluk oranları da video kalitesi, ışık durumu ve araç üstü marka stickerlarının da birer parametre olması sebebinden dolayı tanımada düşüklük olabilir ama bunu günlük trafik hayatına uygularsak araç benzerlikleri formula camiasından daha az olacağı için tanıma oranları daha yüksek olması tahmin ediyoruz.

Kaynak Arastırması

Tian-Hao Wu ve arkadaşları yolov5 algoritması kullanarak araç tespiti gerçekleştirmişlerdir. Tespit edilen araçların birbirleriyle arasındaki mesafeyi tespit ederek araçlar arasındaki olması gereken mesafeyi ilk kullanıcıya göstermektedir. Chengpeng Wang ve arkadaşları araçların tespiti yapılarak motorlu araçların egzoz tespitini gerçekleştirmişlerdir. Bu amaçla tespit edilen egzoz görüntülerinin araç gölgeleriyle karıştırılmaması için veri setini genişletmişlerdir. Tespiti yapılan egzoz görüntüleri trafikteki egzoz gazı çıkışında büyük bir veri elde etmişlerdir. Yu Zhang ve arkadaşları araç tespitinde yolov5 algoritmasını kullanmışlardır. Yapılan çalışma üzerinde doğruluk oranını arttırmak için Flip-Mosaic algoritmasını kullanmışlardır. Çalışmada gösteriyor ki kullanılan algoritma araç tespit oranını arttırmıştır. Yeting Huang ve arkadaşları oluşturdukları yapıda araçların tespit edilerek daha güvenli sürüş için kullanıcıya asistan desteği vermeyi amaçlamışlardır. Bu sayede kullanıcı sebepli kazaların önüne geçilerek daha güvenli bir sürüş hedeflenmiştir. Geliştirilen sürüş algılama sistemi trafik kazalarının azaltacak önlemler almıştır. Ali Abdulazeez Mohammedbaqer Qazzaz ve arkadaşları yaşadıkları bölgede araçların sınıflandırılmasından bahsetmişlerdir. Bu amaçla bölgedeki sınıflandırılan araçlara göre tespit gerçekleştirilmiştir. Gerçekleştirilen tespit öncelikle aracın varlığının tespiti. Ardından aracın görevi ve son olarak bulunduğu sınıfı da tespit edilerek nihai çıktıya ulaşılmıştır. Yapılan projenin sebebi ise son yıllarda artan araç sayısını yapay zeka algoritmaları sayesinde tespit ederek sınıflandırılmasının yapılmasıdır. Tübitak araştırmacılarının yaptığı çalışmada insansız hava araçları sayesinde trafik algılama ve trafik kontrolü gerçekleştirme işlemi yapılmıştır. Yapılan bu tespitte insansız hava aracının sağladığı yüksek hareket kabiliyeti sayesinde daha başarılı bir tespit ve takip gerçekleştirilecektir.

Yapılan literatür taramasında görüldüğü üzere derin öğrenme tabanlı nesne tespitinin kullanım alanı ve çeşitleri olarak çok geniş alana sahiptir. Bu doğrultuda amaç, zaman ve insan gücünden kazanç sağlanmasıdır. Özellikle trafikte araç tespiti ve trafik kontrolü alanında geliştirilen yapay zeka algoritmaları için tespit edilecek nesnenin doğru ve güvenli bir şekilde hedef noktasına teslim edilmesi gerekmektedir. Bu projede gerçekleştirilecek olan araç marka tespiti (yolov5) algoritması aynı amaç doğrultusunda tasarlanacaktır. Projenin özgün değeri literatür taramalarında örneği ile karşılaşılmayan tespiti gerçekleştirilen aracın markasının da tespiti gerçekleştirilmesidir. Bu amaç doğrultusunda trafikte araç marka tespiti CRM analitiği, markaların egzoz gazına göre değerleri bulunması, güvenli sürüş gibi alanlarda başarı sağlayacaktır. Trafik akışının yoğun olduğu alanlarda tasarlanan derin öğrenme algoritması sayesinde yoğun trafikte aracın tespiti sağlanacaktır. Bu sayede hızlı bir tespit gerçekleştirilecektir. Doğru bir şekilde tespit edilen aracın markasının doğru bir şekilde tespit edilmesi amaçlanmıştır. İnsan ve yapay zeka iş birliği içerisinde oluşturulan yapı doğrultusunda işlemler hızlı ve akıcı bir şekilde gerçekleştirilecektir. Sektöre getireceğimiz bu çözümler neticesinde insan ve yapay zeka iş birliği bir adım daha ileriye taşınacaktır.

Materyal ve yöntem

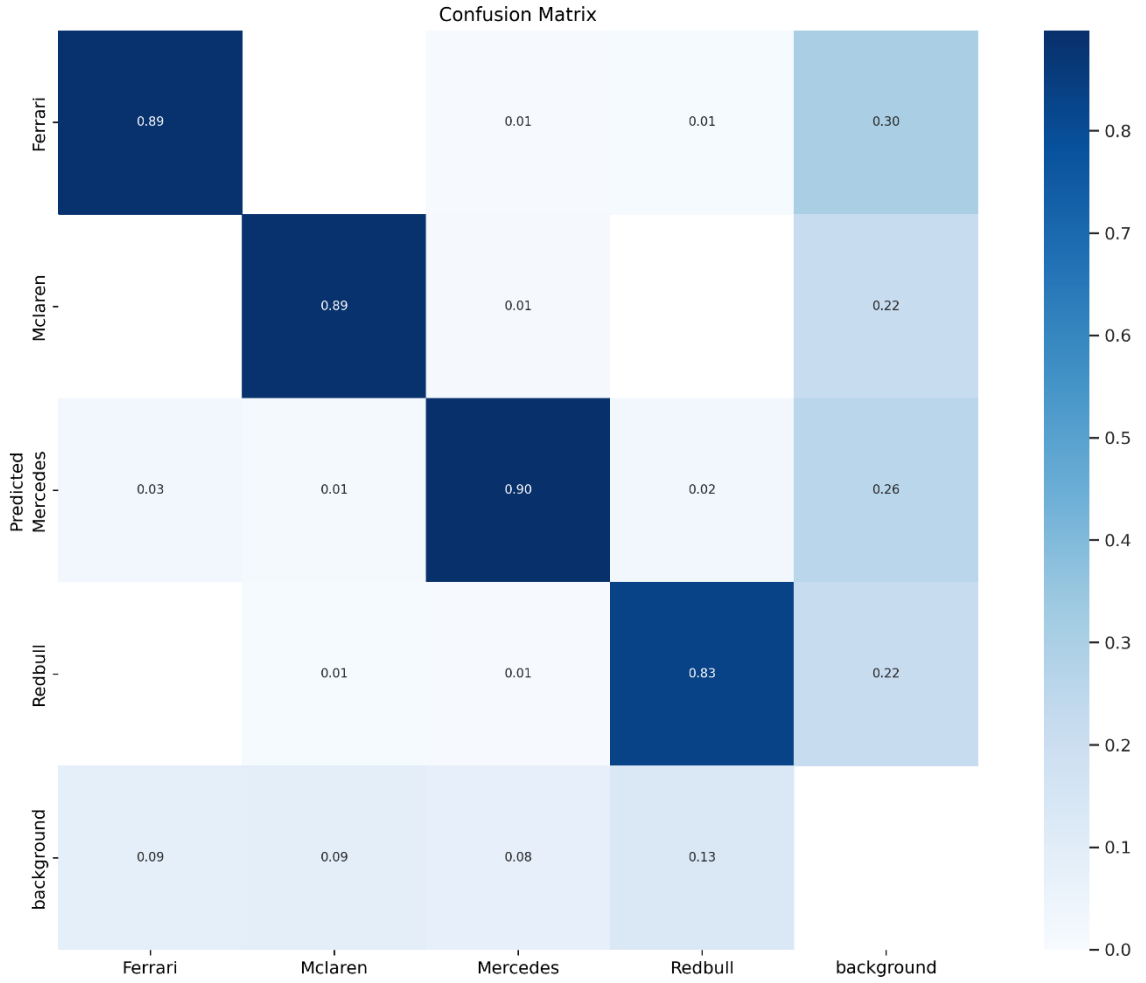


miro

Kod yapısı

```
1 import torch #pytorch kütüphanesini dahil ederek makine öğrenmesini kullanıyoruz.
2 import numpy as np #numpy kütüphanesini dahil ederek diziler üzerinde çalışmayı sağlıyoruz.
3 import cv2 #opencv kütüphanesini dahil ederek görüntüyü işlememizi sağlıyoruz.
4 from time import time #time kütüphanesini dahil ederek sayılar ve zamansal işlemler adına kullanıyoruz
5
6
7 class CustomDetector: #Class'ımızı oluşturuyoruz
8
9
10 def __init__(self, capture_index, model_name):
11
12     self.capture_index = capture_index #Hangi kamerayı kullancağıımız
13     self.model = self.load_model(model_name) #Hangi modeli kullancağıımız
14     self.classes = self.model.names
15     self.device = 'cuda' if torch.cuda.is_available() else 'cpu' #Lokalde kurmuş olduğumuz cuda sayesinde işlemlerimizi cpu yerine cpu'dan yapmayı seçiyoruz olmazsa cpu ile devamını istiyoruz.
16     print("Using Device: ", self.device) #Kullancağıımız birimi belirliyoruz
17
18 def get_video_capture(self): #Videodan görüntü alıyoruz
19
20     return cv2.VideoCapture("C:/Users/pc/vs_code/Z11_yolo_wx/f1_2_Trin.mp4 ") #Kullancağıımız videonun yolunu yazıyoruz
21
22 def load_model(self, model_name):
23
24     if model_name:
25         model = torch.hub.load('C:/Users/pc/yolov5-master', 'custom', source='local', path=model_name, force_reload=True) #Pytorch hub'dan Yolov5 modelini indiriyoruz
26     else:
27         model = torch.hub.load("C:/Users/pc/yolov5-master/data", 'custom', path='yolov5s.pt', source='local') #Bunu module geri döndürüyoruz
28     return model
29
30 def score_frame(self, frame): #Videodan aldığımız görüntüyü modele sokarak ondan tahmin oranı alıyoruz
31
32     self.model.to(self.device)
33     frame = [frame]
34     results = self.model(frame)
35     labels, cord = results.xyxy[0][:, -1], results.xyxy[0][:, :-1]
36     return labels, cord
37
38 def class_to_label(self, x):
39
40     return self.classes[int(x)] #Classlarımızı labela dönüştürüyoruz.
41
42 def plot_boxes(self, results, frame): #Aradığımız objenin hangi koordinatlar içinde olduğunu belirliyoruz.
43
44     labels, cord = results
45     n = len(labels)
46     x_shape, y_shape = frame.shape[1], frame.shape[0]
47     for i in range(n):
48         row = cord[i]
49         if row[4] >= 0.3:
50             x1, y1, x2, y2 = int(row[0]*x_shape), int(row[1]*y_shape), int(row[2]*x_shape), int(row[3]*y_shape)
51             bgr = (0, 255, 0)
52             cv2.rectangle(frame, (x1, y1), (x2, y2), bgr, 2)
53             cv2.putText(frame, self.class_to_label(labels[i]), (x1, y1), cv2.FONT_HERSHEY_SIMPLEX, 0.9, bgr, 2)
54     return frame
55
56 def __call__(self): #Videomuzu açarak aranan nesnenin nerede olduğunu hangi nesne olduğunu yazıyoruz.
57
58     cap = self.get_video_capture()
59     assert cap.isOpened()
60
61     while True:
62
63         ret, frame = cap.read()
64         assert ret
65
66         frame = cv2.resize(frame, (1280,720)) #video penceremizin boyutunu ayarlıyoruz
67
68         start_time = time() #video zamanını başlatır.
69         results = self.score_frame(frame)
70         frame = self.plot_boxes(results, frame)
71
72         end_time = time()
73         fps = 1/np.round(end_time - start_time, 2)
74
75         cv2.putText(frame, f'FPS: {int(fps)}', (20,70), cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,255,0), 2) #video ile alakalı fps,fon vs. gerekli parametreleri giriyoruz
76
77         cv2.imshow('YOLOv5 Detection', frame) #penceremizin adını yazıyoruz
78         if cv2.waitKey(5) & 0xFF == ord('q'): # pencereden çıkmak için 'q' tuşuna basılmasını giriyoruz
79             break
80
81     cap.release() #kodu serbest bırakıyoruz
82     cv2.destroyAllWindows() #çıkış isteminceyken kapatıyoruz ekranları.
83
84
85 alperen='yolov5s.pt'
86 alperen="C:/Users/pc/vs_code/Z11_yolo_wx/best3.pt"
87
88 detector = CustomDetector(capture_index=0, model_name=alperen) #burada eğitilen ağırlık dosyasını giriyoruz
89 detector()
90
91
92
```

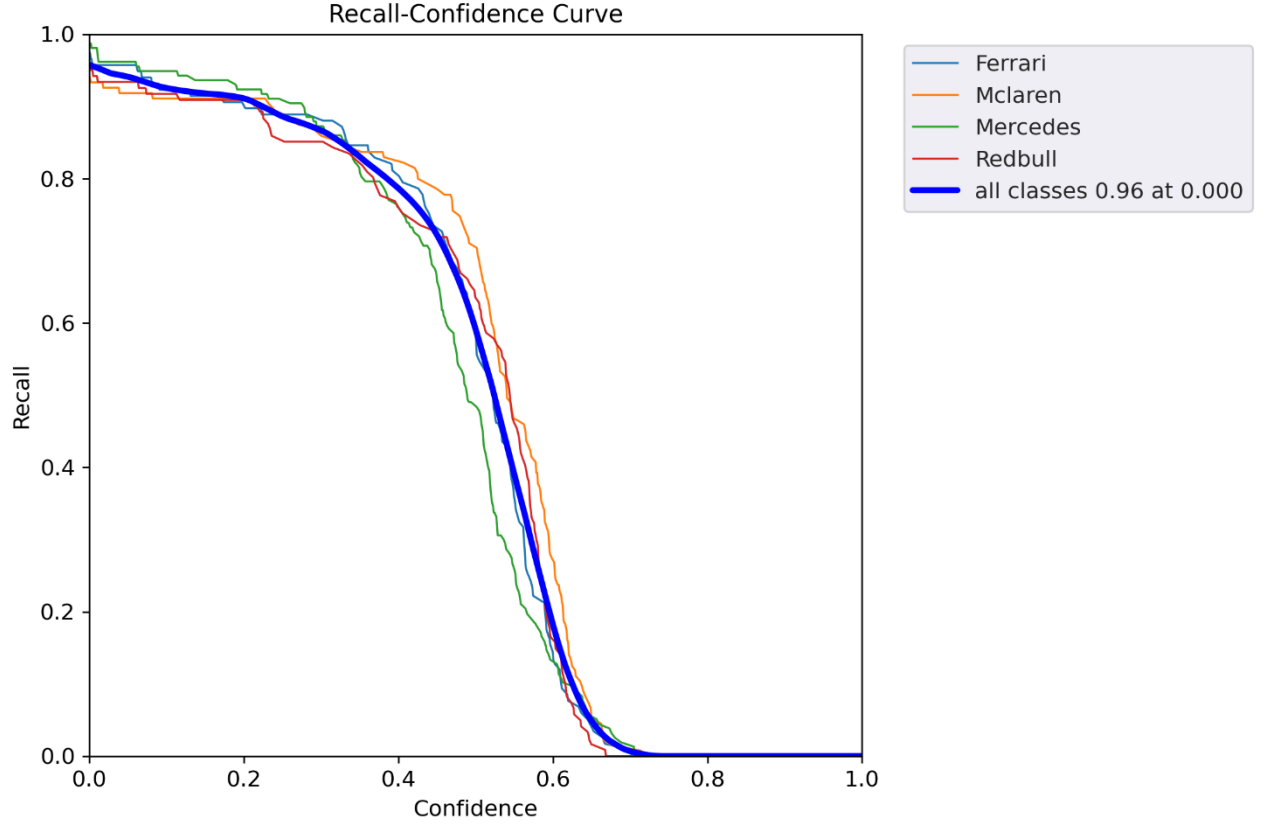
Sonuçlar



Confusion Matrix kullanarak yaptığımız classification (sınıflandırma) modelimizin performansını yorumlayabiliriz. Aynı zamanda Confusion matrix yardımı ile farklı metrik değerlerimizi de bulabilir ve model başarılarımızı yaptığımız işleme göre detaylı şekilde değerlendirebiliriz.

Matrisin her satırı, tahmin edilen bir sınıftaki örnekleri temsil ederken, her sütun gerçek bir sınıftaki örnekleri temsil eder (veya tersi). Biz dikeyde true label, yatayda predicted label aldık. Yukarıda da görüldüğü üzere modelimizin başarı oranı köşegenler matrisimizin doğruluk oranı ile tespit edilmektedir.

İsimlerin kaynağı sistemin iki sınıfı karıştırıp karıştırmadığını (yani genellikle birini diğeriyle yanlış etiketleyip etiketlemediğini) görmeyi kolaylaştırdığı gerçeğinden kaynaklanmaktadır.



ROC eğrisi (alıcı çalışma özelliği eğrisi), tüm sınıflandırma eşiklerinde bir sınıflandırma modelinin performansını gösteren grafikdir

Bu eğri iki parametreyi çizer:

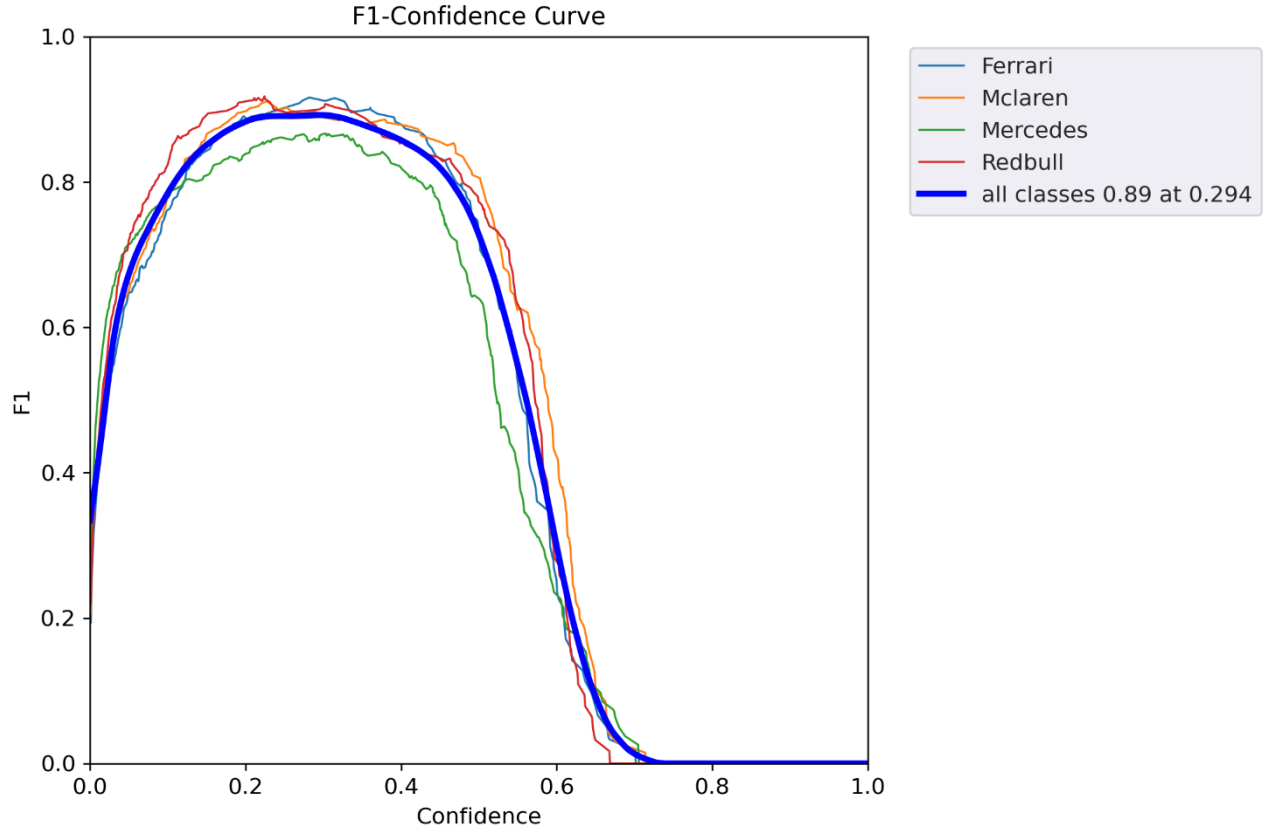
- Gerçek Pozitif Oran
- Yanlış Pozitif Oran
- **Gerçek Pozitif Oranı (TPR)**, geri çağırmanın eş anlamlısıdır ve dolayısıyla şu şekilde tanımlanır:

$$TPR = \frac{TP}{TP + FN}$$

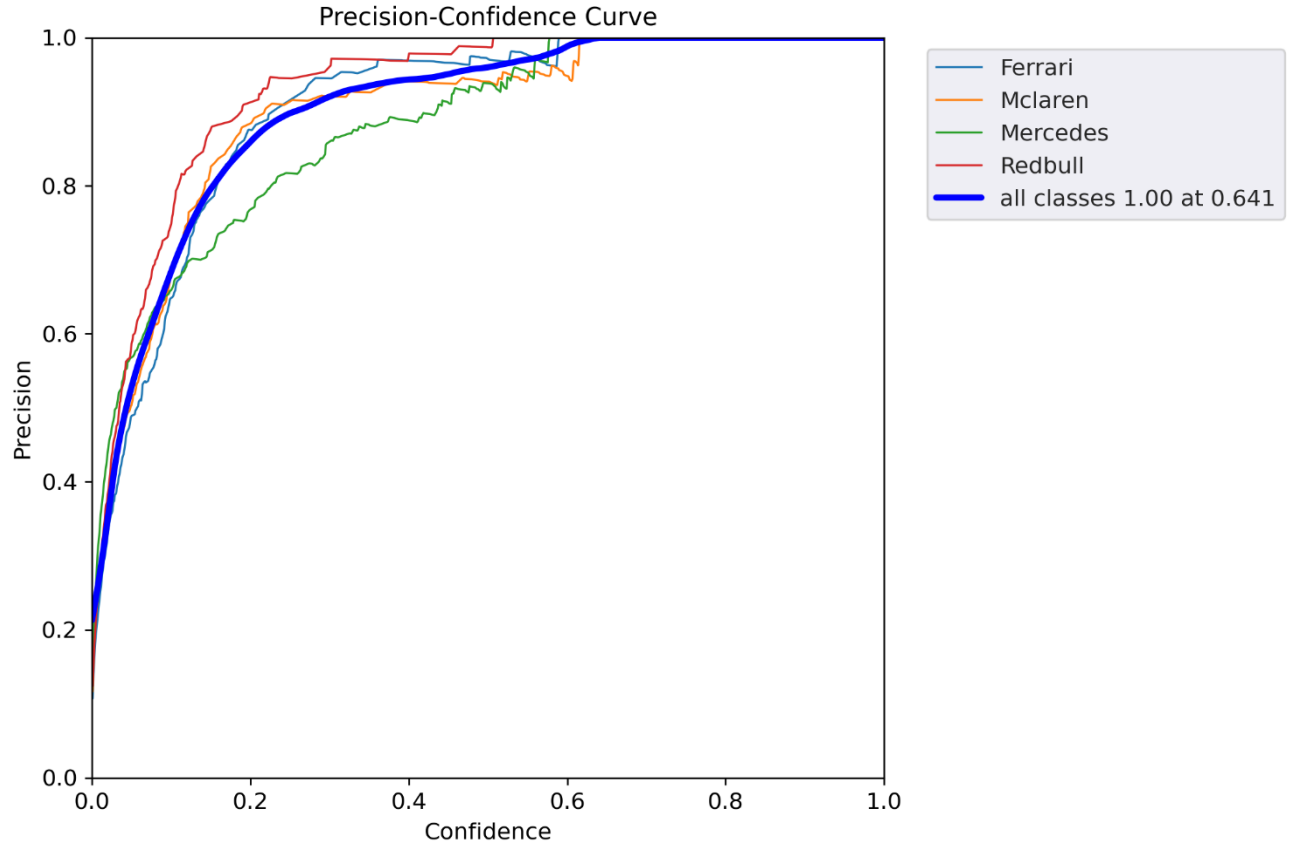
- **Hatalı Pozitif Oranı (FPR)** şu şekilde tanımlanır:

$$FPR = \frac{FP}{FP + TN}$$

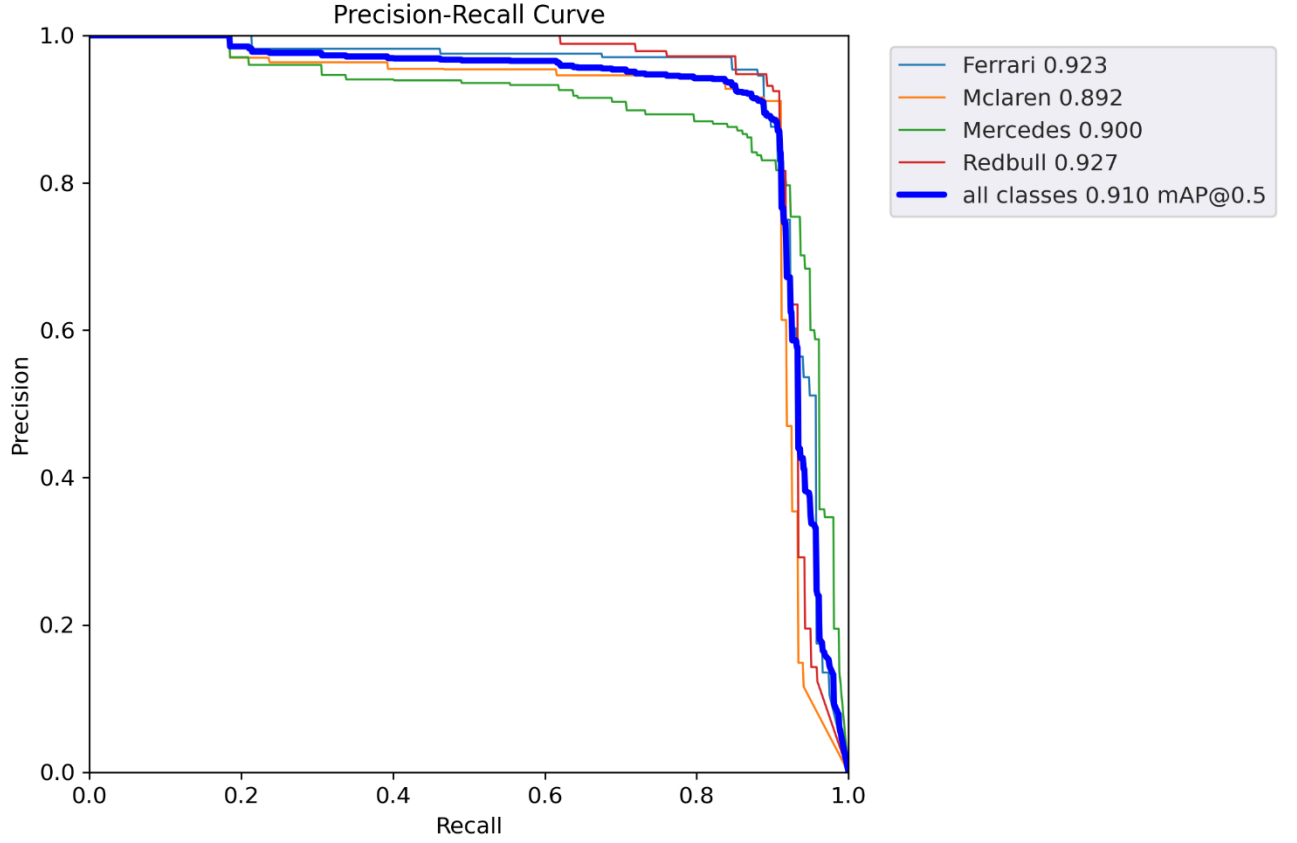
ROC eğrisi, farklı sınıflandırma eşiklerinde TPR ve FPR karşılaştırması yapar. Sınıflandırma eşikini düşürmek, daha fazla ögeyi pozitif olarak sınıflandırır. Bu da hem Yanlış Pozitif'i hem de Gerçek Pozitif'i artırır. Aşağıdaki şekilde tipik bir ROC eğrisi gösterilmektedir.



Confidence curves(güven eğrileri) son derece istatistiksel amaçlı muhteşem yaratılmışlardır. İlgilenilen bir parametre için bir güven eğrisi, nokta tahminini ve etrafındaki belirsizliği aynı anlaşılır resimde gösteren, başlı başına bir çıkarımsal özetir

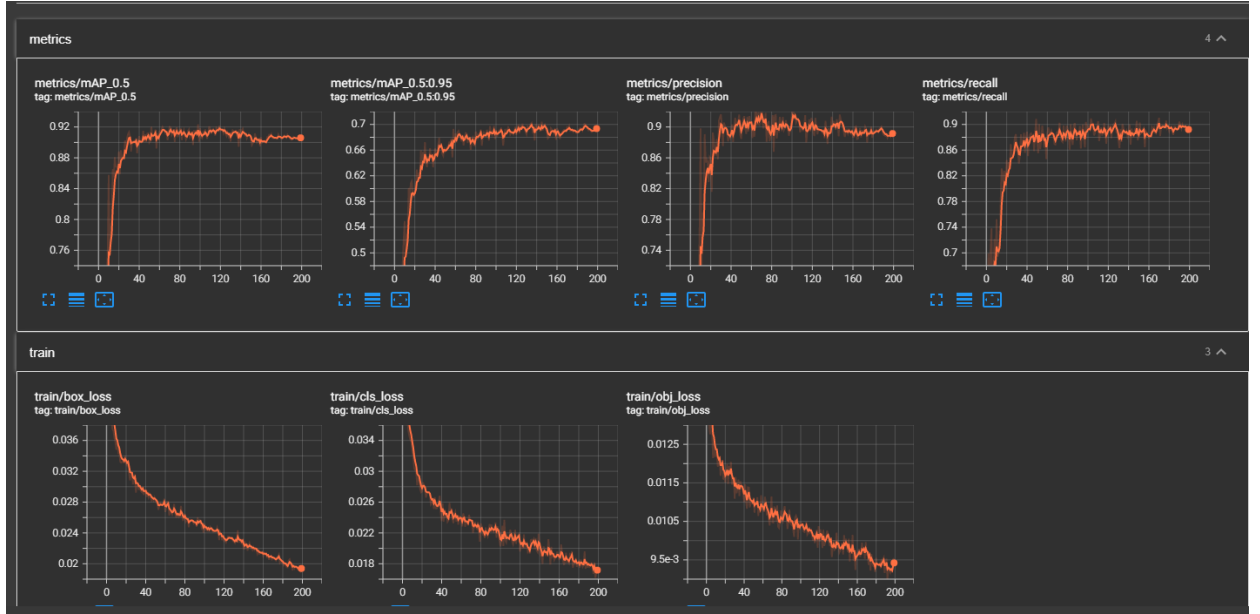


Confidence curves(güven eğrileri)'nin daha hassas ayrıntılı grafiksel halidir.



Precision confidence curve (Kesinlik-Geri Çağırma Eğrisi), sınıflar çok dengesiz olduğunda tahminin başarısının yararlı bir ölçüsüdür. Bilgi erişiminde kesinlik, sonuç alaka düzeyinin bir ölçüsüdür, geri çağırma ise kaç tane gerçekten alakalı sonucun döndürüldüğünün bir ölçüsüdür.

Precision confidence curve (Kesinlik-geri çağırma eğrisi), farklı eşikler için kesinlik ve geri çağırma arasındaki değiş tokuşu gösterir. Eğrinin altındaki yüksek bir alan hem yüksek geri çağırmayı hem de yüksek kesinliği temsil eder; burada yüksek kesinlik, düşük yanlış pozitif oranıyla ve yüksek geri çağırma, düşük yanlış negatif oranıyla ilgilidir. Her ikisi için de yüksek puanlar, sınıflandırıcının doğru sonuçlar verdiğini (yüksek kesinlik) ve ayrıca tüm olumlu sonuçların çoğunu döndürdüğünü (yüksek hatırlama) gösterir.



mAP:

Ortalama kesinlik değeri yukarıdaki grafikten okunmaktadır. Yukarıdaki örnekte görüleceği üzere $mAP = 0.9$ civarı olarak gözükmektedir. Eğitim devam ederken bu ölçüm değerine bakabilir ve duruma göre müdahale edebilirsiniz. Böylece zaman kaybını önleyebiliriz.

Precision (Kesinlik):

“Tahmin ettiğim örnekler içerisinde gerçekten kaç tanesi doğru” sorusunun cevabıdır.

$$Precision = \frac{TP}{TP + FP}$$

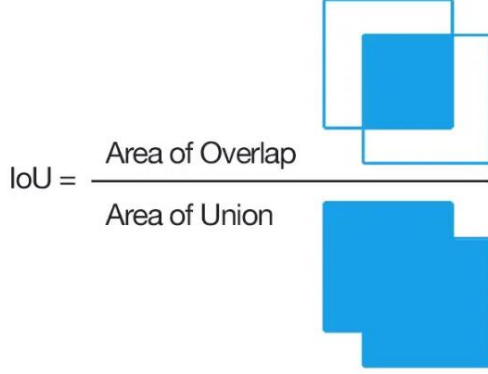
Recall (Duyarlılık):

“Pozitif sınıfa ait örneklerden kaç tanesini doğru tahmin ettim?” sorusunun cevabıdır.

$$Recall = \frac{TP}{TP + FN}$$

IoU (Intersection over union):

IoU iki sınır arasındaki örtüşmeyi ölçer. Bunu, tahmin ettiğimiz nesnenin sınırı ile gerçek nesne sınırının ne kadar örtüşeceğini ölçmek için kullanıyoruz. Bir eşik belirlenir. Mesela $IoU \geq 0.5$ 'ten büyükse tahmin doğrudur.



mAP50 nedir:

Nesne algılama için mAP, tüm sınıflar için hesaplanan AP'nin ortalamasıdır. mAP@0.5, IOU eşiği 0.5'te hesaplanan mAP olduğu anlamına gelir. Ortalama Hassasiyetin (AP) genel tanımı, hassasiyet-geri çağırma eğrisi altındaki alanı bulmaktır. Modelin kesinliğini çizme ve modelin güven eşiğinin bir fonksiyonu olarak geri çağırma işlemi, kesinlik geri çağırma eğrisidir. Bizim içinde önemli olan ele almamız gereken değer kriter budur.

```
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
```

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	501	530	0.905	0.88	0.91	0.705
Ferrari	501	117	0.929	0.889	0.923	0.739
McLaren	501	135	0.915	0.876	0.892	0.683
Mercedes	501	157	0.828	0.904	0.9	0.704
Redbull	501	121	0.949	0.851	0.927	0.694

Results saved to runs/train/exp

Veri Kümesi Kontrolü

Resim Sayısı:

➤ 2496

- 0 missing annotations
- 169 null examples

Ortalama Resim Boyutu:

➤ 2.07 mp(MegaPiksel)

- 0.01Mp'den
- 12.13Mp'ye kadar

Medyan Görüntü oranı:

➤ 1920x1080p

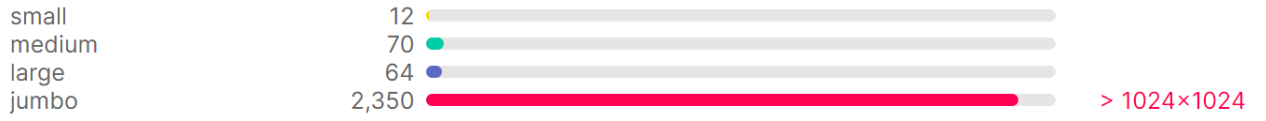
Etiketlenen Sınıflar Arası Denge:

Sınıf Dengesi, her nesneden kaç tane olduğunu gösterir ve sınıf dengesini/dengesizliğini kolayca görselleştirir

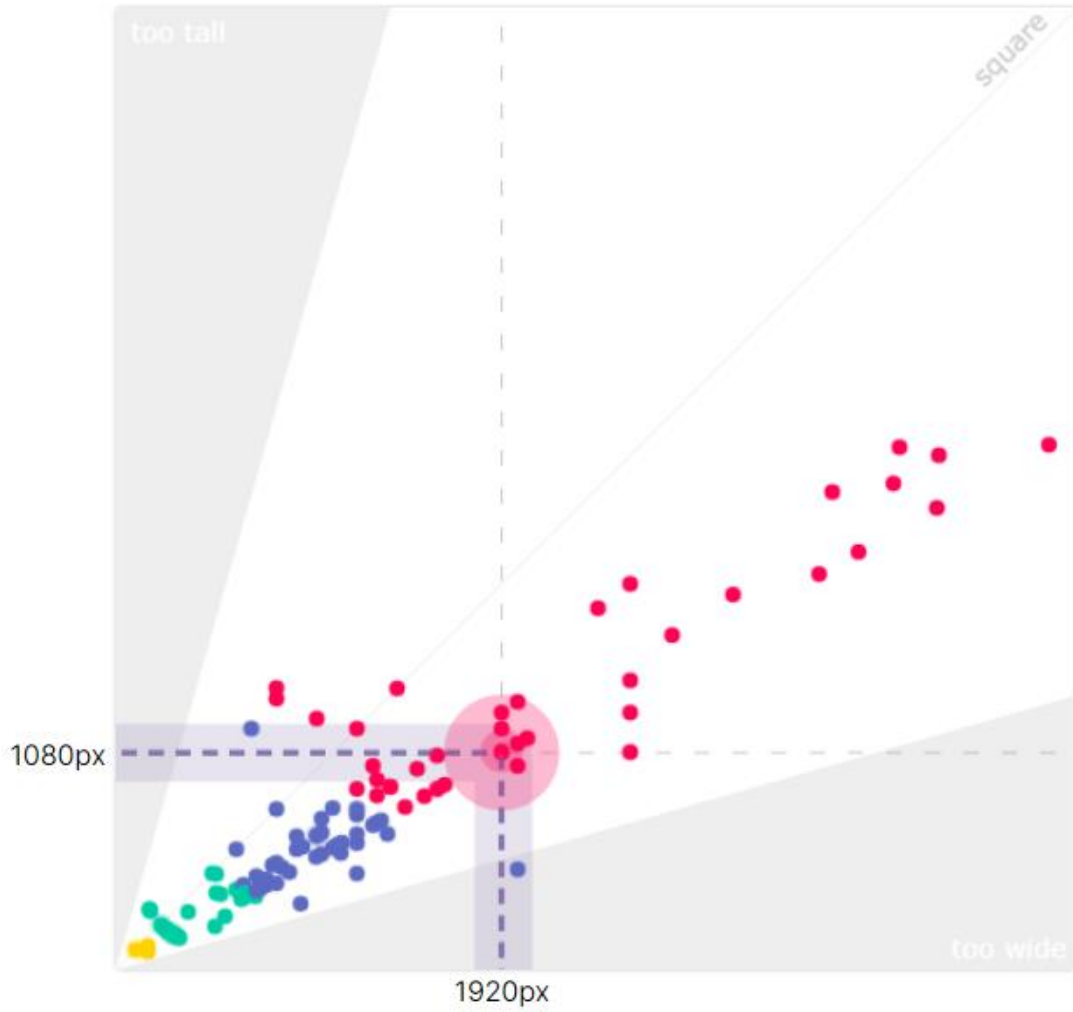
Class Balance



Etiketlenen Sınıflar Arası Denge:



Burada boyut bilgisi alıyoruz, boyut bilgisi önemlidir çünkü boyut değişiklikleri hakkında bilgi vermede yardımcıdır, eğer biz resimlerimizin boyutlarını değiştirirsek çoğu modelimizin aşağı ya da yukarı sıkışmaları gerekecektir. Sıkışmadan kastettiğimiz basitçe söylemek gerekirse, germe, verilerinizi ölçeklendirme anlamına gelir. Medyan resim ortalamasında da belirttiğimiz gibi resimlerimizin ortalaması 1920x1080'dir. tabiki de bazıları bundan büyük, küçük olabilir



En Boy Dağılım Oranı:

Aspect Ratio Distribution



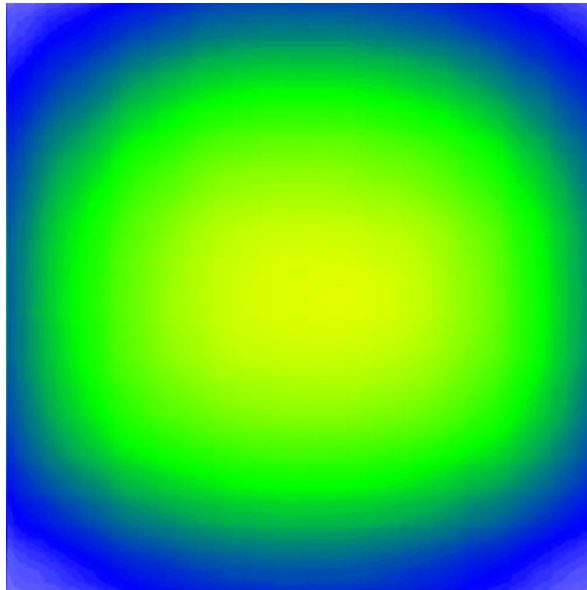
Isı Haritaları:

Bir ısı haritası (veya ısı haritası), bir olgunun büyüklüğünü iki boyutta renk olarak gösteren bir veri görselleştirme tekniğidir. Renkteki varyasyon, ton veya yoğunluğa göre olabilir, okuyucuya fenomenin nasıl kümelendiğine veya uzayda nasıl değiştiğine dair bariz görsel ipuçları veriyor.

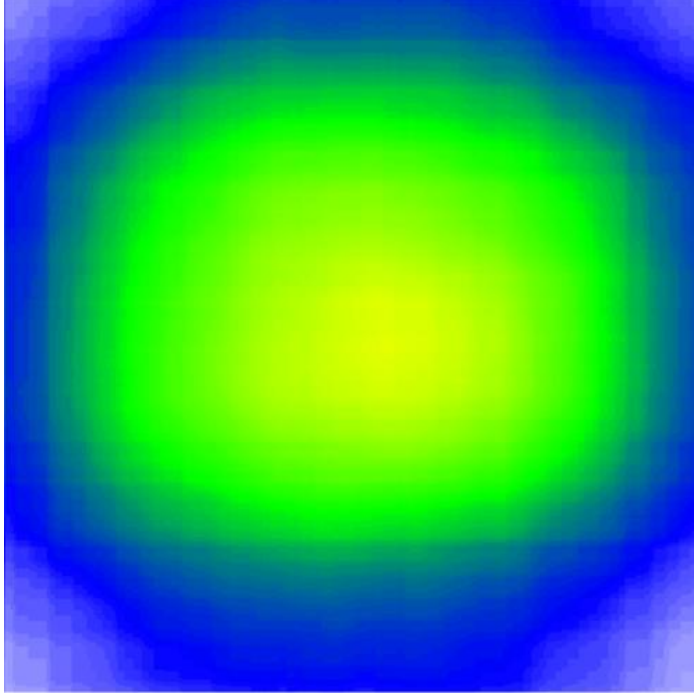
Isı haritalarımızın birbirlerine benzemesinin sebebi merak edilecek olursa verileri etiketleme sırasında etiketlediğimiz görseller içerisindeki Formula 1 araçlarının konumsal pozisyonlarından kaynaklanmaktadır.

Eğer 2500 kadar olan veri setimizde hepsini poligonal olarak etiketleme yapsaydık bu daha uzun bir süreç almasının yanı sıra ısı haritasında da daha gözle görülür şekilde ve renksel değişimlerin gözlenmesi söz konusu olabilirdi.

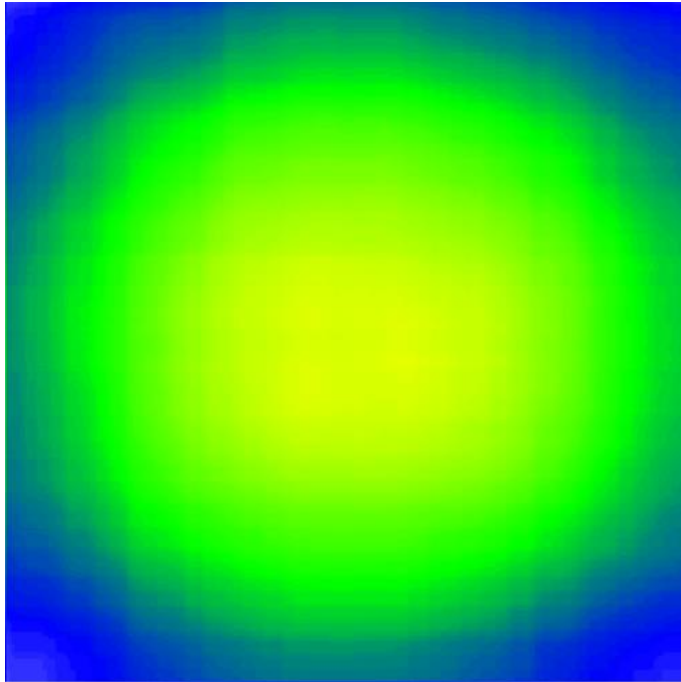
- Tüm Sınıfların;



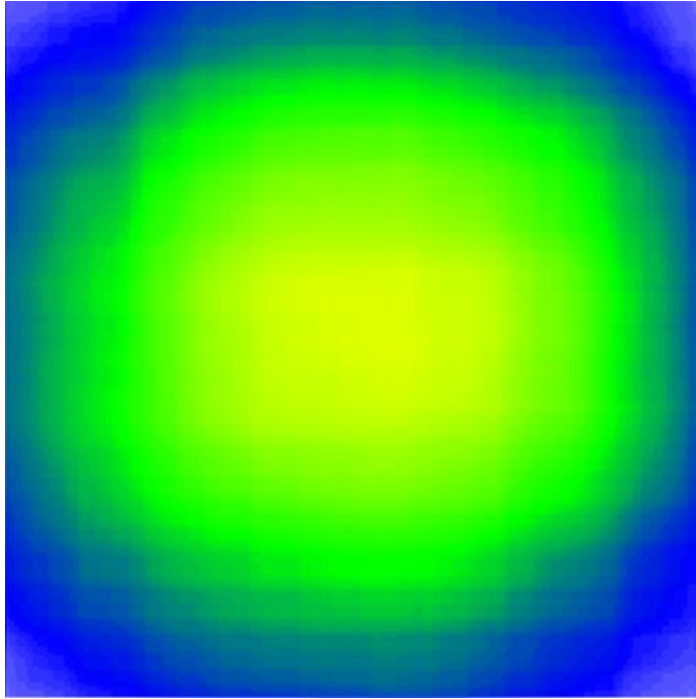
- Redbull(572 görsel);



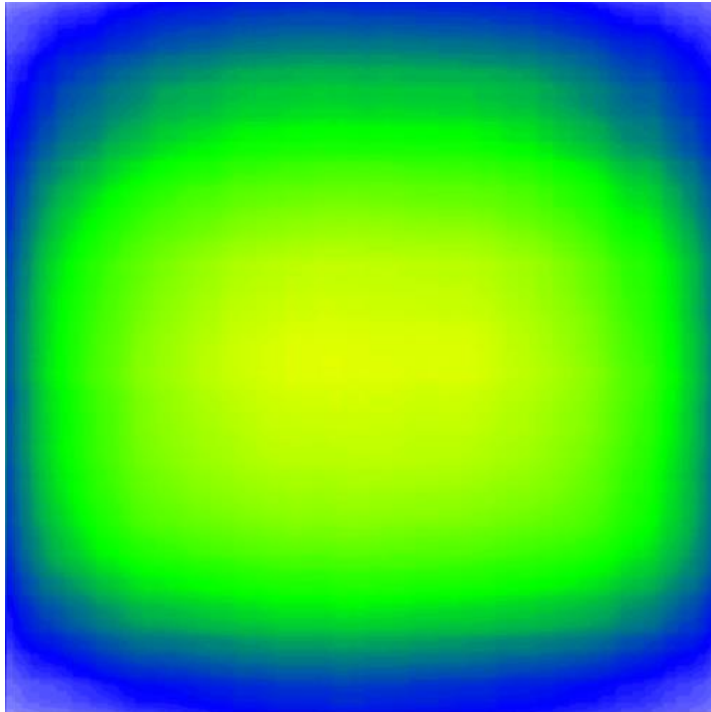
- Ferrari (582 Görsel);



- McLaren (610 Görssel);

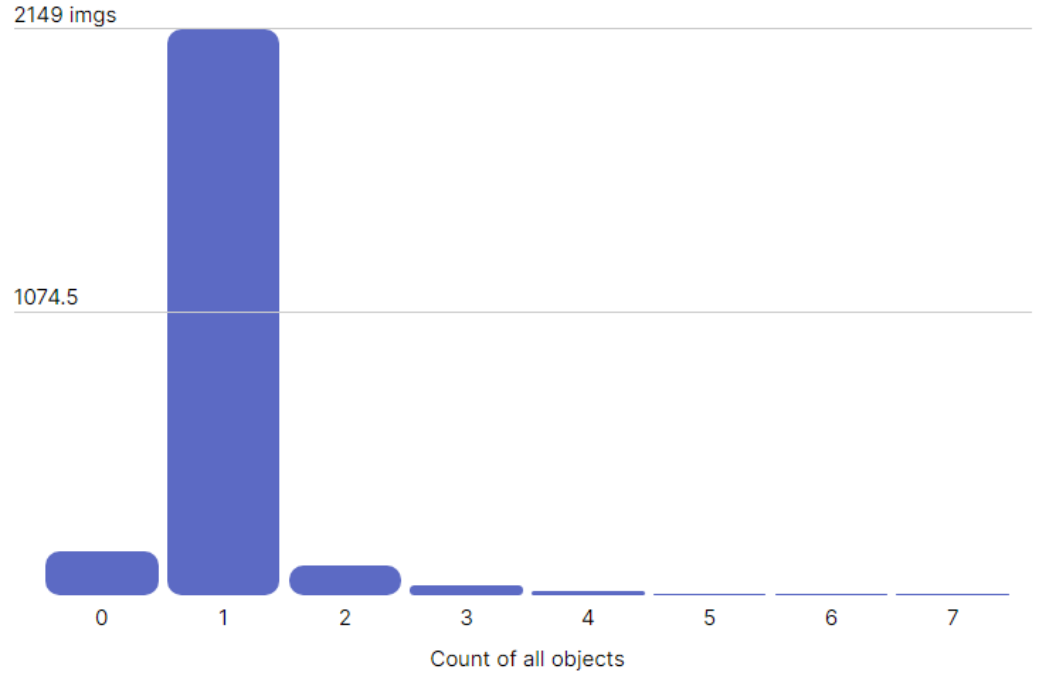


- Mercedes (850 Görssel);

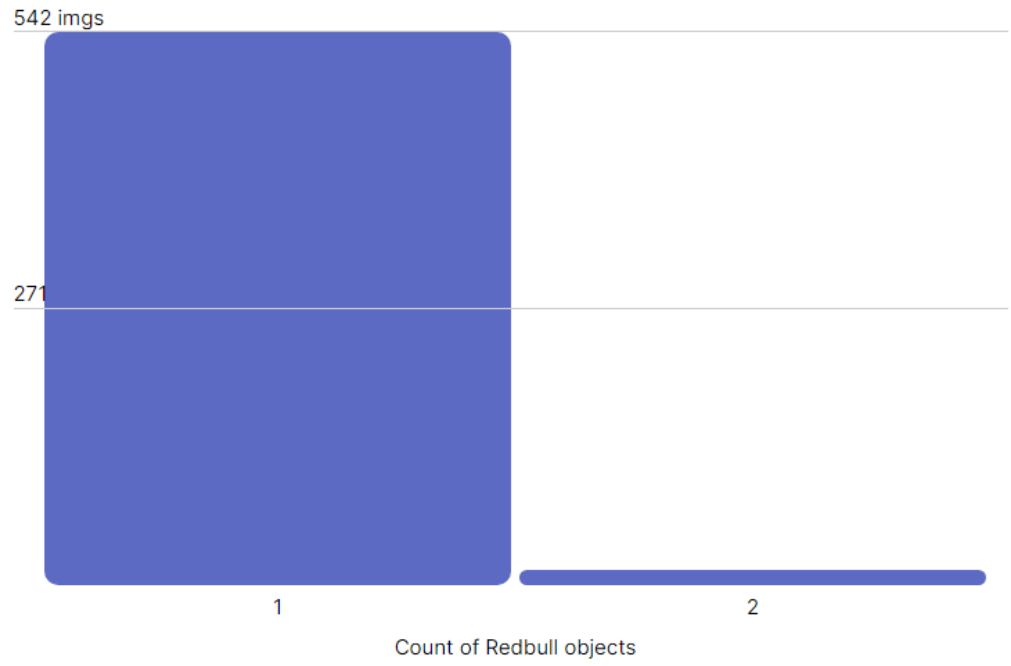


Görüntüye Göre Nesne Sayımı Histogramı

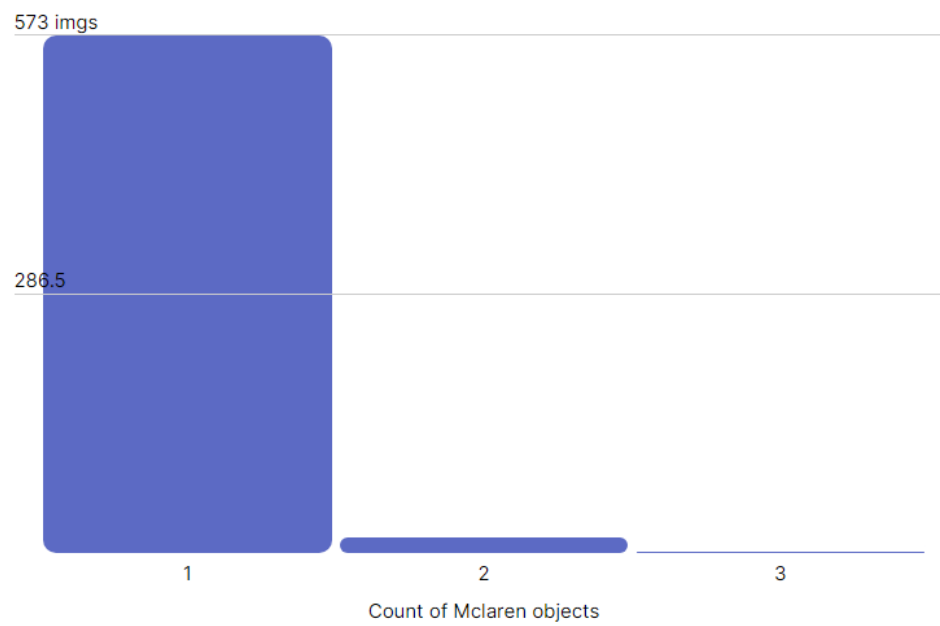
- Tüm Sınıfların;



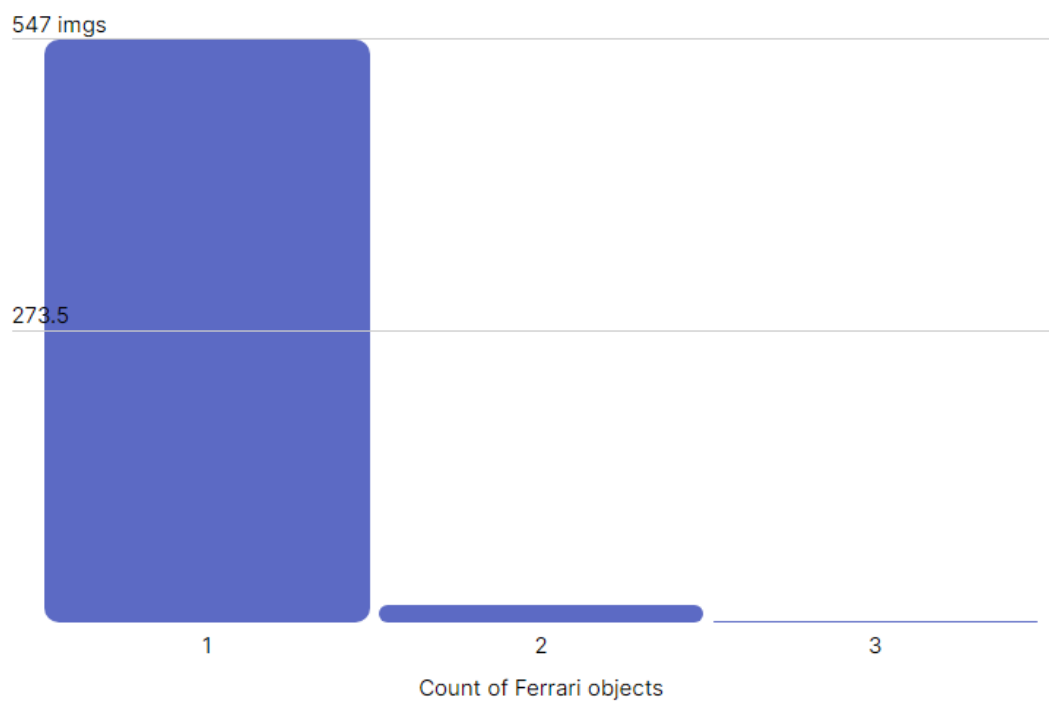
- Redbull;



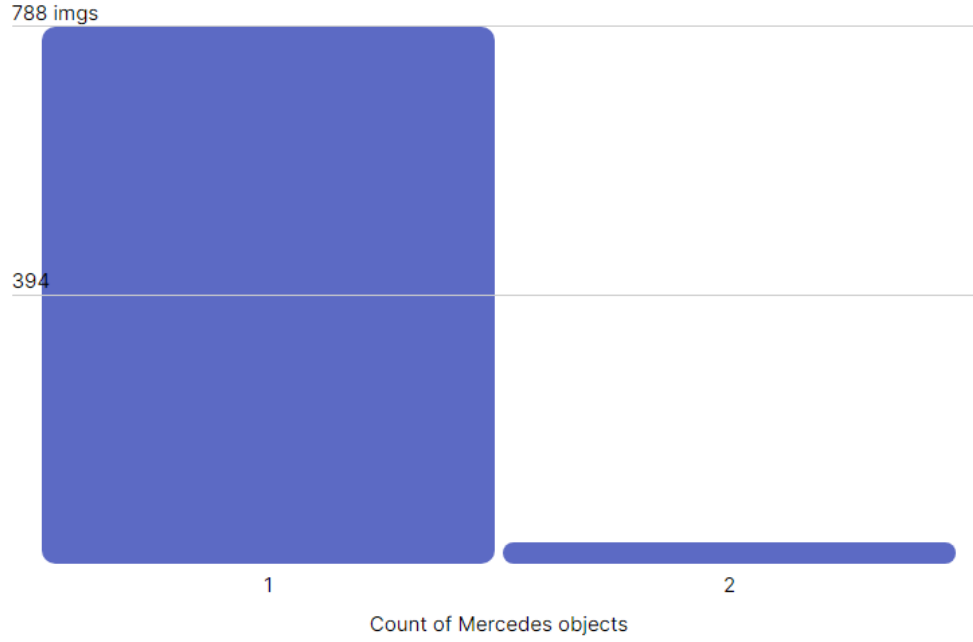
- McLaren;



- Ferrari;



- Mercedes;



Projenin Dünyaya ve bize olan özel katkısı

Hayata geçirmeyi planladığımız projede fayda sağlamak istediğimiz nokta, akan trafik üzerinde geçen araçların markalarının tespit edilerek gözlem altında tutulmasıdır. Bunun sebebi ise akan trafikte bulunan araçların oransal olarak hangi markaya ait olduğu yani bir nevi CRM (customer relationship management) müşteri ilişkiler yöntemi analizi yapmaktır.

Bu sayede gerçek zamanlı olarak markaları, kazaları ve kurlsız davranışları takip ederek bir müşteri analizi kolaylığı sağlamaktadır.

Projenin başarılı bir şekilde hayata geçirilmesinde dikkat edilmesi gereken hususlar vardır. Örnek verecek olursak trafikte hareket eden aracın marka tespiti doğruluğunu yüksek olması için kullanılacak veri setinin yeterli sayıda olması gerekmektedir.

Bir sonraki dikkat edilmesi gereken noktalardan bir tanesi de kullanılacak olan kameranın konumu, ortamdaki doğru ışık pozisyonu önemlidir. Çünkü kamera üzerinde yapılacak olan algılamada kameranın başarılı şekilde tespitini yapması gerekmektedir. Kullanılacak olan algoritmanın optimizasyon parametrelerinin doğru bir şekilde seçilmesi sonucunda doğruluk oranını arttırmasını sağlayacaktır.

Nerelerde kullanılabilir ve ülkemizde entegre edilebilme durumu.

Bildiğimiz üzere her ülkenin kendi iklim koşulları, kendi araç seçme farklılıkları sürüşte olan teknik farklı gereklilikler nedeniyle alınan araçlarda farklılık gösterebilir. Mesela bunları birer istatistikle örnekleyecek olursak;

IN NEARLY EVERY COUNTRY



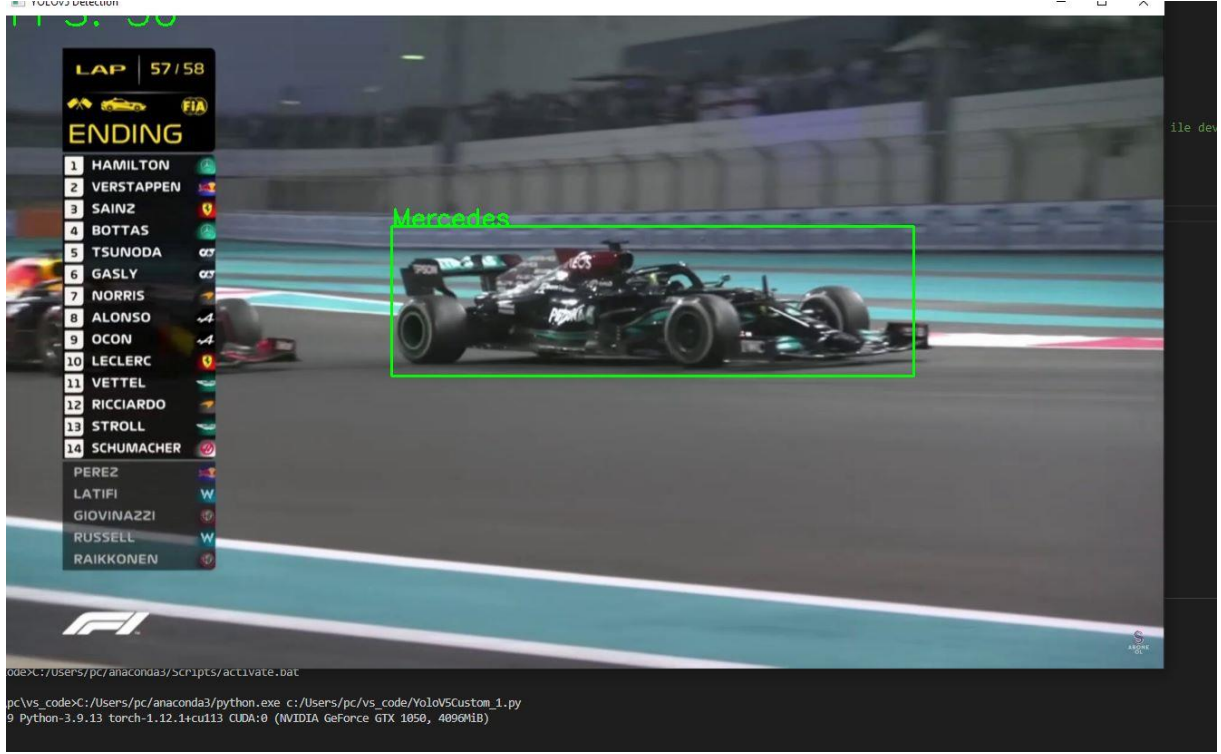
Bu yüzden bizce ülkemizde kullanımı olmayan, kullanımı çok az olan, kullanımı yasak olan araçlardan daha çok tercih edilen, her bölgede alınan bayilik verileri ile topladığımız araç markalarının verilerinin sayısını daha fazla tutarak her ülkede bunun gibi entegrasyonel değişiklikler yapılabilir.

Dikkat çekmek istediğim bir husus var, bu konu ülkemizde x model Araç 'tan hiç yok o zaman onun verilerini çok çok az tutalım, onun verilerinin hiç almayalım değil. Çok olan araç modellerinin (en çok satanlardan bahsediyoruz)

bunları daha fazla verisi ile yüksek doğruluk oranına sahip oluruz böylelikle ülkesel uygulama yapmış oluruz.

Çıktısını aldığımız görüntüler

Lokalden:



Colab:





Kaynakça

- Huang, Y. (2021). A Safety Vehicle Detection Mechanism Based. *2021 IEEE 6th International Conference on Smart Cloud*. Hangzhou, China.
- Qazzaz, A. P. (2022). Car Detection and Features Identification Based on.
- Tübitak Bilim Araştırma Vakfı. (2020). DERİN ÖĞRENME YÖNTEMLERİ KULLANARAK GERÇEK. tübitak .
- Wang, C. (2021). A High-Precision Fast Smoky Vehicle Detection. *2021 IEEE International Conference on Artificial Intelligence and Industrial Design* . China .
- Wu, T.-H. (2021). Real-Time Vehicle and Distance Detection Based on. *2021 the 3rd World Symposium on Artificial Intelligence*. China.
- Zhang, Y. (2022). Real-Time Vehicle Detection Based on Improved YOLO v5. *Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University*., China.