

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



YARI GÜDÜMLÜ VERİ ETİKETLEME ARACI

16011038 – TALHA BACAK
16011706 – DUYGU ERDURAN

BİLGİSAYAR PROJESİ

Danışman
Dr. Öğr. Üyesi Hamza Osman İLHAN

Temmuz, 2021

TEŞEKKÜR

Bu projeyi tamamlamamızda tüm sabrı ve özverisiyle konu ile ilgili bilgi ve tecrübesini bizimle paylaşarak bizi her aşamada yönlendiren Danışmanımız Dr. Öğr. Üyesi Hamza Osman İlhan'a en derin şükranlarımızı sunarız.

Çalışmalarımız boyunca maddi manevi destekleriyle bizi hiçbir zaman yalnız bırakmayan, bize güzel bir çalışma ortamı sağlamaya çabalayan en büyük motivasyonumuz olan ailemize ve arkadaşlarımıza da sonsuz teşekkür ederiz.

TALHA BACAK
DUYGU ERDURAN

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
ÖZET	viii
ABSTRACT	ix
1 Giriş	1
2 Ön İnceleme	2
3 Fizibilite	4
3.1 Teknik Fizibilite	4
3.1.1 Yazılım Fizibilitesi	4
3.1.2 Donanım Fizibilitesi	4
3.2 İletişim Fizibilitesi	5
3.3 İşgücü ve Zaman Planlaması	5
3.4 Ekonomik Fizibilite	6
3.5 Yasal Fizibilite	6
4 Sistem Analizi	8
4.1 Projenin Amacı	9
4.2 Taslak Diyagramı	9
4.3 Veri Akış Diyagramı	9
4.4 Use Case Diyagramı	9
5 Sistem Tasarımı	11
5.1 Yazılım Tasarımı	11
5.1.1 Fotoğraf Seçimi	11
5.1.2 Manuel Etiketleme	11
5.1.3 Etiket Bilgilerinin Import Edilmesi	12

5.1.4	Model ile Etiketleme	12
5.1.5	Fotoğrafı Resize Etme	12
5.1.6	Etiket Bilgilerinin Kaydedilmesi	12
5.2	Veri Tabanı Tasarımı	12
5.3	Girdi-Çıktı Tasarımı	12
6	Uygulama	16
6.1	Uygulamanın Temel Yapısı	16
6.2	MAIN MENU	18
6.2.1	Photo	18
6.2.2	Manuel	18
6.2.3	Autonomous-Run	18
6.2.4	Custom Model-Import	19
6.2.5	Load	19
6.2.6	Save	20
6.3	LABEL	20
6.3.1	Label Name	20
6.3.2	Etiketlere Müdahale	21
7	Deneyisel Sonuçlar	23
8	Performans Analizi	24
8.1	Deneyisel Sonuçların Değerlendirilmesi	24
9	Sonuç	26
	Referanslar	28
	Özgeçmiş	29

KISALTMA LİSTESİ

BOW	Bag Of Words
CNN	Convolutional Neural Networks
GTSDDB	German Traffic Sign Detection Benchmark
GUI	Graphical User Interface
İHA	İnsansız Hava Aracı
LOOCV	Leave One-Out Cross Validation
R-CNN	Region-Convolutional Neural Networks
RESNET	Residual Neural Network
sn	saniye
TLS	Trafik Levha Sınıflandırması
TLT	Trafik Levhası Tespiti
VAD	Veri Akış Diyagramı
YOLO	You Only Look Once
fMRI	Functional Magnetic Resonance Imaging

ŞEKİL LİSTESİ

Şekil 3.1	Gantt Diyagramı	5
Şekil 3.2	Organizasyon Şeması	6
Şekil 4.1	YOLOv5 modellerinin performans değerleri [6]	8
Şekil 4.2	Taslak Diyagramı	9
Şekil 4.3	Veri Akış Diyagramı	10
Şekil 4.4	Use-Case Diyagramı	10
Şekil 5.1	State diyagramı	11
Şekil 5.2	PyQt5 ile oluşturulan kullanıcı ara yüzü	13
Şekil 5.3	Girdinin Alınması (Fotoğraf)	13
Şekil 5.4	Girdi olarak alınan fotoğraf	14
Şekil 5.5	Çıktı olarak YOLO formatında kaydedilen txt dosyası	14
Şekil 5.6	Çıktı olarak Pascal formatında kaydedilen xml dosyasının bir kısmı	15
Şekil 6.1	MAIN MENU	16
Şekil 6.2	LABEL	17
Şekil 6.3	Photo Label	17
Şekil 6.4	Message Label	17
Şekil 6.5	Photo	18
Şekil 6.6	Manuel	18
Şekil 6.7	Autonomous-Run	19
Şekil 6.8	Kullanılabilecek Modeller	19
Şekil 6.9	Autonomous-Run	19
Şekil 6.10	Modeller için gerekli dosyaların import edilmesi	19
Şekil 6.11	Load	20
Şekil 6.12	Save	20
Şekil 6.13	Label Name Grubu	20
Şekil 6.14	Etiketlere müdahale edebilecek butonlar	21
Şekil 6.15	Etiket Tablosu	21
Şekil 6.16	Etiket isim seçimi	21
Şekil 6.17	Etiketlenmiş fotoğraf	22

TABLO LİSTESİ

Tablo 3.1	Donanım Fizibilitesi	5
Tablo 6.1	Kütüphane bilgileri	16
Tablo 7.1	Fotoğrafların etiket bilgileri	23
Tablo 7.2	Etiketleme süre bilgileri	23

YARI GÜDÜMLÜ VERİ ETİKETLEME ARACI

TALHA BACAK
DUYGU ERDURAN

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Dr. Öğr. Üyesi Hamza Osman İLHAN

Proje esas olarak; YOLO modelleri üzerinde manuel etiketleme ve otonom etiketleme sağlayan windows işletim sistemin üzerinde çalışan yarı güdümlü veri etiketleme aracı gerçekleştirilmesidir. Manuel olarak kullanıcı el ile dikdörtgen şeklinde etiketleme yapılabilirken; otonom etiketlemede ise, herhangi bir YOLO modeli import edebilir ya da uygulamadaki hazır modeller kullanılabilir. Hazır modellerde COCO veri setiyle eğitilmiş YOLOv5s ve YOLOv4-tiny esas alınmıştır.

Etiketleme yöntemlerinin etiketleme süre bilgileri karşılaştırılmıştır.

Anahtar Kelimeler:

Otonom etiketleme, Manuel etiketleme, COCO veri seti, YOLOv5s, YOLOv4-tiny

ABSTRACT

SEMI-GUIDED DATA ANNOTATION TOOL

TALHA BACAK
DUYGU ERDURAN

Department of Computer Engineering
Computer Project

Advisor: Assist. Prof. Dr. Hamza Osman İLHAN

In the project, It is the implementation of a semi-guided data labeling tool running on the windows operating system that provides manual labeling and autonomous labeling on YOLO models. Rectangular labeling can be done manually by the user; In autonomous labeling, any YOLO model can be imported or ready-made models in the application can be used. Ready models are based on YOLOv5s and YOLOv4-tiny trained with the COCO dataset.

Labeling time information of labeling methods were compared.

Keywords:

Autonomous labeling, Manual labeling, COCO dataset, YOLOv5s, YOLOv4-tiny

1 Giriş

Günümüzde yaygınlaşan derin öğrenme algoritmalarının, görüntü ve videolarda nesne tespiti tanıma uygulamalarında kullanılması yaygınlaşmıştır. Nesne tespit ve tanıma uygulamaları son yıllarda güvenlik, savunma, doğal afetler (sel, deprem ve yangın vb.), sağlık (salgınların yayılımının önlenmesi vb.), tarım, ormancılık alanlarında birçok problemlere çözüm bulmaktadır.

Nesne tespit, tanıma uygulamalarında oldukça yaygın kullanılan metodlardan Konvolüsyonel Sinir Ağları (CNN) algoritmaları gelmektedir. Popüler olan CNN algoritmalarından YOLO, "You Only Look Once" yani "Sadece Bir Kez Bak" Algoritma adının neden bu olduğunun cevabı ise; nesne tespitini tek seferde yapabilecek kadar hızlı olmasıdır. YOLO algoritması çalışmaya başladığında görüntülerdeki veya videolardaki nesneleri ve bu nesnelerin koordinatlarını aynı anda tespit eder.

Bu projede derin öğrenme modelleri kullanarak nesne tespiti yapıp ve nesnelerin etiketlenmesi amaçlanarak Windows işletim sistemi üzerinde çalışan GUI kullanılarak yarı güdümlü veri etiketleme aracı yapılması istenmektedir. Veri etiketleme 2 ana yöntemden oluşturulacak şekilde tasarlanmaktadır. Biri manuel etiketleme, diğeri ise otonom etiketlemedir. Manuel etiketlemede mouse ile etiketleme yapılabilir. Otonom etiketlemede ise COCO data setiyle eğitilmiş YOLOv5s, YOLOv4-tiny modelleri kullanılabilir ya da başka YOLO modelleri import edilebilir halde olması hedeflenmektedir. Oluşturulan etiketler YOLO formatında txt dosyasına ya da Pascal formatında xml dosyasına kaydedilebilir. Bu veriler YOLO formatında txt dosyalarından ya da Pascal formatında xml dosyalarından import edilebilir. Bütün YOLO modelleri kullanılabilir ve import edilebilecek biçimde uygulamanın kapsamı da genişletilebilir olması amaçlanmaktadır. Farklı özellikli etiketlemelerde harcanan zaman bilgisi karşılaştırılarak etiketlemede hangi yöntemin ne derece avantajlı olduğunun saptanması beklenmektedir.

2 Ön İnceleme

Son yıllarda, derin öğrenme (Deep Learning) ile nesne tespiti (Object Detection) projelerinin başarıları giderek artmıştır. Bunun en önemli nedeni günümüzde derin öğrenme ağlarının (Deep Learning Network) başarı oranının artması ve veri toplamanın, yaygınlaşan sosyal medya kullanımıyla beraber daha kolay bir hale gelmesi örnek olabilir. Bu bölümde bizim projemize benzer projeleri inceleyip, kullandıkları yapıları, veri kümeleri, başarı oranlarını ele alacağız. Metod olarak Faster R-CNN, YOLO üzerinde durularak Coco veri seti kullanılan ve yarı güdümlü etiketleme araçları baz alınarak literatür taraması gerçekleştirilmiştir. Bu kapsamda öne çıkan çalışmalar aşağıda özetlenmiştir.

İlk olarak projemize çok benzeyen derin öğrenme yöntemleri kullanarak gerçek zamanlı araç tespit uygulamasını inceledik. Uygulama YOLO Algoritması yardımıyla geliştirilmiştir. YOLO algoritmasının performansı, konvolüsyonel sinir ağı yapıları yardımıyla arttırılmaya çalışılmıştır. 50m ve 100m'den çekilen 500 adet fotoğraf kümesinden oluşan veriyi Faster R-CNN ile YOLO 'yu karşılaştırmışlardır. Faster R-CNN ve YOLO algoritmalarının her ikisi de GeForce 1060 GPU Ekran kartı ve 16GB Ram'e sahip bilgisayarlar üzerinde test edilmiştir. Karşılaştırma sonucunda skor olarak YOLO Algoritması daha başarılı olmuştur. Çalışılan testler sonunda YOLO algoritmasının tespit oranını % 4,3 arttırdığı ve 400x400 giriş değerlerinde 60fps değerine ulaşabildiği bilgisine varılmıştır. Yapılan bu çalışmada çıkan sonuçlara göre gerçek zamanlı araç tespiti için kullanılabilecek bir yapı ortaya çıkarmıştır. Çalışma sonucunda Yapılan testler nitecesinde Trafik izleme, hedef tespiti, araç sayımı, park uygulamaları gibi birçok alanda kullanılabileceği düşünülmektedir. [1]

İkinci olarak incelediğimiz çalışma ise, araç üzerinde bulunan kamera algılayıcısı ile çekilen trafik sahne resimleri üzerinde trafik işaret levhası tespitini uygun şekilde eğitilmiş olan Konvolüsyonel Sinir Ağları (CNN) yöntemiyle çalışılmıştır. 2 etap olarak ele alınan çalışma; bir görüntüdeki potansiyel trafik levhalarını tespit etmek TLT iken TLS seçilen bölgelerdeki trafik levhasının sınıfını belirlemek ya da ilgili bölgede trafik levhası olmadığına karar verme olarak ele alınmıştır. Algoritma

Alman Trafik İşareti Algılama Karşılaştırması (GTSDb) veriseti üzerinde test edilmiş olup başarılı sonuçlara ulaşılmış olup, işleme maliyeti bakımından da düşük olduğu tespit edilmiştir. Karmaşık ve akıllı bir yapı olması sayesinde CNN bu sınıflandırma sürecinde daha iyi çalışıldığı ön plana çıkmıştır fakat karmaşık yapısı sebebiyle yapılan sınıflandırmanın uzun sürdüğü tespit edilmiştir. Bu çalışmada ortalama çalışma/işlem süresi deney sonuçlarına göre bir resim için yaklaşık 0.05 saniye arttığı tespit edilmiş olup, bunun çıkarımının performans düşüşü kabul edilebilir bir seviyede olduğu anlaşılmıştır. [2]

Üçüncü olarak incelediğimiz çalışma ise, insansız hava aracı (İHA) ile havadan çekilmiş görüntülerden nesne (silah) tespiti yapılması amaçlanmış olup, derin öğrenme algoritmalarından R-CNN çeşitleri kullanılmıştır. R-CNN metodunun kullanılmasının sebebi ise diğer algoritmalara göre resimler üzerinde daha iyi başarı almaları olarak açıklanmıştır. Çalışmada doğru tahmin oranını görebilmek için ResNet101 algoritması denenmiştir. İHA ile havadan çekilmiş 200 adet görüntü kullanılmıştır. Yapılan eğitim sonucunda veri seti üzerinde R-CNN mimarisi ve ResNet101 mimarisiyle % 99 doğruluk oranı ve hassasiyet ile görüntü tespitinde başarı sağlanmıştır. [3]

Dördüncü olarak incelediğimiz çalışmada Bebeklik döneminde fMRI ile ortaya çıkan implant öncesi kortikal aktivasyon paternlerinin koklear implantasyondan 2 yıl sonra dil performansı ile ilişkili olduğu hipotezini doğrulanmıştır. Araştırma 44 bebek ve küçük çocuk sedasyon ile klinik olarak endike bir MRI beyin çalışması üzerinde yapılmıştır. 23 katılımcıda SNHL sahipken, diğer 21 katılımcı normal işitme (NH) kontrolleri test edilmiştir. Tedavi başlangıcından 2 yıl sonrasına kadar incelemesi olan uzun süre üstünde çalışılmıştır. CI ameliyatından iki yıl sonra, CI'yi takiben işitsel, konuşma ve dil sonuçlarını değerlendirmek için test edilmiştir. Otomatik parametre seçimli yarı denetimli model iç içe geçmiş LOOCV yaklaşımı kullanılarak değerlendirmiş olup BoW21 ve BoW44 için daha iyi performans elde edilmiştir. [4]

Son olarak incelediğimiz çalışma ise, tam otomatik derin öğrenme yöntemlerini, yarı otomatik yöntemleri ve manuel yöntemleri kullanmayı destekleyen tıbbi görüntü açıklama (RIL-Contour) yazılımıdır. Yarı otomatik özelliği için Minimal Etkileşimli Hızlı Organ Segmentasyonu (MIROS) algoritmasından yararlanılmıştır. Kullanıcı arayüz yapılandırılırken çok yönlü tasarlanmıştır. Proje Karın BT fotoğraflarını 35 benzersiz organ ve dokuyu birbirinden ayırmayı içermektedir. Projede, 132 vaka bölümü içeren yapı bir haftada başarılı bir şekilde sonuçlandırılmıştır. Görüntülenen veriler NIFTI formatıyla dışarı aktarılırken görüntülenemeyen veriler CSV ve Excel formatında aktarılmaktadır. [5]

Fizibilitedeki amaç projenin olabilirliğini incelemektir. Bu amaçla bu kısımda projenin ekonomik, yasal, teknik ve zaman fizibiliteleri ayrı ayrı oluşturularak gerçekleştirilebilirliği incelenmiştir.

3.1 Teknik Fizibilite

Proje için kullanılacak yazılım ve donanım özelliklerini bu başlık altında inceledik.

3.1.1 Yazılım Fizibilitesi

Proje için gerekli yazılımsal ihtiyaçlar belirlendi.

- İşletim Sistemi: Windows10
- Uygulama Geliştirme Dili: Görüntü işleme ve derin öğrenme için bir çok hazır kütüphane bulunması, kaynak ve doküman yönünden zengin olması sebebiyle: Python v3.8.5
- Uygulama Geliştirme Ortamı: Anaconda Spyder ve Qt designer
- Kullanılacak Framework ve Python Kütüphaneleri: Torch, Cv2, Numpy, Xml, Sys ve PyQt5

3.1.2 Donanım Fizibilitesi

Bizim proje için kullanacağımız donanımsal özellikler tablo 3.1 gibidir. Ön inceleme aşamasında yaptığımız işlemler için gerekli olacak disk kapasitesi 500 MB olarak düşünülmüştür.

İşlemci:	Intel i7-7700HQ CPU 2.80GHz
Sistem türü:	64 bit işletim sistemi, x64 tabanlı işlemci
RAM Kapasitesi:	16 GB
Ekran kartı:	Nvidia GTX 1050
Disk Kapasitesi:	1 TB HDD

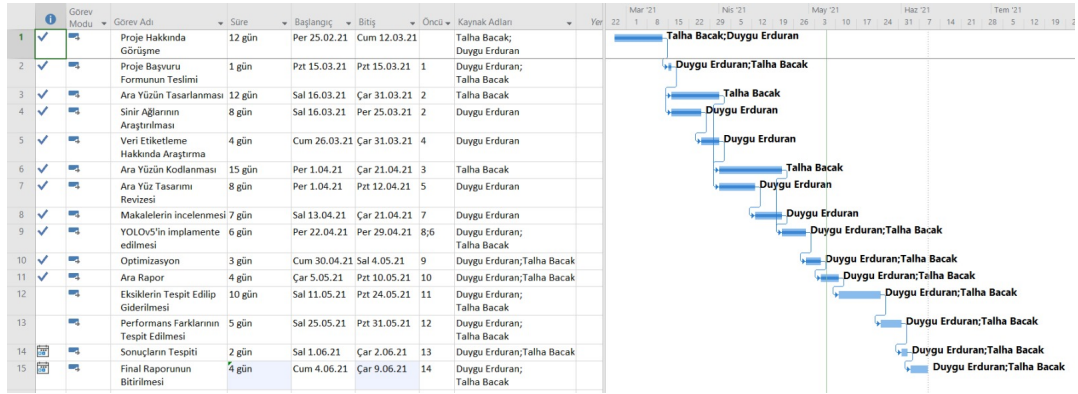
Tablo 3.1 Donanım Fizibilitesi

3.2 İletişim Fizibilitesi

İletişim Fizibilitesi başlığı altında iletişim teknolojileri ve buna bağlı olarak kullanılması uygun olan cihaz/yazılım vb. teknik özellikler incelenmiştir. Proje boyunca uzaktan yürütülmesi gereken kısımlarda iletişim kanalı olarak Zoom, Discord uygulamaları kullanılması tercih edilmiştir. Ayrıca uzaktan masaüstü erişimi içinse AnyDesk uygulamasından yararlanılması projenin devamlılığı için yeterli görülmüştür.

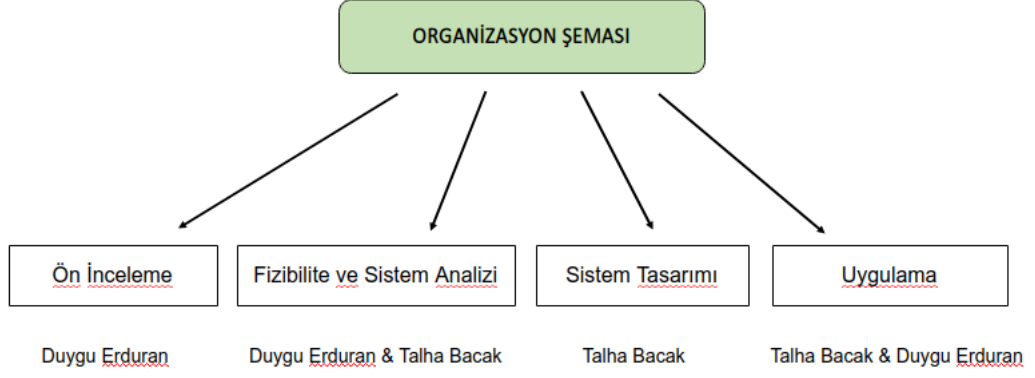
3.3 İşgücü ve Zaman Planlaması

Bu aşamada projenin önemli tarihleri ve zaman çizelgesi belirlenmiştir. Önemli tarihleri ve planlanan sürecin ne ölçüde gerçekleştiğini gösteren Gantt diyagramı Şekil 3.1 'de eklenmiştir.



Şekil 3.1 Gantt Diyagramı

Proje raporuna da konu olan aktivitelerin organizasyon şeması Şekil 3.2 'deki gibidir.



Şekil 3.2 Organizasyon Şeması

3.4 Ekonomik Fizibilite

Projede kullanılan tüm yazılım, kütüphane ve veri setinin açık kaynak kodlu, ücretsiz sağlanan versiyonlar olması ve projeyi geliştirme sırasında kullanılan donanım cihazlarının kişisel kullanılan cihazlar olması, proje geliştiricilerinin herhangi bir ücret talebi olmadığı için bu projede mali bir yükümlülük bulunmamaktadır.

3.5 Yasal Fizibilite

Proje 5846 sayılı "Fikir ve Sanat Eserleri Kanunun" 2.maddesine göre ilim ve edebiyat eseri kapsamındadır. 2.MADDE: Herhangi bir dil ve yazı ile ifade olunan eserler ve her biçim altında ifade edilen bilgisayar programları ve bir sonraki aşamada program sonucu doğurması koşuluyla bunların hazırlık tasarımları" ilim ve edebiyat eseri kapsamındadır.

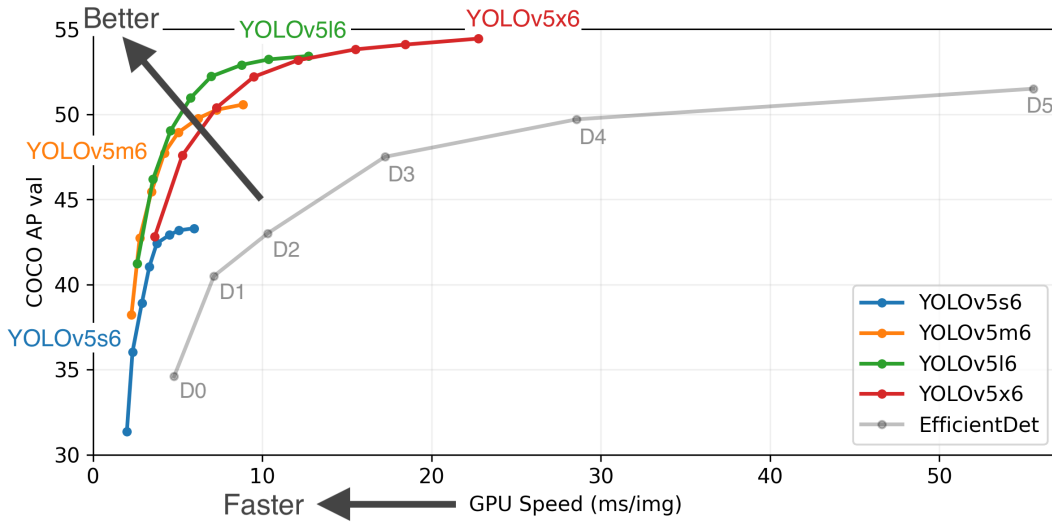
Gerekli görülen yerlerde alıntı yapılan yerler için aynı kanunun 35. maddesine uygun bir şekilde alıntı yapılmıştır.

35.MADDE: Kanunumuzda iktibas serbestisi düzenleyen 35. maddede “Bir eserden aşağıdaki hallerde iktibas yapılması caizdir: 1. Alenileşmiş bir eserin bazı cümle ve fıkralarının müstakil bir ilim ve edebiyat eserine alınması; 2. Yayınlanmış bir bestenin en çok tema, motif, pasaj ve fikir nevinden parçalarının müstakil bir musiki eserine alınması; 3. Alenileşmiş güzel sanat eserlerinin ve yayınlanmış diğer eserlerin, maksadın haklı göstereceği bir nispet dahilinde ve münderecatını aydınlatmak maksadıyla bir ilim eserine konulması; 4. Alenileşmiş güzel sanat eserlerinin ilmi konferans veya derslerde konuyu aydınlatmak için projeksiyon ve buna benzer vasıtalarla gösterilmesi. İktibasın belli olacak şekilde yapılması lazımdır. İlim eserlerinde, iktibas hususunda kullanılan eserin ve eser sahibinin adından başka bu kısmın alındığı yer belirtilir." hükmü yer almaktadır. Bu hükümde belirtilen sınırlar

dahilinde eser sahibinin iznini almaksızın eserin kullanılmasının mümkün olduđu belirtilmiřtir.

4 Sistem Analizi

Sistem kullanıcının ara yüzden fotoğraf seçmesiyle başlar. Kullanıcı isterse manuel bir şekilde dikdörtgen şeklinde etiketleme yapacaktır. Eğer otomatik bir şekilde etiketlemek isterse derin öğrenme algoritmalarından COCO datasetiyle eğitilmiş YOLOv5s (şekil 4.1'e bakabilirsiniz) ve YOLOv4-tiny modellerini kullanabilecektir. Ayrıca kullanıcı isterse kendi eğittiği YOLO modellerini de kullanabilir. Kullanıcı daha sonra etiketler üzerinde düzenlemeler yapabilecek ve etiket bilgilerini dosyaya yazdırabilecektir. Bunların dışında önceden etiketlenmiş fotoğraf verilerini de import ederek fotoğraf üzerinde etiketleri görebilecektir. Pascal formatında .xml ve YOLO formatında .txt uzantılı dosyalarda işlemler yapılacaktır. İnceleme sonucunda gereksinimleri belirlemek amacıyla taslak diyagramı, veri akış diyagramı ve use case diyagramı oluşturuldu. Manuel olarak elle tek tek etiketleme ve otomatik olarak derin öğrenme modelleri ile etiketleme yapacak bir ara yüz için gereksinimler belirlendi.



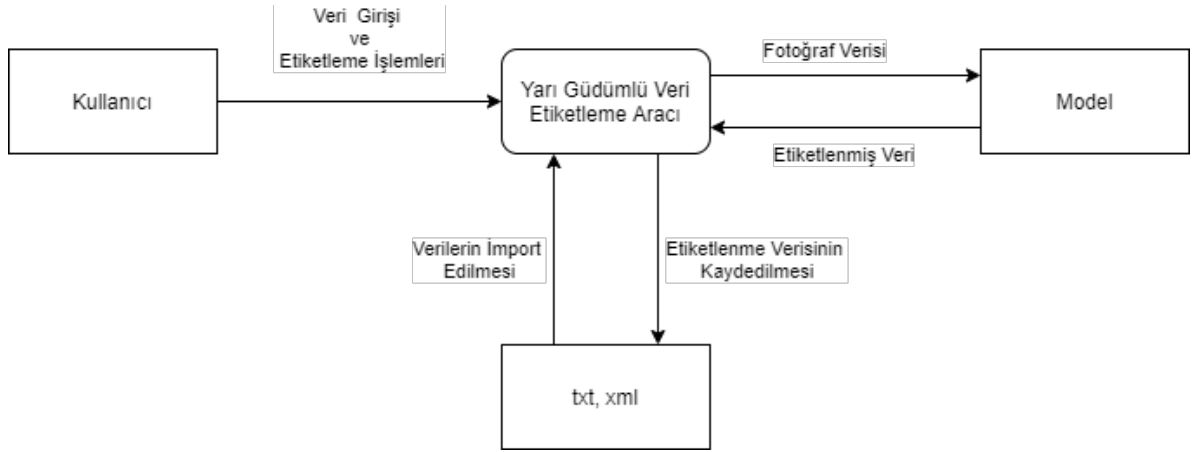
Şekil 4.1 YOLOv5 modellerinin performans değerleri [6]

4.1 Projenin Amacı

Projemiz daha önceden eğitilmiş olan model yardımıyla, yarı-gözetimli bir şekilde yeni veri setleri oluşturulmasını çok daha kolaylaştıracaktır. Daha efektif bir şekilde veri setlerinin eğitilmesine katkı sağlayacaktır.

4.2 Taslak Diyagramı

Sistem ile varlıklar arasındaki durumu ilişkisel olarak temsil etmek için kullanılır. Burada sistem "YARI GÜDÜMLÜ VERİ ETİKETLEME ARACI" olarak belirlendi, varlık olarak "Kullanıcı", "Model (Derin Öğrenme Modeli)" ve ".txt/.xml dosyaları" olarak tanımlanmıştır. Oluşturulan tasarım şekil 4.2'deki gibidir.



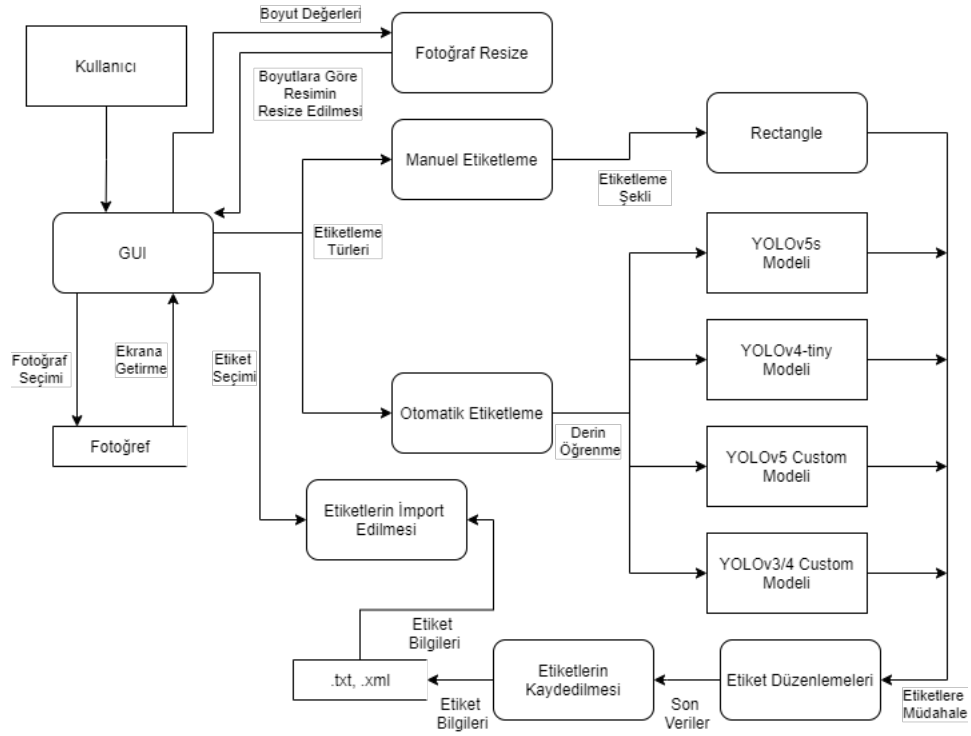
Şekil 4.2 Taslak Diyagramı

4.3 Veri Akış Diyagramı

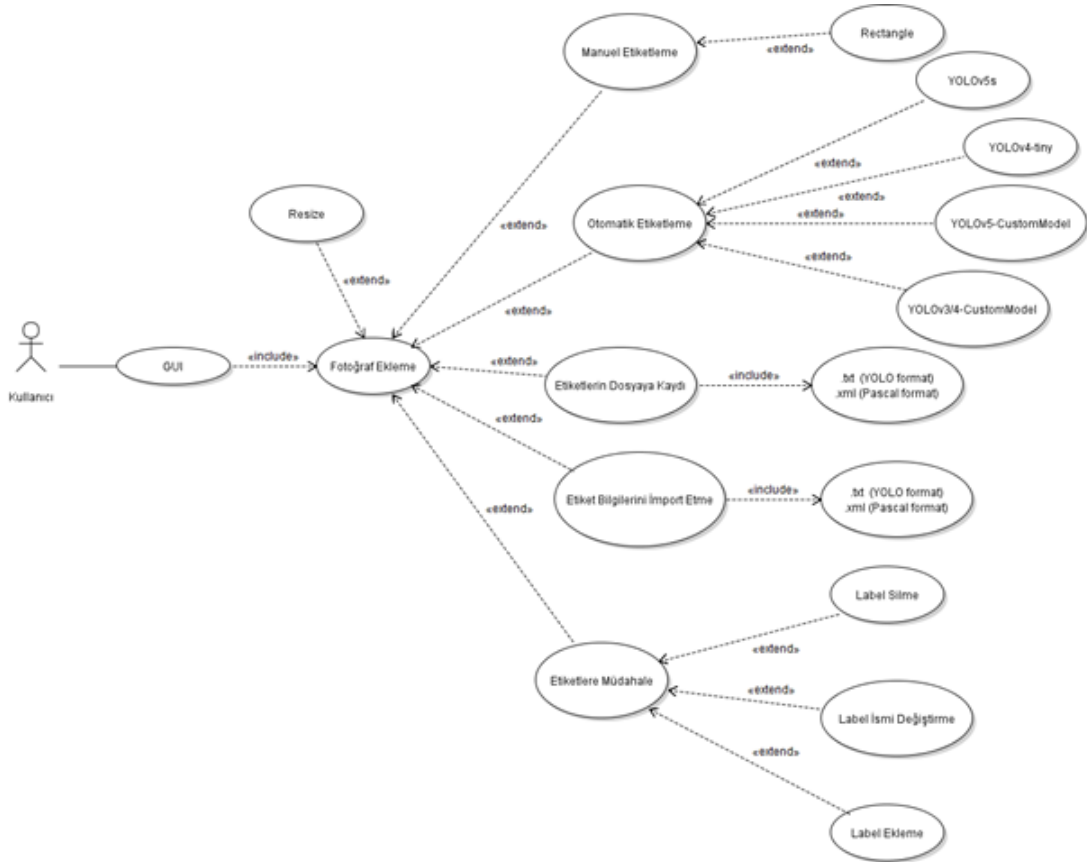
Bu kısımda taslak diyagramda belirlenen varlıklar ve sistem arasındaki ilişkiyi geliştirecek süreçlere göre daha da detaylandırarak şekil 4.3'de görselleştirilmiştir.

4.4 Use Case Diyagramı

Projemizin işleyişini kullanım durumunda gerçekleşecek olayları açıklayan use case diyagramı oluşturularak tasarımın işleyişi hakkında temel bilgilendirilme şekil 4.4'de şematize edilmiştir.



Şekil 4.3 Veri Akış Diyagramı



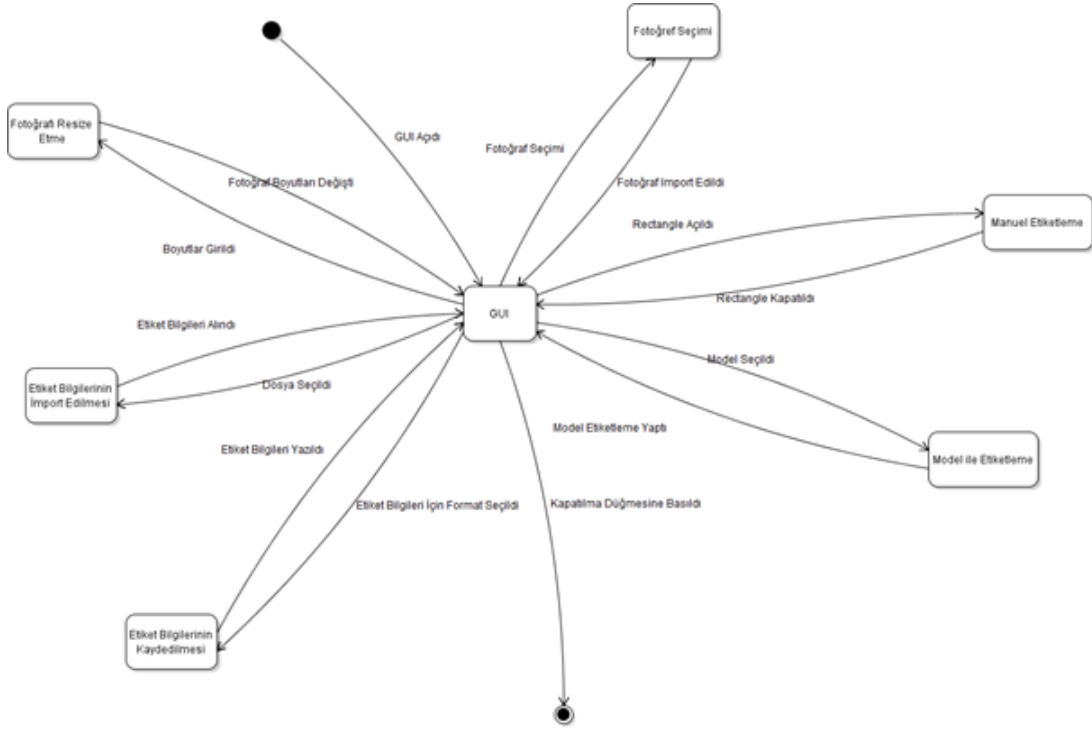
Şekil 4.4 Use-Case Diyagramı

5

Sistem Tasarımı

5.1 Yazılım Tasarımı

Yazılım tasarımında sistemin ana işlemleri belirlenmiş, state diyagramı ile şekil 5.1'deki gibi gösterilmiştir. Daha sonra bu ana işlemler ayrıntılarıyla açıklanmıştır.



Şekil 5.1 State diyagramı

5.1.1 Fotoğraf Seçimi

Kullanıcı ara yüzü açtığında bilgisayarındaki istediği fotoğrafı import edebilir.

5.1.2 Manuel Etiketleme

Etiketleme yöntemi olarak rectangle seçildiğinde kullanıcı manuel olarak dikdörtgen şeklinde etiketlemeler yapabilecektir.

5.1.3 Etiket Bilgilerinin Import Edilmesi

Önceden hazır olarak etiketlenmiş datanın bulunduğu Pascal ya da YOLO formatındaki dosyalardaki etiket bilgileri alınabilir.

5.1.4 Model ile Etiketleme

Etiketleme yöntemi olarak COCO datasetiyle eğitilmiş YOLOv5s ve ya YOLOv4-tiny modelleri seçilebilir. Ayrıca kullanıcı tarafından eğitilmiş bir YOLO modeli de eklenerek otomatik olarak etiketlemeler yapılabilir.

5.1.5 Fotoğrafi Resize Etme

Fotoğraf kullanıcıdan alınan X ve Y değerlerine göre yeniden boyutlandırılır.

5.1.6 Etiket Bilgilerinin Kaydedilmesi

Etiketleme işlemleri bittikten sonra kullanıcı Pascal ya da YOLO formatında etiket bilgilerini kaydedebilecektir. Sonrasında ise isterse yeni bir fotoğraf ekleyebilecek ya da uygulamadan çıkabilecektir.

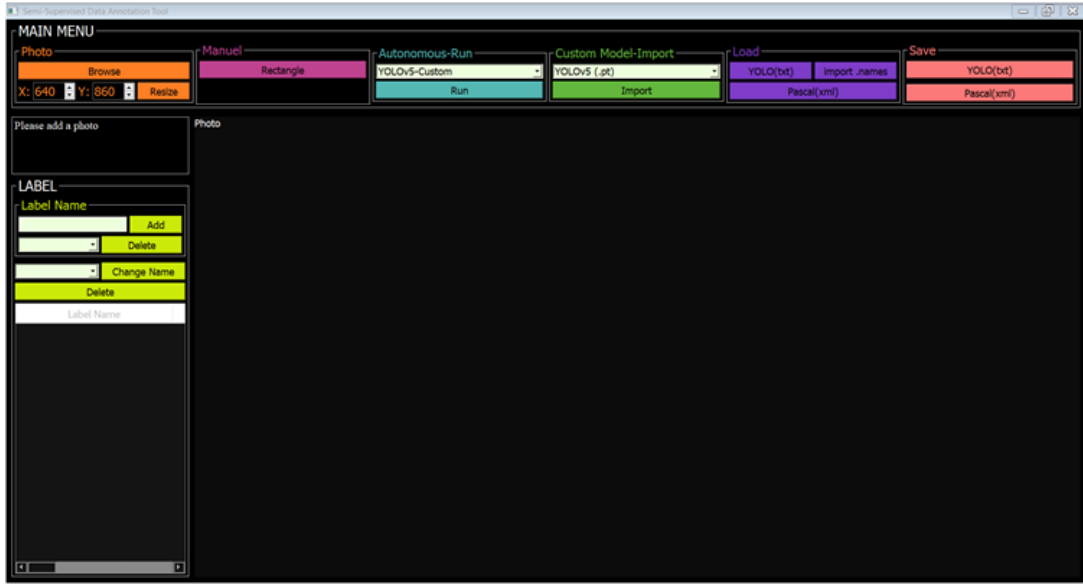
5.2 Veri Tabanı Tasarımı

Uygulamada dışardan alınacak verilerin ilki olarak fotoğraf kullanılacaktır. Bilgisayarda erişimi olan her fotoğraf uygulamaya dahil edilebilecektir. İkinci olarak, uygulamada etiketlenen fotoğrafın etiket bilgilerini Pascal ya da YOLO formatında bir dosya oluşturarak bu dosyaya yazacaktır. Üçüncü olarak, önceden etiketlenmiş Pascal ya da YOLO formatındaki veriler de import edilebilecektir. Bunların dışında derin öğrenme modellerimizde COCO veri seti kullanılacaktır. COCO veri setiyle eğitilmiş olan YOLOv5s ve YOLOv4-tiny modelleri kullanılacaktır. Kullanıcı isterse sonradan eğitilmiş YOLO modellerini de kullanabilecektir.

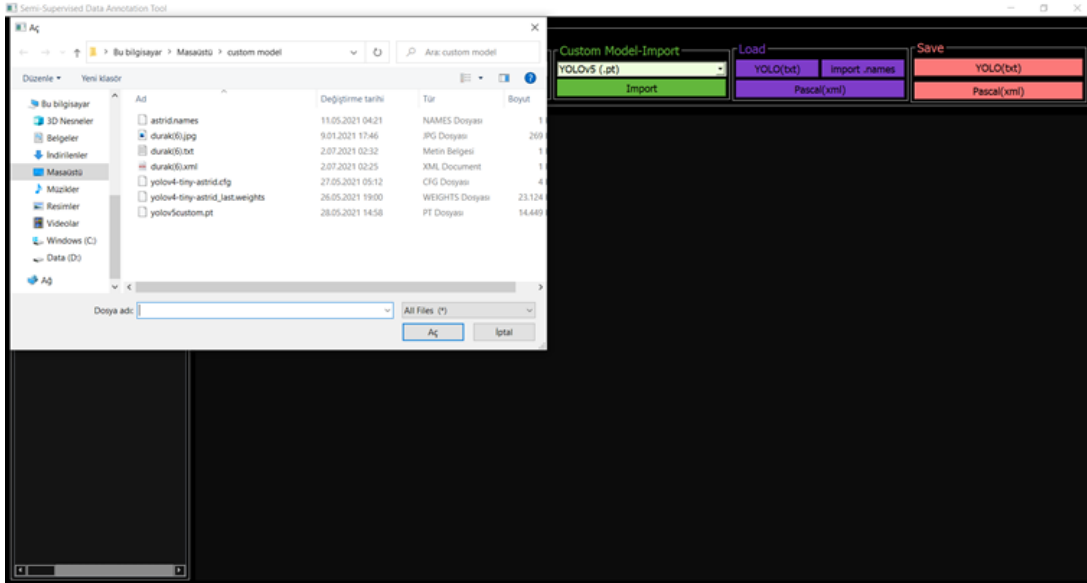
5.3 Girdi-Çıktı Tasarımı

Geliştirilen ara yüz üzerinden alınarak etiketlenen verilerin gerekli bilgileri uygun görülen bir formatta çıktı olarak kaydedilecektir. Örneğin GUI açıldığında şekil 5.2'deki gibi açılacaktır. Browse tuşuna basıldığında şekil 5.3'deki gibi dosya seçme ekranı gelecektir. Fotoğraf seçildikten sonra şekil 5.4 gibi resim ekrana gelecektir. Etiketlendikten sonra etiketler YOLO formatında kaydedilmek istenilirse şekil 5.5'deki gibi, Pascal formatında kaydedilmek istenilirse de şekil 5.6'daki gibi çıktı dosyası

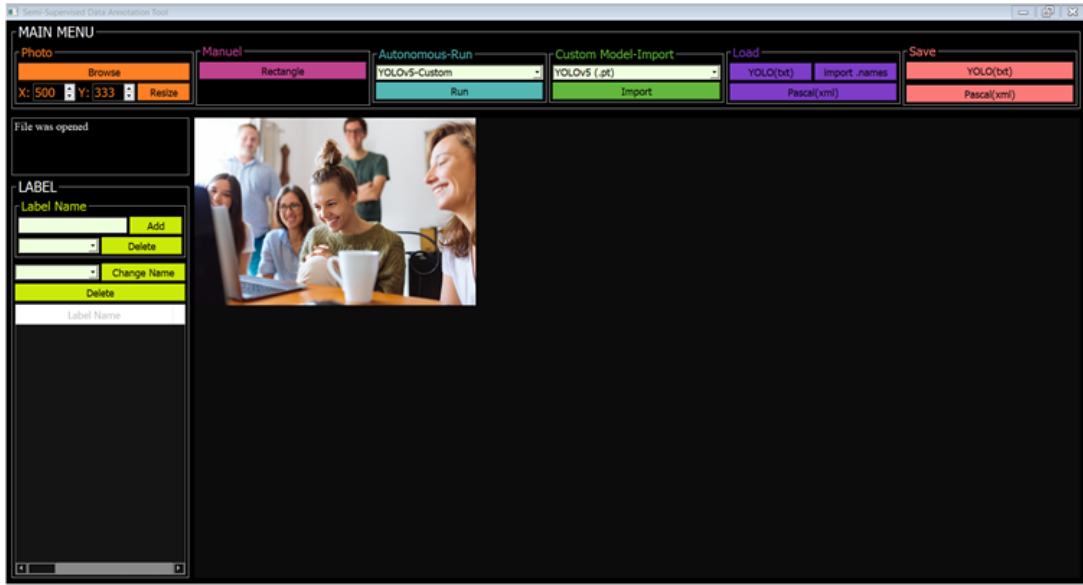
oluşacaktır. Etiketli verilerin import edilmesi de benzer şekilde "Load" kısmındaki butonlardan yapılır. Model import edilmek istenilirse "Custom Model-Import" kısmından yapılacaktır.



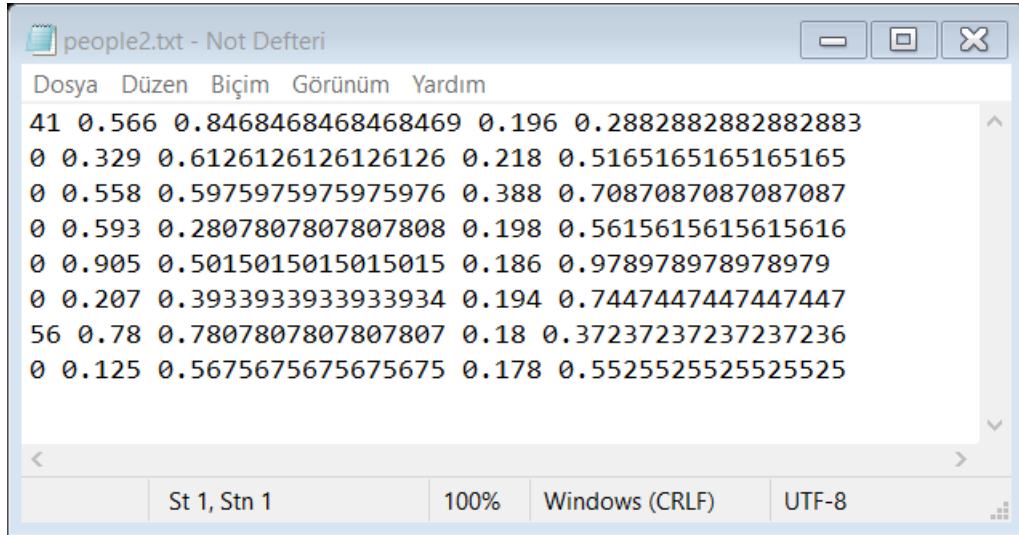
Şekil 5.2 PyQt5 ile oluşturulan kullanıcı ara yüzü



Şekil 5.3 Girdinin Alınması (Fotoğraf)



Şekil 5.4 Girdi olarak alınan fotoğraf



Şekil 5.5 Çıktı olarak YOLO formatında kaydedilen txt dosyası


```

<annotation>
  <folder>Resim</folder>
  <filename>/people2.jpg</filename>
  <path>C:/Users/msi-nb/Desktop/Resim/people2.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>500</width>
    <height>333</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>cup</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>234</xmin>
      <ymin>234</ymin>
      <xmax>332</xmax>
      <ymax>330</ymax>
    </bndbox>
  </object>
  <object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>110</xmin>
      <ymin>118</ymin>
      <xmax>219</xmax>
      <ymax>290</ymax>
    </bndbox>
  </object>
  <object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>182</xmin>
      <ymin>81</ymin>
      <xmax>276</xmax>
      <ymax>276</ymax>
    </bndbox>
  </object>
</annotation>

```

Şekil 5.6 Çıktı olarak Pascal formatında kaydedilen xml dosyasının bir kısmı

6

Uygulama

Uygulama PyQt5 ile oluşturulmuş bir ara yüzle kullanılmaktadır. Bu yüzden ara yüz üzerinden uygulama anlatılacaktır.

Uygulamada kullanılan kütüphaneler tablo 6.1’de belirtilmiştir.

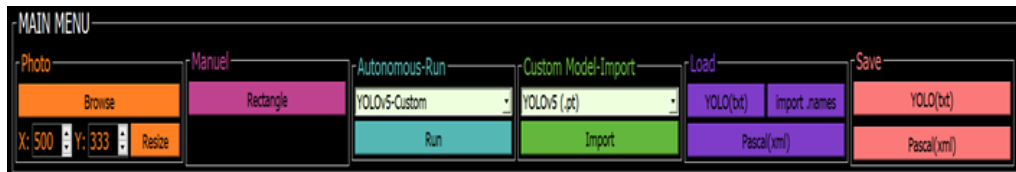
Kullanılan Kütüphaneler
torch
cv2
numpy
PyQt5
lxml
xml
sys

Tablo 6.1 Kütüphane bilgileri

6.1 Uygulamanın Temel Yapısı

Uygulamanın tasarımı 4 temel alandan oluşmaktadır. Bunlar:

- Temel işlemlerin yapıldığı "MAIN MENU" (bakınız şekil 6.1).
- Etiket verileriyle ilgili kısmın bulunacağı "LABEL" (bakınız şekil 6.2).
- Resmin bulunacağı label (bakınız şekil 6.3).
- Kullanıcıyı bilgilendirmek için bulunan label (bakınız şekil 6.4).



Şekil 6.1 MAIN MENU

Label

Label Name

Add

cup

Delete

cup

Change Name

Delete

Label Name

1 cup

2 person

3 person

4 person

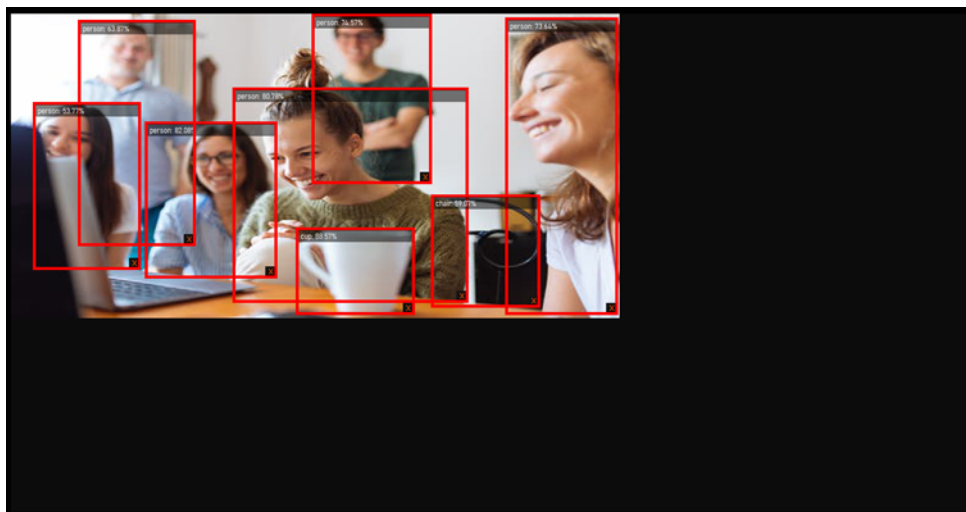
5 person

6 person

7 chair

8 person

Şekil 6.2 LABEL



Şekil 6.3 Photo Label

```
.txt was writed
Writing format: YOLO
```

Şekil 6.4 Message Label

6.2 MAIN MENU

Bu kısımda programın genel işleyişi yönetilmektedir. Fotoğrafın import edilmesi ve yeniden boyutlandırılması, manuel etiketleme, otomatik etikleme (modellerin import edilmesi ve seçilmesi), etiket bilgilerinin kaydedilmesi ve import edilmesi gibi özellikler açıklanacaktır.

6.2.1 Photo

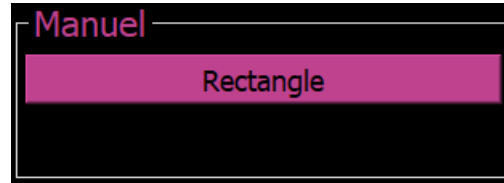
Photo grubunun 2 fonksiyonu vardır ve bunun için şekil 6.5'deki gibi 2 buton bulunmaktadır. Bunların ilki "Browse" tuşudur ve fotoğrafların import edilmesini sağlar. Diğeri ise "Resize" tuşudur ve girilen değerleri alarak fotoğraf yeniden boyutlandırılır.



Şekil 6.5 Photo

6.2.2 Manuel

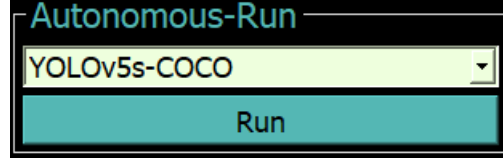
Manuel grubunun tek fonksiyonu vardır ve bu fonksiyon şekil 6.6'daki gibi "Rectangle" yazan butona basıldığında manuel olarak dikdörtgen şeklinde etiketleme yapılabilmektedir.



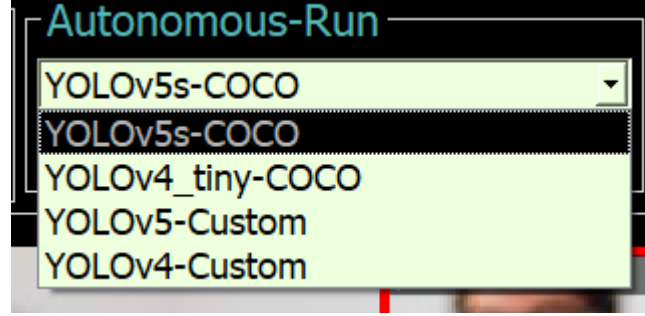
Şekil 6.6 Manuel

6.2.3 Autonomous-Run

Otonom olarak çalışması istenilen model bu kısımda (şekil 6.7) çalıştırılır. Çalışacak model şekil 6.8'de belirtilenler arasından seçilebilir. Custom yazan modeller dışardan import edilmelidir.



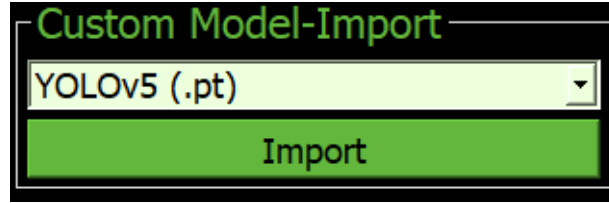
Şekil 6.7 Autonomous-Run



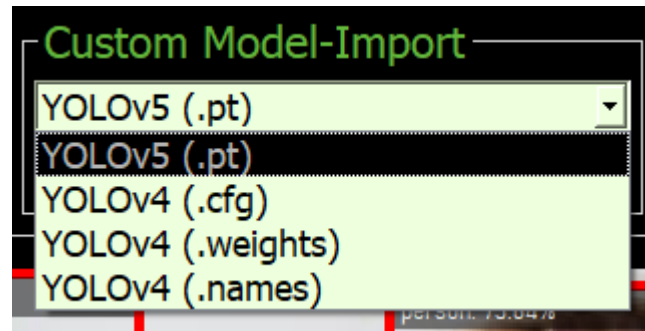
Şekil 6.8 Kullanılabilecek Modeller

6.2.4 Custom Model-Import

Otonom olarak çalışması istenilen custom-modeller bu kısımdan (şekil 6.9) import edilir. Gerekli dosyalar burdan tek tek import edilir, import edilecek dosyalar şekil 6.10'da belirtilenlerden seçilir.



Şekil 6.9 Autonomous-Run

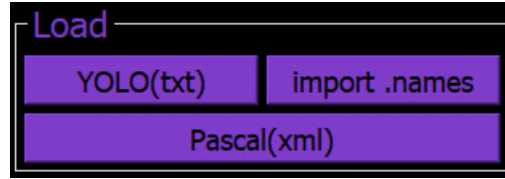


Şekil 6.10 Modeller için gerekli dosyaların import edilmesi

6.2.5 Load

Önceden etiketlenmiş verilerin kontrol edilmesi amacıyla bu yapı kullanılabilir. Bu kısımda YOLO formatındaki ve Pascal formatındaki önceden etiketlenmiş ve bu

bilgilerin bulunduğu dosyalar import edilebilir. YOLO formatında import etmek için önce "import .names" butonuna basılarak sınıf isimlerinin olduğu dosya alınmalıdır. Hangi formatta import edileceği şekil 6.11’de gözüken butonlardan seçilebilir.



Şekil 6.11 Load

6.2.6 Save

Bu kısımda ise yapılan etiketlemeler YOLO ya da Pascal formatında kaydedilebilir. Şekil 6.12’de gözüken butonlara tıklanılarak istenilen formatta dosya oluşturulup etiket bilgileri yazdırılır. Yazdırılan dosya etiketlenen fotoğrafın isminde ve etiketlenen fotoğrafın bulunduğu dosya yolunda oluşturulur.



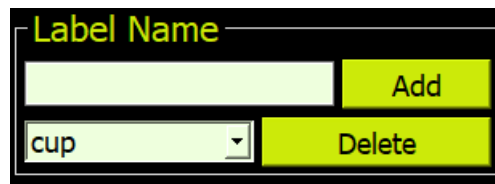
Şekil 6.12 Save

6.3 LABEL

Bu kısımda etiketler hakkında bilgi edinilebilir ve etiketlere müdahale edilebilir. Etiket ismi ekleme ve silme, etiketlenmiş verinin isminin değiştirilmesi ve yeni etiketlenecek verinin isminin seçilmesi, etiketlerin silinmesi gibi özellikler açıklanacaktır.

6.3.1 Label Name

Bu kısımda yeni etiket sınıflarının isimleri eklenebilir ya da bulunan etiket sınıflarının ismi silinebilir. Bu işlemler şekil 6.13’te görünen yerden yapılmaktadır.



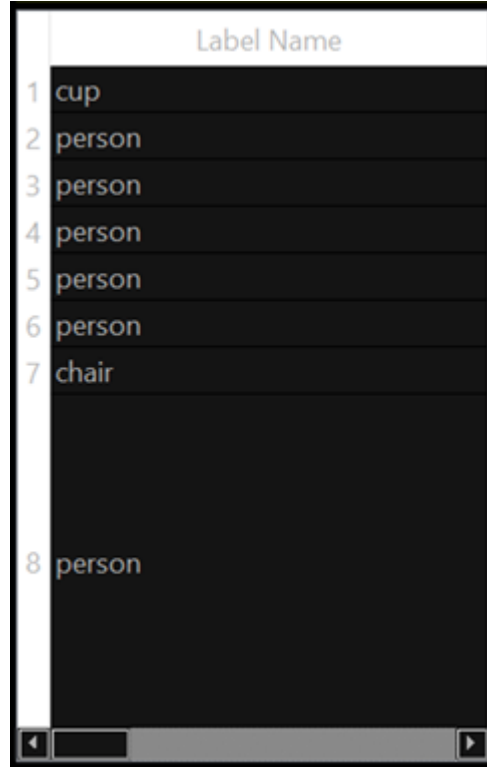
Şekil 6.13 Label Name Grubu

6.3.2 Etiketlere Müdahale

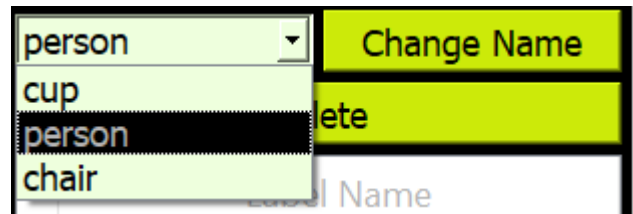
Etiketlere müdahale etmek için şekil 6.14'te görülen kısım kullanılmaktadır. Manuel olarak yeni etiketleme yapılırken burada (şekil 6.16'daki gibi değiştirilebilir) seçili olan isim etiket sınıfının ismi olarak atanmaktadır. Eğer önceden oluşturulmuş bir etiket ismi değiştirilmek istenilirse şekil 6.15'te gözüken etiket isimleri seçilip şekil 6.14'teki "Change Name" butonuna tıklandığında seçili olan isimle (şekil 6.16'da seçildiği gibi) değiştirilecektir. Etiketlerin silinmesi için ise şekil 6.15'teki etiketlerden biri seçilerek şekil 6.14'teki "Delete" butonuna basılması gerekir.



Şekil 6.14 Etiketlere müdahale edebilecek butonlar

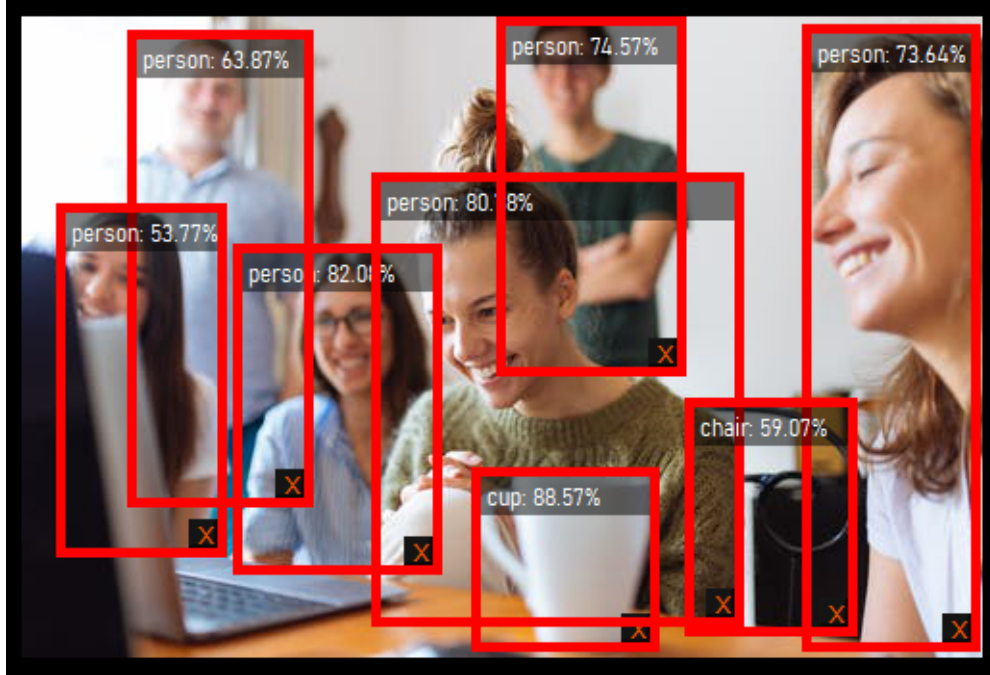


Şekil 6.15 Etiket Tablosu



Şekil 6.16 Etiket isim seçimi

Etiketler ara yüz üzerinden silinmek istenilirse şekil 6.17'deki gibi etiketlenmiş olan fotoğraf üzerindeki etiketlerde bulunan 'X' işaretine tıklanarak silme işlemi yapılır.



Şekil 6.17 Etiketlenmiş fotoğraf

7 Deneysel Sonuçlar

Bu bölümde uygulamanın temel kullanım amacı olan manuel etiketleme ve otonom etiketleme ele alınacaktır. Manuel etiketlemede kullanıcılar etiketleri ve etiket sınıflarını kendileri oluşturmuştur. Otonom etiketlemede ise etiketler ve etiket sınıfları seçilen derin öğrenme modeli tarafından oluşturulmuştur. Derin öğrenme modeli olarak COCO datasetiyle eğitilmiş YOLOv5s kullanılmıştır. 10 kullanıcıya 5 fotoğrafı önce otonom etiketlemeleri sonra ise manuel etiketlemeleri istenmiştir. Bütün kullanıcılar aynı 5 fotoğraf üzerinde etiketleme yapmıştır. Kullanıcılar bu 5 fotoğrafta belirtilen cisimleri etiketlemiştir. Fotoğrafların etiket bilgileri tablo 7.1’de gösterilmiştir. Bu etiketleme işlemleri için geçen süreler tablo 7.2’de belirtilmiştir. Belirtilen süreler 5 fotoğrafın sırayla etiketlenmesi sonucu geçen toplam süredir.

Fotoğraf	Etiket Sayısı	Etiket Sınıf Sayısı
1	4	2
2	1	1
3	4	1
4	8	3
5	5	1

Tablo 7.1 Fotoğrafların etiket bilgileri

Kişi	Otonom Etiketleme Süresi(sn)	Manuel Etiketleme Süresi(sn)
1	27	123
2	22	104
3	21	94
4	22	96
5	29	102
6	32	95
7	33	86
8	25	104
9	27	89
10	24	97

Tablo 7.2 Etiketleme süre bilgileri

8 Performans Analizi

Uygulamamızda manuel etiketleme ve otonom etiketleme olmak üzere 2 temel etiketleme yöntemi bulunmaktadır. Manuel etiketlemede kullanıcı tarafından tek tek dikdörtgen şeklinde etiketlemeler yapılabilmektedir. Otonom etiketlemede ise herhangi bir YOLO modeli import edilebilir ya da uygulamadaki hazır modeller kullanılabilir. Deneysel sonuçlar için uygulamada hazır bulunan COCO data setiyle eğitilmiş YOLOv5s modeli kullanılmıştır. Modeller import edilirken geçen süreler tablo 7.2’de belirtilen süreler dahil edilmemiştir. Modellerin import edilmesi ve ya indirilmesi gereken modeller varsa bunlar donanım ve internet hızına bağlı olarak kısa bir süre vakit alacaktır.

Uygulamadaki hazır kullanılabilecek modeller YOLOv5s ve YOLOv4-tiny olduğundan modeller oldukça hızlı çalışmaktadır. Dışardan import edilen modeller ile etiketleme yapılırken model yapısına göre etiketleme hızı değişebilir.

8.1 Deneysel Sonuçların Değerlendirilmesi

Bu bölümde, elde edilen deneysel sonuçlar değerlendirilecektir. Değerlendirme kriteri olarak harcanan zaman ele alınmıştır. Değerlendirilecek veri tablo 7.2’de bulunmaktadır.

10 kişinin harcadığı otonom etiketleme ve manuel etiketleme sürelerinin ortalaması alınarak zaman açısından oluşan fark değerlendirilecektir. Otonom etiketleme için harcanan ortalama süre **26,2** saniyedir. Manuel etiketleme için harcanan ortalama süre **99** saniyedir. Görüldüğü üzere manuel etiketleme otonom etiketlemenin **3,77** katı sürmüştür. Bu deneyde her kullanıcı 5 fotoğrafta toplam 22 etiket ve 8 etiket sınıfı oluşturmuştur. Fotoğraf sayısı, etiket sayısı ve ya etiket sınıfı sayısı arttıkça otonom etiketleme ile manuel etiketleme arasındaki harcanan zaman oranı artarak devam edecektir.

Otonom etiketleme, az sayıda etiketleme yapılırken manuel etiketlemeye göre

avantajlı olduđu gibi etiketlenecek veri arttıkça da avantajını artırarak devam ettirecektir. Otonom etiketlemede oluşabilecek dezavantaj ise bütün etiketlemeleri hatasız yapamamasıdır. Buradaki hata oranı kullanılacak modele göre değışiklik gösterecektir.

Bu projede derin öğrenme modelleri kullanarak nesne tespiti yapılmış ve nesnelerin etiketlenmesi sağlanarak windows üzerinde çalışan GUI kullanılarak yarı güdümlü veri etiketleme aracı yapılması amaçlanmıştır.

Veri etiketleme 2 ana yöntemle ayrılmıştır. Biri manuel etiketleme, diğeri ise otonom etiketlemedir. Manuel etiketlemede mouse ile etiketleme yapılabilir. Otonom kısımda ise COCO data setiyle eğitilmiş YOLOv5s, YOLOv4-tiny modelleri kullanılabilir ya da başka YOLO modelleri import edilebilir.

Projede otonom etiketleme ve manuel etiketleme karşılaştırılarak otonom etiketlemenin avantajları ve dezavantajları incelendi. Data setlerin oluşturulması oldukça vakit almakta ve yanlış etiketlemelerle sık sık karşılaşmaktadır. Bu sorunları daha aza indirebilecek bir yaklaşım ortaya konulmaya çalışıldı.

Bu çalışmayla elde edilen sonuçlara göre otonom etiketleme manuel etiketlemeye göre oldukça hızlı çalışmaktadır. Otonom etiketleme nesne tespitinde yanlış ya da eksik etiketleme yapabilmektedir. Bundan dolayı uygulamada otonom etiketleme yapıldıktan sonra etiketlere manuel olarak müdahale edilebilmektedir. Oluşturulan etiketler YOLO formatında txt dosyasına ya da Pascal formatında xml dosyasına kaydedilebilir.

Bu proje üzerinde çalışırken, önceden etiketlenmiş verilerde zaman zaman hatalar olduğunu ve data setlerinde birçok hata olduğunu tespit ettik. Bu sebeple uygulamaya önceden etiketlenmiş verilerin yüklenip kontrol edilebileceği bir özellik eklenmiştir. Bu veriler YOLO formatında txt dosyalarından ya da Pascal formatında xml dosyalarından import edilebilir. Bütün YOLO modelleri kullanılacak ve import edilebilecek biçimde uygulamanın kapsamı da genişletilmiştir. Kullanıcıların düşük çözünürlüklü fotoğraflardaki nesneleri daha belirgin görebilmesi için fotoğrafın yeniden boyutlandırılabilmesi özelliği eklenmiş ve bu durumun etiket bilgilerinde oluşturabileceği hatalar engellenmiştir.

Projede etiketlere müdahale edilirken daha kullanıcı dostu çözümler sunulabilir. YOLO dışında farklı derin öğrenme modelleri de kullanılabilecek şekilde geliştirilerek kapsamı genişletilebilir. Windows dışı platformlara da uygulanabilecek şekilde bir geliştirme yapılabilir.

Referanslar

- [1] H. S. DIKBAYIR and H. İ. BÜLBÜL, “Derin öğrenme yöntemleri kullanarak gerçek zamanlı araç tespiti,” *TÜBAV Bilim Dergisi*, vol. 13, no. 3, pp. 1–14,
- [2] M. Çetinkaya and T. Acarman, “Trafik işaret levhası tespiti için derin öğrenme yöntemi,” *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*, vol. 3, no. 2, pp. 140–157,
- [3] M. Burgaz, “Derin öğrenme algoritmaları kullanarak insansız hava araçları ile silah tespiti,” M.S. thesis, Batman Üniversitesi Fen Bilimleri Enstitüsü, 2020.
- [4] L. Tan, S. K. Holland, A. K. Deshpande, Y. Chen, D. I. Choo, and L. J. Lu, “A semi-supervised support vector machine model for predicting the language outcomes following cochlear implantation based on pre-implant brain fmri imaging,” *Brain and behavior*, vol. 5, no. 12, e00391, 2015.
- [5] K. A. Philbrick, A. D. Weston, Z. Akkus, T. L. Kline, P. Korfiatis, T. Sakinis, P. Kostandy, A. Boonrod, A. Zeinoddini, N. Takahashi, *et al.*, “Ril-contour: A medical imaging dataset annotation tool for and with deep learning,” *Journal of digital imaging*, vol. 32, no. 4, pp. 571–581, 2019.
- [6] Ultralytics. (2020). “Yolov5,” [Online]. Available: <https://github.com/ultralytics/yolov5> (visited on 01/05/2021).