



**TC
BARTIN ÜNİVERSİTESİ
MÜHENDİSLİK, MİMARLIK VE TASARIM
FAKÜLTESİ**

BİLGİSAYAR MÜHENDİSLİĞİ

AUTO CENSORİNG WITH AI

LİSANS BİTİRME TEZİ

Seyfullah KURT

Tez Danışmanı:

Dr. Öğr. Üyesi Hakan ÖCAL

BARTIN-2023

TEZ ETİK VE BİLDİRİM SAYFASI

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını, aksinin ortaya çıkması durumunda her türlü yasal sorumluluğu üstlendiğime dair beyanda bulunuyorum.

Öğrenci Adı ve SOYADI

İmza

Gün. Ay. Yıl

İÇİNDEKİLER

ÖZET	ii
Tablolar Listesi	iii
Şekiller Listesi	iii
Grafikler Listesi.....	iii
1. ÖZGÜN DEĞER.....	1
1.1. Konunun Önemi, Araştırma Önerisinin Özgün Değeri ve Araştırma Sorusu/Hipotezi.....	1
1.2. Amaç ve Hedefler	1
2. YÖNTEM.....	2
2.1. Sansürlenmek İstenilen Obje İçin Veri Seti Oluşturulması.	4
2.2. Sansürlenmek İstenilen Obje İçin Model Oluşturulması.	5
2.3. Sansürlenmek İstenilen Objenin Algılanması ve Sansürlenmesi	8
Sonuç ve Öneriler	15
Kaynakça.....	15

ÖZET

Günümüzde birçok yayın kuruluşu ve sosyal medya yayıncısı için yayınlarına sansür yapmak iş gücü ve zaman gerektirmektedir ve bu da bir maliyet oluşturmaktadır. Projenin amacı bir video ve/veya canlı bir görüntüdeki istenmeyen bölümleri derin öğrenme sayesinde otomatik ve gerçek zamanlı olarak sansürlemek ve sansürleme işini daha hızlı ve düşük maliyetli halde gerçekleştirmesini sağlamaktır. Proje bir video kaydını ve/veya canlı bir görüntüyü girdi olarak aldıktan sonra kullanıcının önceden belirlemiş olduğu kriterlere dayanarak girdideki istenmeyen görüntüyü (Hakaret içeren el, Yüz, Alkollü içecekler, Sigara, Cinsel içerikli eşyalar, Cinsel organ vb.) sansürlemek için görüntünün olduğu kısmı gerçek zamanlı görüntü işleme sayesinde sansürleyen bir programdır. Bu amaç doğrultusunda, öncelikle veri setleri toplanmış ve sansürleme işlemleri üzerinde çalışılmıştır. Veri setleri, gerçek zamanlı görüntü işleme için kullanılan en son teknolojilerle oluşturulmuştur. Bu veri setleri farklı durumlar için farklı modeller oluşturmak ve modellerin doğruluğunu artırmak için kullanılmıştır. Analiz yöntemleri ise, derin öğrenme algoritmaları ve görüntü işleme teknikleridir.

Anahtar Kelimeler: Yapay zeka, Sansür, Gerçek zamanlı görüntü işleme, Derin öğrenme

Tablolar Listesi

Tablo 1 Derin öğrenme kütüphaneleri 3

Tablo 2 Sınıflandırma yöntemleri 3

Şekiller Listesi

Şekil 1 RoboFlow fotoğraf ekleme ve istenilen özelliği kare içine alma 4

Şekil 2 RoboFlow veri seti 5

Şekil 3 Ultralytics veri seti 5

Şekil 4 Ultralytics hub model eğitim ayarları 6

Şekil 5 Ultralytics ve collab arasında bağlantı kurma komutları 6

Şekil 6 Ultralytics'i collaba indirme ve içeri aktarma 7

Şekil 7 Hub login ve model eğitimi komutları ve çıktıları 7

Şekil 8 Eğitim sürecinin çıktısı 7

Şekil 9 Yüz tespiti ve sansürlenmesi 9

Şekil 10 Yüz sansürlenmesi 10

Şekil 11 Sigara tespiti ve sansürlenmesi 11

Şekil 12 Sigara sansürlenmesi 12

Şekil 13 Hakaret içeren el hareketi tespiti ve sansürlenmesi 13

Şekil 14 Hakaret içeren el hareketi sansürlenmesi 14

Grafikler Listesi

Grafik 1 Epoch sayısına göre modelin doğruluk oranları grafiği 8

Grafik 2 Epoch sayısına göre modelin kayıp grafiği 8

1. ÖZGÜN DEĞER

1.1. Konunun Önemi, Araştırma Önerisinin Özgün Değeri ve Araştırma Sorusu/Hipotezi

Bu araştırma önerisi, günümüzde yayın kuruluşlarının ve sosyal medya yayıncılarının yayınlarında sansür yapıp iş gücü ve zaman harcamak zorunda kalmalarından kaynaklanan maliyeti azaltmayı hedefleyen bir derin öğrenme modeli tasarlamayı amaçlamaktadır. Projenin özgün değeri, gerçek zamanlı görüntü işleme yöntemi kullanarak otomatik sansürleme işlemini hızlı ve düşük maliyetli hale getirmesi ve küçük yayın kuruluşları, yayın girişimleri ve yayıncılar için sansürleme yöntemini daha erişilebilir hale getirmesidir. Bu araştırmanın sorusu, "Gerçek zamanlı görüntü işleme ve derin öğrenme yöntemleri kullanarak otomatik sansürleme nasıl yapılabilir ve küçük yayın kuruluşları, yayın girişimleri ve yayıncılar için nasıl daha erişilebilir hale getirilebilir?" şeklinde ortaya konulmuştur. Bu araştırma önerisi, yapay zeka, sansür, gerçek zamanlı görüntü işleme ve derin öğrenme gibi konulara katkı sağlamayı hedeflemektedir. Bu çalışma, derin öğrenme modellerinin gerçek zamanlı görüntü işleme için nasıl kullanılabileceğini ve otomatik sansürleme için ne kadar etkili olabileceğini göstererek, bu alanda önemli bir adım atacaktır. Bu araştırma, gelecekte yayın sektöründe sansürleme işlemini daha verimli ve düşük maliyetli hale getirebilecek birçok fırsat sunmaktadır.

1.2. Amaç ve Hedefler

Bu araştırmanın amacı, gerçek zamanlı görüntü işleme yöntemleri kullanarak otomatik sansürleme yapabilen bir derin öğrenme modeli tasarlamaktır. Bu model, istenmeyen bölümleri otomatik olarak algılayıp sansürleyerek yayın kuruluşlarının ve sosyal medya yayıncılarının iş gücü ve zamanını azaltmayı hedeflemektedir. Bu amaç doğrultusunda, projenin hedefleri şunlardır: Derin öğrenme ve gerçek zamanlı görüntü işleme yöntemlerini kullanarak bir otomatik sansürleme modeli tasarlamak. Modelin gerçek zamanlı olarak çalışabilmesi için performansını optimize etmek. Modelin hassasiyetini artırmak için farklı veri setleri kullanmak ve eğitim sürecini iyileştirmek. Modelin sağlıklı bir şekilde çalışması için tüm teknik sorunları çözmek ve doğru bir şekilde entegre etmek. Modelin yayın sektöründe kullanılabilirliğini ve

erişilebilirliğini artırmak için küçük yayın kuruluşları, yayın girişimleri ve yayıncılar için uygun bir maliyetle sunulabilecek bir hizmet modeli oluşturmak. Bu hedefler, gerçek zamanlı görüntü işleme ve derin öğrenme yöntemlerinin kullanımını artırarak, sansürleme işlemini daha hızlı, daha verimli ve daha düşük maliyetli hale getirmeyi hedeflemektedir. Bu araştırmanın sonucunda, otomatik sansürleme işlemleri için yeni bir yöntem geliştirilecek ve yayın sektöründe sansürleme işlemlerinin daha verimli bir şekilde yapılması sağlanacaktır.

2. YÖNTEM

Bu projede, gerçek zamanlı görüntü işleme yöntemleri kullanarak otomatik sansürleme yapabilen bir derin öğrenme modeli tasarlanmıştır. Bu amaç doğrultusunda, öncelikle sansürlenmek istenilen objeler hakkında görüntüye dayalı veriler toplanmış ve planlanan objeler için derin öğrenme modelleri oluşturulmuş ve sansürleme işlemleri için bu modelleri kullanan bir program hazırlanmıştır. Veri setleri, gerçek zamanlı görüntü işleme için kullanılan en son teknolojilerle oluşturularak modellerin doğruluğunu artırmak için farklı veri setleri kullanılmıştır.

Tablo 1 Derin öğrenme kütüphaneleri

Kütüphane	Dil	Yıl	Açıklama	GitHub Yıldız Sayısı	GPU Çalışma Zamanı Performansı
PyTorch (Collobert vd., 2017)	Python	2017	GPU desteği, hız ve esneklik	45k	2.88
Tensorflow (TensorFlow, 2020)	Python, C++	2020	Otomatik resim yazısı oluşturma yazılımı, tek bir API ile birden fazla GPU ve CPU'ya dağıtma olanağı	152k	5.57
Keras (Keras: The Python Deep Learning API, 2020)	Python	2020	Aktivasyon fonksiyonları ve optimize ediciler	50.4k	-
Digits (NVIDIA DIGITS, 2017)	C++	2017	Çoklu GPU üzerinde verimli, tamamen etkileşimli	4k	-
Theano (Theano Development Team, 2016)	Python	2016	CPU ve GPU üzerinde verimli bir şekilde çalışır, matematiksel hesaplamalar kolay	9.3k	1.40
Caffe (Jia vd., 2014)	Python, C++	2014	Görüntü sınıflandırma ve görüntü bölümlendirme, GPU desteği	31.2k	2.45
Deep learning 4j (DeepLearning4j, 2020)	Java, C, C++, Python	2020	Görselleştirme aracı	11.9k	-

Bu kütüphanelerin arsında GPU desteği olan Theano, TensorFlow, Caffe ve Torch kullanılarak birkaç veri kümesi ve model üzerinde çalışma zamanı performansı karşılaştırılmıştır. Theano en hızlı şekilde çalışmış peşinden Caffe, Torch gelmiş ve en yavaş çalışmış olan ise TensorFlow olduğu görülmüş. Fakat kullanılması en kolay olan ve esnekliği yüksek olduğu için PyTorch seçilmiştir.

Tablo 2 Sınıflandırma yöntemleri

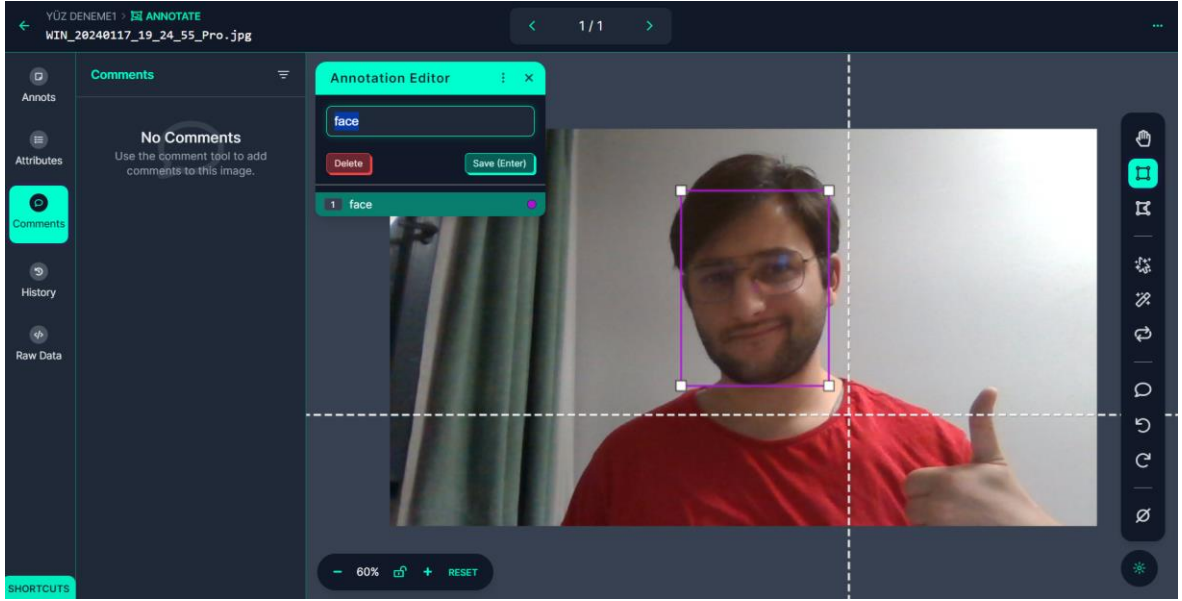
Sınıflandırma Yöntemi	Maliyet	Başarı Oranı	Yorum
Şekil Tabanlı	Düşük	Orta	Işık değişimine hassas. Farklı şekillerde oluşan nesnelerde zayıf.
Hareket Tabanlı	Yüksek	Orta	Farklı karelerdeki nesneleri tanımda başarılı.
Renk Tabanlı	Yüksek	Yüksek	Işık değişimine duyarlı.
Doku Tabanlı	Yüksek	Yüksek	Öznitelik çıkarma aşaması fazla, maliyet yüksek.

Sonuçta baktığımızda ise kullanılması en muhtemel dil Python, kullanılan ekran kartı nvidia markası olduğu için nvidia ile en uyumlu şekilde çalışan cuda kütüphanesi. Derin öğrenme modeli için tensorflow kütüphanesi planlanlanmaktadır. Tasarlanacak veri seti için açık veri setleri ve kendimizin oluşturacağı görüntüler kullanılacaktır Kullanımın kolaylaşması ve teknolojinin erişilebilirliğini artırmak için bir arayüz tasarlanılacak fakat arayüz için kullanılacak program henüz kesin değildir. Projede kullanılması gereken programlar ve projenin işleyişi bu şekilde planlanmıştır.

2.1. Sansürlenmek İstenilen Obje İçin Veri Seti Oluşturulması.

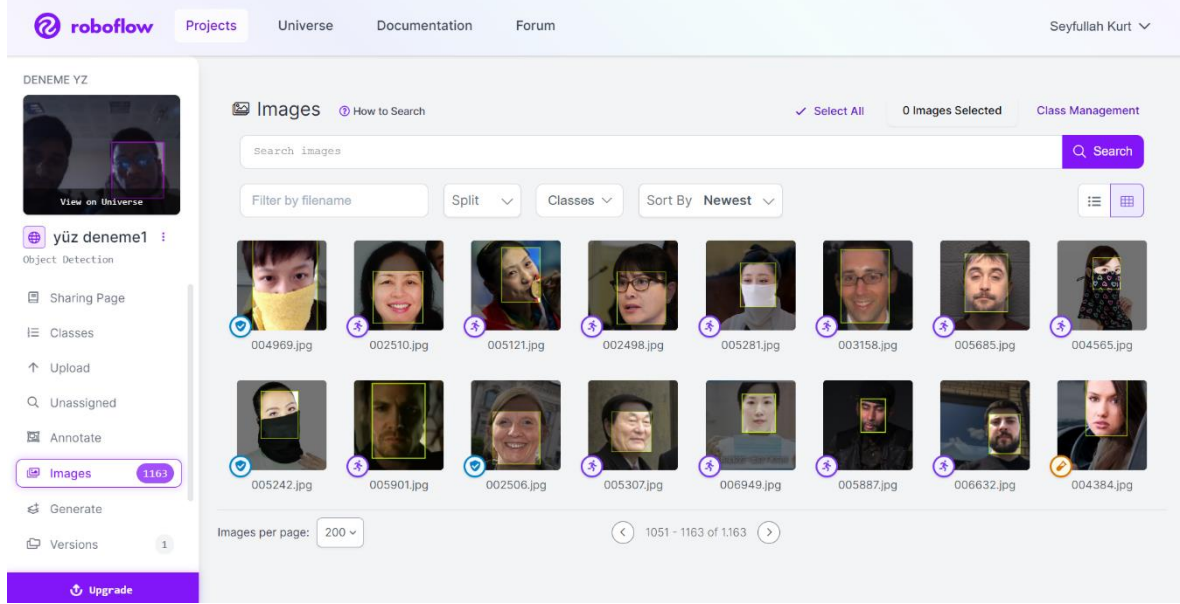
Oluşturacağımız model için öncelikle veri seti oluşturarak başladık bunun için RoboFlow Kullanıldı ve Roboflow'a kendi oluşturduğumuz veriler girildi. Şekil 2 de gözüktüğü gibi yüz veri setimiz için fotoğraf yüklenildi ve istenilen özellik kare içine alındı.

Şekil 1 RoboFlow fotoğraf ekleme ve istenilen özelliği kare içine alma



Sonrasında ise RoboFlow’da bulunan açık verisetleri kullanılarak kendi oluşturduğumuz verisetini Şekil 3 teki gibi geliştirerek modelin doğruluğunu arttırmaya katkıda bulunmaya çalışıldı.

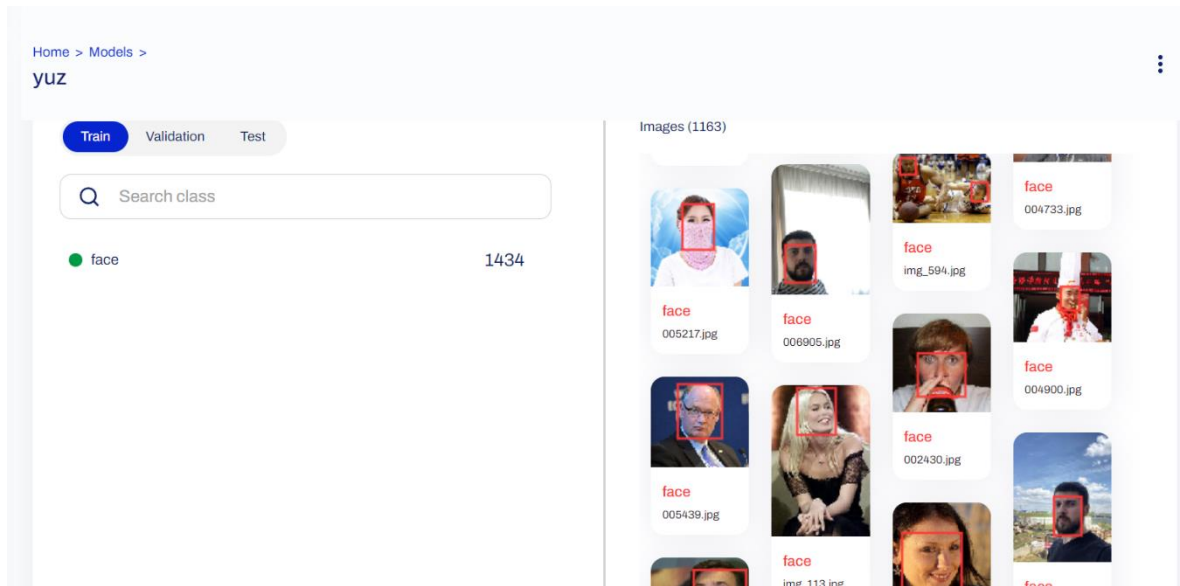
Şekil 2 RoboFlow veri seti



2.2. Sansürlenmek İstenilen Obje İçin Model Oluşturulması.

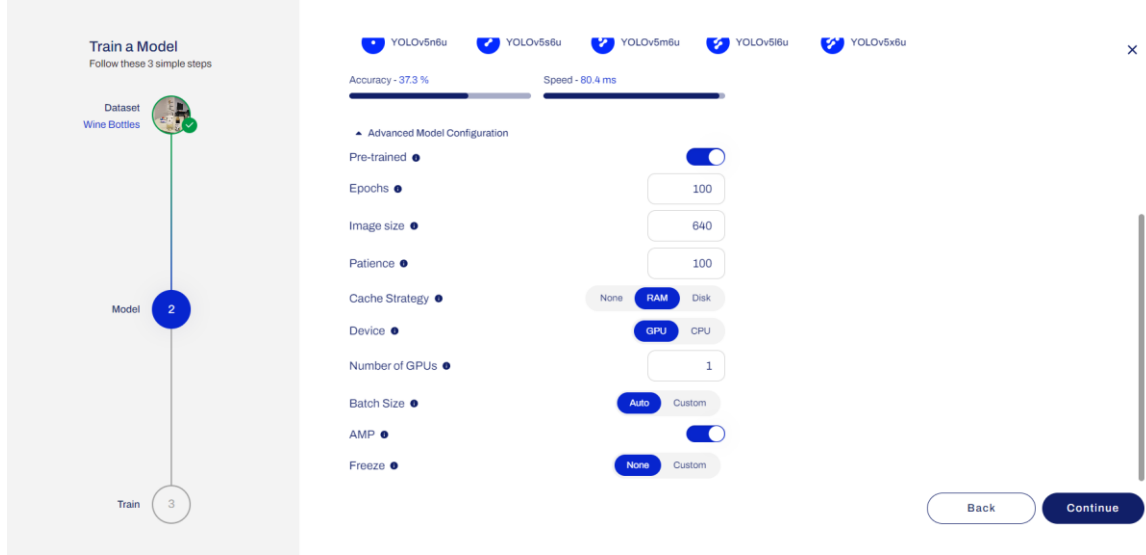
RoboFlow’da oluşturulan veri seti model oluşturmak için Şekil 4 te gösterildiği gibi Ultralytics hub’a yüklendi.

Şekil 3 Ultralytics veri seti



Sonrasında ise bu veriseti ile Şekil 4'te ki gibi model oluşturulmak üzere için en optimize şekilde eğitilebilmesi olabilmelerini sağlamak için ayarlandı ve bu ayarlar için önceden eğitilmiş yolo modellerinden yararlanıldı.

Şekil 4 Ultralytics hub model eğitim ayarları



Sonrasında ise şekil 5'teki gibi modeli eğitebilmek için Ultralytic hub, Google Collab'a bağlanıldı

Şekil 5 Ultralytics ve collab arasında bağlantı kurma komutları



Şekil x teki gibi collab üzerinde Ultralyticy kuruldu ve içeri aktarıldı

Şekil 6 Ultralytics'i collaba indirme ve içeri aktarma

```
✓ 18 sn. Setup

Pip install ultralytics and dependencies and check software and hardware.

%pip install ultralytics # install
from ultralytics import YOLO, checks, hub
checks() # checks

Ultralytics YOLOv8.1.2 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 26.3/78.2 GB disk)
```

Sonrasında ise şekil 7’deki gibi Yolo girişi yapıldı ve Model eğitime başlandı ve yaklaşık 1 gün içerisinde 100 epoch ile eğitim tamamlandı.

Şekil 7 Hub login ve model eğitimi komutları ve çıktıları

```
hub.login('2cb7afd51d6d62ebdbfa0d76a61b9619f2422a8a3')

model = YOLO('https://hub.ultralytics.com/models/XJ1zXVvds4YaHZPhoL4')
results = model.train()

... Ultralytics HUB: New authentication successful ✓
Ultralytics HUB: View model at https://hub.ultralytics.com/models/XJ1zXVvds4YaHZPhoL4
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt to 'yolov8n.pt'...
100%| 6.23M/6.23M [00:00<00:00, 77.5MB/s]
Ultralytics YOLOv8.1.2 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=https://app.roboflow.com/ds/yH4EuT1u9e?key=DxOzxPalp6, epochs=100, ti
Downloading https://app.roboflow.com/ds/yH4EuT1u9e to 'yH4EuT1u9e'...
100%| 149M/149M [00:01<00:00, 91.1MB/s]
Unzipping yH4EuT1u9e to /content/datasets/yH4EuT1u9e...: 100%| 462/462 [00:00<00:00, 524.82file/s]
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%| 755k/755k [00:00<00:00, 15.8MB/s]
Overriding model.yaml nc=80 with nc=1
```

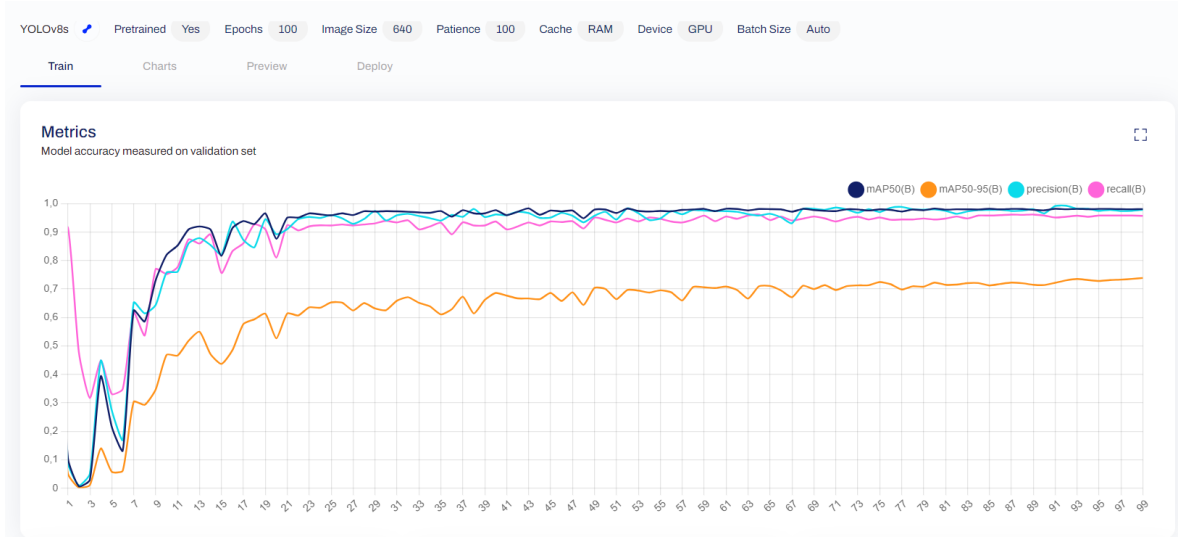
Eğitim sürecinin çıktısı Şekil 8’de belirtilmiştir.

Şekil 8 Eğitim sürecinin çıktısı

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
7/100	10.9G	1.099	1.014	1.053	594	640: 100% [00:02<00:00, 1.49it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% [00:00<00:00, 0.915 0.636]
all	34	540	0.919	0.507		
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
8/100	11.9G	1.064	0.9355	1.018	677	640: 100% [00:02<00:00, 1.30it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% [00:00<00:00, 0.846 0.618]
all	34	540	0.861	0.297		
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
9/100	10.8G	1.003	0.8709	0.9919	551	640: 100% [00:01<00:00, 1.51it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% [00:00<00:00, 0.771 0.583]
all	34	540	0.849	0.285		
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
10/100	11.2G	0.9814	0.8951	0.992	570	640: 100% [00:02<00:00, 1.50it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100% [00:00<00:00, 0.472 0.345]
all	34	540	0.755	0.254		

Sonuç olarak Grafik 1 ve Grafik 2’de görünen eğitim sonuçlarına sahip modele ulaşıldı ve bu sürecinin tamamını kopyalayan farklı özellikler için 3 model daha oluşturuldu.

Grafik 1 Epoch sayısına göre modelin doğruluk oranları grafiği



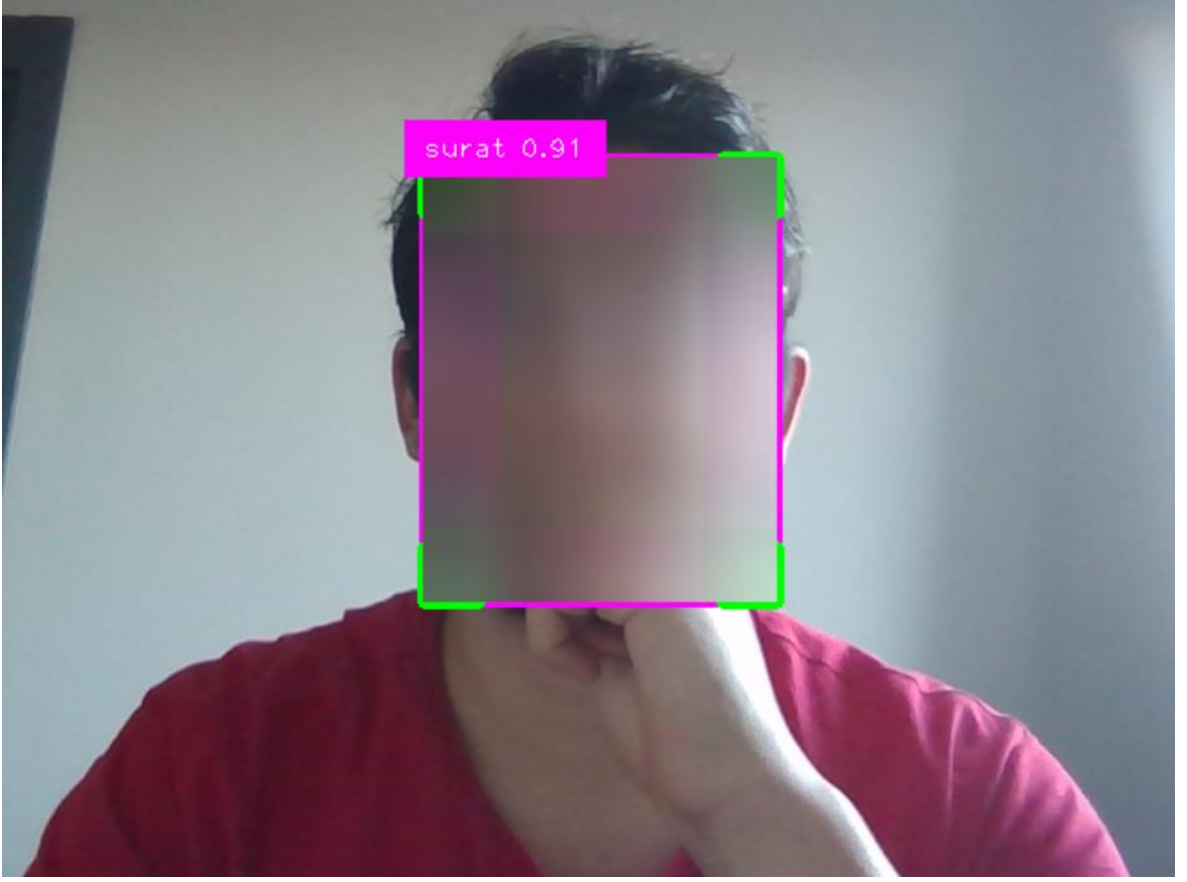
Grafik 2 Epoch sayısına göre modelin kayıp grafiği



2.3. Sansürlenmek İstenilen Objenin Algılanması ve Sansürlenmesi

Elde ettiğimiz modeller çalıştırılarak modelin sınıflandırma algoritmasında bize ne gibi sonuçlar vereceği test edildi. Planlanan algoritma ve kütüphaneleri kullanarak kişilerin yüz takibi ve sansürlenmesi uygulanmaya çalışıldı. Sonuçlar Şekil 9'dan Şekil 14'e kadar gösterilmiştir

Şekil 9 Yüz tespiti ve sansürlenmesi



Şekil 10 Yüz sansürlenmesi



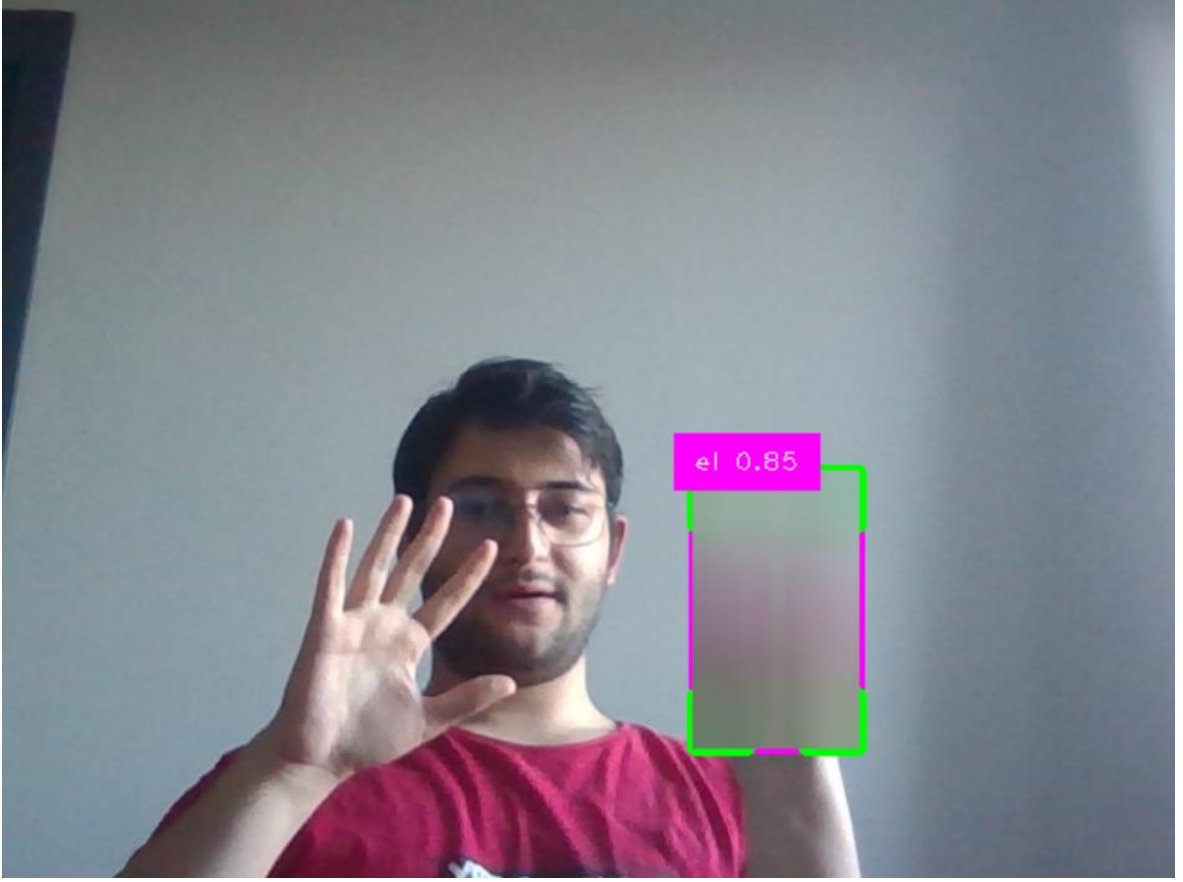
Şekil 11 Sigara tespiti ve sansürlenmesi



Şekil 12 Sigara sansürlenmesi



Şekil 13 Hakaret içeren el hareketi tespiti ve sansürlenmesi



Şekil 14 Hakaret içeren el hareketi sansürlenmesi



Sonuç ve Öneriler

Sonuç olarak elimizde İstedğimiz özellikleri sansürlememizi sağlayacak 4 model oldu ve bu modelleri gerçek zamanlı çalıştırabileceğimiz bir algoritma tasarlandı. Bu modeller gelecekte sansürleme işleminin maliyetlerini düşürecek şekilde kullanılması planlanmaktadır.

Kaynakça

- (tarih yok). *Temelli, S. C. (2014). Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması (Yüksek Lisans Tezi).*
- Alya'a R. Ali*, Ban N. Dhannoon. (2019). Real Time Multi Face Blurring on Uncontrolled Environment. (tarih yok).
- Güler, O. &. (tarih yok).
- <https://www.innova.com.tr/tr/blog/buyuk-veri-blog/derin-ogrenme-hakkinda-bilmeniz-gereken-her-sey>. (tarih yok).
- Tan, F. G., Yüksel, A. S., Aydemir, E., & Ersoy, M. (2021). Derin Öğrenme Teknikleri İle Nesne Tespiti Ve. (tarih yok).
- TOMBAK, M. E. (2019). Python ve OpenCV ile Yüz Tanıma ve Otomatik Blur Uygulaması (Yüksek Lisans. (tarih yok).