

2209/A

ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEK
PROGRAMI

SONUÇ RAPORU

PROJE BAŞLIĞI: Görüntüdeki Zararlı İçerikleri Sansürleme

PROJE YÜRÜTÜCÜSÜNÜN ADI: Gürkan GÜR

DANIŞMANININ ADI: Erhan TURAN

GENEL BİLGİLER

PROJENİN KONUSU	Görüntüdeki Zararlı İçerikleri Sansürleme
PROJE YÜRÜTÜCÜSÜNÜN ADI	Gürkan GÜR
DANIŞMANIN ADI	Erhan TURAN
PROJE BAŞLANGIÇ VE BİTİŞ TARİHLERİ	29.09.2023 – 20.09.2024

PROJE YÜRÜTÜCÜSÜNÜN ADI – SOYADI - İMZA	DANIŞMANIN ADI – SOYADI - İMZA
Gürkan GÜR	Erhan TURAN

Tarih : 20.09.2024

[1] Proje Sunum Linki: <https://github.com/2gur1kan/Harmful-Content-Censorship-Practice>

1.GİRİŞ

Günümüz dijital dünyasında, video ve görsel içeriklerin hayatımızdaki yeri her geçen gün derinleşirken, bu içeriklerin yaygınlaşmasıyla birlikte zararlı unsurların da hızla yayılmaya başladığı bir gerçektir. Özellikle sigara gibi sağlığa ciddi zararlar veren alışkanlıkların dijital içeriklerde normalleştirilmesi, toplum sağlığını tehdit eden sinsi bir tehlike olarak karşımıza çıkmaktadır. Sigaranın yalnızca fiziksel sağlığı değil, aynı zamanda zihinsel ve sosyal sağlığı da olumsuz etkilediği bilinirken, bu tür içeriklerin genç bireyler üzerinde kötü alışkanlıkların yerleşmesine yol açma riski göz ardı edilmemelidir. Dijital platformlarda bu tür zararlı unsurların tespiti ve sansürlenmesi, bu bağlamda büyük önem taşımaktadır.

Ancak, genellikle sigara gibi spesifik içeriklerin sansürlenmesi profesyonel televizyon ve video işleme yazılımlarının yetkinlik alanında kalmakta ve bu tür yazılımlar genellikle yüksek maliyetlerle erişilebilir durumdadır. Gelişen derin öğrenme modelleri ise, bu tür işlemleri genel kullanıcıların erişimine açabilecek sade ve ücretsiz çözümler sunma potansiyeline sahiptir. Bu proje, işte bu noktada devreye giriyor. Amacımız, mevcut video dosyalarında sigara görüntülerini tespit ederek, bu içerikleri bulanıklaştırma yoluyla izleyicilerden gizleyen bir uygulama geliştirmek. Üstelik, bu uygulamayı açık kaynaklı ve ücretsiz bir şekilde sunarak, daha geniş bir kullanıcı kitlesine ulaşmayı hedefliyoruz. Böylece, yalnızca profesyonel yayıncıların değil, sıradan kullanıcıların da dijital içeriklerinde zararlı unsurları sansürleyebilmelerine olanak tanıyoruz.

Geliştirilen bu uygulama, sigara gibi zararlı içeriklerin tespit edilmesi ve ardından bulanıklaştırılması için derin öğrenme modellerini devreye sokmaktadır. Başlangıçta sigara nesnesine odaklanılmış olsa da, benzer yöntemlerle başka nesneler üzerinde de uygulanabilir bir yapıya sahiptir. Derin öğrenme dünyasında, eğitilen modellerin başarısı, büyük ölçüde kullanılan veri setlerinin niteliğine bağlıdır. Bu projede, veri seti sürekli olarak yenilenmiş ve model eğitimleri, bu güncellemeler doğrultusunda rafine edilmiştir. Ancak, mevcut sınırlı kaynaklar doğrultusunda oluşturulan veri setinin, daha iyi sonuçlar elde edebilmek için gelecekte daha da geliştirilmesi gerektiği açıktır.

Bu çözüm, dijital içeriklerin güvenliğini artırmak adına sigara gibi zararlı maddeleri algılayıp, görünmeden önce bulanıklaştırarak, özellikle genç bireylerin maruz kalabileceği olumsuz etkileri en aza indirmeyi hedeflemektedir. Aynı zamanda, içerik üreticileri ve tüketicileri için daha güvenli bir dijital deneyim sunarak, toplumsal sağlığı koruma yolunda küçük ama önemli bir adım atmaktadır.

2. RAPOR DÖNEMLERİNDE YAPILAN ÇALIŞMALAR

Projenin ilk aşaması olarak, geliştirilecek yapay zeka modeli için uygun bir veri seti oluşturulması gerektiği fark edilmiştir. Bu doğrultuda, internet üzerindeki farklı kaynaklar detaylı bir şekilde incelenmiş, ancak karşılaşılan veri setlerinin büyük bir çoğunluğunun telif hakkına tabi olması, kullanım açısından yasal engeller ortaya çıkarmıştır. Bu sorunun üstesinden gelebilmek amacıyla alternatif bir yöntem arayışına girilmiş ve veri setinin sıfırdan oluşturulmasının en uygun çözüm olacağına karar verilmiştir.

Veri seti oluşturma sürecinde, gönüllülerden yardım alınarak çeşitli görseller toplanmış ve aynı zamanda film sahnelerinden elde edilen kareler de bu sürece dahil edilmiştir. Görsellerin dikkatli bir şekilde seçilmesine özen gösterilmiş, yalnızca sigara içen kişilerin değil, aynı zamanda sigaraya benzeyen nesneler (kalem, pipet, çubuk vb.) tutan kişilerin de dahil olduğu kareler tercih edilmiştir. Bu sayede, modelin yalnızca sigara algılamasının ötesinde, sigaraya benzer nesneleri yanlışlıkla sigara olarak varsaymasının önüne geçilmesi hedeflenmiştir.

Toplanan bu görseller, Ultralytics şirketinin YOLO (You Only Look Once) V5 ve V8 modelleri için kullanılabilir hale getirilmiştir. YOLO, nesne tespiti için popüler bir derin öğrenme modeli olduğundan, verilerin bu modelin gereksinimlerine uygun şekilde düzenlenmesine özen gösterilmiştir. Bu süreçte, her bir görsel titizlikle etiketlenmiş ve modelin eğitimi için gerekli formatlara dönüştürülerek veri seti yapılandırılmıştır. Ancak, mevcut veri sayısının yetersiz olduğu fark edilmiştir. Bu nedenle, proje süreci boyunca ek veri arayışına devam edilerek veri setinin zenginleştirilmesi amaçlanmıştır.

Danışman hocanın tavsiyeleri doğrultusunda, projenin kullanıcı arayüzü (UI) tasarımının Qt Designer kullanılarak gerçekleştirilmesine karar verilmiştir. Bu aracın, arayüz geliştirme sürecini hızlandırdığı ve görsel düzenlemeleri daha kolay hale getirdiği göz önünde bulundurularak tercih edilmiştir. Tasarım sürecine başlamadan önce, uygulamanın hangi işlevleri yerine getirmesi gerektiğine dair kapsamlı notlar alınmış ve bu notlar, projenin temel yapısının belirlenmesinde yol gösterici olmuştur.

Benzer işlevlere sahip uygulamaların arayüzleri incelenmiş ve yapılan bu incelemeler sonucunda, projede eksik olabileceği düşünülen özellikler ve değişkenler tespit edilmiştir. Başlangıçta gözden kaçan ek işlevler ve değişkenler projeye entegre edilerek, uygulamanın daha işlevsel ve kullanıcı dostu bir yapıya sahip olması hedeflenmiştir.

Kullanıcı arayüzü tasarımında, her bir bileşenin dikkatle yerleştirilmesine özen gösterilmiştir. Tasarımın, kullanıcılar açısından kolay anlaşılır ve etkili olmasına özellikle dikkat edilmiştir.

Bu bağlamda, butonların yerleşimi, menülerin düzeni ve görsel unsurların uyumu gibi detaylar titizlikle ele alınmıştır. Kullanıcı deneyiminin iyileştirilmesi amacıyla, her bileşenin mantıklı ve akıcı bir şekilde yerleştirildiği bir yapı oluşturulmuştur.

Tasarım sürecinde, kullanıcıların minimum çaba ile en verimli sonuçları elde edebilmeleri sağlanmaya çalışılmıştır. Bu doğrultuda, bileşenlerin boyutları, renkleri ve konumları optimize edilmiştir. Tasarımın hem estetik hem de işlevsel yönleri dengelenmiş, kullanıcının rahatça etkileşimde bulunabileceği bir arayüz oluşturulmuştur.

Tasarım tamamlandıktan sonra, uygulamanın işlevlerinin düzenlenmesi sürecine geçilmiştir. Bu aşamada, her bir arayüz bileşeni, ilgili işlevlerle bağlantılandırılmıştır. Örneğin, butonlara tıklanıldığında gerçekleştirilecek işlemleri belirleyen kodlar yazılmış ve hata durumlarında kullanıcılara anlamlı geri bildirim sağlayan mesajlar eklenmiştir.

Son olarak, tüm bileşenler bir araya getirilerek uygulamanın faal hale gelmesi sağlanmıştır. Uygulamanın sorunsuz çalıştığından emin olunabilmesi için çeşitli testler yapılmış ve gerekli iyileştirmeler gerçekleştirilmiştir. Bu süreçte karşılaşılan zorluklar gözden geçirilmiş ve danışman hocadan alınan geri bildirimler doğrultusunda uygulamaya ek düzenlemeler yapılmıştır.

Projenin ilerleyen aşamalarında, yapay zeka modellerinin eğitimi, sistem kapasitesi doğrultusunda başlatılmıştır. Model eğitim süreci, projenin en kritik aşamalarından biri olarak ele alınmış ve bu adım titizlikle yürütülmüştür. İlk olarak, sistemin donanım kaynakları ve işlem gücü dikkate alınarak, model eğitimlerinin optimize edilmesi sağlanmıştır. Eğitim sürecinin mümkün olduğunca verimli hale getirilmesi amacıyla uygun parametreler belirlenmiş ve her bir model için en iyi sonuçların elde edilmesine odaklanılmıştır.

Bu süreçte, farklı model yapıları denenmiş ve eğitimler tekrarlanmıştır. İlk eğitimlerde elde edilen sonuçlar, belirlenen hedeflere tam olarak ulaşmadığından, model parametreleri ve veri seti üzerinde düzenlemeler yapılmış, eğitimler birkaç kez yinelenmiştir. Eğitim süreçlerinde karşılaşılan sorunlar analiz edilmiş ve her tekrarda modeli daha iyi hale getirmek için gerekli değişiklikler uygulanmıştır.

Modelin performansını artırmak amacıyla, hiperparametre ayarları gözden geçirilmiştir. Öğrenme oranı, epoch sayısı gibi parametrelerde çeşitli kombinasyonlar denenmiştir. Ayrıca, modelin doğruluğunu artırmak için veri setine yönelik çeşitli veri ön işleme teknikleri uygulanmıştır. Veri artırma (data augmentation) yöntemleriyle modelin daha genel geçer sonuçlar üretmesi sağlanmış ve ek veri setleri oluşturulmuştur.

Her model eğitimi sonrasında, elde edilen sonuçlar dikkatle incelenmiş ve modelin performansı değerlendirilmiştir. Modelin doğruluk, hassasiyet ve hatırlama (recall) gibi

performans metriklerine odaklanılmış ve bu metrikleri iyileştirmek için gerekli düzenlemeler yapılmıştır. Modelin en iyi performansı gösterdiği eğitim konfigürasyonları kaydedilmiş ve bu konfigürasyonlar üzerinden ilenmiştir.

Model eğitim sürecinde, tek bir modelle sınırlı kalınmamış; farklı mimariler ve algoritmalar denenmiştir. Bu yöntemle, projenin amaçlarına en uygun model belirlenmiştir. Ayrıca, her eğitim sürecinde modelin aşırı öğrenme (overfitting) gibi olumsuz durumlarla karşılaşmaması için dikkatli bir şekilde çalışılmıştır.

3.SONUÇ

Aşağıdaki görsellerde, farklı yapay zeka modellerinin ve bu modellerin sahip olduğu özelliklerin, yapılan testlerde göstermiş oldukları performans sonuçları yer almaktadır. Bu performans değerlendirmeleri, her modelin çeşitli kriterlere göre test edilmesi ve sonuçların analiz edilmesiyle elde edilmiştir. Her bir model, belirli bir eğitim sürecinden geçirilmiş ve bu süreçlerin sonucunda elde edilebilen en iyi performans durumları görsellerde iliştilmiştir.

Modellerin eğitimi ve performans testleri sırasında, sınırlı sistem kaynaklarıyla çalışılmıştır. Özellikle, 4 GB sanal belleğe sahip Nvidia RTX 3050 Ti ekran kartı ile donatılmış bir dizüstü bilgisayar kullanılmıştır. Bir noktadan sonra donanım yetersiz kaldığı için 8 GB sanal belleğe sahip Nvidia GTX 1080 içeren bir sistem ile çalışmalara devam edilmiştir. Bu donanım, yoğun hesaplama gerektiren derin öğrenme işlemleri sırasında önemli bir rol oynamıştır. Ancak, sınırlı bellek ve işlem kapasitesi nedeniyle bazı zorluklar yaşanmıştır. Buna rağmen, modellerin mümkün olan en iyi sonuçları vermesi için eğitim süreçleri dikkatlice optimize edilmiştir.

Modellerin performansını değerlendirirken, çeşitli metrikler göz önünde bulundurulmuştur. Bu metrikler arasında doğruluk (accuracy), hassasiyet (precision), hatırlama (recall) ve F1 skoru gibi önemli ölçütler bulunmaktadır. Ayrıca, modellerin eğitim süreleri, bellekte kapladıkları alan ve işlem yükleri de analiz edilmiştir. Tüm bu faktörler, model seçimi ve performans değerlendirmesi sürecinde kritik bir rol oynamıştır.

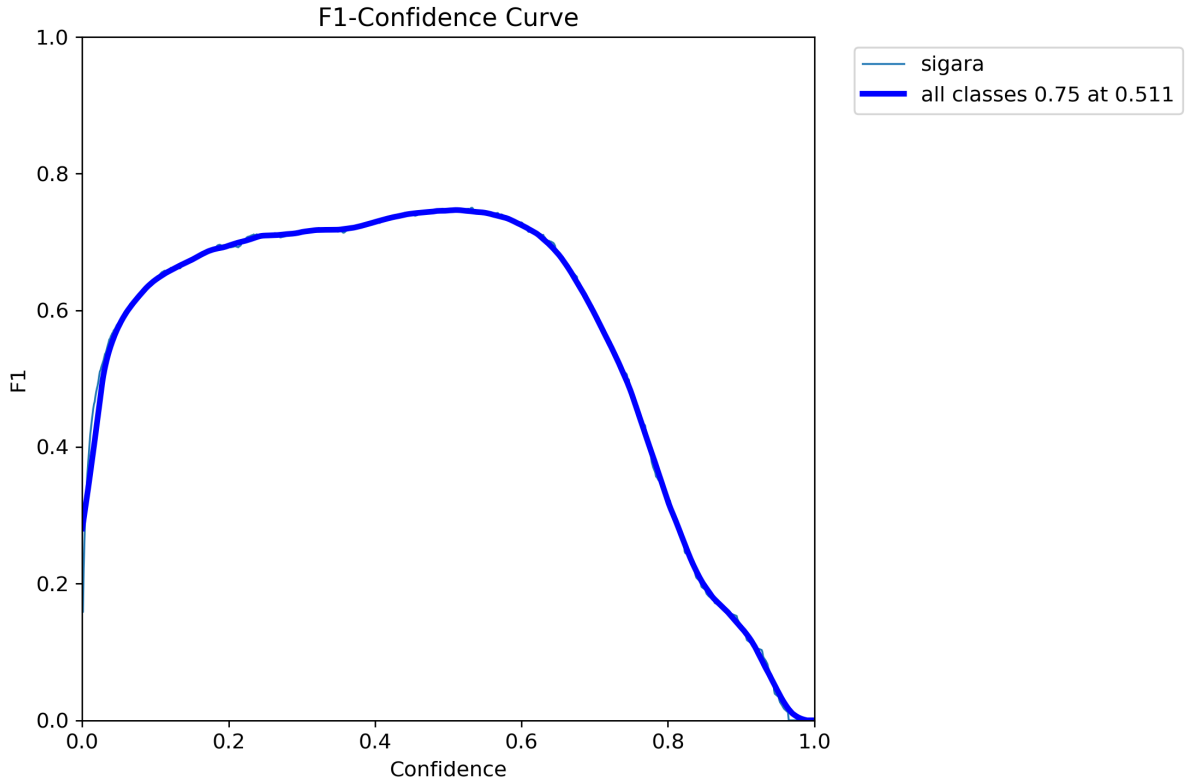
Elde edilen sonuçlar, her modelin farklı senaryolarda nasıl performans gösterdiğini anlamamıza yardımcı olmuş ve projeye en uygun modelin seçilmesi konusunda yol gösterici olmuştur. Bu süreçte karşılaşılan zorluklar, sistem kaynaklarının sınırları dahilinde çözümler üretilerek aşılmış ve projeye değerli bir katkı sağlanmıştır.

3.1. F1 Skorları

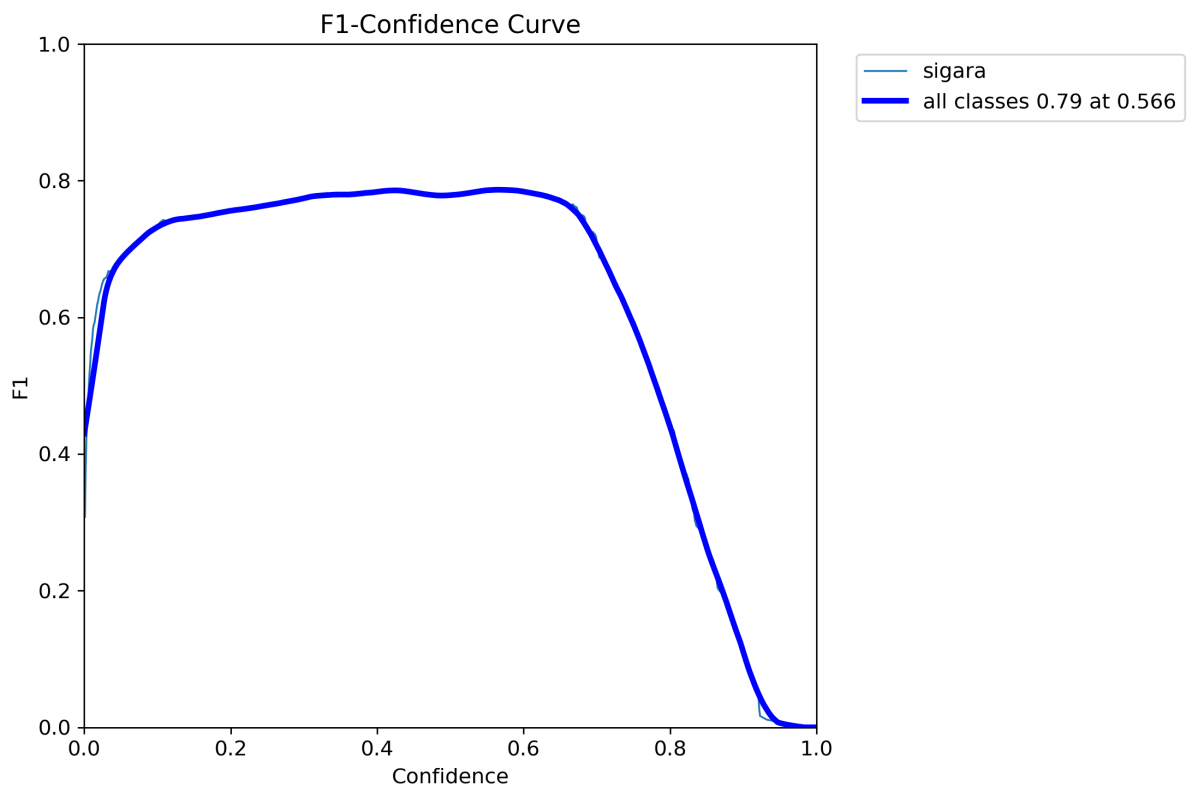
Model eğitimi sırasında yapılan çeşitli denemeler sonucunda, en istikrarlı sonuçları veren yapılandırmanın 100 epoch olduğu tespit edilmiştir. Eğitimler yapılırken auto batch sistemi kullanılmıştır. Bu sistem modellerin eğitimi sırasında sistemin kaldırabileceği en iyi batch değerlerini sistem girilmesine yardımcı bulunmuştur. Deneyler sırasında elde edilen sonuçlar arasındaki farklar minimal düzeyde olsa da, bu parametrelerle elde edilen sonuçlar en yüksek başarı seviyesine ulaşmıştır. Bu nedenle, eğitim sonuçları makalede belirtilen bu parametreler çerçevesinde sunulmaktadır.

Yapılan model eğitimleri sonucunda, en iyi performansın YoloV8 modeli ile eğitilen modellerde elde edildiği gözlemlenmiştir. YoloV5 ile karşılaştırıldığında, YoloV8'in daha yüksek karmaşıklığa sahip modellerde üstün performans sergilediği tespit edilmiştir. Eğitim süreçlerinden elde edilen en iyi sonuçlar aşağıda sunulmuştur.

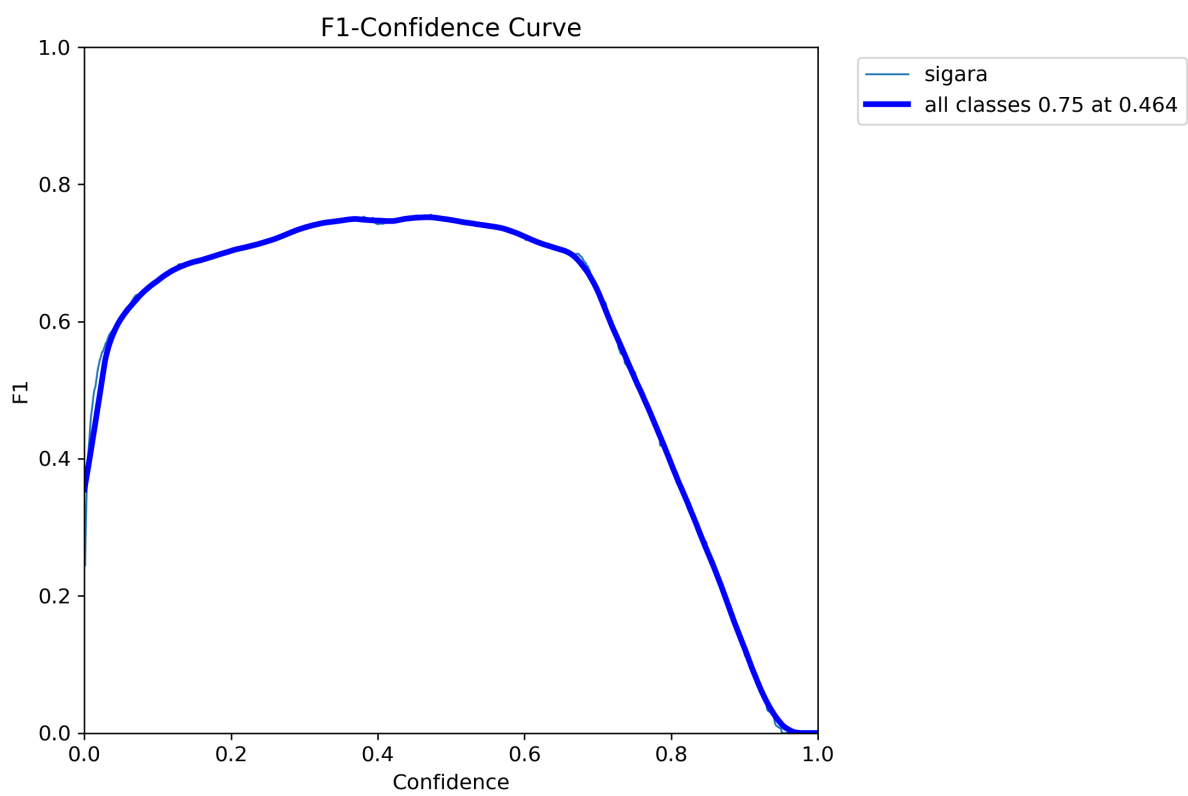
F1 skorlarının hesaplanması için kullanılan görseller oluşturulan veri setinin içerisinden seçilmiştir.



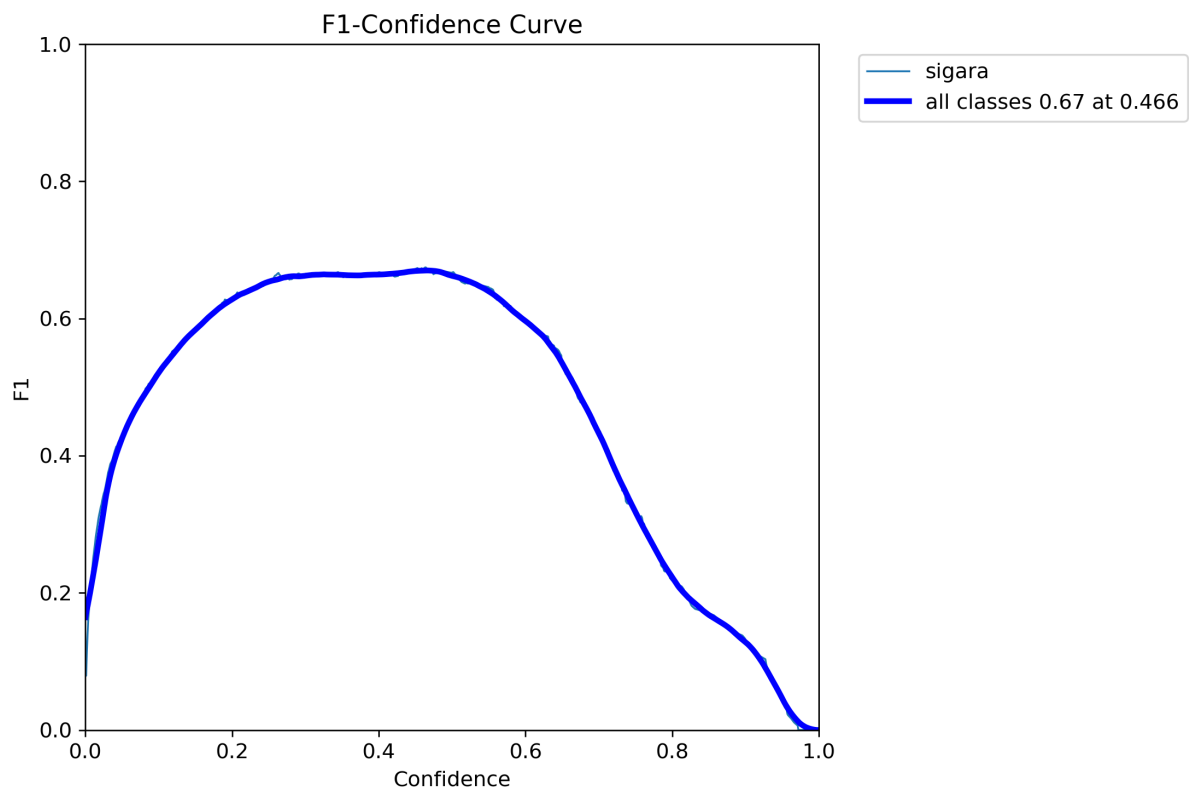
YoloV5nu: 100 epoch, 0.75 accuracy.



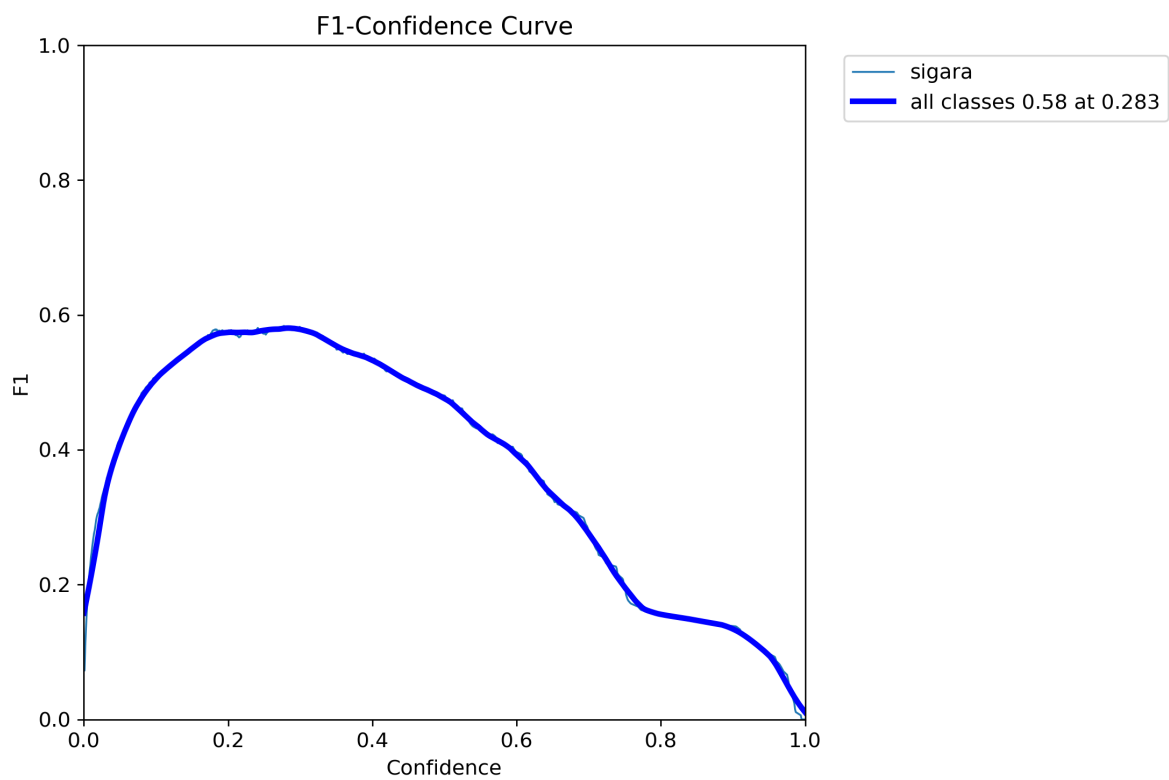
YoloV5su: 100 epoch, 0.79 accuracy.



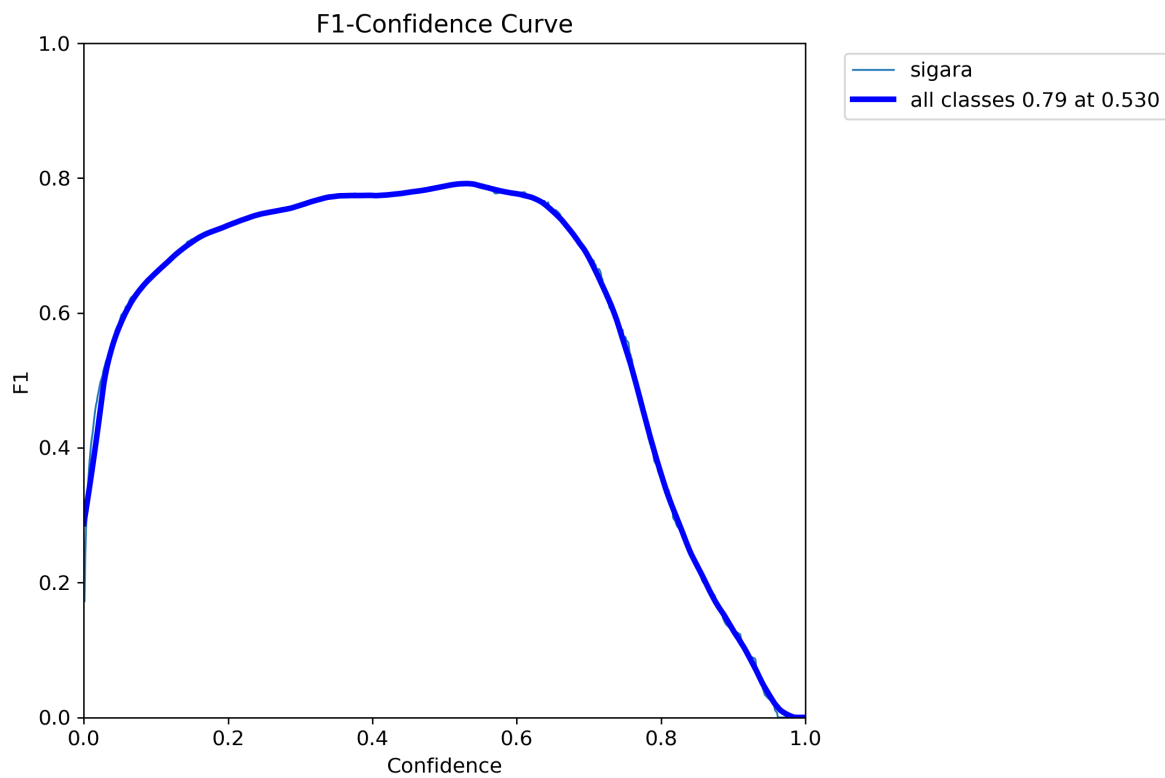
YoloV5mu: 100 epoch, 0.75 accuracy.



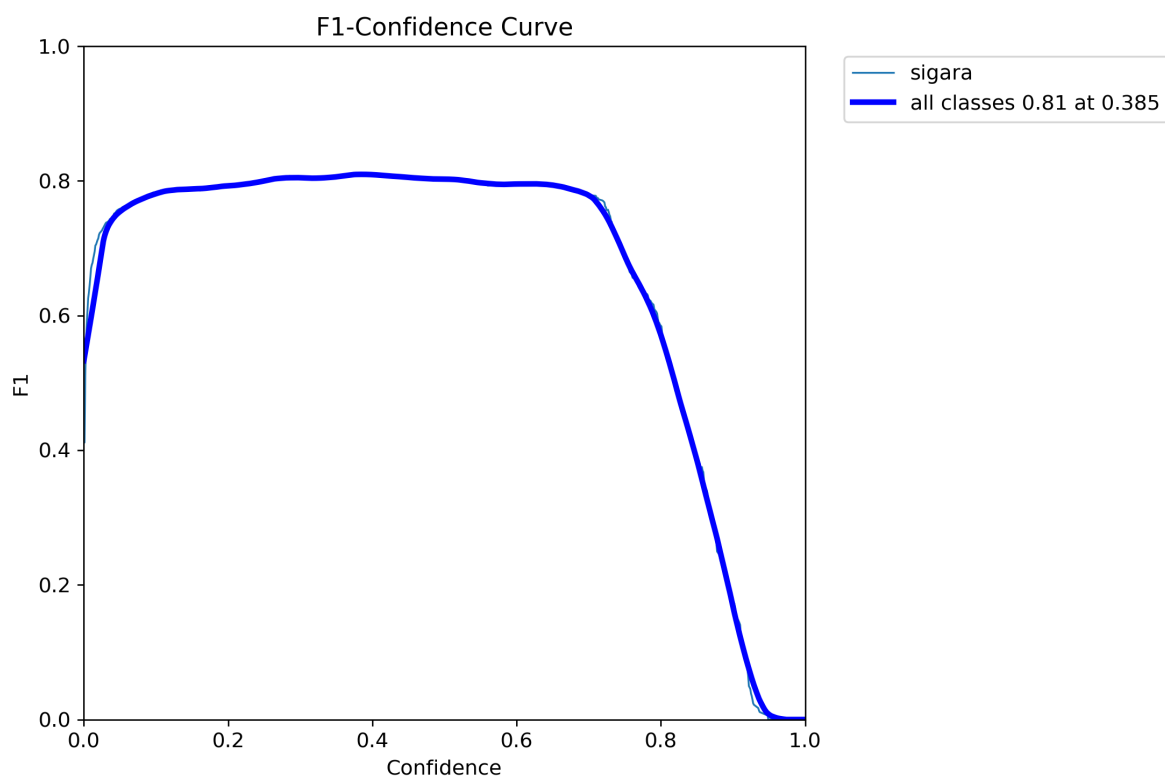
YoloV5lu: 100 epoch, 0.67 accuracy.



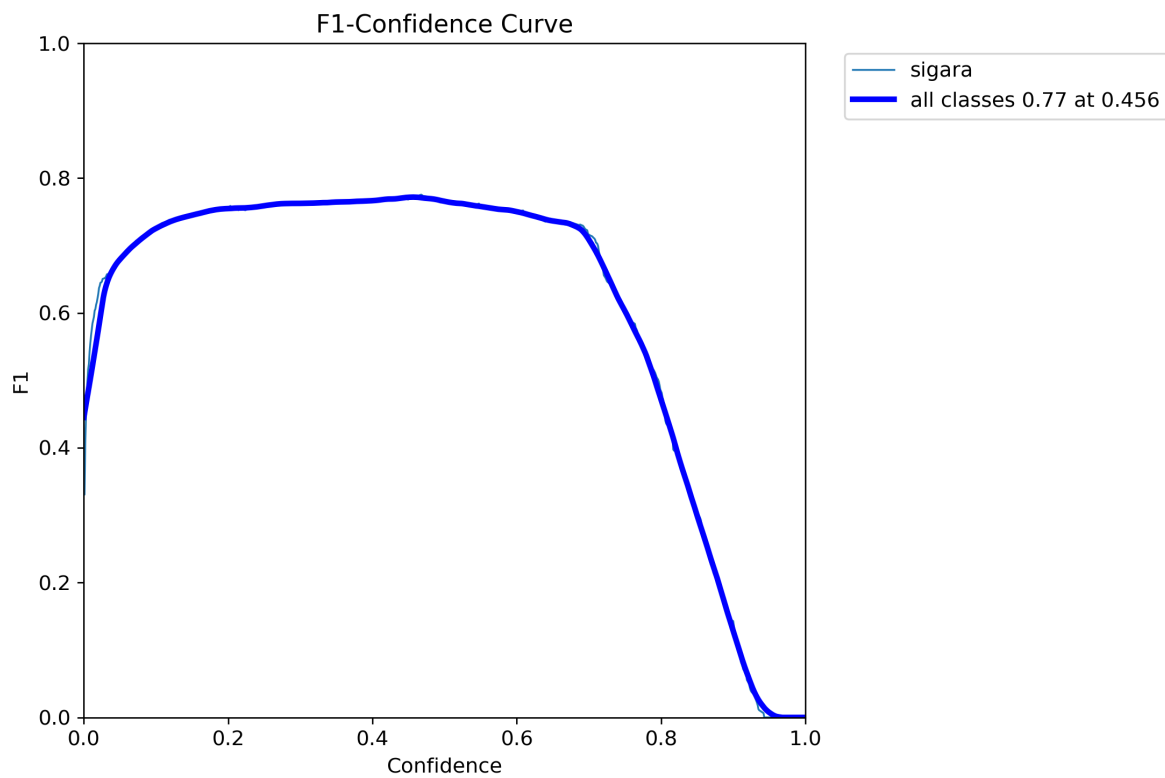
YoloV5xu: 100 epoch, 0.58 accuracy.



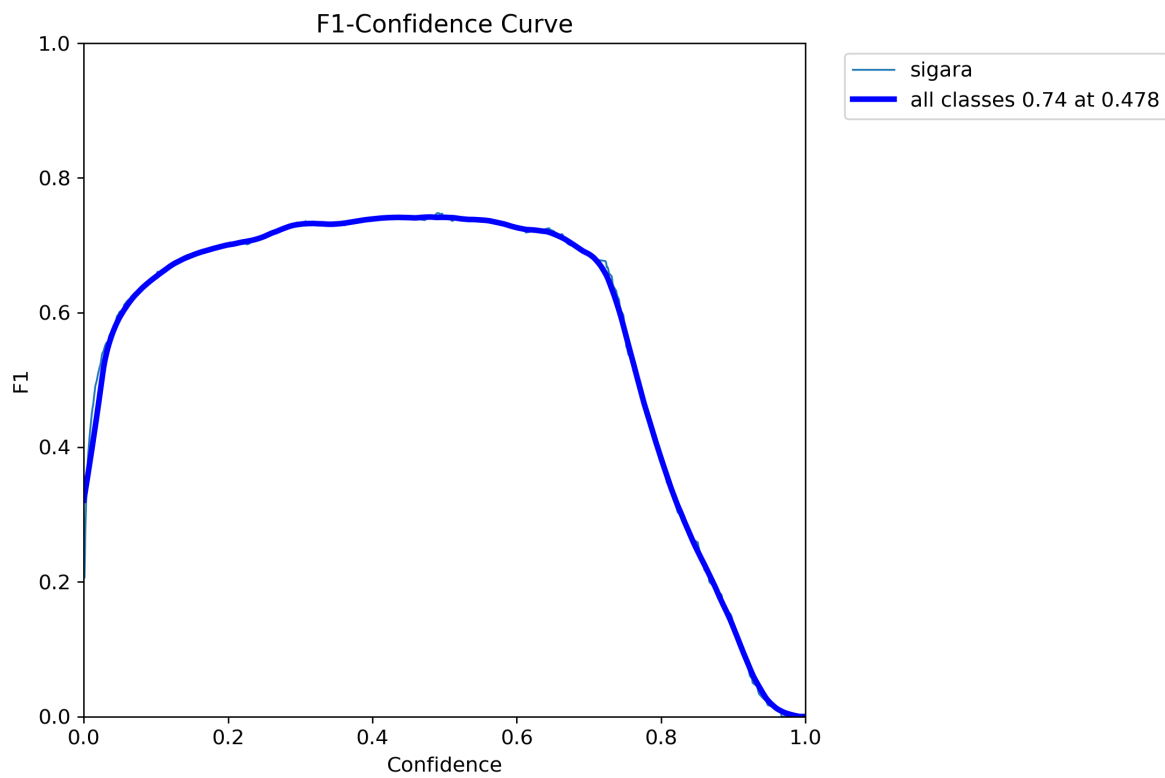
YoloV8n: 100 epoch, 0.79 accuracy.



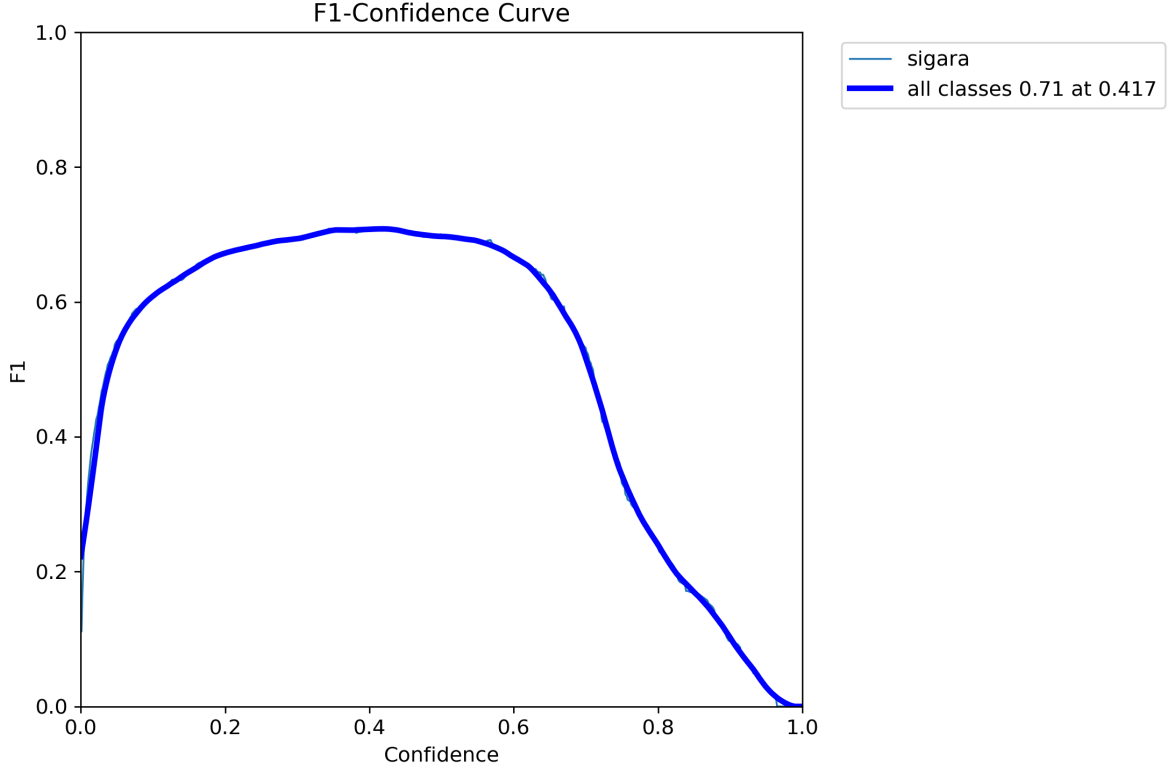
YoloV8s: 100 epoch, 0.81 accuracy.



YoloV8m: 100 epoch, 0.77 accuracy.



YoloV8l: 100 epoch, 0.74 accuracy.

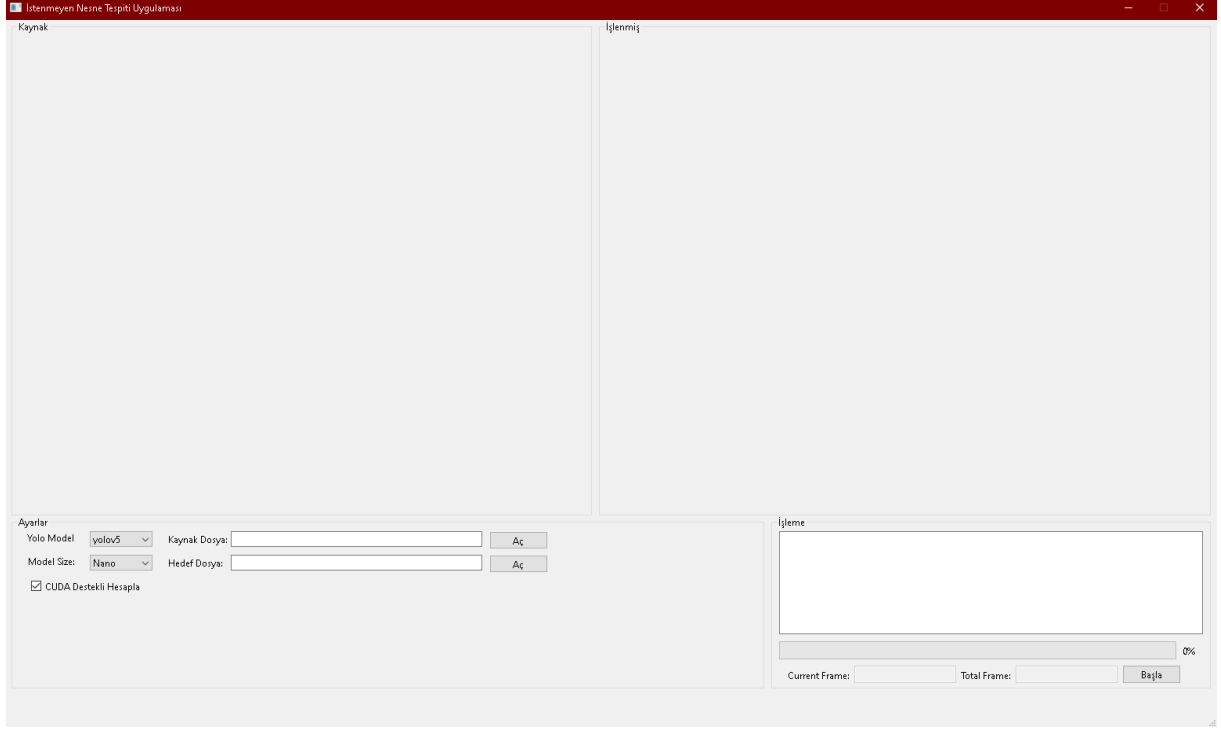


YoloV8x: 100 epoch, 0.71 accuracy.

3.2. Uygulama

Uygulama başlatıldığında, öncelikli olarak kontrol paneli yüklenmektedir. Bu aşamada, eğer sistemde CUDA desteği mevcutsa, "CUDA Destekli Hesaplama" seçeneği otomatik olarak etkinleştirilmekte ve varsayılan olarak işaretlenmektedir. Yapılan testlerde, CUDA destekli hesaplama ile CPU tabanlı hesaplama arasında yaklaşık %80 oranında bir performans farkı tespit edilmiştir. Bu nedenle, "CUDA Destekli Hesaplama" seçeneğinin varsayılan olarak aktif hale getirilmesi önerilmektedir.

Uygulamada, kullanıcıların ihtiyaçlarına en uygun modeli seçebilmesi ve test süreçlerini kolaylaştırabilmesi amacıyla bir model seçme opsiyonu sunulmaktadır. Bu seçenek, modelin karmaşıklığına göre değişen performans ve işlem süreleri arasında denge kurulmasına olanak tanır. Model seçenekleri, Nano'dan Xlarge'a doğru genişleyen bir skalada sunulmakta olup, model karmaşıklığı arttıkça işlem süresi de orantılı olarak uzamaktadır. Yapılan testlerde, her bir model değişimiyle birlikte, işlem sürelerinde yaklaşık iki katına varan artışlar gözlemlenmiştir.



Uygulama Arayüzü

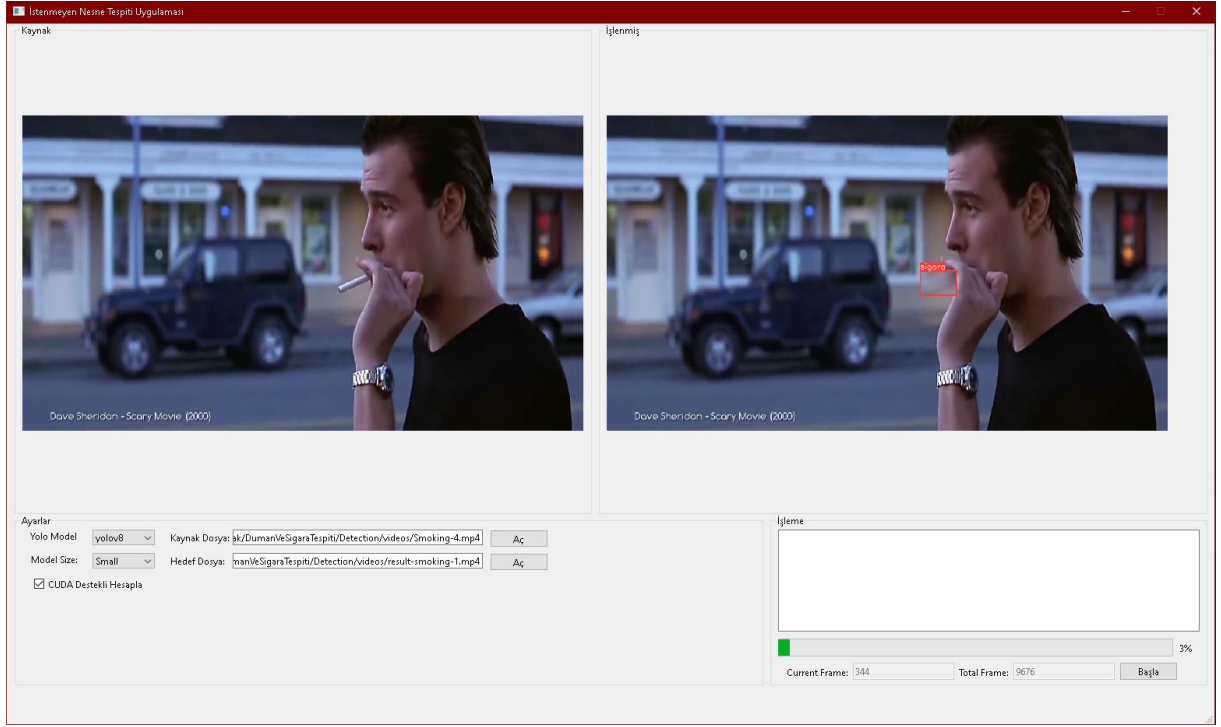
Örneğin, 360 piksel çözünürlükte ve 30 fps hızında, 10 dakikalık bir videonun sansürlenme süreci, Nano modelinde ortalama 15 dakika sürerken, Small modelinde bu süre 25 ila 30 dakikaya kadar çıkabilmektedir. Ancak, bu sürelerin sistemden sisteme değişiklik gösterdiği unutulmamalıdır. Kullanılan donanım özelliklerine ve aynı anda çalışan arka plan uygulamalarına bağlı olarak, işlem süreleri uzayabilmektedir. Bu nedenle, performans ölçümleri yapılırken test ortamlarının dikkate alınması büyük önem taşımaktadır.

Uygulamanın kullanım aşamasında, işlem yapılacak kaynak dosya "Kaynak Dosya" bölümünden seçilmektedir. Ardından, işlenmiş verilerin kaydedileceği hedef dosyanın belirlenmesi gerekmektedir. Bu hedef dosya, işlem tamamlandığında sonuçların kaydedileceği dosyadır.

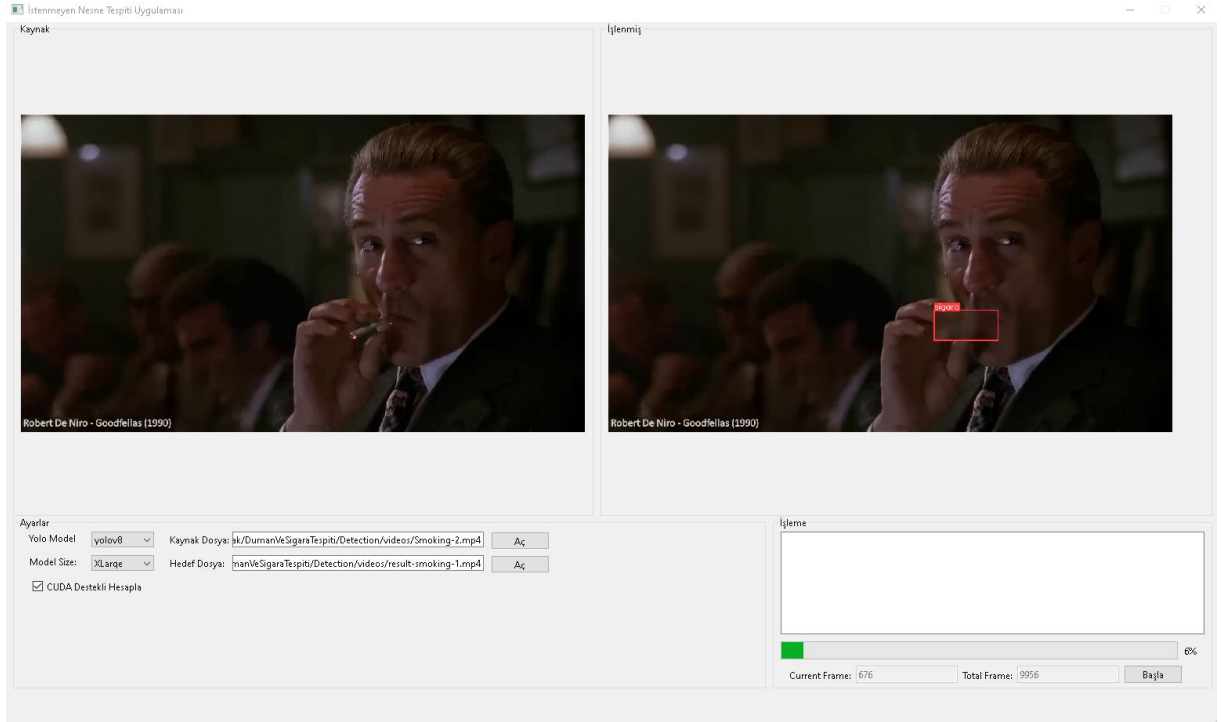
Tüm ayarlar ve seçimler yapıldıktan sonra, "Başla" seçeneği ile işlem başlatılır ve arka planda sansürleme süreci gerçekleşir. İşlem sırasında, video kare kare analiz edilerek sonuçlar kullanıcıya arayüz üzerinde anlık olarak yansıtılır. Bu özellik, kullanıcıya işlemin ilerleyişini takip etme ve sürecin detaylarını gözlemleme imkanı sunar.

4.ÇIKTILAR

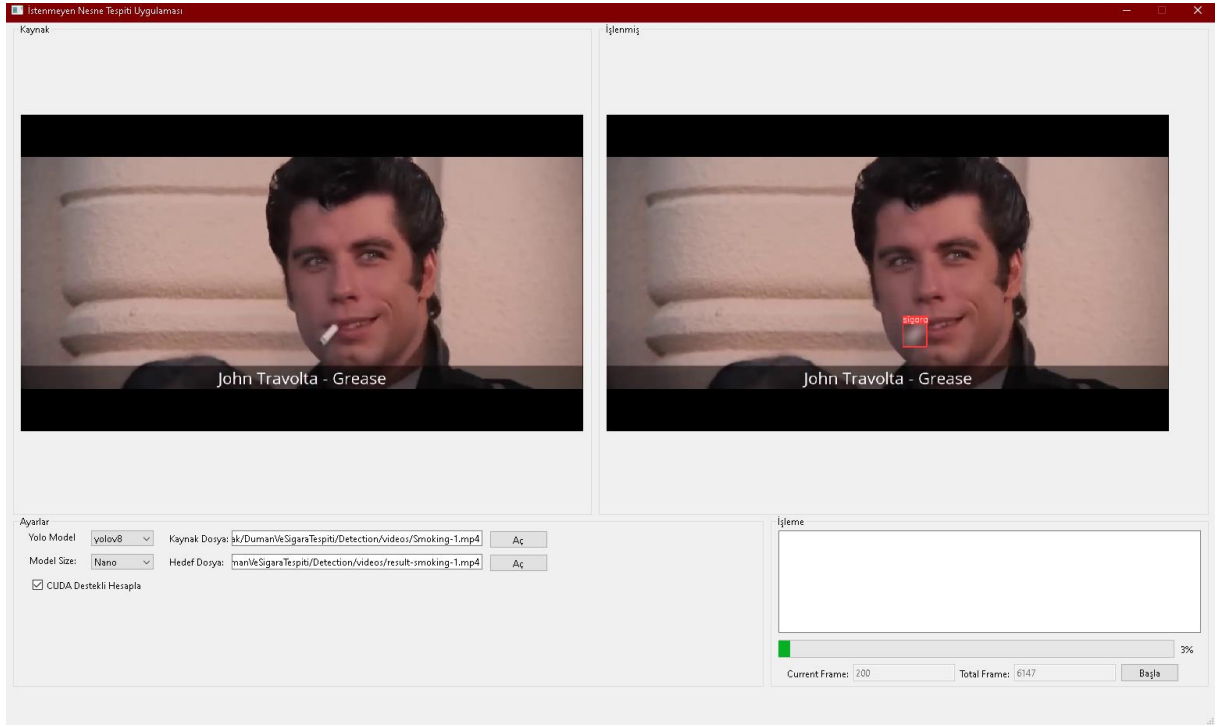
Belirtilen github sayfası üzerinden projenin dosyalarına ulaşabilirsiniz. Tespitin yapıldığı alanların daha rahat gözlenmesi için verilen resimlerin çıktıları alınırken kırmızı bir çerçeve ve tag kullanılmıştır. Proje yüklenirken bu işlem deactive edilmiştir.



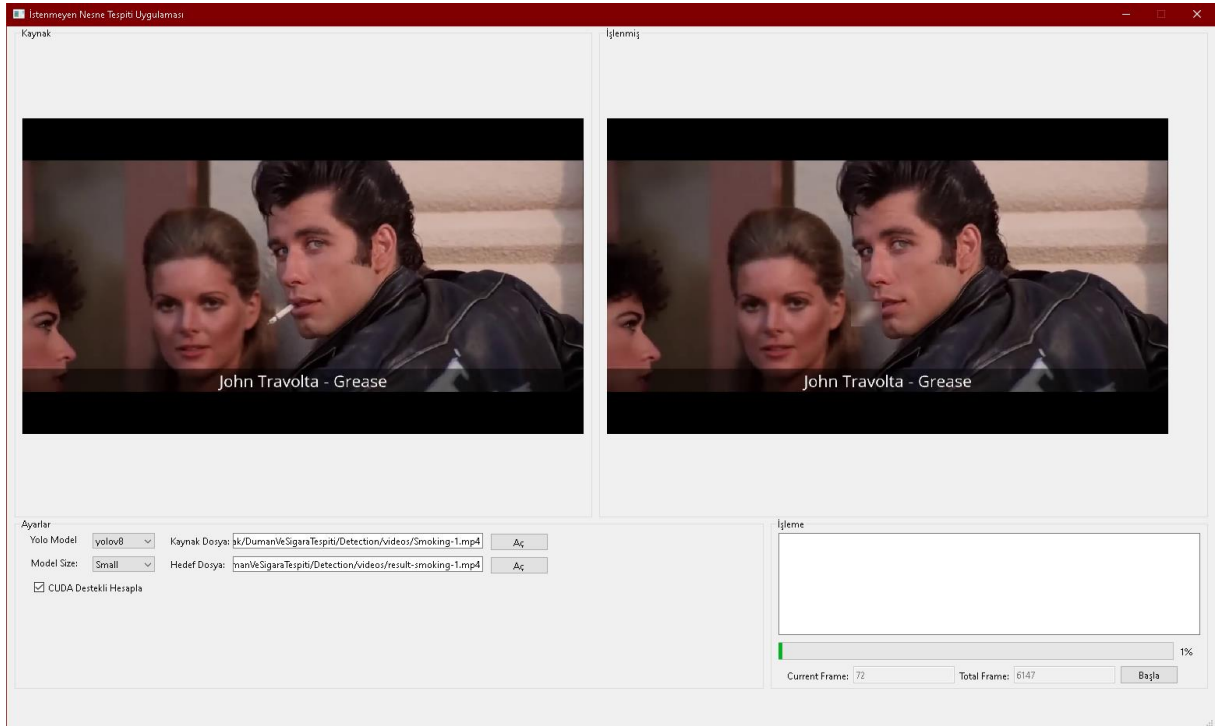
II YoloV8s ile uygulamanın çalıştırılması



III YoloV8 x ile uygulamanın çalıştırılması



IV YoloV8 n ile uygulamanın çalıştırılması



V Uygulamanın en son perde de göstereceği görüntü

5. PROJE İLE İLGİLİ HARCAMA KALEMLERİ

Proje kapsamında maddi destek istenmediği için rapor edilecek herhangi bir harcama kalemi bulunmamaktadır. Proje esnasında yürütücü öğrencinin kişisel dizüstü bilgisayar ve üniversitesinde bulunan bilgisayar sistemleri tarafından model hesaplamaları gerçekleştirilmiştir.