

TAKIM İSMİ:Glieser

Nesne tespiti için YOLO4 kullanılmış

Darknetin ve YOLOnun buildini almak için Vscode 2019 kullandık

Compile etmek için Cmake yazılımını kullandık ve OpenCV,Nvidia,CUDA,görüntü işlemeyi GPU üzerinden yapmamızı sağladı.Yapay zeka öğretimi için Darknet yazılımını kullandık

Veri seti için dataset v6 kullanıldı.

Toparladıkları yazılımları bir arada bulunabilmesi ve farklı programlar içerisinde kullanılan kodların eş zamanlı çalışmasını sağlamak amacıyla Cmake yazılımı kullanmışlar.

Veri setinin görüntüleri algılayabilmesi ve tanımlayabilmesi amacıyla OpenCV kullanmışlar.

Nvidia CUDAnın görüntüyü GPU üzerinden almaları ise daha stabil ve daha hızlı veri almaşarını sağlamış

TAKIM İSMİ:ADYU AI TEAM

Algoritmalar ve özgünlük kısmında puanları çok kırılmış,özgünlük ve yazılım mimarisi ve sistem donanımlarını daha fazla geliştirmeyi düşünmüşler,algoritmalar üzerine daha detaylı açıklama yapmışlar

Labelimg programını kullanarak resimlerini etiketlemişler,bu çok zaman aldığı için GitHub ortamında geliştirilmiş otomatik resim etiletleme yapan algoritmayı kullanmışlar data set oluşturmak için

Nesne tespit için SSD ve YOLO modeller denenmiş ama en isabetlisi Faster R-CNN ,zamanın iyi kullanılması ve nesne tespit oranındaki başarısından dolayı en isabetlisi buymuş YOLO nun insan içeren detaylı resimlerde doğruluk oranı düşükmüş

Yurt dışında çeşitli üniversitelerin araştırma amaçlı yayınladığı iha görüntülerini veri seti olarak kullanmayı düşünmüşler

Google ın açık kaynak kodlu Tenserflow Object Detection API modülünü kullanmışlar

Veri seti oluştururken karşılaştığımız nokta uçan araba iniş ve uçan ambulans iniş resimlerinin azlığı ile ilgili,bunun için de resim çoğaltma ile data augmentation metotlarını kullanarak datasetimiz içindeki uçan ambulans iniş ve uçan araba iniş resimleri ile zenginleştiriyoruz

Bir diğer nokta normalde Faster R-CNN dahil diğer algoritmalarda da bir threshold değeri var Faster R-CNN algoritmasında tresholdu %90 a ayarlayınca doğruluk payı bu değer ve üzerindeki nesnelerin düzgün bir şekilde tesbit edildiği gözlemlenmektedir. Halbuki %80

doğruluk değerlerinde bile insan etiketli nesnelerde bulunabilmektedir. Ama trershold değeri yüksek

ayarlandığından bu puan kaybına neden olmaktadır. Bizlerde bunun üzerine geçebilmek için %70

treshold değeri ile %90 treshold değerleri arasında tesbit edilebilen nesne bölgelerinin kesilip bunları

büyütüp enhance edilerek tekrardan hızlı bir şekilde mesela YOLO gibi bir algoritmayla nesne tanıma

algoritmasına sokularak buradaki nesnelerin tesbiti sağlanabilir. Bunun bizim doğruluk oranına

çok ciddi etki edeceğini düşünüyorlarmış. Bununla ilgili Hibrid olarak son yıllarda çıkan Dynamic head algoritması ile Faster R-CNN algoritmasını birleştirmeyi düşünmüşler Dynamic Head: Unifying Object Detection Heads with Attentions” adlı makale bu konuda çok güzel başarılar ortaya koymuştur.

Bir diğer özgünlük metodu için Sypros GIDARIS in “LocNet: Improving Localization Accuracy

for Object Detection” adli makalesinde paylaşılan LocNET isimli metodu kullanarak IoU doğruluğunu

artırmak için nesnenin çerçeve değerlerini doğruluğunu artırmayı amaçlamışlar

Ek olarak Wei JIANG’ın “Improve Object Detection by Data Enhancement based on GAN” adlı makalesinde bahsettiği resimlerin görüntü kalitesini artırma metodunu da kullanmışlar. Çünkü ufak detaylar görüntü kalitesi artırıldığında daha da belirginleşiyormuş.

Bir diğer yöntem olarak layer segmentasyonu oluşturarak tespit etmek istedikleri nesnenin arka plan pixel hareketlerini sabitleyip hareket esnasında oluşan bilgi kayıplarını azaltıp doğruluk oranını artırmak istemişler . Bununla beraber gerçek zamanla hızlı nesne tespiti yapabilmek için algoritmalarında fast tracking optimizasyonu yapmak istiyorlarmış. Bunları yaparken Jiangjian Xiao`nun “Vehicle and Person Tracking in UAV Videos “ adlı makalesinden faydalanacaklarmış .

Eğik açılı resimlerle de data setlerini zenginleştirmişler.

TAKIM İSMİ =MERGEN

Nesne tespiti için YOLOV4 kullanılmış.

Veri etiketleme işlemi Labelmg ortamında gerçekleştirilmiştir.

Bu projede etiketli görüntüler üzerinden nesne tespitine dayalı danışmanlı derin öğrenme modelleri kullanılacaktır.

Öznitelik yayılımını arttırmak ve özniteliklerin yeniden kullanılmasını ve özellik haritalarının birleştirilmek adına DenseNet kullanılmıştır.

YOLOv4 ün head bölümünde YOLOv3 yer almaktadır, burada yüksek yoğunluklu tahminler yapılır.

Tek bir çerçevede birden fazla nesneyi tanıyabilen gerçek zamanlı bir nesne tanıma modellerinden oluşan YOLOv4 ve YOLOv4-tinny algoritmalarının birleştirilerek hibrit bir yapının oluşturulması amaçlanmıştır.

TAKIM ADI:FEZA AI TAKIMI

Yolo algoritması kullanılmıştır.Makine öğrenmesi kütüphaneleri de kullanılarak hedeflenen “%70 ve üzeri” nesne tespit oranına ulaşmayı amaçlamışlardır.

YOLO V4 ve YOLO V5 algoritmaları diğer YOLO versiyonlarına göre daha gelişmiştir. Takım, danıştıkları akademisyenlerden de gelen geri bildirimleri de göz önüne alarak kullanım kolaylığından dolayı YOLO V5 algoritmasını projede kullanmaya karar vermiştir.

Kullandıkları YOLOv5 modeli, diğer YOLO modellerine nazaran daha büyük boyutta olup YOLOv5x modelidir.Bu modelin daha fazla nesne tespiti sağladığı görülmüştür.

Algoritmamızın eğitilmesi için gerekli miktarlarda UAP ve UAİ görsellerinin bulunmaması ön kabulünden yola çıkılarak yapay UAP ve UAİ görselleri oluşturan bir sistemin geliştirilmesine karar kılınmıştır.

Open cv kütüphanesi kullanılarak yapay görseller oluşturulmuştur. Lakin, rapor sonuçlarında hakem heyetinden gelen geri bildirimler doğrultusunda bu görsellerin gerekli niteliklerde olmadığına karar kılınmıştır

Sistem araç tespitleri noktasında problem yaşamamaktadır ancak yaya tespitleri konusunda hala istenilen orana erişilememiştir. Tespit edilen nesnelerin sınıflandırılması ve hedeflenen tespit oranına erişilebilmesi için takım çalışmalarını sürdürmektedir.

TAKIM İSMİ:YOLOva

Nesne tespitinde YOLOv5 kullanılmış.

Veriler nesne tespit işlemine girmeden önce filtreler uygulanır, bu filtreler araçların arka plandan daha kolay ayırt edilmesini sağlar. Çalışmada, G'MIC filtrelerinden “Freaky Details” filtresi kullanılmıştır. Nesnelerin tespitinin yüksek başarı ile gerçekleştirilebilmesi için ön işlem aşaması olarak filtrelerin uygulanmasıyla görsellerin daha belirgin hale getirilmesi sağlanmıştır.

“Super Resolution” algoritmalarının kullanılmamasına karar verilmiştir çünkü bu algoritmaların gerçek zamanlı çalışmaya uygun olmadığı ve yarışma esnasında vakit ve performans kaybına neden olacağı öngörülmüştür.

Nesnenin bir sonraki fotoğraftaki konumunun belirlenmesi için çalışmalara başlanmış bunun için Kalman filtresi ile çalışmalara başlanmıştır.

VisDrone2019 Veri Setinin yarışmada kullanılması için etiket formatı düzenlendi, açısı uygun olmayan fotoğraflar temizlendi ve model eğitiminde kullanılmaya başlandı.

VEDAI Veri Seti otomatik hedef tespit tanıma algoritmalarının kıyaslanması için oluşturulmuş olan veri setleridir.

Sentetik Veri Seti Veri setinin yeterli olmaması durumunda Unreal Engine 5 oyun motoru kullanılarak üretilecek olan veridir.

YOLOV5 sürümü kendisinden daha önceki sürümlerde kullanılan derin öğrenme modeli oluşturmada kullanılan Darknet Framework yerine PyTorch Framework kullanmaktadır.

SAHI (Slicing Aided Hyper Inference) küçük nesne tespiti için genel dilimleme destekli çıkarım ve ince ayar yapmaktadır. Herhangi bir nesne tespit algoritmasının üstüne eklenip tespit oranları arttırılabilmektedir.

Nesne tespiti yapılmadan önce tespiti yapılacak görsel kalitesinin arttırılmasında kullanılacak Fast SRGAN algoritmasının testleri yapılacak, uygun olup olmadığı bu algoritmanın çalışma süresinin yarışma süresini aşp aşmayacağı test edilecektir.

Araç tespitinde özellikle SAHI algoritmasının kullanılması ile daha iyi tespit oranı sağlandığı görülmüştür. Fazla ışık almayan görsellerde insanların tespitinin oranının az olduğu görülmüştür. Bunun düzeltilmesi için farklı filtreler kullanılmasının uygun olduğu düşünülmüştür. Taşıtların tespiti oranının diğer sınıflara göre daha az olduğu görülmüştür. Bu sorunun çözümü için taşıt verilerinin olduğu görsellerin sentetik veri olarak arttırılması düşünülmüştür.

TAKIM İSMİ: BigBrainet

Veri seti olarak COCO ve VisDrone kullanılmıştır.

Videoların içerisinde hareket bulanıklığı olan frame'ler süzgeç ve filtreler yardımı ile netleştirilmiş. Etiketsiz verilerin etiketlenmesi için kullanılması planlanan LabelImg açık kaynak kodlu yazılım

işlevsellik, kullanım kolaylığı ve performans açısından test edilmiş. El ile etiketleme sırasında program arayüzünün basitliği ve işlevselliği nedeniyle LabelImg yazılımı el ile işaretleme işlemlerinde kullanılmaya devam edilecekti.

YOLOv5 modelinin versiyonları kullanılarak kapsamlı deneysel çalışmalar gerçekleştirilmiştir. Bu versiyonlar arasında en yüksek performans YOLOv5l6 modeli ile elde edilmiştir. Bu deneysel çalışmalarda, özgün olarak elde edilen frame'lerin yanı sıra COCO ve VisDrone veri setlerinde yarışma için paylaşılan veri kümesindekilere benzeyen görüntülerin çıkartılarak kullanılmıştır. Bu sayede, veri çeşitliliği artırılarak geliştirilen modelin öğrenme performansı artırılması amaçlanmıştır.

YOLO algoritmasına dayalı modelleri detaylı olarak araştırdıktan sonra, gerçek zamanlı çalışma hızı yüksek ve küçük obje tespitindeki yüksek performansından dolayı PyTorch temelli YOLOv5 mimarisi kullanılmış.