

ADYU AI

Labelimg resimleri etiketlemeye yarar

Github ortamında geliştirilmiş otomatik resim etiketleme yapan bir algoritma kullanılmış.

Faster rcnn ve yolov4 dyhead

Stanford + dronela kendileri çekmiş

Yolo ve ssd hassas değil -> faster rcnn seçilmiş

Bölge önerilerini cnn ile beslemek yerine erişimli bir özellik haritası oluşturup cnn girdi görüntüsü ile besliyorlar. Konvolisyon özellik haritasından tekliflerin bölgesini tespit ediyor ve bunlara kareleri çarpıtıyorlar. ROI havuzlama katmanı kullanarak tam bağlı bir katman besleyebilmeleri için sabit bir boyutta şekillendiriyorlar.

Youtube UAV dataset

Otomatik etiketleme AutoLabel algoritması

Koordinatlar daha doğru bir şekilde bulunduğunda ROI oranının daha efektif olmasını sağlar

Data augmentation araba ve ambulans iniş resimleri

IoU doğruluğunu arttırmak için LocNet isimli metodu kullanmaya çalışmışlar

Ai Lab—Horizon

Yolor yapısı gereği aynı nesnelerin farklı ortam ve şartlarda görerek eğitilmesi algoritmanın kararlılığını oldukça arttırdı.

Pexels—> veri arttırma

TPH-Yolov5 Algoritması drone görüntüleri için geliştirilmiş özgün bir sürüm

YOLOR, içerisindeki derin öğrenme katmanları sayesinde veri seti eğitimi sırasında aldığı veriler ile önce açık bilgiyi öğrenir, sonrasında ise daha alt katmanlarda örtük bilgi denilen nesne detaylarını öğrenir. Bu sayede sadece eğitimde verilen görüntülerle değil de bir nevi bilinçaltı oluşturarak kendi tecrübeleriyle de nesneleri tespit edebilmektedir

Weighted Boxes Fusion:

Nesne tespit algoritmalarında tespitlerin karışmaması veya üst üste binmemesi için filtreleme algoritması. Birleştirilen tüm modellerin yanlış tahminler yaptığı durumlarda bile doğruya yakın sonuçlar üretir.

Detaylı sınıflandırma (her taşıt cinsi için) yapılmış.

Videonun görüntü kalitesini interpolasyon yaparak arttırmışlar.

Daha fazla UAP ve UAİ alanları üretmek için RAGE oyun motoru kullanarak gerçekçi sahneler simüle etmişler daha fazla UAP ve UAİ verisi üretmişler.

Avion ai

VisDrone Veri setinin çekim açısının yarışmanın amacını hizmet etmediği, Stanford veri setinin ise kendi veri setlerindeki dengesizlik oranını arttırdığı görüldü. İnsan ve taşıt gibi fazla bulunan objelerin yoğunluğunu azaltmak için eleme işlemi yapıp dengelenmiş veriyi veri setine dahil ederek veri setinin genişliğini artırmayı hedeflemişler.

Yoloyu iniş alanlarının doluluk tespiti aşamasında kullanmışlar

CNN nesne UAİ UAP alanları, tespit modeli. Bounding boxların kırılıp dolu ve boş sınıflandırması yapar.

PR, ROC eğrileri, F1skoru, mAP değerleri araştırma ve deney kısmı

Küçük nesnelerin tespiti: sliding uygulaması, yolunun yolt versiyonu, yoltv5

Videodadan Elde edilen görüntüler MakeSenseAi kullanılarak işaretlendi insan taşıt gibi sınıflara ayrıldı.

Kendi veri setlerini oluşturmuşlar. Stanford Drone veri seti kullanmışlar

YOLT (You Only Look Twice) YOLO üzerinden türetilmiş, uzay ve hava görüntüleri

kullanılarak eğitilmiş bir modeldir. YOLT modeli küçük nesneleri tanımda daha başarılı.

Sorunlar:

Modelin nesnelerin genellemesinde veya eğitildiğinden farklı açılarda karşısına çıkması durumunda zorlanması

Çözüm: Rescaling ve Rotation resim çoğaltma teknikleri

- Kuş sürüşü gibi küçük objelerin tespitinde zorlanması

- YOLO'nun birden fazla downsampling layer kullandığı için obje tespitinde görece biraz daha vurdumduymaz çalışıyor olması

Çözüm:

- o Yeni bir network(YOLT) üzerinde daha yoğun bir final katmanı kullanmak
- o Daha önceden YOLO ağı üzerinde sıkıştırılarak (down-sampled) incelenen objelerde küçük obje tespitini efektif kılmak için bu yoğun katmanları tekrar netleştirmek (upsample)
- o Farklı ölçeklerde tespit edicilerin çalıştırılarak hem küçük hem de büyük obje tespitini efektif kılmak

Çözünürlükte düşüş için yöntemler: Clahe, Copy-Pasting , Rotating, Rescaling

GAN kullanarak hava şartlarından etkilenmeyecek düzeye ulaşmasını sağlarlar

Yolonun backboneu üzerinde değişiklik— CSPNet (hız)

Küçük nesnelerin tespiti için copy-paste algoritmasının aktifken daha iyi sonuç verdiği görülmüş.

sorun Background FP'lerin insan sınıfında yüksek olması.

Bunun temel sebebi insan sınıfı diğer objelere kıyasla çok daha küçük objelerden oluştuğu için bazen insan olmayan objelere de insan sınıflandırması yapıyor.

“YOLT, YOLO'ya göre başarısız oldu ancak daha önce test ettiğimiz metriklere göre başarılı olmasını bekliyorduk. Burada en büyük hatamız veri seti değişkenini deney yaparken sabit tutmamak oldu. Model mevcut veri seti üzerinde çok iyi başarı gösterse de küçük insan gruplarının etiketli fotoğrafı yoktu.”

CodeArt

Teknofestin verdiği veri setinde UAP ve UAİ alanlarının yetersiz olmasından dolayı verilere manuel görsel manipülasyona uygulanmış ve yeterli sayıda UAP ve UAİ alanı elde edilmiş. 3 algoritma kullanılmış bunlar YOLOv4, Faster RCNN, SSD.

Öncelikle videonun çerçevelere (framelere) bölünmesi ile saniyede beş kare olacak şekilde elde edilen fotoğraflara bölünmüş. Bu fotoğraflar denetimli öğrenme (supervised learning) algoritmalarına etiketlenmiş. Doğruluk oranını artırabilmek için görüntü işleme fonksiyonları ile veri seti üzerinde düzenlemeler yapılacak. VisDrone veri seti kullanılarak veri genişletilmiş.

R-CNN modeli resmi parçalara ayırma işlemini kayan pencere yöntemi olan Selective Search algoritmasıyla yapmaktadır. Selective Search algoritması nesne bulunma olasılığının yüksek olduğu bölgeleri hızlı bir şekilde tespit etmektedir, öğrenme içermeyen sabit bir algoritmadır.

Tespit edilen bölgenin özellikleri çıkarıldıktan sonra SVM ile sınıflandırılarak nesne tespit işlemi gerçekleştirilir.

Araç gözetimi ve trafik izleme Senaryosunu YOLOv3 algoritması Daha sağlam ve daha yüksek hatırlamaya sahip.

Yolo algılama ve sınıflandırmayı içerir. Tek bir sinir ağı ile nesne tespiti yapar.

Yolo nesnelerin konumlarını gösteren sınırlayıcı kutuları (bounding box) bulmak için SxS boyutunda ızgara (grid) şeklinde bölmüştür daha sonra her bir ızgara hücresi için bir nesne tanımı yapmıştır birbirine komşu ve aynı sınıfı gösteren ızgara hücresi birleştirilerek sınırlama kutusu elde edilmiştir algoritma görüntüyü tek bir seferde sinir ağından geçirerek resimdeki tüm nesnelerin sınıfını ve koordinatlarını tahmin eder Her bir ızgara kendi içinde alanda nesnenin olup olmadığını varsa orta noktasının içinde olup olmadığını orta noktası da içinde ise uzunluğunu yüksekliğini ve hangi sınıftan olduğunu bulmakla sorumludur her ızgara için ayrı bir tahmin türü oluşturulur vektörlerin içine koordinatlar nesnenin boyutu sınıfı iliştilir.

SSD (Single Shot Multibox Detector), tek bir derin sinir ağı kullanarak tek seferde nesne tanıma işlemi yapmaktadır SSD nesne tespiti özellik haritasının çıkarılması ve nesneleri algılamak için evrişim filtrelerinin uygulanması olmak üzere iki aşamada çalışmaktadır. hız önemliyse SSD kullanılmalı, daha doğru tespit önemliyse Faster R-CNN kullanılmalıdır.

Veri çoğaltma teknikleri kernel filtreleri, rastgele silme, görsel birleştirme, renk uzayı dönüşümleri, düşmanca eğitim, çekişmeli üretici ağı, geometrik döndürme, kırpma, oklüzon, yoğunluk işlemleri, gürültü enjeksiyonu, kombinasyon ve yakınlaştırmadır.

Veri setini çoğaltmak OverFitting'e yol açabilir modelin ezberlenmesi engellenmesi için EarlyStop uygulanmış.

Modelin değerlendirme kriterleri olarak karmaşıklık matrisi, duyarlılık, hassasiyet ve IoU kullanılacaktır.

Karışıklık Matrisi, gerçek değerlerinin bulunduğu bir dizi test verisi ile modelin tahmin değerlerinin karşılaştırılmasını sağlayan bir tablodur.

BigBrainNet

Yolov5- evrişim sinir ağı kullanır- nesnelerin konum bilgilerini verir üst üste anlamamızı sağlar çalışma prensibi görüntüyü bölgelere ayırır her bölgedeki nesneleri kapsayıcı kutu çizer daha sonra her bir bölgede nesne bulunma olasılığını hesaplar hesaplanan skorlarına bağlı olarak nesneleri tahmin eder. Non maximum suppression: en yüksek güven skoruna sahip olan bounding box kalır.

COCO ve VisDrone veri seti

İha ile veri toplanmış

Görüntüleri maksimum verim elde edilebilecek hale getirme

Labelimg ile etiketleme

İniş kalkış alanlarının tespiti için gerekli logolar drone yardımı ile elde edilen görüntülere yerleştirilerek oluşturulan veri kümesi de eğitim setine katılmıştır

KTR aşaması: görüntü boyutları yükseltilerek nesne tespitine etkileri gözlenmekte.

Bingöl STEM Roboklup-UYZ

CenterNetHourGlass104 ağı özellikle insan tespitinde çok iyi

YOLO ve SSD gibi çapa tabanlı dedektörlerde karşılaşılan aynı nesneyi birden fazla tespit etme ve bunu aşmak için kullanılan Non Max Supression (NMS) algoritmasına ihtiyaç duymaz ve ek bir algoritma ihtiyacı ortadan kalkar.

Programlama dili olarak Python'ı Anaconda-Spyder ve Anaconda-Jupyter ortamlarında kullanmışlar.

Yapay zekâ kütüphanesi olarak; train işlemleri için Tensorflow-1.14-gpu ve genel kullanım ve test işlemleri için daha hızlı olduğundan dolayı Tensorflow 2.2.0'ı kullanmışlar.

“Resimde OpenCV” kütüphanesi kullanılarak bütün bir dairenin olup olmadığına bakılacaktır. Dairenin bulunması durumunda inişe uygun aksi taktirde inişe uygun değil değerleri JSON dosyasına işlenecektir”

Detaylı taşıt sınıflandırma yapılmış.

Feza ai

Yolov5 (Konvolüsyonel)

Görüntüler ekstrem hava koşullarını içermelidir farklı zamanlarda çekilmiş olmalıdır.

Nesne tespit algoritmaları Coco veri setinde karşılaştırılır.

Grid SxS

Non max supression: nesneyi kapsayan birden fazla sınırlayıcı kutucuk çizilir güven skoru düşük olanlar temizlenir bu şekilde en yüksek skor elde edilir.

PyTorch Hub içerdiği nesne sınıfı parametreleri. Modelin sonuçlarının JSON olarak aktarılmasına yarar

Yapay görsellerin daha gerçekçi olması ve üç boyutta sahip olması istendiği için oyun motoru olan rockstar advanced game engine kullanılmıştır.

Adobe photoshop ile uap ve uai nesneleri için ör görseller oluşturulmuştur .

Adobe phptpshop ile kırışıklık efekti.

Glieser

Yolov4 tek bir gpu ile görüntü alınabilir. Visual studio. Compile için Cmake yazılımı. Opencv, nvdia cuda(stabil ve daha hızlı veri alımı) görüntü işlemeyi gpu üzerinden yaptırma. Yapay zeka için Darknet yazılımı kullanılmış. İnternet bağlantısı olmadan çalışabilen uygulamalar.

Veri seti ihtiyacı için Google destekle Open images dataset v6 kullanıldı buradaki sınıfların kısa bir açıklaması open image datasetin sitesinde mevcutmuş.

Yolo çalışma prensibi üç tekniğe dayanır:

Residual blocks- ızgaralara bölünür

Sınırlayıcı kutu regresyonu- nesnelerin etrafını saran kutu

Intersection over union- kutuların üst üste binışı

Cmake derleyici

OpenCv (gerçek zamanlı bilgisayar görüşü uygulamalarında kullanılır kütüphane) veri setinin görüntüleri algılayabilmesi ve tanımlayabilmesi.

LunAi

Stanford Drone Dataset kullanılmış ve insan nesnesi eksik olduğu için VISDRONE kullanılmış. Yolox (daha yüksek doğruluk) simOTA(etiket ataması) yı kullanır.

Etiketleme LabelImg yardımıyla eğitim sırasında mAP.5 metriği kullanılarak yapılan değerlendirmeler sonucu en yüksek performans gösteren checkpointler tutulmuştur. Bunun nedeni eğitim sırasında oluşan bottleneck'leri mümkün olduğunca çabuk düzelterek eğitime devam etmektir. WANDB üzerinden performans takibi yapılmıştır.

Test Data Augmentation farklı augmentation teknikleriyle çoğaltılarak prediction almamızı ve var olan frame üzerinde daha genel, doğru sonuç çıkarmamıza yarayan bir tekniktir. False positiveleri artırır ancak yarışmada kullanılacak olan mAP metriği üzerinde olumlu etkisi vardır.

Mergen

ESA Evrimsel sinir ağı -CNN

Yolov4

Supervised learning etiketli görüntüler üzerinden Derin öğrenme algoritmaları insan beyninin işleyiş mekanizmasından ilham alan yapay sinir ağı katmanlarından oluşmaktadır

Python, opencv kütüphanesi, darknet sinir ağı çerçevesi

Csp darknet53 kategorizasyon (girdinin karmaşıklığı azaltılır)

Backbone: görsellerin önemli özelliklerini vurgulamak için kullanılır(cspdarknet)

Neck: özellik piramitlerini oluşturmak için kullanılır aynı nesneyi farklı boyut ve ölçeklerde algılamamızı sağlar(spp ağ yapısı)

Panet: görüntüde sunulan değişen sayıda özelliği yerleştirmek için sınıf etiketlerini tahmin eder

Head: neckden veri alır ve tanıma işlemi

Özgün noktaları gürültünün tespit edilmesi gürültü eklenmesi ve temizlenmesi.
Görsel piksellerinin değişmesine karşılık ön işleme ve bölütleme yöntemi yapılır bölütleme yönteminde görselin bileşenleri ve nesnelerini ayrılarak modelin görüntüyü daha net şekilde oluşturması sağlanır.

Görsellerin kırılması ile veri girişindeki çözünürlük korunur küçük nesneler dahi hızlı tanınır

Zorlu hava koşullarında araç tespitinin zorlaştığı sırada görüntüye gürültü eklenir ve gürültünün tespit edilmesi sağlanır.

YOLOva

Yolov5 Darknet framework yerine pytorch framework kullanır

Visdrone 2019 veri seti (görseller)

VEDAI hedef tespit tanıma etiket formatı uyarlanması

!!!GMIC Freaky Details filtresi ile araçların kolay ayırt edilmesi sağlanır.

SAHI – küçük nesne tespiti için kullanılır.

ZeroAlpha

Eğitimler Google Colab üzerinde Nvidia P100 ekran kartları kullanılarak yapılmakta.

CycloMedia ve Google Earth çevrimiçi sitelerle birlikte GF-2 ve JL-1 isimli uydulardan alınmış görüntüler içermektedir veri setleri

Yolov516

Veri arttırma teknikleri:

HueSaturationValue: Giriş görüntüsünün, rastgele olarak renk tonunu (Hue), renk doygunluğunu (Saturation) ve ışık değerini (Value) değiştirmektedir.

GaussianBlur: Giriş görüntüsüne rastgele bir Gauss filtresi uygulamaktadır.

GaussNoise: Giriş görüntüsüne rastgele olarak Gauss gürültüsü uygulamaktadır.

RandomBrightnessContrast: Giriş görüntüsünün, rastgele olarak parlaklığını (Brightness) ve kontrastını değiştirmektedir.

CLAHE: Giriş görüntüsüne sınırlı uyarlanabilir histogram eşitleme (Limited Adaptive Histogram Equalization) uygulamaktadır.

Yukarıdaki veri arttırma teknikleri dışında YOLOv5 içinde hazır bir şekilde bulunan veri arttırma teknikleri de vardır. Bunlar arasında eğitim sırasında kullanılanlar şöyledir:

hsv_h: Görüntünün renk tonu üzerinde değişiklik yapar.

hsv_s: Görüntünün renk doygunluğu üzerinde değişiklik yapar.

hsv_v: Görüntünün renk ışık değeri üzerinde değişiklik yapar.

degrees: Görüntünün açısını değiştirir.

translate: Görüntünün piksellerini kaydırır.

scale: Görüntüyü yeniden boyutlandırır.

flipud: Görüntüyü yukarı-aşağı yönde çevirir.

fliplr: Görüntüyü sağ-sol yönde çevirir.

mosaic: Görüntü parçalara ayırıp bu parçaları tekrar rastgele bir şekilde birleştirir.

GAiA Tech

Örnek videonun her karesini. JPG uzantılı resim dosyasına dönüştürmüşler YOLOv4 formatına uygun etiketlemişler (you only look Once) tek tip etiket formatıyla çalışmışlar gerektiğinde etiket dönüşümü yapmışlar. Veri seti veri inceleme yöntemleri ile hangi sınıftan kaç adet etiketleme yapıldığı incelenmiş veri çoğaltma yapılmış, YOLO'nun 4 farklı türevi, SSD ve Faster R-CNN algoritmaları denenmiş.

Kullanılan verilerde 60 ila 90 derece arası çekilmiş görüntüler ile veri seti oluşturulmuş. Yarışma videosundaki araç insan UAP ve UAİ sınıfları etiketlemiş. Toplamda 51.349 adet veri bulunmakta. Farklı hava koşulları göz önünde bulundurularak veri setleri eklenmiş nesnelerin tam görünmedi durumlarda etiketleme yapmışlar. Yayaların hem ayakta durduğu hem oturduğu veriler etiketlemiş.

Nesne tespiti: 1. Öneri odaklı tek aşamalı, 2. Regresyon odaklı iki aşamalı iki aşamalı olan daha yüksek hassasiyete sahip ancak daha yavaş bu sebeple YOLOv4 algoritmasının doğruluğu daha iyi olduğu için bunu kullanmışlar.

Faster R—CNN: iki aşamalı nesne tespiti algoritması. İlk aşamada Konvolüsyonel katman bölge teklif önerisi ile nereye bakılacağını bildirir ikinci aşamada R—CNN Algılayıcısı ilk aşamadan gelen tahmin konumlarını kullanarak nesne tespiti için bölge Önerisinde bulunur. Bu mimari YOLOv4e göre çok yavaş kaldığından kullanımı uygun bulunmamış.

SSD: YOLO ile benzer çalışır. İki aşamadan oluşur özellik haritalarına çıkarılması ve nesneleri algılamak için Konvolüsyonel filtrelerin uygulanması. Şu şekilde çalışır giriş görüntüsünü çeşitli boyutlarda ızgaralara bölerek farklı sınıflar ve farklı en boy oranları için tahminde bulunur bu tahminlerden yüksek puanlı ve eşleşen tahminlere karar verir. Çok tahminde bulunduğundan dolayı hız açısından tercih edilmemiştir.

YOLOv4: Tek aşamalı ve iki aşamalı nesne tespiti modelleri vardır iki aşamalı modeller nesne lokalizasyonu ve sınıflandırması birbirinden ayrı bir şekilde yapar. Tek aşamalı modeller aynı anda tahminlerde bulunur görseli tek aşamada değerlendirir sadece bir kez bakar.

Görüntüye girdi olarak alır birbirine komşu ve aynı sınıfları gösteren bölümleri birleştirerek sınırlayıcı kutu oluşturur sınırlayıcı Kutular tahminlerin hesaplanması sonucu oluşur. Görüntüyü ızgaralara bölerek her ızgara için puanlama yapar her ızgara için ayrı bir tahmin faktörü oluşturur Sınıflandırma ile farklı ölçekleri ve boyutlara sahip çoklu sınırlayıcı Kutular tahmin edilmektedir. Gerçek zamanlı nesne tespiti yapar.

Görüntü kalitesi yüksek olan resimleri modelin yeniden boyutlandırıp detayları kaybetmesini engellemek amacıyla SAHI kütüphanesi kullanılmış. Modelin istediği ölçülerde resmi parçalara ayırmak için gerekli hesaplamalar yapılır yeni oluşan etiketlemelerin koordinatları hesaplanır bu sayede görüntü üzerindeki veri kaybı önlenir.

Image Ops tekniği sayesinde görüntüyü etiketlemek için birçok yöntem kullanılabilir. Kontrast eşitlemesi Autocontrast ile yapılırken; Equalize ile de görüntünün histogramı eşitlenmektedir. Görüntünün tersini elde etmek için Invert, görüntüdeki renk kanallarının bit değerlerini düşürmek için Posterize ya da belirlenen eşik değerine göre görüntünün piksellerinin tersini almak için Solarize yöntemi kullanılabilir. Derin öğrenme bağlamında kullanılan veri büyütme, daha fazla etiketli veri oluşturma basit bir yolu sağlar; verilerinde rastgele kontrast/parlaklık değişimi, gürültü ve bulanıklık, döndürme, perspektif projeksiyonu ve kumaş desenleri uygulanmış.

Görüntü üzerinde mozaik işlemi yapılarak veri kaybı en az seviyeye indirilmiş veriler training, validation ve test olmak üzere 3 ayrı bölüme ayrılmıştır. Model eğitirken 6000 epoch yapılmıştır. Modelin ezberleme ihtimaline karşı **early stop** kullanılmıştır. Hata oranı azaltılması için futbol ve basketbol sahaları trafik duası futbol kalesi gibi veriler etiketsiz olarak eklenmiştir.

Değerlendirme kriterleri olarak karışıklık matrisi (confusion matrix), duyarlılık (recall), hassasiyet (precision), f1 puanı (f1-score) ve kesişim bölgeleri (IoU) kullanılacaktır.

Karışıklık maddesi bir sınıflandırma algoritmasının performansını tanımlar.

Doğruluk, sınıfların ortalama hassasiyet (AP) değerlerinin aritmetik ortalaması (mAP) alınarak bulunmaktadır.

Duyarlılık, isabet oranı olarak bilinmektedir ve yüksek olması beklenmektedir. Sınıflayıcının ne kadar TP ya da TN değeri doğru tahmin ettiğinin bir ölçüsüdür.

Hassasiyet, pozitif olarak sınıflandırılanlar arasında doğru sınıflandırılmış örneklerin oranını değerlendirir. Duyarlılık gibi hassasiyetin de yüksek olması beklenmektedir

F-1 puanı, sınıflandırıcının ne kadar iyi performans gösterdiğinin bir ölçüsüdür. Duyarlılık ve hassasiyetin harmonik ortalamasıdır.

IoU, her bir sınırlayıcı kutu için, tahmin edilen sınırlayıcı kutu ile gerçek sınırlayıcı kutu arasındaki örtüşmeyi ölçmek için kullanılır.