Proje Adı: Nesne Tanıma Algoritmaları

Yıl: 2020

Danışmanın Adı: Tuba Bozdemir Şenol

Okul: Işıkkent eğitim Kampüsü

Sinan Yücebaş

Kriter A -Araştırma

Benim ilham kaynağım daha çok etrafımdaki gözlemlerimden ve düşünmenin ve görmenin bende yarattığı heyecandan gelmektedir. Bu heyecan ve merak öyledir ki benim aklımda birçok soru ve fikir ortaya çıkardı. Beş yıldır programlama üzerinde eğitim alıyorum ve bu eğitimler beni, düşünce yapısı olarak çok etkiledi. O zamandan bu yana elektronik aygıtlara olan bakış açım çok değişti. Eskiden programlama konusunda aklıma soyut fikirler ve düşünceler geliyordu ama beş yıl içerisinde fikirlerim daha elle tutulur ve mantık çerçevesine uyar bir biçime geldi. Bu projeyi yapma düşüncesi birkaç yıldır kafamda yatıyordu ama ne yazık ki bu projeyi yapabilecek bir bilgi kapasitesine sahip değildim. Bu yüzden bu projeyi yapabilmek için düşünce yapımın ve programlama bilgimin gelişmesini bekledim ve en sonunda bu yıl bu projeyi hayata geçirebileceğime karar kıldım. Her yıl daha zor, daha iddialı ve benim sınırlarımı daha zorlayacak projeler seçmeye dikkat ettim. Bu sayede hem kendimdeki gelişmeyi gözle görebilecektim, hem de kendimi rahatlık çerçevesinden çıkartıp kendime ebedi bir gelişme katabilecektim. Bu yıl ise benim amacım insanların hayatını hem kolaylaştıran hem de güvenliklerini sağlayan bir algoritmayı insanlara sunup onları bu konuda bilgilendirip insanların bu programları daha doğru ve bilinçli kullanmalarını sağlamak. Ben projem için “Bilimsel ve teknik inovasyon” küresel bağlamının doğru olduğunu seçtim çünkü benim yaptığım proje şu anda çok hızlı gelişmekte ve birçok yeni teknikler bulunmakta. Benim bu projeden umduğum ise şu anki yazılan nesne ve yüz tanıma gibi algoritmaların nasıl çalıştığını ve prensiplerini kavramak ve bu algoritmaların yazımını kendi seviye ve kapasiteme göre öğrenip uygulamaktır. Bu algoritmalar ile bilgileri öğrenmek için udemy, İnternet siteleri ve Stanford Üniversitesi’nin “computer science 231n” adlı dersinden yararlandım. Ama bu kaynaklardan en çok kullandığım udemy oldu çünkü ben bu algoritmalar ile ilgili fonksiyon, terim, data setleri ve bazı tarz algoritmaları burada detaylı bir biçimde öğrendim.

Kriter B -Planlama

Ben bu projeye başlamadan önce zaten programlama üzerine birçok bilgiye sahiptim bu yüzden projeme ilk programlama dilini öğrenmek yerine direk projeme giriştim. Bu projede ben terim ve kavrama konusunda zorlanacağımı biliyordum o yüzden kavramları öğrenmeye ve mantığını kavramaya yazdan başladım. Bu konuda yaparken Türkçe kaynak bulmak zordu çünkü bu konu yeni adı duyulmaya başlayan ve hep bir yenilenme içerisinde olduğundan Türkçe yazılar bulamadım. O yüzden İngilizce sayfa ve dökümanlardan yararlandım. Konum zaten karışıkken İngilizce de üzerine binince anlamak benim için hayli zor oldu. Erken başlamam burada çok işe yaradı. Proje süreci başlayınca ilk olarak daha önceden de karar kıldığım konuyu kesinleştirdim ve sonrasına bu konuyu kapsayan bir küresel bağlam seçtim. Konumu kesinleştirdikten sonra projeme daha çok ağırlık verdim. Konumla ilgili daha noktasal dersler ve bilgiler almaya başladım. Projemin iskeletini yaparken ne kadar çok ilerlersem o kadar çok hata almaya başladım. Bu sorun için bilgisayar öğretmenime ve tanıdığım bilgisayar, makine ve elektrik - elektronik profesör ve mühendislerine danıştım. İhtiyacım olan tüm program ve fonksiyonları belirleyip o konularla ilgili dersler dinleyip yazılar okudum. Program ve kütüphaneleri bilgisayarıma kurdum. Doğru çalışıp çalışmadıklarını kontrol ettim. Hata aldığım zaman yardım aldım veya doğaçlama ve araştırma yoluyla sorunuma çözüm buldum. Araştırıp bulduğum fonksiyon, dataset, terim ve algoritmaları bilgisayarımda dosyaladım. Bitiş tarihine yaklaşırken algoritmamda son değişikleri yaptım ve algoritmamı hatalardan arındırdım. Doğru çalıştığından emin olmak için içine birkaç örnek girdi koydum ve çıktılarla karşılaştırarak doğruluk oranına baktım. Süreç çizelgesi ektedir (11.sf).

Kriter C -Harekete Geçme

Benim projem algoritmalar ve bilgisayar olduğu için sayılamayacak kadar çok problem çıktı. Bu problemleri çoğunlukla internetten araştırarak veya tanıdığım ve bu işte bilgisi olan insanlara sorarak çözmeyi başardım. Ama bazı problemlerde ne yazık ki bana yardım edebilecek kimse yoktu o yüzden bu problemleri kendi denemelerimle çözdüm. Ama kendim çözmeye çalışınca bazen başka problemlere yol açtığım da oldu. Bu nedenle bazı kısımlarda uzun zaman takıldım ve bu nedenle projem biraz yavaş ilerledi ama bazen doğaçlama gitmek ve soruna sorunla yanıt vermek bir çözüm de olabiliyor. Algoritmamı yazarken birçok dataset, terim ve fonksiyonlardan yararlandım. Mesela kullandığım R-CNN adında bir algoritma türünden yararlandım. Bu algoritmalar nesne tanıma algoritmaları için neredeyse vaz geçilmez diyebiliriz. Bu algoritmalar nesne tanıma algoritmalarında bütün işi yapan algoritmalardır. Bu algoritmaların çalışma prensibi basittir. Genel olarak R-CNN programları bir fotoğraf karesini alır ve her pikselini tarar ve şüpheli bulduğu yani bir nesne veya insanın olabileceği yerleri belirli filtrelerden geçir ve bize fotoğraftaki nesneleri çıktı olarak gösterir. Ben kendi nesne tanıma algoritmamda

Faster R-CNN adlı bir R-CNN çeşidini kullandım. Bu R-CNN çeşidi ise daha gelişmiş bir filtrelemeye sahip ve daha hızlı bir şekilde bize sonuç sunuyor. Bu R-CNN çeşidinin bu kadar hızlı ve isabetli olmasının sebebi ise prosedürlerin sırasıdır. Normal R-CNN algoritmasında ilk fotoğraf taranıyor ama Faster R-CNN'de fotoğraf ilk filtreden geçiriliyor. Sonra filtreden geçirilen fotoğraf taranıyor ve şüpheli bölgeler çıktı olarak sunuluyor. Bu prosedür sırasının yaptığı fark çok büyüktür. Mesela normal R-CNN algoritması bize 49 saniyede çıktı verirken, Faster R-CNN 0.2 saniye gibi hızlı bir sürede bize çıktı veriyor. Bu nedenlerden dolayı ben algoritmamda Faster R-CNN kullandım ki bize anlık ve doğru sonuçlar versin.

Yaptığım bu proje tabii ki daha iyi ve daha doğruluk oranı yüksek olacak şekilde tasarlanıp programlanabilirdi ama benim bu projede yapma amacım öğrenmek ve insan hata yapmadan

öğrenemez ve gelişme yaptıklarımdan sonuç çıkartıp iyileştirerek olur. Ayrıca benim kapasitem ve olanaklarım el verdiği kadar iyi yapmaya çalıştım ama bir öğrenci olduğum için hata ve zayıf noktalar var. Hatalarımı düzeltmek ve kendimi geliştirmek için daha çok kuzenimden yardım aldım. Kuzenim makine mühendisi ama programlama ve algoritmalar konusunda birçok bilgisi var. O yüzden bir sorun veya sorum olduğunda ona danıştım. Bu projede materyal olarak sadece bilgisayar ve çıkardığım notlarımı kullandım. Çünkü program yazmak materyallerle değil fikir ve bilgiyle olan bir şeydir. Aslında program yazmayı kitap yazmaya benzetebiliriz. Kitap ta düşünce, hayal gücü ve fikirlerle yazılır. Ama algoritmayı daha çok belli bir kitleye yönelik bir kitap olarak düşünebiliriz.

Projemi yaparken kendimi algoritma mantığı ve programlama konusunda çok geliştirdim. Algoritma mantığı konusunda gelişmem günlük hayattaki düşünme becerimi de çok etkiledi. Artık birçok şeye daha yapıcı ve çözümsel bir biçimde yaklaşabiliyorum.

Kriter D -Yansıtma

Ben bu projeyi yaparken süreç içerisinde sabırlı olmanın aslında ne kadar olayları çözücü olduğunun farkına vardım. Mesela algoritmamda bir kütüphane yüklenemediğinde hemen stres olmak yerine daha sabırla yaklaşıp problemin tam olarak nerede olduğunu bulmak ve onu çözmek için araştırma yapınca aslında sabrın projedeki büyük yerini farkına vardım. Algoritma işleri düşünce ve bilginin eseri olduğu kadar sabrın da eseri olduğunu kavradım. Örnek verecek olursak insanlar bazı büyük çaplı yapay zeka ve nesne tanıma datasetleri ve programları için yıllarını veriyor ve bu işler sabırla olmayacak şeyler değil. Ben bu projemi sabırlı olarak tamamladığımı söyleyebilirim. Hata çıkınca sabırla onu hallettim. Bu sayede stresime yenik düşüp daha çok sorunlar ortaya çıkarmadım.

Benim kendi yapabilirliklerim açısından bu projenin en iyisi olduğunu düşünüyorum. Çünkü bu proje gayet karmaşık fonksiyonlar, denklem ve topoloji konusunda birçok bilgi içeriyor

ve bazen de gerektiriyor. Bu nedenle bu algoritmaları yazmak benim seviyem için yeterince yüksek ve aşılması zor bir noktaydı. Programlama konusunda uzun zamandır uğraşıyorum ve bu konu benim ilgimi en çok çeken ve bana anlaşılması en zor gelen konuydu. Bu konu üzerinde çalışmak beni çok zorladı ama bu konumun da üstünde bir proje yapmaya çalışmak benim seviyem için uygun ve sağlıklı olmazdı.

Bu projeyi tekrar yapacak olsaydım. Kesinlikle proje konusuyla ilgili 2 yıl öncesinden araştırma yapıp öğrenmeye başlardım çünkü bu konu çok çeşitli bilgiye ev sahipliği yapıyor ve içerisinde birçok detay var. Ben yazın bu projeye başlamış olmama rağmen bitirmek ve bu bilgileri sindirmek çok zamanımı aldı. Ama bence programlama konusunda çok doğru zamanda python ve visual studio konularında eğitim almışım yoksa bu iki programı da bu yıla bıraksaydım kesinlikle projemi yetiştiremez ve kendime hiçbir şey katamazdım. Ben ortaya çıkardığım sonuçtan son derece memnun ve mutluyum. Çünkü bu konu hakkında çok fazla merak ve heyecan içerisindeydim ve merak duyduğum bu konu üzerinde bir sonuç elde etmek beni çok mutlu etti.

Sözcük Sayısı

1984

Kaynakça/Alıntılanan Kaynaklar

Kendime özel internet defterim:
http://localhost:8888/notebooks/object_detection_tutorial.ipynb

IBM Knowledge Center, www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zconcepts/zconc_datasetintro.htm.

Thomas-Krenn. “Cmd Commands under Windows.” Thomas, Thomas-Krenn.AG, 27 May 2019, www.thomas-krenn.com/en/wiki/Cmd_commands_under_Windows.

Rouse, Margaret. “What Is Image Recognition? - Definition from WhatIs.com.” SearchEnterpriseAI, TechTarget, 6 May 2017, searchenterpriseai.techtarget.com/definition/image-recognition.

Journal, IJCSMC. “Analysis of Face Recognition Using Manhattan Distance Algorithm with Image Segmentation.” IJCSMC, www.academia.edu/7561145/Analysis_of_Face_Recognition_using_Manhattan_Distance_Algorithm_with_Image_Segmentation_.

“Anaconda Python/R Distribution - Free Download.” *Anaconda*, www.anaconda.com/distribution/.

İzlediğim Stanford Üniversitesi’nin dersleri:

https://www.youtube.com/watch?v=d14TUNcbn1k&list=PL3FW7Lu3i5JvHM8ljYj-----zLfQRF3EO8sYv&index=4

https://www.youtube.com/watch?v=vT1JzLTH4G4&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=1

https://www.youtube.com/watch?v=OoUX-nOEjG0&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=2

https://www.youtube.com/watch?v=h7iBpEHGVNc&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=3

https://www.youtube.com/watch?v=bNb2fEVKeEo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=5

https://www.youtube.com/watch?v=wEoyxE0GP2M&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=6

https://www.youtube.com/watch?v=_JB0AO7QxSA&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=7

https://www.youtube.com/watch?v=6SlgtELqOWc&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=8

https://www.youtube.com/watch?v=DAOcjicFr1Y&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=9

https://www.youtube.com/watch?v=6niqTuYFZLQ&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=10

https://www.youtube.com/watch?v=nDPWywWRIRo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=11

https://www.youtube.com/watch?v=6wcs6szJWMY&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=12

https://www.youtube.com/watch?v=5WoItGTWV54&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=13

https://www.youtube.com/watch?v=lvoHnicueoE&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=14

https://www.youtube.com/watch?v=eZdOkDtYMoo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=15

https://www.youtube.com/watch?v=ClfsB_EYsVI&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=16

"A Beginner's Guide to Generative Adversarial Networks (GANs)." *Pathmind*, pathmind.com/wiki/generative-adversarial-network-gan.

Gandhi, Rohith. "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms." *Medium*, Towards Data Science, 9 July 2018, towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

"Online Courses - Anytime, Anywhere." *Udemy*, www.udemy.com/course/bilgisayar-gorusu/learn/lecture/9544010#content.

Brownlee, Jason. "Introduction to Python Deep Learning with Keras." *Machine Learning Mastery*, 13 Sept. 2019, machinelearningmastery.com/introduction-python-deep-learning-library-keras/.

"NumPy¶." NumPy, numpy.org/.

Elfouly, Sharif. "R-CNN." *Medium*, Towards Data Science, 18 July 2019, towardsdatascience.com/r-cnn-3a9beddfd55a.

Berber, Burak. "CNN - RCNN - FAST RCNN - FASTER RCNN Nedir?" Teknofesor, 30 Oct. 2019, teknofesor.com/cnn-rcnn-fast-rcnn-faster-rcnn-nedir/.

Ekler

Cifar10 Veriseti Nedir?

CIFAR-10 veri seti, 10 sınıftan ve 3 kanalda 32 x 32 piksel boyutunda 60.000 renkli görüntü içerir. Her sınıfta 6.000 resim vardır. Eğitim seti 50.000 görüntü içerirken, test setlerinde 10.000 görüntü vardır. Örnek olarak aşağıda 10 tane

ardır.

ıe tanıma programının girdi olarak

egorilerden birine atamaktır.

Tensorflow nedir?

Derin öğrenme ve nesne tanıma algoritmalarında en çok kullanılan kütüphanelerden birisidir. Tensorflow, Google tarafından geliştirilen açık kaynaklı bir kütüphanedir. Esnek yapıya sahip bir kütüphanedir. Birçok Cpu veya Gpu'da kurma imkanı sunabilmektedir.

R-CNN nedir?

CNN nedir?

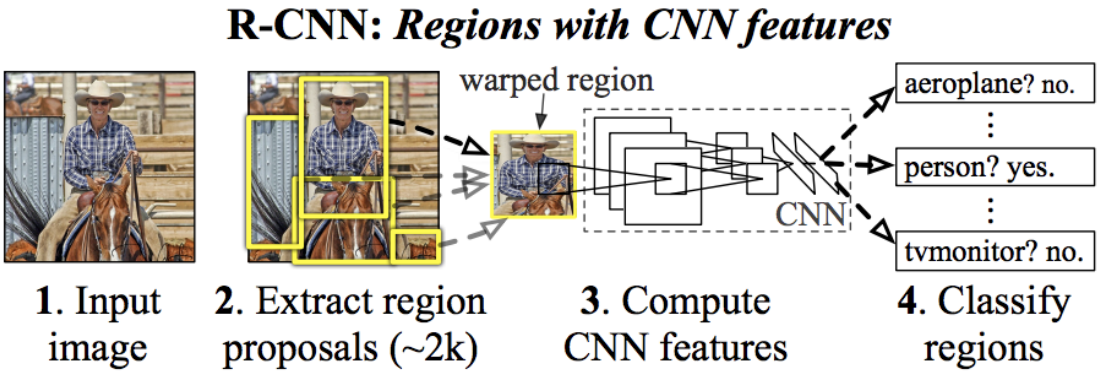
CNN resim ve video işleme için geliştirilen bir derin öğrenme ağ çeşididir. CNN genel olarak 4 katmandan oluşur. Bunlar: Convolution layer, Relu layer, Pooling layer ve Flattening layer.

Convolution layer katmanında resmin üzerinde bir filtre gezdirilir. Filtre, bulunduğu piksellerde hesaplamalar yapar. Filtrenin gezdirilmesi sonucunda bir matris ortaya çıkar ve bu matrise "feature map" denir. Bir CNN ağında birden çok filtre gezdirilir ve filtreler özelliklerine göre şekillenir.

Relu layer katmanı Convolution layer katmanından sonra gelir ve gelen veride eksi değerleri sıfır yapar. Bunun için Relu aktivasyon fonksiyonunu kullanır. Bu sayede 0 olan yerlerde bir nesnenin olmadığı sonucuna varılır.

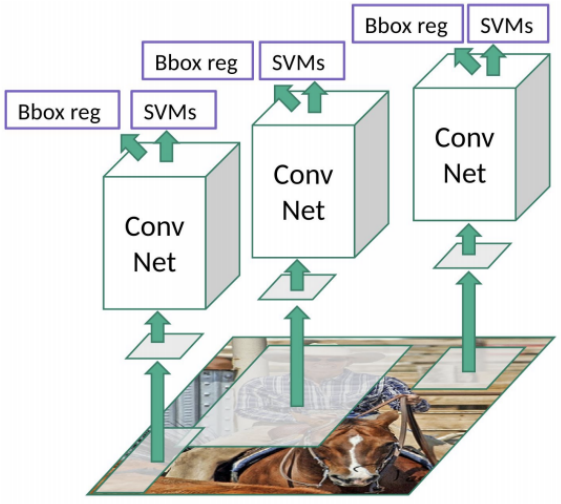
Pooling layer katmanında elimizdeki feature maplerinin boyutları küçültülerek havuzlama yapılır. Boyutlarının küçültülme nedeni ise elimizdeki parametrelerin sayısının azaltılması ve en marjinal ya da kritik parametreleri tutmaktır.

Flattening layer katmanı elimizde matrisi tek bir vektöre çevirir ve neronlar aracılığıyla yapay sinir ağına aktarılır



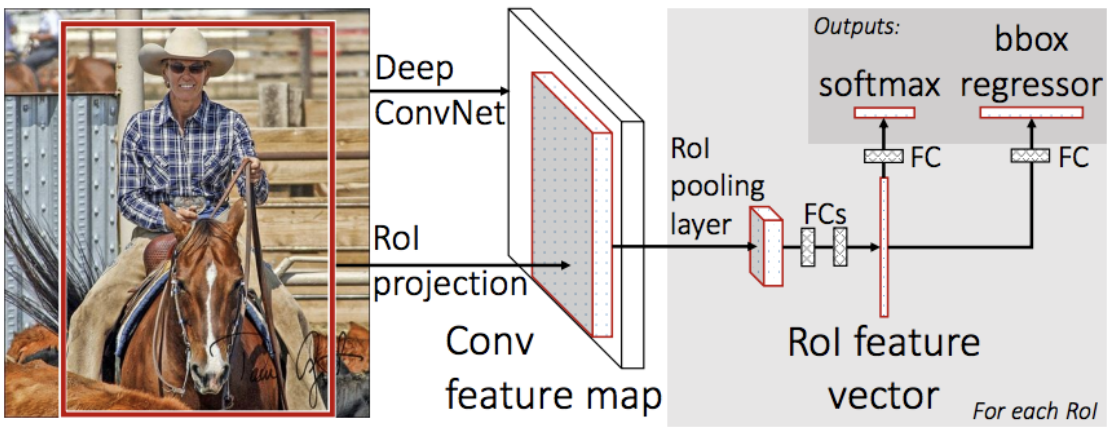
R-CNN nedir?

Resim öncelikle bölge önerilerine yani şüphe duyulan bölgelere bölünür. Daha sonra her bölge için CNN (ConvNet) uygulanır ve çıkan özellik mapleri için sınıflandırma algoritması olan SVM kullanılır. Sonrasında nesnenin var olup olmadığı kontrol edilip ayıklanır. Sonrasında linear regrasyon modellemesi ile bölgelerin boyutları belirlenir ve yapay sinir ağına gönderilir. Bu yöntemin en büyü sıkıntısı zamandır. Her bölge ayrı ayrı CNN'den geçirildiği için eğitim 84 saat sürer. Fotoğraflarda tahmin süresi orta



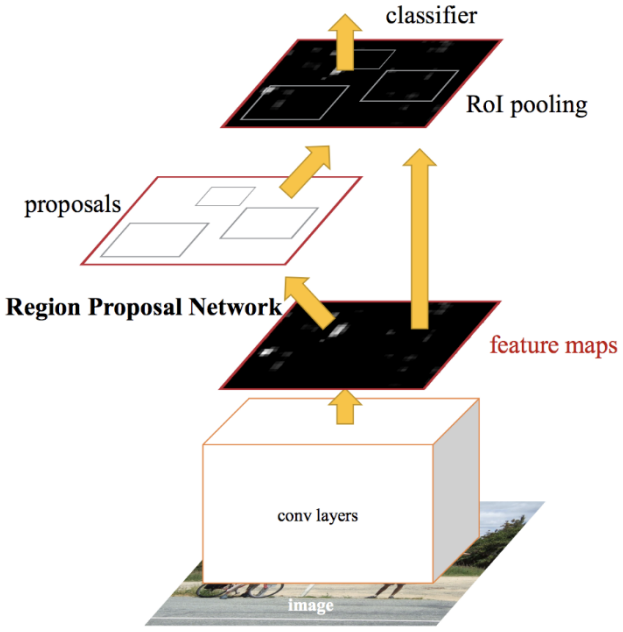
Fast R-CNN nedir?

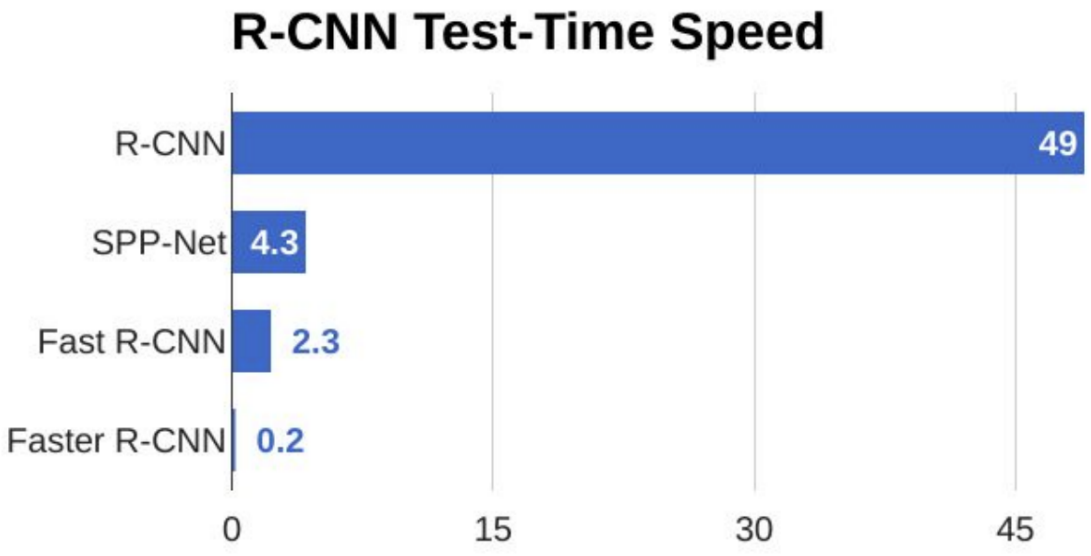
Bu yöntemin R-CNN'den farkı önce resmi bölgelere bölmek yerine CNN uygulanır ve bu sayede her bölge için ayrı ayrı uygulamaya gerek kalmaz. Sonrasında oluşan harita üzerinde bölge önerileri yapılır. Ayrıca sınıflandırma metodu olarak SVM yerine yapay sinir ağları katmanları içerisinde gerçekleşen softmax classifacition kullanılır. CNN' sadece bir kere kullandığı için zamandan tasarruf sağlar. Eğitim süresi 8.50 saattir. Tahmin süresi ise yaklaşık 2.5 saniyedir.



Faster R-CNN nedir?

Bu yöntemde de ilk CNN uygulanıp özellik haritası oluşturuluyor. Bölge önerileri kısmında seçici bölge araması yerine ayrı bir bölge önerisi ağı oluşturarak bölgeleri seçiyoruz. Geri kalan kısımlar Fast R-CNN ile neredeyse aynı. Bu teknikle tahmin süresini 0.3 saniyeye kadar düşürüyoruz.

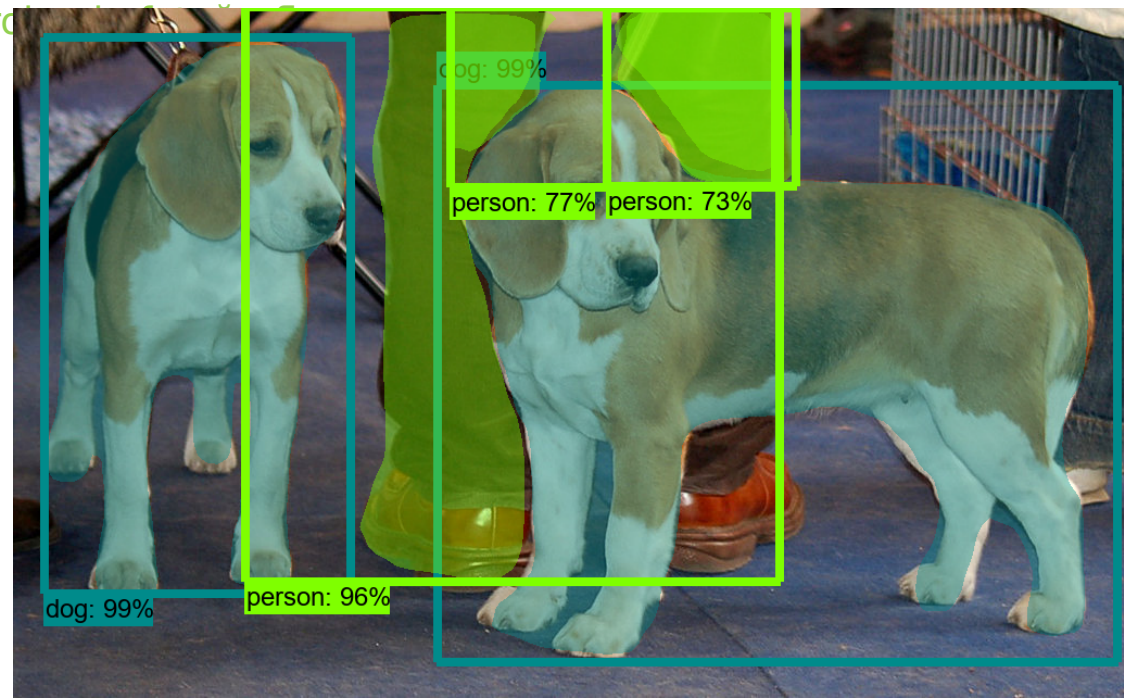


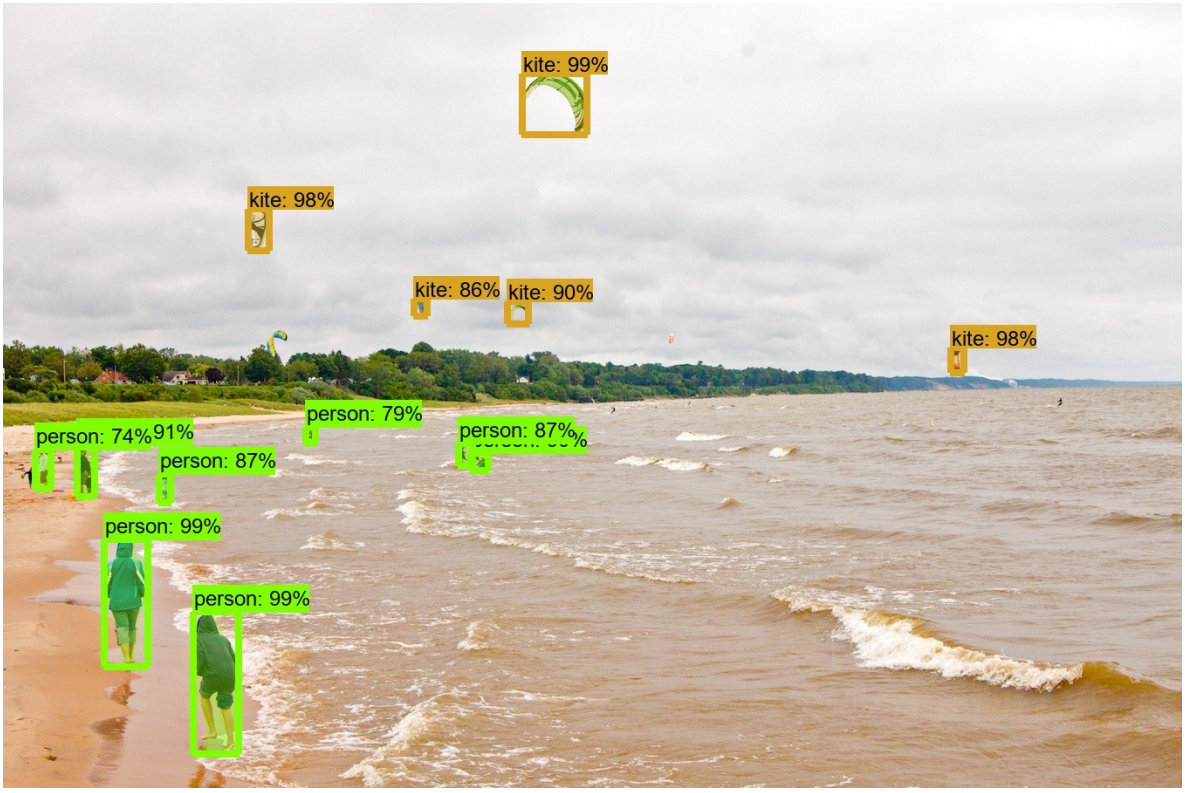


Süreç Çizelgesi

Yapılış Tarihi	Amaç ve Süreç
21.10.2019 Toplantı	Proje konusu hakkında bilgi verildi ve proje konusu kararlaştırıldı.
4.11.2019 Toplantı	Küresel bağlam seçildi. Proje konusu nesne tanıma algoritmaları olarak seçildi. Küresel bağlam bilimsel ve teknik inovasyon olarak seçildi.
5.11.2019 – 22.12.2019 Proje çalışması	Kullanılacak program, matematiksel işlemler ve algoritmalar seçildi ve derlendi. Programın yazımına başlandı.
23.12.2019 Toplantı	Süreç günlüğünün ilk hali teslim edildi. Araştırma ve yapılanlar hakkında geri bildirim alındı.
24.12.2019 – 4.01.2020 Proje çalışması	Program yazımında farklı datasetler ve kütüphaneler kullanıldı. Farklı stratejik algoritmalar araştırılıp denendi.
24.12.2019 – 4.01.2020 Raporlar	Örnek rapor incelenerek uyarlama yapıldı. Arşivlenmiş bilgilerin rapora yayılacağı düzen belirlendi.
3.02.2020 Toplantı	Rapor taslağı teslim edilip geri bildirim alındı. Eksikler düzeltildi ve taslak son halini almaya başladı.

Pr






```
Administrator: Anaconda Prompt (Anaconda3)

(base) C:\Windows\system32>C:\tensorflow1\models\research
'C:\tensorflow1\models\research' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Windows\system32>cd C:\tensorflow1\models\research

(base) C:\tensorflow1\models\research>C:\protoc\bin\protoc object_detection/protos/*.proto --python_out=.

(base) C:\tensorflow1\models\research>cd object_detection

(base) C:\tensorflow1\models\research\object_detection>jupyter notebook
[I 15:30:29.504 NotebookApp] JupyterLab extension loaded from C:\Users\Windows\Anaconda3\lib\site-packages\jupyterlab

[I 15:30:29.505 NotebookApp] JupyterLab application directory is C:\Users\Windows\Anaconda3\share\jupyter\lab
[I 15:30:29.507 NotebookApp] Serving notebooks from local directory: C:\tensorflow1\models\research\object_detection
[I 15:30:29.507 NotebookApp] The Jupyter Notebook is running at:
[I 15:30:29.507 NotebookApp] http://localhost:8888/?token=e8a68bce1e1a213b417a8725ab8f3776b2897401b351084a
[I 15:30:29.507 NotebookApp]  or http://127.0.0.1:8888/?token=e8a68bce1e1a213b417a8725ab8f3776b2897401b351084a
[I 15:30:29.507 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation)
.
[C 15:30:29.550 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Windows/AppData/Roaming/jupyter/runtime/nbserver-29344-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=e8a68bce1e1a213b417a8725ab8f3776b2897401b351084a
     or http://127.0.0.1:8888/?token=e8a68bce1e1a213b417a8725ab8f3776b2897401b351084a
[I 15:31:36.135 NotebookApp] Kernel started: 60d41d11-f6fe-4186-b1cc-d210178282de
[I 15:33:15.917 NotebookApp] Saving file at /object_detection_tutorial.ipynb
[I 15:33:48.126 NotebookApp] Interrupted...
[I 15:33:48.126 NotebookApp] Shutting down 1 kernel
[I 15:33:53.152 NotebookApp] Kernel shutdown: 60d41d11-f6fe-4186-b1cc-d210178282de

(base) C:\tensorflow1\models\research\object_detection>
```

```
198
199
200 def run_inference_for_single_image(model, image):
201   image = np.asarray(image)
202   # The input needs to be a tensor, convert it using `tf.convert_to_tensor`.
203   input_tensor = tf.convert_to_tensor(image)
204   # The model expects a batch of images, so add an axis with `tf.newaxis`.
205   input_tensor = input_tensor[tf.newaxis,...]
206
207   # Run inference
208   output_dict = model(input_tensor)
209
210   # All outputs are batches tensors.
211   # Convert to numpy arrays, and take index [0] to remove the batch dimension.
212   # We're only interested in the first num_detections.
213   num_detections = int(output_dict.pop('num_detections'))
214   output_dict = {key:value[0, :num_detections].numpy()
215                  for key,value in output_dict.items()}
216   output_dict['num_detections'] = num_detections
217
218   # detection_classes should be ints.
219   output_dict['detection_classes'] = output_dict['detection_classes'].astype(np.int64)
220
221   # Handle models with masks:
222   if 'detection_masks' in output_dict:
223     # Reframe the the bbox mask to the image size.
224     detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
225               output_dict['detection_masks'], output_dict['detection_boxes'],
226                image.shape[0], image.shape[1])
227     detection_masks_reframed = tf.cast(detection_masks_reframed > 0.5,
228                                        tf.uint8)
229     output_dict['detection_masks_reframed'] = detection_masks_reframed.numpy()
230
231   return output_dict
232
233
234 # Run it on each test image and show the results:
235
236 # In[ ]:
237
238
239 def show_inference(model, image_path):
240   # the array based representation of the image will be used later in order to prepare the
241   # result image with boxes and labels on it.
```