



**YAZILIM MÜHENDİSLİĞİ
GÖRÜNTÜ İŞLEME**

ÖZGE GÜRBÜZ

MAYIS , 2022

AMAÇ

Görüntü işleme, bilgisayarların görüntüler ve videolardaki nesneleri ve kişileri tanımlamasını ve anlamasını sağlamaya odaklanan bir bilgisayar bilimi alanıdır. Diğer yapay zeka türleri gibi görüntü işleme de insan yeteneklerini kopyalayan görevleri gerçekleştirmeyi ve otomatikleştirmeyi amaçlar. Bu durumda, görüntü işleme hem insanların görme biçimini hem de gördüklerini anlamlandırma biçimini kopyalamaya çalışır.

PROJENİN İÇERİĞİ

1. Görüntü işleme aşamaları
2. YOLOv4 modeli ve Google Colaboratory ile eğitim ve madeni para tanıma
3. OpenCV ile madeni para tanıma ve toplamı hesaplama

1. Görüntü İşleme Aşamaları

- a) Veri alımı: Sayısal görüntü sayısal kamera ile elde edilir.
- b) Ön işleme: Sayısal görüntütüyü kullanmadan önce bazı ön işlemlerden geçirmek gereklidir. Bunlar; görüntü iyileştirme, görüntü onarma, görüntü sıkıştırma işlemleridir.
- c) Özellik seçimi veya çıkarımı: Birçok doğrusal veya doğrusal olmayan boyut indirgeme yönteminden yararlanılmaktadır.
- d) Model Oluşturma: Eğiticisiz öğrenme yöntemleri etiket bilgileri olmaksızın (verinin ne olduğunu bilmeksizin) veride gizli bulunan yapıyı ortaya çıkarmaya çalışır. Eğiticili öğrenme yöntemlerinde, eğiticisiz yöntemlerden farklı olarak etiketli veriden yararlanılarak model oluşturulmaya çalışılır. Takviyeli öğrenme yöntemlerinde, eğiticili öğrenme yöntemlerinden farklı olarak, eğitim modeli oluşturulurken her adımda etiket bilgileri sağlayan bir eğitici olmaması durumunda sistem, ödül ve ceza kavramlarında yararlanılarak istenilen modeli en başarılı biçimde oluşturmaya çalışır.
- e) Son işleme: Elde edilen sınıflandırma veya kümeleme sonuçlarının iyileştirilmesi için yapılan her türlü işlem son işleme başlığı altında işlenebilir.

2. Yolov4 Modeli İle Eğitim Ve Madeni Para Tanıma

Madeni para tanıma ve toplamını hesaplama projesinde YOLOv4 modeli ve Google Colaboratory kullanılmıştır. Google Colab (Google Colaboratory), yapay zeka ve derin öğrenme projeleri üzerinde çalışanlar için etkileşimli, tamamen bulut tabanlı, kullanımı kolay ve ortak çalışmaya dayalı bir programlama ortamıdır. YOLOv4 (You Only Look Once), nesne algılamalarını gerçekleştirmek için derin evrişimli sinir ağlarını kullanan son teknoloji bir algoritmadır. YOLO algoritmasının diğer algoritmalarдан daha hızlı olmasının sebebi resmin tamamını tek seferde nöral bir ağdan geçiriyor olmasıdır. YOLO algoritması görüntüler üzerinde tespit ettiği nesnelerin çevresini bounding box ile çevreler.

Data Toplama / Etiketleme :

Data etiketleme işlemi LabelImg programı ile yapılmıştır.



Örnek olması açısından 1 adet görüntü etiketlenmiştir ve sonucu YOLO formatında export edildiğinde aşağıdaki şekilde bir görüntü matrise sahip olmuştur.



1.txt - Not Defteri

	Dosya	Düzen	Bİçim	Görünüm	Yardım
1	0.726389	0.240972	0.226852	0.318056	
4	0.553241	0.278472	0.147222	0.220833	
2	0.432407	0.416667	0.196296	0.269444	
2	0.306019	0.527083	0.200926	0.268056	
0	0.256944	0.772917	0.182407	0.243056	
3	0.388889	0.713194	0.216667	0.315278	
4	0.612037	0.830556	0.170370	0.222222	
1	0.536111	0.564583	0.238889	0.312500	
3	0.728704	0.579861	0.203704	0.318056	
0	0.807870	0.430556	0.156481	0.230556	

- Görüntüler etiketledikten sonra DarkNet aracılığı ile Google Colab üzerinden kendi YOLO modelimiz eğitilir. YOLO algoritmasının çalıştırılabilmesi için Darknet bizden bazı dosyalar beklemektedir. Bunlardan bir tanesi de .cfg dosyasıdır.
- Config dosyası YOLO içerisindeki sinir ağının başarısını,hızını vs etkileyebilecek özelliklerini bizden talep eder.
- YOLO'nun bizden beklediği bir başka dosya ise .names dosyası.Names içerisinde sınıfların adlarını tutar.Madeni para tespiti uygulaması için 1 lira,50 kurus,25 kurus,10 kurus ve 5 kurus olmak üzere 5 class vardır.
- Ayrıca bütün dosyaların DarkNet klasöründeki konumunu belirten .data uzantılı dosyayı da bizden talep eder.
- Config dosyanızda yapmanız gereken değişiklikler :
(Burada verilen değerler bu değişkenlerin önerilen değerleridir.)
 - Batch=64 ve subdivision 16
 - Max_batches değerini (2000* eğitilen sınıf sayısı) değerine eşitlenir.
 - Steps değerleri (%80 of Max_batches),(%90 of Max_batches) yapılır.
 - [yolo] başlığı altındaki classes değerleri eğitim yaptığınız sınıf sayısı ile değiştirilir.
 - Filters değişkenleri de (eğitim yaptığınız sınıf sayısı + 5)*3 değerine eşitlenir.
 - Aynı klasör içinde obj.data isimle bir dosya oluşturarak içine eğitim yapacağınız nesne sayısını, eğitim yaparken kullanacağınız train.txt,text.txt ve obj.names isimli dosyaların adreslerini ve eğitim sonucu bulduğunuz ağırlıkları kaydedeceğiniz dizin yazılır.

classes = 5
train = data/train.txt
valid = data/test.txt
names = data/obj.names
backup = backup

```

+ Kod + Metin
+ '/content/darknet'

Eğitime Başlama
!./darknet detector train spot_data/spot.data spot_yolov4.cfg yolov4.conv.137 -map -dont_show
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.824752), count: 12, class_loss = 1.510454, iou_loss = 6.511600, total_bbox = 6591570, rewritten_bbox = 0.247923 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.861424), count: 11, class_loss = 4.001787, iou_loss = 120.228615, total_bbox = 6591570, rewritten_bbox = 0.247923 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.882443), count: 56, class_loss = 5.306380, iou_loss = 198.750534, total_bbox = 6591647, rewritten_bbox = 0.247920 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.803073), count: 16, class_loss = 0.935812, iou_loss = 11.463598, total_bbox = 6591647, rewritten_bbox = 0.247920 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.849451), count: 10, class_loss = 3.418629, iou_loss = 118.314117, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.902724), count: 49, class_loss = 5.399725, iou_loss = 160.963791, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.8666281), count: 24, class_loss = 2.949909, iou_loss = 17.749325, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000002, iou_loss = 0.000000, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.892590), count: 3, class_loss = 0.000024, iou_loss = 9.535302, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.742810), count: 1, class_loss = 0.000010, iou_loss = 0.425107, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000221, iou_loss = 0.000000, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.737377), count: 7, class_loss = 6.164109, iou_loss = 6.626099, total_bbox = 6591730, rewritten_bbox = 0.247917 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.696451), count: 16, class_loss = 12.174070, iou_loss = 4.634880, total_bbox = 6591730, rewritten_bbox = 0.247916 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000003, iou_loss = 0.000000, total_bbox = 6591730, rewritten_bbox = 0.247916 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.904319), count: 21, class_loss = 0.031726, iou_loss = 47.996639, total_bbox = 6591730, rewritten_bbox = 0.247916 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.844876), count: 13, class_loss = 0.073276, iou_loss = 7.875033, total_bbox = 6591730, rewritten_bbox = 0.247914 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000025, iou_loss = 0.000000, total_bbox = 6591730, rewritten_bbox = 0.247914 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.849828), count: 16, class_loss = 3.182071, iou_loss = 26.444944, total_bbox = 6591730, rewritten_bbox = 0.247914 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.805424), count: 19, class_loss = 1.429391, iou_loss = 8.027161, total_bbox = 6591826, rewritten_bbox = 0.247913 %

```

- Eğitime, Transfer Learning yaparak eğitilmiş ağıın ağırlıkları ile başlanmıştır(yolov4.conv.137).
- Eğitim aşamasını tamamlayıp .weights uzantılı dosyaya sahip olunmuştur. Artık model hazır.

TEST :

```

+ Kod + Metin
+ '/content/darknet'

Test
!./darknet detector test spot_data/spot.data spot_yolov4.cfg /content/drive/MyDrive/spot_weights/backup/spot_yolov4_last.weights /content/darknet/data/1.jpg
4 sn.
15 conv   64      1 x 1/ 1    104 x 104 x 64 -> 104 x 104 x 64 0.089 BF
16 conv   64      3 x 3/ 1    104 x 104 x 64 -> 104 x 104 x 64 0.797 BF
17 Shortcut Layer: 14, wt = 0, wn = 0, outputs: 104 x 104 x 64 0.001 BF
18 conv   64      1 x 1/ 1    104 x 104 x 64 -> 104 x 104 x 64 0.089 BF
19 conv   64      3 x 3/ 1    104 x 104 x 64 -> 104 x 104 x 64 0.797 BF
20 Shortcut Layer: 17, wt = 0, wn = 0, outputs: 104 x 104 x 64 0.001 BF
21 conv   64      1 x 1/ 1    104 x 104 x 64 -> 104 x 104 x 64 0.089 BF

```



CoinDetector.ipynb ☆

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

+ Kod + Metin

```

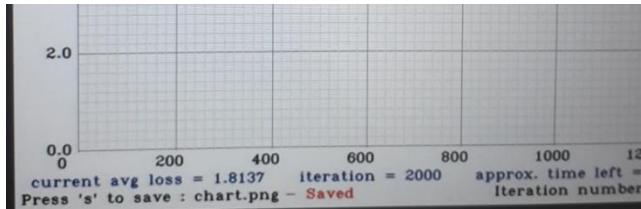
155 conv 1024      3 x 3/ 1    13 x 13 x 512 -> 13 x 13 x 1024 1.595 BF
156 conv 512       1 x 1/ 1    13 x 13 x 1024 -> 13 x 13 x 512 0.177 BF
157 conv 1024      3 x 3/ 1    13 x 13 x 512 -> 13 x 13 x 1024 1.595 BF
158 conv 512       1 x 1/ 1    13 x 13 x 1024 -> 13 x 13 x 512 0.177 BF
159 conv 1024      3 x 3/ 1    13 x 13 x 512 -> 13 x 13 x 1024 1.595 BF
160 conv 30        1 x 1/ 1    13 x 13 x 1024 -> 13 x 13 x 30 0.010 BF
161 yolo

[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedyNMS (1), beta = 0.600000
Total BFLOPS 59.592
avg_outputs = 490384
Allocate additional workspace_size = 52.43 MB
Loading weights from /content/drive/MyDrive/spot_weights/backup/spot_yolov4_last.weights...
seen 64, trained: 128 K-images (2 Kilo-batches_64)
Done! Loaded 162 layers from weights-file
Detection layer: 139 - type = 28
Detection layer: 150 - type = 28
Detection layer: 161 - type = 28
/content/darknet/data/1.jpg: Predicted in 42.836000 milli-seconds.

10 kurus: 100%
25 kurus: 100%
50 kurus: 100%
25 kurus: 100%
1 lira: 100%
5 kurus: 100%
5 kurus: 100%
1 lira: 100%
50 kurus: 100%
10 kurus: 100%

```

Buradaki görselde, verilen fotoğraftaki paraların başarı oranı görülmektedir.



Anlık Ortalama loss değeri (1.8137) görülmektedir.

Eğitime devam etme :

CoinDetector.ipynb ☆

Dosya Düzene Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

+ Kod + Metin spot_data/spot_images/IMG_000000000000.jpg

RAM Disk Düzenleme

Eğitime devam

```
[ ] ./darknet detector train spot_data/spot.data spot_yolov4.cfg /content/drive/MyDrive/spot_weights/backup/spot_yolov4_last.weights -dont_show
```

Görsütlenen çıkış son 5000 satırı kısaltıldı.

1981: 1.178304, 1.808675 avg loss, 0.000130 rate, 13.600040 seconds, 126784 images, 0.317371 hours left
Loaded: 0.000061 seconds
Very small path to the image:
Very small path to the image:
Very small path to the image:
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.937200), count: 1, class_loss = 0.000068, iou_loss = 11.236680, total_loss = 11.236748
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.925490), count: 39, class_loss = 0.000517, iou_loss = 121.060547, total_loss = 121.061066
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.907975), count: 17, class_loss = 0.001540, iou_loss = 12.769183, total_loss = 12.770723
total_bbox = 1026818, rewritten_bbox = 0.237822 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.759959), count: 2, class_loss = 1.133919, iou_loss = 28.737415, total_loss = 29.871334
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.944073), count: 19, class_loss = 0.000070, iou_loss = 43.567589, total_loss = 43.567657
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.910457), count: 11, class_loss = 0.000125, iou_loss = 7.650755, total_loss = 7.650880
total_bbox = 1026850, rewritten_bbox = 0.237815 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000007, iou_loss = 0.000000, total_loss = 0.000007
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.932728), count: 3, class_loss = 0.000001, iou_loss = 9.043129, total_loss = 9.043130
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.940510), count: 1, class_loss = 0.000020, iou_loss = 0.847429, total_loss = 0.847448
total_bbox = 1026854, rewritten_bbox = 0.237814 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.939761), count: 1, class_loss = 0.000034, iou_loss = 4.566301, total_loss = 4.566335
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.934943), count: 42, class_loss = 0.000218, iou_loss = 137.328613, total_loss = 137.328827
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.887776), count: 16, class_loss = 0.008392, iou_loss = 16.940273, total_loss = 16.948666
total_bbox = 1026913, rewritten_bbox = 0.237800 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000003, iou_loss = 0.000000, total_loss = 0.000003
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.923821), count: 14, class_loss = 0.059291, iou_loss = 23.265251, total_loss = 23.324541
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.911855), count: 18, class_loss = 0.019149, iou_loss = 9.269444, total_loss = 9.288594
total_bbox = 1026945, rewritten_bbox = 0.237793 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.000000), count: 1, class_loss = 0.000061, iou_loss = 0.000000, total_loss = 0.000061
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.914358), count: 29, class_loss = 0.496007, iou_loss = 79.046303, total_loss = 79.542313
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.906791), count: 12, class_loss = 1.017554, iou_loss = 14.694226, total_loss = 15.701780

1981 iterasyonda, 1.80 avg loss ve 126784 çoğaltılmış resim ile istenildiği takdirde eğitime devam edilebilmektedir.

```
(next mAP calculation at 1800 iterations)
Last accuracy mAP@0.50 = 88.85 %, best = 92.06 %
1789: 3.406183, 2.388879 avg loss, 0.001300 rate, 13.951226 seconds, 114496 images, 0.841509 hours left
Loaded: 0.000029 seconds
Very small path to the image:
```

Accuracy değeri : 88.85% ve best değeri ise : 92.06% dir.

3. OpenCV İle Madeni Para Tanıma Ve Toplami Hesaplama

OpenCV, numpy ve matplotlib kütüphaneleri import edildikten sonra cfg ve last.weight dosyaları da dosyaya eklenir. Gerekli kodlar yazıldıktan sonra ise sonuçlar elde edilir.

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'CoinDetector' with files like 'main.py', 'yolo4.cfg', and 'yolo4_last.weights'. The main editor window shows the Python code for coin detection, which includes importing cv2, dnn, and other modules, defining a function to detect coins, and calculating the total value. The terminal below shows the execution of the script and the resulting output, which lists various coin types and their counts, followed by a total value of 'TOPLAM = 3.8 TL'. The status bar at the bottom indicates the use of NumPy scientific mode.

```
boxes_list.append([start_x, start_y, int(box_width), int(box_height)])  
max_ids = cv2.dnn.NMSBoxes(boxes_list, confidences_list, 0.5, 0.4)  
  
for max_id in max_ids:  
    max_class_id = max_id  
    box = boxes_list[max_class_id]  
  
    start_x = box[0]  
    start_y = box[1]  
    box_width = box[2]  
    box_height = box[3]
```

Run: main
D:\AfterFormat\CoinDetector\venv\Scripts\python.exe D:/AfterFormat/CoinDetector/main.py
['5 kurus', '10 kurus', '25 kurus', '50 kurus', '1 lira'
1 lira
10 kurus
50 kurus
5 kurus
25 kurus
50 kurus
10 kurus
5 kurus
1 lira
25 kurus
TOPLAM = 3.8 TL

Process finished with exit code 0

Version Control Run Python Packages TODO Python Console Problems Terminal Services
Looks like you're using NumPy. Would you like to turn scientific mode on? // Use scientific mode. Keep current layout // Scientific mode provides a tool window layout optimized for data scienc... (yesterday 2 16:1 CRLF UTF-8 4 spaces Python 3.8 (CoinDetector)

