

PREBROJAVANJE OBJEKATA NA POKRETNOSTI TRACI

Loreana Oluić, Nemanja Milutinović I Dušan Antić
Fakultet tehničkih nauka, Novi Sad

Uvod

Detekcijom i prebrojavanjem objekata možemo omogućiti vođenje preciznih statistika o prodaji koje bi pomogle u daljoj organizaciji rada marketa.

Skup podataka

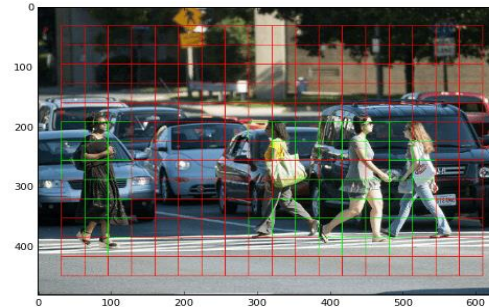
Za potrebe treniranja neuronske mreže koristili smo slike namirnica (banana, jabuka, sendviča, sargarepe i flaše) koje smo pronašli na internetu i ručno labelirali. Za našu mrežu koristili smo ukupno 177 slika koje smo podelili u trening I testni skup

Trening skup : 160 slika (90 %)
Testni skup: 17 slika (10 %)

YOLO je algoritam koji koristi neuronsku mrežu da bi pružio mogućnost detekcije objekata u realnom vremenu. Algoritam je popularan zbog svoje brzine i preciznosti. Koristi se za razne primene od detekcije saobraćajnih signala, ljudi, životinja... Dva osnovna pitanja u detekciji objekata su:
1) Sta je objekat
2) Gdje je objekat
Detekcija objekata u YOLO se radi kao problem regresije i daje verovatnocu klase detektovanih slika. Glavne karakteristike algoritma su brzina, velika preciznost I sposobnost učenja.

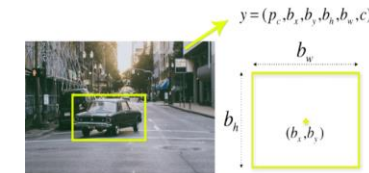
Yolo (you only look once)

Algoritam radi prateći tri tehnike:
1) Rezidualni blokovi

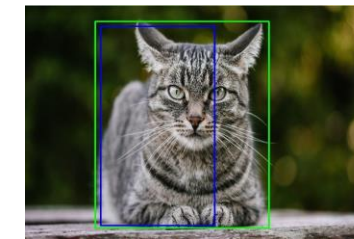


Slika je podeljena u mrežu gdje je svaka ćelija iste dimenzije. Svaka ćelija će detektovati objekat koji se pojavljuje u njoj.

2) Bounding box regresija
Bounding box je kontura koja ograničava objekat na slici. Sadrži širinu, visinu, klasu i centar konture



3) Intersection over union
YOLO koristi IOU da bi napravio okvir oko objekta na slici. Svaka ćelija predviđa okvir i postotak sigurnosti u predviđanje, YOLO se trudi da predviđeni okvir i stvarni okvir budu jednaki.



Rezultati

Računanje MAP (mean average precision)

Yolov3_custom:

```
detections_count = 118, unique_truth_count = 55
class_id = 0, name = Banana, ap = 87.50% (TP = 3, FP = 1)
class_id = 1, name = Apple, ap = 99.35% (TP = 13, FP = 0)
class_id = 2, name = Sandwich, ap = 100.00% (TP = 12, FP = 0)
class_id = 3, name = Carrot, ap = 95.24% (TP = 6, FP = 1)
class_id = 4, name = Bottle, ap = 91.60% (TP = 13, FP = 1)

for conf_thresh = 0.25, precision = 0.94, recall = 0.85, F1-score = 0.90
for conf_thresh = 0.25, TP = 47, FP = 3, FN = 8, average IoU = 73.79 %
```

Yolov3:

```
detections_count = 383, unique_truth_count = 55
class_id = 0, name = Banana, ap = 76.75% (TP = 2, FP = 0)
class_id = 1, name = Apple, ap = 74.46% (TP = 8, FP = 2)
class_id = 2, name = Sandwich, ap = 31.63% (TP = 4, FP = 4)
class_id = 3, name = Carrot, ap = 54.59% (TP = 2, FP = 1)
class_id = 4, name = Bottle, ap = 46.13% (TP = 4, FP = 1)

for conf_thresh = 0.25, precision = 0.71, recall = 0.36, F1-score = 0.48
for conf_thresh = 0.25, TP = 20, FP = 8, FN = 35, average IoU = 47.42 %
```

Yolov3-tiny:

```
detections_count = 376, unique_truth_count = 55
class_id = 0, name = Banana, ap = 7.57% (TP = 0, FP = 1)
class_id = 1, name = Apple, ap = 37.59% (TP = 7, FP = 4)
class_id = 2, name = Sandwich, ap = 13.61% (TP = 3, FP = 14)
class_id = 3, name = Carrot, ap = 2.38% (TP = 0, FP = 0)
class_id = 4, name = Bottle, ap = 2.81% (TP = 0, FP = 3)

for conf_thresh = 0.25, precision = 0.31, recall = 0.18, F1-score = 0.23
for conf_thresh = 0.25, TP = 10, FP = 22, FN = 45, average IoU = 19.29 %
```

Upoređivanjem rezultata dobijenih različitim algoritmima uviđamo da smo najbolje rezultate dobili primjenom našeg prilagođenog. Dok je najlošije rezultate dao tiny algoritam što je bilo i očekivano