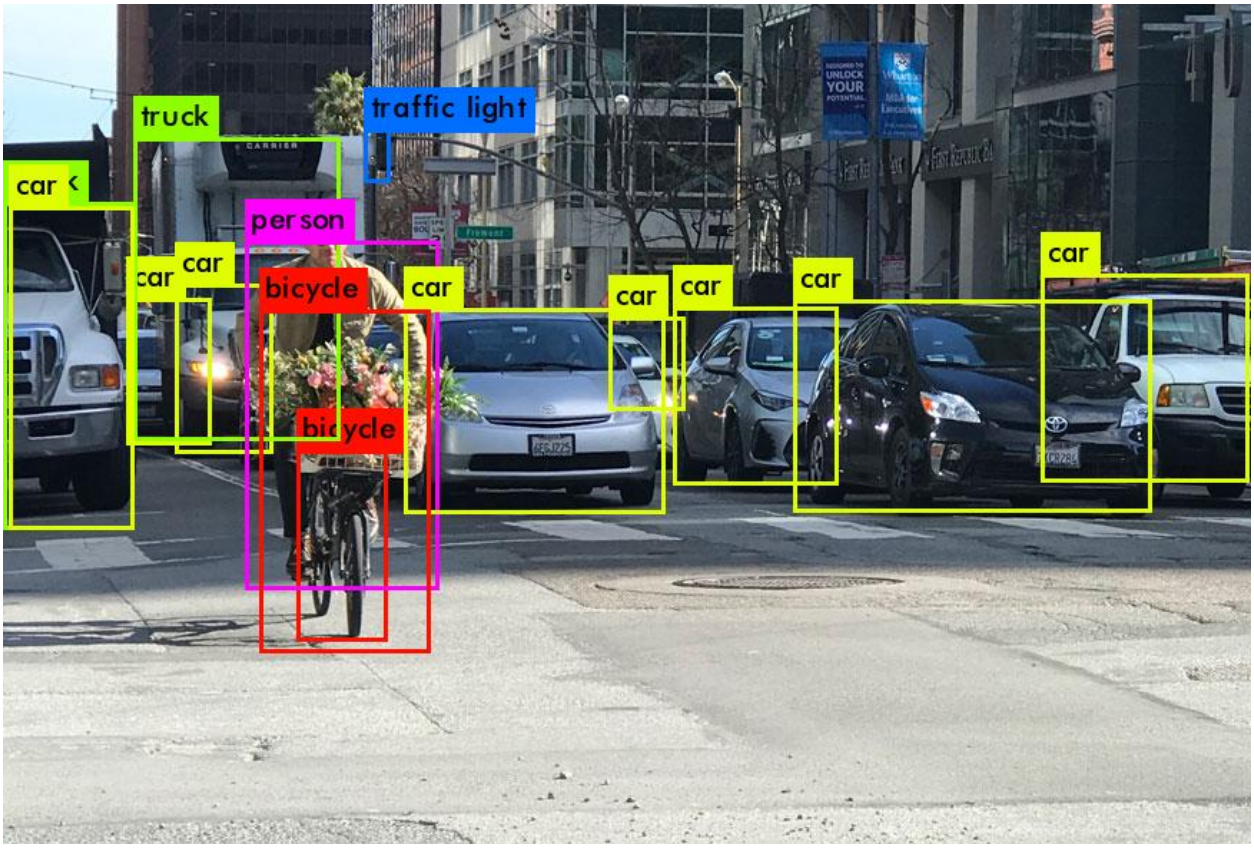
# Projektni zadatak 1
## Priprema skupa podataka za detekciju objekata pomoću YOLO algoritma

## UVOD

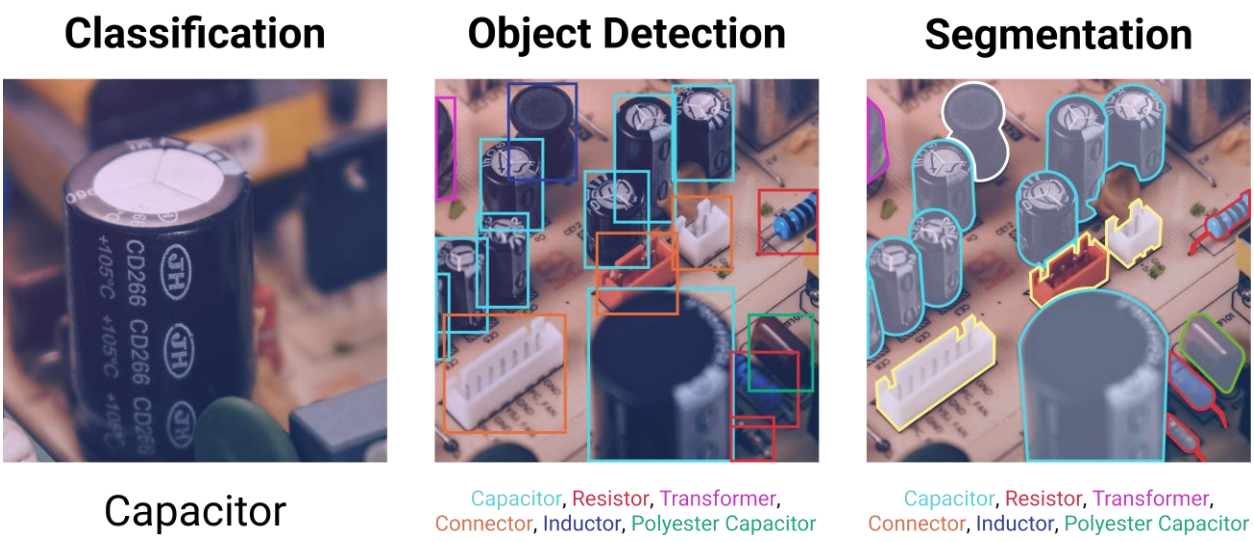
YOLO (eng. *You Only Look Once*) je algoritam za detekciju objekata u realnom vremenu, koji je objavljen 2015. godine. Od prve verzije, algoritam i sama arhitektura modela su unapređivani u nekoliko iteracija. Ovaj algoritam je zasnovan na dubokim konvolucionim neuronskim mrežama (eng. *deep CNN – Convolutional Neural Network*).




**Ilustracija 1 Primer detekcije objekata pomoću YOLOv3 modela**

U prvom projektnom zadatku, potrebno je izvršiti pripremu skupa podataka za treniranje ovog modela za detekciju željenog objekta. U drugom delu zadatka će biti odrađena detekcija i segmentacija.

Vaš zadatak u okviru prvog dela projektnog zadatka je da pripremite vaš skup podataka za detekciju pomoću *Ultralytics* YOLOv8 modela.




**Ilustracija 2. Primer detekcije i segmentacije objekata na slici pomoću Ultralytics YOLOv8 modela**

Ovaj zadatak se sastoji iz 3 celine: pravljenje skupa podataka (slikanje objekta koji će biti detektovan), anotacija skupa podataka i augmentacija skupa podataka.

Kada se napravi skup podataka za određeni zadatak mašinskog učenja, on se obično deli na tri skupa podataka: trening, validacioni i test. Ova tri skupa su međusobno isključiva, odnosno, ne smeju sadržati iste podatke. Trening skup čini većinu celog skupa podataka. On služi da na njemu treniramo naš model da obavi željeni zadatak. Toko treniranja model pokuša da obavi svoj zadatak (npr. detekcija objekata) i da na osnovu tih rezultata detekcije podešavamo parametre svog modela kako bismo poboljšali performanse modela tokom treniranja. Validacioni skup služi da proverimo da li je naš model istreniran i da zaustavimo treniranje ako jeste ili sprečimo negativne efekte prekomernog treniranja. Test skup služi za konačnu meru performansi našeg modela, pri čemu njega ne koristimo za podešavanje parametara tokom treniranja.

Kako YOLO algoritam spada u klasu algoritama sa učenjem sa nadziranjem (eng. *supervised learning*), potrebno je definisati labele koje označavaju tačan izlaz iz algoritma za odgovarajući ulaz u algoritam, koje će se koristiti u toku faze treniranja. U YOLO algoritmu, to predstavljaju pravougaonici koji ograničavaju objekat (eng. *bounding-box*) koji treba detektovati na slici i koji definišu klasu objekta (videti središnju sliku u Ilustraciji 2). Dodatno, treba odraditi i anotaciju validacionog skupa, kako bi mogla biti određena mera performanse algoritma (npr. koliko dobro se poklapaju naš ručno označen bounding-box i bounding-box koji odredi naš model).

Augmentacija skupa podrazumeva proširenje skupa podataka na kojem se trenira algoritam, na osnovu postojećih podataka. Recimo, kada su u pitanju slike, moguće je menjati boju, kontrast, širinu/visinu slike, iseći određeni deo slike, filtriranje slike, dodavanje okluzija... Ovo se radi u cilju poboljšanja performansi algoritma, kako bi model koji se trenira mogao da radi nad „novim“ podacima, bez da moramo da prikupljamo nove podatke.

# ZADACI

**NAPOMENA: VODITI SE SKRIPTOM flow.sh i dokumentacijom s interneta**

## Zadatak 1

Uslikati željeni objekat u više scenarija, iz različith uglova, sa različitim brojem objekata na slici. Gledati da skup podataka bude što raznovrsniji. Skup podataka podeliti na trening skup i validactioni skup (neko pravilo podele je obično 80% za trening i 20% za validacioni i test skup).

## Zadatak 2

Nakon uspešno realizovanog zadatka 1, potrebno je odraditi anotaciju skupa podataka pomoću alata labelImg, tako što će se svakoj slici pridružiti labela.

labelImg omogućava da se putem grafičkog interfejsa (eng. *Graphical User Interface*) generišu labele za tražene klase u našem skupu podataka. To se radi tako što se postavi bounding-box oko objekta na slici, pri čemu će se generisati odgovarajuća labela za taj objekat.

Kako bi program pravilno radio, potrebno je definisati nekoliko stvari:

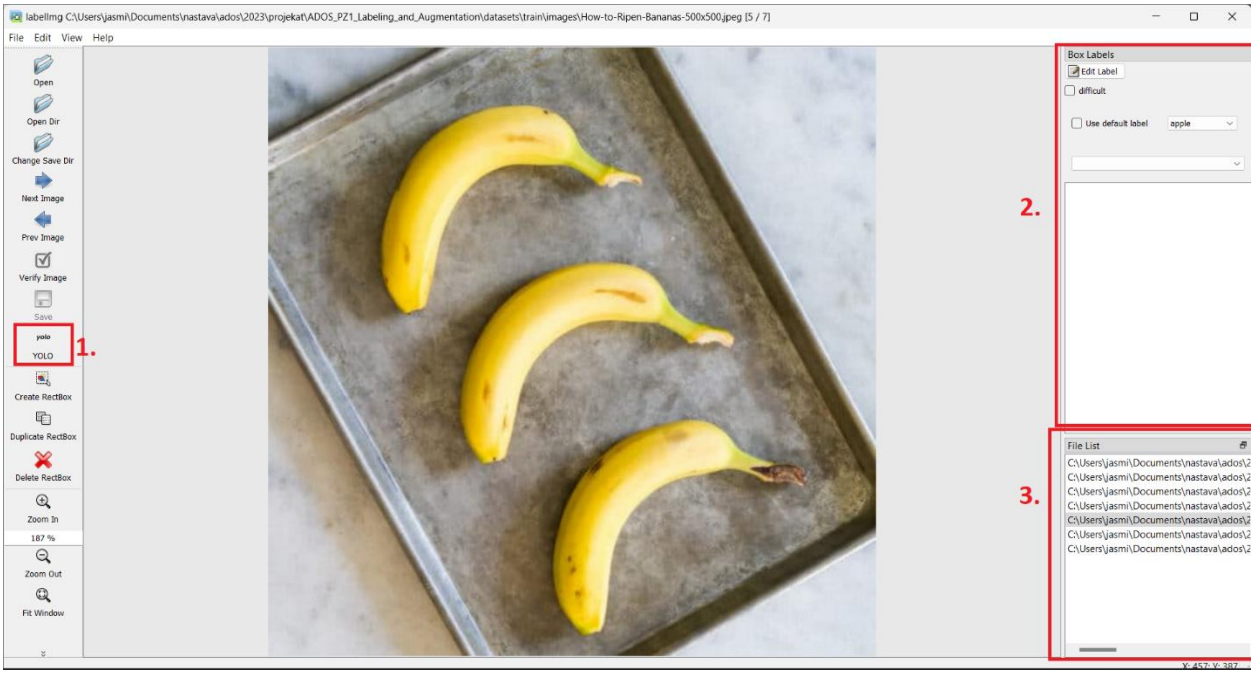
- putanju do direktorijuma gde se nalaze slike koje treba labelirati,
- datoteku predefined_classes.txt gde treba upisati klase objekata koji ćete detektovati,
- putanju do direktorijuma gde će se skladištiti labele u YOLO formatu (kao <naziv_slike>.txt).

Program se pokreće iz komandne linije, gde argumente komandne linije predstavljaju prethodno navedene 3 stavke, tim redom. Pogledati *flow.sh* skriptu za više detalja.

Labele za YOLO detektciju su u formatu koji predstavlja klasu, centar bounding-box-a objekta i njegovu širirnu i visinu relativno u odnosu na širinu i visinu slike:

*<class> <X-coordinate> <Y-coordinate> <relative_width> <relative_height>*

# Uputstvo za labelImg:



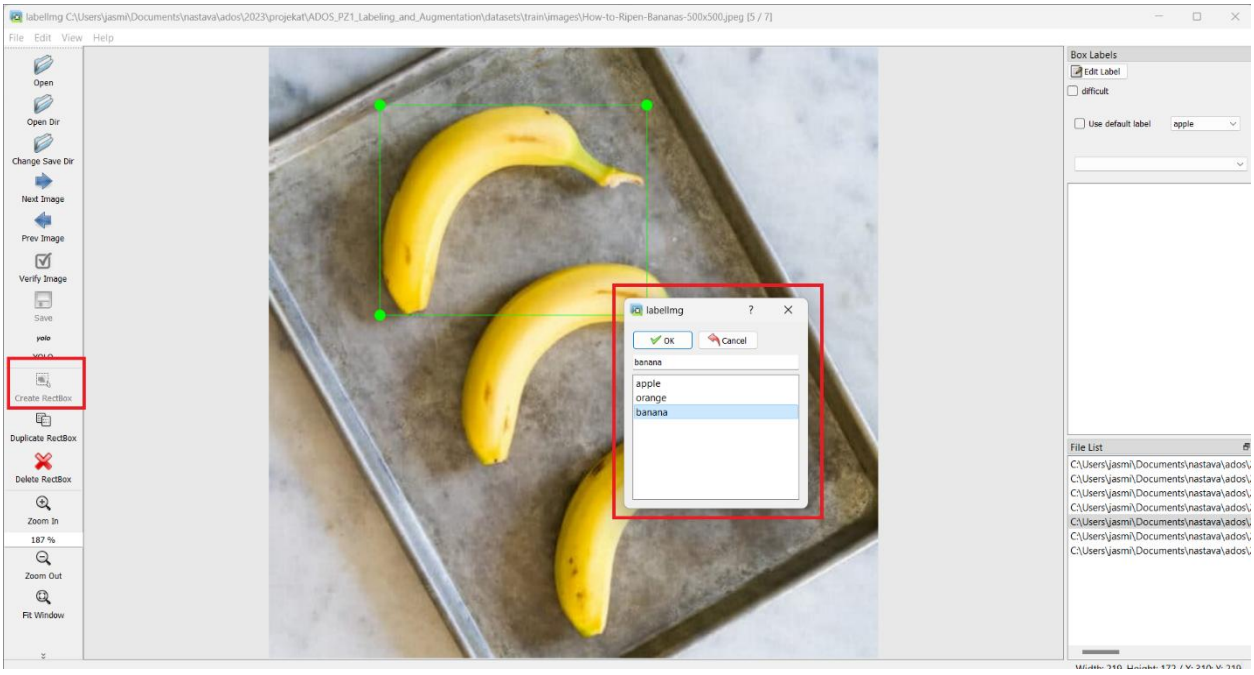
**Ilustracija 3 labelImg alat**

1. Promeniti format labela u YOLO (ukoliko već nije postavljeno) klikom na to dugme (prozor obeležen sa 1.).
2. U prozoru obeleženom sa 2. nalaziće se dodeljene labele na slici. Na početku, spisak će verovatno biti prazan.
3. U prozoru 3, nalazi se spisak slika iz direktorijuma sa slikama.

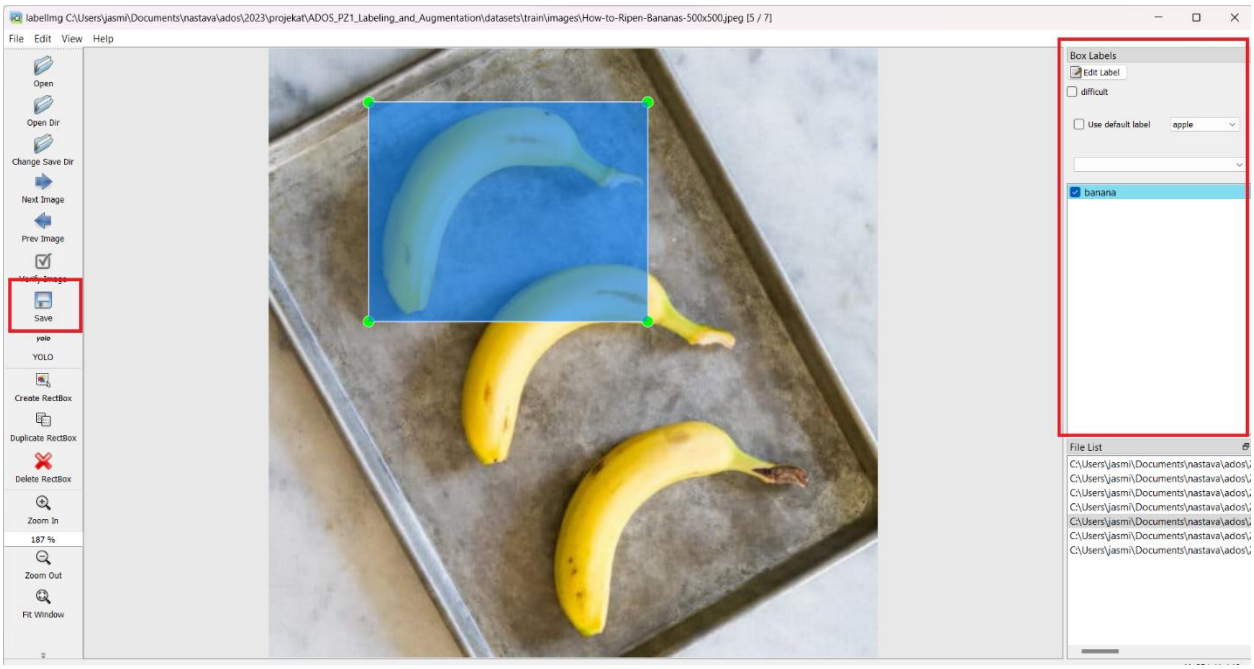
Bounding-box se pravi tako što se klikne na Create RectBox i napravi se pravougaonik oko objekta. Zatim, klikne se na odgovarajuću labelu u pop-up prozoru i na OK. Ukoliko ste dobro definisali putanju do predefined_classes.txt, vaša labela će biti na spisku labela u pop-up prozoru. Ukoliko ne, proverite da li ste prosledili dobru putanju pri pokretanju programa.

Nakon što ste odabrali labelu, ona će se pojaviti na spisku labela na toj slici. Nakon svake labelirane slike, obavezno kliknuti na Save dugme, pa onda preći na sledeću sliku.

Ilustracije 4 i 5 grafički objašnjavaju ove korake.




**Ilustracija 4 Pravljenje labele**




Ilustracija 5 Spisak labela i čuvanje labele u .txt fajlu

# Zadatak 3

U ovom koraku obavlja se augmentacija skupa podataka. U datoteci URL.txt u postavci zadatka, nalazi se link do tabele u kojoj se nalaze navedene 4 vrste augmentacije za vaš objekat. Ove augmentacije treba da implementirate sami. Dat je Qt projekat koji se može koristiti tokom implementacije kako biste potvrdili da vaša augmentacija dobro radi. Na osnovu komande u flow.sh koja radi augmentaciju, možete napisati svoju skriptu da odradi sve vrste augmentacije nad svim slikama iz skupova podataka, tj. možete da automatizujete proces. Možete napisati bash/batch ili python skriptu...

```
#######
# Augmentation.

./ImageDSP/build/linux/build-ImageDSP-Desktop-Debug/ImageDSP \
    --prog-name "Augmentation 1" \
    --in-img "datasets/train/images/IMG001.jpg" \
    --out-img "datasets/train/images/IMG001_a1_threshold=150_compress=40.jpg" \
    --params "150;40"
```

Ilustracija 6 Primer komande za pokretanje programa za augmentaciju skupa podataka iz skripte flow.sh