

# Osnovi računarske inteligencije – predlog projekta

## 1. Naziv teme

Automatska detekcija registarskih tablica.

## 2. Definicija problema

Potrebno je razviti sistem koji može automatski da detektuje i lokalizuje registarske tablice na automobilima sa ulaznih slika ili videa sa kamere. Problem uključuje prepoznavanje tablica u različitim uslovima (osvetljenje, vremenske prilike, uglovi snimanja, delimične zaklonjenosti), kao i izdvajanje područja tablice radi daljeg procesiranja metodama obrade slike, nakon čega se rezultujuća slika prosleđuje modelu za detekciju karaktera.

## 3. Motivacija

Automatska detekcija registarskih tablica nalazi primenu u:

- sistemima nadziranja i kontrole saobraćaja,
- naplati putarina,
- parking sistemima,
- bezbednosnim i nadzornim sistemima.

Ovim se smanjuje potreba za ljudskom intervencijom, ubrzava proces kontrole i unapređuje bezbednost.

## 4. Skup podataka

Zbog ograničenih računarskih resursa i nemogućnosti korišćenja velikih javno dostupnih skupova podataka, kreiran je sopstveni skup podataka. Fotografije automobila su prikupljene sa sajta *polovniautomobili.com*. Nakon prikupljanja, slike su ručno anotirane korišćenjem alata **Label Studio**, pri čemu su registarske tablice označene bounding-box okvirima.

Konačan skup podataka je prilagođen formatu koji koristi **YOLO** model (YOLO Darknet txt format), čime je obezbeđena kompatibilnost sa okruženjem za treniranje.

## 5. Način pretprocesiranja podataka

Nakon prikupljanja i anotacije, nad skupom podataka su sprovedene sledeće faze pretprocesiranja:

Promena dimenzija slika – sve slike se skaliraju na rezoluciju prilagođenu YOLO modelu (npr. 640×640 piksela), radi ujednačenosti i efikasnijeg treninga.

Augmentacija podataka – primenjuju su tehnike kao što su rotacija, horizontalno ogledanje, promena osvetljenja i kontrasta, zamućenje i dodavanje šuma. Na ovaj način se povećava raznovrsnost skupa podataka i postiže robusnost modela na različite uslove u realnom okruženju.

Podela na skupove – podaci se dele na trening, validacioni i test skup, čime se omogućava objektivna evaluacija performansi modela.

## 6. Metodologija

### 6.1 Opis procesa

Cilj sistema je **potpuna automatizacija procesa prepoznavanja registarskih tablica** sa ulazne slike. Metodologija obuhvata sledeće korake:

1. Prikupljanje i anotacija podataka
  - Anotacija tablica je izvršena pomoću *Label Studio* alata i eksportovana u YOLO format.
  - Nad ovim podacima vrši se treniranje YOLO modela.
2. Detekcija registarskih tablica (YOLO)
  - Istreniran model YOLO koristi se za prepoznavanje i lokalizaciju registarskih tablica na slici.
  - Model vraća **koordinate bounding box-ova** sa pripadajućim verovatnoćama.
3. Izdvajanje i obrada slike tablice
  - Ulazna slika se seče na region tablice na osnovu YOLO bounding box koordinata.
  - Nad isečenom slikom primenjuju se klasične tehnike obrade slike
4. Optičko prepoznavanje karaktera (OCR)
  - Obradena slika tablice se prosleđuje OCR modelu (*PaddleOCR*, *Tesseract*, ...).
  - OCR vraća prepoznati niz karaktera sa tablice.
5. Izlaz korisniku
  - Dobijeni niz karaktera predstavlja konačno prepoznatu registarsku tablicu.

### 6.2 Ulaz i izlaz sistema

- Ulaz: slika automobila (JPEG/JPG format).
- Izlaz: tekstualni string koji predstavlja registarsku tablicu.

### 6.3 Dijagram metodologije

Grafički dijagram procesa:



## 7. Način evaluacije

Evaluacija sistema se vrši na dva nivoa. Prvo se meri uspešnost detekcije tablica korišćenjem metrika kao što su *preciznost (Precision)*, *odziv (Recall)*, *mAP (mean Average Precision)* i *IoU (Intersection over Union)*, kako bi se ocenila tačnost lokalizacije. Nakon toga, procenjuje se kvalitet OCR prepoznavanja pomoću *tačnosti karaktera (Character Accuracy)* i *tačnosti cele tablice (Word/Plate Accuracy)*, odnosno poređenjem izlaza modela sa stvarnim tekstom tablice. Na kraju se meri i end-to-end tačnost kompletnog sistema, koja pokazuje procenat slučajeva u kojima je tablica od slike do teksta prepoznata u potpunosti.

## 8. Tehnologije

- Programski jezik: Python
- Biblioteke: OpenCV, PyTorch, NumPy, *Ultralytics YOLO*, PaddleOCR ili Tesseract...
- Alat za anotaciju: Label Studio

## 9. Relevantna literatura

- YOLOv11 dokumentacija - <https://docs.ultralytics.com/models/yolo11/>
- PaddleOCR dokumentacija - <https://github.com/PaddlePaddle/PaddleOCR>
- Tesseract dokumentacija - <https://tesseract-ocr.github.io/tessdoc/>