



UNIVERZITET U БЕОГРАДУ - МАТЕМАТИЧКИ ФАКУЛТЕТ

## Prepoznavanje položaja tela

Andreea C. Citcauan

Septembar, 2024.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Različite implementacije</b>	<b>3</b>
2.1	YOLO model . . . . .	3
2.2	OpenPose model . . . . .	6
2.3	Sopstveni model . . . . .	8
<b>3</b>	<b>Zaključak</b>	<b>9</b>
<b>4</b>	<b>Literatura</b>	<b>11</b>

# 1 Uvod



Figura 1: Detekcija ljudske poze

Detekcija ljudskog položaja, takođe poznata kao detekcija ključnih tačaka, predstavlja tehniku u oblasti računarskog vida koja omogućava identifikaciju različitih položaja tela na slikama i na snimcima. Iako se detekcija poze može primeniti i na različite objekte, posebno se ističe interesovanje za detekciju ljudske poze zbog širokog spektra praktičnih primena, poput primene u raznim sistemima za video nadzor, sportu i fitnesu, zdravlju kao i u animacijama i virtuelnoj realnosti.

Za implementaciju ovog projekta koristimo konvolutivne neuronske mreže. One su posebno pogodne za analizu slika i video zapisa. Koriste se kako bi prepoznale ključne tačke na telu, kao što su zglobovi, ramena, kolena i kukovi na osnovu dostupnih, labeliranih (anotiranih) podataka.

## 2 Različite implementacije

Za rešenje ovog problema dostupna su razna gotova rešenja u vidu modela koji pružaju širok spektar mogućnosti. U ovom radu demonstriraćemo YOLOv8 i OpenPose modele a potom ćemo pokušati da implementiramo sopstveni model za prepoznavanje položaja tela osoba.

### 2.1 YOLO model

U dinamičnom svetu veštačke inteligencije i mašinskog učenja, jedno od najvećih dostignuća je razvijanje efikasnih i pouzdanih modela za procenu položaja. Ultralytics, predvodnik u AI tehnologiji, napravio je značajan iskorak svojim modelom Ultralytics YOLOv8. Model YOLOv8 predstavlja najmoderniju verziju algoritma YOLO (You Only Look Once), koji je postavio nove standarde u prepoznavanju objekata u oblasti računarske vizije, donoseći inovacije koje su značajno unapredile ovu disciplinu.

*YOLOv8 arhitektura*<sup>1</sup> se može uopšteno podeliti na tri glavne komponente:

- **Backbone**-Predstavlja konvolutivnu neuronsku mrežu (CNN) koja je odgovorna za ekstrakciju karakteristika iz ulazne slike. YOLOv8 koristi prilagođenu CSPDarknet53 osnovu, koja primenjuje delimične međuslojne veze kako bi poboljšala protok informacija između slojeva i povećala tačnost
- **Neck** - Vrat, poznat i kao ekstraktor karakteristika, spaja mape karakteristika iz različitih faza glavne strukture (backbone) kako bi prikupio informacije na različitim skalamama. YOLOv8 arhitektura koristi novi C2f modul umesto tradicionalne Feature Pyramid Network (FPN) mreže. Ovaj modul kombinuje semantičke karakteristike visokog nivoa sa prostornim informacijama niskog nivoa, što poboljšava tačnost detekcije, posebno kod manjih objekata.
- **Head** - Glava je odgovorna za pravljenje predikcija. YOLOv8 koristi više modula za detekciju koji predviđaju granične kutije, skorove objektivnosti i verovatnoće klasa za svaku celiju mreže u mapi karakteristika. Ove predikcije se zatim agregiraju kako bi se dobile konačne detekcije.

---

<sup>1</sup> YOLOv8 arhitektura

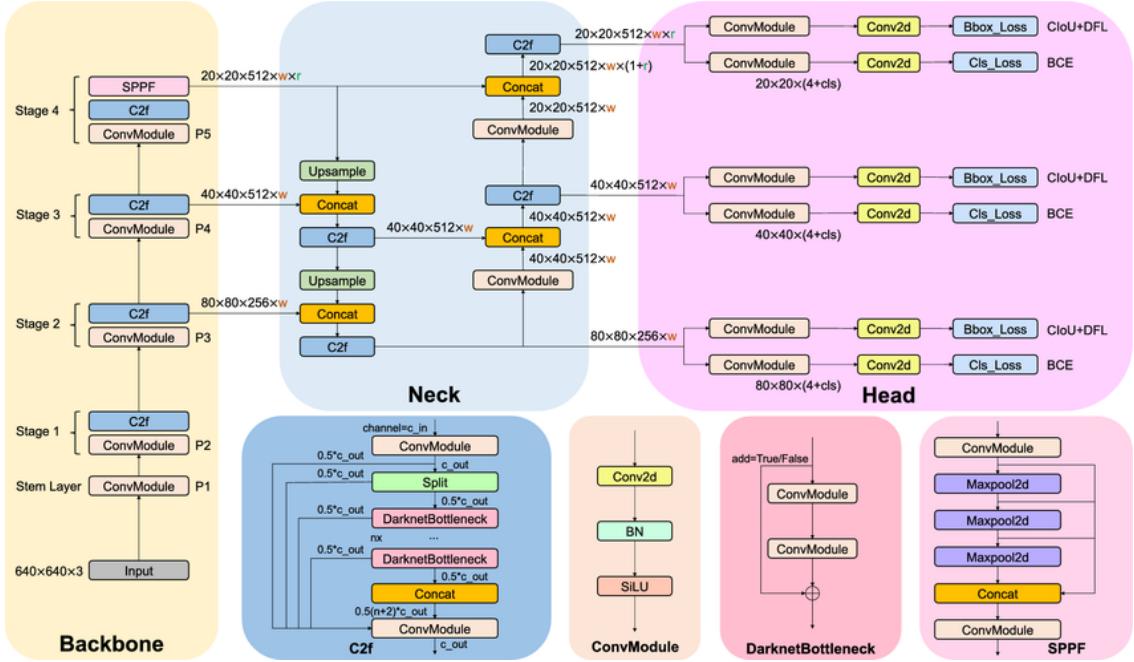


Figura 2: Detaljna reprezentacija arhitekture YOLOv8 modela

Za demonstraciju izražajnosti ove arhitekture dovoljno je koristiti unapred trenirani YOLOv8 model<sup>2</sup>. Kako bismo koristili pomenuti model u python-u, neophodno je importovati biblioteku *Ultralytics*.

Primena ovog modela nije ograničena samo na određivanje pozicije, već YOLOv8 omogućuje opcije poput segmentacije, detekcije, klasifikacije i praćenje objekata. Kako bismo model koristili za estimaciju položaja, neophodno je učitati model sa odgovarajućom opcijom na sledeći način:

```
model = YOLO('yolov8m-pose.pt')
```

Model je moguće i trenirati na skupu podataka *COCO8-pose*, kako bi se prikazalo napredovanje u treningu i određene statistike. U kodu broj epoha postavljen je na 100 i veličina slika na 640x640 piksela.

<sup>2</sup>Kod za učitavanje YOLOv8 modela

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
1/100	86	1.15	0.8659	0.2324	0.7755	1.543	7	640: 100%  ██████████  1/1 [60:15]
<00:00, 15.63s/it]	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)
-95: 100%  ██████████  1/1 [00:05<00:00, 5.24s/it]	all	4	14	0.987	0.786	0.897	0.744	R mAP50 mAP50
0.527								
Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
2/100	86	1.075	1.974	0.3615	0.7686	1.397	13	640: 100%  ██████████  1/1 [60:14]
<00:00, 14.67s/it]	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)
-95: 100%  ██████████  1/1 [00:04<00:00, 4.60s/it]	all	4	14	0.919	0.857	0.909	0.776	R mAP50 mAP50
0.526								
Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size
3/100	86	0.8633	1.459	0.3747	0.6395	1.028	17	640: 100%  ██████████  1/1 [60:13]
<00:00, 13.62s/it]	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)
-95: 100%  ██████████  1/1 [00:04<00:00, 4.43s/it]	all	4	14	0.922	0.844	0.906	0.774	R mAP50 mAP50
0.521								

Figura 3: Trening YOLOv8 modela

Što se vizualizacije rezultata tiče, rezultati modela predstavljeni su na narednim slikama (4a) i 4b) koje nisu iz skupa podataka za YOLOv8, gde se jasno vide ključne tačke detektovane na telu.



Figura 4: Rezultati YOLOv8 modela

## 2.2 OpenPose model

Još jedan model koji je dao značajne rezultate je OpenPose model<sup>3</sup>. Koristili smo unapred trenirani model koji je učitan pomoću OpenCV biblioteke. Model je dizajniran tako da prepoznaće ključne tačke na telu i povezuje ih kako bi formirao skelet na slici. Ovaj model je zasnovan na dubokim neuronskim mrežama i treniran je na velikim skupovima podataka.

**Arhitektura**<sup>4</sup> modela se sastoji od sledećih ključnih komponenti:

- **Ulazni sloj** - Model prihvata slike unapred definisane veličine. Na početku, slike prolaze kroz proces *normalizacije i preprocesiranja* kako bi se prilagodile arhitekturi mreže.
- **Skriveni slojevi** - Model sadrži više konvolutivnih slojeva koji omogućavaju ekstrakciju karakteristika iz slika. Ovi slojevi identifikuju obrasce kao što su ivice, uglovi i oblici na slikama, dok kasniji slojevi identifikuju složenije strukture, uključujući delove tela.
- **Izlazni sloj** - Izlaz modela su mape tačnosti (*confidence maps*) i polja afiniteta (*affinity fields*) za svaku od definisanih ključnih tačaka na telu. Svaka mapa prikazuje verovatnoću da se određena ključna tačka nalazi na specifičnom mestu na slici, čime se pruža informacija o povezivanju delova tela.
- **Postprocesiranje** - Nakon što model generiše mape tačnosti, koriste se metode kao što je *minMaxLoc* za lociranje pozicija ključnih tačaka.

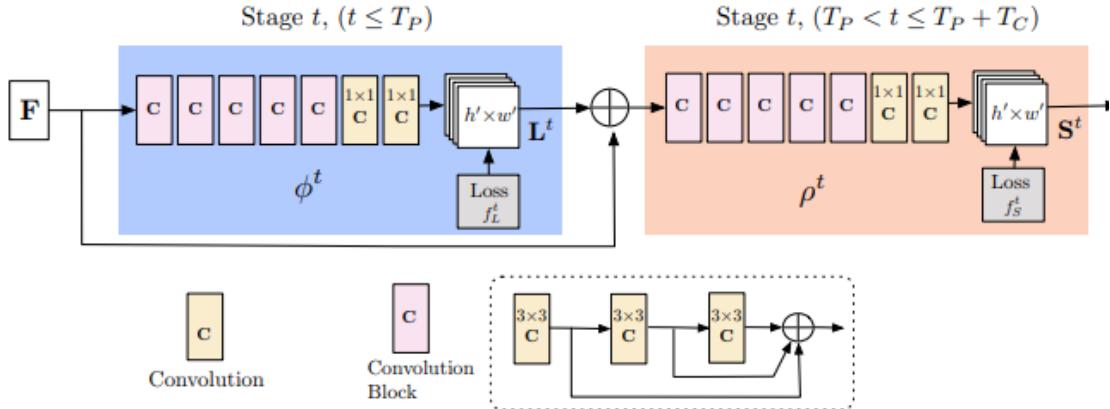


Figura 5: Detaljna reprezentacija arhitekture OpenPose modela

<sup>3</sup>Link ka implementaciji modela

<sup>4</sup>Arhitektura OpenPose modela

## Primena OpenPose modela

Učitane su težine iz datoteke *graphOpt.pb*. Ovaj model je prethodno treniran na skupu podataka koji sadrže slike sa označenim ključnim tačkama, poput nosa, ramena, laktova, kolena i drugih.

Kako bi slika mogla da bude obrađena kroz mrežu, izvršeno je skaliranje slike na dimenzije 368x368 piksela. Prag detekcije je postavljen na vrednost od 0.2, što znači da će samo tačke sa tačnošću većim od 20% biti uzete u obzir za prikazivanje.

Za svaku ključnu tačku na telu, kao što su *nos*, *vrat*, *ramena*, *laktovi* i *zglobovi*, definisana je odgovarajuća pozicija u nizu. Ove tačke su mapirane na indeksе koji odgovaraju pozicijama u izlazu mreže. Takođe, definisani su parovi tačaka koje će se povezivati linijama kako bi se formirao skeletni prikaz na slici.

Slika je zatim konvertovana u **blob format**, što je neophodan korak kako bi mogla da se obradi kroz neuronsku mrežu. Blob format podrazumeva skaliranje slike na odgovarajuće dimenzije i normalizaciju njenih vrednosti piksela, kako bi model mogao pravilno da interpretira podatke. Ovaj korak osigurava da je slika pravilno pripremljena za predikciju.

Model zatim, vrši *predikciju* nad ulaznom slikom, što rezultira mapama tačnosti za svaku ključnu tačku na telu.

Rezultati detekcije ključnih tačaka prikazuju se na narednim slikama(**6a** i **6b**) koje nisu iz skupa podataka za OpenPose model. Ovaj prikaz omogućava jasnu identifikaciju poze osobe na slici.

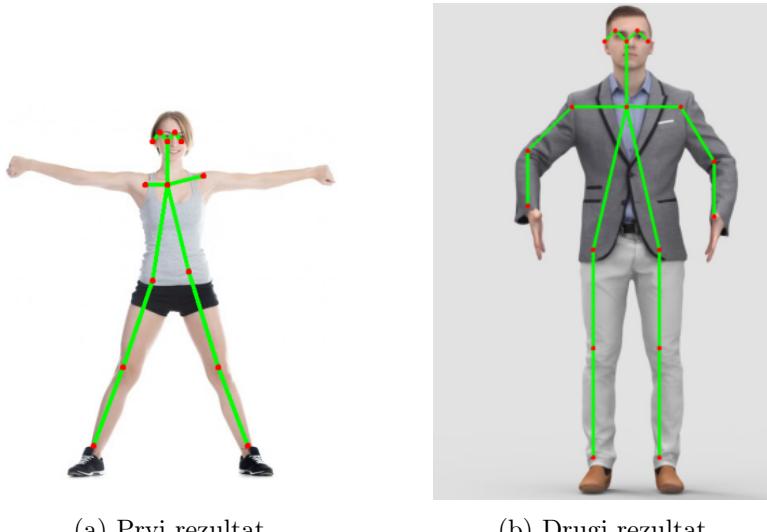


Figura 6: Rezultati OpenPose modela

## 2.3 Sopstveni model

Skup podataka [MPII Human Pose Database](#), na kojem je testiran model, sastoji se od slika i pratećeg *CSV* fajla. U svakom redu CSV fajla nalaze se naziv slike i odgovarajuće X i Y koordinate ključnih tačaka na telu. Ukoliko neka ključna tačka nije prisutna u kadru, njene koordinate dobijaju vrednost -1, čime se obeležava njen odsustvo.

Prvobitna ideja što se tiče funkcije gubitka bila je da se koristi funkcija MSELoss (*Mean Squared Error Loss*). Međutim, zbog specifičnosti skupa podataka, gde delovi tela koji nedostaju na slici imaju vrednost -1, bilo je neophodno ignorisati ove tačke prilikom izračunavanja gubitka. Stoga je razvijena omotač(wrapper) klasa koja implementira funkciju sličnu MSELoss, ali uz upotrebu maske koja omogućava da se ne uzimaju u obzir tačke sa vrednošću -1. Ovaj pristup omogućava preciznije merenje gubitka i poboljšava performanse modela.

U arhitekturi modela, ulaz u mrežu su slike dimenzija 64x64 piksela u RGB formatu. Na početku, slike prolaze kroz tri konvolucionala sloja, gde svaki sloj primenjuje filtere za izdvajanje različitih karakteristika sa slike. Prvi sloj koristi 16 filtera, drugi sloj koristi 32 filtera, dok treći sloj koristi 64 filtera, čime se povećava broj apstrahovanih karakteristika sa svakim slojem. Svaki konvolucionali sloj sadrži *ReLU aktivaciju i Batch Normalizaciju* kako bi se stabilisao proces treniranja i ubrzala konvergencija.

Svaki konvolucionali sloj je takođe praćen ReLU aktivacionom funkcijom, koja uvodi nelinearnost u mrežu. Nakon prolaska kroz tri konvolucionala sloja, slike se preoblikuju u vektor kako bi mogле biti obrađene u potpuno povezanim slojevima. Poslednji sloj generiše 28 izlaza, pri čemu svaki par izlaza predstavlja x i y koordinate ključnih tačaka na telu.

### 3 Zaključak

Tokom treninga, primećujemo značajno opadanje greške na trening podacima kroz epohe. Gubitak na test podacima ne pokazuje slične rezultate, tj. ne opada, što sugerise mogućnost preprilagodjavanja modela(overfitting). Međutim, to ne mora nužno biti slučaj, zato što čak i na slikama gde se pogrešno obeležavaju ključne tačke, one nisu proizvoljne, već predstavljaju ključne tačke nekog drugog objekta. Dakle, promenu funkcije greške dovodimo u vezi sa činjenicom da je skup podataka takav da čak iako na slici ima više objekata, samo je jedan anotiran.

Predstavljamo slike koje ilustruju relativno uspešne predikcije modela za detekciju ključnih tačaka.

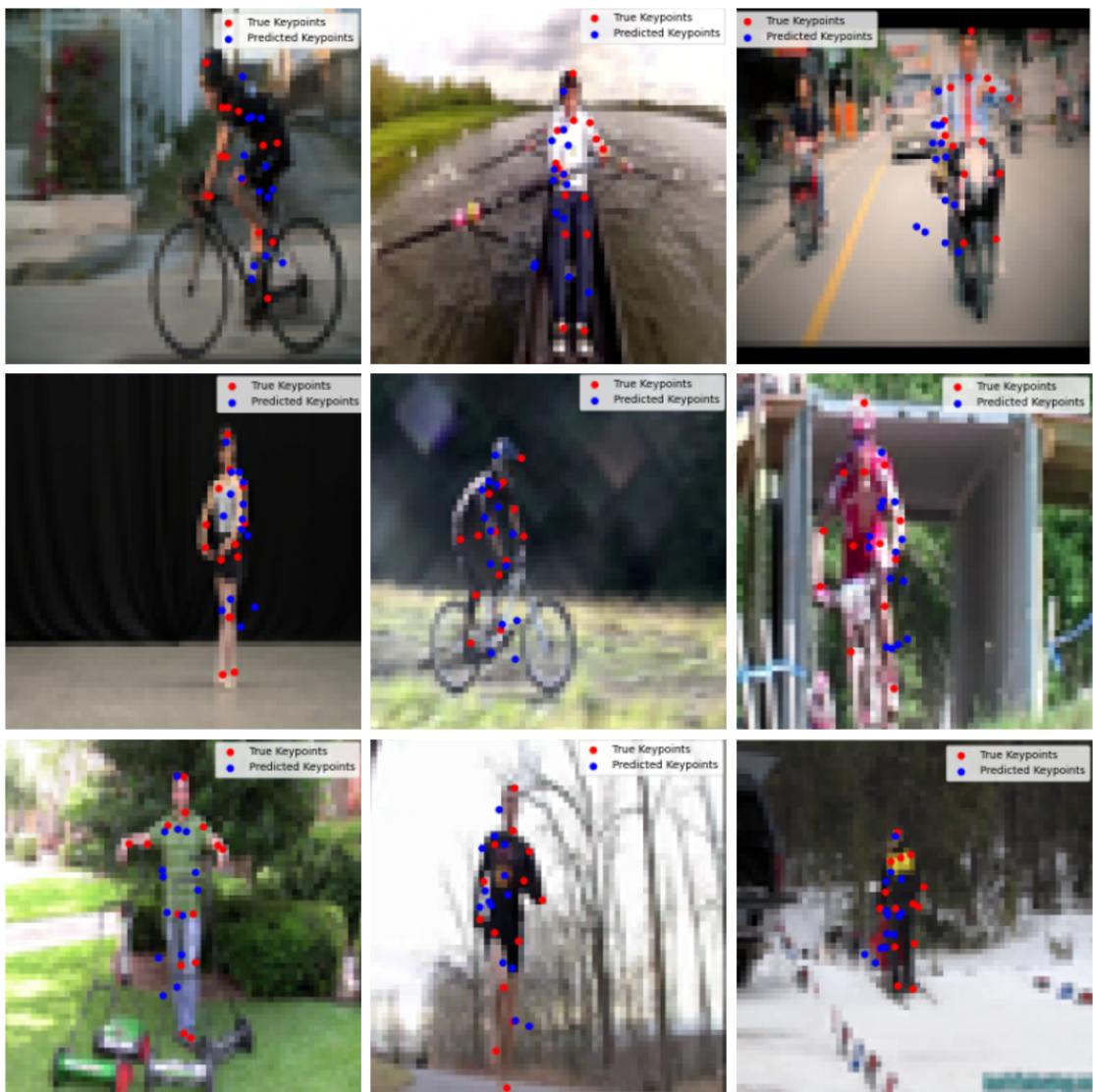


Figura 7: Uspešne predikcije modela kroz različite epohe

Naredne slike ukazuju na izazove sa kojima se model susreće u situacijama kada je prisutno više objekata na slikama.



Figura 8: Neuspešne predikcije modela kroz različite epohe

## 4 Literatura

- [1] YOLOv8 Documentation. [Ultralytics YOLOv8](#). Ultralytics, 2023.
- [2] Ultralytics. ["Pose Estimation — YOLOv8 Documentation."](#). Ultralytics, 2023.
- [3] OpenCV. ["OpenPose Sample Code."](#). GitHub, 2023
- [4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh. *OpenPose: Real-time Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2017.
- [5] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. ["MPII Human Pose Database."](#)
- [6] Materijali sa kursa *Računarska inteligencija*