

# Primjena YOLO modela za detekciju bolesti na biljkama

Matija Roginić, JMBAG: 0036530135

## Motivacija:

Motivacija za ovaj rad je svladavanje potrebnih tehnologija na datasetu slične tematike onom koji je u planu za diplomski rad. Naime, hobi mi je vinogradarstvo i u posljednjih nekoliko godina borimo se s problemom bolesti vinove loze, Zlatne žutice. S obzirom na to da se bolest vrlo brzo širi vinogradima, cilj je što prije otkriti zaražene trsove i ukloniti ih iz vinograda. Stoga će tema diplomskog rada biti detekcija te bolesti pomoću YOLO modela. Zbog toga sam odlučio rad na ovom predmetu napraviti na način da se bolje upoznam s tehnologijom i primjenom na sličnom datasetu.

## Uvod:

Cilj rada na ovom predmetu je upoznavanje s tehnologijama i modelima koji se koriste za računalni vid, prepoznavanje objekata i segmenata slike. Prvo što sam napravio je istraživanje koji modeli su trenutno najbolji za dani zadatak (State-of-the-art). U dogovoru s mentorom, sljedeći korak je bio odabir modela i primjena na problemu detekcije bolesti na lišću biljaka. Za model sam izabrao YOLOv9 što je trenutno najnovija i najbolja verzija od niza YOLO modela. Dataset koji sam koristio je dataset s bolestima biljaka koji je javno dostupan i kojeg sam preuzeo s Roboflowa. <sup>[1]</sup>

## Opis modela:

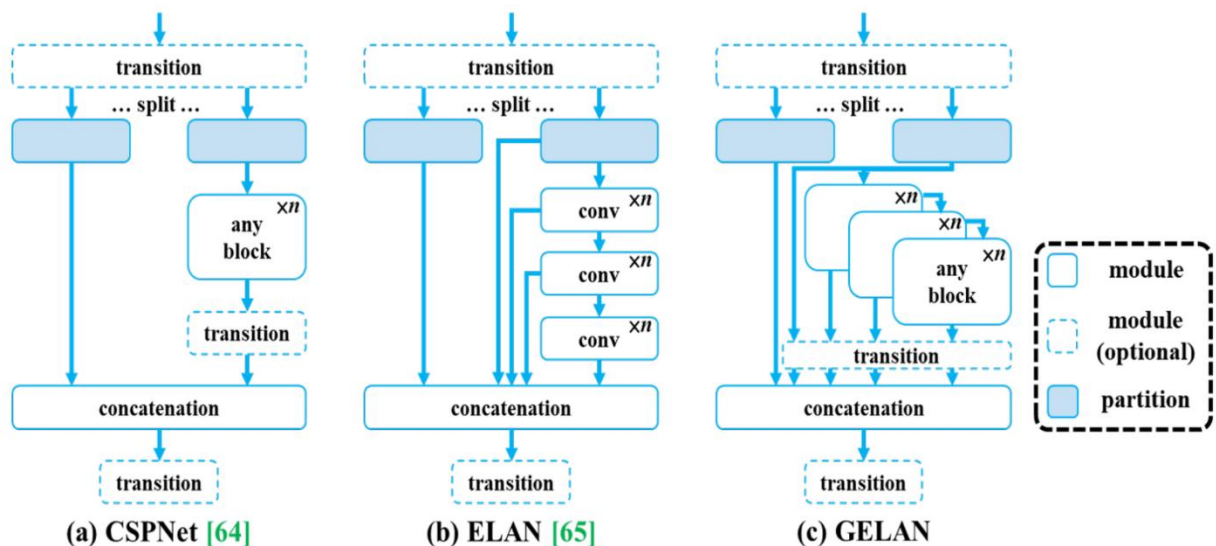
Model YOLOv9 je najnoviji član YOLO „obitelji“ modela s trenutno najboljim performansama od svih YOLO modela. YOLO predstavlja okvir za treniranje modela za bilo koji od 3 zadataka (detekciju objekata, segmentaciju dijelova slike, te klasifikaciju slike). Postoji 4 verzije YOLOv9 modela, a to su redom: S, M, C, E. Uzlazno su poredani po veličini i točnosti koju pružaju. Dakle, S je najmanji i najmanje točan od navedenih, ali je i najbrži. S druge strane, E je najveći i najtočniji, ali i najsporiji. Verzija koju ćemo koristiti za ovaj zadatak je C. Općenito, YOLO modeli su brzi modeli u usporedbi s drugim modelima koji se koriste u ovom području. Brzina se temelji na tome što oni svaku sliku pogledaju samo jednom i zatim se prebacuju na sljedeću. Iz toga dolazi i sam naziv modela: YOLO – You only look once. Model sam po sebi je dosta složen, s kombinacijom konvolucijskih, 2D konvolucijskih te pooling slojeva koji su posloženi u dva dijela (glavu i kralježnicu modela). Na izlazu modela nalazi se sloj Detect koji se sastoji od dvije obične konvolucije, zatim 2D konvolucije te dijela koji izračunava gubitak (box loss i clf loss). Novost u odnosu na prethodne verzije YOLO modela je način na koji mreža sprečava gubitak informacija tijekom transmisije podataka kroz slojeve mreže. <sup>[2]</sup>

$$I(X, X) \geq I(X, f_{\theta}(X)) \geq I(X, g_{\phi}(f_{\theta}(X)))$$

Slika 1: Prikaz gubitka informacije kroz slojeve – Bottleneck

Ono što YOLOv9 koristi zove se *Programmable Gradient Information (PGI)*. PGI omogućuje potpuno očuvanje ulazne informacije potrebne za izračun funkcije cilja, a zatim i pouzdan izračun gradijenata potrebnih za ažuriranje težina mreže.

Sljedeća novost u YOLOv9 modelu je *Generalized Efficient Layer Aggregation Network (GELAN)*. Obilježje te arhitekture je da maksimalno iskorištava parametre modela čime poboljšava točnost, a mreža zadržava računalnu učinkovitost i brzinu. [3]

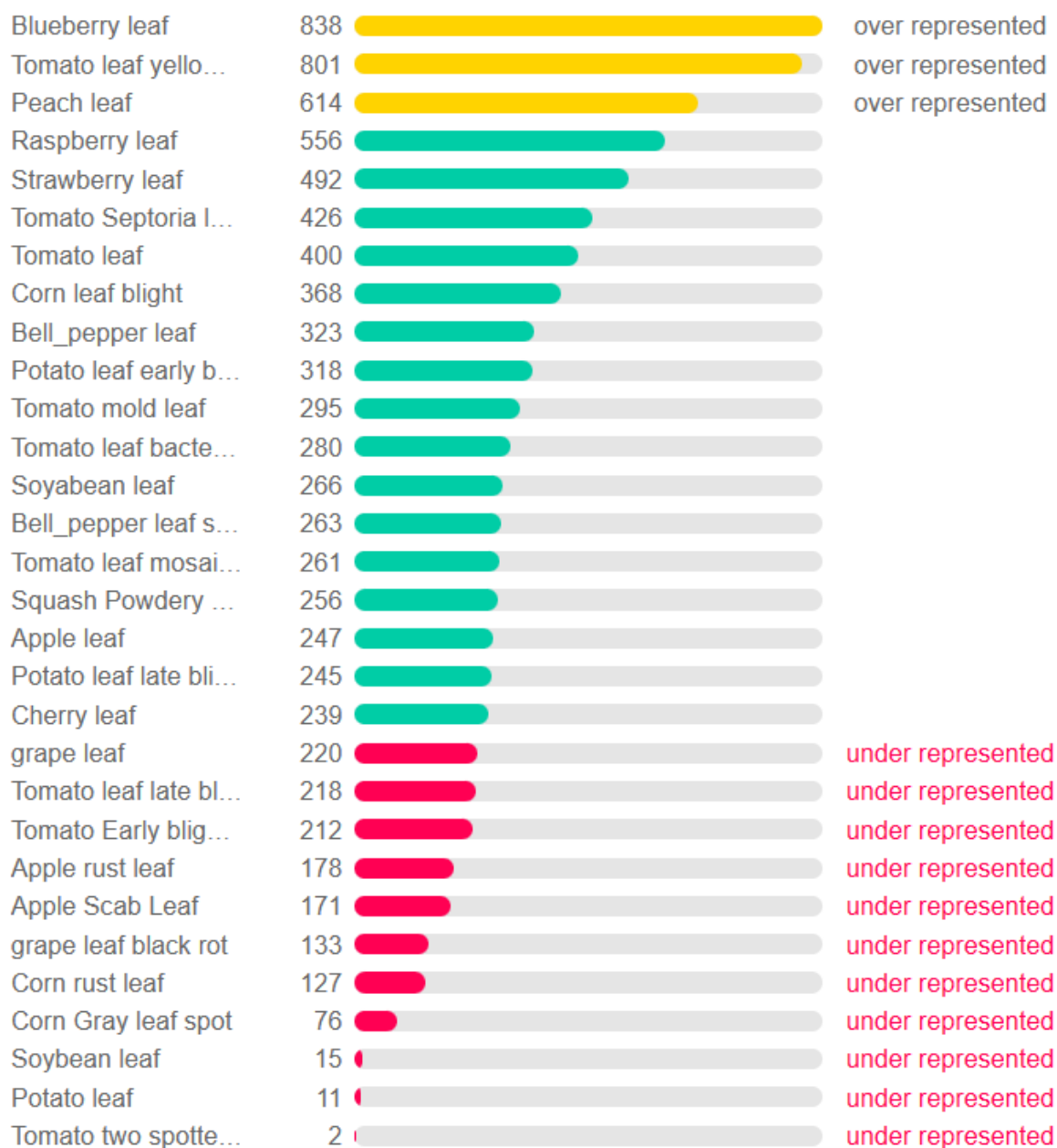


Slika 2: Arhitektura GELAN

## O datasetu:

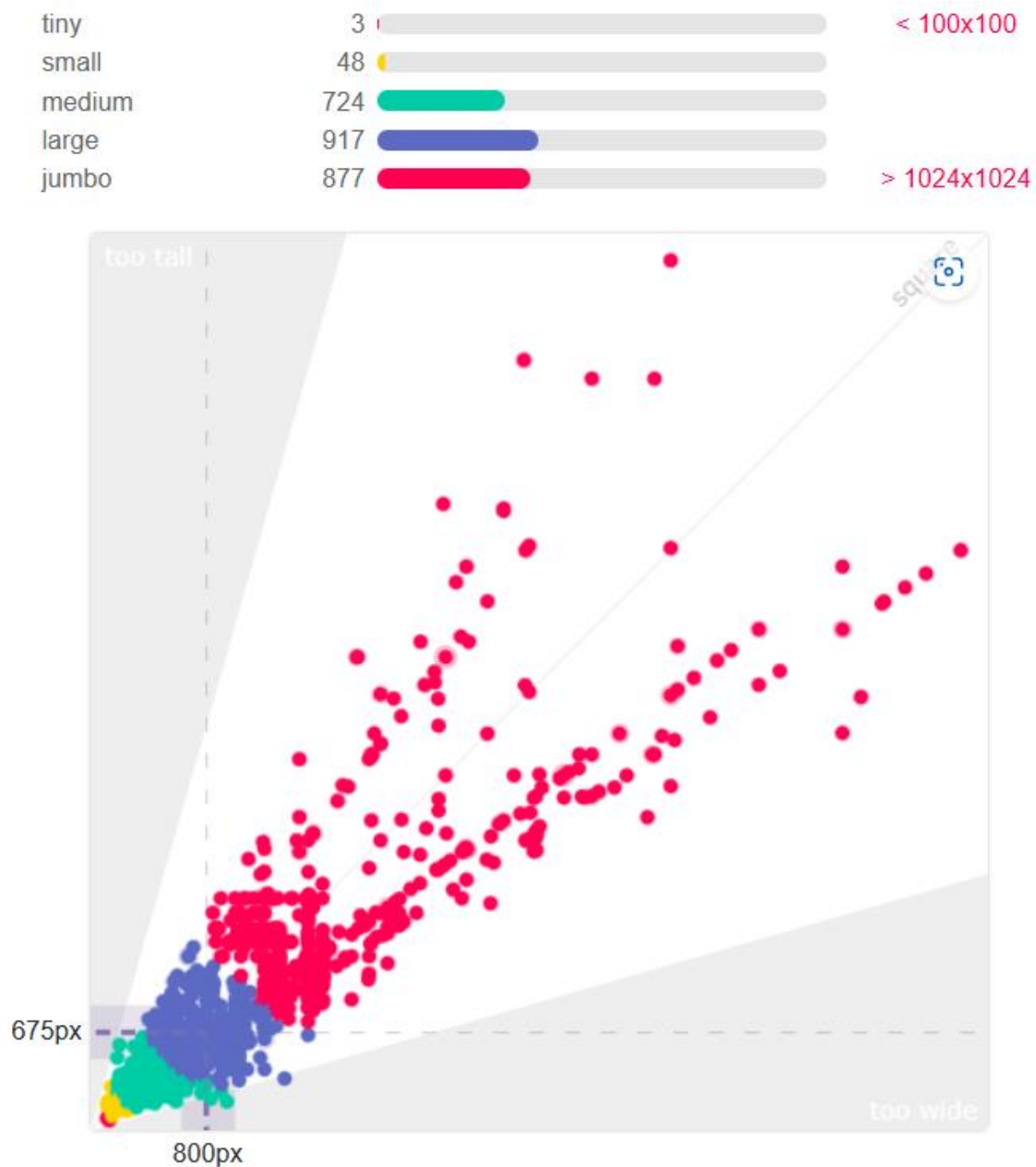
Dataset koji ćemo koristiti u ovom zadatku javno je dostupan na Roboflowu te se sastoji od 2569 slika s označenim objektima. Ukupni broj označenih objekata je 8851. Sam dataset je već pripremljen za YOLO model, te je podijeljen u skup za treniranje i testiranje u omjeru (0.9: 0.1). Svaki skup sadrži slike i pripadnu .txt datoteku za svaku sliku u kojoj se nalazi popis oznake svake bolesti biljke na slici, te njezin položaj na slici. Dataset također sadrži .yaml datoteku koja se sastoji od putanje do skupova za treniranje i testiranje te broja klasa i oznaka svake klase. Iz toga vidimo da se slike klasificiraju u ukupno 30 različitih klasa.

## Class Balance



Slika 3: Distribucija broja primjera po klasama

Iz gornje slike možemo uočiti potencijalan problem, a to je da dataset nije niti približno jednoliko distribuiran, te da je Roboflow 11 od 30 klasa označio kao da su podzastupljene. Taj problem se pogotovo može vidjeti u posljednje 4 klase.



Slika 4: distribucija primjera po veličini

Gornja slika nam prikazuje distribuciju primjera (slika) po veličini. To nam također može predstavljati problem jer sve slike na ulazu u model moraju biti jednake veličine, što znači da ćemo ih morati svesti na jednu univerzalnu veličinu čime može doći do pogoršanja kvaliteta slika i gubitka korisnih informacija.

## Treniranje modela:

	from	n	params	module	arguments
0		-1 1	1856	ultralytics.nn.modules.conv.Conv	[3, 64, 3, 2]
1		-1 1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
2		-1 1	212864	ultralytics.nn.modules.block.RepNCSPeLan4	[128, 256, 128, 64, 1]
3		-1 1	164352	ultralytics.nn.modules.block.ADown	[256, 256]
4		-1 1	847616	ultralytics.nn.modules.block.RepNCSPeLan4	[256, 512, 256, 128, 1]
5		-1 1	656384	ultralytics.nn.modules.block.ADown	[512, 512]
6		-1 1	2857472	ultralytics.nn.modules.block.RepNCSPeLan4	[512, 512, 512, 256, 1]
7		-1 1	656384	ultralytics.nn.modules.block.ADown	[512, 512]
8		-1 1	2857472	ultralytics.nn.modules.block.RepNCSPeLan4	[512, 512, 512, 256, 1]
9		-1 1	656896	ultralytics.nn.modules.block.SPpELan	[512, 512, 256]
10		-1 1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12		-1 1	3119616	ultralytics.nn.modules.block.RepNCSPeLan4	[1024, 512, 512, 256, 1]
13		-1 1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15		-1 1	912640	ultralytics.nn.modules.block.RepNCSPeLan4	[1024, 256, 256, 128, 1]
16		-1 1	164352	ultralytics.nn.modules.block.ADown	[256, 256]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18		-1 1	2988544	ultralytics.nn.modules.block.RepNCSPeLan4	[768, 512, 512, 256, 1]
19		-1 1	656384	ultralytics.nn.modules.block.ADown	[512, 512]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21		-1 1	3119616	ultralytics.nn.modules.block.RepNCSPeLan4	[1024, 512, 512, 256, 1]
22	[15, 18, 21]	1	5605930	ultralytics.nn.modules.head.Detect	[30, [256, 512, 512]]

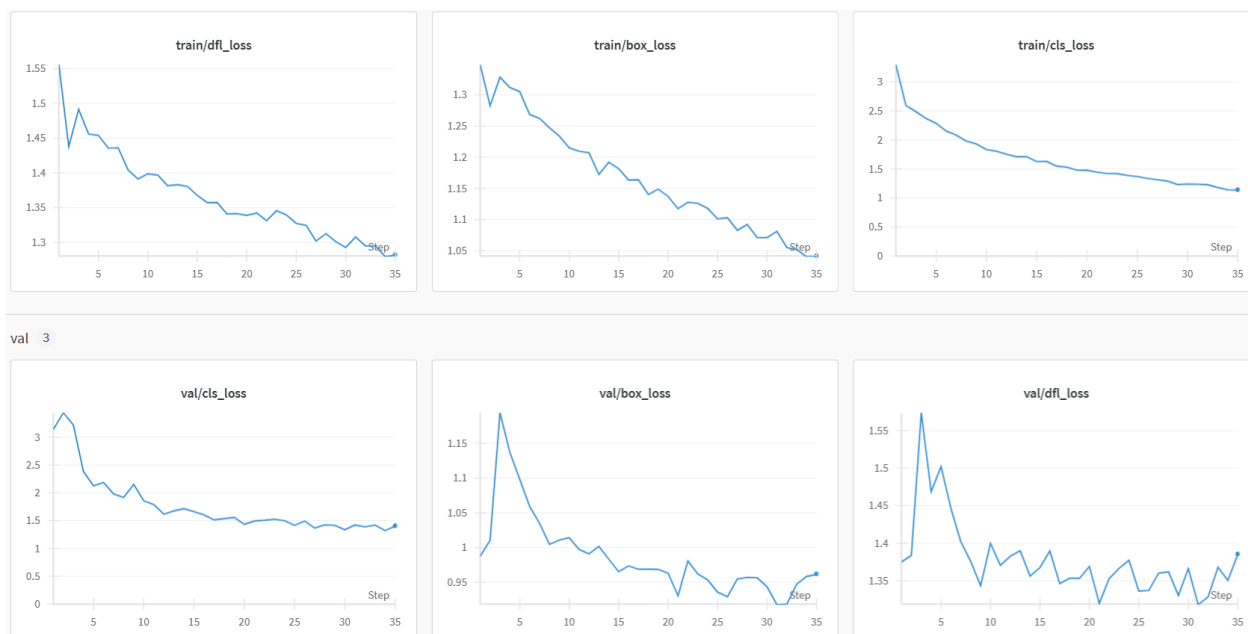
YOLOv9c summary: 618 layers, 25552362 parameters, 25552346 gradients, 103.8 GFLOPs

Slika 5: :Arhitektura modela

Sljedeći korak je treniranje modela. Slika iznad prikazuje samu arhitekturu modela, sloj po sloj, broj parametara, vrstu sloja i argumente koje sloj prima. Model je inicijaliziran na način da su promijenjeni neki hiperparametri: broj epoha je postavljen na 100, a patience na 5 što znači da ako u 5 epoha zaredom ne postoji poboljšanje u metrici validacije, trening se prekida. Uz to, argument `weight_decay` je postavljen na 0.001, a dropout na 0.2. Prvi od ta dva nam govori o tome koliko će djelovati L2 regularizacija, a drugi o tome da će se 20% neurona u mreži nasumično gasiti. Oba argumenta, isto kao i patience nam pomažu u tome da se model ne prenauči. Ostali hiperparametri nisu mijenjani, odnosno oni su onakvi kakvi su bili preddefinirani u modelu. Ono što sam promijenio je dimenzija slike, na 416×416 što je dimenzija na koju smo postavili veličinu slika u datasetu. Treniranje je provedeno preko platforme Kaggle, na njihovim grafičkim karticama zbog nepostojanja vlastite grafičke kartice na osobnom laptopu.

## Rezultati:

Nakon provedenog treninga, ono što možemo vidjeti da je pod utjecajem argumenta patience, trening zaustavljen nakon 35 epoha. Iz toga možemo zaključiti da je dobra odluka bila postaviti argument patience na više od 0 jer da smo trenirali punih 100 epoha, model bi vjerojatno postao dosta prenaučan. Sljedeće što ćemo prikazati su grafovi gubitaka tijekom epoha.



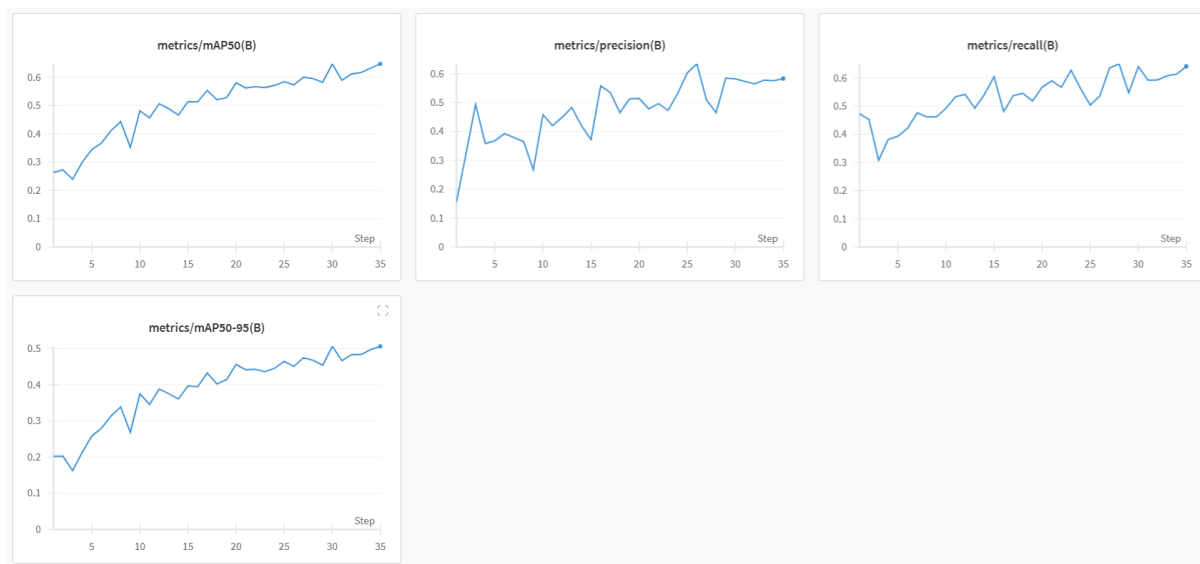
Slika 6: Grafovi pogrešaka na skupu za trening i validaciju

Prvo da objasnimo vrste funkcija gubitka čiji grafovi su gore prikazani. Box loss nam govori o tome kolika je greška u predviđanju granica objekata na slici, Cls loss prikazuje pogrešku u predviđanju klasa bolesti biljaka sa slike, dok Dfl loss mjeri grešku u tome je li objekt prisutan na slici ili nije. Iz grafova možemo vidjeti da se vrijednost funkcija gubitka smanjuje kontinuirano na skupu za treniranje, dok na skupu za validaciju vidimo da funkcije gubitaka padaju do nekog trenutka, a zatim su ili konstantne ili variraju. Konkretno brojeve ću prikazati u tablici ispod:

Tablica 1: Vrijednosti funkcija gubitaka

	Box loss		Cls loss		Dfl loss	
	Minimum	Krajnje	Minimum	Krajnje	Minimum	Krajnje
train	1.0420	1.0420	1.1430	1.1430	1.2810	1.2830
val	0.9196	0.9620	1.3300	1.4100	1.3190	1.3860

Iz tablice vidimo da se u sve tri funkcije gubitka na skupu za validaciju, minimum dostiže u nekom trenutku pa onda iznos raste do trenutka prekida treniranja.



Slika 7: Grafovi metrika modela

Recall (odziv) je mjera koja iznosi omjer između broja stvarno pozitivnih i zbroja stvarno pozitivnih i lažno negativnih. Precision (preciznost) mjera koja iznosi omjer između broja stvarno pozitivnih i zbroja stvarno pozitivnih i lažno pozitivnih. Iz grafova vidimo da odziv doseže vrijednost od 0.64, a preciznost vrijednost od 0.58 iako je u jednom dijelu treniranja bila iznad 0.6. Sljedeće metrike koje promatramo je mjera mAP50 i mAP50-95. mAP50 je mjera koja promatra omjer presjeka i unije između predviđenih i stvarnih okvira za svaki objekt na slici. U ovom slučaju taj omjer je fiksiran na 0.5, pa mjera govori o tome u kojem broju slučajeva je omjer presjeka i unije veći ili jednak od 0.5. U ovom slučaju, iznos te metrike doseže 0.64 do trenutka prekida testiranja. Posljednja metrika, mAP50-95 uzima prosjek mAP mjera s time da omjer presjeka i unije povećavamo od 0.5 do 0.95 s korakom od 0.05. U ovom slučaju, ona doseže 0.51 u trenutku prekida testiranja.

Za daljnju analizu možemo pogledati ostale grafove na sljedećem linku: [YOLOv8 Workspace – Weights & Biases \(wandb.ai\)](#)

### Zaključak:

Za kraj, moramo komentirati rezultate. Iz vrijednosti metrika vidimo da model postiže solidne rezultate, iako je daleko od savršenog. Uzrok tome možemo tražiti u samom datasetu, njegovom nebalansiranosti te razlici u veličini slika. Kasnijim postavljanjem veličine na  $416 \times 416$  sigurno gubimo na kvaliteti. Ta dva razloga nas sprečavaju u postizanju boljeg rezultata, no to je problem na kojeg ne možemo utjecati. Da bismo to potvrdili, provjerit ćemo kako se naš model ponaša ako provedemo validaciju samo na slikama onih klasa kojih u originalnom datasetu ima jako malo. Uzeli smo u obzir 3 klase slika kojih ima najmanje (Soybean leaf, Potato leaf, Tomato two spotted spider mites leaf) te smo u obzir uzeli slike iz tih klasa koje se nalaze u valid dijelu originalnog dataseta. Tih slika ima ukupno 8, te smo na njima pokrenuli validaciju modela. Rezultati koje smo dobili su sljedeći: Precision iznosi 0.4 (naspram 0.58 u cijelom valid datasetu), Recall 0.6 (naspram 0.64 u cijelom valid datasetu), mAP50 iznosi 0.497 (naspram 0.64 u

cijelom valid datasetu), a mAP50-95 0.433 (naspram 0.51 u cijelom valid datasetu). Ti rezultati su potvrdili sumnje da jedan dio uzroka nesavršenosti rezultata leži upravo u nebalansiranosti dataseta. Istu analizu mogli bismo provesti i za problem s veličinom slika u originalnom datasetu, no za pristup podacima koje slike su koje veličine moramo uzeti pretplatu na Roboflow jer nam besplatni dio nažalost ne pruža tu informaciju. Problemi koje smo ranije naveli mogli bi se izbjeći na način da se kod prikupljanja slika pazi na veličinu slika i balansiranoost dataseta. Dataset prikupljen na taj način bi sigurno doveo do boljih rezultata. To je definitivno nešto na što moram obratiti pažnju kod prikupljanja vlastitog dataseta kojeg ću koristiti pri izradi diplomskog rada.

## Literatura:

[1] Singh et. al., 17.04.2024., [PlantDoc Dataset \(roboflow.com\)](https://roboflow.com/dataset/plantdoc)

[2] Jocher et.al, 21.04.2024., YOLOv9: A Leap Forward in Object Detection Technology, [YOLOv9 - Ultralytics YOLOv8 Docs](https://ultralytics.com/yolov9)

[3] Ankan Ghosh, YOLOv9: Advancing the YOLO Legacy, 22.01.2024., [YOLOv9: Advancing the YOLO Legacy \(learnopencv.com\)](https://learnopencv.com/yolov9-advancing-the-yolo-legacy/)