

Τμήμα Μηχανικών Η/Υ & Πληροφορικής  
Πανεπιστήμιο Ιωαννίνων



# Εντοπισμός Σημείων Ενδιαφέροντος σε Εικόνες με Τεχνικές Βαθιάς Μάθησης

---

Αθανάσιος Καλυβιώτης  
Επιβλέπων Καθηγητής: Αριστείδης Λύκας

# Πίνακας Περιεχομένων

---

- 01 **Σημεία Ενδιαφέροντος**  
Τι είναι η ανίχνευση σημείων ενδιαφέροντος και που χρησιμοποιείται.
- 02 **Συνελικτικά Δίκτυα**  
Λίγα λόγια για τη δομή των συνελικτικών δικτύων.
- 03 **Ο αλγόριθμος YOLO**  
Σύντομη ιστορική αναδρομή εκδόσεων και βελτιώσεων.
- 04 **Πείραμα εντοπισμού λαβής**  
Περιγραφή προβλήματος εντοπισμού λαβής σε κούπες με τη χρήση του YOLOv8.
- 05 **Ενσωμάτωση σε Ρομποτικό Βραχίονα**  
Αυτόνομη ανασήκωση κούπας από τη λαβή με χρήση ρομποτικού βραχίονα.



# 01

## Σημεία Ενδιαφέροντος

---

Τι είναι σημεία ενδιαφέροντος;





# Τι είναι σημείο ενδιαφέροντος

Σημείο ενδιαφέροντος (keypoint)  
μπορεί να είναι οποιοδήποτε χαρακτηριστικό ενός  
αντικειμένου.

Για παράδειγμα σημεία ενδιαφέροντος σε ένα  
πρόβλημα εκτίμησης στάσης ανθρώπου είναι  
οι αρθρώσεις του.



# Εφαρμογές

---

Η ανίχνευση σημείων ενδιαφέροντος έχει πολλές εφαρμογές και μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων όπως:

- Αναγνώριση Προσώπου και Βιομετρία
- Συστήματα αναγνώρισης χειρονομιών
- Motion Tracking αντικειμένων

# Πως γίνεται η αναγνώριση των σημείων ενδιαφέροντος;

---

Η αυτόνομη ανίχνευση σημείων ενδιαφέροντος γίνεται με την χρήση μιας υποκατηγορίας Νευρωνικών Δικτύων που ονομάζονται Συνελικτικά Δίκτυα.

# 02

## Συνελικτικά Δίκτυα (CNNs)

---

Λίγα λόγια για τις αρχές των συνελικτικών δικτύων.

# Διαστρωμάτωση Συνελικτικών Δικτύων

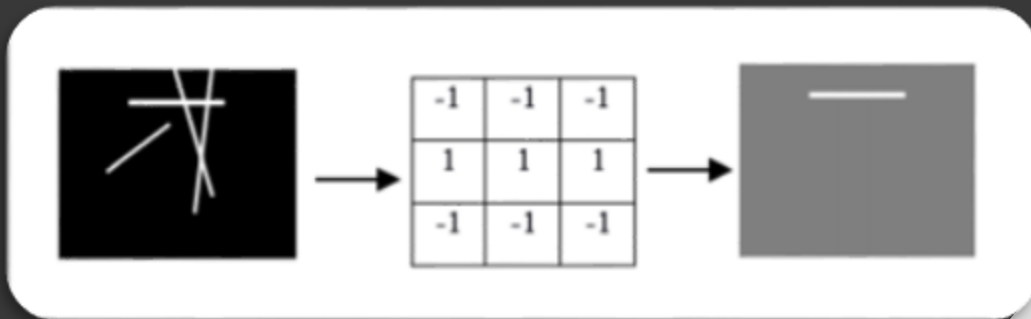
Ένα συνελκτικό δίκτυο αποτελείται από 4 βασικά στρώματα.

Συνέλιξη → Max Pooling → ReLU → Πλήρως συνδεδεμένο  
δίκτυο

Συνήθως αυτή η αλληλουχία **επαναλαμβάνεται** και στο τέλος ακολουθεί ένα **πλήρως συνδεδεμένο δίκτυο**, το οποίο συνδέει κάθε εξαγόμενο χαρακτηριστικό με κάθε κρυμμένο νευρώνα και ενισχύει τις υπολογιστικές δυνατότητες περιέχοντας **την πλειοψηφία των παραμέτρων** του δικτύου.



# Πυρήνας (Kernel)



- Ο πυρήνας ή φίλτρο είναι ένας πίνακας βαρών μικρότερης διάστασης από το στρώμα στο οποίο εφαρμόζεται, αλλά με ίδιο βάθος με αυτό της εισόδου (ή πίνακα χαρακτηριστικών).
- Τα φίλτρα σε ένα CNN αναλύουν τις αλλαγές στις τιμές των εικονοστοιχείων για να εντοπίσουν τα όρια των αντικειμένων, λαμβάνοντας υπόψη την τιμή φωτεινότητας και τα κανάλια χρώματος.
- Χρησιμοποιούνται πολλαπλά φίλτρα για την αναγνώριση διαφορετικών μοτίβων της εικόνας εισόδου μέσω της πράξης της συνέλιξης.

# Πράξη της συνέλιξης (Convolution)

- Η **συνέλιξη** στα συνελικτικά δίκτυα είναι η **ολίσθηση του φίλτρου** με την εικόνα εισόδου και ο υπολογισμός του **εσωτερικού γινομένου** κάθε τιμής του φίλτρου με τις αντίστοιχες της εισόδου.
- Το **βήμα (Stride)** καθορίζει τον αριθμό των εικονοστοιχείων με τα οποία το φίλτρο μετακινείται πάνω από την είσοδο, με μεγαλύτερα βήματα να μειώνουν περισσότερο τις διαστάσεις της εξόδου.
- Για την αποφυγή απώλειας πληροφορίας λόγω της μείωσης των διαστάσεων, χρησιμοποιείται η τεχνική της **πλήρωσης (Padding)**.

0	0	0	0	0	0	0	0
0	3	3	4	4	7	0	0
0	9	7	6	5	8	2	0
0	6	5	5	6	9	2	0
0	7	1	3	2	7	8	0
0	0	3	7	1	8	3	0
0	4	0	4	3	2	2	0
0	0	0	0	0	0	0	0

$6 \times 6 \rightarrow 8 \times 8$

\*

1	0	-1
1	0	-1
1	0	-1

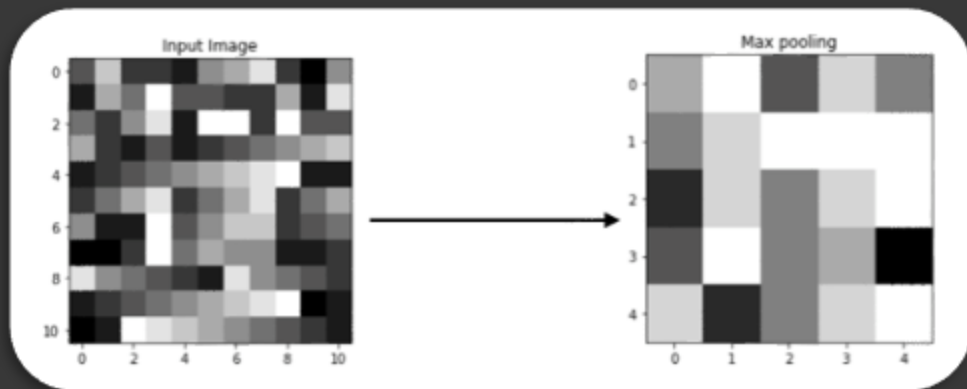
$3 \times 3$

=

-10	-13	1					
-9	3	0					

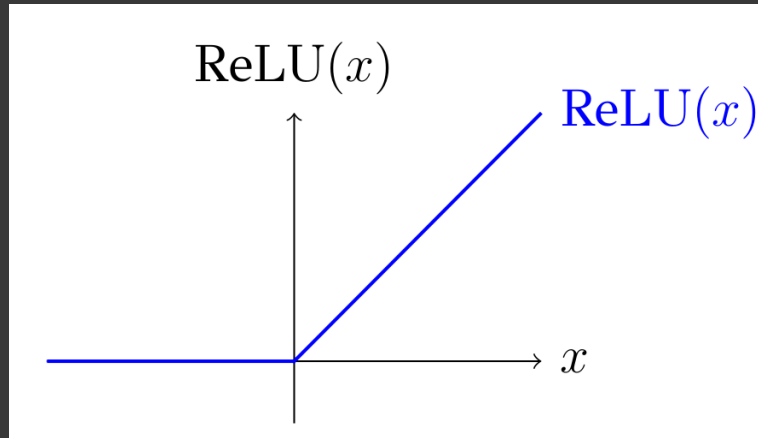
$6 \times 6$

# Max Pooling (Ομαδοποίηση)



- Τα επίπεδα συγκέντρωσης (Pooling Layers) **μειώνουν τις διαστάσεις** της εξόδου από το επίπεδο συνέλιξης και ομαδοποιούν την πληροφορία για πιο αποτελεσματική επεξεργασία.
- Το επίπεδο συγκέντρωσης **μειώνει τον αριθμό των παραμέτρων** και των υπολογισμών στο δίκτυο, βελτιώνοντας την αποδοτικότητα και αποτρέποντας το φαινόμενο της **υπερεκπαίδευσης**.

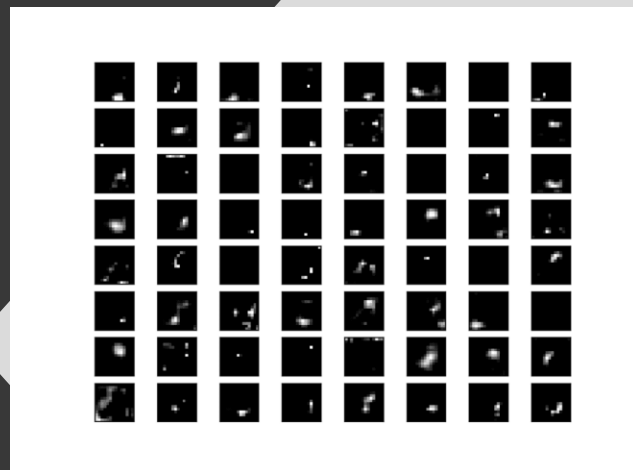
# Συνάρτηση Ενεργοποίησης ReLU



$$\text{ReLU}(x) = \max(0, x)$$

- Η **ReLU** και η παραλλαγή της **Leaky ReLU** βοηθούν στην αντιμετώπιση του προβλήματος της εξαφάνισης κλίσης και υποστηρίζουν την **αραιή ενεργοποίηση**, επιταχύνοντας τη μάθηση και μειώνοντας την πιθανότητα υπερεκπαίδευσης.
- Η γραμμικότητα και η **υπολογιστική απλότητα** της ReLU την καθιστούν ιδανική για σύγχρονες υλοποιήσεις, καθώς απαιτεί ελάχιστους υπολογισμούς.

# Χάρτης Χαρακτηριστικών



- Το αποτέλεσμα συνέλιξης της εισόδου με ένα φίλτρο δημιουργεί ένα πλέγμα τιμών που ονομάζεται Χάρτης Χαρακτηριστικών (Feature Map). Αυτοί αποτελούν την είσοδο προς το επόμενο επίπεδο του νευρωνικού δικτύου.
- Ενώ αρχικά εντοπίζουν ευδιάκριτα τα χαρακτηριστικά της εισόδου, διαχωρίζοντας το αντικείμενο από το παρασκήνιο, στα βαθύτερα στρώματα η σαφήνεια της αναπαράστασης μειώνεται.

# 03

## YOLOv8

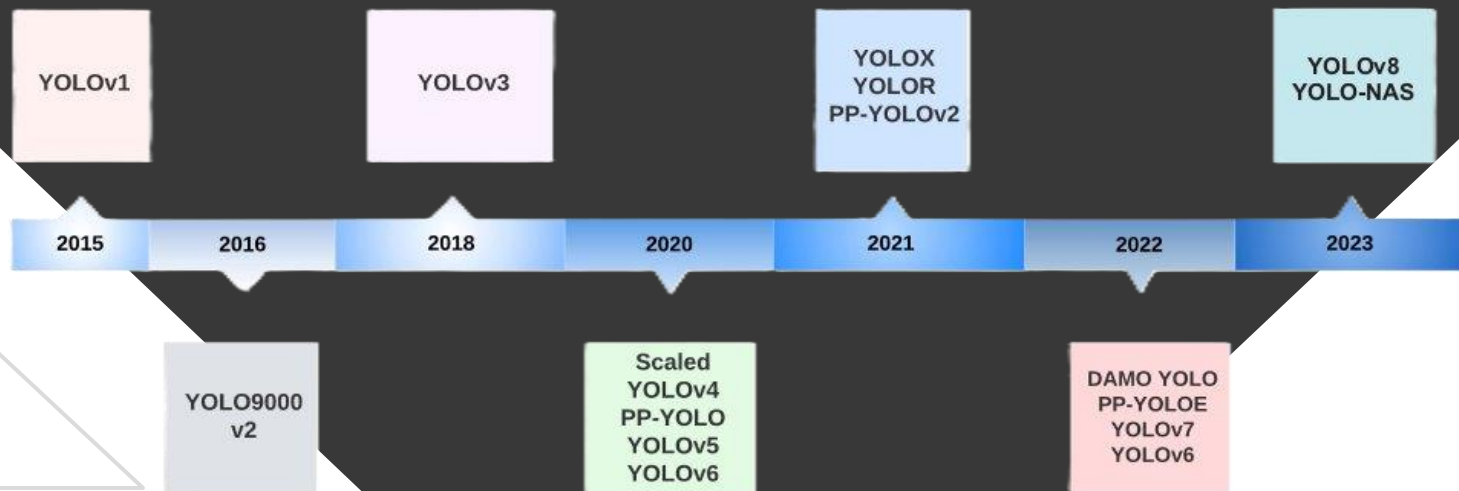
---

Το δίκτυο YOLO και οι  
δυνατότητές του.



# You Only Look Once

Το YOLO, που σημαίνει "You Only Look Once", είναι ένας ανοιχτού-κώδικα αλγόριθμος που επιτρέπει γρήγορη και αποδοτική αναγνώριση αντικειμένων σε πραγματικό χρόνο. Η ταχύτητα του αλγορίθμου οφείλεται στο γεγονός ότι η αναγνώριση πραγματοποιείται με ένα μόνο πέρασμα από το νευρωνικό δίκτυο, σε αντίθεση με άλλες πολυβηματικές μεθόδους. Από το 2015 μέχρι σήμερα έχουν γίνει αρκετές βελτιώσεις στον αλγόριθμο.





# Ταχύτητα

VS

# Ακρίβεια

---

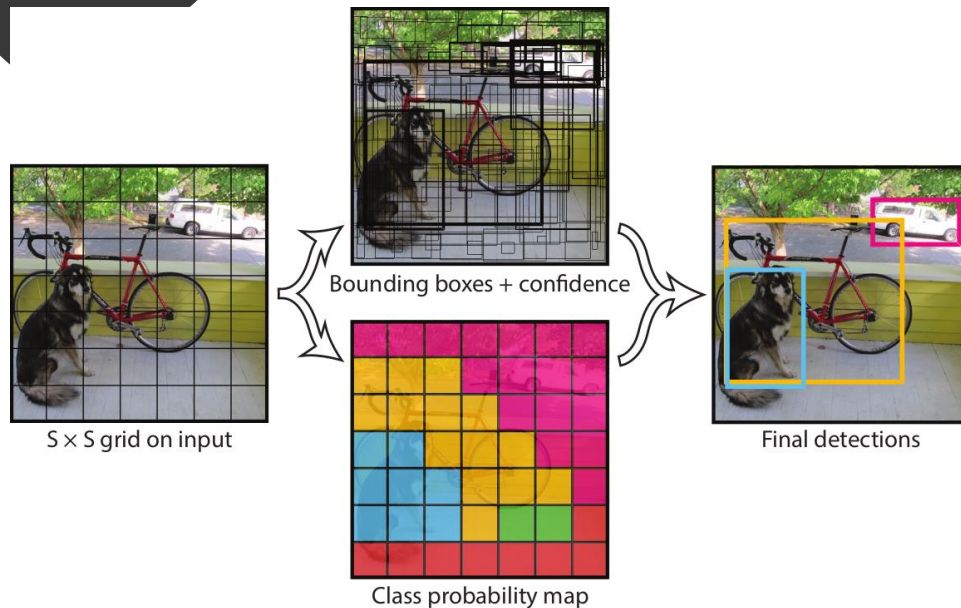
Σε όλες τις εκδόσεις δεν λαμβάνεται μόνο υπόψη η βελτίωση της ακρίβειας των εκτιμήσεων αλλά ταυτόχρονα η εγγύηση υψηλής ταχύτητας εκτέλεσης.

Το αποτέλεσμα είναι ένα **ισορροπημένο δίκτυο** με εντυπωσιακή ακρίβεια χωρίς μεγάλη θυσία της υπολογιστικής απλότητας.



# Ο Αλγόριθμος YOLO

- Η εικόνα χωρίζεται σε πλέγμα.
- Κάθε πλέγμα υπολογίζει περιγράμματα με τιμή **confidence** και πιθανή κατηγορία.
- Χρησιμοποιείται ο αλγόριθμος **Non-Maximum Supression** (NMS), που με τη χρήση της τιμής IoU γίνεται το φιλτράρισμα των bounding boxes και προκύπτουν οι τελικές προβλέψεις.
- Η τιμή **IoU** (Intesection over Union) είναι ενδεικτική του **βαθμού επικάλυψης** των περιγραμμάτων

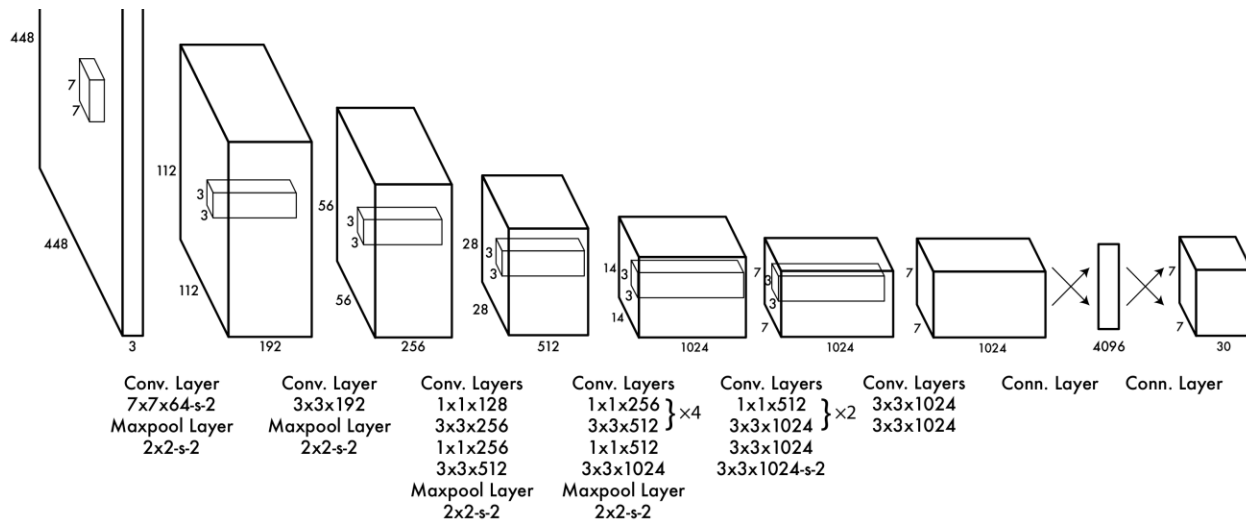


$$\text{Precision} = \frac{\text{Intersection}}{\text{Detected box}}$$

$$\text{Recall} = \frac{\text{Intersection}}{\text{Object}}$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

# Αρχιτεκτονική YOLO



- Η είσοδος είναι μία εικόνα 448 X 448 pixel και έπειτα από διαδοχικά συνελκτικά επίπεδα καταλήγουμε σε μία έξοδο 7 X 7 X 30, καθώς έχουμε 7 X 7 κελιά πλέγματος όπου το καθένα προβλέπει 2 περιγράμματα.



# Περιορισμοί YOLO

- Αδυναμία στον εντοπισμό λεπτομερών χαρακτηριστικών αντικειμένων.
- Αδυναμία ανίχνευσης πολλαπλών αντικειμένων ίδιας κατηγορίας από το ίδιο κελί.
- Δυσκολία εκτίμησης περιγράμματος σε αντικείμενα ασυνήθιστων διαστάσεων.



## Anchor Boxes

Προκαθορισμένα περιγράμματα (5)  
για κάθε κελί του πλέγματος



## Batch Normalization

Χρήση κανονικοποίησης παρτίδας για  
αποφυγή υπερεκπαίδευσης και  
καλύτερη γενίκευση

---

# YOLO v2

---



## Fine grained features

Στην αρχιτεκτονική το YOLO  
v2 ενσωματώνει ένα passthrough  
επίπεδο που συνδυάζει χάρτες  
χαρακτηριστικών στοιβάζοντας γειτονικά  
χαρακτηριστικά



## Darknet-19

Χρήση του Darknet-19 ως  
backbone και κατάργηση Fully  
Connected Επιπέδων

# YOLO v3

---

## Objectness

Υπολογισμός νέας τιμής objectness  
με χρήση logistic regression

## Strided Convolutions

Αντικατάσταση Max-Pooling  
με Strided Convolutions

## Binary Cross-Entropy

Αντικατάσταση συνάρτησης ενεργοποίησης  
softmax με binary cross-entropy

# YOLO v5

---



## Pytorch

Υλοποίηση χρησιμοποιώντας την βιβλιοθήκη Pytorch αντί για το Darknet



## CSPDarknet53

Χρήση του μεγαλύτερου και ταχύτερου backbone CSPDarknet53, εξέλιξη του Darknet-19



## Εκπαίδευση Anchors

Τα anchors εκπαιδεύονται για να προσαρμοστούν καλύτερα στα δεδομένα χρησιμοποιώντας γενετικούς αλγορίθμους



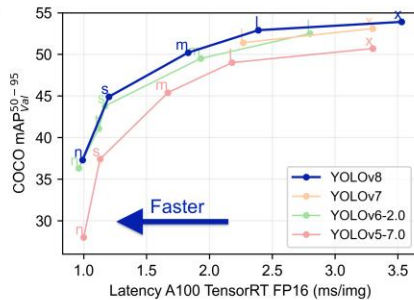
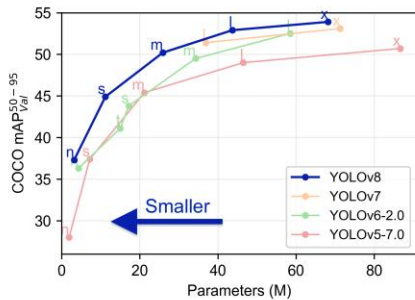
## Παραλλαγές Μεγέθους

Το YOLO v5 δίνει 5 επιλογές μεγέθους δικτύου (n, s, m, l, x)



# YOLO v8

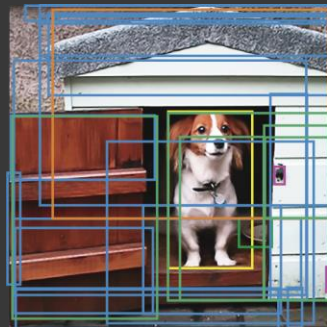
Το **YOLOv8**, η πιο πρόσφατη έκδοση από την ultralytics, αποτελεί μια δημοφιλή λύση για προβλήματα αναγνώρισης αντικειμένων, ανίχνευσης σημείων, τμηματοποίησης, παρακολούθησης και κατηγοριοποίησης. **Βασισμένο στο YOLOv5**, παρουσιάζει **σημαντική βελτίωση** στην ακρίβεια και την ταχύτητα σε σχέση με τους προκατόχους του.



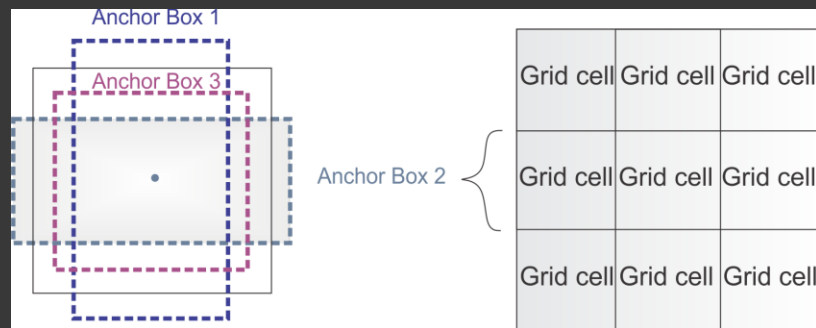
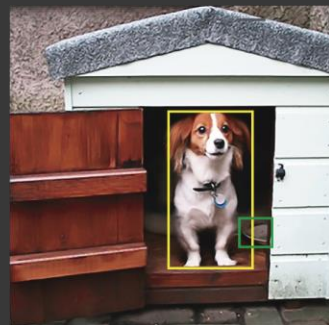


# Anchor-free Ανίχνευση

Η σημαντικότερη αλλαγή στην έκδοση 8 του YOLO είναι η απουσία **anchor boxes** για την αναγνώριση αντικειμένων, γνωστή ως **anchor-free detection**. Το μοντέλο προβλέπει απευθείας το κέντρο του αντικειμένου χωρίς να εξαρτάται από anchor boxes, μειώνοντας τον αριθμό των προβλέψεων και επιτρέποντας ταχύτερη εκτέλεση του αλγόριθμου **Non-Maximum Suppression (NMS)**.



➡  
**NMS**





# Scalability Δικτύου

Το YOLO v8 προσφέρει διάφορα προ-εκπαιδευμένα μοντέλα για pose-estimation. Κάθε εκπαιδευμένο δίκτυο έχει διαφορετικό μέγεθος και πλήθος παραμέτρων, παρέχοντας ευελιξία στη σχέση ταχύτητας-ακρίβειας.

Model	size (pixels)	mAP <sub>pose</sub> 50-95	mAP <sub>pose</sub> 50	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-pose	640	50.4	80.1	131.8	1.18	3.3	9.2
YOLOv8s-pose	640	60.0	86.2	233.2	1.42	11.6	30.2
YOLOv8m-pose	640	65.0	88.8	456.3	2.00	26.4	81.0
YOLOv8l-pose	640	67.6	90.0	784.5	2.59	44.4	168.6
YOLOv8x-pose	640	69.2	90.2	1607.1	3.73	69.4	263.2
YOLOv8x-pose-p6	1280	71.6	91.2	4088.7	10.04	99.1	1066.4

# 'Αλλες Βελτιώσεις

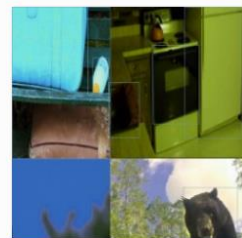
- Μία καινοτομία της νέας έκδοσης του YOLO είναι η χρήση του **μοσσαϊκού** κατά την επαύξηση του συνόλου δεδομένων, που δημιουργεί μοσσαϊκά από **τυχαίες αποκοπές εικόνων** κατά την εκπαίδευση. Αυτό βελτιώνει την αποτελεσματικότητα της εκπαίδευσης συνδυάζοντας εικόνες με διαφορετικά αντικείμενα και κλάσεις. Η τεχνική διακόπτεται στις τελευταίες 10 εποχές της εκπαίδευσης για καλύτερα αποτελέσματα.
- Επίσης, η δυνατότητα **τμηματοποίησης εικόνας** με το νέο μοντέλο "**yolov8-seg**" και οι δομικές αλλαγές αποτελούν σημαντικές βελτιώσεις.



aug\_-319215602\_0\_-238783579.jpg



aug\_-1271888501\_0\_-749611674.jpg



aug\_1462167959\_0\_-1659206634.jpg



aug\_1474493600\_0\_-45389312.jpg



aug\_1715045541\_0\_603913529.jpg



aug\_1779424844\_0\_-589696888.jpg

# Αρχιτεκτονική

## Backbone

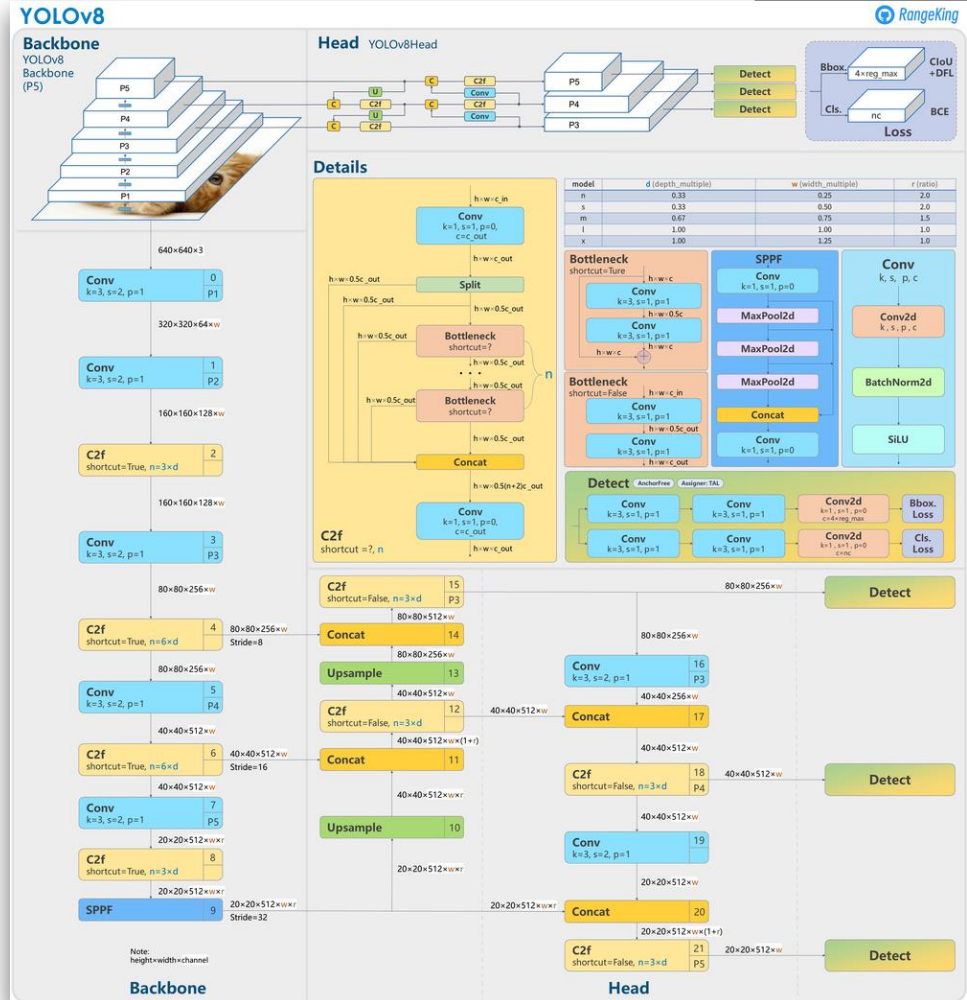
Εξαγωγή χαρακτηριστικών,  
βάση όλου του δικτύου.

## Neck

Συνδυασμός χαρακτηριστικών από  
πολλά επίπεδα αναλύσεων.

## Head

Εξαγωγή τελικού  
αποτελέσματος.



# 04

## Πείραμα εντοπισμού λαβής κούπας

---

Περιγραφή εφαρμογής του YOLOv8 για τον εντοπισμό λαβής σε κούπες.

# Μοντελοποίηση

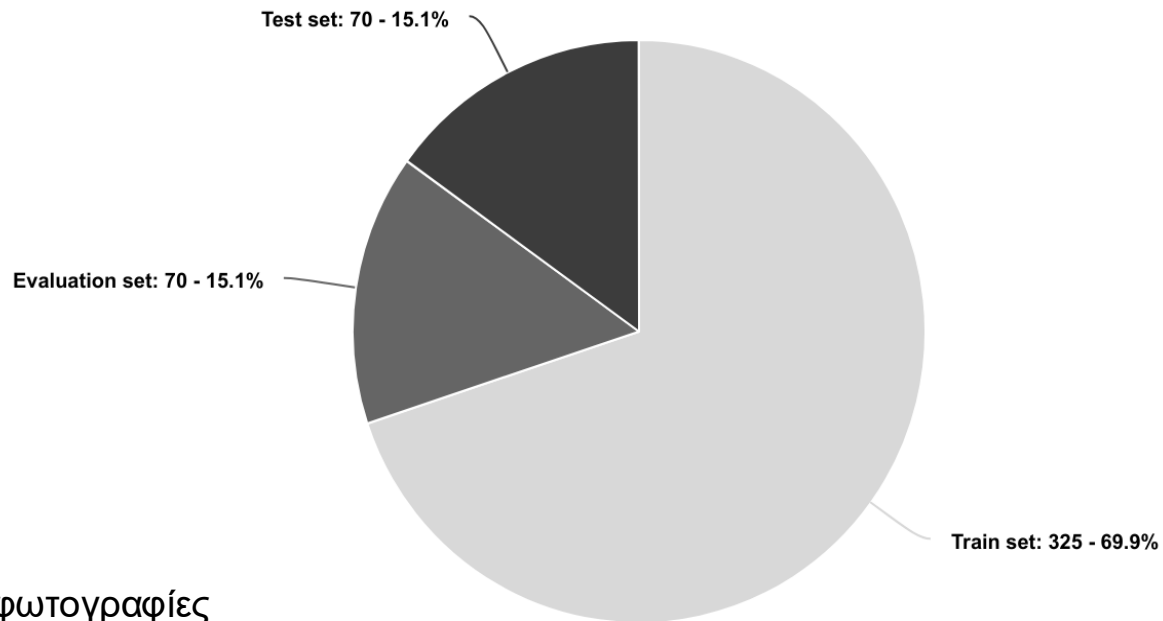
Η λαβή μιας κούπας έχει χαρακτηριστικό **ελλειψοειδές σχήμα** και βρίσκεται στην **πλαιινή πλευρά**. Σε αυτό το πείραμα, η θέση και το σχήμα της λαβής προσδιορίζονται με **τρία σημεία**: ένα στο πάνω σημείο σύνδεσης με την κούπα, ένα στο κάτω σημείο, και ένα στο μέσο, το πιο απομακρυσμένο από το κέντρο της κούπας. Αυτά τα σημεία καθορίζουν τον **προσανατολισμό, τα άκρα και το γενικό σχήμα** της λαβής, διευκολύνοντας την αναγνώριση χωρίς να αυξάνεται η πολυπλοκότητα της μάθησης.



# Σύνολο Δεδομένων



# Σύνολο Δεδομένων



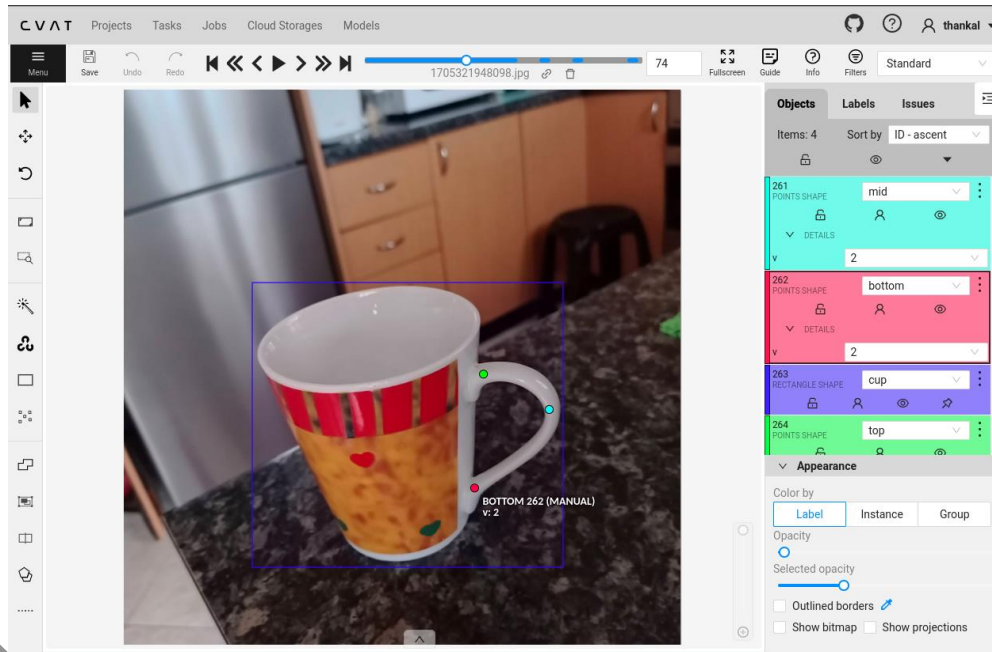
Συνολικά 465 φωτογραφίες

Train set Evaluation set Test set



# Επισήμανση Δεδομένων

Η επισήμανση δεδομένων είναι αναγκαία για την εκπαίδευση του μοντέλου. Εργαλεία αυτοματισμού, όπως το **CVAT** (Computer Vision Annotation Tool), υποστηρίζουν αυτή τη διαδικασία, παρέχοντας ένα εύχρηστο περιβάλλον για αποδοτική διαχείριση μεγάλων όγκων δεδομένων. Για κάθε φωτογραφία καθορίζουμε χειροκίνητα το **περίγραμμα** της κούπας και τα **3 keypoints** με την κατάλληλη τιμή **visibility**.



```
<image id="0" name="1702935368059.jpg" width="2992" height="2992">
  <box label="cup" source="manual" occluded="0" xtl="1869.20"
    ytl="517.98" xbr="2377.52" ybr="1093.86" z_order="0">
  </box>
  <points label="top" source="manual" occluded="0" points="
    2266.97,804.50" z_order="0">
    <attribute name="v">2</attribute>
  </points>
  <points label="bottom" source="manual" occluded="0" points="
    2241.18,948.94" z_order="0">
    <attribute name="v">2</attribute>
  </points>
  <points label="mid" source="manual" occluded="0" points="
    2349.51,897.35" z_order="0">
    <attribute name="v">2</attribute>
  </points>
</image>
<image id="1" name="1702935368070.jpg" width="2992" height="2992">
  <box label="cup" source="manual" occluded="0" xtl="1720.15"
    ytl="489.82" xbr="2256.65" ybr="1052.11" z_order="0">
  </box>
  <points label="mid" source="manual" occluded="0" points="
    2132.84,562.04" z_order="0">
    <attribute name="v">2</attribute>
  </points>
  <points label="bottom" source="manual" occluded="0" points="
    2158.64,727.12" z_order="0">
    <attribute name="v">2</attribute>
  </points>
  <points label="top" source="manual" occluded="0" points="
    2089.04,629.11" z_order="0">
    <attribute name="v">2</attribute>
  </points>
</image>
```



# Keypoint Visibility Flag

---

0

Σημείο εκτός ορίων εικόνας. Σε αυτή τη περίπτωση οι συντεταγμένες του σημείου δεν λαμβάνονται υπόψη.

1

Σημείο κρυμμένο αλλά μέσα στην εικόνα.

2

Ορατό σημείο.

# Εκπαίδευση

Το YOLOv8 μας δίνει την δυνατότητα να ρυθμίσουμε με ακρίβεια τις υπερπαραμέτρους της εκπαίδευσης ώστε να βελτιστοποιήσουμε το μοντέλο ανάλογα με το dataset και τις ανάγκες μας. Τέτοιες υπερπαραμέτροι είναι:

- epochs
- Batch\_size
- patience
- imgsz
- lr0
- optimizer
- label\_smoothing

- box
- cls
- pose
- kobj
- dfl

- degrees
- flip\_lr
- scale
- shear
- perspective
- translate

CLI εντολή:

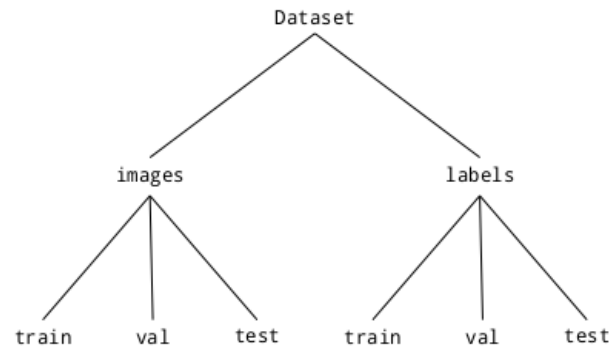
```
yolo task=pose mode=train model=yolov8s-pose.pt data='config.yaml' epochs=200 batch=4 patience=20 imgsz=640  
...
```

# Εκπαίδευση

Για την εκπαίδευση του μοντέλου αναγνώρισης λαβών, χρησιμοποιήθηκαν εικόνες από κούπες τραβηγμένες από διαφορετικές γωνίες αλλά **πάντα σε ορθό κατακόρυφο προσανατολισμό**. Η ανίχνευση της σωστής κατεύθυνσης της κούπας (ορθά ή ανάποδα) είναι ένα δύσκολο πρόβλημα και έτσι στο συγκεκριμένο πείραμα, θεωρείται δεδομένο ότι η κούπα είναι σε όρθια θέση.

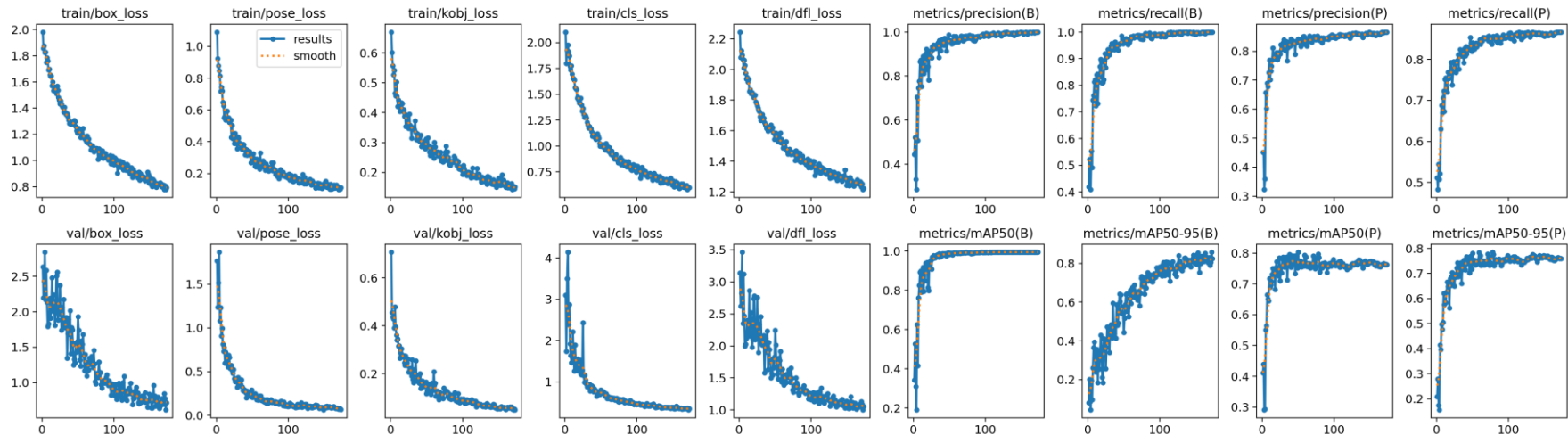
Η εκπαίδευση του μοντέλου χρησιμοποιεί **μεταφορά μάθησης** από το **yolov8s-pose**, το οποίο είναι εκπαιδευμένο στην εκτίμηση στάσης ανθρώπινου σώματος. Αντί για 17 keypoints, το νέο μοντέλο εκτιμά μόνο 3.

	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.628	0.495	0.521	0.24	0.531	0.441	0.418	0.248
Epoch 3/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.26G	1.696	0.9561	1.363	1.64	1.924	2	640: 100%		117/117	[00:18<00:0
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.177	0.49	0.149	0.0445	0.159	0.447	0.131	0.0905
Epoch 4/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.26G	1.734	0.8377	1.384	1.626	1.931	2	640: 100%		117/117	[00:18<00:0
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.543	0.353	0.375	0.117	0.48	0.458	0.395	0.245
Epoch 5/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.26G	1.658	0.7484	1.324	1.583	1.871	1	640: 100%		117/117	[00:18<00:0
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.742	0.766	0.8	0.398	0.635	0.656	0.554	0.397
Epoch 6/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.26G	1.635	0.7314	1.256	1.494	1.877	2	640: 100%		117/117	[00:18<00:0
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.772	0.742	0.796	0.382	0.618	0.633	0.568	0.432
Epoch 7/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.27G	1.622	0.6935	1.233	1.544	1.874	3	640: 100%		117/117	[00:18<00:0
	Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Pose(P)	R	mAP50	mAP50-95
	all	465	465	0.757	0.695	0.808	0.414	0.549	0.624	0.546	0.412
Epoch 8/200	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances	Size			
	1.26G	1.586	0.6685	1.249	1.483	1.836	4	640: 100%		117/117	[00:18<00:0



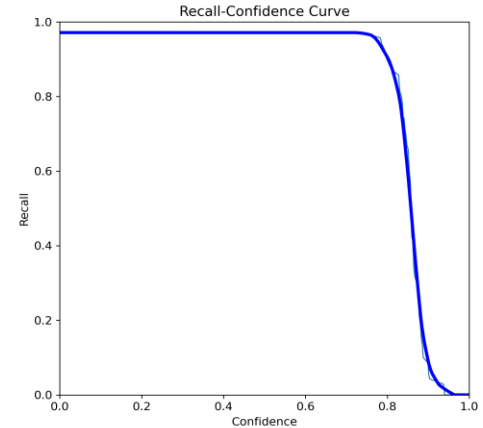
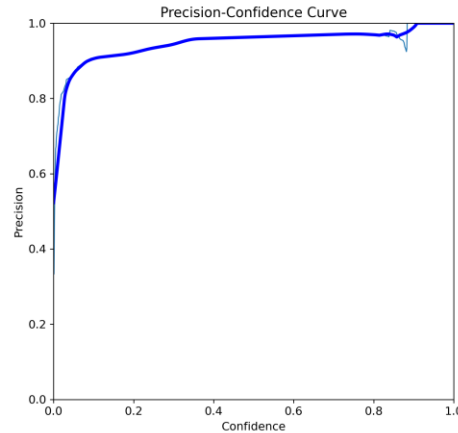
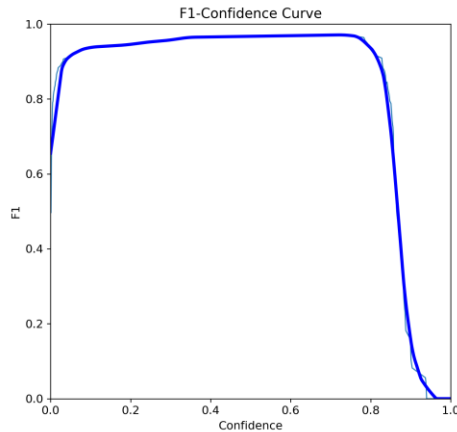
# Εκπαίδευση

Σε αυτό το στιγμιότυπο, παρατηρούμε την έξοδο στο τερματικό κατά τη διάρκεια της εκπαίδευσης, όπου μπορούμε να παρακολουθούμε την πρόοδο του μοντέλου και τις μετρικές επίδοσης. Το μοντέλο εκπαιδεύτηκε με **batch size=4** για 173 εποχές, 27 λιγότερες από τις καθορισμένες 200. Η εκπαίδευση τερματίστηκε πρόωρα λόγω της παραμέτρου `patience=20`, που αποφεύγει την υπερεκπαίδευση τερματίζοντας την εκπαίδευση αν δεν υπάρχει βελτίωση για `k` συνεχόμενες εποχές. Τα βέλτιστα βάρη επιτύνθηκαν έπειτα από 153 εποχές.



# Αξιολόγηση Επίδοσης

Τα γραφήματα παρακάτω απεικονίζουν τις επιδόσεις του μοντέλου σύμφωνα με τις κύριες μετρικές, παρέχοντας σημαντικές πληροφορίες για την ικανότητα γενίκευσής του και τις επιδόσεις του σε ρεαλιστικές συνθήκες.

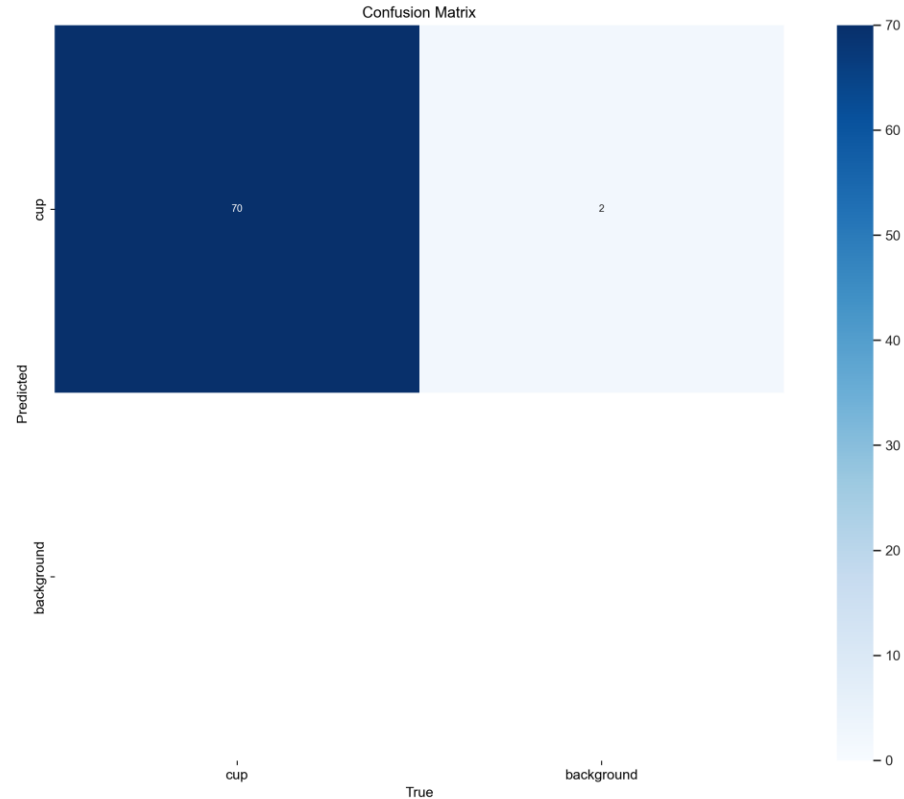


CLI εντολή:

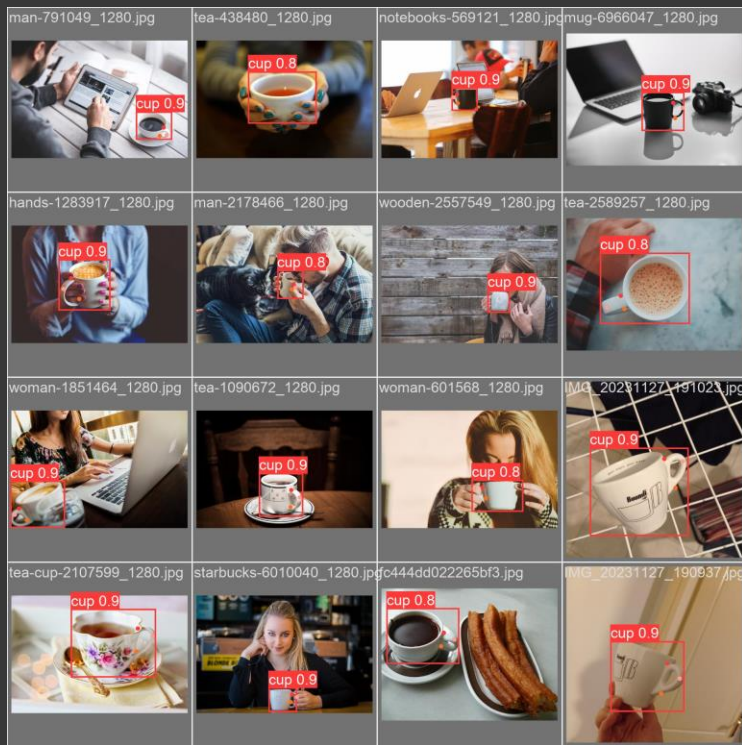
```
yolo val data='config.yaml' imgsz=640 model='./runs/pose/train34/weights/best.pt' plots=True split=test
```

# Confusion Matrix

Εδώ φαίνεται ο πίνακας σύγχυσης του μοντέλου (Confusion Matrix), πάνω στο σύνολο ελέγχου. Το μοντέλο εντόπισε σωστά και τις 70 κούπες όμως είχε 2 επιπλέον ψευδώς θετικές προβλέψεις.



# Σύντομη ζωντανή επίδειξη λειτουργίας του μοντέλου στην ανίχνευση λαβών



# 05

## Ενσωμάτωση σε Ρομποτικό Βραχίονα

---

Χρήση ρομποτικού βραχίονα για την αυτόνομη  
ανασήκωση κούπας από τη λαβή.



# Βήματα Υλοποίησης

---

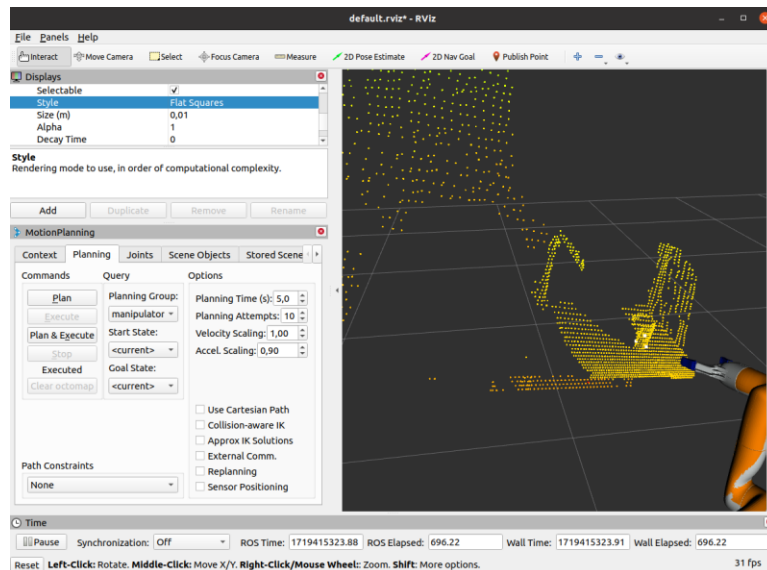
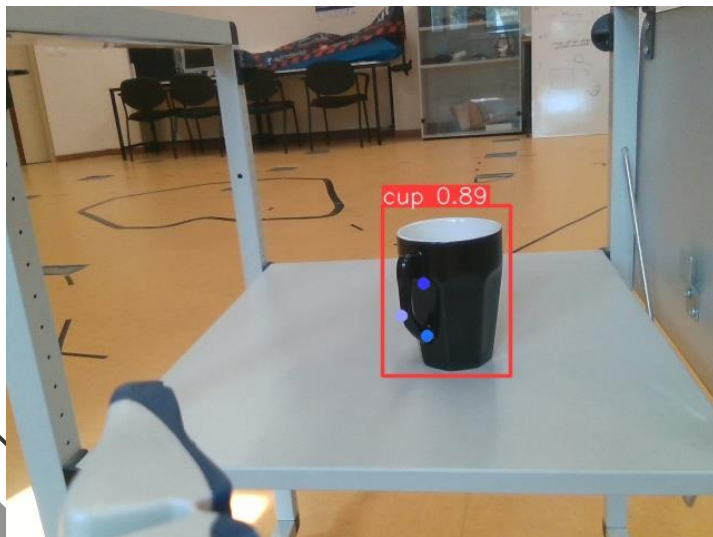
Ο σκοπός του πειράματος είναι να προσδιοριστούν οι ακριβείς συντεταγμένες των σημείων, ώστε να τοποθετηθεί ιδανικά ο βραχίονας και αφού σφίξει την λαβή, να ανυψώσει την κούπα. Είναι σημαντικό ότι ο εντοπισμός του YOLOv8 τρέχει διαρκώς στο παρασκήνιο, έχοντας σαν είσοδο τη ζωντανή ροή από την κάμερα που βρίσκεται στο άκρο της λαβίδας του βραχίονα.

Ο βραχίονας είναι στερεωμένος στο πάνω μέρος ενός αμαξιδίου και η κούπα βρίσκεται εντός του εύρους κίνησής του.



# Καθορισμός Θέσης Σημείων

Το πρώτο βήμα είναι η μετατροπή των σημείων από 2D σε **3D συντεταγμένες**, με χρήση του **point cloud** της κάμερας για να εξάγουμε τις αποστάσεις των keypoints στον πραγματικό κόσμο.

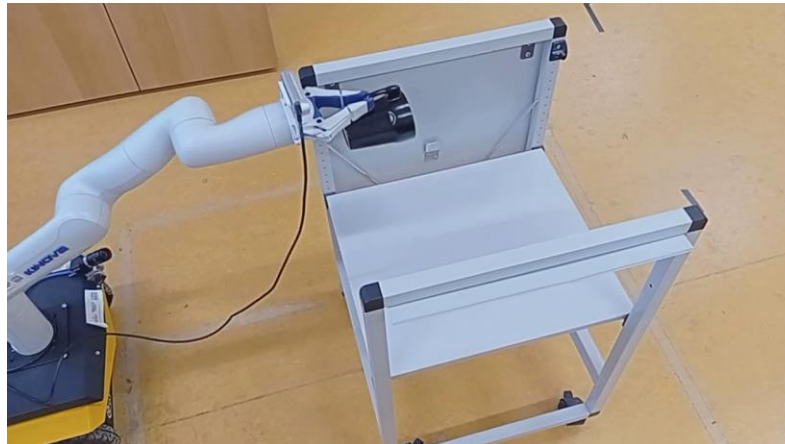


# Υπολογισμός Τελικής Θέσης Βραχίονα

---

Γίνεται μετασχηματισμός του συστήματος συντεταγμένων ώστε να υπολογιστούν οι συντεταγμένες των keypoints ως προς τη βάση του βραχίονα.

Η τελική θέση υπολογίζεται μέσω αντίστροφης κινηματικής και αριθμητικής επίλυσης με βελτιστοποίηση. Έπειτα, γίνεται το **motion planning** και ο βραχίονας μεταβαίνει στην τελική θέση για να πιάσει την λαβή.



# Βίντεο Επίδειξης Grasping



<https://www.youtube.com/watch?v=9cweFydTjcQ>



# Σας Ευχαριστώ

CREDITS: This presentation template was created by [Slidesgo](#).