

## Όραση Υπολογιστών: Εργασία 4

---

### A' ΜΕΡΟΣ

Ζητείται να υλοποιηθεί αρχιτεκτονική συνελκτικού δικτύου σε Python που θα αφορά στο πρόβλημα της ανίχνευσης αντικειμένου (object detection) και συγκεκριμένα στο πρόβλημα ανίχνευσης σημάτων οδικής κυκλοφορίας (Traffic Sign Detection). Επιτρέπεται η χρήση των βιβλιοθηκών [Keras-Tensorflow](#), αλλά και η βιβλιοθήκη [Pytorch](#). Οι υποβολές των υλοποιήσεων θα γίνονται με τη μορφή Jupyter Notebook, ενώ για την ανάπτυξη και δοκιμή θα χρησιμοποιηθεί η πλατφόρμα [Google Colab](#).

Στα πλαίσια της εργασίας θα πρέπει να εκπαιδευτούν και να υποβληθούν προς σύγκριση ΔΥΟ αρχιτεκτονικές:

- i. Ο two-stage ανιχνευτής Faster-RCNN που παρουσιάστηκε στο εργαστήριο.
- ii. Ο one-stage ανιχνευτής YOLOv3. Για τη χρήση του συγκεκριμένου ανιχνευτή έχετε τη δυνατότητα να επιλέξετε μια δικιά σας υλοποίηση πέραν της διαθέσιμης από τη βιβλιοθήκη [ultralytics](#).

Κάθε μια από τις προσεγγίσεις θα πρέπει να συνοδεύεται από:

A) **Πλήρη περιγραφή της αρχιτεκτονικής** και των επιπέδων που χρησιμοποιούνται.

B) **Ποσοτική εκτίμηση της επίδοσής τους** στο σύνολο δοκιμής, ως το ποσοστό επιτυχίας ανίχνευσης με βάση τη μετρική mAP.

Γ) **Περιγραφή και δικαιολόγηση της διαδικασίας εκπαίδευσης** (εποχές εκπαίδευσης, input size, batch size, callbacks, προ-επεξεργασία, data augmentation).

### B' ΜΕΡΟΣ

Στο δεύτερο μέρος της εργασίας ζητείται η χρήση του μοντέλου **Grounding DINO**, το οποίο αξιοποιεί **γλωσσικές προτροπές (text prompts)** σε συνδυασμό με οπτική πληροφορία για την ανίχνευση αντικειμένων, χωρίς να απαιτείται επιπλέον επανεκπαίδευση ή fine-tuning στο συγκεκριμένο dataset. Το μοντέλο θα εφαρμοστεί **σε zero-shot learning**, δηλαδή χωρίς καμία προσαρμογή στις εικόνες του dataset που χρησιμοποιήθηκε στο Α' μέρος.

**Για το B' μέρος της εργασίας ζητείται:**

- i. Να γίνει χρήση του **προεκπαιδευμένου μοντέλου Grounding** για την **ανίχνευση σημάτων κυκλοφορίας**, δίνοντας κατάλληλες **φυσικές γλωσσικές προτροπές (text prompts)** όπως:  
"stop sign, speedLimit25 sign, yield sign"
- ii. Να πραγματοποιηθεί **ποιοτική και ποσοτική αξιολόγηση** της επίδοσης του μοντέλου στις εικόνες δοκιμής του ίδιου dataset που χρησιμοποιήθηκε στο Α' μέρος.
- iii. Να πραγματοποιηθεί **σύγκριση με τα αποτελέσματα του Α' μέρους** τόσο σε επίπεδο ακρίβειας ανίχνευσης (mAP & recall), όσο και σε ποιοτικό επίπεδο (ενδεικτικά παραδείγματα επιτυχών/αποτυχημένων ανιχνεύσεων).

## **Υποδείξεις & σημειώσεις:**

Σημείωση 1: Για την παρούσα εργασία παρέχεται το σύνολο δεδομένων Lisa Traffic Sign Dataset. Το συγκεκριμένο σύνολο δεδομένων είναι χωρισμένο σε τρία υποσύνολα 'train', 'val' και 'test'. Αποτελεί σημαντικό βήμα η σωστή προσπέλαση των δεδομένων (εκόνων και labels) για την εκπαίδευση των νευρωνικών σας. Ενδέχεται να χρειάζεται ο μετασχηματισμός του dataset structure ώστε να ταιριάζει με τις ανάγκες του εκάστοτε νευρωνικού δικτύου. Από το τον παρακάτω σύνδεσμο να γίνει η λήψη του συνόλου δεδομένων:

[LISA Traffic Sign Dataset download link](#)

Στο αρχείο annotations\_with\_splits.csv θα βρείτε το groundtruth το οποίο απαρτίζεται από τις συντεταγμένες και την κλάση του κάθε αντικειμένου.

Το σύνολο δεδομένων απαρτίζεται από τις εξής 9 κλάσεις:

0: keepRight

1: merge

2: pedestrianCrossing

3: signalAhead

4: speedLimit25

5: speedLimit35

6: stop

7: yield

8: yieldAhead

Σημείωση 2: Για την εκπαίδευση και τη δοκιμή του yolov3 θα χρειαστεί να υλοποιήσετε και να αναρτήσετε στο colab το κατάλληλο data.yaml. Αυτό θα πρέπει να περιλαμβάνει το path του dataset, τα relative paths των train set, val set και test set, καθώς και την αριθμημένη λίστα των κλάσεων με τις οποίες θα εκπαιδευτεί το μοντέλο. Χρησιμοποιήστε ως βάση το αντίστοιχο που έχει αναρτηθεί στο υλικό του εργαστηρίου.

Επιπλέον, για το yolov3 θα χρειαστεί να διαμορφώσετε το dataset ώστε να έχει την εξής δομή:

```

db_lisa_tiny/
├── train/
│   ├── images/
│   │   ├── image1.png
│   │   ├── image2.png
│   │   └── ...
│   └── labels/
│       ├── image1.txt
│       ├── image2.txt
│       └── ...
├── val/
│   ├── images/
│   │   └── ...
│   └── labels/
│       └── ...
├── test/
│   ├── images/
│   │   └── ...
│   └── labels/
│       └── ...
└── data.yaml

```

Σημείωση 2: Δίνεται το αρχείο *'template.ipynb'* το οποίο περιέχει κώδικα για απευθείας μεταφόρτωση της βάσης δεδομένων, χωρίς τη χρήση Google Drive. Θα το βρείτε στον παρακάτω σύνδεσμο:

<https://vc.ee.duth.gr:6960/index.php/s/JnNhDD9w3wCXZwd>

## **Βοηθητικά σχόλια & Πηγές**

### **Επαύξηση δεδομένων (Data Augmentation)**

Συνήθως, τα βαθιά νευρωνικά δίκτυα επιτυγχάνουν καλύτερα αποτελέσματα όταν είναι διαθέσιμο μεγάλο πλήθος δεδομένων εκπαίδευσης. Είναι δυνατή η παραγωγή επιπλέον τεχνητών δεδομένων εκπαίδευσης από τις διαθέσιμες εικόνες, μέσω της εφαρμογής μετασχηματισμών. Κάποιοι μετασχηματισμοί που εφαρμόζονται συχνά είναι οι εξής:

- Flipping
- Random cropping
- Rotation
- Zoom

[tf.keras.preprocessing.image.ImageDataGenerator](#) | TensorFlow Core v2.7.0

[How to Configure Image Data Augmentation in Keras \(machinelearningmastery.com\)](#)

<https://docs.pytorch.org/vision/0.21/transforms.html>

<https://medium.com/@BurtMcGurt/a-practical-guide-to-data-augmentation-in-pytorch-with-examples-and-visualizations-761ad5c2a903>

### Χωρικές διαστάσεις

Οι αρχιτεκτονικές συνελκτικών νευρωνικών δικτύων που έχουμε δει στο εργαστήριο έχουν το εξής κοινό χαρακτηριστικό: Καθώς “βαθαίνει” το δίκτυο, οι χωρικές διαστάσεις των feature maps μειώνονται. Δύο τρόποι για να “διατηρήσουμε” τις χωρικές διαστάσεις των feature maps είναι οι εξής:

1. Αντί για το μοτίβο conv  $\rightarrow$  pool  $\rightarrow$  conv  $\rightarrow$  pool, μπορούμε να εφαρμόσουμε πολλαπλά επίπεδα συνέλιξης πριν από κάποιο επίπεδο pooling (π.χ. conv  $\rightarrow$  conv  $\rightarrow$  conv  $\rightarrow$  pool)
2. Χρήση padding “same” στα επίπεδα συνέλιξης

(βλ. [Conv2D layer \(keras.io\)](https://keras.io/layers/conv2d/))

### Batch size

Όσο μεγαλύτερο batch size, τόσο καλύτερα! Όμως, η μνήμη της κάρτας γραφικών δεν επαρκεί, έτσι επιλέγουμε το batch size στο μέγιστο δυνατό (μέχρι την τιμή εκείνη που εμφανίζεται memory error).

Επιπλέον υλικό:

<https://keras.io/callbacks/>

[A Gentle Introduction to Batch Normalization for Deep Neural Networks \(machinelearningmastery.com\)](https://machinelearningmastery.com/a-gentle-introduction-to-batch-normalization-for-deep-neural-networks/)

[A Gentle Introduction to Dropout for Regularizing Deep Neural Networks \(machinelearningmastery.com\)](https://machinelearningmastery.com/a-gentle-introduction-to-dropout-for-regularizing-deep-neural-networks/)