

Όραση Υπολογιστών: Εργασία 3

A' ΜΕΡΟΣ

Ζητείται να υλοποιηθεί πρόγραμμα σε Python με τη χρήση της βιβλιοθήκης OpenCV το οποίο θα αφορά στο πρόβλημα της ταξινόμησης πολλαπλών κλάσεων (multi-class classification). Το πρόγραμμα υλοποιείται με την εκτέλεση των παρακάτω βημάτων :

1. Παραγωγή οπτικού λεξικού (visual vocabulary) βασισμένη στο μοντέλο Bag of Visual Words (BOVW). Η δημιουργία του λεξικού να γίνει με τη χρήση του αλγορίθμου K-Means χρησιμοποιώντας όλες τις εικόνες του συνόλου εκπαίδευσης (train).
2. Εξαγωγή περιγραφέα σε κάθε εικόνα εκπαίδευσης (train) βασισμένου στο μοντέλο BOVW χρησιμοποιώντας το λεξικό που προέκυψε κατά το βήμα 1.
3. Με δεδομένο το αποτέλεσμα του βήματος 2, να υλοποιηθεί η λειτουργία ταξινόμησης μιας εικόνας κάνοντας χρήση των δυο παρακάτω ταξινομητών :
 - a. Του αλγορίθμου k-NN
 - b. Του σχήματος one-versus-all όπου για κάθε κλάση εκπαιδεύεται ένας SVM ταξινομητής
4. Αξιολόγηση του συστήματος: Χρησιμοποιώντας το σύνολο δοκιμής (test), να μετρηθεί η ακρίβεια του συστήματος (και στις δύο περιπτώσεις ταξινομητών) που εκφράζεται ως το ποσοστό των επιτυχών ταξινομήσεων, συνολικά και ανά κλάση.

Κατά την αξιολόγηση να ελέγχετε την επίδραση των παρακάτω εμπλεκομένων παραμέτρων:

- a. Μέγεθος του λεξικού οπτικών λέξεων (Βήμα 1). Να δοκιμάσετε τουλάχιστον πέντε (5) διαφορετικές τιμές λεξικού και να επιλέξετε αυτή για την οποία έχουμε τη μεγαλύτερη ακρίβεια.
- b. Αριθμός των πλησιέστερων γειτόνων (k) που εξετάζει ο k-NN (Βήμα 3a). Ποια είναι η βέλτιστη τιμή για το συγκεκριμένο σενάριο;

B' ΜΕΡΟΣ

Ζητείται να υλοποιηθεί αρχιτεκτονική συνελικτικού δικτύου σε Python με τη χρήση της βιβλιοθήκης [Keras-Tensorflow](#), που θα αφορά στο πρόβλημα της ταξινόμησης πολλαπλών κλάσεων (multi-class classification). 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 υλοποιηθούν ΔΥΟ αρχιτεκτονικές δικτύων:

- i. Ένα **ΜΗ προ-εκπαιδευμένο** δίκτυο το οποίο θα δημιουργηθεί αποκλειστικά για το τρέχον πρόβλημα ταξινόμησης, με τη διαδικασία που έχει αποτυπωθεί κατά το εργαστήριο.
- ii. Ένα **προ-εκπαιδευμένο** δίκτυο της επιλογής σας, που θα συνοδεύεται και από την αντίστοιχη ανάλυση του δικτύου που επιλέχθηκε. Να συμβουλευτείτε τον παρακάτω σύνδεσμο, όπου παρατίθεται η λίστα με τα προ-εκπαιδευμένα δίκτυα που είναι διαθέσιμα:

Keras Applications

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

Γ' ΜΕΡΟΣ

Αφού ολοκληρώσετε επιτυχώς το Α' και Β' μέρος της εργασίας θα πρέπει να κάνετε μια συγκριτική ανάλυση μεταξύ των διαφορετικών αλγορίθμων ταξινόμησης:

- a. Κλασσικών ταξινομητών (k-NN & SVM)
- b. Νευρωνικών Δικτύων

Η συγκριτική ανάλυση θα πρέπει να βασίζεται στις διαφορές της επεξεργασίας των δεδομένων, στην εκπαίδευση και το finetuning των αλγορίθμων, αλλά και στις δυνατότητες αυτών βάσει της αξιολόγησης που κάνατε στο Α' και Β' μέρος.

Υποδείξεις & σημειώσεις:

Σημείωση 1: Οι αλγόριθμοι του Α' και Β' μέρους θα εκπαιδευτούν και θα δοκιμαστούν σε 2 κοινά σύνολα δεδομένων με σκοπό τη σύγκριση τους.

Συγκεκριμένα, δίνεται ο φάκελος 'caltech-transportation' που περιέχει τις δύο βάσεις εικόνων 'caltech-transportation_train' και 'caltech-transportation_test'. Η πρώτη θα χρησιμοποιηθεί για την εκπαίδευση του συστήματος και η δεύτερη για την δοκιμή και την αξιολόγησή του. Οι βάσεις αυτές αποτελούν υποσύνολο της βάσης εικόνων [Caltech-101 Dataset](#)

Download link: <https://vc.ee.duth.gr:6960/index.php/s/4PIELUtjzbSmfYf>

Έπειτα, οι αλγόριθμοι του Α' και Β' μέρους θα εκπαιδευτούν και θα αξιολογηθούν στις εικόνες του φακέλου GTSRB, ο οποίος με τη σειρά του περιλαμβάνει τις δύο βάσεις εικόνων 'train_modified' και 'test_modified'. Η πρώτη θα χρησιμοποιηθεί για την εκπαίδευση του συστήματος και η δεύτερη για την δοκιμή και την αξιολόγησή του. Οι βάσεις αυτές αποτελούν υποσύνολο της βάσης δεδομένων [GTSRB - The German Traffic Sign Recognition Benchmark](#).

Download link: <https://vc.ee.duth.gr:6960/index.php/s/t8BYhmNQLLPs8Rp>

Σημείωση 2: Για τη δημιουργία του SVM (Μέρος Α, Ερώτημα 3b) να χρησιμοποιήσετε Linear kernel, με την εντολή: `svm.setKernel(cv.ml.SVM_LINEAR)`

Σημείωση 3: Δίνεται το αρχείο 'template.ipynb' το οποίο περιέχει κώδικα για απευθείας μεταφόρτωση

της βάσης δεδομένων GTSRB. Μπορείτε να το χρησιμοποιήσετε τροποποιώντας το και για τη βάση δεδομένων ‘caltech-transportation’.

<https://vc.ee.duth.gr:6960/index.php/s/xBToGNYSaNr66PS>

Βοηθητικά σχόλια & Πηγές

SVM one verus all

Το **One-vs-All (OvA)** είναι μια στρατηγική που χρησιμοποιείται για την αντιμετώπιση προβλημάτων πολυκατηγορικής ταξινόμησης (multi-class classification) με αλγόριθμους που αρχικά υποστηρίζουν μόνο δυαδική ταξινόμηση (binary classification), όπως ο SVM.

Πώς λειτουργεί το One-vs-All; Για κάθε κλάση εκπαιδεύεται ένα ξεχωριστό μοντέλο SVM. Το κάθε μοντέλο εκπαιδεύεται ώστε να ξεχωρίζει μία συγκεκριμένη κατηγορία από όλες τις υπόλοιπες. Για παράδειγμα, αν έχουμε 3 κατηγορίες (Α, Β, Γ):

- Το μοντέλο 1 θα εκπαιδευτεί να διακρίνει την κατηγορία **Α** από τις κατηγορίες **Β** και **Γ**.
- Το μοντέλο 2 θα εκπαιδευτεί να διακρίνει την κατηγορία **Β** από τις κατηγορίες **Α** και **Γ**.
- Το μοντέλο 3 θα εκπαιδευτεί να διακρίνει την κατηγορία **Γ** από τις κατηγορίες **Α** και **Β**.

1. Εκπαίδευση:

- Σε κάθε μοντέλο, τα δείγματα της συγκεκριμένης κλάσης σημειώνονται με την ετικέτα 1 (θετική κατηγορία), ενώ τα υπόλοιπα σημειώνονται με την ετικέτα 0 (αρνητική κατηγορία).

2. Πρόβλεψη (Prediction):

- Για ένα νέο δείγμα, περνάμε το δείγμα από όλα τα μοντέλα OvA.
- Κάθε μοντέλο επιστρέφει μία τιμή εμπιστοσύνης (confidence score) ή απόφασης. Η τελική κλάση είναι αυτή για την οποία το αντίστοιχο μοντέλο επιστρέφει τη μέγιστη τιμή εμπιστοσύνης.

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

Χωρικές διαστάσεις

Οι αρχιτεκτονικές συνελικτικών 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 → pool → conv → pool, μπορούμε να εφαρμόσουμε πολλαπλά επίπεδα συνέλιξης πριν από κάποιο επίπεδο pooling (π.χ. conv → conv → conv → pool)
2. Χρήση padding “same” στα επίπεδα συνέλιξης

(βλ. [Conv2D layer \(keras.io\)](#))

Χρήση συνάρτησης flow_from_directory:

Στην περίπτωση που χρησιμοποιηθεί η συνάρτηση flow_from_directory προσοχή στις παραμέτρους → (https://keras.io/preprocessing/image/#flow_from_directory)

Προσοχή: το target_size έχει default τιμή (256,256).

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)

A Gentle Introduction to Dropout for Regularizing Deep Neural Networks
(machinelearningmastery.com)