

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ - ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ**

**Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας**

**Ακαδημαϊκό Έτος 2021-2022 (Εαρινό Εξάμηνο)**

**Εργαστηριακές Ασκήσεις - Μέρος Β**

**Διαδικαστικά Ζητήματα**

1. Υπενθυμίζουμε ότι οι εργασίες είναι ατομικές. Κάθε άτομο θα πρέπει να επιλέξει ένα από τα θέματα που ακολουθούν.
2. Η τεχνική αναφορά που θα υποβάλετε θα πρέπει να περιλαμβάνει συνοπτική περιγραφή του θέματος, επαρκή σχολιασμό των αποτελεσμάτων τα οποία προκύπτουν σε κάθε περίπτωση και επίσης θα πρέπει να συνοδεύεται από τον κώδικα που γράψατε.
3. Προτεινόμενο περιβάλλον ανάπτυξης για τις διάφορες επεξεργασίες είναι το περιβάλλον Matlab. Υλοποίηση σε άλλα περιβάλλοντα (π.χ. Python) είναι επίσης αποδεκτή.
4. Η παράδοση της άσκησης θα γίνει ηλεκτρονικά στο eclass μέσω της ενότητας Εργασίες. Διευκρινίσεις θα δίνονται μέσω της Ενότητας “Συζητήσεις” του μαθήματος στο e-class και μέσω email στα [st1003586@ceid.upatras.gr](mailto:st1003586@ceid.upatras.gr) (Αλέξανδρος Γκίλλας – Θέμα 1), [pipeirigkos@ceid.upatras.gr](mailto:pipeirigkos@ceid.upatras.gr) (Νικόλαος Πιπερίγκος - Θέμα 2 & Θέμα 4) και [egeorgatos@ceid.upatras.gr](mailto:egeorgatos@ceid.upatras.gr) (Ευάγγελος Γεωργάτος – Θέμα 3).
5. Προθεσμία παράδοσης ορίζεται η **03/07/2021 12.00μμ**, σε περίπτωση που επιθυμείτε να σταλεί βαθμός τον Ιούλιο. Η προθεσμία παράδοσης για τον Σεπτέμβριο θα ανακοινωθεί αργότερα.

## ΘΕΜΑ 1: Κατηγοριοποίηση Εικόνων – Σύγκριση δύο αντιπροσωπευτικών μεθόδων

### Μέρος Α: Κατηγοριοποίηση Εικόνων με χρήση Συνελικτικών Νευρωνικών Δικτύων (Convolutional Neural Networks - CNN)

Στο πλαίσιο αυτού του ερωτήματος, το ζητούμενο είναι η εκπαίδευση ενός νευρωνικού δικτύου για την κατηγοριοποίηση εικόνων. Το συγκεκριμένο πρόβλημα είναι πρόβλημα επιβλεπόμενης μάθησης, γεγονός που σημαίνει ότι υπάρχει διαθέσιμη μία βάση δεδομένων με εικόνες  $x \in X$  και κάθε εικόνα περιγράφεται από μία ετικέτα  $y \in Y, Y = \{1, \dots, C\}$  σχηματίζοντας ζεύγη της μορφής  $d_i = \{x_i, y_i\}, d_i \in D$ .

Το πρόβλημα εκπαίδευσης ενός νευρωνικού δικτύου (όπως και πλήθος προβλημάτων που εμφανίζονται στην μηχανική μάθηση) μπορεί να αναπαρασταθεί ως ένα πρόβλημα ελαχιστοποίησης και πιο συγκεκριμένα σύμφωνα με το παρακάτω πρόβλημα στοχαστικής βελτιστοποίησης:

$$\min_W F(X, W), \quad F(X, W) = \mathbb{E}_{X \sim P} \{\ell(W, X)\} \quad (1)$$

όπου  $P = P(x, y)$  η κατανομή η οποία περιγράφει στατιστικά τα δεδομένα και  $W$  η μήτρα η οποία περιέχει το σύνολο των παραμέτρων του νευρωνικού δικτύου και  $\ell(\cdot)$  μία συνάρτηση κόστους η οποία μοντελοποιεί το κόστος για κάθε λανθασμένη πρόβλεψη για κάθε στοιχείο του συνόλου  $X$ .

Η γνώση της κατανομής  $P(x, y)$  στην πλειοψηφία των προβλημάτων είναι δύσκολη και ακριβή πληροφορία οπότε και το πρόβλημα βελτιστοποίησης (1) λύνεται προσεγγιστικά, προσεγγίζοντας τον τελεστή Expectation (δηλ. τον  $\mathbb{E}\{\cdot\}$ ) από τον αριθμητικό μέσο των διαθέσιμων δεδομένων τα οποία έχουν παραχθεί από την κατανομή  $P(x, y)$ , καταλήγοντας στο παρακάτω πρόβλημα ελαχιστοποίησης εμπειρικού ρίσκου:

$$\min_W F(X, W), F(X, W) = \frac{1}{|D|} \sum_{i=1}^{|D|} f_i(W) \quad (2)$$

όπου  $f_i(W) = \ell(x_i, y_i; W)$  το κόστος της λανθασμένης πρόβλεψης για το στοιχείο  $\{x_i, y_i\}$  δεδομένου των παραμέτρων του νευρωνικού δικτύου  $W$ .

Ο αλγόριθμος που θα χρησιμοποιηθεί για την επίλυση του προβλήματος (2) είναι η μέθοδος Stochastic Gradient Descent (SGD). Αρχικά, ο συγκεκριμένος επαναληπτικός

αλγόριθμος υπολογίζει μία στοχαστική προσέγγιση του διανύσματος κλίσης της (2). Στη συνέχεια ανανεώνει τις παραμέτρους  $W$  σύμφωνα με την αρνητική διεύθυνση του gradient βάση της παρακάτω επαναληπτικής διαδικασίας

$$W_{t+1} = W_t - \mu \hat{\nabla} F(X, W_t)$$

όπου  $\mu$  (βήμα ή ρυθμός μάθησης) είναι μία μικρή θετική σταθερά η οποία ελέγχει τον ρυθμό με τον οποίο «κινείται» η μήτρα των παραμέτρων  $W$  προς την κατεύθυνση που υποδεικνύει το διάνυσμα κλίσης σε κάθε επανάληψη.

**Η αρχιτεκτονική του Νευρωνικού δικτύου που ζητείται να υλοποιήσετε περιγράφεται αναλυτικά στην Εικόνα 1.**

**Ερωτήματα**

1. Αρχικά, κατεβάστε το MNIST από τον παρακάτω σύνδεσμο <http://yann.lecun.com/exdb/mnist/> και απεικονίστε σε ένα figure μία εικόνα από κάθε κλάση. Οι εικόνες είναι χωρισμένες σε δεδομένα εκπαίδευσης (training data) και δεδομένα ελέγχου (testing data).
2. Υλοποιήστε την επαναληπτική διαδικασία του αλγορίθμου Stochastic Gradient Descent για την εκπαίδευση ενός NN το οποίο ελαχιστοποιεί την συνάρτηση κόστους (2). Πιο συγκεκριμένα, θα πρέπει να υλοποιήσετε τα παρακάτω βήματα:
  - a) Χωρίστε τα δεδομένα εκπαίδευσης σε μη επικαλυπτόμενα υποσύνολα (mini-batches) μεγέθους  $b$ .
  - b) Χρησιμοποιώντας σε κάθε επανάληψη ένα διαφορετικό mini-batch υπολογίστε μία (στοχαστική) προσέγγιση για το gradient  $\hat{\nabla}_{W_t} F(X; W_t)$ .
  - c) Χρησιμοποιώντας το gradient που υπολογίσατε στο προηγούμενο βήμα ανανεώστε τις παραμέτρους του NN σύμφωνα με την εξίσωση του SGD.

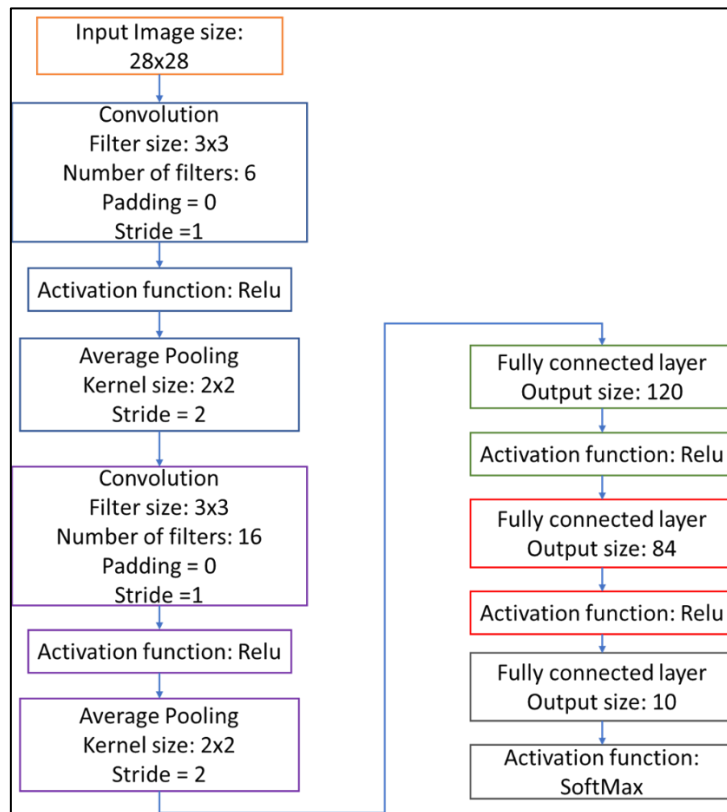
Τα παραπάνω βήματα (α)-(c) επαναλαμβάνονται μέχρι να συγκλίνει ο SGD σε κάποιο τοπικό ελάχιστο της συνάρτησης (2). Ακόμη, προκειμένου να εξασφαλιστεί η σύγκλιση του SGD είναι χρήσιμο ο SGD να επεξεργαστεί πολλές φορές το σύνολο δεδομένων. Το κάθε «πέρασμα» των δεδομένων σηματοδοτεί μία εποχή (epoch). Συνεπώς, θα είναι χρήσιμο να χρησιμοποιήσετε αρκετές εποχές προκειμένου να εξασφαλιστεί η σύγκλιση του SGD.

3. Χρησιμοποιώντας το σύνολο δεδομένων ελέγχου (testing dataset) του MNIST στο τέλος κάθε εποχής υπολογίστε το σφάλμα κατηγοριοποίησης (loss) και την ακρίβεια (accuracy) του μοντέλου σας και απεικονίστε τα σε δύο διαφορετικές γραφικές παραστάσεις.
4. Υπολογίστε (μέσω συνάρτησης που θα κατασκευάσετε εσείς) τον confusion matrix με βάση τα αποτελέσματα που εξάγατε.

**Υποδείξεις:**

A) Προτείνεται η χρήση του MATLAB Deep Learning Toolbox ή οποιουδήποτε άλλου λογισμικού της αρεσκείας σας (Keras – Tensorflow, Pytorch).

B) Για την υλοποίηση των βημάτων (α)-(ε) μπορείτε να χρησιμοποιήσετε τις έτοιμες συναρτήσεις που προσφέρουν τα παραπάνω λογισμικά.



**Εικόνα 1.** Αρχιτεκτονική του Συνελκτικού Νευρωνικού Δικτύου

## **Μέρος Β: Κατηγοριοποίηση Εικόνων με χρήση Histogram of Oriented Gradients και Support Vector Machines**

Χρησιμοποιήστε το MNIST dataset από το προηγούμενο ερώτημα.

### ***Υπόβαθρο:***

Ο περιγραφέας χαρακτηριστικών «Histogram of Oriented Gradients» (HOG) είναι ένα κλασικό εργαλείο της Επεξεργασίας Εικόνας και της Υπολογιστικής όρασης για την ανίχνευση αντικειμένων. Με την τεχνική αυτή ομαδοποιούνται οι τιμές της κατεύθυνσης της κλίσης της εικόνας σε μια συγκεκριμένη υποπεριοχή της, με στόχο την καλύτερη περιγραφή της δομής και του σχήματος του αντικειμένου. Για τον λόγο αυτό, κατασκευάζεται ένα ιστόγραμμα με βάση το πλάτος και την κατεύθυνση της κλίσης της εικόνας. Πιο συγκεκριμένα:

1. Για κάθε patch  $8 \times 8$  της εικόνας, υπολογίζεται η κλίση των pixel περιλαμβάνει, ως προς τις γραμμές και στήλες  $G_x$ ,  $G_y$ . Στη συνέχεια, και πάλι για κάθε pixel, υπολογίζεται το πλάτος και η κατεύθυνση της κλίσης, δηλαδή  $g = \sqrt{G_x^2 + G_y^2}$  και  $\theta = \arctan \frac{G_y}{G_x}$

2. Για το ίδιο patch, κατασκευάζεται ένα ιστόγραμμα 9 bins, όπου τα (αριστερά) όρια κάθε bin τίθενται ίσα με  $0^\circ, 20^\circ, 40^\circ, 60^\circ, 80^\circ, 100^\circ, 120^\circ, 140^\circ, 160^\circ$ . Παραδείγματος χάριν, εάν  $\theta = 20^\circ$  τότε στο  $1^\circ$  bin θα προστεθεί η αντίστοιχη τιμή του  $g$ . Εάν όμως  $\theta = 10^\circ$ , τότε στα 2 πρώτα bin προστίθεται η τιμή  $\frac{g}{2}$ .
3. Σε επικαλυπτόμενα παράθυρα  $16 \times 16$ , τα 4 ιστογράμματα των 9 bins συγκεντρώνονται σε ένα τελικό διάνυσμα μεγέθους  $36 \times 1$ , το οποίο κανονικοποιείται διαιρώντας κάθε στοιχείο του με το μέτρο του διανύσματος.
4. Στο τέλος της διαδικασίας, όλα τα κανονικοποιημένα διανύσματα  $36 \times 1$  συγκεντρώνονται σε ένα συνολικό διάνυσμα, το οποίο αποτελεί τον περιγραφέα HOG της εικόνας. Η οπτική απεικόνιση των χαρακτηριστικών HOG γίνεται με τα κανονικοποιημένα διανύσματα (ιστογράμματα) κάθε patch  $8 \times 8$ .

Οι κατηγοριοποιητές Support Vector Machines (SVM) είναι ένα εργαλείο της Στατιστικής και Μηχανικής Μάθησης. Στο πρόβλημα της δυαδικής κατηγοποίησης, στοχεύουν να διαχωρίσουν σε 2 διακριτές κλάσεις το χώρο στον οποίο βρίσκονται τα δεδομένα εκπαίδευσης, με την χρήση διανυσμάτων υποστήριξης. Ορίζεται συνεπώς ένα πρόβλημα βελτιστοποίησης το οποίο μεγιστοποιεί την απόσταση των δύο υπερεπιπέδων (κλάσεων), στα οποία διαμοιράζονται τα διανύσματα εκπαίδευσης. Τα διανύσματα εκπαίδευσης που βρίσκονται πάνω στα υπερεπίπεδα ονομάζονται διανύσματα υποστήριξης επειδή υποστηρίζουν τα υπερεπίπεδα και συμβάλλουν στην επίλυση του προβλήματος. Με παρόμοιο τρόπο, τα SVM επεκτείνονται και στο πρόβλημα της κατηγοριοποίησης περισσότερων των 2 κλάσεων.

### **Ερωτήματα:**

- [1] Υπολογίστε τα χαρακτηριστικά HOG του συνόλου δεδομένων εκπαίδευσης με έτοιμη συνάρτηση της MATLAB. Στη συνέχεια, εκπαιδεύστε ένα SVM με τα χαρακτηριστικά αυτά.
- [2] Με την χρήση του εκπαιδευμένου SVM, παρουσιάστε ενδεικτικά αποτελέσματα κατηγοριοποίησης για το σύνολο των δεδομένων ελέγχου. Πειραματιστείτε με διαφορετικό μέγεθος patch για τα HOGs, τόσο κατά την εκπαίδευση όσο και κατά τον έλεγχο.
- [3] Υπολογίστε (μέσω συνάρτησης που θα κατασκευάσετε εσείς) τον confusion matrix με βάση τα αποτελέσματα που εξάγατε. Συγκρίνεται τα αποτελέσματα με τα αποτελέσματα που προέκυψαν με την χρήση του Συνελικτικού δικτύου από το Α μέρος.
- [4] **Bonus:** Υπολογίστε τα HOG χωρίς την έτοιμη συνάρτηση της MATLAB.

### **Βιβλιογραφία**

- [1] R. C. Gonzalez, “Deep Convolutional Neural Networks [Lecture Notes],” IEEE Signal Process. Mag., vol. 35, no. 6, pp. 79–87, Nov. 2018.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016, <http://www.deeplearningbook.org>

[3]. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[4]. P. E. Rybski, D. Huber, D. D. Morris and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features," 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 921-928, doi: 10.1109/IVS.2010.5547996.

[5]. X. Yuan, L. Cai-nian, X. Xiao-liang, J. Mei and Z. Jian-guo, "A two-stage hog feature extraction processor embedded with SVM for pedestrian detection," 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 3452-3455, doi: 10.1109/ICIP.2015.7351445.

## **ΘΕΜΑ 2: Αξιολόγηση της απόδοσης Συνελκτικών Νευρωνικών Δικτύων κατά την ανίχνευση και κατηγοριοποίηση αντικειμένων κυκλοφοριακής σκηνής**

### **1. Εισαγωγή**

Τα Συνελκτικά Νευρωνικά Δίκτυα (ΣΝΝ) χρησιμοποιούνται ευρέως από την ακαδημαϊκή κοινότητα και τη βιομηχανία για εφαρμογές αυτόνομης οδήγησης. Τα ανταγωνιστικά τους πλεονεκτήματα προσφέρονται κυρίως κατά την ανίχνευση και κατηγοριοποίηση αντικειμένων, όπως οχήματα (επιβατηγά, φορτηγά κ.α.), πεζοί, δικυκλιστές κτλ., μέσω της καταγραφής τους από εικόνες. Το ΣΝΝ θα πρέπει να εκπαιδευτεί με ένα μεγάλο σύνολο κατάλληλα επιλεγμένων εικόνων εκπαίδευσης, μία διαδικασία η οποία συνήθως απαιτεί μεγάλη υπολογιστική και επεξεργαστική ισχύ. Μετά την εκπαίδευση, ακολουθεί η αξιολόγηση του ΣΝΝ με ένα σύνολο εικόνων ελέγχου. Η έξοδος του ΣΝΝ (όπως φαίνεται και στην **Εικόνα 1**) αποτελείται από 2Δ bounding boxes (με τις κατάλληλες συντεταγμένες), την κλάση του αντικειμένου (π.χ. όχημα) και την πιθανότητα ακριβούς ταξινόμησης στη συγκεκριμένη κλάση. Στα πλαίσια της Εργασίας αυτής, θα πειραματιστείτε με την αξιολόγηση της εξόδου των ΣΝΝ σε ό,τι αφορά τις μετρικές Intersection over Union (IoU), True Positive (TP), False Positive (FP), False Negative (FN), Precision, Recall και Area Under Curve (AUC). Το state-of-the-art ΣΝΝ στο οποίο θα στηριχθείτε είναι το SqueezeDet [1].

**Περιβάλλον υλοποίησης:** Αρχικά, θα κατεβάσετε το περιβάλλον Anaconda για Python. Στη συνέχεια, θα ανοίξετε το Anaconda prompt και θα εκτελέσετε την εντολή `git clone https://github.com/BichenWuUCB/squeezeDet.git`, που κατεβάζει από το github τον κώδικα πάνω στον οποίον θα δουλέψετε. Πλέον, θα εργαστείτε μέσα στον φάκελο SqueezeDet. Με την εντολή `pip install -r requirements.txt` θα εγκατασταθούν οι απαραίτητες εκδόσεις των πακέτων που χρειάζονται για να λειτουργήσει ο κώδικας. Εάν για τον οποιονδήποτε λόγο εργαστείτε με νεότερη έκδοση του Tensor Flow, θα



προκύψουν ζητήματα συμβατότητας που όμως είναι εύκολα αντιμετωπίσιμα. Έπειτα, κατεβάστε τον φάκελο που περιέχει τις προεκπαιδευμένες παραμέτρους των τεσσάρων δικτύων που δίνονται (resnet50, squeezeDet, squeezeDetPlus, vgg16), και αποθηκεύστε τον στον υποφάκελο data. Τέλος, εάν έχουν γίνει όλα σωστά, δίνοντας την εντολή `python ./src/demo.py` θα σας εμφανιστεί η παρακάτω εικόνα:



**Εικόνα 1:** Ενδεικτική έξοδος του SqueezeDet

Η ίδια διαδικασία περιγράφεται αναλυτικά και εδώ:  
<https://github.com/BichenWuUCB/squeezeDet>

## 2. Ερωτήματα:

1. Κατεβάστε το dataset εικόνων με το οποίο θα πειραματιστείτε από εδώ: [http://www.cvlibs.net/download.php?file=data\\_object\\_image\\_2.zip](http://www.cvlibs.net/download.php?file=data_object_image_2.zip) και το ground truth (σε μορφή .txt αρχείου) για κάθε αντικείμενο της εικόνας από εδώ: [http://www.cvlibs.net/download.php?file=data\\_object\\_label\\_2.zip](http://www.cvlibs.net/download.php?file=data_object_label_2.zip). Το dataset ελέγχου στο οποίο θα εργαστείτε θα αποτελείται από τις πρώτες 200 εικόνες. Η μορφή του ground truth .txt αρχείου περιγράφεται εδώ: [https://github.com/bostondiditeam/kitti/blob/master/resources/devkit\\_object\\_readme.txt](https://github.com/bostondiditeam/kitti/blob/master/resources/devkit_object_readme.txt). Να λάβετε υπόψιν σας μόνο όσα αντικείμενα αντιστοιχούν σε Car, Pedestrian και Cyclist.
2. Παρουσιάστε ενδεικτικά αποτελέσματα (5 ~ 8 εικόνες) ανίχνευσης αντικειμένων που παράγει το SqueezeDet.  
**Bonus:** Παρουσιάστε την έξοδο (bounding boxes) του SqueezeDet συνδυαστικά με τα ground truth αντικείμενα (δίνοντας το αντίστοιχο χρώμα).
3. Υπολογίστε το IOU για κάθε ανιχνευμένο αντικείμενο. Επίσης, υπολογίστε το μέσο IOU ανά frame και το τελικό μέσο IOU του dataset ελέγχου για το SqueezeDet και ένα οποιοδήποτε διαφορετικό δίκτυο (που θα το αναφέρετε). Αξιολογήστε την απόδοσή τους.

Προσοχή: Τα δίκτυα επιστρέφουν το κεντρικό σημείο του bounding box. Θα πρέπει να μετατρέψετε το σημείο αυτό σε μορφή συμβατή με το ground truth για τον σωστό υπολογισμό του IOU. Για το λόγο αυτό, χρησιμοποιείτε τη συνάρτηση `bbox_transform()` που βρίσκεται στο αρχείο `util.py`.

4. Επιλέξτε πέντε διαφορετικές τιμές IOU οι οποίες θα λειτουργούν ως κατώφλια (thr) υπολογισμού των TP, FP, FN για το σύνολο των δεδομένων του dataset ελέγχου.

Εάν  $\text{IOU} \geq \text{thr}$  για ανιχνευμένο αντικείμενο, τότε  $\text{TP} = \text{TP} + 1$ .

Εάν  $\text{IOU} < \text{thr}$  για ανιχνευμένο αντικείμενο, τότε  $\text{FP} = \text{FP} + 1$ .

Εάν το δίκτυο δεν εντόπισε αντικείμενο που αναφέρεται ως ground truth, τότε  $\text{FN} = \text{FN} + 1$ .

5. Υπολογίστε τις μετρικές True Positive Rate (TPR) και False Positive Rate (FPR) με βάση τον τελικό αριθμό TPs, FPs, FNs και TNs: 1) για κάθε κατώφλι και 2) για τα 2 δίκτυα. Αναφερθείτε στα βασικά σημεία των μετρικών TPR και FPR.
6. Για κάθε δίκτυο, κατασκευάστε την γραφική παράσταση Receiver Operating Characteristics (ROC) με βάση τις πέντε τιμές των TPRs και FPRs και υπολογίστε το AUC με τον κανόνα του τραπεζίου (έτοιμη συνάρτηση της Python). Αναφερθείτε στα βασικά σημεία του AUC και αξιολογήστε τα δύο δίκτυα με βάση την συγκεκριμένη μετρική.

## Βιβλιογραφία

- [1] B. Wu, A. Wan, F. Iandola, P. H. Jin and K. Keutzer, "SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 446-454.

## ΘΕΜΑ 3: Αποθορυβοποίηση Εικόνας μέσω αραιής αναπαράστασης και λεξικών μάθησης

Τα τελευταία χρόνια παρουσιάζεται έντονο ενδιαφέρον από την επιστημονική κοινότητα για την μελέτη της θεωρίας της αραιής αναπαράστασης και την μάθηση υπερπλήρων λεξικών. Με την χρήση ενός υπερπλήρους λεξικού  $D \in R^{n \times K}$ ,  $K > n$ , το οποίο περιέχει  $K$  άτομα (διανύσματα στηλών), ένα σύνολο από σήματα  $Y = (y_1, y_2, \dots, y_N)$ ,  $y_i \in R^n$  δύναται να εκφραστεί ως ένας αραιός γραμμικός συνδυασμός των ατόμων αυτών, σύμφωνα με την σχέση :

$$y_i \approx Dx_i, i = 1, \dots, N \text{ ή } Y \approx DX$$

Όπου το  $x_i \in R^K$  είναι το αραιό διάνυσμα που περιέχει ένα μικρό αριθμό από μη μηδενικούς συντελεστές και  $X$  είναι το μητρώο που περιέχει τα διανύσματα  $x_i$ .

Το μαθηματικό μοντέλο για την αναπαράσταση του σήματος  $y_i$  δίνεται ως:

$$\min \|y_i - Dx_i\|_2^2 \text{ s.t. } \|x_i\|_0 \leq s$$



Όπου  $\|\cdot\|_0$  είναι νόρμα-0 η οποία μετράει το πλήθος των μηδενικών στοιχείων και το  $s$  καθορίζει το ακριβές πλήθος των μη μηδενικών στοιχείων. Το  $s$  στην βιβλιογραφία αναφέρεται ως βαθμός αραιότητας (sparsity level).

Το εν λόγω πρόβλημα καθίσταται αδύνατο να επιλυθεί σε πολυωνυμικό χρόνο. Για την επίλυση του έχουν αναπτυχθεί διάφορες τεχνικές οι οποίες παρουσιάζουν αξιόλογα αποτελέσματα. Η πιο διαδεδομένη κατηγορία αλγόριθμων είναι οι άπληστοι (greedy) αλγόριθμοι όπως ο αλγόριθμος OMP (Orthogonal Matching Pursuit) [2].

Το βασικό πλεονέκτημα του μοντέλου της μάθησης λεξικών έγκειται στο γεγονός ότι το λεξικό καθώς και το μητρώο  $X$  με τους συντελεστές αραιής αναπαράστασης δύναται να υπολογιστούν ταυτόχρονα. Το πρόβλημα της αραιής αναπαράστασης και της μάθησης λεξικού μπορεί να εκφραστεί και σε μορφή μητρώων. Οι παράμετροι του μοντέλου καθορίζονται επιλύοντας το ακόλουθο πρόβλημα βελτιστοποίησης:

$$\min_{D, X} \|Y - DX\|_F \text{ s.t. } \|x_i\|_0 \leq s \quad i = 1 \dots N$$

Όπου  $\|\cdot\|_F$  είναι η νόρμα Frobenius.

Για την επίλυση του παραπάνω προβλήματος έχουν αναπτυχθεί αρκετοί αλγόριθμοι η κεντρική ιδέα των οποίων είναι ο διαχωρισμός του προβλήματος σε δύο στάδια. Το πρώτο στάδιο αφορά την εύρεση της αραιής κωδικοποίησης και το δεύτερο την ανανέωση του λεξικού. Ο πιο διαδεδομένος αλγόριθμος που αξιοποιεί την ιδέα αυτή είναι ο αλγόριθμος K-SVD [3]. Η λύση του προβλήματος προσεγγίζεται επαναληπτικά.

Έστω το σύνολο σημάτων  $Y$ , το πρώτο στάδιο του αλγορίθμου υλοποιεί την εκτίμηση της αραιής κωδικοποίησης  $x_i$ , θεωρώντας το λεξικό  $D$  γνωστό:

$$\min_{x_i} \|y_i - Dx_i\|_2 \text{ s.t. } \|x_i\|_0 \leq s \quad i = 1 \dots N$$

Το στάδιο αυτό μπορεί να υλοποιηθεί με την χρήση του αλγορίθμου OMP (Orthogonal Matching Pursuit).

Το δεύτερο στάδιο περιλαμβάνει το βήμα της ανανέωσης του λεξικού, διατηρώντας το  $X$  σταθερό. Το λεξικό  $D$  υπολογίζεται ελαχιστοποιώντας την ακόλουθη σχέση:

$$\min_D \|Y - DX\|_F$$

Το στάδιο αυτό μπορεί να υλοποιηθεί με την χρήση του αλγορίθμου K-SVD.

Στα πλαίσια της άσκησης θα μελετηθεί το πρόβλημα της αποκατάστασης εικόνας έπειτα από την προσθήκη λευκού Gaussian θορύβου.

Έστω η παραμορφωμένη εικόνα  $\Phi$  είναι διάστασης  $M \times M$ . Χρησιμοποιώντας μόνο την παραμορφωμένη εικόνα είμαστε σε θέση να ανακτήσουμε με ακρίβεια την αρχική με την χρήση της μάθησης λεξικού και της αραιής αναπαράστασης. Η εικόνα  $\Phi$  χωρίζεται

σε επικαλυπτόμενες περιοχές (overlapped patches) διάστασης ( $\sqrt{m} \times \sqrt{m}$ ). Κάθε patch το μετατρέπουμε σε διάνυσμα στήλης  $y_i \in R^m$ . Τοποθετώντας αυτά τα διανύσματα με τη σειρά καθώς σαρώνουμε την εικόνα σχηματίζεται το μητρώο  $Y$  που περιέχει όλα τα δυνατά επικαλυπτόμενα patches. Εν συνεχεία εφαρμόζουμε τον αλγόριθμο K-SVD προκειμένου να αναπαραστήσουμε κάθε διάνυσμα patch ως γινόμενο ενός λεξικού και ενός αραιού διανύσματος. Με τον τρόπο αυτό δημιουργούμε τα νέα διανύσματα, τα οποία τα μετατρέπουμε εκ νέου σε περιοχές  $\sqrt{m} \times \sqrt{m}$ . Λαμβάνοντας υπόψιν ότι κάθε pixel της εικόνας συμμετέχει σε πολλά patches ( $m$ ), η τιμή κάθε νέου pixel της τελικής εικόνας προκύπτει από το μέσο όρο των εντάσεων του pixel σε όλα τα patches που συμμετέχει.

Θεωρήστε την εικόνα **butterfly.png** και πραγματοποιήστε τα παρακάτω βήματα.

1. Υποβαθμίστε την εικόνα με λευκό Gaussian θόρυβο ώστε να έχει λόγο σήματος προς θόρυβο 20dB.
2. Τεμαχίστε την παραμορφωμένη εικόνα σε επικαλυπτόμενες περιοχές (8x8), και δημιουργήστε το μητρώο  $Y$  όπως περιεγράφηκε. Με την χρήση του αλγόριθμου OMP και K-SVD μάθετε ένα λεξικό και βρείτε την αραιή κωδικοποίηση κάθε patch που περιέχει το μητρώο  $Y$ . Για τον αλγόριθμο OMP χρήσιμη είναι η εργασία [2], ενώ για τον αλγόριθμο K-SVD η εργασία [3].
3. Υλοποιήστε τον αλγόριθμο της αποθορυβοποίησης με την χρήση λεξικών όπως αυτός περιγράφεται αναλυτικά στην εργασία [1]. Συγκρίνετε την αρχική εικόνα (χωρίς θόρυβο), την ενθόρυβη και την τελική χρησιμοποιώντας τη μετρική του PSNR (Peak Signal-to-Noise Ratio).

**Ερώτημα με Bonus:** Καθώς η υπολογιστική πολυπλοκότητα του αλγορίθμου OMP αποτελεί μια κρίσιμη σχεδιαστική παράμετρο για την εκμάθηση του λεξικού, υλοποιήστε τον αλγόριθμο batch-OMP [4] ο οποίος αποτελεί μία βελτιστοποιημένη έκδοση του OMP αλγόριθμου. Επαναλάβετε τα ερωτήματα και συγκρίνετε τους χρόνους εκτέλεσης.

#### **Βιβλιογραφία:**

- [1] M. Elad and M. Aharon, Image Denoising Via Sparse and Redundant representations over Learned Dictionaries, IEEE Trans. on Image Processing, Vol. 15, no. 12, pp. 3736-3745, December 2006.
- [2] K. Engan, S. O. Aase, and J. H. Hakon-Husoy, "Method of optimal directions for frame design," in IEEE Int. Conf. Acoust., Speech, Signal Process., 1999, vol. 5, pp. 2443–2446
- [3] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," IEEE Transactions on Signal Processing, Volume: 54 , Issue: 11 , Nov. 2006, Page(s): 4311 – 4322.

[4] Rubinstein, R., Zibulevsky, M., & Elad, M. (2008). Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. CS Technion, 40.

[5] Michael Elad. 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing (1st ed.). Springer Publishing Company Incorporated.

#### **Θέμα 4: Δημιουργία περιοχής ενδιαφέροντος (region of interest ή ROI) και κατάλληλης μάσκας σε εικόνες, για την ανίχνευση οχημάτων που κινούνται σε αυτοκινητόδρομο**

Ο τομέας της αυτόνομης οδήγησης είναι από τους ταχύτερα εξελισσόμενους στη βιομηχανία και στην έρευνα και ανάπτυξη. Ετησίως, πολλές εφαρμογές αναπτύσσονται και υλοποιούνται σε διάφορους αντίστοιχους κλάδους. Βασικό ρόλο στην αυτόνομη οδήγηση παίζουν οι αισθητήρες που βοηθούν το όχημα και τον οδηγό να αντιληφθούν το περιβάλλον στο οποίο βρίσκονται. Ένας από αυτούς είναι η οπτική κάμερα. Στα βασικά πλεονεκτήματά της περιλαμβάνονται το χαμηλό κόστος εγκατάστασης της στο όχημα και ο όγκος της πληροφορίας που παρέχει. Προκειμένου όμως η πληροφορία αυτή να καταστεί χρήσιμη, απαιτούνται πολλές και διαφορετικές διαδικασίες και τεχνικές επεξεργασίας. Έτσι, εάν ο στόχος είναι να εντοπιστούν τα οχήματα που προπορεύονται από το όχημα μας, πρωταρχικό ρόλο παίζει η δημιουργία τη περιοχής ενδιαφέροντος. Ο σχηματισμός της συμβάλλει στη σημαντική μείωση του μεγέθους του προβλήματος, εφόσον επεξεργάζεται ένα μικρό τμήμα της εικόνας, και επιπλέον αυξάνεται κατά πολύ η δυνατότητα επιτυχούς ανίχνευσης του οχήματος, αφού δεν περιλαμβάνεται πια «θόρυβος» ή άλλη άχρηστη «πληροφορία». Μετά τη δημιουργία της περιοχής ενδιαφέροντος, ακολουθεί ο σχηματισμός μίας μάσκας, μέσω της οποίας επιτυγχάνεται ο στόχος του εντοπισμού των οχημάτων του δρόμου. **Στο πλαίσιο της εργασίας αυτής, καλείστε να αναπτύξετε τεχνικές δημιουργίας της περιοχής ενδιαφέροντος και της κατάλληλης μάσκας.**



**Εικόνα 1:** Εικόνα από κίνηση σε αυτοκινητόδρομο



**Εικόνα 2:** Περιοχή ενδιαφέροντος



**Εικόνα 3:** Μάσκα για την ανίχνευση οχημάτων

### **Ερωτήματα:**

- 1) Υλοποιήστε 2 αλγορίθμους δημιουργίας περιοχής ενδιαφέροντος στην ακολουθία των frames που σας δίνεται (**april21.avi**). Αναφερθείτε στα βασικά τους σημεία.
- 2) α) Υποβαθμίστε τα frames με 2 διαφορετικά είδη θορύβου (π.χ. Gaussian και Salt & Pepper). Εφαρμόστε ξανά τους 2 αλγορίθμους. β) Αφαιρέστε τα 2 είδη θορύβου, με όποια τεχνική επιθυμείτε και εφαρμόστε και πάλι τους αλγορίθμους.

Καταγράψτε όλες τις παρατηρήσεις σας.

- 3) Με τη χρήση κατωφλίωσης ακμών και ό,τι άλλο θεωρείτε εσείς αναγκαίο, εργαστείτε στην περιοχή ενδιαφέροντος των εικόνων και υπολογίστε μία μάσκα κατάλληλη για την ανίχνευση του οχήματος. Καταγράψτε όλες τις παρατηρήσεις σας.

Για την επεξεργασία κάποιου frame, μπορείτε να χρησιμοποιήσετε χρήσιμη πληροφορία προηγούμενων από αυτό frames. Κατασκευάστε ένα video που θα περιέχει τις εικόνες που αποτελούνται μόνο από την περιοχή ενδιαφέροντος. Τα ερωτήματα θα υλοποιηθούν στο σύνολο των εικόνων από τις οποίες αποτελείται το video.

### **Βιβλιογραφία:**

[3.1] Yi, Xiang, and Bingjian Wang. "Obstacles detection method of vehicles based on image analysis." In *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1-5. IEEE, 2017.

[3.2] Yunzhou, Zhang, Sun Pengfei, Li Jifan, and Meng Lei. "Real-time vehicle detection in highway based on improved Adaboost and image segmentation." In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 2006-2011. IEEE, 2015.

[3.3] Tzomakas, Christos, and Werner von Seelen. "Vehicle detection in traffic scenes using shadows." In *Ir-Ini, Institut fur Nueroinformatik, Ruhr-Universitat*. 1998.

[3.4] Yan, Gang, Ming Yu, Yang Yu, and Longfei Fan. "Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification." *Optik* 127, no. 19 (2016): 7941-7951.

[3.5] Baek, Jang Woon, Kee-Koo Kwon, and Soo-In Lee. "Mono-camera based vehicle detection using effective candidate generation." In *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, pp. 1-2. IEEE, 2014.