

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

Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας
Ακαδημαϊκό Έτος 2020-2021 (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 στο τέλος της αναφοράς ως Παράρτημα.
3. Προτεινόμενο περιβάλλον ανάπτυξης για τις διάφορες επεξεργασίες είναι το περιβάλλον Matlab, με απαραίτητα τα toolboxes signal και image. Υλοποίηση σε άλλα περιβάλλοντα (π.χ. Python) είναι επίσης αποδεκτή.
4. Προθεσμία παράδοσης ορίζεται η 17/07/2021 12.00μμ, σε περίπτωση που επιθυμείτε να σταλεί βαθμός τον Ιούλιο. Η προθεσμία παράδοσης για τον Σεπτέμβριο θα ανακοινωθεί αργότερα.
5. Διευκρινίσεις θα δίνονται μέσω της Ενότητας “Συζητήσεις” του μαθήματος στο e-class και μέσω email στα kentrosn@gmail.com (Νικόλαος Κέντρος - Θέμα 3), st1003586@ceid.upatras.gr (Αλέξανδρος Γκίλλας – Θέμα 2), piperigkos@ceid.upatras.gr (Νικόλαος Πιπερίγκος - Θέμα 1 & Θέμα 4) και trigka@ceid.upatras.gr (Μαρία Τρίγκα).

Θέμα 1: Αξιολόγηση της απόδοσης Συνελκτικών Νευρωνικών Δικτύων κατά την ανίχνευση και κατηγοριοποίηση αντικειμένων κυκλοφοριακής σκηνής

Εισαγωγή

Τα Συνελκτικά Νευρωνικά Δίκτυα (ΣΝΝ) χρησιμοποιούνται ευρέως από την ακαδημαϊκή κοινότητα και τη βιομηχανία σε πολλές εφαρμογές, μεταξύ άλλων και για προβλήματα 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 μεγάλες ανάγκες υπολογιστικής και επεξεργαστικής ισχύος. 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 2D 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 [1.1].

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 τессάρων 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>

Ζητούμενα

1. Κατεβάστε το dataset εικόνων με το οποίο θα πειραματιστείτε από εδώ: 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. 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. 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 απόδοση 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 $IOU \geq thr$ για ανιχνευμένο αντικείμενο, τότε $TP = TP + 1$.

Εάν $IOU < thr$ για ανιχνευμένο αντικείμενο, τότε $FP = FP + 1$.

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

5. Υπολογίστε τις μετρικές Precision και Recall με βάση τον τελικό αριθμό TPs, FPs και FNs: 1) για κάθε κατώφλι και 2) για τα 2 δίκτυα. Αναφερθείτε στα βασικά σημεία των μετρικών Precision και Recall.
6. Κατασκευάστε την γραφική παράσταση Precision (-y) vs Recall (-x) και υπολογίστε το AUC με τον κανόνα του τραpezίου (έτοιμη συνάρτηση της Python). Αναφερθείτε στα βασικά σημεία του AUC και αξιολογήστε τα δύο δίκτυα με βάση τη συγκεκριμένη μετρική.

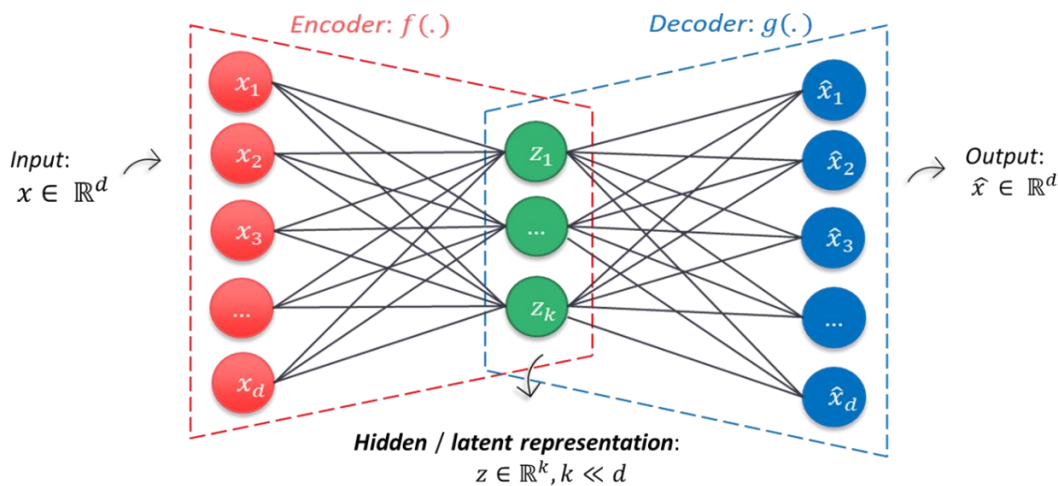
Βιβλιογραφία

[1.1] 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.

Θέμα 2: Ανάκτηση εικόνας από βάση δεδομένων

Στόχος αυτού του θέματος καθίσταται η εξοικείωση σας με τη διαδικασία της ανάκτησης εικόνων καθώς και με γραμμικές και μη γραμμικές τεχνικές μείωσης της διάστασης δεδομένων (linear and non-linear dimensionality reduction). Για τον σκοπό αυτό, σας δίνεται μια βάση εικόνων (αρχείο: **Database.rar**), που είναι ουσιαστικά ένας φάκελος με έναν αριθμό από εικόνες, από τις οποίες ορισμένες μοιάζουν μεταξύ τους και θα πρέπει να αναπτύξετε κατάλληλες ρουτίνες για την ανάκτησή τους εφαρμόζοντας τεχνικές μείωσης διάστασης δεδομένων.

Λαμβάνοντάς υπόψιν ότι οι εικόνες είναι σήματα που αναπαρίστανται σε διανυσματικούς χώρους με πολύ μεγάλη διάσταση, ο βασικός σκοπός των τεχνικών μείωσης διάστασης είναι να μειώσουν αυτόν τον πολυδιάστατο χώρο και να αναπαραστήσουν τα σήματα σε έναν χώρο μικρότερης διάστασης, επιχειρώντας να διατηρήσουν την δομή και την πληροφορία των αρχικών σημάτων. Αυτός ο χώρος δύναται να έχει πολύ μικρότερη διάσταση, ειδικά σε περιπτώσεις που υπάρχει υψηλή συσχέτιση μεταξύ των δεδομένων. Οι εν λόγω τεχνικές χωρίζονται σε δύο βασικές κατηγορίες: στις γραμμικές μεθόδους και στις μη γραμμικές μεθόδους μείωσης διάστασης.



Εικόνα 2. Δομή ενός autoencoder.

Αναλυτικότερα, στα πλαίσια της εν λόγω άσκησης καλείστε να υλοποιήσετε ένα σύστημα ανάκτησης εικόνων χρησιμοποιώντας μια γραμμική μέθοδο μείωσης διάστασης, συγκεκριμένα την Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - **PCA**) καθώς και ένα δεύτερο σύστημα ανάκτησης εικόνων που βασίζεται στους Αυτοκωδικοποιητές (**Autoencoders**), οι οποίοι αποτελούν μία μη γραμμική μέθοδο μείωσης διάστασης.

Σχετικά με την PCA, ο βασικός πυρήνας της είναι ουσιαστικά μια διάσπαση ιδιοτιμών και επιμέρους συνιστωσών που ονομάζονται ιδιο-εικόνες. Από την άλλη, οι Autoencoders είναι τεχνητά νευρωνικά δίκτυα τα οποία προβάλλουν τα δεδομένα της εισόδου σε έναν χώρο χαμηλότερης διάστασης και έπειτα ανακατασκευάζουν στην έξοδο τους την είσοδο. Σύμφωνα με την εικόνα 2, βασίζονται σε μία αρχιτεκτονική κωδικοποιητή- αποκωδικοποιητή (Encoder - Decoder), όπου ο encoder κωδικοποιεί τα δεδομένα υψηλής διάστασης σε ένα χώρο χαμηλής διάστασης (ονομάζεται latent space) και ο decoder παίρνει αυτά τα δεδομένα χαμηλής διάστασης και προσπαθεί να ανακατασκευάσει τα αρχικά πολυδιάστατα δεδομένα.

Στο φάκελο **test** δίνονται 10 εικόνες με τις οποίες θα πειραματιστείτε για τον έλεγχο των συστημάτων ανάκτησης που θα κατασκευάσετε με την χρήση της PCA και των Autoencoders. Δίνοντας μία εικόνα από τον φάκελο test ως είσοδο σε αυτό το σύστημα θα πρέπει αυτό να την εντοπίζει στην βάση δεδομένων.

Ζητούμενα:

α) **Σύστημα ανάκτησης PCA:** Κατασκευάστε ένα σύστημα ανάκτησης εικόνων μέσω της PCA. Χρησιμοποιήστε τις εικόνες στο αρχείο **Database.rar** ως training σετ ώστε να υπολογίστε την PCA των εικόνων της συγκεκριμένης βάσης. Εν συνεχεία, με βάση τις εικόνες στον test φάκελο εξάγετε ποσοστά επιτυχούς ανάκτησης. Η επιτυχής ανάκτηση αναφέρεται στην εύρεση της καλύτερης δυνατής προσέγγισης, δηλαδή της εγγύτερης στη δοθείσα βάση εικόνων. Για να εξάγετε τα ποσοστά επιτυχούς ανάκτησης πειραματιστείτε με την διάσταση του χώρου όπου προβάλλονται οι εικόνες διατηρώντας τις 100, 50 και 10 κύριες συνιστώσες για κάθε εικόνα. Είναι σημαντικό για κάθε εικόνα να αναγνωρίζονται οι κύριες συνιστώσες για λόγους υπολογιστικής πολυπλοκότητας. Σχολιάστε τα αποτελέσματα.

β) **Σύστημα ανάκτησης Autoencoders:** Κατασκευάστε ένα σύστημα ανάκτησης εικόνων με την χρήση των Autoencoders. Όπως και στο προηγούμενο ερώτημα, χρησιμοποιήστε τις εικόνες στο αρχείο **Database.rar** ώστε να εκπαιδεύσετε το δίκτυο σας και να προβάλλετε τις εικόνες της βάσης στον χώρο χαμηλότερης διάστασης μέσω του encoder δικτύου. Εν συνεχεία, χρησιμοποιήστε τις εικόνες στον test φάκελο και αφού τις προβάλλετε και αυτές στον χώρο χαμηλότερης διάστασης μέσω του encoder, εξάγετε ποσοστά επιτυχούς ανάκτησης. Για την εξαγωγή των ποσοστών επιτυχούς ανάκτησης, πειραματιστείτε όπως και πριν με την διάσταση του χώρου προβάλλοντας τις εικόνες σε ένα χώρο με διάσταση 100, 50 και 10. Ουσιαστικά θα πρέπει να εκπαιδεύσετε 3 ξεχωριστούς autoencoders, έναν για κάθε διάσταση που ζητείται.

Σημείωση: Σχετικά με την υλοποίηση του συστήματος, κατασκευάστε autoencoders οι οποίοι αποτελούνται από ένα layer encoder και ένα layer decoder. Επίσης, πειραματιστείτε και με την συνάρτηση ενεργοποίησης που θα χρησιμοποιήσετε στον encoder και στο decoder.

Σχολιάστε και συγκρίνετε τα αποτελέσματα με το προηγούμενο ερώτημα.

Βιβλιογραφία

- [2.1] Lindsay I Smith, “A tutorial on Principal Components Analysis”, Cornell University, 2002.
- [2.2] Jonathon Shlens, “A tutorial on Principal Components Analysis”, arXiv preprint, 2014.
- [2.3] Turk, M. and Pentland A., “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.
- [2.4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016, <http://www.deeplearningbook.org> .
- [2.5] Van Der Maaten, L., Postma, E. & Van den Herik, J. (2009). Dimensionality reduction: a comparative review. J Mach Learn Res, 10, 66-71.
- [2.6]. David Charte, Francisco Charte, Salvador Garcia, Maria J. del Jesus, and Francisco Herrera, “A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines,”Information Fusion, vol. 44, pp. 78–96, Nov 2018.

Θέμα 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 το μητρώο X που περιέχει τα διανύσματα x_i .

Το μαθηματικό μοντέλο για την αναπαράσταση του σήματος y_i δίνεται:

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

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 πολωνυμικό χρόνο. Για την επίλυση του έχουν αναπτυχθεί διάφορες τεχνικές οι οποίες παρουσιάζουν αξιόλογα αποτελέσματα. Η πιο διαδεδομένη κατηγορία αλγόριθμων είναι οι άπληστοι αλγόριθμοι (greedy algorithms) όπως ο αλγόριθμος OMP (Orthogonal Matching Pursuit) [3.2].

Το βασικό πλεονέκτημα του μοντέλου της μάθησης λεξικών έγκειται στο γεγονός ότι το λεξικό καθώς και το μητρώο X με τους συντελεστές αραιής αναπαράστασης δύνανται 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:

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

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 [3.3]. Η λύση του προβλήματος προσεγγίζεται επαναληπτικά.

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

$$\min_{x_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \text{ s.t. } \|\mathbf{x}_i\|_0 \leq s \quad i = 1 \dots N$$

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 \mathbf{X} σταθερό. Το λεξικό \mathbf{D} υπολογίζεται ελαχιστοποιώντας της ακόλουθη σχέση:

$$\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F$$

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 Φ είναι διάστασης $M \times M$. Χρησιμοποιώντας μόνο την παραμορφωμένη εικόνα δύναται να ανακτήσουμε με ακρίβεια την αρχική με τη χρήση της μάθησης λεξικού και της αραιής αναπαράστασης. Η εικόνα Φ χωρίζεται σε επικαλυπτόμενες περιοχές (overlapped patches) διάστασης $(\sqrt{m} \times \sqrt{m})$. Κάθε patch το μετατρέπουμε σε διάνυσμα στήλης $y_i \in \mathbb{R}^m$. Τοποθετώντας αυτά τα διανύσματα με την σειρά καθώς σαρώνουμε την εικόνα σχηματίζεται το μητρώο \mathbf{Y} που περιέχει όλα τα δυνατά επικαλυπτόμενα patches. Εν συνεχεία εφαρμόζουμε τον αλγόριθμο K-SVD προκειμένου να αναπαραστήσουμε κάθε διάνυσμα patch ως γινόμενο ενός αραιού διανύσματος και ενός λεξικού. 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 τον μέσο όρο των εντάσεων του pixel σε όλα τα patches που συμμετέχει.

Ζητούμενα

Θεωρήστε την εικόνα **barbara.jpg** 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 \mathbf{Y} όπως περιεγράφηκε. Με τη χρήση του αλγορίθμου OMP και K-SVD μάθετε ένα λεξικό και βρείτε την αραιή κωδικοποίηση κάθε patch που περιέχει το μητρώο \mathbf{Y} . Για τον αλγόριθμο OMP χρήσιμη είναι η εργασία [3.2], ενώ για τον αλγόριθμο K-SVD η εργασία [3.3].

3. Υλοποιήστε τον αλγόριθμο της αποθορυβοποίησης με την χρήση λεξικών όπως αυτός περιγράφεται αναλυτικά στην εργασία [3.1]. Συγκρίνετε την αρχική εικόνα (χωρίς θόρυβο), την ενθόρυβη και την τελική χρησιμοποιώντας την μετρική του PSNR (Peak Signal-to-Noise Ratio).

Bonus Ερώτημα: Καθώς η υπολογιστική πολυπλοκότητα του αλγορίθμου OMP αποτελεί μια κρίσιμη σχεδιαστική παράμετρο για την εκμάθηση του λεξικού, υλοποιήστε τον αλγόριθμο batch-OMP [3.4] ο οποίος αποτελεί μία βελτιωμένη έκδοση του OMP αλγορίθμου. Επαναλάβετε τα ερωτήματα και συγκρίνετε τους χρόνους εκτέλεσης.

Βιβλιογραφία

- [3.1] 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.
- [3.2] 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
- [3.3] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," IEEE Trans. Signal Process., to be published.
- [3.4] Rubinstein, R., Zibulevsky, M., & Elad, M. (2008). Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. CS Technion, 40.
- [3.5] 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 σχεδιάζονται 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. Τα μεγαλύτερα πλεονεκτήματα της είναι 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 της 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. **Στα πλαίσια της εργασίας αυτής, καλείστε να αναπτύξετε τεχνικές δημιουργίας της περιοχής ενδιαφέροντος και της κατάλληλης μάσκας.**



Εικόνα 3. Εικόνα από κίνηση σε αυτοκινητόδρομο



Εικόνα 4. Περιοχή ενδιαφέροντος



Εικόνα 5. Μάσκα για την ανίχνευση οχημάτων

Ζητούμενα

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

[4.1] 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.

[4.2] 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.

- [4.3] Tzomakas, Christos, and Werner von Seelen. "Vehicle detection in traffic scenes using shadows." In *Ir-Ini, Institut für Neuroinformatik, Ruhr-Universität*. 1998.
- [4.4] 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.
- [4.5] 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.