

Θέμα 1 (2.4 μονάδες)

Μια ψηφιακή εικόνα 8 bit έχει ιστόγραμμα στο οποίο τα επίπεδα του γκρι κατανέμονται εξίσου στην περιοχή από 160 έως 220 (ομοιόμορφη κατανομή). Για καθεμία από τις παρακάτω επεξεργασίες να σχεδιάσετε το παραγόμενο ιστόγραμμα, τη συνάρτηση μετασχηματισμού που εφαρμόζεται, και να περιγράψετε τις διαφορές μεταξύ της αρχικής εικόνας και της εικόνας εξόδου.

$L=8 \rightarrow$ επίπεδο φωτεινότητας $\in (0, 255)$, αρχικό ομοιόμορφο σε $(160, 220)$

1.1 Υπολογισμός του αρνητικού της εικόνας

συνάρτηση: $s = 255 - r$

η κατανομή αντιστρέφεται γύρω από 127.5

μ/ε τιμών: $(160, 220) \rightarrow (35, 95)$

η αρνητική εικόνα είναι πιο σκοτεινή

η εξόδου: ομοιόμορφο σε $(35, 95)$

1.2 Προσθήκη 50 (επιπέδων γκρι) στην τιμή του κάθε εικονοστοιχείου

συνάρτηση: $s = \begin{cases} r+50, & r \leq 205 \\ 255, & r > 205 \end{cases}$

το ιστόγραμμα έχει κορεσμό στο φωτεινό άκρο

στην εικόνα περιοχές που ήταν ήδη φωτεινές θα γίνουν λευκές

μ/ε τιμών: $(160, 220) \rightarrow (220, 255)$

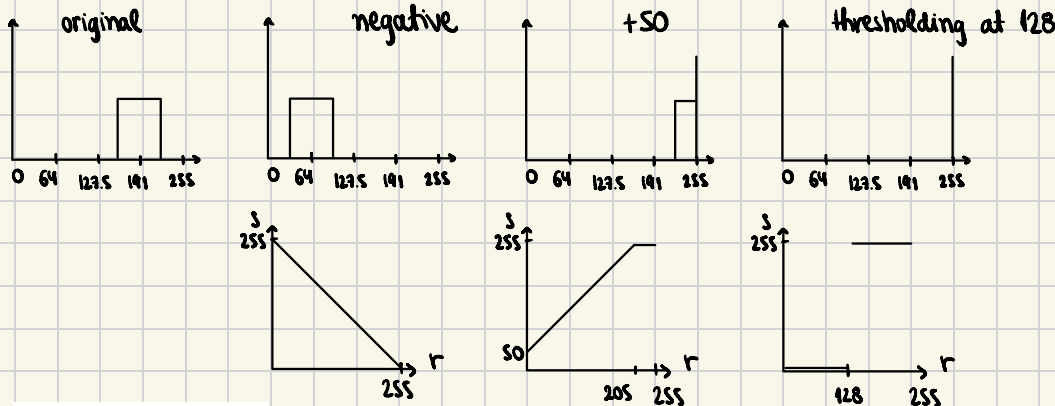
θα μοιάζει λίγο θεθωρισμένη / ξεθωριασμένη

1.3 Εφαρμογή κατωφλίωσης, όπου το κατώφλι επιλέγεται ως το επίπεδο γκρι 128

συνάρτηση: $s = \begin{cases} 0, & r \leq 128 \\ 255, & r > 128 \end{cases}$

ολόκληρη η εικόνα θα γίνει λευκή και καμία πληροφορία δεν θα διατηρηθεί

μ/ε τιμών: $(160, 220) \rightarrow 255$



Θέμα 2 (1.8 μονάδες)

Εφαρμόζοντας ένα φίλτρο στην εικόνα αριστερά, λαμβάνουμε την εικόνα δεξιά. Τι φίλτρο νομίζετε ότι εφαρμόσαμε; Δικαιολογήστε την απάντησή σας. Προτείνετε τουλάχιστον ένα τρόπο για να βελτιωθεί η δεξιά εικόνα.

Το **median filter** είναι ένα μη γραμμικό φίλτρο που αντικαθιστά την τιμή κάθε pixel με τη διάμεσο τιμή της γειτονιάς του, και χρησιμοποιείται συχνά για την αφαίρεση θορύβου τύπου salt & pepper. Ωστόσο, όταν εφαρμόζεται με μεγάλο kernel, τείνει να εξαλείφει τη λεπτομέρεια και να δημιουργεί τεχνητή εξομάλυνση στην υφή.



Η εικόνα στα δεξιά προκύπτει από την εφαρμογή ενός φίλτρου ενδιάμεσης τιμής (median filter) στην εικόνα αριστερά, πιθανότατα με μεγάλο παράθυρο.

Η υπόθεση αυτή βασίζεται στα εξής οπτικά χαρακτηριστικά της δεξιάς εικόνας:

- Η υφή της έχει αλλοιωθεί, εμφανίζοντας μεγάλες περιοχές με ομαλή φωτεινότητα και χαμηλή τοπική αντίθεση.
- Το αποτέλεσμα δεν είναι ισотροπικά θολό όπως με το Gaussian blur, αλλά φαίνεται δομικά αλλοιωμένο, όπως συμβαίνει συχνά με το median filter σε εικόνες με πολύπλοκη υφή.

πρόταση βελτίωσης:

- Μείωση μεγέθους του median φίλτρου (π.χ. σε 3x3), ώστε να διατηρηθούν οι ακμές και να περιοριστεί η υπερεξομάλυνση.
- Εναλλακτικά, χρήση τεχνικών ενίσχυσης ακμών, όπως το unsharp masking, για να αποκατασταθεί η τοπική αντίθεση.
- Ή ακόμα καλύτερα, χρήση edge-preserving φίλτρων, όπως το bilateral filter ή το guided filter, τα οποία μειώνουν τον θόρυβο διατηρώντας παράλληλα τις λεπτές υφές και τις ακμές.

Θέμα 3 (3 μονάδες)

3.1 Επιλέξτε έναν τυπικό αριθμό bits ανά εικονοστοιχείο μιας ακολουθίας εικόνων. Ποιος θα ήταν ο ρυθμός κωδικοποίησης κάθε εικονοστοιχείου όσον αφορά στη φωτεινότητα και στις χρωματικές συνιστώσες (ξεχωριστά), εάν οι βαθμοί συμπίεσης για τη φωτεινότητα και τις χρωματικές συνιστώσες είναι 10 και 40 αντίστοιχα;

B (bits/pixel): πλήθος bits ανά pixel

R (bits/pixel): ρυθμός κωδικοποίησης

C (διαστάσεις): βαθμός συμπίεσης $\sim C_y=10, C_c=40$

Τυπική εικόνα YCbCr 4:4:4, όπου:

$B_y = B_{Cb} = B_{Cr} = 8 \text{ bits/pixel}$

$\Rightarrow B_{\text{pixel}} = B_y + B_{Cb} + B_{Cr} = 8 + 8 + 8 = 24 \text{ bits/pixel}$

$R_y = B_y / C_y = 8 / 10 = 0.8 \text{ bits/pixel} \sim \text{φωτεινότητα}$

$R_c = (B_{Cb} + B_{Cr}) / C_c = 16 / 40 = 0.4 \text{ bits/pixel} \sim \text{χρωματικές συνιστώσες}$

3.2 Επιλέγουμε κωδικοποίηση JPEG των ανωτέρω εικόνων του βίντεο. Πόσα bits θα αντιστοιχούσαν, κατά μέσο όρο, σε ένα τυπικό μπλοκ φωτεινότητας;

Στη JPEG κωδικοποίηση κάθε εικόνα χωρίζεται σε block 8×8 pixels ανά κανάλι.

Κάθε 4 block περιλαμβάνει $N = 8 \times 8 = 64$ pixel και έχουμε ρυθμό κωδικοποίησης $R_y = 0.8 \text{ bits/pixel}$

Αριθμοί bits σε block φωτεινότητας: $B_{\text{block}} = R_y \cdot N = 0.8 \cdot 64 = 51.2 \text{ bits/block}$

3.3 Σε συνέχεια του ερωτήματος Β, πόσα bits θα αντιστοιχούσαν, κατά μέσο όρο, σε μία εικόνα φωτεινότητας διαστάσεων 480×720 στην κωδικοποίηση JPEG;

αριθμοί pixels εικόνας: $N_{\text{pixels}} = W \times H = 720 \cdot 480 = 345.600 \text{ pixels}$

bits εικόνας: $B_{\text{image}} = N_{\text{pixels}} \cdot R_y = 345.600 \cdot 0.8 = 276.480 \text{ bits}$

3.4 Υποθέτοντας ότι 240.000 bits είναι διαθέσιμα για να κωδικοποιήσουν μία εικόνα 4:4:4, ποια θα ήταν η μέγιστη οριζόντια ανάλυση, γνωρίζοντας ότι η κάθετη ανάλυσή της είναι 500 γραμμές;

$B_{\text{avail}} = 240.000 \text{ bits}$ διαθέσιμα για κωδικοποίηση

$R_{\text{total}} = R_y + R_c = 0.8 + 0.4 = 1.2 \text{ bits/pixel}$ για 4:4:4

Συνολικός αριθμός pixel: $N_{\text{pixels}} = W \cdot H = W \cdot 500$

Συνολικό διαθέσιμο bitrate: $B_{\text{avail}} = N_{\text{pixels}} \cdot R_{\text{total}} \Rightarrow W = \frac{B_{\text{avail}}}{R_{\text{total}} \times H} = \frac{240.000}{1.2 \times 500} = 400 \text{ pixels}$

3.5 Αν η εικόνα ήταν 4:1:1, η ανάλυση αυτή θα ήταν ίδια, μεγαλύτερη ή μικρότερη; Γιατί;

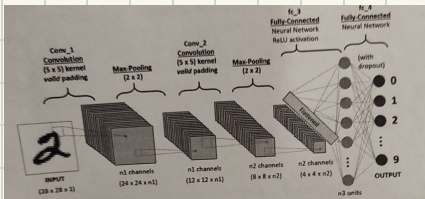
Στο 4:4:4 κάθε pixel: $B_r = B_g = B_b = 8 \text{ bits} \Rightarrow B_{\text{pixel}} = 24 \text{ bits}$

Στο 4:1:1 κάθε ομάδα 4 pixel: $4B_r + 1B_g + 1B_b = 4 \cdot 8 + 1 \cdot 8 + 1 \cdot 8 = 48 \text{ bits per 4 pixels}$
κάθε pixel: $48/4 = 12 \text{ bits} \Rightarrow B_{\text{pixel}} = 12 \text{ bits}$

Κάθε pixel απαιτεί λιγότερα bits (50% λιγότερα) $\Rightarrow$ μεγαλύτερη ανάλυση

Θέμα 4 (3.3 μονάδες)

4.1 Στο σχήμα φαίνεται ένα βαθύ νευρωνικό δίκτυο. Να εξηγήσετε τη λειτουργία του.



$\sim$ Συνελκτικό νευρωνικό δίκτυο - CNN

input: ασπρόμαυρη εικόνα $28 \times 28 \times 1$

output: ταξινόμηση σε 10 classes (0-9)

- 1 Input: εικόνα $28 \times 28 \times 1$ πιθανότατα χειρόγραφοι αριθμοί
 - 2 Conv-1: συνελίξη με kernel 5×5 , χωρίς padding (valid padding)
 $\sim$ χωρική διάσταση: $28 - 5 + 1 = 24 \Rightarrow 24 \times 24 \times n_1$ feature maps
 - 3 Max-Pooling (2×2): μειώνει χωρική διάσταση κατά 2, σε κάθε 2×2 block κρατάει max
 $\sim$ χωρική διάσταση: $24/2 = 12 \Rightarrow 12 \times 12 \times n_1$ feature maps
 - 4 Conv-2: συνελίξη με kernel 5×5 , χωρίς padding (valid padding)
 $\sim$ χωρική διάσταση: $12 - 5 + 1 = 8 \Rightarrow 8 \times 8 \times n_2$ feature maps
 - 5 Max-Pooling (2×2): μειώνει χωρική διάσταση κατά 2, σε κάθε 2×2 block κρατάει max
 $\sim$ χωρική διάσταση: $8/2 = 4 \Rightarrow 4 \times 4 \times n_2$ feature maps
 - 6 Flattening: $4 \times 4 \times n_2$ feature maps $\sim$ vector με length $n_3 = 16n_2$
 - 7 Fully-Connected Layer (ReLU): οι n_3 μονάδες εισόδου συνδέονται με όλους του νευρώνες του hidden layer
 $\rightarrow$ χρησιμοποιείται ReLU για μη γραμμικότητα
 - 8 Fully-Connected layer (με dropout): ανενεργονοί τυχαίο ποσοστό μονάδων κατά την εκπαίδευση
 $\rightarrow$ σκοπός: αποφυγή overfitting
 - 9 Output layer: περιέχει 10 μονάδες εξόδου, πιθανή ενεργοποίηση με soft-max, ώστε να παράξει πιθανότητες
- Εντοπισμός χαρακτηριστικών μέσω συνελίξεων
► Μείωση διαστάσεων με pooling
► Εκμάθηση μη γραμμικών σχέσεων στα fully connected layers

4.2 Στα συνελκτικά δίκτυα η χρήση 'parameter sharing' επιτρέπει τα εξής:

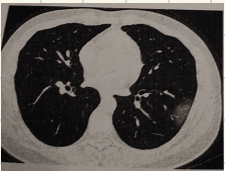
- μείωση του συνολικού αριθμού των άγνωστων παραμέτρων
- εφαρμογή ανίχνευσης features σε πολλαπλές θέσεις των εικόνων εισόδου
- το sharing των παραμέτρων που υπολογίστηκαν για μια κατηγορία εικόνων και σε άλλες διαφορετικές κατηγορίες (transfer learning)

1 Μείωση παραμέτρων: Στα CNNs, αντί κάθε φίλτρο να έχει ξεχωριστό βάρος για κάθε pixel εισόδου, χρησιμοποιούμε το ίδιο σύνολο βαρών (kernel) σε όλη την εικόνα. Αυτό μειώνει δραματικά τον αριθμό των παραμέτρων, π.χ. από εκατομμύρια (σε πλήρως συνδεδεμένα) σε λίγες χιλιάδες.

2 Ανίχνευση features σε πολλές θέσεις: Επειδή το kernel "σαρώνει" την εικόνα, μπορεί να ανιχνεύσει το ίδιο pattern (π.χ. μια ακμή) ανεξάρτητα από το πού βρίσκεται – κάτι κρίσιμο για εικόνες όπου τα αντικείμενα εμφανίζονται σε διαφορετικές θέσεις.

3 Μεταφορά μάθησης (Transfer Learning): Εφόσον τα φίλτρα μαθαίνουν γενικά patterns (π.χ. γραμμές, υφές), μπορούν να επαναχρησιμοποιηθούν και σε άλλες εφαρμογές – ακόμη και σε διαφορετικά dataset. Αυτό κάνει το transfer learning πολύ αποτελεσματικό.

4.3 Έχετε στη διάθεσή σας αξονικές τομογραφίες των πνευμόνων υγιών ατόμων και ασθενών (όπως η εικόνα παρακάτω), και πρέπει να ανιχνεύσετε αν δείχνουν την ύπαρξη COVID-19. Περιγράψτε σύντομα πώς θα σχεδιάζατε ένα σύστημα που θα μάθαινε να υλοποιεί αυτόματα αυτή την ανίχνευση.



Για την αυτόματη ανίχνευση COVID-19 από αξονικές τομογραφίες πνευμόνων, θα σχεδιάζα ένα σύστημα βασισμένο στη μηχανική μάθηση, και συγκεκριμένα σε συνελκτικά νευρωνικά δίκτυα (CNNs), τα οποία είναι κατάλληλα για ταξινόμηση εικόνων και ανίχνευση μορφολογικών χαρακτηριστικών.

◆ Βήματα σχεδίασης του συστήματος:

1. Συλλογή και labeling δεδομένων: Δημιουργούμε σύνολο δεδομένων με εικόνες CT από υγιή άτομα και ασθενείς με COVID-19, συνοδευόμενες από κατάλληλες ετικέτες (labels).
2. Προεπεξεργασία εικόνων: Κανονικοποίηση pixel values, Ανακατασκευή ή resize σε κοινές διαστάσεις, Ενδεχομένως εξαγωγή περιοχής ενδιαφέροντος (ROI), Εφαρμογή data augmentation (π.χ. περιστροφή, flip), για αύξηση γενίκευσης
3. Επιλογή και εκπαίδευση μοντέλου: Χρήση αρχιτεκτονικών CNN (ή 3D-CNN εάν επεξεργαζόμαστε όγκους εικόνων αντί για μεμονωμένα slices), Παράδειγμα μοντέλων: VGG, ResNet, DenseNet, Εφαρμογή Transfer Learning με fine-tuning προεκπαιδευμένων μοντέλων (π.χ. ImageNet)
4. Αξιολόγηση του συστήματος: Χρησιμοποιούμε κατάλληλες μετρικές, όπως: accuracy, binary cross-entropy (ως loss function), precision, recall, F1-score, AUC, ώστε να διασφαλίσουμε την διαγνωστική ακρίβεια του μοντέλου