

# Ψηφιακή Επεξεργασία Εικόνας -Εργασία 2-

## Οπτική αναγνώριση χαρακτήρων

A. Ντελόπουλος

Άνοιξη 2023

### Εισαγωγή

Στην παρούσα εργασία θα ασχοληθείτε με την αυτόματη, οπτική αναγνώριση χαρακτήρων. Συγκεκριμένα, σας δίνεται μία εικόνα η οποία περιέχει κείμενο και θα πρέπει να βρείτε τους χαρακτήρες που το αποτελούν. Τα βασικά βήματα είναι (α) η προεπεξεργασία της εικόνας ώστε να είναι κατάλληλη για επεξεργασία και να εντοπισθούν λέξεις, (β) η περιγραφή των χαρακτήρων με χρήση του διακριτού μετασχηματισμού Fourier, και (γ) η “ανάγνωση” της φωτογραφίας, εκπαιδεύοντας και αξιολογώντας ένα σύστημα ταξινόμησης.

### 1 Προεπεξεργασία και εντοπισμός λέξεων

Στο πρώτο ζητούμενο, θα πρέπει να προεπεξεργαστείτε την εικόνα ώστε να αποκόψετε χωρία στα οποία θα αναζητήσετε τον κάθε χαρακτήρα. Το πρώτο βήμα είναι η αναίρεση της περιστροφής που έχει υποστεί μια εικόνα κειμένου.

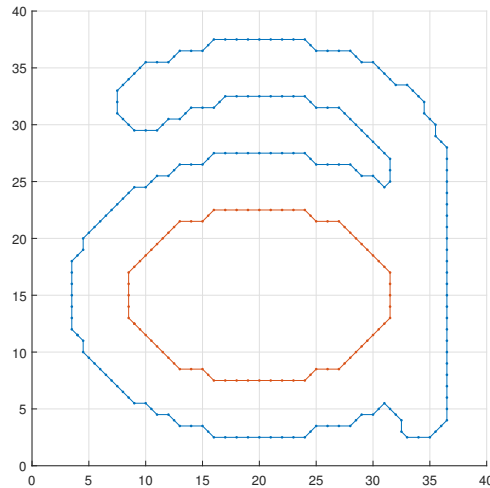
Αρχικά, θα πρέπει να χρησιμοποιήσετε τον διδιάστατο διακριτό μετασχηματισμό Fourier (DFT), αφού πρώτα θολώσετε κατάλληλα την εικόνα, ώστε να “συνενωθούν” τα γράμματα κάθε γραμμής. Παρατηρώντας το κατάλληλο τμήμα του μέτρου του DFT (ή του λογαρίθμου του μέτρου) μπορείτε να εντοπίσετε την μέγιστη συχνότητα μεταβολής που αντιστοιχεί στην μεταβολή της φωτεινότητας από γραμμή (κειμένου) σε γραμμή. Με αυτόν τον τρόπο θα έχετε μία πρώτη εκτίμηση της γωνίας περιστροφής, ωστόσο, η μέθοδος αυτή δεν είναι απόλυτα ακριβής, κυρίως λόγω της περιορισμένης ανάλυσης του DFT.

Στη συνέχεια, για να εντοπίσετε ακριβώς την γωνία, θα πρέπει να κάνετε μία σειριακή αναζήτηση γύρω από την πρώτη εκτίμηση της γωνίας. Σε κάθε υποψήφια γωνία, θα αναίρεσετε την περιστροφή, και θα υπολογίσετε την προβολή της φωτεινότητας στον κατακόρυφο άξονα. Κριτήριο επιλογής θα πρέπει να είναι ότι όσο πιο καλή είναι η προσέγγιση της πραγματικής γωνίας, τόσο μεγαλύτερες θα είναι και οι μεταβολές της φωτεινότητας στην προβολή.

Υλοποιήστε τη συνάρτηση

```
angle = findRotationAngle(x)
```

η οποία δέχεται ως είσοδο την εικόνα  $x$  (η οποία θα είναι η εικόνα `text.png` που σας δίνεται) και επιστέφει την εκτίμηση της γωνίας περιστροφής `angle` σύμφωνα με τη μεθοδολογία που περιγράφεται παραπάνω.



Σχήμα 1: Παράδειγμα περιγράμματος για το γράμμα “a”. Υπάρχουν δύο περιγράμματα, ένα εσωτερικό και ένα εξωτερικό.

Επιπλέον να υλοποιήστε τη συνάρτηση

```
y = rotateImage(x, angle)
```

η οποία δέχεται ως είσοδο την εικόνα  $x$  και τη γωνία  $angle$  και επιστέφει την εικόνα  $y$  αφού έχει εφαρμόσει στην εικόνα εισόδου περιστροφή ίση με τη γωνία εισόδου και έχει συμπληρώσει με κατάλληλο padding τα κενά.

## 2 Περιγραφή περιγράμματος

### 2.1 Εντοπισμός περιγράμματος

Για να μπορέσετε να κάνετε την αναγνώριση του κάθε χαρακτήρα, θα πρέπει να υπάρχει μία περιγραφή η οποία θα είναι κατάλληλη. Αρχικά θα πρέπει να εντοπίσετε το περίγραμμα του κάθε γράμματος. Τα γράμματα που θα εντοπίσετε σας δίνονται σε υψηλή ανάλυση σε χωριστή εικόνα. Ένα παράδειγμα περιγράμματος δίνεται στο Σχήμα 1. Το περίγραμμα θα το αναγνωρίσετε με κατάλληλους μορφολογικούς μετασχηματισμούς, και συγκεκριμένα υπολογίζοντας την *dilated* εκδοχή της εικόνας και αφαιρώντας από αυτή την αρχική. Στην συνέχεια πρέπει να εφαρμόσετε και *thinning* ώστε να μειώσετε το πάχος των ακμών. Τέλος, θα πρέπει να εντοπίσετε ένα ή περισσότερα περιγράμματα για το κάθε γράμμα.

Για το δεύτερο ζητούμενο, υλοποιήστε τη συνάρτηση

```
c = getcontour(x)
```

η οποία δέχεται ως είσοδο μία εικόνα  $x$  η οποία περιέχει ένα γράμμα και εντοπίζει τα περιγράμματά της. Επιστρέφει το  $c$  το οποίο είναι ένα cell array, με ένα cell για κάθε περίγραμμα που εντοπίσατε. Κάθε cell περιέχει έναν πίνακα διαστάσεων  $N \times 2$  όπου  $N$  είναι ο αριθμός των σημείων που περιγράφουν το περίγραμμα, και κάθε γραμμή του πίνακα είναι οι συντεταγμένες του αντίστοιχου σημείου. Σημειώστε ότι για τα γράμματα που σας δίνονται, υπάρχουν πάντα ένα ή δύο περιγράμματα.

Παρουσιάστε (στην αναφορά) τα περιγράμματα για τα γράμματα “a”, “f”, “l” και “e”.

## 2.2 Αναπαράσταση και σύγκριση περιγραμμάτων

Στο τρίτο ζητούμενο της εργασίας, θα εφαρμόσετε τα παραπάνω ώστε να αναγνωρίσετε τους χαρακτήρες του κειμένου που σας δίνεται. Αρχικά θα πρέπει να υλοποιήσετε τον ακόλουθο περιγραφέα περιγράμματος. Έστω ένα περίγραμμα (θυμηθείτε ότι ένα γράμμα μπορεί να έχει ένα, δύο ή τρία περιγράμματα) του οποίου τα σημεία είναι τα  $x[i], y[i]$ ,  $i = 0, 1, \dots, n-1$ , όπου  $n$  είναι ο αριθμός των σημείων τα οποία περιγράφουν το περίγραμμα. Αρχικά ορίζουμε την μιγαδική ακολουθία

$$r[i] = x[i] + jy[i] \quad (1)$$

όπου  $j = \sqrt{-1}$ . Στη συνέχεια, υπολογίζουμε τον DFT της ακολουθίας, έστω  $R[i]$ . Ο περιγραφέας του περιγράμματος είναι το μέτρο της  $R[i]$ ,  $|R[i]|$ , εξαιρώντας από αυτή τον πρώτο όρο, δηλαδή για  $i = 1, 2, \dots, n-1$ .

Το κριτήριο απόστασης για δύο τροχιές  $r_1$  και  $r_2$  είναι το τετραγωνικό σφάλμα των περιγραφέντων τους  $|R_1|$  και  $|R_2|$ , δηλαδή

$$d(|R_1|, |R_2|) = \sum_{i=1}^{n-1} (|R_1[i]| - |R_2[i]|)^2 \quad (2)$$

Όσο μικρότερο είναι, τόσο περισσότερο μοιάζουν. Παρατηρήστε ότι οι περιγραφείς  $|R_1|$  και  $|R_2|$  πρέπει να έχουν το ίδιο μήκος.

## 3 Αναγνώση χαρακτήρων - Σύστημα ταξινόμησης

Για να υλοποιήσετε και να επικυρώσετε την αυτόματη ανάγνωση χαρακτήρων, πρέπει να εκπαιδεύσετε ένα σύστημα ταξινόμησης χαρακτήρων με βάση το κείμενο αναφοράς που σας δίνεται (text1.png - text1.txt).

Για τη δημιουργία του dataset πρέπει να εντοπίσετε ένα ένα τα γράμματα της εικόνας κειμένου και να τα αντιστοιχίσετε με τους χαρακτήρες ascii. Πρώτα θα πρέπει να αναιρέσετε την περιστροφή της εικόνας, και να χρησιμοποιήσετε την κατακόρυφη προβολή της φωτεινότητας για να εντοπίσετε και να ξεχωρίσετε τις γραμμές κειμένου μεταξύ τους. Στη συνέχεια, για κάθε γραμμή κειμένου, χρησιμοποιήστε την προβολή της φωτεινότητας στον οριζόντιο άξονα ώστε να ξεχωρίσετε τις λέξεις. Επειτα, θα πρέπει να ξεχωρίσετε τα γράμματα της κάθε λέξης, με μία απλή επαναληπτική διαδικασία. Για κάθε χαρακτήρα του κειμένου, εφαρμόστε αρχικά (πριν από την ανίχνευση των περιγραμμάτων) το μορφολογικό τελεστή opening ώστε να εξομαλύνετε μικρά “σκουπίδια” που μπορεί να υπάρχουν στην περιοχή του γράμματος.

Χωρίστε το dataset που δημιουργήσατε σε σύνολο εκπαίδευσης (train set) και δοκιμής (test set), χρησιμοποιώντας διαχωρισμό 70:30 ώστε να εκπροσωπείται κάθε χαρακτήρας αντίστοιχα.

Με βάση τον αριθμό των περιγραμμάτων που έχει κάθε χαρακτήρας, χωρίζεται αρχικά σε μία από τις τρεις δυνατές τάξεις χαρακτήρων (π.χ. I: 1ης τάξης, b: 2ης τάξης, B: 3ης τάξης). Για κάθε μια τάξη χαρακτήρων εκπαιδεύστε ξεχωριστά έναν ταξινομητή χαρακτήρων χρησιμοποιώντας τον αλγόριθμο μηχανικής μάθησης K-κοντινότεροι γείτονες (KNN) με βάση την αναπαράσταση των περιγραμμάτων του.

Αξιολογήστε την απόδοση του ταξινομητή σας χρησιμοποιώντας το σύνολο δοκιμών. Συγκεκριμένα υπολογίστε τον πίνακα σύγχυσης (confusion matrix), ο οποίος δείχνει πόσα δείγματα ταξινομήθηκαν σωστά και λανθασμένα για κάθε κλάση. Επιπλέον, υπολογίστε

τη σταθμισμένη ακρίβεια (weighted accuracy), η οποία είναι η μέση ακρίβεια κάθε κλάσης σταθμισμένη με τον αριθμό των δειγμάτων της συγκεκριμένης κλάσης.

Για να μπορέσετε να υπολογίσετε την απόσταση μεταξύ δύο περιγραμμάτων πρέπει αυτά να έχουν ίδιο μήκος, δηλαδή ίδιο αριθμό σημείων. Ο απλούστερος τρόπος για να το πετύχετε είναι να κάνετε κάποιου είδους παρεμβολή (πχ γραμμική) στις μιγαδικές ακολουθίες, έτσι ώστε οι ακολουθίες  $r_1$  και  $r_2$  που ανήκουν στην τάξη  $i$  να έχουν το ίδιο μήκος  $N_i$ . Για την παρεμβολή προτείνεται η χρήση της συνάρτησης `interp1`. Δοκιμάστε διαφορετικές τιμές για τα  $N_1, N_2, N_3$  και σχολιάστε τις παρατηρήσεις σας.

Για τους χαρακτήρες 2ης και 3ης θα χρειαστεί να αντιστοιχίσετε πρώτα τα περιγράμματα, π.χ. για τους 2ης τάξης "μέσα"/"έξω" και για τους 3ης τάξης "έξω"/"μέσα πάνω"/"μέσα κάτω".

Αναλύστε τα αποτελέσματα και συζητήστε πιθανούς τρόπους βελτίωσης της απόδοσης του συστήματος OCR.

Τέλος, με βάση το εκπαιδευμένο σύστημα υλοποιήστε την συνάρτηση

```
lines = readtext(x)
```

όπου  $x$  είναι μία εικόνα κειμένου, και `text` είναι ένα cell array που κάθε στοιχείο του περιέχει το κείμενο που βρήκατε από την αντίστοιχη γραμμή κειμένου. Προτείνεται να μην υλοποιήσετε τα πάντα μέσα σε αυτή τη συνάρτηση, αλλά να δομήσετε τον κώδικά σας σε μικρές συναρτήσεις με συγκεκριμένη λειτουργικότητα, τις οποίες θα χρησιμοποιήσετε στην `readtext` ώστε να να είναι πιο κατανοητός ο κώδικάς σας.

Εφαρμόστε την παραπάνω συνάρτηση στην εικόνα `text2.png` και υπολογίστε το αντίστοιχο confusion matrix και weighted accuracy, σύμφωνα με την αποκωδικοποίηση `ascii text2.txt`. Η εργασία με το μεγαλύτερο weighted accuracy θα λάβει bonus 1 μονάδα.

## Για την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Η αναφορά θα πρέπει να επιδεικνύει την ορθή λειτουργία του κώδικά σας στις εικόνες που σας δίνονται.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλίσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για οποιαδήποτε είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για τις εικόνες που σας δίνονται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα, καθώς και να τηρούνται τα ακόλουθα:

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM.zip`, όπου AEM είναι τα τέσσερα ψηφία του A.E.M. του φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα σε MATLAB ή Python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Εάν η εργασία υλοποιηθεί σε Python θα πρέπει μαζί με τα αρχεία κώδικα να συμπεριλάβετε και το αρχείο `environment.yml` με τα πακέτα που έχετε χρησιμοποιήσει.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.

- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε τις εικόνες που σας δίνονται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ “Thumbs.db”, “.DS\_Store”, “.directory”.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “\$”, “%” κλπ.