

Άσκηση 5

Εφαρμογές της Ψηφιακής Επεξεργασίας Σημάτων

Περιεχόμενα

SVD.....	2
Διαδικασία	2
1).....	2
2).....	3
α).....	3
β)	6
3).....	6
4).....	7
10 singular values	7
3 singular values.....	9
PCA.....	11
Διαδικασία	11
2).....	11
3).....	12
4).....	14
10 components	14
3 components.....	14
Eigenfaces.....	14
Διαδικασία	14
1).....	14
2).....	15
3).....	16
4).....	17

Ο κώδικας μπορεί να βρεθεί εδώ:

<https://github.com/GrigorisTzortzakis/Applications-of-Digital-Signal-Processing/tree/main/Exercise%205>

SVD

Διαδικασία

1)

Αρχικά, θέλουμε να δούμε τις διαστάσεις των 3 βίντεο και μετά να εκτελέσουμε SVD. Κάθε ένα από αυτά φορτώθηκε στη MATLAB ώστε να ανακτηθεί η τρισδιάστατη μεταβλητή video, η οποία έχει διαστάσεις ύψους, πλάτους και αριθμό καρέ. Στο αρχείο bootstrap η μεταβλητή video είναι διαστάσεων 120×160×1500, στο campus είναι 128×160×1430 και στο lobby.mat 128×160×800.

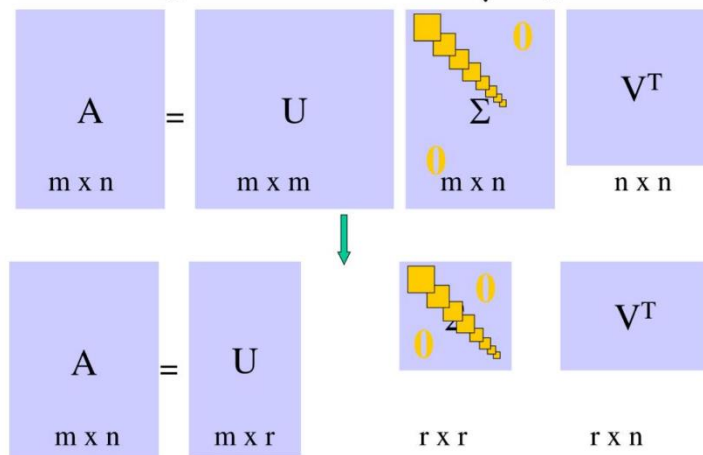
Στη συνέχεια, τα τρισδιάστατα δεδομένα κάθε βίντεο μετασχηματίστηκαν σε δισδιάστατο πίνακα δεδομένων X διαστάσεων $T \times N$, όπου T είναι το πλήθος των καρέ και $N = H \times W$ ο συνολικός αριθμός pixel ανά καρέ. Ο μετασχηματισμός έγινε με τη λειτουργία `reshape(video, H*W, T)`, ώστε κάθε σειρά του X να περιέχει ένα διάνυσμα μήκους N που αντιστοιχεί στη διανυσματοποίηση ενός frame. Τέλος, χρησιμοποιούμε την έτοιμη εντολή για τον υπολογισμό της `svd`.

```
=== Processing bootstrap.mat ===
Original video: 120 × 160 × 1500 → X is 1500 × 19200
U size: 19200 × 19200 (should be N × N = 19200 × 19200)
Sigma size: 19200 × 1500 (should be N × T = 19200 × 1500)
V size: 1500 × 1500 (should be T × T = 1500 × 1500)

=== Processing campus.mat ===
Original video: 128 × 160 × 1430 → X is 1430 × 20480
U size: 20480 × 20480 (should be N × N = 20480 × 20480)
Sigma size: 20480 × 1430 (should be N × T = 20480 × 1430)
V size: 1430 × 1430 (should be T × T = 1430 × 1430)

=== Processing lobby.mat ===
Original video: 128 × 160 × 800 → X is 800 × 20480
U size: 20480 × 20480 (should be N × N = 20480 × 20480)
Sigma size: 20480 × 800 (should be N × T = 20480 × 800)
V size: 800 × 800 (should be T × T = 800 × 800)
```

The Singular Value Decomposition



<https://www.slideserve.com/race/lecture-19-singular-value-decomposition>

<https://www.mathworks.com/help/matlab/ref/double.svd.html>

2)

Πριν εφαρμόσουμε την μέθοδο που ζητείτε, παρατηρούμε τα αρχικά βίντεο. Το bootstrap περιέχει κόσμο να παίρνει φαγητό σε δίσκους, το campus είναι ένα πάρκο όπου έχει άνεμο, κουνιούνται οι φυλλωσιές και περνάνε μερικά άτομα και οχήματα και το lobby είναι ένα γραφείο όπου μετα από κάποιο χρόνο περνάνε 2 άτομα και σβήνουν τα φωτά.



α)

Bootstrap:

Παρατηρούμε ότι πλέον τα άτομα έχουν γίνει ένα «blur» και εμφανίζονται σαν απλές σκιές που κινούνται. Επιπλέον, αντί να αλλάζουν τοποθεσίες, πολλά άτομα παραμένουν στο ίδιο σημείο και έχουν ελάχιστη μετακίνηση. Το αξιοσημείωτο είναι ότι τα άτομα με λευκή μπλούζα πρακτικά δεν εμφανίζονται στο βίντεο, έχουν παραμείνει μόνο άτομα με σκούρα ρούχα.

Τέλος, οι δίσκοι που υπήρχαν πλέον δεν εμφανίζονται καθόλου και το βίντεο φαίνεται σαν να είναι σε fast forward.



Campus:

Το βίντεο φαίνεται σαν να είναι σε slow motion, καθώς η κίνηση των δέντρων από τον άνεμο είναι πολύ πιο αργή και σε αρκετά μικρότερο βαθμό σε σχέση με το αρχικό βίντεο. Τα αυτοκίνητα που περνάνε εμφανίζονται στιγμιαία ως ένα «blur» και το αξιοσημείωτο της υπόθεσης είναι ότι οι άνθρωποι που εμφανίζονταν έχουν εξαφανίσει πλήρως από το βίντεο.



Lobby:

Τα άτομα εμφανίζονται και πάλι ως ένα στιγμιαίο «blur». Τα φωτά που κλείνουν εμφανίζονται με τον ίδιο τρόπο με το αρχικό βίντεο, ενώ το αξιοσημείωτο είναι ότι το άτομο με την λευκή μπλούζα εμφανίζεται για πάρα πολύ λίγο, ενώ το άτομο με την μαύρη μπλούζα υπάρχει για περισσότερα frame μέσα στο βίντεο.



Συμπερασματικά, όταν ανακατασκευάζουμε το βίντεο χρησιμοποιώντας μόνο τις δέκα κορυφαίες ιδιοτιμές, ουσιαστικά απορρίπτονται όλες οι λεπτομέρειες που αλλάζουν γρήγορα σε κάθε καρέ και διατηρούνται μόνο οι κυρίαρχες, αργές δομές. Στην πράξη, αυτό σημαίνει ότι τα στατικά στοιχεία, όπως τοίχοι, δάπεδα και άλλα χαρακτηριστικά του φόντου, παραμένουν ίδια. Όλα τα υπόλοιπα θεωρούνται θόρυβος και αφαιρούνται από τη νέα εκδοχή του βίντεο.

Ως αποτέλεσμα, οι φιγούρες των ανθρώπων δεν εμφανίζονται πλέον ως διακριτά σώματα αλλά ως ασαφείς, θαμπές σκιές. Οι ακριβείς μορφές τους και οι έντονες αντιθέσεις στα ρούχα τους εξαφανίζονται, γιατί κατανέμονται σε πολλές ανώτερες ιδιοτιμές που έχουν μηδενιστεί. Μικρά αντικείμενα, όπως οι δίσκοι, εξαφανίζονται επειδή καταλαμβάνουν ελάχιστα pixels και η κίνησή τους δεν ακολουθεί το κυρίαρχο μοτίβο του φόντου. Ομοίως, τα αυτοκίνητα που περνούν γρήγορα μέσα στο πλάνο εμφανίζονται ως θολές γραμμές αντί για καθαρές, συμπαγείς μορφές. Τέλος, στην πραγματικότητα δεν έχει αλλάξει η ταχύτητα του βίντεο, απλά δημιουργείτε η ψευδαίσθηση λόγω των παραμορφώσεων.

β)

Είδαμε παραπάνω ότι το κάθε βίντεο επηρεάζεται με τον δικό του τρόπο, αφού στο πρώτο εξαφανίζονται οι άνθρωποι με λευκά ρούχα, στο δεύτερο δεν υπάρχουν καθόλου και στο τρίτο υπάρχουν αλλά για περιορισμένο χρονικό διάστημα. Αυτό που είναι ίδιο σε όλα είναι ότι θολώνονται οι λεπτομέρειες και εμφανίζονται μόνο για ένα περιορισμένο χρονικό διάστημα ενώ διατηρούνται σταθερά όλα τα κύρια χαρακτηριστικά, όπως το φόντο και γενικά ότι είναι ακίνητο.

Αναλύοντας, όταν ανακατασκευάζετε κάθε σκηνή χρησιμοποιώντας μόνο τις πρώτες δέκα ιδιοτιμές, διατηρείτε ουσιαστικά ότι σε κάθε βίντεο έχει τη μεγαλύτερη «ενέργεια». Στο `boosttrap.mat`, το φόντο και οι άνθρωποι με σκούρα ρούχα δημιουργούν υψηλή διακύμανση, οπότε παραμένουν ευκρινή, ενώ οι άνθρωποι με άσπρα ρούχα έχουν παρόμοιο χρώμα με το `background`, άρα εξαφανίζονται.

Στο `campus.mat`, τα κτίρια μαζί με τα δέντρα δημιουργούν ισχυρά μοτίβα που επιβιώνουν την περικοπή, αλλά τα γρήγορα αυτοκίνητα που διασχίζουν το πλάνο μειώνονται σε στιγμιαίες θολούρες ή εξαφανίζονται τελείως. Επιπροσθέτως, παρόλο που ο πεζός παραμένει στην οθόνη για πολλά καρέ, η εμφάνισή του είναι πολύ λιγότερο «συνεκτική» σε σχέση με τα αυτοκίνητα. Ένα όχημα που περνάει διατηρεί κατά γενικό κανόνα σταθερό σχήμα, μέγεθος, χρώμα και τροχιά, έτσι, παρόλο που εμφανίζεται για μικρό χρονικό διάστημα, κάθε του `frame` ευθυγραμμίζεται χωρικά και φασματικά με τα υπόλοιπα. Αντίθετα, ένας άνθρωπος που περπατάει αλλάζει συνεχώς στάση, σιλουέτα και ακόμη και φαινομενική φωτεινότητα (σκιές, προσανατολισμός σώματος).

Στο `lobby.mat`, είναι σημαντικό να παρατηρήσουμε το χρώμα του `background`, το οποίο είναι ανοιχτόχρωμο. Για αυτόν ακριβώς τον λόγο εμφανίζεται παραπάνω ο άνθρωπος με τα σκούρα ρούχα, αφού δημιουργεί μεγάλη αντίθεση και άρα παραμένει ακόμα και με 10 συνιστώσες. Ο λόγος που δεν εξαφανίζεται εντελώς ο άνθρωπος με ανοιχτόχρωμα ρούχα, όπως στο πρώτο βίντεο, είναι επειδή δημιουργεί επίσης ένα τοπικό, υψηλής αντίθεσης «στίγμα» σε σχέση με το σκούρο `background` του χώρου όταν κλείνουν τα φώτα κατά την διάρκεια του βίντεο. Στο πρώτο βίντεο ο φωτισμός είναι σταθερός για όλη του την διάρκεια.

3)

Κρατώντας μόνο τις 3 πρώτες ιδιάζουσες τιμές, χάνουμε μεγάλο ποσοστό της ενέργειας (έχουμε μόνο περίπου 30% της αρχικής ενέργειας). Ως αποτέλεσμα, στο πρώτο βίντεο εξαφανίζονται σχεδόν πλήρως οι άνθρωποι και μπορούμε ελάχιστα να διακρίνουμε την σιλουέτα τους, ενώ ταυτόχρονα μερικοί από αυτούς εμφανίζονται σαν «φαντάσματα» (μια άσπρη σιλουέτα). Στο δεύτερο

βίντεο, χάνονται πλήρως τα αυτοκίνητα και οι άνθρωποι. Επιπλέον, η κίνηση των δέντρων από τον άνεμο δεν είναι εμφανής πάρα μόνο σε ένα σημείο όπου κινούνται στιγμιαία, όταν πλέον η μεταβολή τους είναι πολύ μεγάλη. Τέλος, στο τρίτο βίντεο δεν εμφανίζονται καθόλου οι άνθρωποι και το μόνο που έχει διατηρηθεί είναι το κλείσιμο των φωτών προς το τέλος.

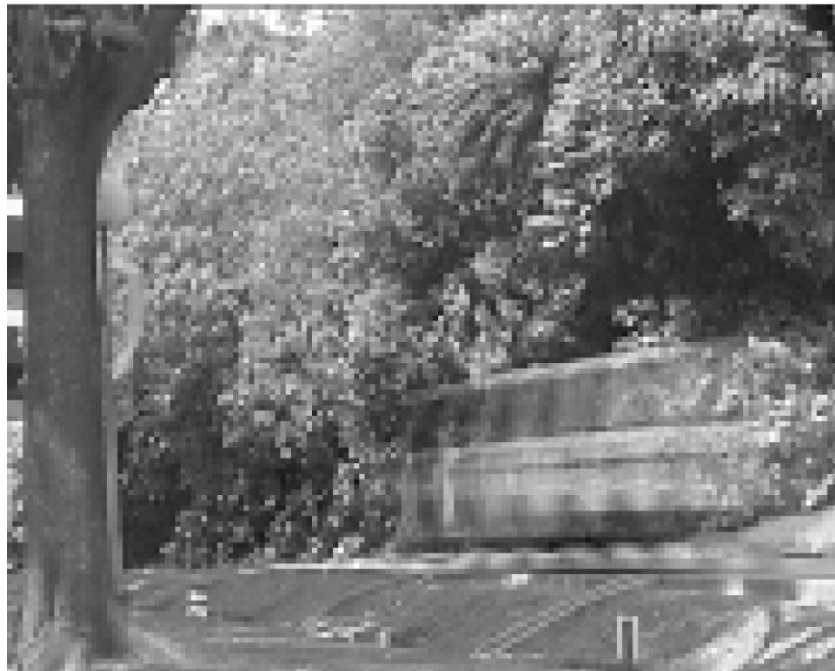
4)

10 singular values

Τα αποτελέσματα είναι πολύ καλύτερης ποιότητας σε σχέση με το ερώτημα 2. Οι άνθρωποι, τα αυτοκίνητα και οι λεπτομέρειες έχουν διατηρηθεί σε μεγάλο βαθμό και πλέον είναι ευδιάκριτα, καθώς και οι κινήσεις τους. Αυτό συμβαίνει επειδή όταν συμπιέζουμε όλα τα 800 καρέ ταυτόχρονα με απλή SVD, αναγκάζουμε ολόκληρη την ακολουθία του βίντεο, κάθε αλλαγή φωτισμού, κίνηση αντικειμένων και μετατόπιση φόντου, να χωρέσει σε μόλις δέκα ιδιάζουσες τιμές. Με την πάροδο του χρόνου, η σκηνή μπορεί να εξελιχθεί με πολλούς διαφορετικούς τρόπους και οι απουσία singular values απλώς δεν μπορεί να προσαρμοστεί σε νέες κινήσεις ή αλλαγές φωτισμού που συμβαίνουν αργότερα στην ακολουθία.

Προσθέτοντας, η διαίρεση των ίδιων 800 frame σε οκτώ ανεξάρτητα μπλοκ των 100 αλλάζει ριζικά αυτόν τον περιορισμό. Μέσα σε κάθε μπλοκ, η σκηνή είναι πιο στατική, δηλαδή τα αντικείμενα κινούνται λιγότερο, ο φωτισμός παραμένει πιο σταθερός και η στατιστική δομή δεν αλλάζει τόσο δραματικά. Υπολογίζοντας SVD για κάθε μπλοκ, επιτρέπουμε στις συναρτήσεις βάσης να «επαναμορφώνονται» κάθε 100 καρέ, αποτυπώνοντας τοπικές κινήσεις και υφές πολύ πιο πιστά από ότι θα μπορούσε η κλασσική μέθοδος.

Τέλος, αυτή η προσέγγιση επίσης αποτρέπει τη συσσώρευση σφαλμάτων. Σε ένα κλασσικό μοντέλο, οποιοδήποτε λάθος στην προσέγγιση των πρώτων καρέ μπορεί να μεταδοθεί και στα επόμενα. Όμως, με χρήση block, τα σφάλματα σε ένα τμήμα παραμένουν περιορισμένα σε αυτό και δεν προκαλούν στρέβλωση για τα επόμενα frame.





3 singular values

Τα αποτελέσματα είναι ακόμη πιο εντυπωσιακά για την συγκεκριμένη περίπτωση. Στο ερώτημα 3, είδαμε πως στο πρώτο βίντεο οι άνθρωποι εξαφανίζονται σχεδόν πλήρως ενώ στα αλλά 2 είχαμε ολοκληρωτική απώλεια των λεπτομερειών που εμφανίζονται. Τώρα, διατηρούμε λεπτομέρεια που σε μερικά σημεία δεν υπήρχε ούτε με 10 ιδιάζουσες τιμές (ερώτημα 2 με απλή svd, προφανώς το μπλοκ με τις 10 τιμές είναι καλύτερο από αυτό με 3).

Συγκεκριμένα, όταν εφαρμόζουμε απλή SVD και κρατάμε τις πρώτες 10 τιμές, όλες οι κινήσεις, οι αλλαγές φωτισμού και οι μεταβάσεις σκηνών σε αυτά τα 800 καρέ πρέπει να αναπαρασταθούν από αυτές τις ίδιες δέκα τιμές. Αντιθέτως, χωρίζοντας το βίντεο σε οκτώ μπλοκ των 100 frame και κρατώντας μόνο τρεις singular values ανά μπλοκ, επιτρέπουμε ουσιαστικά τη χρήση μέχρι και 24 συνολικών βασικών εικόνων (3 ανά μπλοκ) για να αποτυπώσουμε το περιεχόμενο.

Επιπλέον, η SVD ανά μπλοκ προσαρμόζει τη βάση της στις τοπικές αλλαγές. Ας υποθέσουμε ότι στα καρέ 1-100 κάποιος περπατάει από αριστερά προς τα δεξιά και στα καρέ 101-200 η σκιά του ήλιου σαρώνει το γκαζόν. Μία απλή SVD θα «ξεπλέξει» αυτούς τους δύο πολύ διαφορετικούς ρυθμούς κινήσεων στις κορυφαίες δέκα δομές της, έτσι ώστε τα ιδιοδιανύσματα πέραν των πρώτων λίγων να περιγράφουν μικρά «υπολείμματα» ή θόρυβο. Αντίθετα, με την block SVD, τα τρία ιδιοδιανύσματα στρέφονται ακριβώς στα ισχυρότερα feature εκείνου του παραθύρου.





PCA

Διαδικασία

2)

Τα αποτελέσματα είναι σχεδόν ίδια οπτικά με ελάχιστες αλλαγές. Αναλυτικότερα, κατά την ανακατασκευή με SVD, ο αλγόριθμος θεωρεί το συνολικό μέσο καρέ ως απλώς μια ακόμη συνιστώσα που πρέπει να αποτυπωθεί. Αυτό σημαίνει ότι το στατικό φόντο και τα δυναμικά μέρη της σκηνής συχνά αναμειγνύονται μαζί σε αυτές τις κύριες συνιστώσες.

Αντιθέτως, η PCA ξεκινά πάντα αφαιρώντας ρητώς τη μέση εικόνα καρέ-ανά-καρέ. Στην ουσία, το στατικό φόντο αφαιρείται και αποθηκεύεται χωριστά. Στη συνέχεια, υπολογίζουμε τα ιδιοδιανύσματα της συνδιακύμανσης αυτών των κεντραρισμένων καρέ, προβάλλουμε κάθε κεντραρισμένο καρέ στον υποχώρο που ορίζεται από τις κορυφαίες κύριες συνιστώσες και τελικά αυτό προσθέτετε ξανά τη μέση εικόνα. Οπτικά, αυτό σημαίνει ότι το στατικό φόντο εμφανίζεται σχεδόν τέλεια στην ανακατασκευή και οι διατηρούμενες κύριες συνιστώσες περιγράφουν μόνο αποκλίσεις από αυτό το μέσο.

Μαθηματικά, τόσο η SVD όσο και η παρέχουν low-rank προσεγγίσεις, αλλά διαφέρουν ελαφρώς. Η SVD στο μητρώο δεδομένων βρίσκει την rank k προσέγγιση που ελαχιστοποιεί το συνολικό τετράγωνο σφάλμα σε όλες τις

τιμές των pixel, ενώ η PCA βρίσκει τις κατευθύνσεις που μεγιστοποιούν την διακύμανση στα κεντραρισμένα δεδομένα, που ισοδυναμεί με την ελαχιστοποίηση του σφάλματος ανακατασκευής μόνο στα κεντραρισμένα frame.

3)

Η διαφορά σε αυτή την περίπτωση σε σχέση με την svd είναι πλέον εμφανής. Στο πρώτο βίντεο υπάρχει ελάχιστη βελτίωση, κυρίως στην μείωση ανθρώπων «φάντασμα», καθώς δεν εμφανίζονται τόσο έντονες λευκές σιλουέτες. Επιπλέον, στο βίντεο 2 η κίνηση των δέντρων από τον άνεμο είναι πολύ καλύτερη και μπορούμε να διακρίνουμε μερικά οχήματα. Τέλος, στο τρίτο βίντεο μπορούμε να διακρίνουμε τους ανθρώπους μερικές στιγμές.

Αναλυτικότερα, με την SVD, οι πρώτες μερικές ιδιοτιμές αφιερώνονται σχεδόν αποκλειστικά στην αναπαράσταση του μέσου frame (του στατικού φόντου), καθώς αυτό έχει τη μεγαλύτερη «ενέργεια» (διακύμανση) στο βίντεο. Όταν διατηρήσουμε μόνο τρεις συνιστώσες, σχεδόν όλος ο «προϋπολογισμός» πηγαίνει στην αναπαράσταση του φόντου, και δεν απομένει αρκετή «χωρητικότητα» για κωδικοποίηση των χαμηλότερης ενέργειας συμβάντων π.χ. ενός ανθρώπου που περπατά ή ενός αυτοκινήτου που περνά.

Αντιθέτως, η PCA αφαιρεί ρητά πρώτα αυτό το σταθερό φόντο και το χειρίζεται ως ξεχωριστή μέση εικόνα. Όταν υπολογίζουμε τα ιδιοδιανύσματα της κεντροποιημένης σειράς frame, οι κυρίαρχες κατευθύνσεις διακύμανσης αντιστοιχούν άμεσα στην κίνηση και τις αλλαγές, δηλαδή τα δέντρα που κουνιούνται από τον άνεμο, τα διερχόμενα οχήματα, τους ανθρώπους που εισέρχονται ή εξέρχονται από τη σκηνή.





4)

10 components

Τα αποτελέσματα είναι πρακτικά ίδια με αυτά της svd. Με «μπλοκ» 100 καρτέ, τόσο η PCA (με αφαίρεση και επαναπρόσθεση του μέσου όρου) όσο και η SVD καταλήγουν να μοντελοποιούν ουσιαστικά τον ίδιο υποχώρο και αυτό εξηγεί γιατί οι ανακατασκευές τους φαίνονται ίδιες.

3 components

Τα συμπεράσματα είναι ίδια με αυτά που βγάλαμε στο αντίστοιχο ερώτημα για την svd. Επιπλέον, η pca έχει σαφώς καλύτερα αποτελέσματα για τους λογούς που έχουμε εξηγήσει παραπάνω.

Eigenfaces

Διαδικασία

1)

Για το dataset, χρησιμοποιήσαμε το ίδιο με αυτό της άσκησης 4, δηλαδή το orl faces dataset. Έχουμε 6 συνολικές κατηγορίες που έχει κάθε εικόνα: gender, smile, facial hair, pose, long/short hair και glasses. Άρα, έχουμε 6 δυαδικές επιλογές για την κάθε εικόνα. Για την χρήση αυτού του dataset, έχουμε 80/20 split, δηλαδή 8 εικόνες για training και 2 για testing από τον κάθε φάκελο.

image	folder	gender	hair_long	facial_hair	pose_tumed	smile	glasses	split
/content/drive/MyDrive/s1		1	0	0	0	0	0	train
/content/drive/MyDrive/s1		1	0	0	1	0	0	eval
/content/drive/MyDrive/s1		1	0	0	1	0	0	train
/content/drive/MyDrive/s1		1	0	0	0	0	0	train
/content/drive/MyDrive/s1		1	0	0	1	0	0	train
/content/drive/MyDrive/s1		1	0	0	1	0	0	train
/content/drive/MyDrive/s1		1	0	0	0	0	0	train
/content/drive/MyDrive/s1		1	0	0	0	0	0	train
/content/drive/MyDrive/s1		1	0	0	1	0	0	eval
/content/drive/MyDrive/s10		0	1	0	1	1	0	train
/content/drive/MyDrive/s10		0	1	0	0	0	0	eval
/content/drive/MyDrive/s10		0	1	0	0	1	0	train
/content/drive/MyDrive/s10		0	1	0	1	1	0	train
/content/drive/MyDrive/s10		0	1	0	1	1	0	train
/content/drive/MyDrive/s10		0	1	0	0	0	0	train
/content/drive/MyDrive/s10		0	1	0	0	0	0	train
/content/drive/MyDrive/s10		0	1	0	1	0	0	train
/content/drive/MyDrive/s10		0	1	0	0	0	0	train
/content/drive/MyDrive/s10		0	1	0	1	0	0	train
/content/drive/MyDrive/s10		0	1	0	0	0	0	eval

2)

Η μέθοδος των eigenfaces για την αναγνώριση προσώπων ξεκινά με την μετατροπή κάθε εικόνας προσώπου διαστάσεων $N \times M$ σε ένα διάνυσμα στο R^{NM} . Έχοντας K εκπαιδευτικές εικόνες I_1, \dots, I_K , η κάθε μία «ξετυλίγεται» σε στήλη $c_i \in R^{NM}$ και όλα τα διανύσματα συγκεντρώνονται στον πίνακα δεδομένων $C = [c_1 c_2 \dots c_K]$. Υπολογίζεται το «μέσο πρόσωπο»:

$$c = \frac{1}{K} \sum_{i=1}^K c_i$$

και το αφαιρούμε από κάθε εικόνα ώστε να κεντράρουμε τα δεδομένα $v_i = c_i - c$. Ο πίνακας V περιγράφει τις αποκλίσεις κάθε προσώπου από το μέσο πρόσωπο. Για να αποφύγουμε τον υπολογισμό του μεγάλου πίνακα συνδιακύμανσης εκτελούμε SVD στη μικρότερη μήτρα:

$$X^T = \frac{v}{\sqrt{K}}$$

Η SVD μας επιτρέπει να εξάγουμε τα πρώτους k δεξιά ιδιοδιανύσματα (συνδεδεμένα με τις μεγαλύτερες ιδιοτιμές) και όταν τα απεικονίσουμε πίσω στον χώρο των pixel, λαμβάνουμε τα ιδιοπρόσωπα q_1, \dots, q_k . Συνεχίζοντας, σχηματίζουμε τον πίνακα Q , όπου κάθε κεντραρισμένο πρόσωπο v προβάλλεται στον υποχώρο των ιδιοπροσώπων μέσω του διανύσματος βάρος:

$$w = Q^t v \in R^k$$

το οποίο συμπυκνώνει τα πιο σημαντικά χαρακτηριστικά του. Για να ανακατασκευάσουμε το πρόσωπο, υπολογίζουμε $v' = Qw$ και επαναφέρουμε το μέσο πρόσωπο $c' = v' + c$. Για testing, προβάλλουμε το πρόσωπο δοκιμής με τον ίδιο τρόπο και συγκρίνουμε το w_{test} με τα αποθηκευμένα βάρη των εκπαιδευτικών εικόνων.



Η εικόνα που βλέπουμε είναι ο μέσος όρος όλων των ευθυγραμμισμένων προσώπων στο σύνολο εκπαίδευσής. Κάθε τιμή pixel προκύπτει αθροίζοντας την ένταση αυτού σε κάθε εικόνα και διαιρώντας με τον αριθμό των εικόνων. Συνεπώς, όλες οι ιδιόμορφες λεπτομέρειες, όπως το ακριβές σχήμα των ματιών και οι σκιές από διαφορετικό φωτισμό σχεδόν εξαφανίζονται. Αυτό που μένει είναι τα ευρύτερα, δομικά χαρακτηριστικά κοινά σε όλα τα πρόσωπα του συνόλου.

3)

Η φάση αξιολόγησης του συστήματος ξεκινάει με τη φόρτωση των παραγόμενων αντικειμένων από την εκπαίδευση, δηλαδή του μέσου προσώπου, του πίνακα Q και των διανυσμάτων βαρών W_{train} για όλες τις εικόνες εκπαίδευσης. Για κάθε εικόνα αξιολόγησης, το πρώτο βήμα είναι η ανάγνωση του ακατέργαστου διανύσματος pixel c_{test} , η αφαίρεση του μέσου προσώπου για κεντράρισμα και η προβολή του αποτελέσματος στην βάση των eigenfaces:

$$W_{test} = Q^T(c_{test} - c')$$

Το αποτέλεσμα είναι ένα k -διάστατο διάνυσμα w_{test} που κωδικοποιεί την «ένταση» εμφάνισης κάθε eigenface στη νέα εικόνα. Μόλις κάθε εικόνα αξιολόγησης αναπαρασταθεί ως διάνυσμα βαρών, η αναγνώριση

απλοποιείται σε μια αναζήτηση nearest neighbor στον k -διάστατο χώρο. Τότε, υπολογίζεται η απόσταση ℓ_2 ανάμεσα σε κάθε «δοκιμαστικό» διάνυσμα και όλα τα εκπαιδευτικά διανύσματα:

$$d_{ij} = \left\| \mathbf{w}_{train,i} - \mathbf{w}_{test,j} \right\|_2$$

Τέλος, για κάθε εικόνα j , επιλέγεται εκείνη η εκπαιδευτική εικόνα i που ελαχιστοποιεί την απόσταση d_{ij} .

4)

Σε αυτό το ερώτημα, μετράμε πόσο καλά αποδίδει ο αλγόριθμος για κάθε μεμονωμένο άτομο, υπολογίζοντας δύο κλασικές διαγνωστικές μετρικές, την sensitivity ή recall και την specificity. Ξεκινάμε από τον πίνακα C , όπου κάθε γραμμή i αντιστοιχεί στο πραγματικό subject s_i και κάθε στήλη j στο προβλεπόμενο subject s_j . Το στοιχείο C_{ij} μετρά πόσες φορές μια εικόνα του s_i ταξινομήθηκε ως s_j .

Προσθέτοντας, για ένα συγκεκριμένο άτομο s , οι true positive (TP) είναι οι εικόνες του s που αναγνωρίστηκαν σωστά ως s , δηλαδή το διαγώνιο στοιχείο C_{ss} . Οι false negative (FN) είναι οι εικόνες του s που ταξινομήθηκαν λανθασμένα ως κάποιος άλλος. Τέλος, οι true negative (TN) είναι όλες οι εικόνες που δεν ανήκουν στο s και δεν προβλέφθηκαν ως s .

Έπειτα, ορίζουμε για κάθε άτομο s :

- Sensitivity:

$$Sens_s = \frac{TP_s}{TP_s + FN_s}$$

- Specificity:

$$Spec_s = \frac{TN_s}{TN_s + FP_s}$$

	sensitivity	specificity
s1	1.0	1.000
s10	0.0	1.000
s11	1.0	0.987
s12	1.0	1.000
s13	1.0	1.000
s14	1.0	1.000
s15	1.0	1.000
s16	1.0	1.000
s17	1.0	1.000
s18	1.0	0.987
s19	0.5	1.000
s2	1.0	1.000
s20	1.0	1.000
s21	1.0	1.000
s22	1.0	1.000
s23	1.0	1.000
s24	1.0	1.000
s25	1.0	1.000
s26	1.0	1.000
s27	1.0	1.000
s28	1.0	1.000
s29	1.0	1.000
s3	1.0	1.000
s30	1.0	1.000
s31	1.0	1.000
s32	1.0	1.000
s33	1.0	1.000
s34	1.0	1.000
s35	1.0	1.000
s36	1.0	1.000
s37	1.0	1.000
s38	1.0	0.974
s39	1.0	1.000
s4	1.0	1.000
s40	1.0	1.000
s5	0.5	1.000
s6	1.0	1.000
s7	1.0	1.000
s8	1.0	1.000
s9	1.0	1.000

Βλέπουμε από τα αποτελέσματα ότι σε όλα τα subject, το specificity είναι εξαιρετικά υψηλά (με εξαίρεση τα s11, s18 και s38). Αυτό σημαίνει ότι ο ταξινομητής σχεδόν ποτέ δεν μπερδεύει πρόσωπα, δηλαδή οι ψευδείς θετικές ταξινομήσεις είναι πρακτικά ανύπαρκτες.

Αντιθέτως, το sensitivity έχει μεγάλη διακύμανση. Αν και οι περισσότερες εικόνες έχουν άριστο σκορ, αυτές στον φάκελο s10 έχουν ευαισθησία 0. Αυτό υποδεικνύει ότι καμία από τις εικόνες δεν αναγνωρίστηκε σωστά και όλες ταξινομήθηκαν λανθασμένα ως κάποιος άλλος subject. Επιπλέον, στον s19 και s5 έχουμε ευαισθησία 0.5, δηλαδή τυχαία επιλογή.

Συμπερασματικά, το σύστημα μας λειτουργεί με μεγάλη ακρίβεια και έχει μόνο μερικές εξαιρέσεις που πιθανόν να οφείλονται σε συγκεκριμένες αλλαγές στην γωνιά των εικόνων.