

**Τμήμα
Μηχανικών
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**“ΑΛΓΟΡΙΘΜΟΙ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ ΚΑΤΑΜΕΤΡΗΣΗΣ ΑΝΘΡΩΠΩΝ ΣΕ
ΕΛΕΓΧΟΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ”**

ΑΚΡΙΒΟΣ ΜΑΡΙΟΣ Α.Μ:1638

ΚΟΡΚΟΝΤΖΗΛΑΣ ΙΩΑΝΝΗΣ Α.Μ:1717

ΕΠΙΒΛΕΠΩΝ: ΤΣΑΚΑΝΙΚΑΣ ΒΑΣΙΛΕΙΟΣ

ΑΝΤΙΠΡΙΟ 2018

Περίληψη

Στην παρούσα εργασία γίνεται εκτενής αναφορά στο επιστημονικό αντικείμενο της μηχανικής όρασης. Με τον όρο μηχανική όραση εννοούμε την προσπάθεια εξομοίωσης της ανθρώπινης όρασης από κάποια συσκευή, όπως ένα ρομπότ ή άλλο ηλεκτρονικό σύστημα. Η μηχανική όραση βασίστηκε πάνω σε ήδη υπάρχοντα επιστημονικά αντικείμενα, όπως η επεξεργασία εικόνας, η αναγνώριση προτύπων και η μηχανική μάθηση. Για να καταλήξουμε στην ψηφιακή αναγνώριση προσώπων πρέπει πρώτα να έχουμε την δυνατότητα αποθήκευσης των δεδομένων της εικόνας με τέτοιο τρόπο ώστε στην συνέχεια να μπορούμε να την αναλύσουμε και να την επεξεργαστούμε σε σχετικά σύντομο χρόνο. Για τον λόγο αυτό οι ψηφιακές εικόνες αποθηκεύονται ως δισδιάστατοι πίνακες, όπου κάθε στοιχείο του πίνακα είναι ένα διάνυσμα ακέραιων αριθμών, οι οποίοι περιγράφουν κάποιες ιδιότητες της εικόνας στο συγκεκριμένο τμήμα το οποίο αναπαριστούν. Το τι ακριβώς σημαίνουν οι αριθμοί αυτοί το καθορίζουν τα χρωματικά μοντέλα. Με τον όρο χρωματικά μοντέλα εννοούμε έναν τρισδιάστατο συνήθως χώρο, όπου κάθε χρώμα αναπαρίσταται από ένα σημείο. Υπάρχουν διάφορα χρωματικά μοντέλα, η ανάλυση των οποίων όμως ξεφεύγει από τα όρια της παρούσας εργασίας. Στην εργασία αυτή, γίνεται εκτενής αναφορά στο χρωματικό μοντέλο RGB. Συγκεκριμένα για το πρότυπο RGB κάθε pixel χαρακτηρίζεται από τρεις τιμές φωτεινότητας μια για το κόκκινο, μια για το πράσινο και τέλος μια για το μπλε. Ουσιαστικά έχουμε τρεις διαφορετικές εικόνες, μια για κάθε χρώμα. Στην συνέχεια οι εικόνες αυτές προστίθενται και η πρόσθεση των τριών πρωτευόντων χρωμάτων δημιουργεί τα υπόλοιπα χρώματα σε ποικίλες αναλογίες. Το πρότυπο αυτό χρησιμοποιείται όταν η εικόνα προβάλλεται σε οθόνη ηλεκτρονικού υπολογιστή.

Για να μπορέσουμε να επεξεργαστούμε μια εικόνα πρέπει να εφαρμόσουμε σε αυτή κάποιο μετασχηματισμό. Στην παρούσα εργασία γίνεται λόγος για τον δισδιάστατο μετασχηματισμό Haar, ο οποίος βασίζεται στην θεώρηση των κυματιδίων. Κύριο πλεονέκτημα του μετασχηματισμού συνίσταται στο ότι δειγματοληπτεί τα δεδομένα εισόδου σε πολλαπλές αναλύσεις με χαμηλή έως υψηλή ανάλυση και υπολογίζει τις διαφορές γειτονικών ζευγών.

Τέλος γίνεται αναφορά στους ταξινομητές, οι οποίοι είναι αλγόριθμοι που χρησιμοποιούνται για να επιτύχουμε την ταξινόμηση των προτύπων σε όσο το δυνατόν ομογενείς από άποψη χαρακτηριστικών κλάσεις. Υπάρχει μια ευρεία γκάμα ταξινομητών. Η επιλογή του κατάλληλου ταξινομητή σχετίζεται άμεσα με το είδος των χαρακτηριστικών και το είδος της εφαρμογής όπου θα χρησιμοποιηθεί. Η επιλογή των κατάλληλων χαρακτηριστικών και του ταξινομητή πρέπει να γίνει ταυτόχρονα. Με την τεχνική της ενδυνάμωσης (boosting) μπορούμε να μετατρέψουμε έναν απλό (ή ασθενή) ταξινομητή σε ισχυρό. Ένας τέτοιος αλγόριθμος ενδυνάμωσης είναι ο Adaboost, ο οποίος μπορεί να χρησιμοποιηθεί για την ενίσχυση της απόδοσης οποιουδήποτε αλγορίθμου εκμάθησης μηχανών. Η ενδυνάμωση επιτυγχάνεται με την κατασκευή ενός μοντέλου από τα δεδομένα εκπαίδευσης, στη συνέχεια δημιουργείται ένα δεύτερο μοντέλο που επιχειρεί να διορθώσει τα σφάλματα του πρώτου μοντέλου κ.ο.κ. Τα μοντέλα προστίθενται έως ότου προβλεφθεί απόλυτα το σετ εκπαίδευσης ή προστεθεί ένας μέγιστος αριθμός μοντέλων.

Περιεχόμενα

Περίληψη.....	2
1. Εισαγωγή στην ψηφιακή εικόνα.....	6
2. Τρόποι αναπαράστασης της ψηφιακής εικόνας	11
2.1 Αναπαράσταση ψηφιακής εικόνας.....	11
2.2 Χρωματικά μοντέλα: το πρότυπο RGB.....	15
3. Μηχανική όραση	20
3.1. Ιστορική αναδρομή.....	22
3.2 Βασικές τεχνολογίες.....	23
4. Τα χαρακτηριστικά του Haar.....	27
5. Μηχανική όραση και ταξινομητές.....	35
5.1 Εισαγωγή στους ταξινομητές	35
5.2 Ενίσχυση απόδοσης- Αλγόριθμος Adaboost.....	40
Βιβλιογραφία.....	46
Υλοποίηση	48
Βιβλιοθήκη OpenCV	48
Τεχνικές ανίχνευσης προσώπου και χαρακτηριστικών	51
Ανάλυση εφαρμογής	59
ΠΑΡΑΡΤΗΜΑ.....	77

Πίνακας Διαγραμμάτων

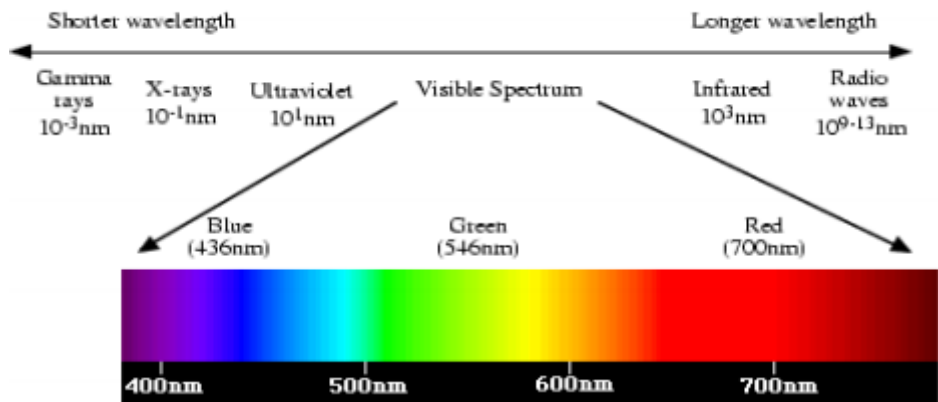
Διάγραμμα 1. Περιοχή RGB	18
Διάγραμμα 2. Μετασχηματισμοί εικόνας.....	28

Πίνακας Εικόνων

Εικόνα 1. Ορατή περιοχή φάσματος	6
Εικόνα 2. Εικονοστοιχεία (pixel) εικόνας.....	8
Εικόνα 3. Κβαντισμός εικόνας με 1-bit (2 επίπεδα), 3 bit (8 επίπεδα), 6- bit (64 επίπεδα)....	10
Εικόνα 4. Ανάλυση εικόνας	10
Εικόνα 5. Γειτνίαση pixels	12
Εικόνα 6. Τα σύνολα N_4 και N_8 ενός pixelp.	13
Εικόνα 7. Γειτνίαση.....	13
Εικόνα 8. Συνεκτικότητα a. 4-connected, b. 8-connected, m- connected	15
Εικόνα 9. Ενδιάμεσο χρώμα.....	17
Εικόνα 10. Πρόσθεση πρωτεύοντων χρωμάτων.....	17
Εικόνα 11. Χρήση έγχρωμης εικόνας σε τόνους γκρι.....	19
Εικόνα 12. Αναπαράσταση τμήματος εικόνας στον υπολογιστή.....	21
Εικόνα 13. Αυτόνομο σύστημα πλοήγησης του Πανεπιστημίου CarnegieMellon	24
Εικόνα 14. α. 2-D εικόνα bitmap, β. 2-D εικόνα vector.....	26
Εικόνα 15. Μονοδιάστατες συναρτήσεις Haar.....	30
Εικόνα 16. Εικόνες βάσης του 2DHaar για $N=8$	33
Εικόνα 17. Παράδειγμα εκτέλεσης Adaboost	45
Εικόνα 18. Σωστό αποτέλεσμα αλγορίθμου.	68
Εικόνα 19. Λάθος αποτέλεσμα αλγορίθμου. Εντοπίστηκαν 2 λιγότερα άτομα	69
Εικόνα 20. Αποτέλεσμα εκτέλεσης	70
Εικόνα 21. Αποτέλεσμα εκτέλεσης	70
Εικόνα 22. Αποτέλεσμα εκτέλεσης	71
Εικόνα 23. Αποτέλεσμα εκτέλεσης	71
Εικόνα 24. Αποτέλεσμα εκτέλεσης αλγορίθμου με φυτά. Με την κόκκινη ακμή σημειώνεται η εσφαλμένη έξοδος και με την πράσινη η σωστή έξοδος.	73
Εικόνα 25. Αποτέλεσμα εκτέλεσης	74
Εικόνα 26. Αποτέλεσμα εκτέλεσης	74
Εικόνα 27. Αποτέλεσμα εκτέλεσης	75
Εικόνα 28. Αποτέλεσμα εκτέλεσης	75

1. Εισαγωγή στην ψηφιακή εικόνα

Το σύστημα της ανθρώπινης όρασης είναι υπεύθυνο να αντιλαμβάνεται τα φωτεινά ερεθίσματα και ο εγκέφαλος τα επεξεργάζεται ώστε να γίνεται αντιληπτή συνολική εικόνα. Το χρώμα που αντιλαμβάνεται ο άνθρωπος αποτελεί μια υποκειμενική αντίληψη των συχνοτήτων του φωτός στην ορατή περιοχή του φάσματος, η οποία είναι από 400 nm μέχρι και 700 nm. Η εικόνα που βλέπει το μάτι είναι αναλογική, δηλαδή είναι συνεχής αναπαράσταση στον χώρο.

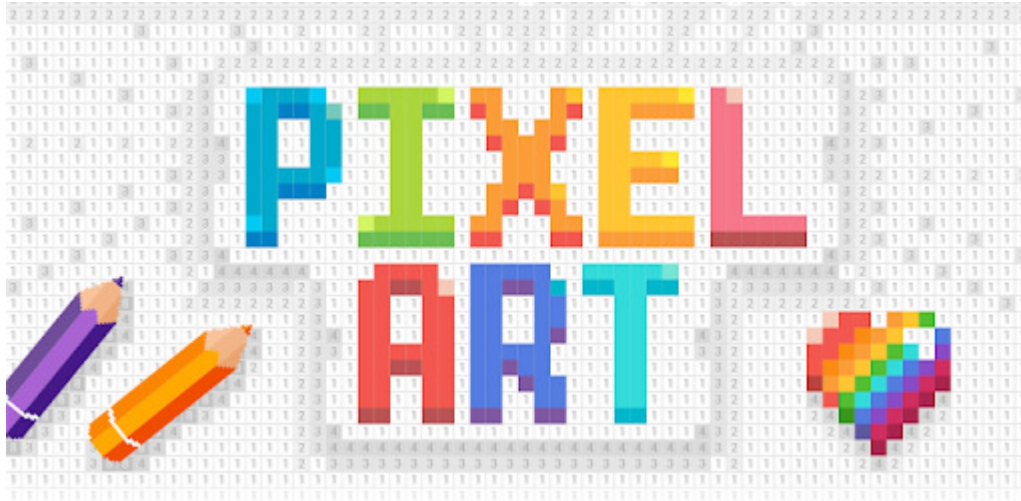


Εικόνα 1. Ορατή περιοχή φάσματος

Μια ψηφιακή εικόνα είναι μια απεικόνιση σε έναν διακριτό χώρο δύο διαστάσεων. Για να προκύψει μια ψηφιακή εικόνα από μια αναλογική εικόνα που βλέπει το μάτι πρέπει να γίνει ψηφιοποίηση της αναλογικής εικόνας. Η συνεχής εικόνα διαιρείται σε N σειρές και M στήλες. Τα ορθογώνια που σχηματίζονται ονομάζονται εικονοστοιχεία (pixels) και μπορούμε να τα διακρίνουμε στην εικόνα 2. Κάθε pixel χαρακτηρίζεται από ένα σύνολο τιμών που καθορίζουν την φωτεινότητα του, η οποία είναι κοινή για όλο το pixel. Το τι ακριβώς προσδιορίζει η τιμή αυτή εξαρτάται από το πρότυπο αναπαράστασης που έχει επιλεγεί για την συγκεκριμένη εικόνα.

Η ψηφιοποίηση μιας εικόνας περιλαμβάνει δυο διαδικασίες, την δειγματοληψία και τον κβαντισμό. Κατά την δειγματοληψία λαμβάνονται κάποιες τιμές φωτεινότητας της αναλογικής εικόνας για κάθε pixel και στην συνέχεια υπολογίζεται ο μέσος όρος αυτών των τιμών. Κατά τον κβαντισμό η μέση τιμή που παρήγαγε η διαδικασία της δειγματοληψίας για το pixel αντιστοιχίζεται σε μια ακέραια τιμή που αντιπροσωπεύει το σύνολο τιμών της περιοχής στην οποία ανήκει η συγκεκριμένη μέση τιμή. Με άλλα λόγια όλη η ορατή περιοχή του φάσματος χωρίζεται σε ένα συγκεκριμένο πλήθος επιπέδων (για παράδειγμα 256 επίπεδα), όπου καθένα από αυτά τα επίπεδα αντιπροσωπεύεται από ένα αριθμό. Έτσι οποιαδήποτε μέση τιμή φωτεινότητας έχει υπολογιστεί για ένα pixel, αυτή αντιστοιχίζεται στον αντιπροσωπευτικό αριθμό που έχει επιλεγεί για το επίπεδο στο οποίο ανήκει. Στην εικόνα 3 διακρίνουμε πως φαίνεται μια εικόνα αν την κβαντίσουμε με διαφορετικό πλήθος επιπέδων. Είναι αισθητή η διαφορά όσο τα επίπεδα γίνονται περισσότερα. Βέβαια όσα περισσότερα επίπεδα έχουμε αυξάνεται και το μέγεθος αποθήκευσης της εικόνας. Ο αριθμός που δηλώνει πόσα bits χρησιμοποιούνται για την αποθήκευση της πληροφορίας χρώματος του κάθε pixel τον ονομάζουμε *βάθος χρώματος*. Τελικά μια ψηφιακή εικόνα μεγέθους $M \times N$, αποτελείται από MN pixels και χαρακτηρίζεται από τουλάχιστον MN επί πλήθος bits βάθος χρώματος τιμές ανάλογα με το πρότυπο που επιλέχθηκε για να αναπαρασταθεί¹.

¹ Αναλυτική παρουσίαση των προτύπων αναπαράστασης γίνεται σε επόμενα κεφάλαια.

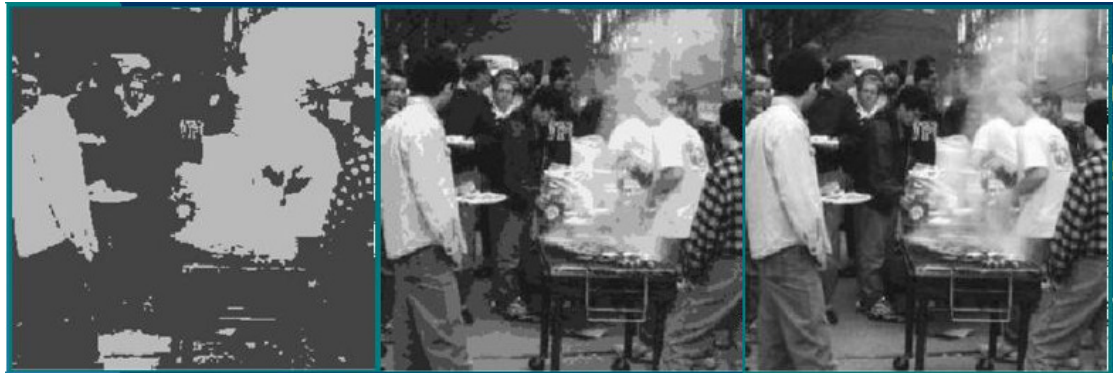


Εικόνα 2. Εικονοστοιχεία (pixel) εικόνας

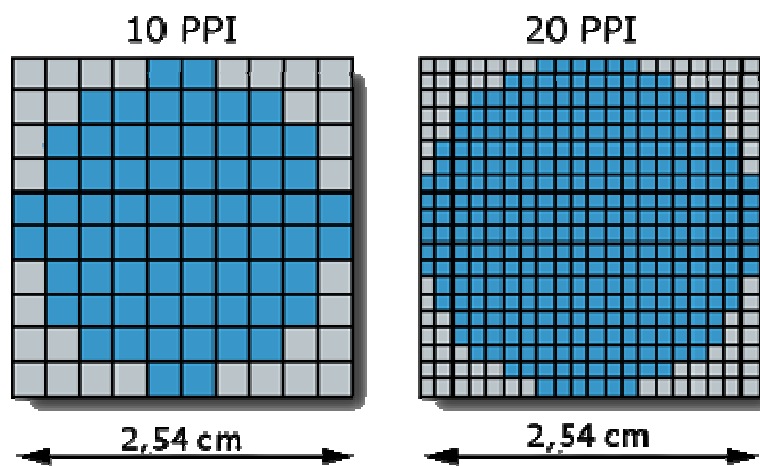
Ουσιαστικά το βάθος χρώματος ενός pixel εκφράζει τον αριθμό από τα ενδιάμεσα βήματα (διαβαθμίσεις) που είναι δυνατόν να οριστούν ανάμεσα στα δύο άκρα της κλίμακας μέτρησης (π.χ. άσπρο-μαύρο ή όρια υπέρυθρου-υπεριώδους στις έγχρωμες εικόνες). Παράλληλα, το βάθος μας δείχνει πόσα bits χρησιμοποιούνται για την καταγραφή της αξίας κάθε pixel και μπορεί να είναι από 1 και πάνω bits. Οι διαθέσιμες διαβαθμίσεις είναι το αποτέλεσμα που μας δίνει η τιμή του βάθους του pixel σαν δύναμή του 2. Μια μονοχρωματική εικόνα με βάθος 2 bits έχει $2^2 = 4$ διαβαθμίσεις, ενώ μια εικόνα των 3 bits θα έχει $2^3 = 8$ διαβαθμίσεις. Στην περίπτωση που ένα σύστημα συλλαμβάνει μονοχρωματικές εικόνες με ακρίβεια 8-bit, κάθε pixel καταγράφει οκτώ δυαδικά επίπεδα άρα μπορεί να διακρίνει 2^8 βαθμίδες (επίπεδα) άρα 256 διαφορετικές τιμές. Επομένως μπορεί να καταγράψει το μαύρο σαν 0, το άσπρο σαν 255 και ανάμεσά τους, 254 βαθμίδες (τόνους) του γκρι.

Με αντίστοιχο τρόπο αναπαριστούμε τις έγχρωμες εικόνες, δηλαδή χρησιμοποιώντας βαθμίδες με κλιμακούμενες διαφορετικές τιμές ενδιάμεσα. Η διαφορά είναι ότι διατηρούμε παραπάνω από μια τιμές χρώματος για κάθε pixel. Μεγαλύτερη ανάλυση για την αναπαράσταση του βάθους μιας εικόνας θα γίνει στο επόμενο κεφάλαιο.

Το μέγεθος που δείχνει από πόσα pixels αποτελείται μια ψηφιακή εικόνα στην μονάδα του μήκους λέγεται *ανάλυση* και μετριέται σε ppi, δηλαδή pixels per inch (εικονοστοιχεία ανά ίντσα). Στην εικόνα 4, παρουσιάζεται ένας κύκλος όπως φαίνεται με ανάλυση 10 ppi και 20 ppi. Είναι προφανές ότι η μεγαλύτερη τιμή ppi συνεπάγεται καλύτερη σχηματική απεικόνιση του αντικειμένου. Όσα περισσότερα pixels δηλαδή διαθέτει μια εικόνα για ένα δεδομένο μέγεθος, τόσο περισσότερη πληροφορία μπορεί να περιγράψει. Ας σημειώσουμε πως η εικόνα με την μεγαλύτερη ανάλυση, δεν είναι αυτόματα καλύτερη από εκείνη με την μικρότερη ανάλυση. Η εκτίμηση πρέπει να γίνεται πάντα με βάση κριτήρια συνδεδεμένα με τη χρήση, για την οποία έχει δημιουργηθεί κάθε εικόνα. Το να έχει μια εικόνα μεγάλη ανάλυση δεν είναι πανάκεια ούτε και πάντα αυτό που προτιμάμε. Καθώς το μέγεθος της εικόνας σε Bytes (και συνήθως σε Megabytes) μεγαλώνει ταχύτατα όσο πληθαίνουν τα pixel της, ένα αρχείο μεγαλύτερης ανάλυσης από αυτή που πραγματικά χρειαζόμαστε είναι σπατάλη σε χώρο αποθήκευσης και χρόνο επεξεργασίας. Εξάλλου, η ύπαρξη μεγάλου αριθμού από pixels σε μια εικόνα, από μόνη της, δεν αποτελεί εγγύηση για την ποιότητα της εικόνας. Όπως είδαμε και προηγουμένως, την ακρίβεια της καταγραφής της πληροφορίας που αποθηκεύεται σε ένα pixel την καθορίζει το βάθος χρώματος του pixel. Όσο μεγαλύτερο το βάθος εικονοστοιχείου, τόσο περισσότερη η λεπτομέρεια της πληροφορίας που αποθηκεύεται για ένα pixel.



Εικόνα 3. Κβαντισμός εικόνας με 1-bit (2 επίπεδα), 3 bit (8 επίπεδα), 6- bit (64 επίπεδα)



Εικόνα 4. Ανάλυση εικόνας

2. Τρόποι αναπαράστασης της ψηφιακής εικόνας

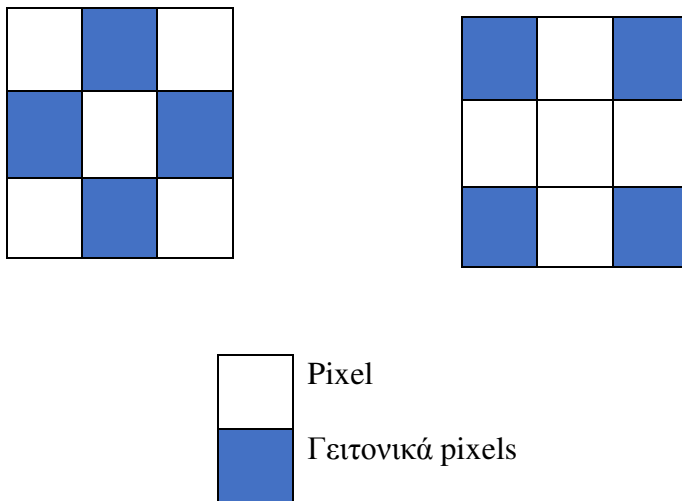
2.1 Αναπαράσταση ψηφιακής εικόνας

Μια ψηφιακή εικόνα αποθηκεύεται ως δισδιάστατος πίνακας ακεραίων αριθμών $I(M,N)$, όπου I είναι το όνομα του πίνακα, M το πλήθος των γραμμών και N το πλήθος των στηλών του:

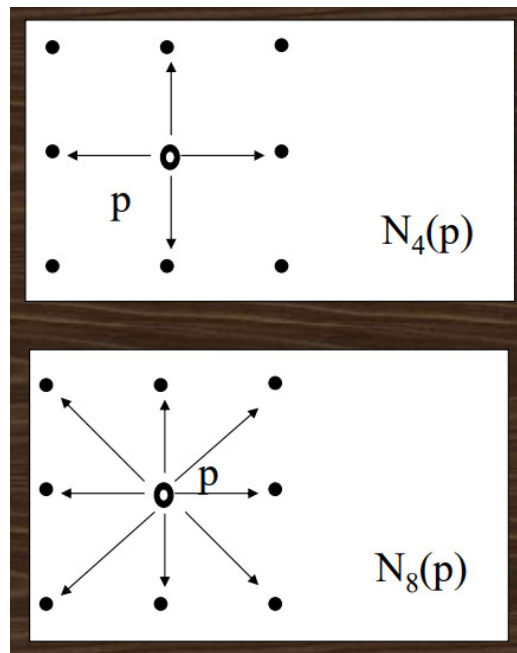
$$I = \begin{bmatrix} I(0,0) & \cdots & I(0,N-1) \\ \vdots & \ddots & \vdots \\ I(M-1,0) & \cdots & I(M-1,N-1) \end{bmatrix}$$

Όταν αναφερόμαστε στο στοιχείο του πίνακα $I(i,j)$ εννοούμε το στοιχείο (ή το pixel) στην γραμμή i και την στήλη j του πίνακα. Το πλήθος των bits ανά pixel που χρησιμοποιείται για να εκφράσει το βάθος χρώματος (ή την φωτεινότητα) του συγκεκριμένου pixel, αποθηκεύεται ως αριθμός του δεκαδικού συστήματος στο αντίστοιχο στοιχείο του πίνακα I . Για παράδειγμα αν μια εικόνα αποθηκεύεται χρησιμοποιώντας 256 διαφορετικές αποχρώσεις του γκρι (δηλαδή για κάθε ένα pixel χρειαζόμαστε 8 bits πληροφορίας βάθους χρώματος), τότε κάθε στοιχείο του πίνακα θα περιέχει έναν αριθμό από το 0 έως και το 255, που αντιστοιχεί στο βάθος χρώματος του συγκεκριμένου σημείου της εικόνας. Επειδή οι πίνακες αυτοί περιέχουν ακέραιους αριθμούς μπορούν να γίνουν σε αυτούς αριθμητικές πράξεις με στόχο την επεξεργασία της εικόνας ή ακόμα και της συμπίεσης της, ώστε να καταλαμβάνει τον λιγότερο δυνατό χώρο μνήμη του συστήματος. Για τις έγχρωμες εικόνες χρησιμοποιούνται περισσότερες τιμές για να εκφραστεί η έγχρωμη απόχρωση. Υπάρχουν διάφορες αναπαραστάσεις για την έγχρωμη εικόνα όπως η RGB, HIS και η CMYK, τις οποίες θα δούμε σε επόμενο κεφάλαιο.

Οι τιμές στον πίνακα της εικόνας δεν είναι εντελώς ανεξάρτητες μεταξύ τους. Έτσι έχουν επικρατήσει οι παρακάτω ορισμοί για την περιγραφή της συσχέτισης των pixel. Κάθε pixel $I(i,j)$ έχει δυο οριζόντιους και δύο κάθετους γείτονες τους $I(i,j-1)$, $I(i,j+1)$, $I(i-1,j)$ και $I(i+1,j)$, αντίστοιχα όπως παρουσιάζεται και στο αριστερό τετράγωνο της εικόνας 5. Αυτό το σύνολο των pixels ονομάζεται 4-neighbors του $I(i,j)$ και συμβολίζεται ως $N_4(I)$. Καθένα από αυτά τα pixels του συνόλου έχει μια μονάδα απόσταση σε σχέση με το $I(i,j)$. Επιπρόσθετα το pixel $I(i,j)$ έχει και τους διαγώνιους γείτονες: $I(i+1,j+1)$, $I(i-1,j-1)$, $I(i+1,j-1)$ και $I(i-1,j+1)$, όπως φαίνεται και στο δεύτερο τετράγωνο δεξιά της εικόνας 5. Το σύνολο αυτών των pixel συμβολίζουμε με $N_D(I)$ και κάθε ένα από αυτά τα pixel απέχει από το $I(i,j)$ κατά την ευκλείδεια απόσταση 1,414. Το σύνολο των pixels των $N_4(I)$ και $N_D(I)$ είναι γνωστό ως το σύνολο $N_8(I)$, όπως φαίνεται στην εικόνα 6. Κάποιοι γείτονες pixels μπορεί να βρίσκονται εκτός ορίου της εικόνας, όπως για παράδειγμα όταν αναφερόμαστε σε pixel που βρίσκεται σε μια πλευρά της εικόνας, π.χ. $I(0,10)$.



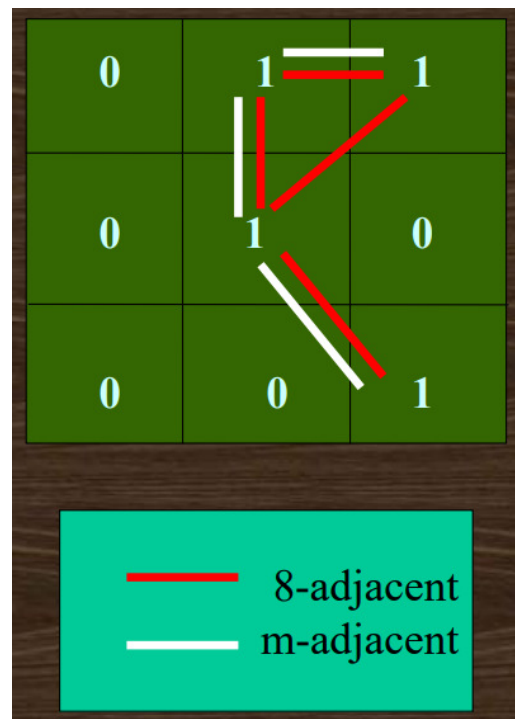
Εικόνα 5. Γειτνίαση pixels



Εικόνα 6. Τα σύνολα N_4 και N_8 ενός pixel p .

Δυο pixels ονομάζονται συνδεδεμένα αν είναι γείτονες και τα επίπεδα του χρώματος το οποίο περιέχουν πληροί κάποια συγκεκριμένα κριτήρια ομοιότητας. Για παράδειγμα σε μια δυαδική εικόνα (των 2 bits) δυο pixel είναι συνδεδεμένα αν είναι 4-neighbors και έχουν ίδια τιμή. Για εικόνες που ανήκουν σε αποχρώσεις του γκρι ισχύει το παρακάτω θεώρημα:

Έστω σύνολο V που περιέχει τα επίπεδα του γκρι χρώματος, τότε έχουμε τις παρακάτω γειτνιάσεις:



Εικόνα 7. Γειτνίαση

4-adjacency: Δυο pixels p και q με τιμές από το σύνολο V είναι 4-adjacency αν το q είναι στο σύνολο $N_4(p)$.

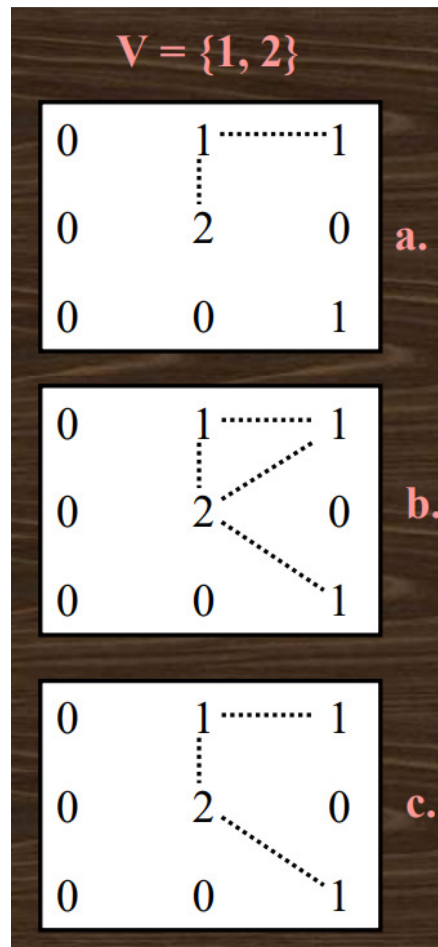
8-adjacency: Δυο pixels p και q με τιμές από το σύνολο V είναι 8-adjacency αν το q είναι στο σύνολο $N_8(p)$.

m -adjacency: Δυο pixels p και q με τιμές από το σύνολο V είναι m -adjacency αν:

- το q είναι στο σύνολο $N_4(p)$.
- το q είναι στο σύνολο $N_D(p)$ και στα σύνολα $N_4(p)$ και $N_D(p)$ δεν υπάρχουν pixels με κοινές τιμές.

Τέλος, με τον όρο συνεκτικότητα (connectivity) ορίζουμε μια γειτνίαση pixel με κάποιον τρόπο. Έστω σύνολο V που περιέχει τα επίπεδα του γκρι χρώματος, τότε έχουμε τις παρακάτω συνεκτικότητες ανάμεσα σε δυο pixels p και q :

- 4-connected: αν το q είναι στο σύνολο $N_4(p)$
- 8-connected, αν το q είναι στο σύνολο $N_8(p)$
- m -connected, αν και μόνο αν ισχύει:
 - το q να είναι στο σύνολο $N_4(p)$ ή
 - το q να είναι στο σύνολο $N_D(p)$ και τα σύνολα $N_4(p)$ και $N_4(q)$ δεν έχουν κοινά στοιχεία.



Εικόνα8. Συνεκτικότητα. a. 4-connected, b. 8-connected, m- connected

2.2 Χρωματικά μοντέλα: το πρότυπο RGB

Η επιστήμη από πολύ παλιά έχει μελετήσει τον τρόπο με τον οποίο βλέπουμε τα χρώματα και τους έχει προσδώσει διάφορες ιδιότητες για να τα χαρακτηρίσει όπως η απόχρωση (Hue- υπερέχον μήκος κύματος), ο χρωματική καθαρότητα (Saturation- το ποσό του λευκού χρώματος της απόχρωσης), η λαμπρότητα (Brightness- πόσο φως προέρχεται από μια πηγή), η φωτεινότητα (Luminance- η πυκνότητα της έντασης του

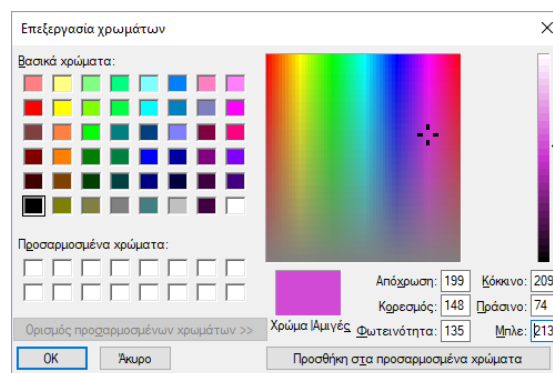
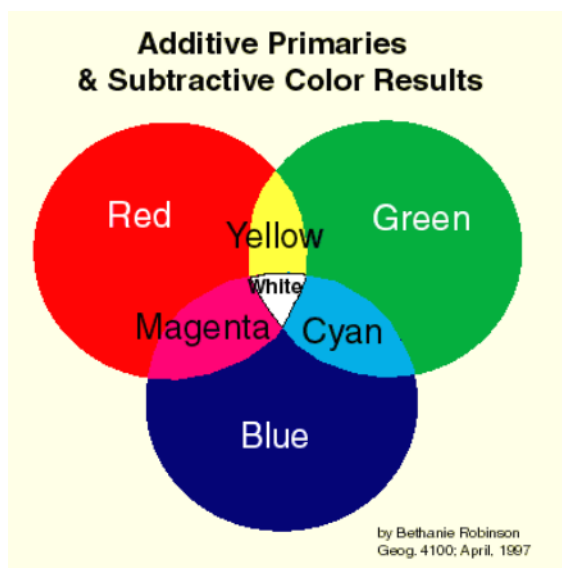
φωτός) και άλλα. Πολλές εργασίες μελέτησαν την δυνατότητα περιγραφής χρωμάτων με μονοσήμαντο τρόπο, λαμβάνοντας υπόψη μόνο κάποιες από τις παραπάνω ιδιότητες. Οι μέθοδοι που αναπτύχθηκαν οδήγησαν σε συστήματα κατάταξης τα οποία είναι γνωστά ως *Χρωματικά Μοντέλα ή Χρωματικά Συστήματα* (Colour Models). Τα χρωματικά μοντέλα έδωσαν μεγάλη ώθηση στην αναπαράσταση και γενικότερα στην χρήση χρωμάτων σε όλα τα μέσα (οθόνες, σαρωτές, εκτυπωτές). Ο σκοπός της χρήσης χρωματικών μοντέλων είναι να διευκολύνεται ο ορισμός των χρωμάτων στα πλαίσια της τυποποίησης. Στην ουσία, ένα χρωματικό μοντέλο είναι ένας τρισδιάστατος συνήθως χώρος όπου κάθε χρώμα αναπαρίσταται από ένα σημείο. Για τον λόγο αυτό τα χρωματικά μοντέλα χαρακτηρίζονται και ως χρωματικοί χώροι.

Σήμερα χρησιμοποιούνται διάφορα χρωματικά μοντέλα, η επιλογή των οποίων καθορίζεται σε μεγάλο βαθμό από τον τομέα στον οποίο θέλουμε να τα εφαρμόσουμε. Τα πιο συνηθισμένα χρωματικά μοντέλα είναι το Red- Green- Blue (RGB), Cyan- Magenta- Yellow (CMYK), Hue- Lightness- Saturation (HLS: απόχρωση- φωτεινότητα- καθαρότητα), Hue- Saturation- Brightness (HSB: απόχρωση- καθαρότητα- λάμψη) και άλλα μοντέλα. Κοινό χαρακτηριστικό τους είναι πως υιοθετούν τρεις παραμέτρους, δηλ. χρειάζονται τρεις ανεξάρτητες τιμές για να προσδιορίσουν μαθηματικά κάποιο χρώμα.

Μια έγχρωμη εικόνα αναπαρίσταται ως διάνυμα τιμών. Συγκεκριμένα για το πρότυπο RGB κάθε pixel χαρακτηρίζεται από τρεις τιμές φωτεινότητας μια για το κόκκινο, μια για το πράσινο και τέλος μια για το μπλε. Ουσιαστικά έχουμε τρεις διαφορετικές εικόνες, μια για κάθε χρώμα. Στην συνέχεια οι εικόνες αυτές προστίθενται και η πρόσθεση των τριών πρωτευόντων χρωμάτων δημιουργεί τα υπόλοιπα χρώματα σε ποικίλες αναλογίες, όπως μπορούμε να διαπιστώσουμε και από την εικόνα 9. Για τον λόγο αυτό το πρότυπο RGB χαρακτηρίζεται ως προσθετικό. Η

απουσία όλων των χρωμάτων έχει ως αποτέλεσμα το μαύρο χρώμα, ενώ αντίθετα η παρουσία και των τριών χρωμάτων στην μέγιστη ποσότητα δημιουργεί το λευκό χρώμα. Αναμειγνύοντας τα χρώματα σε διαφορετικές ποσότητες μπορούμε να δημιουργήσουμε όλα τα ενδιάμεσα χρώματα στις διάφορες αποχρώσεις.

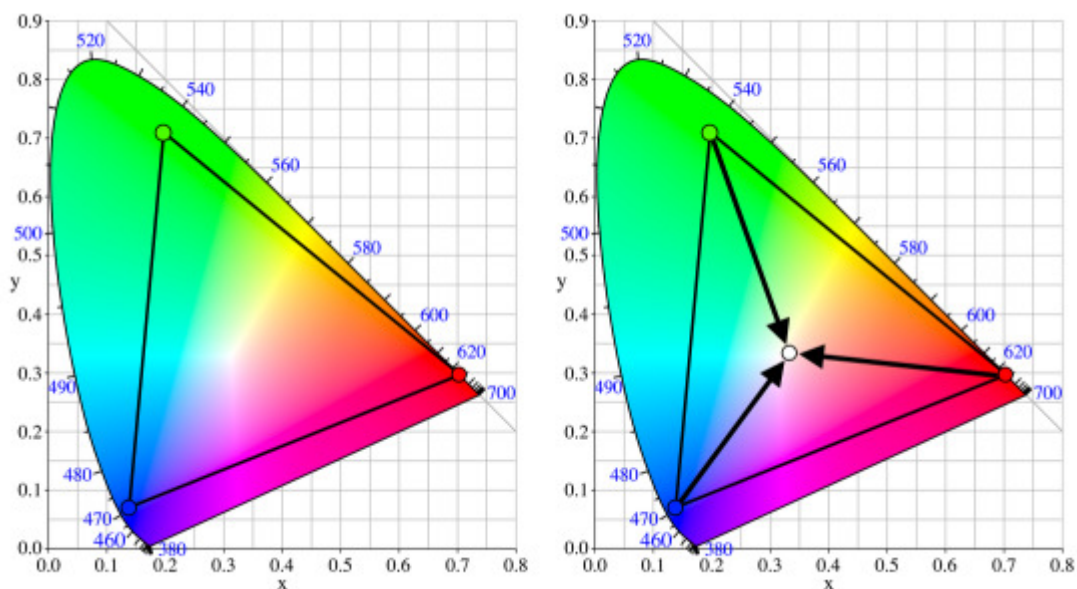
Ο χρωματικός χώρος RGB αντιστοιχεί σε ένα τρίγωνο όπως φαίνεται στο χρωματικό διάγραμμα 1. Όλα τα RGB χρώματα σχηματίζονται με την κατάλληλη ανάμειξη των προτύπων χρωμάτων που αντιστοιχούν στις κορυφές του τριγώνου. Συνεπώς όλα τα RGB χρώματα απεικονίζονται στο εσωτερικό του τριγώνου και συνεπώς όπως βλέπουμε και από το χρωματικό διάγραμμα 1, ο χρωματικός χώρος RGB δεν καλύπτει όλα τα χρώματα του οπτικού φάσματος.



Εικόνα 10. Πρόσθεση πρωτευόντων χρωμάτων

Οι περισσότεροι αλγόριθμοι οι οποίοι Εικόνα 9. Ενδιάμεσο χρώμα

χρησιμοποιούν χρώμα επεξεργάζονται τις RGB εικόνες ξεχωριστά ως εικόνες σε γκριζούς τόνους και στην συνέχεια τις προσθέτουν με τις αντίστοιχου χρώματος εικόνες για να πάρουν το τελικό αποτέλεσμα, όπως χαρακτηριστικά φαίνεται στην εικόνα 11.



Διάγραμμα 1. Περιοχή RGB

Το μοντέλο RGB χρησιμοποιείται όταν οι ακτίνες προσπίπτουν άμεσα στο μάτι του παρατηρητή, όπως οι οθόνες ηλεκτρονικών υπολογιστών και οι τηλεοράσεις, ενώ δε χρησιμοποιούνται σε περιπτώσεις απορρόφησης ή διάθλασης. Τα χρωματισμένα pixel παράγονται με ανάφλεξη κόκκινου, πράσινου και μπλε με βομβαρδισμό ηλεκτρονίων στον φώσφορο των οθονών. Το μοντέλο σχετίζεται στενά με τον τρόπο με τον οποίο αντιλαμβάνεται το χρώμα ο αμφιβληστροειδής του ανθρώπινου ματιού. Τέλος πρέπει να σημειωθεί ότι δεδομένου ότι το εύρος των χρωμάτων ποικίλει το μοντέλο εξαρτάται άμεσα από την συσκευή, η οποία το προβάλλει. Για παράδειγμα, μια εικόνα εμφανίζεται με πιο σκούρα χρώματα σε συστήματα με Windows, σε αντίθεση με τα συστήματα που χρησιμοποιούν MacOS, γιατί η σταθερή απόσταση των χρωμάτων

του πρότυπου RGB είναι πιο σκούρα στα Windows σε σχέση με τοMacOS.(Yam & Papadakis , 2003)



Εικόνα 11. Χρήση έγχρωμης εικόνας σε τόνους γκρι

Εκτός από το παραπάνω RGB χρωματικό μοντέλο έχουν προταθεί και άλλα RGB μοντέλα με κυριότερο το standardRGB (sRGB). Συγκεκριμένα, οι εταιρείες Hewlett-Packard και Microsoft ανέπτυξαν το sRGB χρωματικό μοντέλο για την χρησιμοποίηση του σε οθόνες, εκτυπωτές αλλά και στο διαδίκτυο. Στόχος ήταν να αναπτυχθεί ένα απλό χρωματικό μοντέλο που θα παρέχει τα ίδια χρώματα ανεξάρτητα από τις συσκευές που εφαρμόζεται. Με τον τρόπο αυτό μπορεί να επιτευχθεί ομοιόμορφη απεικόνιση και μετάδοση των εικόνων από τις διάφορες συσκευές και λογισμικά. Για παράδειγμα είναι επιθυμητό ό,τι βλέπουμε στην οθόνη ενός υπολογιστή ακριβώς αυτό να παίρνουμε χρωματικά κατά την εκτύπωση. Ο sRGB είναι ένας μη γραμμικός χρωματικός χώρος που χρησιμοποιείται σε πολλές εφαρμογές, συμπεριλαμβανομένων του διαδικτύου, του AdobePhotoshop, το MicrosoftWindows και πολλούς εκτυπωτές.

3. Μηχανική όραση

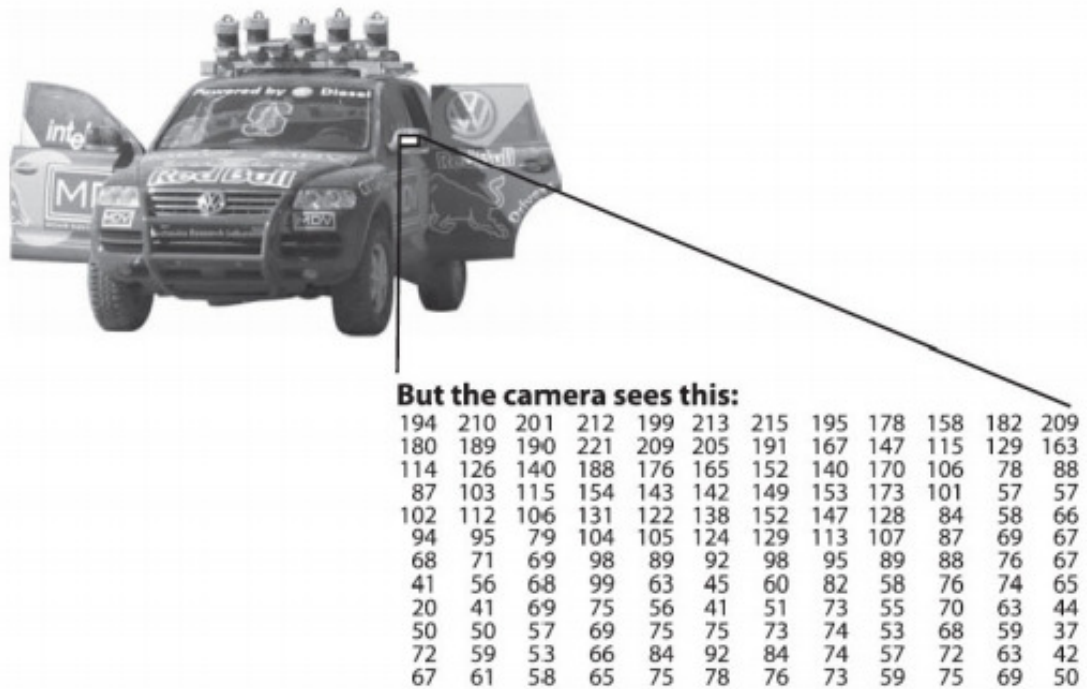
Με τον όρο μηχανική όραση εννοούμε την προσπάθεια εξομοίωσης της ανθρώπινης όρασης από κάποια συσκευή, όπως ένα ρομπότ ή άλλο ηλεκτρονικό σύστημα. Ουσιαστικά με την μηχανική όραση γίνεται λήψη της πληροφορίας που μας παρέχει μια εικόνα, ένα βίντεο, όψεις από πολλαπλές κάμερες ή πολυδιάστατες εικόνες από έναν ψηφιακό σαρωτή και στην συνέχεια, γίνεται επεξεργασία ώστε να παρθεί από το σύστημα μια απόφαση, η οποία οδηγεί σε κάποια ενέργεια. Αποτελεί μέρος της τεχνητής νοημοσύνης. Μια απόφαση μπορεί να αφορά τα σημεία στα οποία απεικονίζονται πρόσωπα ή η αναγνώριση ύπαρξης κάποιων συγκεκριμένων αντικειμένων από μια βάση. Μια ενέργεια, η οποία θα μπορούσε να γίνει ύστερα από την απόφαση είναι η απομόνωση των προσώπων από την υπόλοιπη εικόνα ή ο υπολογισμός ιστογραμμάτων βάση μιας απεικόνισης.

Στα συστήματα μηχανικής όρασης, ο υπολογιστής λαμβάνει ένα πίνακα αριθμών από μια κάμερα ή από το δίσκο. Πρέπει να φροντίσει να αναγνωρίσει ένα αντικείμενο με βάση ένα σύστημα αναγνώρισης προτύπων, ίσως να καθορίσει στην κάμερα την εστίαση, τον χρόνο κλείστρου και άλλες λεπτομέρειες που το ανθρώπινο μάτι έχει μάθει εμπειρικά να κάνει.

Η εικόνα 12 δείχνει την φωτογραφία ενός αυτοκινήτου. Ένα μέρος από τον καθρέπτη του οδηγού βρίσκεται σε πλαίσιο, όπως παρατηρούμε. Ο υπολογιστής όμως αντιλαμβάνεται το τμήμα αυτό της φωτογραφίας εντελώς διαφορετικά. Βλέπει έναν πίνακα ακεραίων αριθμών που περιγράφουν τη ζητούμενη πληροφορία. Η διαδικασία μηχανικής όρασης έγκειται στο να μετατρέψουμε αυτή την πληροφορία σε κάτι

εκμεταλλεύσιμο, και ειδικά στην πληροφορία στην οποία πραγματικά αντιστοιχεί, δηλαδή στο τμήμα του καθρέπτη του αυτοκινήτου.

Η μηχανική όραση μπορεί επίσης να περιγράψει ως συμπλήρωμα της βιολογικής όρασης. Όσον αφορά την βιολογική όραση μελετώνται ηοπτική αντίληψη στους ανθρώπους και τα ζώα με αποτέλεσμα μοντέλα για το πώς αυτά τα συστήματα λειτουργούν υπό το πρίσμα των φυσιολογικών διαδικασιών. Η μηχανική όραση από την άλλη μελετά και περιγράφει τα τεχνητά συστήματα όρασης που εφαρμόζονται σε λογισμικό ή/και σε υλικούπολογιστών. Η διεπιστημονική ανταλλαγή μεταξύ της βιολογικής και υπολογιστικής όρασης αποδεικνύεται όλο και περισσότερο καρποφόρα και για τους δύο τομείς.



Εικόνα 12. Αναπαράσταση τμήματος εικόνας στον υπολογιστή

Η μηχανική όραση επιδιώκει να εφαρμόσει θεωρίες και μοντέλα στην κατασκευή μηχανικών συστημάτων με δυνατότητα όρασης. Παραδείγματα εφαρμογών τέτοιων συστημάτων είναι τα εξής:

- Έλεγχος διαδικασιών (π.χ. ένα βιομηχανικό ρομπότ ή ένα αυτόνομο όχημα)
- Ανίχνευση συμβάντων (π.χ. οπτική επιτήρηση)
- Οργάνωση πληροφοριών (π.χ. ευρετηριοποίηση βάσεων δεδομένων και ακολουθιών εικόνων)
- Εξομοίωση αντικειμένων και περιβαλλόντων (π.χ. βιομηχανική επιθεώρηση, ιατρική ανάλυση εικόνας ή τοπογραφική εξομοίωση)
- Αλληλεπίδραση χρηστών με υπολογιστικά συστήματα (π.χ. ως είσοδος σε μια συσκευή επικοινωνίας ανθρώπου / μηχανής). (Wikipedia, 2013)

3.1. Ιστορική αναδρομή

Πριν από το 1980 η φυσική όραση είχε μελετηθεί σε βάθος, είχε μοντελοποιηθεί μαθηματικά και είχαν ήδη ξεκινήσει οι πρώτες προσπάθειες για την εξομοίωση της όρασης σε ρομπότ. Στις αρχές του 1980 αναπτύχθηκε ένα πεδίο της πληροφορικής, το οποίο καλείται ψηφιακή επεξεργασία εικόνας. Τότε εμφανίζεται ο όρος *υπολογιστική όραση*. Ο όρος *μηχανική όραση* μέχρι τότε ήταν γνωστός στην βιομηχανία και κυρίως στην ηλεκτρολογία και την ρομποτική. Κατά την διάρκεια της δεκαετίας του 1980 όπου ο κλάδος της ψηφιακής επεξεργασίας εικόνας βρισκόταν σε μεγάλη ανάπτυξη, οι δυο όροι σταδιακά ταυτίστηκαν. Πλέον το επιστημονικό αντικείμενο της μηχανικής όρασης ανήκει στον τομέα της τεχνητής νοημοσύνης, όπου βρίσκει εφαρμογή στην ρομποτική, στην ιατρική και σε δεκάδες ακόμα κλάδους.

Από τις αρχές της δεκαετίας του 1990 και μετά η μηχανική όραση μπαίνει στο επίκεντρο των επιστημόνων και γνωρίζει μεγάλη ανάπτυξη. Συνδέεται με την μηχανική μάθηση και υλοποιούνται αλγόριθμοι όρασης πραγματικού χρόνου σε τέτοιο βαθμό που κανείς μπορεί να τους συναντήσει σε ηλεκτρονικές συσκευές χαμηλού κόστους που απλά είναι εφοδιασμένες με κάμερα.

Στις μέρες μας ο προσανατολισμός της ανάπτυξης της μηχανικής όρασης είναι προς την τρίτη διάσταση. Πλέον η μηχανική όραση επεξεργάζεται και τα δεδομένα που αφορούν στο βάθος της εικόνας (μέσω στερεοσκοπικών καμερών ή αισθητήρες βάθους) για την επίτευξη του στόχου αυτού. Εφαρμογές σε τρισδιάστο επίπεδο βλέπουμε στο Kinect, ένα περιφερειακό που χρησιμοποιείται για την διασύνδεση μεταξύ χρηστών και υπολογιστικών συστημάτων, όπως και σε τρισδιάστες ταινίες και βιντεοπαιχνίδια.

3.2 Βασικές τεχνολογίες

Η τεχνητή νοημοσύνη γενικότερα και πιο ειδικά η μηχανική όραση βασίζονται σε αρκετές τεχνολογίες, όπως την επεξεργασία εικόνας και την αναγνώριση προτύπων. Το αρχικό στάδιο της μηχανικής όρασης είναι η κατανόηση της σκηνής, η οποία επιτυγχάνεται με την ψηφιακά επεξεργασία εικόνας. Η τεχνική της ψηφιακής επεξεργασίας εικόνας περιλαμβάνει κάθε μορφή αλγορίθμου που έχει ως στόχο την επεξεργασία, την ανάλυση και τον χειρισμό των ψηφιακών δεδομένων μιας εικόνας ή ενός βίντεο. Η ψηφιακή επεξεργασία εικόνας επικεντρώνεται σε δύο σημαντικές εργασίες:

- Την βελτίωση των πληροφοριών της εικόνας για καλύτερη ερμηνεία από τους ανθρώπους και
- Την επεξεργασία των δεδομένων της εικόνας για αποθήκευση, μετάδοση και αναγνώριση αυτόνομα από μηχανική αντίληψη (για παράδειγμα συστήματα μηχανικής όρασης).

Η συνέχεια από την επεξεργασία εικόνας στην όραση του υπολογιστή μπορεί να διασπαστεί σε διαδικασίες χαμηλού, μεσαίου και υψηλού επιπέδου. Κατά την επεξεργασία χαμηλού επιπέδου η είσοδος είναι μια εικόνα, ενώ η έξοδος είναι η ίδια η εικόνα απουσία θορύβου ή μεταλλαγμένη ως προς κάποιο χαρακτηριστικό της (για παράδειγμα έχει περιστραφεί ή έχει αυξομειωθεί στο πλάτος ή/και στο μήκος της. Στην επεξεργασία μεσαίου επιπέδου η είσοδος είναι μια εικόνα που έχει προκύψει από το χαμηλό επίπεδο επεξεργασίας και η έξοδος είναι κάποια γνωρίσματά της, όπως αναγνώριση κάποιων αντικείμενων που υπάρχουν σε αυτή ή τμηματοποίηση της σε αντικείμενα. Τέλος η υψηλού επιπέδου επεξεργασία δέχεται στην είσοδο τα γνωρίσματα του προηγούμενου επιπέδου και η έξοδος της είναι η κατανόηση μιας σκηνής ή η αυτόνομη πλοήγηση (εικόνα 13).



Εικόνα 13. Αυτόνομο σύστημα πλοήγησης του Πανεπιστημίου CarnegieMellon

Η μηχανική όραση επίσης έχει βασιστεί στον επιστημονικό κλάδο της πληροφορικής που ασχολείται με τα γραφικά των υπολογιστών. Τα γραφικά των υπολογιστών είναι η τεχνολογία αλγοριθμικής σύνθεσης εικόνων σε ηλεκτρονικό υπολογιστή. Πρόκειται για εικόνες και βίντεο που δημιουργούνται με την βοήθεια προγραμμάτων και εργαλείων που χρησιμοποιούν δεδομένα εικόνας για να συνθέσουν μια νέα. Η είσοδος των υπολογισμών είναι οι περιγραφές μιας οπτικής σκηνής, ενώ η έξοδος είναι οι ψηφιακές εικόνες ή τα βίντεο με ή χωρίς αλληλεπίδραση από τον χρήστη.

Τα γραφικά υπολογιστών απεικονίζονται σε δυο ή σε τρεις διαστάσεις (2-D και 3-D αντίστοιχα). Τα δισδιάστατα γραφικά μπορεί να είναι διανυσματικά (vectorgraphics), δηλαδή αποθηκεύονται ως μαθηματικές διανυσματικές εξισώσεις, οι οποίες περιγράφουν γεωμετρικά σχήματα. Το πλεονέκτημα με αυτόν τον τρόπο αποθήκευσης και αναπαράστασης είναι ότι οι εικόνες μπορούν να αλλάξουν μέγεθος (να αυξομειωθούν οι διαστάσεις τους) χωρίς να υπάρχει καμία απώλεια της ποιότητας της (εικόνα 14β). Η άλλη κατηγορία δισδιάστατων γραφικών είναι τα γραφικά ψηφιδών (bitmapgraphics). Τα γραφικά αυτά προκύπτουν από την απεικόνιση μιας ήδη υπάρχουσας εικόνας για παράδειγμα μιας φωτογραφίας ή μιας εικόνας από σαρωτή. Με αλλαγή των διαστάσεων των εικόνων bitmap σε μέγεθος μεγαλύτερο της ανάλυσης τους προκύπτει μια κοκκοποίηση της εικόνας (εικόνα 14α). Όλα τα χαρακτηριστικά που περιγράψαμε παραπάνω για την αναπαράσταση μιας εικόνας, όπως η ανάλυση, το βάθος χρώματος και το πρότυπο RGB αφορούν τις εικόνες bitmap.



Εικόνα 14. α. 2-D εικόνα bitmap, β. 2-D εικόνα vector

Τα τρισδιάστατα γραφικά (3-D graphics) αφορούν απεικονίσεις εικόνωντριών διαστάσεων στιςδύο διαστάσεις που έχει μια οθόνη και με αυτόν τον τρόπο εμφανίζονται πολύ ρεαλιστικά στον χρήστη. Τα αντικείμενα αυτά δημιουργούνται μέσω μαθηματικών τύπων που δείχνουν τα σημεία από τα οποία αποτελούνται τα αντικείμενα σε κάποιο σύστημα συντεταγμένων και στην συνέχεια προβάλλονται σε σύστημα συντεταγμένων δύο διαστάσεων.

Μια άλλη τεχνολογία πάνω στην οποία βασίστηκε η μηχανική όραση είναι η αναγνώριση προτύπων. Στόχος της αναγνώρισης προτύπων είναι να ταξινομή σε κατηγορίες ή να διαχωρίζει σε ομάδες με βάση κάποια κριτήρια τα δεδομένα που λαμβάνει ως είσοδο. Αυτό γίνεται αυτόματα με την χρήση αλγορίθμων που έχουν δημιουργηθεί για την υλοποίηση της ταξινόμησης αυτής. Η ταξινόμηση μπορεί να γίνει σε εικόνες, σχήματα (σε κυματομορφές), βίντεοή οποιοδήποτε άλλο είδος χρειάζεται να ταξινομηθεί για περαιτέρω επεξεργασία.(Κωτσιόπουλος, 2014)

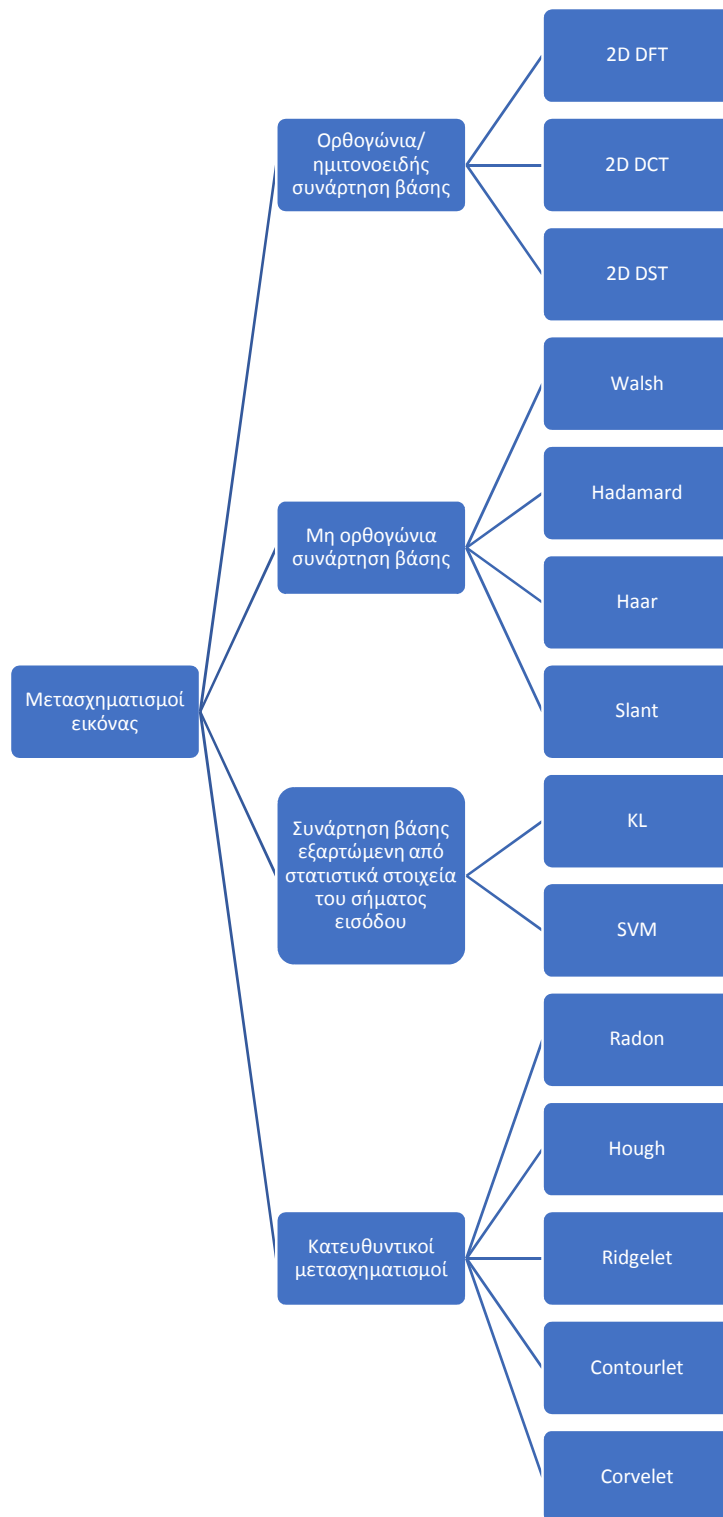
Τέλος, η μηχανική όραση αποτελεί επέκταση της μηχανικής μάθησης. Το 1959, ο Arthur Samuel ορίζει τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί" (Simon, 2013). Το πεδίο αυτό της επιστήμης των υπολογιστών αναπτύσσει αλγορίθμους, οι οποίοι μαθαίνουν από τα δεδομένα εισόδου και στην συνέχεια καθίστανται ικανοί να κάνουν προβλέψεις βάσει των δεδομένων.

4. Τα χαρακτηριστικά του Haar

Για να διασπάσουμε μια εικόνα σε επιμέρους όρους πρέπει να εφαρμόσουμε σε αυτή δισδιάστατους μετασχηματισμούς με την βοήθεια ενός συνόλου πινάκων βάσης που καλούνται εικόνες βάσης. Οι εφαρμογές μετασχηματισμών στην επεξεργασία εικόνας είναι πολυάριθμες. Οι σημαντικότερες από αυτές αφορούν:

- Το φιλτράρισμα των εικόνων στο πεδίο της συχνότητας
- Την εξαγωγή διαφόρων χαρακτηριστικών από τις εικόνες, όπως για παράδειγμα η ανάλυση της υφής των εικόνων και η ανίχνευση ακμών.
- Μια άλλη εφαρμογή είναι η κωδικοποίηση και η συμπίεση εικόνας, όπου η μείωση του εύρους ζώνης επιτυγχάνεται συνήθως με την κατάλληλη διαγραφή (μηδενισμό) των συντελεστών μετασχηματισμού που έχουν αμελητέες τιμές.

Στο παρακάτω διάγραμμα² παρατηρούμε μια κατηγοριοποίηση των σπουδαιότερων μετασχηματισμών.



Διάγραμμα 2. Μετασχηματισμοί εικόνας

Στην παρούσα εργασία θα αναλύσουμε μόνο τον μετασχηματισμό Haar, ο οποίος ανήκει στην κατηγορία των μετασχηματισμών με μη ορθογώνια συνάρτηση βάσης. Ο μετασχηματισμός Haar αποτελεί την πρώτη θεώρηση των κυματιδίων (wavelets)

και είναι το απλούστερο δυνατό κυματίδιο. Ο ορθογώνιος πίνακας Haar περιέχει μόνο τα στοιχεία $-1, 0$ ή 1 πολλαπλασιασμένα με ακέραιες δυνάμεις του $\sqrt{2}$. Το κύριο πλεονέκτημα του μετασχηματισμού αυτού συνίσταται στο ότι δειγματολειτουργεί τα δεδομένα εισόδου σε πολλαπλές αναλύσεις με χαμηλή έως υψηλή ανάλυση και υπολογίζει τις διαφορές γειτονικών ζευγών. Η ανάλυση κλιμακώνεται με δυνάμεις του 2 και αυτό έχει ως αποτέλεσμα η διαφορική ενέργεια να ανιχνεύεται όλο και πιο τοπικά. Το υπολογιστικό κόστος του μετασχηματισμού είναι σχετικά μικρό αφού ο μετασχηματισμός ενός διανύσματος N στοιχείων απαιτεί μόνο $2(N-1)$ προσθέσεις και N πολλαπλασιασμούς. Μερικές από τις εφαρμογές του μετασχηματισμού Haar είναι η συμπίεση εικόνων και video, η εξαγωγή χαρακτηριστικών υφής, η απομάκρυνση του θορύβου από δεδομένα με ανεπιθύμητο θόρυβο κτλ.

Ο μετασχηματισμός Haar βασίζεται στις συναρτήσεις $Haar_{h_k}(x)$ οι οποίες ορίζονται συνήθως στο διάστημα $[0,1]$ και για $k=0,1,\dots, N-1$ όπου $N=2^n$ με $n=1,2,3,\dots$. Πριν προχωρήσουμε στην περιγραφή των συναρτήσεων πρέπει να σημειώσουμε ότι κάθε ακέραιος αριθμός k μπορεί να εκφραστεί ως

$$k = 2^p + q - 1$$

Όπου $0 \leq p \leq n-1$, $q=0$ ή 1 για $p=0$ και $1 \leq q \leq 2^p$ για $p \neq 0$.

Για παράδειγμα, αν $N=4=2^2$ τότε $k=0,1,2,3$ ενώ τα p,q θα είναι ίσα με

$$0=2^0+0-1, 1=2^0+1-1, 2=2^1+1-1 \text{ και } 3=2^1+2-1$$

Δηλαδή

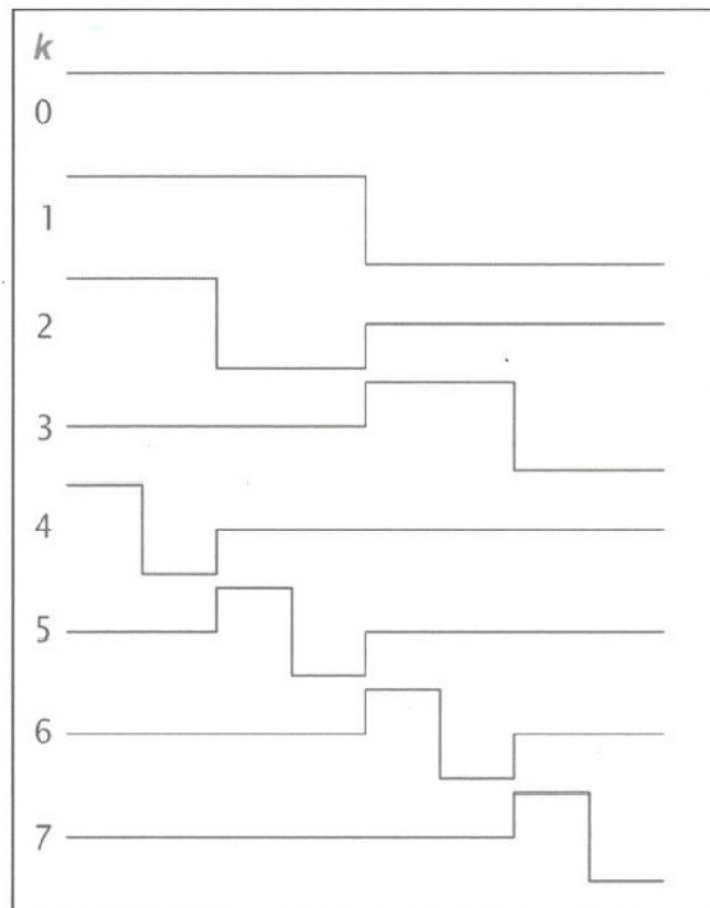
k	0	1	2	3
p	0	0	1	1
q	0	1	1	2

Οι συναρτήσεις Haar μπορούν τώρα να περιγραφούν ως εξής:

$$h_0(x) = h_{00}(x) = \frac{1}{\sqrt{N}}, \text{ όπου } 0 \leq x \leq 1$$

$$h_k(x) = h_{pq}(x) = \frac{1}{\sqrt{N}} \begin{cases} (\sqrt{2})^p, & \text{για } \frac{q-1}{2^p} \leq x \leq \frac{q-\frac{1}{2}}{2^p} \\ -(\sqrt{2})^p, & \text{για } \frac{q-\frac{1}{2}}{2^p} \leq x \leq \frac{q}{2^p} \\ 0, & \text{διαφορετικά για } x \in [0,1] \end{cases}$$

Στην παρακάτω εικόνα παρουσιάζονται οι μορφές των συναρτήσεων $h_k(x)$ για $N=8$.



Εικόνα 15. Μονοδιάστατες συναρτήσεις Haar

Με την κατάλληλη δειγματοληψία των συναρτήσεων $h_k(x)$ δημιουργούνται οι πίνακες Haar, που για $N=2,4,8$ είναι οι ακόλουθοι

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Αν πολλαπλασιάσουμε τον πίνακα H_8 με έναν μονοδιάστατο πίνακα V διαστάσεων 8×1 , μπορούμε να παρατηρήσουμε ότι

- Η πρώτη γραμμή αντιστοιχεί ουσιαστικά στην μέση τιμή του πίνακα V (πολλαπλασιασμένη με $\sqrt{8}$).
- Η δεύτερη γραμμή αντιστοιχεί στην διαφορά των μέσων τιμών του πρώτου μισού από το δεύτερο μισό του πίνακα V .
- Η τρίτη και η τέταρτη γραμμή υπολογίζουν την διαφορά μεταξύ του πρώτου με το δεύτερο και του τρίτου με το τέταρτο τεταρτημόριο του πίνακα V .
- Τέλος, οι τέσσερις τελευταίες γραμμές αντιστοιχούν στην διαφορά γειτονικών τιμών του πίνακα V .

Παρατηρούμε ότι ο πίνακας Haar λειτουργεί διαφορετικά ανάλογα με τις περιοχές του σήματος εισόδου. Σε ορισμένες περιοχές συμπεριφέρεται ως χαμηλοπερατό φίλτρο ενώ σε άλλες ως υψιπερατό. Τα χαμηλότερα φίλτρα αποκόπτουν τις χαμηλές συχνότητες του σήματος εισόδου ενώ τα υψιπερατά αποκόβουν τις υψηλές συχνότητες του σήματος εισόδου. Με αυτό τον τρόπο μπορούμε να ανακαλύψουμε περιοχές της εικόνας όπου το σήμα αλλάζει πολύ γρήγορα.

Ο διδιάστατος διακριτός μετασχηματισμός Haar μπορεί να υπολογιστεί με την χρήση του πίνακα Haar όπως φαίνεται στην παρακάτω σχέση:

$$V=HfH^T$$

Ανάλογα ο αντίστροφος μετασχηματισμός μπορεί να εκφραστεί ως:

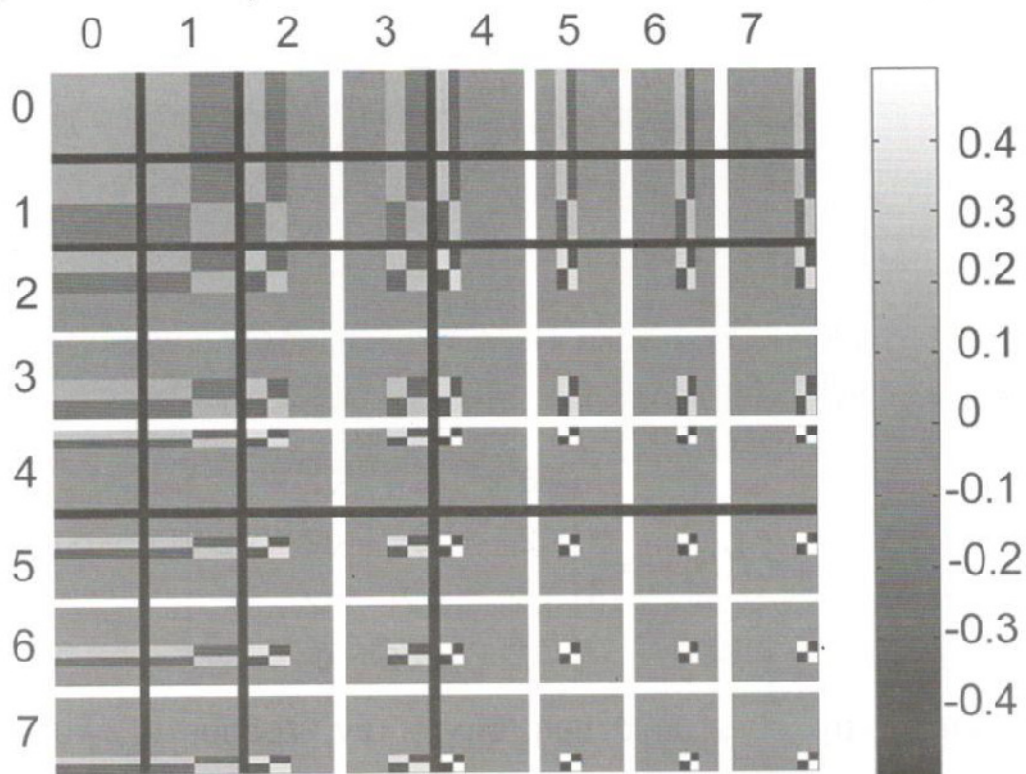
$$f=H^T V H$$

Οι εικόνες βάσης του διδιάστατου διακριτού μετασχηματισμού Haar για μια εικόνα με διαστάσεις 8x8 παρουσιάζονται παρακάτω:

$$H(3,x)H(2,x)^T = \frac{1}{8} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2} \\ \sqrt{2} \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} [\sqrt{2} \quad \sqrt{2} \quad -\sqrt{2} \quad -\sqrt{2} \quad 0 \quad 0 \quad 0 \quad 0] =$$

$$= \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & -2 & -2 & 0 & 0 & 0 & 0 \\ 2 & 2 & -2 & -2 & 0 & 0 & 0 & 0 \\ -2 & -2 & 2 & 2 & 0 & 0 & 0 & 0 \\ -2 & -2 & 2 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Από τις εικόνες βάσης του μετασχηματισμού Haar της εικόνας που ακολουθεί η υποεικόνα 0,0 αντιστοιχεί στην μέση τιμή της εικόνας. Αντίθετα οι εικόνες βάσης υψηλού βαθμού από (4,4) έως (7,7) που αντιστοιχούν στις υψηλές συχνότητες της εικόνας έχουν την ίδια δομική περιγραφή (pattern) διαστάσεων 2x2 η οποία μετατοπίζεται προκειμένου να ανιχνευτούν οι λεπτομέρειες της εικόνας. Επίσης, μπορούμε να παρατηρήσουμε ότι τα διάφορα σχέδια στις εικόνες έχουν διαφορετική ανάλυση και συγκεκριμένα ξεκινούν από μέγεθος 8x8 στην εικόνα 0,0 (χαμηλές συχνότητες) και μειώνονται σε εικόνες μεγέθους 2x2 στην κάτω δεξιά περιοχή (υψηλές συχνότητες). Έτσι οι εικόνες βάσης όπως φαίνεται στην εικόνα 16 μπορούν να χωριστούν ανάλογα με την ανάλυση των σχεδίων τους σε περιοχές.



Εικόνα 16. Εικόνες βάσης του 2DHaar για $N=8$

Επειδή μεγάλο πλήθος εικόνων βάσης αναφέρονται στις υψηλές συχνότητες, δηλαδή στις λεπτομέρειες της εικόνας, μπορούμε πολλές φορές να θεωρήσουμε τους αντίστοιχους συντελεστές αμελητέους (να τους μηδενίσουμε). Με αυτόν τον τρόπο επιτυγχάνεται σημαντική συμπίεση της πληροφορίας χωρίς σημαντική παραμόρφωση. Φυσικά, η ανασύνθεση της εικόνας γίνεται με εφαρμογή του αντίστροφου διακριτού μετασχηματισμού Haar. Έτσι μπορούμε να ανασυνθέσουμε με διαφορετικές λεπτομέρειες διαφορετικές περιοχές της εικόνας.

Τρεις είναι οι κύριοι τρόποι επιλογής των συντελεστών που θα μηδενιστούν. Ο απλούστερος εξ' αυτών βασίζεται στην τυχαία επιλογή τμημάτων του φάσματος και τον μηδενισμό των υπόλοιπων συντελεστών. Σύμφωνα με τον δεύτερο τρόπο, επιλέγεται μια τιμή κατωφλίου και όσοι συντελεστές στο φάσμα έχουν πλάτος μικρότερο αυτής της τιμής μηδενίζονται. Τέλος, ο τρίτος τρόπος βασίζεται στην συνολική ενέργεια στο φάσμα. Αρχικά όλοι οι συντελεστές ταξινομούνται σύμφωνα με αυτό το πλάτος τους από τον μεγαλύτερο προς τον μικρότερο. Στην συνέχεια, ξεκινώντας από τον μεγαλύτερο σε πλάτος συντελεστή αθροίζουμε τα τετράγωνά τους. Το άθροισμα αυτό εκφράζει την ενέργεια στο φάσμα, οπότε όταν το άθροισμα αυτό φτάσει μια επιθυμητή τιμή μηδενίζουμε στην συνέχεια τους υπόλοιπους συντελεστές. Πρέπει πάντως να τονίσουμε ότι ο μετασχηματισμός Haar παρουσιάζει χαμηλή ενεργειακή συμπίεση.

5. Μηχανική όραση και ταξινομητές

5.1 Εισαγωγή στους ταξινομητές

Η ευκολία με την οποία διακρίνουμε ένα πρόσωπο, διακρίνουμε τους χαρακτήρες, ξεχωρίζουμε τα κλειδιά μας ακόμα και μόνο με την αφή και αποφασίζουμε αν ένα φρούτο είναι ώριμο από την μυρωδιά του, είναι διαδικασίες αναγνώρισης προτύπων (patternrecognition). Ο άνθρωπος επιλύει τα προβλήματα αυτά με την εκπαίδευση που έχουν υποστεί τα δισεκατομμύρια κύτταρα του εγκεφάλου του. Στην περιοχή της ψηφιακής επεξεργασίας εικόνων, το πρόβλημα της αναγνώρισης προτύπων ανάγεται στην ταξινόμηση (clustering) ενός συνόλου κατάλληλων και ισχυρών χαρακτηριστικών σε κλάσεις (classes) με την βοήθεια κατάλληλων ταξινομητών (classifiers). Το διάνυσμα των χαρακτηριστικών που χρησιμοποιείται σε ένα σύστημα ταξινόμησης καλείται *πρότυπο* (pattern). Οι ταξινομητές είναι ουσιαστικά αλγόριθμοι που χρησιμοποιούνται για να επιτύχουμε την ταξινόμηση των προτύπων σε κατά το δυνατόν ομογενείς (από την άποψη των χαρακτηριστικών) κλάσεις. Η ταξινόμηση γίνεται συνήθως είτε με επίβλεψη (supervisedclustering) είτε χωρίς επίβλεψη (non-supervisedclustering). Στην πρώτη περίπτωση οι κλάσεις δημιουργούνται καθορίζοντας την επιθυμητή απόκριση του ταξινομητή για κάθε διάνυσμα χαρακτηριστικών (πρότυπο). Στην δεύτερη περίπτωση, οι κλάσεις δημιουργούνται αυτόματα χωρίς να παρέχεται στον ταξινομητή καμία πληροφορία για το που ανήκει κάθε διάνυσμα χαρακτηριστικών.

Στην ψηφιακή επεξεργασία εικόνων χρησιμοποιούμε ταξινομητές που κύρια ανήκουν σε μια από τις ακόλουθες κατηγορίες:

- Στατιστικούς ταξινομητές

- Νευρωνικούς ταξινομητές
- Ταξινομητές που βασίζονται σε έναν ιεραρχικό αλγόριθμο (hierarchicalclustering).
- Τους ταξινομητές που χρησιμοποιούν σαφή (crisp) δεδομένα και σε αυτούς που βασίζονται στην ασαφή λογική (fuzzylogic), οι οποίοι ονομάζονται ασαφείς ταξινομητές
- Ο συνδυασμός νευρωνικών δικτύων και ασαφούς λογικής οδηγεί σε υβριδικούς ταξινομητές, τους επονομαζόμενους νευρο- ασαφείς ταξινομητές (neuro- fuzzyclassifiers).
- Μια άλλη κατηγοριοποίηση αφορά τους αυτό- οργανωμένους (self-organized) ή μη ταξινομητές
- Τέλος, μπορούμε να διακρίνουμε τους ταξινομητές στους οποίους είναι δεδομένος από την αρχή ο αριθμός των κλάσεων και σε αυτούς που έχουν την δυνατότητα προσδιορισμού του βέλτιστου αριθμού των κλάσεων κατά την διαδικασία ταξινόμησης.

Η επιλογή του κατάλληλου ταξινομητή σχετίζεται άμεσα με το είδος των χαρακτηριστικών καθώς και το είδος της εφαρμογής όπου θα χρησιμοποιηθεί. Συνεπώς, κατά τον σχεδιασμό μιας εφαρμογής η επιλογή των κατάλληλων χαρακτηριστικών και του κατάλληλου ταξινομητή πρέπει να γίνει ταυτόχρονα. Παρόλα αυτά, υπάρχουν περιπτώσεις όπου οι ταξινομητές, μετά την αρχική τους χρησιμοποίηση, μπορούν να μας βοηθήσουν να συμπεράνουμε για το πόσο ισχυρά ή μη είναι ορισμένα χαρακτηριστικά που χρησιμοποιήθηκαν.

Στους αλγορίθμους που στηρίζονται στην *σαφή ταξινόμηση* (crispclustering) μας δίνονται M διανύσματα χαρακτηριστικών για παράδειγμα X_1, X_2, \dots, X_M που ανήκουν σε ένα σύνολο S και αναζητούμε τις περιοχές (κλάσεις) S_1, S_2, \dots, S_K έτσι ώστε κάθε διάνυσμα X_i με $i=1, 2, \dots, M$ να τοποθετηθεί σε μόνο μια κλάση.

Ο υπολογισμός της ομοιότητας μεταξύ των χαρακτηριστικών είναι δομικό στοιχείο των τεχνικών ταξινόμησης. Μπορούμε να δεχτούμε ότι ένα διάνυσμα χαρακτηριστικών είναι κοντύτερα σε κάποιο άλλο όταν κάποιο μέτρο ομοιότητας έχει μικρότερη τιμή. Έτσι η μέτρηση ομοιότητας, δηλαδή η εύρεση της απόστασης μεταξύ των χαρακτηριστικών, μπορεί να χρησιμοποιηθεί προκειμένου να ταξινομήσουμε χαρακτηριστικά σε προκαθορισμένες κλάσεις. Η διαδικασία ταξινόμησης με βάση το κριτήριο της ελάχιστης απόστασης οδηγεί στους ταξινομητές της ελάχιστης απόστασης. Είναι συνεπώς αναγκαίο να καθορίσουμε επακριβώς το μέτρο ομοιότητας που θα χρησιμοποιήσουμε, γιατί διαφορετικά μέτρα μπορεί να οδηγήσουν σε διαφορετικές ταξινομήσεις.

Ο ταξινομητής *k-means* στηρίζεται στην ελαχιστοποίηση του αθροίσματος των τετραγώνων των αποστάσεων όλων των στοιχείων σε κάθε κλάση από το κέντρο της κλάσης αυτής (Bow&Tze, 1992), δηλαδή

$$\min \sum_{x \in S_{j(k)}} |x - z_j|^2$$

Όπου $S_{j(k)}$ είναι η κλάση στην k επανάληψη του αλγορίθμου, z_j είναι το κέντρο της κλάσης και ως μέτρο της απόστασης χρησιμοποιείται η Ευκλείδεια απόσταση.

Σε αντίθεση με τον *k-means* ο *c-means* ταξινομητής τοποθετεί και το M στοιχεία που έχουμε κατευθείαν στις K κλάσεις και στην συνέχεια προσδιορίζονται τα κέντρα των

κλάσεων. Τα υπόλοιπα στοιχεία του αλγορίθμου αυτού (κριτήριο ταξινόμησης, σύγκλιση κτλ) είναι ταυτόσημα με τον αλγόριθμο k-means.

Ο ταξινομητής Maxmin είναι ιδιαίτερα χρήσιμος διότι μας επιτρέπει να εκτιμήσουμε το πλήθος των κλάσεων και φυσικά να ταξινομήσουμε τα χαρακτηριστικά στις κλάσεις. Λόγω της αυτόματης και χωρίς επίβλεψη διαδικασίας προσδιορισμού του πλήθους των κλάσεων μπορεί να χρησιμοποιηθεί ως πρώτο στάδιο άλλα και αρχικό σημείο άλλων ταξινομητών, όπως ο k-means που αναφέραμε παραπάνω.

Ο ταξινομητής *ISODATA* θεωρείται μια βελτιστοποίηση των μεθόδων k-means και c-means. Το κριτήριο ταξινόμησης παραμένει πάλι η ελαχιστοποίηση της συνάρτησης:

$$\min \sum_{x \in S_j^{(k)}} |x - z_j|^2$$

Όμως θα πρέπει να ικανοποιούνται και κάποιες άλλες προϋποθέσεις. Ο αριθμός των κλάσεων δεν είναι αυστηρά καθορισμένος και αναφέρεται μόνο ο μέγιστος αριθμός κλάσεων (M). Επίσης, απορρίπτονται κλάσεις με λίγα στοιχεία (καθορίζεται ο ελάχιστος αριθμός στοιχείων σε κάθε κλάση). Επίσης καθορίζεται η μέγιστη τυπική απόκλιση έτσι ώστε παρόμοια στοιχεία να βρίσκονται «κοντά» και η ελάχιστη απόσταση μεταξύ των κλάσεων με στόχο τα ανόμοια στοιχεία να βρίσκονται σε μακρινή απόσταση.

Τα νευρωνικά δίκτυα αποτελούν σημαντικό πεδίο έρευνας τα τελευταία χρόνια. Οι ανθρώπινοι εγκέφαλοι περιέχουν περίπου $1,5 \times 10^{12}$ νευρώνες διαφόρων τύπων, ο καθένας από τους οποίους λαμβάνει σήματα μέσω έως και 104 συνάψεων. Η

απόκριση ενός νευρώνα προκύπτει σε περίπου 1- 10 χιλιοστά του δευτερολέπτου. Η λειτουργία αυτή μας επιτρέπει να αναγνωρίζουμε σύνθετα αντικείμενα πολύ γρήγορα. Για παράδειγμα μπορούμε να αναγνωρίσουμε το πρόσωπο ενός παλαιού φίλου σε μόλις 0.1 δευτερόλεπτα. Αυτό είναι ένας σύνθετος στόχος αναγνώρισης που πρέπει να εκτελεστεί με έναν ιδιαίτερα παράλληλο τρόπο δεδομένου ότι η αναγνώριση γίνεται σε περίπου 100 με 1000 βήματα. Αυτό σημαίνει ότι μόνον παράλληλα συστήματα μπορούν να επιτελέσουν παρόμοιες διεργασίες αναγνώρισης. Τα Τεχνητά Νευρωνικά Δίκτυα αποτελούνται από έναν μεγάλο αριθμό διασυνδεδεμένων μονάδων που τους δίνουν την δυνατότητα να επεξεργαστούν τις πληροφορίες με παράλληλο τρόπο.

Όπως καθένας από τους 10^{12} νευρώνες του ανθρώπινου εγκεφάλου ενσωματώνει τις εισερχόμενες πληροφορίες από όλους τους άλλους νευρώνες που συνδέονται άμεσα ή έμμεσα με αυτόν, ένας τεχνητός νευρώνας αθροίζει όλες τις εισόδους και δημιουργεί μια έξοδο, η οποία μεταφέρει πληροφορίες στους άλλους νευρώνες. Η σύνδεση από τους δενδρίτες ενός νευρώνα ή το σώμα κυττάρων στις λειτουργίες ενός άλλου νευρώνα καλείται σύναψη. Η δύναμη με την οποία δυο νευρώνες επηρεάζουν ο ένας τον άλλον καλείται συνοπτικό βάρος. Σε ένα νευρωνικό δίκτυο όλοι οι νευρώνες συνδέονται με άλλους νευρώνες με συνάψεις των οποίων τα συναπτικά βάρη του μπορούν να έχουν αυθαίρετες τιμές. Στην πραγματικότητα, οι τιμές των συναπτικών βαρών αντιστοιχούν και αντιπροσωπεύουν το αποτέλεσμα ενός ερεθίσματος στο νευρωνικό δίκτυο και παρέχουν με τον τρόπο αυτό στο νευρωνικό δίκτυο την δυνατότητα αναγνώρισης του συγκεκριμένου ερεθίσματος.

Τα Τεχνητά Νευρωνικά Δίκτυα χαρακτηρίζονται από:

- Την τοπολογία του δικτύου

- Τα χαρακτηριστικά των νευρώνων (κόμβων του δικτύωματος) και
- Τους κανόνες εκπαίδευσης ή μάθησης

Η κύρια διαφορά των *ασαφών ταξινομητών* (fuzzyclustering) σε σχέση με τους σαφείς ταξινομητές έγκειται στο ότι κάθε στοιχείο πλέον μπορεί να ανήκει σε περισσότερες από μια κλάσεις. Συγκεκριμένα έχουμε M στοιχεία x_1, x_2, \dots, x_M που ανήκουν στο σύνολο S και αναζητούμε τις περιοχές (κλάσεις) S_1, S_2, \dots, S_K έτσι ώστε κάθε x_i με $i=1, 2, \dots, M$ να ταξινομηθεί σε κλάσεις όπου η ένωση τους μας δίνει το σύνολο S , χωρίς όμως να ισχύει κατ' ανάγκη ότι η τομή δυο κλάσεων παράγει το κενό σύνολο.

Η παραπάνω προσέγγιση είναι πρακτικά πολύ συνηθισμένη και χρήσιμη. Έστω λοιπόν μια διαδικασία ταξινόμησης ανδρών με βάση το ύψος τους στις κλάσεις $\{K, M, \Psi\}$, δηλαδή κοντός, μέτριος και ψηλός αντίστοιχα. Η ταξινόμηση μιας παρατήρησης του δείγματός μας με ύψος για παράδειγμα 1,82 m δύσκολα θα μπορούσε να αποδοθεί μόνο στην κλάση M ή μόνο στην κλάση Ψ . Θα ήταν ίσως πιο κοντά στην πραγματικότητα αν λέγαμε ότι ο άνδρας αυτός ανήκει κατά 70% στην κλάση M και κατά 30% στην κλάση Ψ , δηλαδή ότι είναι μάλλον μέτριος.

5.2 Ενίσχυση απόδοσης- Αλγόριθμος Adaboost

Η ενδυνάμωση (boosting) είναι μια τεχνική που προσπαθεί να δημιουργήσει έναν ισχυρό ταξινομητή από έναν αριθμό απλών ταξινομητών. Αυτό γίνεται με την κατασκευή ενός μοντέλου από τα δεδομένα εκπαίδευσης, στη συνέχεια δημιουργείται ένα δεύτερο μοντέλο που επιχειρεί να διορθώσει τα σφάλματα του πρώτου μοντέλου

κ.ο.κ. Τα μοντέλα προστίθενται έως ότου προβλεφθεί απόλυτα το σετ εκπαίδευσης ή προστεθεί ένας μέγιστος αριθμός μοντέλων.

Ο αλγόριθμος AdaBoost ήταν ο πρώτος πολύ επιτυχημένος αλγόριθμος ενίσχυσης που αναπτύχθηκε για τη δυαδική ταξινόμηση. Είναι το καλύτερο σημείο εκκίνησης για την κατανόηση της ενίσχυσης.

Ο αλγόριθμος AdaBoost χρησιμοποιείται καλύτερα για να ενισχύσει την απόδοση των δέντρων αποφάσεων σε δυαδικά προβλήματα ταξινόμησης. Ο αλγόριθμος αυτός μπορεί να χρησιμοποιηθεί για την ενίσχυση της απόδοσης οποιουδήποτε αλγορίθμου εκμάθησης μηχανών. Χρησιμοποιείται καλύτερα με τους απλούς αλγορίθμους ταξινόμησης (ή ασθενείς αλγορίθμους ταξινόμησης), οι οποίοι δεν ταξινομούν καλά τα δεδομένα απλά πετυχαίνουν μια ταξινόμηση λίγο καλύτερη του 50% επιτυχίας. Ο αλγόριθμος AdaBoost συνδυάζει ένα σύνολο από ασθενείς συναρτήσεις ταξινόμησης με έναν άπληστο αλγόριθμο για να δημιουργήσει τελικά έναν πιο ισχυρό ταξινομητή.

Συγκεκριμένα στα προβλήματα αναγνώρισης εικόνων ο αλγόριθμος AdaBoost είναι εκείνος που επιλέγει τα βασικά χαρακτηριστικά από ένα σύνολο χαρακτηριστικών που θα χρησιμοποιηθούν από τον ταξινομητή. Αυτό αποτελεί αναγκαία διαδικασία, γιατί αν χρησιμοποιούσαμε το πλήρες σύνολο των χαρακτηριστικών η ανίχνευση μιας εικόνας θα ήταν απαγορευτικά χρονοβόρα.

Ο ψευδοκώδικας υλοποίησης του αλγορίθμου Adaboost(Κοτσιας, 2017) παρατίθεται παρακάτω:

1. Έστω ένα σύστημα από παραδείγματα εικόνων $(x_1, y_1) \dots (x_n, y_n)$ όπου $y_i=0,1$ για negative και positive παραδείγματα αντίστοιχα

2. Αρχικοποίηση των βαρών για τα $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}, y_i = 0, 1$, όπου m και l είναι ο αριθμός των negatives και positives αντίστοιχα.

3. Για $t=1, \dots, T$

α) Κανονικοποίηση των βαρών

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

Όπου η παράμετρος w_i αναπαριστά μία κατανομή πιθανότητας.

β) Για κάθε χαρακτηριστικό j , γίνεται εκπαίδευση ενός ταξινομητή h_j ο οποίος είναι περιορισμένος να χρησιμοποιεί μόνο ένα χαρακτηριστικό. Το ποσοστό σφάλματος του ταξινομητή (ε_j) εκτιμάται μέσω της σχέσης:

$$\varepsilon_j = \sum_i h_j(x_i) - y_i$$

γ) Επιλογή του ταξινομητή, με το μικρότερο ποσοστό σφάλματος ε_t

δ) Ενημέρωση των βαρών

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_i}$$

Όπου $\varepsilon_i=0$ αν το δείγμα x_i είναι ορθώς ταξινομημένο διαφορετικά $\varepsilon_i=1$

$$\text{και } \beta_t = \frac{e^{-\varepsilon_t}}{1-e^{-\varepsilon_t}}$$

4. Ο τελικός ισχυρός ταξινομητής είναι:

$$h(x) = 1, \text{ αν } \sum_{t=1}^T a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T a_t$$

$$h(x) = 0, \text{ σε άλλη περίπτωση}$$

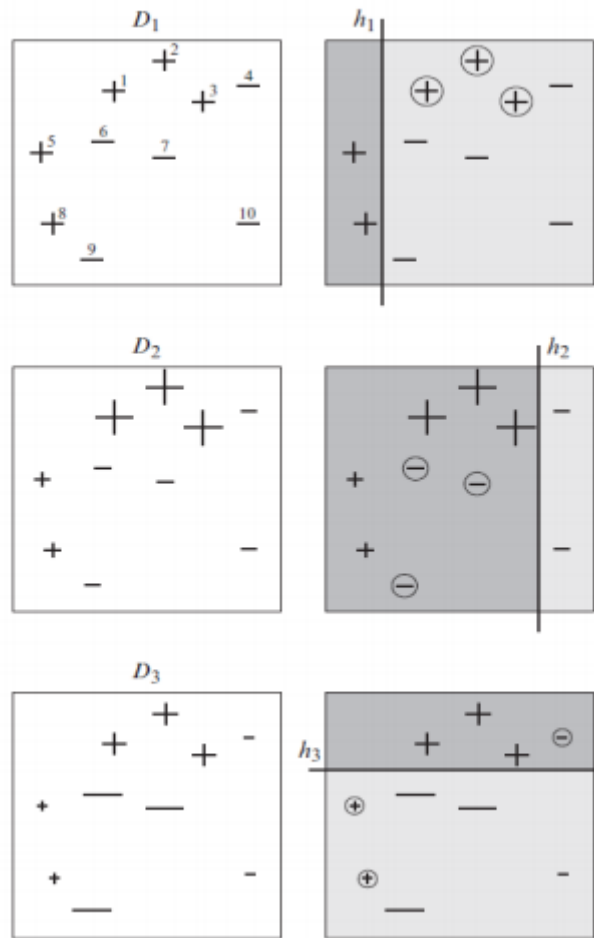
$$\text{όπου } \alpha_t = \log\left(\frac{1}{\beta_t}\right)$$

Ας δούμε ένα απλό παράδειγμα (Κοτσιάς, 2017) εκτέλεσης του παραπάνω αλγορίθμου. Αρχικά, όλα τα δείγματα (θετικά και αρνητικά) παίρνουν μία τιμή βάρους, η οποία είναι ίδια για όλα. Στην πρώτη επανάληψη μάθησης ο αλγόριθμος ταξινομεί όλα τα δείγματα με κάθε διαθέσιμη συνάρτηση ταξινόμησης. Όπως φαίνεται στην εικόνα 17, ο αλγόριθμος ταξινομεί τα σημεία σαν θετικά αν βρίσκονται αριστερά της γραμμής. Έτσι έχουμε αποτυχία για 3 δείγματα τα οποία είναι θετικά, αλλά ταξινομήθηκαν ως αρνητικά. Συνεπώς υπάρχει σφάλμα στην πρώτη επανάληψη $\epsilon_1=0.3$. Τα βάρη των λάθος ταξινομημένων δειγμάτων αυξάνονται, ενώ τα βάρη των σωστά ταξινομημένων μειώνεται, με στόχο την δημιουργία της νέας συνάρτησης ταξινόμησης. Έτσι, στον δεύτερο κύκλο εκμάθησης ο αλγόριθμος ταξινόμησης θα θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον προηγούμενο ταξινομητή. Στην δεύτερη επανάληψη του παραδείγματος μας, δημιουργείται μια άλλη γραμμή βάση της οποίας θα γίνει η ταξινόμηση των θετικών και των αρνητικών σημείων. Όπως φαίνεται και στην εικόνα 17, η γραμμή έχει μετακινηθεί προς τα αριστερά ώστε να ταξινομηθούν σωστά τα 3 θετικά σημεία που είχαν ταξινομηθεί λάθος από την πρώτη επανάληψη. Πλέον έχουμε λάθος ταξινομήσει τρία αρνητικά σημεία. Το σφάλμα ταξινόμησης, όμως είναι μικρότερο από πριν γιατί τα σημεία αυτά έχουν λάβει μικρότερο βάρος από την πρώτη επανάληψη. Πριν ολοκληρωθεί η επανάληψη έχουμε την αύξηση των βαρών των σημείων που ταξινομήθηκαν λάθος και παράλληλα την μείωση των βαρών των σωστά ταξινομημένων σημείων.

Στην τρίτη επανάληψη επιλέγεται πάλι μια νέα γραμμή ταξινόμησης με βάση τα νέα βάρη. Σε αυτή την επανάληψη δεν έχουν ταξινομηθεί σωστά δυο θετικά και ένα

αρνητικό σημείο. Αξίζει να σημειωθεί ότι τα σημεία αυτά στις προηγούμενες επαναλήψεις είχαν ταξινομηθεί σωστά, οπότε έχουν την χαμηλότερη τιμή βάρους από τα υπόλοιπα σημεία.

Τα βήματα του αλγορίθμου επαναλαμβάνονται διαδοχικά, μέχρι να φτάσουμε στο επίπεδο του συνολικού λόγου λανθασμένης ταξινόμησης που επιθυμούμε. Τελικά, ο ισχυρός ταξινομητής προκύπτει από τον συνδυασμό των ασθενών ταξινομητών που επιλέχθηκαν και ένα κατώφλι. Κατά την διαδικασία της ταξινόμησης ενός τμήματος μιας εικόνας από τον ισχυρό ταξινομητή, εφαρμόζονται στο τμήμα όλοι οι ασθενείς ταξινομητές. Τα αποτελέσματα των ασθενών ταξινομητών αθροίζονται, και αν το άθροισμα ξεπερνά το κατώφλι του ταξινομητή, το υπό εξέταση αντικείμενο ταξινομείται ως θετικό, αλλιώς ως αρνητικό. Παρόλο που κάθε ταξινομητής ξεχωριστά δεν καταφέρνει να ταξινομήσει σωστά όλα τα δείγματα του, ο τελικός ισχυρός ταξινομητής καταφέρνει και ταξινομεί σωστά όλα τα δείγματα.



Εικόνα 17. Παράδειγμα εκτέλεσης Adaboost

Βιβλιογραφία

Bow, & Tze, S. (1992). *Pattern Recognition and image pre- processing*.

Simon, P. (2013). *Too Big to Ignore: The Business Case for Big Data*.

Wikipedia. (2013). Ανάκτηση από

https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CF%8C%CF%81%CE%B1%CF%83%CE%B7.

Yam, K. L., & Papadakis, S. E. (2003). *A simple digital imaging method for measuring and analyzing*.

Κοτσιάς, Δ. (2017). *Αλγόριθμοι Ανίχνευσης Προσώπων*. Πάτρα.

Κωτσιόπουλος, Χ. (2014). *Αναγνώριση Προτύπων από Εικόνες*. Πάτρα.

Πρωτονοτάριος, Ι. (2011). *Ανάπτυξη Συστήματος Ενσωματωμένων Αισθητήρων*. Πάτρα.

http://www.cse.iitm.ac.in/~vplab/courses/CV_DIP/PDF/NEIGH_CONN.pdf

Β' ΜΕΡΟΣ

Υλοποίηση

Βιβλιοθήκη OpenCV

Η OpenCV αποτελεί μια βιβλιοθήκη ελευθέρου λογισμικού η οποία αναπτύχθηκε από την εταιρία Intel και αφορά την επεξεργασία εικόνας. Υποστηρίζεται τόσο σε περιβάλλον Windows όσο και σε Unix, MacOS και εκτός των C/C++ υποστηρίζεται και σε άλλες γλώσσες προγραμματισμού όπως είναι η Python και η Java. Πρόκειται για μια διεπαφή που προσφέρει αίσθηση υψηλού επιπέδου γλώσσας προγραμματισμού και εκμεταλλεύεται σε σημαντικό βαθμό το σύστημα εξοπλισμού του υπολογιστή (hardware), προσφέροντας υψηλές ταχύτητες υλοποίησης σύνθετων προγραμμάτων. Επιπλέον, η βιβλιοθήκη μπορεί να χρησιμοποιηθεί σαν πρόσθετο και σε άλλες γλώσσες προγραμματισμού, όπως για παράδειγμα στο Matlab, octave, C# κ.α. Η ανάπτυξη επικοινωνίας ανθρώπου με υπολογιστή, η ανίχνευση, απομόνωση και αναγνώριση αντικειμένων, η ανίχνευση και αναγνώριση προσώπων, η κατανόηση και παρακολούθηση κίνησης είναι μερικά από τα πεδία μηχανικής όρασης που καλύπτει. Οι αλγόριθμοι της OpenCV είναι βελτιστοποιημένοι για επεξεργαστές αρχιτεκτονικής IntelPentium (MMX, Pro, 3, 4). Τέλος, η δημιουργία της OpenCV αποσκοπεί και στην δημιουργία μιας κοινότητας ανοιχτού λογισμικού, σχετική με την μηχανική όραση, η οποία θα αναπτύσσει σύγχρονες μεθόδους επεξεργασίας Η OpenCV σε γενικές γραμμές διαρθρώνεται σε πέντε κύρια μέρη, τέσσερα από τα οποία παρουσιάζονται στο σχήμα. Η CV περιέχει την βασική επεξεργασία εικόνας και τους αλγόριθμους τεχνητής όρασης υψηλότερου επιπέδου. Η ML είναι η βιβλιοθήκη μάθησης μηχανής, που περιλαμβάνει πολλούς στατιστικούς ταξινομητές και εργαλεία ομαδοποίησης(clustering). Η HighGUI περιέχει ρουτίνες

εισόδου/εξόδου και τις συναρτήσεις για την αποθήκευση και φόρτωση βίντεο και εικόνων και η CXCore περιέχει τις βασικές δομές δεδομένων και το περιεχόμενο. Επίσης υπάρχει η CvAux, η οποία περιέχει περιοχές που δεν χρησιμοποιούνται (ενσωματωμένο HMM αναγνώρισης προσώπου) αλλά και πειραματικούς αλγόριθμους (κατάτμηση φόντου, πρόσοψης).

OpenCV και ανίχνευση αντικειμένων

Η ανίχνευση αντικειμένων στην OpenCV στηρίζεται σε ένα σύστημα μηχανικής μάθησης το οποίο είναι ικανό να επεξεργάζεται πολύ γρήγορα μια εικόνα και να πετυχαίνει πολύ υψηλά ποσοστά επιτυχούς ανίχνευσης. Η διαδικασία της ανίχνευσης είναι πολύ πιο αποδοτική όταν βασίζεται στην ανίχνευση χαρακτηριστικών (features), τα οποία κωδικοποιούν πληροφορία σχετικά με το αντικείμενο που πρόκειται να ανιχνευτεί. Τα χαρακτηριστικά αυτά χρησιμοποιούνται σε σύνολα και κωδικοποιούν πληροφορία που αφορά την αντίθεση φωτεινότητας ανάμεσα σε περιοχές της εικόνας, τον προσανατολισμό της αντίθεσης και την χωρική σχέση αυτών των περιοχών. Στην διαδικασία της ανίχνευσης υπάρχουν τρία είδη χαρακτηριστικών, τα διπολυγωνικά, τα τρι-πολυγωνικά και τα τετρα-πολυγωνικά χαρακτηριστικά. Η τιμή των δι-πολυγωνικών χαρακτηριστικών είναι η διαφορά ανάμεσα στο σύνολο των pixels που βρίσκονται μέσα στις δύο πολυγωνικές περιοχές. Η τιμή ενός τριπολυγωνικού χαρακτηριστικού υπολογίζεται αφαιρώντας από το άθροισμα των pixels των εξωτερικών πολυγώνων, το άθροισμα των pixels του εσωτερικού πολυγώνου. Τέλος η τιμή των τετρα-πολυγωνικών χαρακτηριστικών υπολογίζεται από την διαφορά των pixels ανάμεσα στα διαγώνια μέρη του πολυγώνου. Στο Σχήμα X.1 φαίνεται ένα παράδειγμα τέτοιων χαρακτηριστικών. Για τον γρήγορο υπολογισμό των χαρακτηριστικών, σε οποιαδήποτε κλίμακα ή περιοχή, το σύστημα δεν χρησιμοποιεί κατευθείαν τις τιμές φωτεινότητας της εικόνας, αλλά

μια ενδιάμεση αναπαράστασή της, που ονομάζεται πλήρης εικόνα. Η πλήρης εικόνα μπορεί να υπολογιστεί κατευθείαν από την αρχική εικόνα με χρήση απλών πράξεων στις τιμές των pixels της. Συγκεκριμένα, ο τρόπος υπολογισμού της πλήρους εικόνας γίνεται με την χρήση της σχέσης :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

όπου $ii(x, y)$ η τιμή της πλήρους εικόνας στο σημείο (x, y) και $i(x', y')$ η τιμή της πραγματικής εικόνας στο ίδιο σημείο. Η πλήρης εικόνα μπορεί να υπολογιστεί με ένα μόνο πέρασμα πάνω από την πραγματική εικόνα.

OpenCV και διαδικασία μάθησης

Η διαδικασία ανίχνευσης αντικειμένων που χρησιμοποιεί η OpenCV αρχικά προτάθηκε από τον Paul Viola και υλοποιήθηκε από τον Rainer Lienhart. Αρχικά ένας ταξινομητής εκπαιδεύεται με ένα μεγάλο αριθμό θετικών δειγμάτων (εικόνες που περιέχουν το αντικείμενο προς ανίχνευση και έχουν όλες το ίδιο μέγεθος) και με αρνητικά δείγματα (αυθαίρετες εικόνες που δεν περιέχουν το αντικείμενο και είναι ίδιου μεγέθους με τα θετικά δείγματα). Από κάθε θετικό δείγμα εξάγεται ένας πολύ μεγάλος αριθμός χαρακτηριστικών. Πειραματικά έχει αποδειχθεί ότι ένα πολύ μικρό σύνολο αυτών των χαρακτηριστικών είναι αρκετό για την δημιουργία ενός πολύ αποδοτικού ταξινομητή. Για την εξαγωγή αυτών των χαρακτηριστικών χρησιμοποιείται ένας αλγόριθμος μάθησης ο οποίος έχει σχεδιαστεί να επιλέγει μόνο τα πολυγωνικά χαρακτηριστικά τα οποία ανταποκρίνονται καλύτερα στα θετικά και στα αρνητικά δείγματα. Ο αλγόριθμος μάθησης που χρησιμοποιήθηκε είναι ο AdaBoost σε συνδυασμό με naive Bayes. Ο AdaBoost είναι ένας προσαρμοστικός αλγόριθμος με την έννοια ότι οι μεταγενέστεροι ταξινομητές που δημιουργεί

ανταποκρίνονται καλύτερα στα θετικά και αρνητικά δείγματα σε σχέση με τους προγενέστερους. Ο αλγόριθμος καλείται επαναληπτικά για μια σειρά T κύκλων. Σε κάθε κύκλο η κατανομή βαρών των χαρακτηριστικών ανανεώνεται έτσι ώστε να ξεχωρίζουν τα χαρακτηριστικά με βάση την σημαντικότητα τους στην διαδικασία της ταξινόμησης.

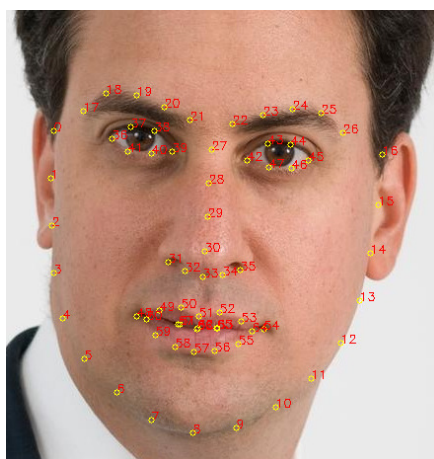
Τεχνικές ανίχνευσης προσώπου και χαρακτηριστικών

Η σημερινή εξέλιξη της τεχνολογίας των υπολογιστών έχει προβλέψει έναν κόσμο με προηγμένα μηχανήματα, όπου η ανθρώπινη ζωή ενισχύεται από την τεχνητή νοημοσύνη. Η τεχνητή όραση για παράδειγμα, έχει ως στόχο να αναπαράγει την ανθρώπινη όραση. Παραδοσιακά, τα συστήματα υπολογιστικής όρασης έχουν χρησιμοποιηθεί σε συγκεκριμένες διεργασίες όπως η εκτέλεση κουραστικών και επαναλαμβανόμενων οπτικών διεργασιών. Οι τρέχουσες εξελίξεις σ' αυτό τον τομέα κινούνται προς ένα γενικευμένο όραμα εφαρμογών όπως είναι οι τεχνικές αναγνώρισης προσώπου και κωδικοποίησης βίντεο. Πολλές από τις τρέχουσες τεχνικές αναγνώρισης προσώπου αναλαμβάνουν τη διαθεσιμότητα μετωπικών προσώπων ίδιου μεγέθους. Στην πραγματικότητα, αυτή η υπόθεση δεν μπορεί να ισχύει λόγω της ποικιλίας της εμφάνισης του προσώπου και των περιβαλλοντικών συνθηκών (Εικόνα 2-1). Αυτές οι εικόνες είναι τυπικές δοκιμαστικές εικόνες που χρησιμοποιούνται στην έρευνα ταξινόμησης προσώπου. Ο αποκλεισμός του φόντου είναι αναγκαίος σ' αυτές τις εικόνες για αξιόπιστες τεχνικές ταξινόμησης προσώπου. Ωστόσο, σε ρεαλιστικά σενάρια εφαρμογής όπως στην Εικόνα 2-2, ένα πρόσωπο θα μπορούσε να υπάρξει σε ένα πιο πολύπλοκο φόντο και σε πολλές πόζες. Τα συστήματα αναγνώρισης που βασίζονται σε τυποποιημένες εικόνες είναι πιθανό να μπερδεύουν κάποιες περιοχές του φόντου ως πρόσωπο. Προκειμένου να διορθωθεί

το πρόβλημα, χρειάζεται ένας οπτικός επεξεργαστής για τον εντοπισμό και την εξαγωγή της περιοχής του προσώπου από το φόντο.

Μοντέλα Κατανεμημένων σημείων (Point distributed Models)

Αυτή η μέθοδος παίρνει τις στατιστικές πληροφορίες του σχήματος σε μια εικόνα και τις συγκρίνει με ένα προκαθορισμένο σύνολο εκπαίδευσης ώστε να καθοριστεί αν το σχήμα είναι ένα κεφάλι. Το μοντέλο κατανεμημένων σημείων δημιουργείται από λογισμικό και εισάγεται σε ένα σύνολο σημείων τα οποία φέρουν ετικέτες. Οι παραλλαγές αυτών των σημείων πρώτα καθορίζονται από τη χρήση του συνόλου εκπαίδευσης που περιλαμβάνει αντικείμενα διαφόρων μεγεθών και στάσεων. Χρησιμοποιώντας την ανάλυση κύριων συνιστωσών, οι παραλλαγές των χαρακτηριστικών σε ένα σύνολο εκπαίδευσης κατασκευάζονται ως ένα εύκαμπτο γραμμικό μοντέλο.



Προσέγγιση βασισμένη στην εικόνα

Η προσέγγιση με βάση τα χαρακτηριστικά συνήθως περιορίζεται στην ανίχνευση ενός προσώπου σε ένα μη πολύπλοκο φόντο με ιδανικές συνθήκες. Εκεί προκύπτει η ανάγκη για τεχνικές που να μπορούν να ανιχνεύσουν πολλά πρόσωπα σε πολύπλοκα φόντα. Γι' αυτό το λόγο αναπτύχθηκε η αναγνώριση προτύπων. Αυτή η τεχνική

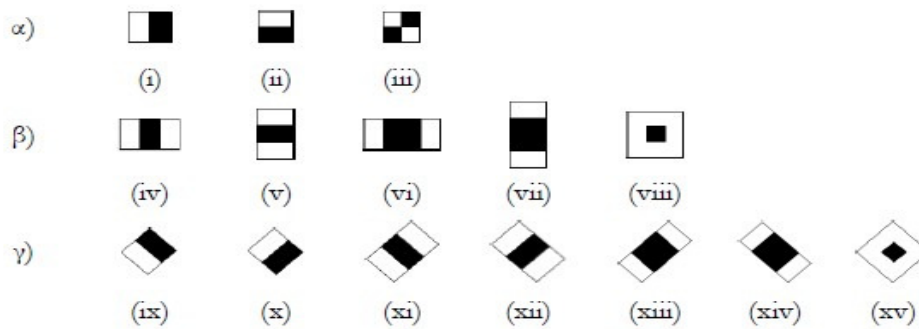
λειτουργεί με την ιδέα ότι το πρόσωπο αναγνωρίζεται με τη σύγκριση μιας εικόνας με τα παραδείγματα των πρότυπων προσώπων. Οι περισσότερες προσεγγίσεις με βάση την εικόνα εφαρμόζουν μια τεχνική σάρωσης παραθύρου για την ανίχνευση προσώπου. Ο αλγόριθμος του παραθύρου σάρωσης στην ουσία είναι μια εξαντλητική αναζήτηση στην εικόνα εισόδου για πιθανές θέσεις του προσώπου σε όλες τις κλίμακες. Συνήθως το μέγεθος του παραθύρου σάρωσης, ο ρυθμός υποδειγματοληψίας και ο αριθμός των επαναλήψεων ποικίλει ανάλογα με μέθοδο. Η προσέγγιση εδώ είναι η ταξινόμηση μιας περιοχής σαν πρόσωπο ή μη πρόσωπο, έτσι ώστε το σύνολο των πρωτότυπων του προσώπου και μη προσώπου πρέπει να εκπαιδευτεί για να ταιριάζει σ' αυτά τα πρότυπα.

Ανίχνευση προσώπου με Haar χαρακτηριστικά

Η ανίχνευση αντικειμένου με τη χρήση ταξινομητών καταρράκτη με βάση Haar χαρακτηριστικά είναι μια αποτελεσματική μέθοδος για την ανίχνευση των αντικειμένων που προτάθηκε από τους Paul Viola και Michael Jones και αναπτύχθηκε από τον Rainer Lienhart. Πρόκειται για μια προσέγγιση βασισμένη στη μηχανική μάθηση όπου η συνάρτηση καταρράκτη έχει εκπαιδευτεί από πολλές θετικές και αρνητικές εικόνες. Στη συνέχεια χρησιμοποιείται για την ανίχνευση αντικειμένων σε άλλες εικόνες. Υπάρχουν 3 συμβολές στο πλαίσιο ανίχνευσης αντικειμένου: Η πρώτη συμβολή είναι μια νέα αναπαράσταση εικόνας που ονομάζεται ακέραια εικόνα και επιτρέπει την πολύ γρήγορη εκτίμηση χαρακτηριστικών. Χρησιμοποιείται ένα σύνολο χαρακτηριστικών το οποίο χρησιμοποιεί Haar-like συναρτήσεις. Για να υπολογιστούν αυτά τα χαρακτηριστικά πολύ γρήγορα σε πολλές κλίμακες έχει εισαχθεί η αναπαράσταση ακέραιας εικόνας (integral image) για εικόνες. Η ακέραια εικόνα μπορεί να υπολογιστεί από μια εικόνα με τη χρήση μερικών εργασιών ανά pixel. Αν υπολογιστεί μια φορά, κάθε ένα από τα

Haar-like χαρακτηριστικά μπορούν να υπολογιστούν σε οποιαδήποτε κλίμακα ή τοποθεσία σε σταθερό χρόνο. Η δεύτερη συμβολή είναι η μέθοδος για την κατασκευή ενός ταξινομητή με την επιλογή ενός μικρού αριθμού σημαντικών χαρακτηριστικών χρησιμοποιώντας AdaBoost. Μέσα σε οποιοδήποτε υπό-παράθυρο της εικόνας ο συνολικός αριθμός των haar-like χαρακτηριστικών είναι πολύ μεγάλος, μεγαλύτερος από τον αριθμό των εικονοστοιχείων. Για να εξασφαλιστεί η γρήγορη ταξινόμηση, η διαδικασία εκμάθησης πρέπει να αποκλείει μια μεγάλη πλειοψηφία των διαθέσιμων χαρακτηριστικών και να επικεντρωθεί σε ένα μικρό σύνολο κρίσιμων χαρακτηριστικών. Η επιλογή των χαρακτηριστικών επιτυγχάνεται με μια απλή τροποποίηση της διαδικασίας AdaBoost: ο αδύναμος μαθητής περιορίζεται έτσι ώστε κάθε ασθενής ταξινομητής να αλλάξει στο να εξαρτάται από μόνο ένα χαρακτηριστικό. Ως αποτέλεσμα αυτού του σταδίου της διαδικασίας ενίσχυσης, το οποίο διαλέγει έναν νέο αδύναμο ταξινομητή, μπορεί να θεωρηθεί ως μια διαδικασία επιλογής χαρακτηριστικών. Το AdaBoost παρέχει έναν αποτελεσματικό αλγόριθμο μάθησης και ισχυρά όρια στην απόδοση γενίκευσης. Η τρίτη σημαντική συμβολή είναι μια μέθοδος για τον επιτυχή συνδυασμό πιο πολύπλοκων ταξινομητών σε μια δομή καταρράκτη η οποία αυξάνει δραματικά την ταχύτητα του ανιχνευτή με την εστίαση της προσοχής σε υποσχόμενες περιοχές της εικόνας. Η έννοια πίσω από την εστίαση της προσοχής είναι ότι είναι συχνά πιθανό να προσδιοριστεί γρήγορα πού μπορεί να προκύψει ένα αντικείμενο σε μια εικόνα. Πιο πολύπλοκη διαδικασία προορίζεται μόνο για αυτές τις υποσχόμενες περιοχές. Το βασικό μέτρο αυτής της προσέγγισης είναι «ψευδώς αρνητικό» ποσοστό της προσοχής της διαδικασίας. Πρέπει να είναι η περίπτωση όπου όλα, ή σχεδόν όλα, τα στιγμιότυπα των αντικειμένων να επιλέγονται από τα προσεκτικά φίλτρα. Η χρησιμοποίηση των συναρτήσεων Haar στο μετασχηματισμό κυματιδίων ξεκινά από την παρατήρηση ότι

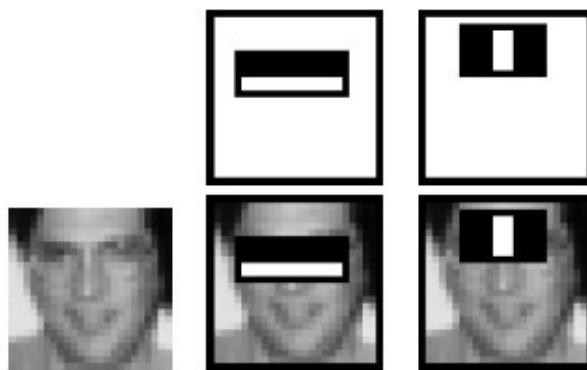
η τιμή της φωτεινότητας κάθε εικονοστοιχείου επηρεάζεται έντονα από τις αλλαγές στο φωτισμό της σκηνής. Αυτή η αλλαγή όμως, επηρεάζει αρκετά ομοιόμορφα όλα τα εικονοστοιχεία της εικόνας. Έτσι, η τιμή μιας συνάρτησης που εξετάζει τη μέση διαφορά ανάμεσα σε δύο ή τρεις περιοχές της ίδιας εικόνας, θα παραμένει σε μεγάλο βαθμό ανεπηρέαστη. Χρησιμοποιώντας, λοιπόν, τις συναρτήσεις Haar, η διαδικασία της ανίχνευσης αντικειμένων δε θα επηρεάζεται από τις διαφορές στη φωτεινότητα από εικόνα σε εικόνα. Οι συναρτήσεις Haar υπολογίζουν τη διαφορά ανάμεσα στους μέσους όρους των τιμών των εικονοστοιχείων δύο (ή τριών) περιοχών. Ας θεωρήσουμε τη συνάρτηση Haar που παριστάνεται με το ορθογώνιο i από την Εικόνα 5-1. Υπολογίζεται ο μέσος όρος των εικονοστοιχείων που βρίσκονται μέσα στο άσπρο ορθογώνιο, καθώς και αυτών που βρίσκονται μέσα στο μαύρο ορθογώνιο. Έπειτα, ο μέσος όρος του μαύρου ορθογωνίου αφαιρείται από τον μέσο όρο του άσπρου. Η τιμή που προκύπτει αποτελεί την τιμή του Haar χαρακτηριστικού. Τα κλασσικά Haar χαρακτηριστικά φαίνονται στην Εικόνα 5-1α. Είναι σχετικά απλά και μπορούν να εντοπίσουν ακμές οριζόντια και κατακόρυφα καθώς και διαγώνιες γραμμές. Για να μπορέσουμε να αναπαραστήσουμε γραμμές, ράβδους και τετράγωνα καλύτερα, προσθέτουμε τα χαρακτηριστικά που φαίνονται στην Εικόνα 5-1β, τα οποία υπολογίζονται χωρίς να αυξάνεται ιδιαίτερα η πολυπλοκότητα. Μια μεγάλη προσθήκη είναι τα χαρακτηριστικά που είναι περιστραμμένα κατά 45° και φαίνονται στην Εικόνα 5-1γ. Με τη χρήση αυτών βελτιώνεται σημαντικά η αναπαράσταση των διαγώνιων σχημάτων. Με την προσθήκη όλων αυτών των χαρακτηριστικών, το σύνολο γίνεται υπερπλήρες και αναπαριστά πολύ καλύτερα την πληροφορία που περιέχεται σε μία εικόνα.



Στην μέθοδο ανίχνευσης που εξετάζουμε, χρησιμοποιήθηκε ο αλγόριθμος AdaBoost τόσο για την επιλογή των χαρακτηριστικών που θα χρησιμοποιηθούν, όσο και για την εκπαίδευση του ταξινομητή. Ο αλγόριθμος εκμάθησης AdaBoost ανήκει στην κατηγορία των αλγορίθμων ενδυνάμωσης (boosting) και χρησιμοποιείται για να αυξήσει την απόδοση ενός οποιουδήποτε απλού αλγορίθμου ταξινόμησης. Ο απλός αλγόριθμος ταξινόμησης λέγεται και ασθενής αλγόριθμος ταξινόμησης, καθώς ακόμα και η καλύτερη συνάρτηση ταξινόμησης που μπορεί να προκύψει από αυτόν, δεν αναμένεται να ταξινομεί καλά τα δεδομένα. Συγκεκριμένα, αρκεί η συνάρτηση ταξινόμησης να έχει απόδοση ελαφρά καλύτερη από την τυχαία ταξινόμηση (50%). Για να αυξήσει, λοιπόν, την απόδοση ενός ασθενούς αλγορίθμου ταξινόμησης, ο AdaBoost συνδυάζει μια συλλογή ασθενών συναρτήσεων ταξινόμησης χρησιμοποιώντας άπληστο αλγόριθμο, ώστε να σχηματίσει από αυτούς έναν ισχυρότερο ταξινομητή. Η βελτίωση του ασθενούς αλγορίθμου ταξινόμησης πραγματοποιείται, καλώντας τον αλγόριθμο να επιλύσει μια αλληλουχία προβλημάτων ταξινόμησης. Αρχικά, όλα τα παραδείγματα (θετικά και αρνητικά) παίρνουν μια τιμή βάρους, η οποία είναι ίδια για όλα. Δίνονται στον αλγόριθμο τα παραδείγματα και πραγματοποιείται ο πρώτος κύκλος εκμάθησης, όπου ο αλγόριθμος ταξινομεί όλα τα παραδείγματα με κάθε διαθέσιμη συνάρτηση ταξινόμησης. Έπειτα, οι συναρτήσεις ταξινόμησης διατάσσονται σύμφωνα με τα αποτελέσματά τους, λαμβάνοντας υπόψη το βάρος κάθε παραδείγματος. Επιλέγεται

έναν μικρό αριθμό συναρτήσεων ταξινόμησης, από αυτές με τα καλύτερα αποτελέσματα, που αποτελούν τον πρώτο ασθενή ταξινομητή. Ο πρώτος κύκλος εκμάθησης ολοκληρώνεται και τα βάρη των παραδειγμάτων ισοσταθμίζονται, δίνοντας μεγαλύτερο βάρος στα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον πρώτο ασθενή ταξινομητή. Έτσι, στον δεύτερο κύκλο εκμάθησης ο αλγόριθμος ταξινόμησης θα θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον προηγούμενο ταξινομητή. Τα βήματα επαναλαμβάνονται διαδοχικά, μέχρι να φτάσουμε στο επίπεδο του συνολικού λόγου λανθασμένης ταξινόμησης που επιθυμούμε. Τελικά, ο ισχυρός ταξινομητής προκύπτει από τον συνδυασμό των ασθενών ταξινομητών που επιλέχθηκαν και ένα κατώφλι. Κατά την διαδικασία της ταξινόμησης ενός υποπαραθύρου εικόνας από τον ισχυρό ταξινομητή, εφαρμόζονται στο υποπάρθυρο όλοι οι ασθενείς ταξινομητές. Τα αποτελέσματα των ασθενών ταξινομητών αθροίζονται, και αν το άθροισμα ξεπερνά το κατώφλι του ταξινομητή, το υπό εξέταση αντικείμενο ταξινομείται ως θετικό, αλλιώς ως αρνητικό. Στη μέθοδο ανίχνευσης αντικειμένων που χρησιμοποιούμε, κάθε ασθενής αλγόριθμος εκμάθησης περιορίζεται στο σύνολο των συναρτήσεων ταξινόμησης που αποτελούνται από ένα μόνο χαρακτηριστικό τύπου Haar. Προφανώς, από ένα μόνο χαρακτηριστικό δε μπορούμε να περιμένουμε ιδιαίτερα χαμηλό λόγο σφάλματος. Σε κάθε στάδιο του αλγορίθμου AdaBoost επιλέγεται το χαρακτηριστικό που διαχωρίζει καλύτερα τα θετικά από τα αρνητικά δείγματα. Για κάθε χαρακτηριστικό, ο ασθενής αλγόριθμος εκμάθησης προσδιορίζει ένα κατώφλι της τιμής του χαρακτηριστικού, που ελέγχοντάς το περιορίζονται οι λανθασμένες ταξινομήσεις από το συγκεκριμένο χαρακτηριστικό στις ελάχιστες δυνατές. Έπειτα, επιλέγεται ως ασθενής ταξινομητής το χαρακτηριστικό τύπου Haar, που, για το δεδομένο κατώφλι του, κάνει τη συνολικά καλύτερη ταξινόμηση. Ο AdaBoost συνεχίζει εκπαιδεύοντας όλους τους

ασθενείς ταξινομητές, μέχρι το σημείο που ο ισχυρός συνολικός ταξινομητής επιτυγχάνει το επίπεδο ταξινόμησης που ζητάμε. Το πρώτο χαρακτηριστικό που επιλέχτηκε επικεντρώθηκε στην περιοχή των ματιών που είναι συνήθως πιο σκούρα απ' ό,τι η περιοχή της μύτης και τα μάγουλα. Αυτό το χαρακτηριστικό είναι σχετικά μεγάλο σε σχέση με το υπό-παράθυρο ανίχνευσης. Το δεύτερο χαρακτηριστικό στηρίζεται στο ότι τα μάτια είναι σκουρότερα απ' ό,τι η γέφυρα της μύτης (εικόνα 5-4).



Η OpenCV έχει τέσσερις μεθόδους ανίχνευσης προσώπων σε κατά μέτωπο στάση (frontal), με τις οποίες έχει ήδη εκπαιδευτεί ο ταξινομητής και είναι αποθηκευμένες σε αντίστοιχα αρχεία xml, τα οποία μπορούν να φορτωθούν ως string. Η μέθοδος που χρησιμοποιήθηκε στην εφαρμογή μας είναι η `haarcascade_frontalface_alt`. Η συνάρτηση `detectMultiScale` βρίσκει ορθογώνιες περιοχές σε μια εικόνα που πιθανό να περιέχουν αντικείμενα όπου έχει εκπαιδευτεί ο ταξινομητής να διακρίνει και επιστρέφει αυτές τις περιοχές ως ένα διάνυσμα ορθογωνίων. Η συνάρτηση σαρώνει πολλές φορές την εικόνα, σε διάφορες κλίμακες, και για κάθε περιοχή εφαρμόζει τον ταξινομητή για να ελέγξει την ύπαρξη κάποιου ανιχνευμένου αντικειμένου. Υπάρχουν κάποιοι προκαθορισμένοι παράμετροι οι οποίοι είναι ρυθμισμένοι για ανίχνευση μεγαλύτερης ακρίβειας, όμως είναι πιο χρονοβόρα.

Ανάλυση εφαρμογής

Η υλοποίηση της εφαρμογής γίνεται σε ένα απλό τερματικό περιβάλλον. Στην συνέχεια περιγράφεται βηματικά η λειτουργία του προγράμματος.

1. Φόρτωση αρχείου εικόνας με χρήση της συνάρτησης `imread`.
2. Εύρεση προσώπων στην εικόνα και της περιοχής του εκάστοτε προσώπου.
Γίνεται χρήση της συνάρτησης `detectMultiScale`.
3. Για κάθε πρόσωπο γίνεται διαγράμμιση αυτού στην εικόνα εξόδου.
4. Εκτύπωση αποτελέσματος και αποθήκευση της φωτογραφίας εξόδου.

Προαπαιτούμενα

1. VisualStudio – CommunityEdition 2017 (χρησιμοποιήθηκε ως περιβάλλον ανάπτυξης, αλλά επίσης προσφέρει κάποιες απαραίτητες βιβλιοθήκες), διαθέσιμο στην διεύθυνση: <https://visualstudio.microsoft.com/>.
2. OpenCV – Έκδοση 3.4.1 (παρέχει ένα πλαίσιο αναγνώρισης προσώπων), διαθέσιμο στην διεύθυνση <https://opencv.org/releases.html>. Κάνουμε extract τα αρχεία στο φάκελο C:\opencv (καθώς θα πρέπει να μπορεί να βρει το πρόγραμμά μας κάποιες βιβλιοθήκες).

Μετά την εγκατάσταση του OpenCV, για να ορίσουμε το μονοπάτι, εκτελούμε σε ένα commandline με δικαιώματα administrator:

```
setx -m OPENCV_DIR C:\OpenCV\Build\x64\vc15
```

Επεξήγηση Κώδικα

Θα δώσουμε μια περιληπτική περιγραφή της λειτουργίας του κώδικα που αναπτύχθηκε. Αρχικά, το πρόγραμμα ζητάει ως είσοδο από τον χρήστη τον φάκελο όπου βρίσκονται αποθηκευμένες οι εικόνες. Συγκεκριμένα, αυτός ο φάκελος πρέπει να έχει ένα αρχείο “data.csv”, το οποίο περιέχει τα ονόματα των αρχείων. Για διευκόλυνση στην ανάπτυξη και στα πειράματά μας, αν ο χρήστης πατήσει enter χωρίς να δώσει όνομα, χρησιμοποιείται από προεπιλογή ο φάκελος «C:/Image_data/» (τονίζουμε ότι χρειάζεται το “/” στο τέλος της διαδρομής αν ο χρήστης δώσει κάποιον άλλο φάκελο). Στη συνέχεια για κάθε αρχείο εκτελείται η συνάρτηση εντοπισμού προσώπων που παρέχεται από το πλαίσιο OpenCV. Μετά, δημιουργείται ένας φάκελος results, στον οποίο δημιουργείται ένα αρχείο results.csv, όπου θα

αποθηκεύουμε το όνομα του αρχείου εικόνας και τον αριθμό προσώπων που ανιχνεύθηκαν για να διευκολυνθούμε μετέπειτα στην αξιολόγηση της απόδοσης. Ύστερα, για κάθε εικόνα, σχεδιάζεται ένα τετράγωνο γύρω από κάθε πρόσωπο που ανιχνεύθηκε και η τελική εικόνα αποθηκεύεται επίσης στο φάκελο results.

Πείραμα 1ο

Κατεβάσαμε από το διαδίκτυο 50 εικόνες, οι οποίες απεικονίζουν τουλάχιστον 4 πρόσωπα. Οι εικόνες κατηγοριοποιήθηκαν στις κατηγορίες Ποδοσφαιρική Ομάδα, Ηθοποιοί, Μουσικό Συγκρότημα, Πολιτικοί και Κολλάζ προσώπων. Ύστερα τρέξαμε το πρόγραμμά μας και πήραμε τα αποτελέσματα που παρουσιάζονται αναλυτικά στον Πίνακα 1, όπου παρουσιάζουμε τα αποτελέσματα καθώς και τις μετρικές TruePositive (TP) (πρόσωπα που ανιχνεύθηκαν), FalseNegative (FN) (πρόσωπα που δεν ανιχνεύθηκαν) και FalsePositive (FP) (μη-πρόσωπα που ανιχνεύθηκαν).

Πίνακας 1. Λίστα εικόνων και αποτελέσματα εντοπισμού προσώπων.

Εικόνα	Περιγραφή	Faces	Detected	TP	FN	FP
1	Κολλάζ προσώπων	15	13	13	2	0
2	Κολλάζ προσώπων	4	4	4	0	0
3	Κολλάζ προσώπων	16	14	14	2	0
4	Ηθοποιοί	7	8	7	0	1
5	Ηθοποιοί	9	8	8	1	0
6	Ηθοποιοί	7	6	6	1	0
7	Ηθοποιοί	7	3	3	4	0
8	Μουσικό Συγκρότημα	4	4	4	0	0
9	Μουσικό Συγκρότημα	4	2	2	2	0
10	Ποδοσφαιρική	11	11	10	1	1

	Ομάδα					
11	Ποδοσφαιρική Ομάδα	11	11	11	0	0
12	Ηθοποιοί	4	3	3	1	0
13	Ηθοποιοί	7	7	7	0	0
14	Ηθοποιοί	8	7	7	1	0
15	Κολλάζ προσώπων	24	21	21	3	0
16	Ηθοποιοί	6	6	6	0	0
17	Πολιτικοί	7	7	7	0	0
18	Ηθοποιοί	6	6	6	0	0
19	Πολιτικοί	36	33	33	3	0
20	Ποδοσφαιρική Ομάδα	16	11	11	5	0
21	Ποδοσφαιρική Ομάδα	11	11	11	0	0
22	Ηθοποιοί	4	4	4	0	0
23	Ηθοποιοί	7	7	7	0	0
24	Ηθοποιοί	12	11	11	1	0
25	Ηθοποιοί	5	5	5	0	0
26	Ηθοποιοί	5	4	4	1	0
27	Ηθοποιοί	5	3	3	2	0
28	Κολλάζ προσώπων	21	21	21	0	0
29	Ηθοποιοί	10	5	5	5	0
30	Κολλάζ προσώπων	12	6	6	6	0

31	Ηθοποιοί	8	8	8	0	0
32	Μουσικό Συγκρότημα	4	4	4	0	0
33	Μουσικό Συγκρότημα	4	4	4	0	0
34	Ηθοποιοί	7	2	2	5	0
35	Ποδοσφαιρική Ομάδα	11	11	11	0	0
36	Ηθοποιοί	8	8	8	0	0
37	Ηθοποιοί	10	9	9	1	0
38	Ηθοποιοί	6	6	6	0	0
39	Ποδοσφαιρική Ομάδα	11	11	11	0	0
40	Ποδοσφαιρική Ομάδα	11	10	10	1	0
41	Ποδοσφαιρική Ομάδα	11	11	11	0	0
42	Ηθοποιοί	7	6	6	1	0
43	Ηθοποιοί	5	5	5	0	0
44	Ποδοσφαιρική Ομάδα	11	11	11	0	0
45	Ηθοποιοί	5	3	3	2	0
46	Ηθοποιοί	4	4	4	0	0
47	Μουσικό	4	2	2	2	0

	Συγκρότημα					
48	Ηθοποιοί	5	5	4	1	1
49	Πολιτικοί	7	5	5	2	0
50	Κολλάζ προσώπων	29	27	27	2	0
	Σύνολο	469	416	411	58	3

Επομένως, η ακρίβεια εντοπισμού προσώπων (δηλαδή το ποσοστό υπαρκτών προσώπων που εντοπίστηκαν) είναι:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{Total} = (411 + 0) / 469 = 87.6\%$$

Το TruePositiveRate ή ευαισθησία (sensitivity), δηλαδή το ποσοστό προσώπων από τα αντικείμενα που εντοπίστηκαν είναι:

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FP}) = 411/416 = 99,3\%$$

Παρατηρούμε πάρα πολύ υψηλή ευαισθησία, δηλαδή πολύ σπάνια ο αλγόριθμος προσδιορίζει ως πρόσωπα αντικείμενα που δεν είναι πρόσωπα. Επίσης, η ακρίβεια είναι αρκετά υψηλή, δεδομένου ότι σε κάποιες από τις εικόνες, όπως αναφέρουμε παρακάτω είναι όντως δύσκολο να εντοπιστούν κάποια πρόσωπα, για παράδειγμα γιατί είναι στραμμένα στο πλάι. Στη συνέχεια στον Πίνακα 2 δίνουμε κάποιους λόγους που πιθανόν ο αλγόριθμος έχασε κάποια πρόσωπα, καθώς μερικές φωτογραφίες ήταν επίτηδες «δύσκολες», για να διαπιστώσουμε τα όρια του αλγορίθμου.

Πίνακας 2. Λίστα εικόνων με λανθασμένο εντοπισμό προσώπων.

No	FN	FP	Σχόλιο για πρόσωπα που δεν ανιχνεύθηκαν
1	2	0	Στη γωνία της εικόνας, ίσως ελαφρώς κομμένα

3	2	0	Στραμμένα πλάγια
4	0	1	(πουκάμισο με γραβάτα)
5	1	0	
6	1	0	Πρόσωπο με μεγάλη κλίση
7	4	0	Πιθανότατα λόγω μεγάλης ψηφιακής επεξεργασίας
9	2	0	Ασπρόμαυρη
10	1	1	Λείπει ένας παίχτης, εντοπίστηκε τμήμα του υποβάθρου
12	1	0	Ο χαρακτήρας Fring φοράει γυαλιά
14	1	0	Στραμμένο πλάγια
15	3	0	Στραμμένα ελαφρώς πλάγια
19	3	0	δεν κοιτούν προς την κάμερα (1 πρόσωπο δεν κοιτά καθόλου, δεν συμπεριλήφθηκε)
20	5	0	Τα 5 πρόσωπα έχουν έντονες εκφράσεις χαράς, πχ ανοιχτό στόμα (3 μερικώς κρυμμένα πρόσωπα δεν συμπεριλήφθηκαν)
24	1	0	Ελαφρώς καλυμμένο με μαλλιά
25	0	0	Το πρόσωπο του Hulk δεν συμπελήφθηκε
26	1	0	έντονη έκφραση (χαμόγελο)
27	2	0	1 πρόσωπο κρύβεται μερικώς από μαλλιά
29	5	0	Η γνωστή οσκαρική selfie του 2014 είναι ελαφρώς κουνημένη, ενώ 5 πρόσωπα είναι μερικώς ορατά
30	6	0	2 πρόσωπα σε μικρή ανάλυση, 4 πλάγια στραμμένα
31	0	0	Ένα πρόσωπο με μουσουλμανική μαντίλα

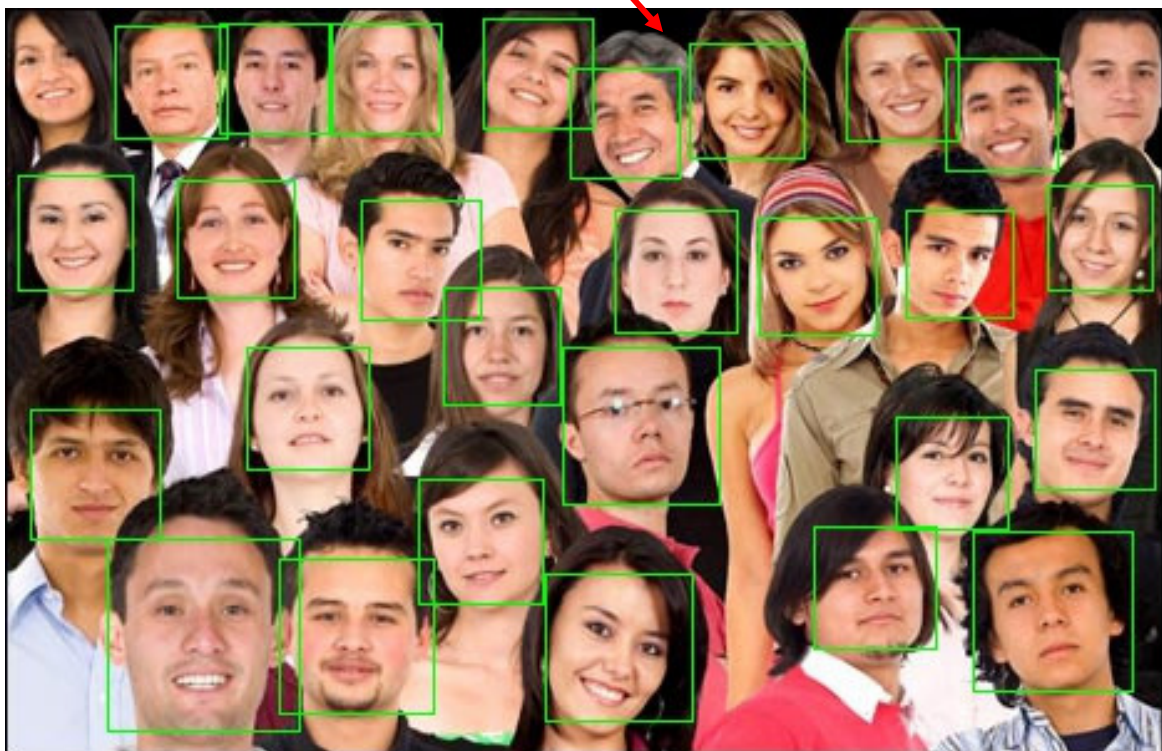
			εντοπίστηκε επιτυχώς
34	5	0	Δοκιμαστική φωτογραφία, καθώς οι περισσότεροι χαρακτήρες φορούν γυαλιά ηλίου, ωστόσο 1 εντοπίζεται
37	1	0	
39	0	0	Τα πρόσωπα με αφρικανικά χαρακτηριστικά εντοπίζονται επιτυχώς
40	1	0	Τα πρόσωπα με ασιατικά χαρακτηριστικά εντοπίζονται επιτυχώς πλην ενός, πιθανόν λόγω έκφρασης (στραβό χαμόγελο)
42	1	0	
43	0	0	Πρόσωπα που δεν είναι ανθρώπινα, όπως οι Yoda, Chewbacca, Darth Vader, δεν εντοπίζονται
45	2	0	Ασπρόμαυρη με μεγάλη ψηφιακή επεξεργασία
47	2	0	1 πρόσωπο αρκετά σκιασμένο στη μία πλευρά
48	1	1	Το FP είναι σε πρόσωπο με έντονη έκφραση (ανοιχτό στόμα)
49	2	0	Στην ιστορική φωτογραφία, χαμηλής ανάλυσης, ο Στάλιν δεν εντοπίζεται, όπως και δύο καθαρά πρόσωπα στην πίσω σειρά
50	2	0	

```
>> Total 2 face object detected on image.  
>> Parsing image file (wgzltbuujjt2ct1kyx4j.jpg).  
>> Face object detected.  
>> Total 5 face object detected on image.  
>> Parsing image file (yalta.jpg).  
>> Face object detected.  
>> Total 5 face object detected on image.  
>> Parsing image file (_47167695_faces_jun_466.jpg).  
>> Face object detected.  
>> Total 27 face object detected on image.  
  
C:\Users\Giannis\Desktop\PeopleDetect\peopleDetect_visual  
t.exe (process 8256) exited with code 0.  
Press any key to close this window . . .
```

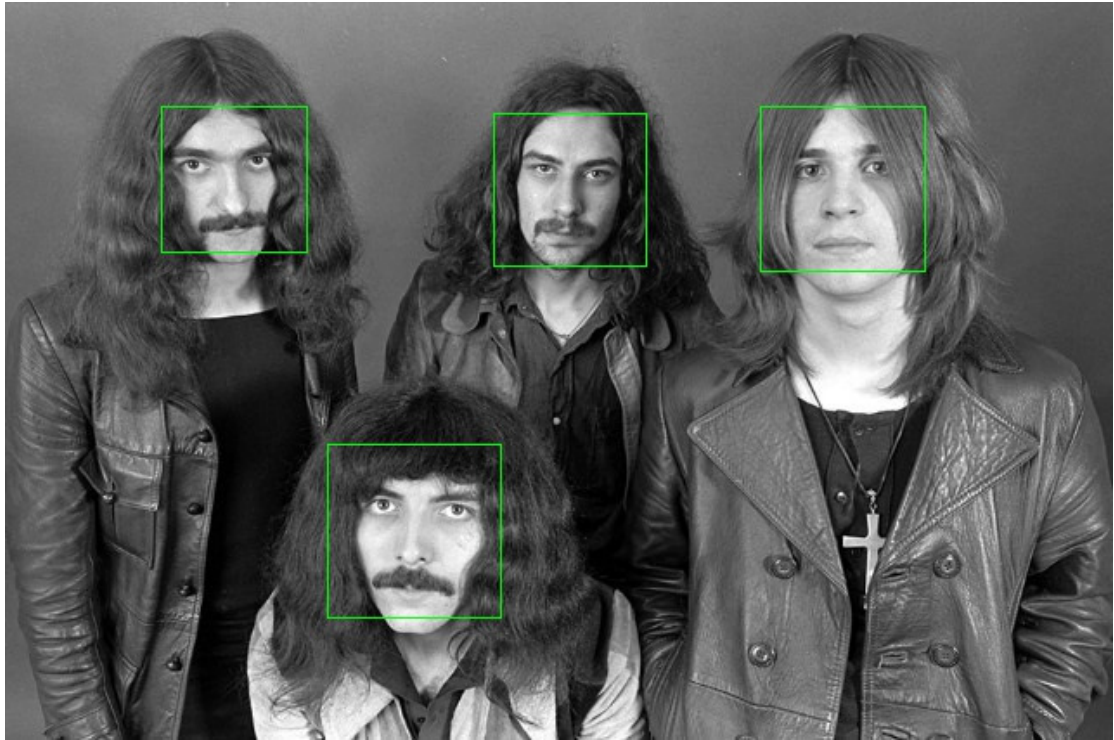


Εικόνα 18. Σωστό αποτέλεσμα αλγορίθμου.

```
>> Total 2 face object detected on image.  
>> Parsing image file (wgzltbuujjt2ct1kyx4j.jpg).  
>> Face object detected.  
>> Total 5 face object detected on image.  
>> Parsing image file (yalta.jpg).  
>> Face object detected.  
>> Total 5 face object detected on image.  
>> Parsing image file (_47167695_faces_jun_466.jpg).  
>> Face object detected.  
>> Total 27 face object detected on image.  
  
C:\Users\Giannis\Desktop\PeopleDetect\peopleDetect_visual  
t.exe (process 8256) exited with code 0.  
Press any key to close this window . . .
```



Εικόνα 19. Λάθος αποτέλεσμα αλγορίθμου. Εντοπίστηκαν 2 λιγότερα άτομα



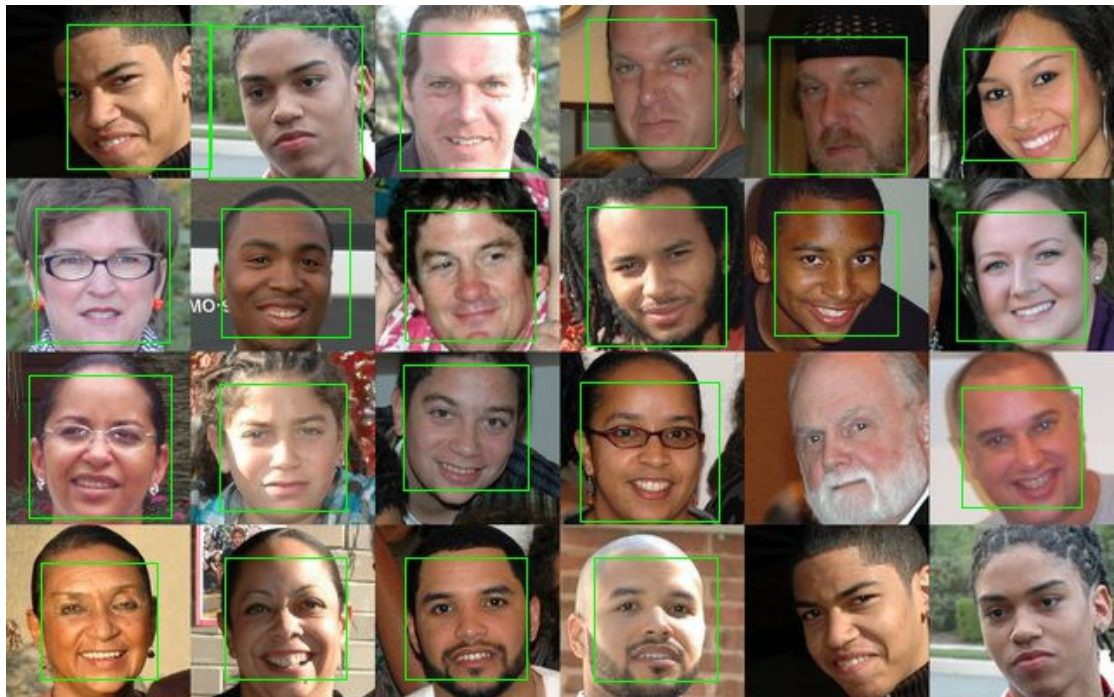
Εικόνα 20. Αποτέλεσμα εκτέλεσης



Εικόνα 21. Αποτέλεσμα εκτέλεσης



Εικόνα 22. Αποτέλεσμα εκτέλεσης



Εικόνα 23. Αποτέλεσμα εκτέλεσης

Πείραμα 2ο

Από τη σελίδα imagenet (<http://image-net.org/>) κατεβάσαμε τα εξής σύνολα δεδομένων:

Όνομα	Κατηγορία	Εικόνες
n09619168	Female	1371
n09624168	Male	1407
n12205694	Plants	1400

Από τις εικόνες των γυναικών διαγράψαμε όσες δεν έδειχναν ένα πρόσωπο να κοιτάζει προς την κάμερα και κρατήσαμε 500 από αυτές. Ομοίως και για τους άντρες, οπότε δημιουργήσαμε ένα φάκελο με 1000 εικόνες προσώπων. Τέλος, για τις εικόνες μη-προσώπων, διαλέξαμε 1000 εικόνες με φυτά.

Εκτελέσαμε τον αλγόριθμο στα δύο αυτά σύνολα εικόνων. Στην περίπτωση των ανθρώπινων προσώπων, ο αλγόριθμος ανίχνευσε τουλάχιστον 1 πρόσωπο σε 868 εικόνες από τις 1000.

Στην περίπτωση των φυτών ανιχνεύθηκαν 5 εικόνες με πρόσωπα στις 1000. Σημειώνεται ότι αρχικά ανιχνεύθηκαν 7 εικόνες, αλλά παρατηρήσαμε ότι όντως σε δύο από αυτές εμφανίζονταν πρόσωπα, οπότε αντικαταστάθηκαν.

Άρα, από αυτό το μεγάλο σύνολο δεδομένων, μπορούμε να εκτιμήσουμε καλύτερα την ακρίβεια και την ευαισθησία του αλγορίθμου:

$$\text{Accuracy} = 868 / 1000 = 86.8\%.$$

$$\text{Sensitivity} = 99,5\%.$$

Οι τιμές αυτές μοιάζουν πολύ με του 1^{ου} πειράματος. Σημειώνουμε πάντως για την περίπτωση της ακρίβειας ότι μετρήσαμε πόσες φορές ανιχνεύθηκε τουλάχιστον ένα πρόσωπο, καθώς υπάρχουν περιπτώσεις όπου οι εικόνες έχουν αρκετά πρόσωπα, οπότε πιθανότατα δεν ανιχνεύθηκαν όλα, αλλά λόγω του όγκου των δεδομένων δεν μπορούμε να υπολογίσουμε την ακρίβεια όπως στο 1^ο Πείραμα. Επίσης, το σύνολο δεδομένων δεν είναι φωτογραφίες προσώπου, αλλά γενικά ανθρώπων, οπότε προσπαθήσαμε να κρατήσουμε φωτογραφίες που δείχνουν το πρόσωπο που μπορεί να το ξεχωρίσει κάποιος, δηλαδή φαίνεται ολόκληρο στην φωτογραφία, και απορρίψαμε εικόνες όπου το πρόσωπο είναι μερικώς κρυμμένο, π.χ. είναι στραμμένο πλάγια.



Εικόνα 24. Αποτέλεσμα εκτέλεσης αλγορίθμου με φυτά. Με την κόκκινη ακμή σημειώνεται η εσφαλμένη έξοδος και με την πράσινη η σωστή έξοδος.



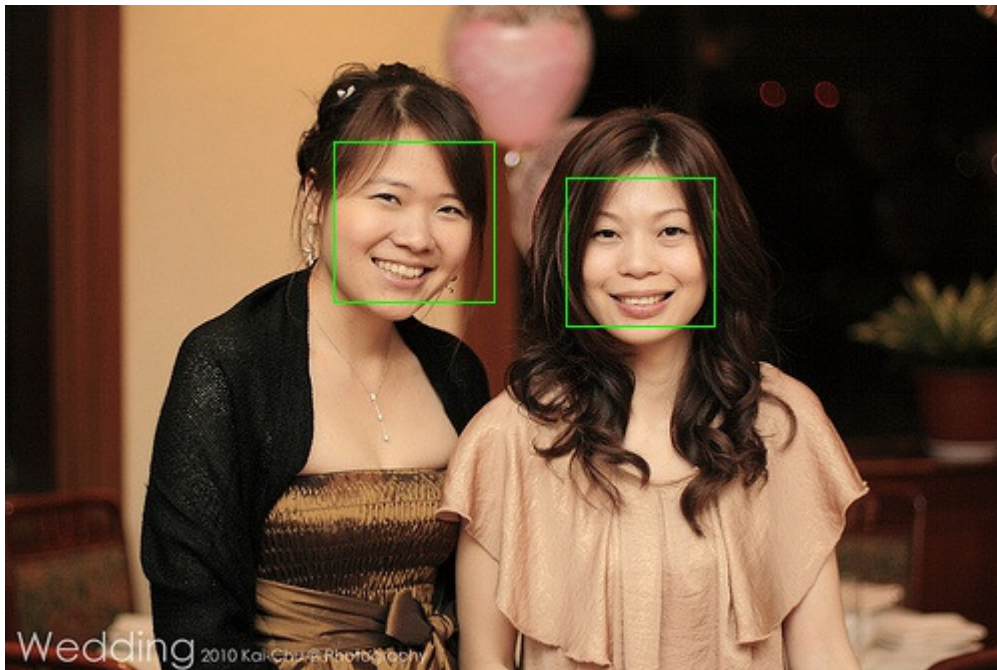
Εικόνα 25. Αποτέλεσμα εκτέλεσης



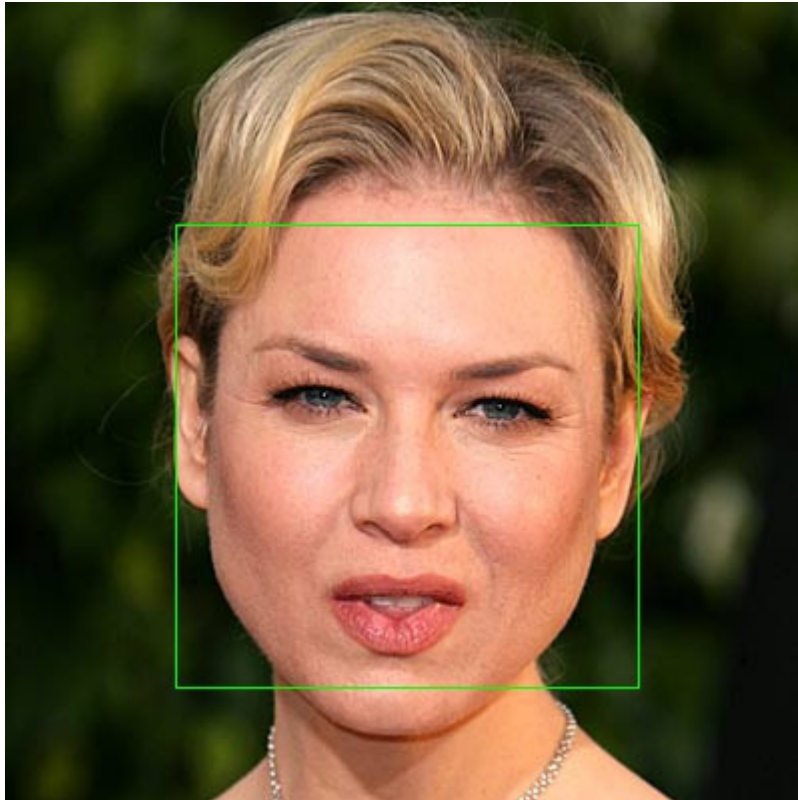
Εικόνα 26. Αποτέλεσμα εκτέλεσης



Εικόνα 27. Αποτέλεσμα εκτέλεσης



Εικόνα 28. Αποτέλεσμα εκτέλεσης



ΠΑΡΑΡΤΗΜΑ

Παρακάτω παρουσιάζεται ο κώδικας που υλοποιήθηκε στην παρούσα εργασία:

```
#include "stdafx.h"
#include "opencv2/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <opencv2/core/core.hpp>
#include <opencv2/ml/ml.hpp>
#include <fstream>
#include <iostream>
#include "string"
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/objdetect/objdetect.hpp"

using namespace std;
using namespace cv;
using namespace cv::ml;

Mat labels;
Mat state;
Mat testData;
/* φορτώνουμε το xml για τον ανιχνευτή προσώπων adaboost */
string face_cascade_name =
"C:/opencv/sources/data/haarcascades/haarcascade_frontalface_alt2.xml";
CascadeClassifier face_cascade;
/* ορίζουμε τον φάκελο στον οποίο έχουμε τις εικόνες που θα ελέγξουμε για την
ύπαρξη προσώπων */
string source = "C:/Image_data/";
/* ορίζουμε το αρχείο στο οποίο έχουμε τα ονόματα των αρχείων των εικόνων που
θα ελέγξουμε για την ύπαρξη προσώπων */
string data_files = "data.csv";
std::vector<String> image_sources;

int read_data(string source) {

    std::ifstream myfile(source+ data_files);
    if (!myfile)
    {
        std::cout <<">> Error opening output file"<< std::endl;
        return -1;
    }
    std::string line;
    while (std::getline(myfile, line))
    {
        //γεμίζουμε ένα vector με τα ονόματα των προς εξέταση εικόνων

        image_sources.push_back(line);
    }

    return 0;
}
```

```

int main(int argc, char** argv)
{
    string dir;

    try
    {
        std::cout <<">> Initializing..."<< endl <<">> Loading OpenCV
cascades."<< endl; /* Φορτώνουμε τον ανιχνευτή adaboost. */
        if(!face_cascade.load(face_cascade_name)) {
            std::cout << (">>Error OpenCV loading Cascade\n") <<
endl;
                return -1;
        }
        std::cout <<">> Cascades loaded."<< endl;
        std::cout <<">> Select folder (empty for default:
'C:/Image_data/', should contain 'data.csv')"<< endl;
        getline(std::cin, dir);
        if(!dir.empty()) {
            source = dir;
        }
        else {
            std::cout <<"Default: ";
        }
        cout <<" "<< source <<" "<< endl;
        //φορτώνουμε τα δεδομένα των εικόνων.
        read_data(source);
        std::cout <<">> Image file names loaded. (Total images "<<
image_sources.size() <<" "<< endl;
        string mkdir_cmd = "mkdir \""+ source +"results\"";
        system(mkdir_cmd.c_str());
        /*δημιουργούμε αρχείο στο οποίο θα αποθηκεύσουμε τα αποτελέσματα από
την ανίχνευση προσώπων των εικόνων*/
        std::ofstream outfile;
        outfile.open(source +"results/results.csv");
        outfile <<"Image;Faces\n";

        for (int i = 0; i < image_sources.size(); i++)
        {
            Mat image;
            /*Δημιουργία ενός vector με ορθογώνια για τον εντοπισμό
τμημάτων της εικόνας εισόδου*/
            std::vector<Rect> faces;
            /* Open image. */
            image = imread(source + image_sources[i]);
            if(!image.data)
            {
                std::cout <<">> Could not open image file ("<<
image_sources[i] <<")."<<std::endl;
                return -1;
            }
            std::cout <<">> Parsing image file ("<< image_sources[i]
<<")."<<std::endl;
            try
            {
                /*εντοπίζουμε αντικείμενα σε διάφορα μεγέθη στην εικόνα εισόδου. Το όρισμα
image περιέχει την προς εξέταση εικόνα. Το faces περιέχει τα ορθογώνια για τον
ορισμό των τμημάτων που περιέχουν το εντοπισμένο πρόσωπο. Το 1.3 είναι η
παράμετρος που καθορίζει πόσο μειώνεται το μέγεθος εικόνας σε κάθε κλίμακα
εικόνας (επιλέχθηκε το 1.3 γιατί είχε καλή απόδοση σε σχέση με τον χρόνο
εκτέλεσης). Το 4 είναι η παράμετρος που καθορίζει τον αριθμό των γειτόνων που
πρέπει να διατηρήσει για κάθε υποψήφιο ορθογώνιο. Ο classifier που

```

χρησιμοποιείται είναι ο CV_HAAR_SCALE_IMAGE. Το τελευταίο όρισμα καθορίζει το μέγεθος του αντικειμένου που θα ανιχνευτεί.

```
        face_cascade.detectMultiScale(image, faces, 1.3,
4, 0 | CV_HAAR_SCALE_IMAGE, Size(30, 30));
        std::cout <<">> Face object detected."<< endl;
    }
    catch (exception ex)
    {
        std::cout <<"detectMultiScale error"<< endl;
    }
    std::cout <<">> Total "<< faces.size() <<" face object
detected on image."<< endl;
    outfile << image_sources[i] <<"<<faces.size() <<"\n";
    /* Parse all faces. */
    for (int j = 0; j <faces.size(); j++)
    {
        /* Locate each face. */
        cv::rectangle(image, Point(faces[j].x,
faces[j].y), Point(faces[j].x + faces[j].width, faces[j].y +
faces[j].height), Scalar(0, 255, 0), 1, 8, 0);
    }
    /* εγγραφή των αποτελεσμάτων σε αρχείο. */
    imwrite(source +"/results/" + image_sources[i], image);

    cv::waitKey(10);
}
outfile.close();
}
catch (exception& e)
{
    std::cout <<"\nexception thrown!"<< endl;
    std::cout << e.what() << endl;
}

return 0;
}
```