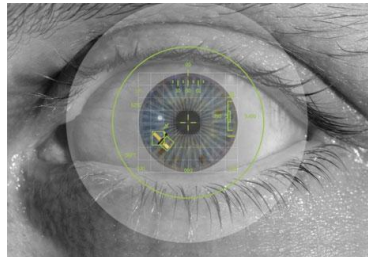


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗΝ ΑΝΑΓΝΩΡΙΣΗ
ΟΠΤΙΚΩΝ ΣΗΜΑΤΩΝ ΜΕΣΩ ΒΛΕΦΑΡΙΣΜΑΤΟΣ
(BLINK DETECTION)**



Πτυχιακή εργασία

Αναστασία Αγγελίδου, Α.Μ. 2519

Θωμά Τσούκα, Α.Μ. 2402

Επιβλέπων: Ιωάννης Καλόμοιρος, Επ. Καθηγητής

Σέρρες, 2013

Περίληψη

Η παρούσα πτυχιακή εστιάζει στην αλληλεπίδραση του ανθρώπου με τον υπολογιστή. Σε περιπτώσεις που ο χρήστης πάσχει από αναπηρία, εμφανίζεται η ανάγκη αλληλεπίδρασης με το υπολογιστικό σύστημα χωρίς τις συμβατικές διεπαφές (ποντίκι, πληκτρολόγιο, κτλ). Στην εργασία αυτή αναπτύχθηκε λογισμικό για την αναγνώριση του βλεφαρίσματος των ματιών (ανοιγόκλεισμα) από τη web-camera γραμμένη στη γλώσσα προγραμματισμού C++ και με τη χρήση της βιβλιοθήκης αλγορίθμων επεξεργασίας εικόνας και όρασης υπολογιστών OpenCV. Αρχικά περιγράφονται διάφορες τεχνικές αναγνώρισης προσώπου και χαρακτηριστικών καθώς και η τεχνική που χρησιμοποιήθηκε για την ανάπτυξη του λογισμικού αναγνώρισης βλεφαρίσματος. Τέλος, εξηγούνται τα βήματα υλοποίησης του προγράμματος, τα προβλήματα που προέκυψαν και τα συμπεράσματα που βγήκαν.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	2
Περιεχόμενα.....	3
Εικόνες.....	5
1. Τι είναι η τεχνητή όραση.....	8
2. Τεχνικές ανίχνευσης προσώπου και χαρακτηριστικών.....	13
2.1 Εισαγωγή.....	13
2.2 Προσέγγιση Βασισμένη στα Χαρακτηριστικά.....	15
2.2.1 Ακμές.....	15
2.2.2 Κατάτμηση Χρώματος.....	16
2.2.3 Μοντέλα Κατανεμημένων Σημείων (Point Distributed Models)	18
2.2.4 Snakes.....	19
2.3 Προσέγγιση Βασισμένη στην Εικόνα.....	20
2.3.1 Νευρωνικά Δίκτυα.....	20
2.3.2 Μηχανές Διανυσμάτων Στήριξης (Support Vector Machine).....	22
2.3.3 Eigenfaces.....	23
3. C++.....	25
3.1 Εισαγωγή.....	25
3.2 Φιλοσοφία.....	26
4. OpenCV.....	27
4.1 Εισαγωγή.....	27
4.2 Ποιοι χρησιμοποιούν την OpenCV.....	27
4.3 Προέλευση	28
4.4 Δομή και Περιεχόμενο.....	30
4.5 Εφαρμογές.....	31
5. Ανίχνευση προσώπου με Haar χαρακτηριστικά.....	32
5.1 Εισαγωγή.....	32
5.2 Χαρακτηριστικά.....	33
5.3 Ακέραια Εικόνα (Integral Image).....	34
5.4 Αλγόριθμος Εκπαίδευσης AdaBoost.....	36

5.5 Καταρράκτης Ταξινομητών.....	38
6. Περιγραφή εφαρμογής αναγνώρισης βλεφαρίσματος.....	41
7. Ανάλυση εφαρμογής.....	46
7.1 Άνοιγμα κάμερας και λήψη πλαισίου.....	46
7.2 Διαδικασία εύρεσης βλεφαρίσματος.....	46
7.3 Άθροισμα απόλυτων διαφορών.....	46
7.4 Κατωφλίωση.....	47
7.5 Μετατροπή αποτελέσματος σε ακέραιο αριθμό.....	48
7.6 Εξίσωση ιστογράμματος.....	49
7.7 Συνάρτηση ανίχνευσης προσώπου.....	49
7.8 Ανίχνευση περιοχής ματιών.....	51
8. Πειράματα.....	53
8.1 Haar ανιχνευτής ματιών.....	53
8.2 Λανθασμένες ανιχνεύσεις.....	54
9. Συμπεράσματα.....	58
9.1 Σε σχέση με την OpenCV.....	58
9.2 Σε σχέση με την εφαρμογή.....	58
9.3 Μελλοντικές βελτιώσεις.....	59
Βιβλιογραφία.....	61
Παράρτημα.....	64
A. Κώδικας.....	64
B. Οδηγίες εγκατάστασης OpenCV.....	68

EΙΚΟΝΕΣ

Εικόνα 1-1 Όραση Υπολογιστή.....	9
Εικόνα 1-2 Οι αλγόριθμοι ανασύστασης της χωρικής δομής από την κίνηση.....	10
Εικόνα 1-3 Αποτέλεσμα ανασύστασης της τρισδιάστατης εικόνας με χρήση στερεοσκοπικής αντιστοίχισης	10
Εικόνα 1-4 Εντοπισμός Ανθρώπου.....	11
Εικόνα 1-5 Ανίχνευση Προσώπου.....	11
Εικόνα 2-1 Τυποποιημένες εικόνες εκπαίδευσης για αναγνώριση προσώπου.....	14
Εικόνα 2-2 Ένα ρεαλιστικό σενάριο ανίχνευσης προσώπου.....	14
Εικόνα 2-3 Αρχική εικόνα.....	16
Εικόνα 2-4 Φίλτρο Sobel.....	16
Εικόνα 2-5 Φίλτρο Canny.....	16
Εικόνα 2-6 Ανίχνευση με μοντέλα κατανεμημένων σημείων.....	19
Εικόνα 2-7 Παράδειγμα ανίχνευσης προσώπου με snakes.....	20
Εικόνα 2-8 Η μέθοδος του Rowley.....	22
Εικόνα 2-9 Γραφική παράσταση Μηχανής Διανυσμάτων Στήριξης.....	23
Εικόνα 2-10 Πιθανά πρόσωπα με Eigenfaces.....	24
Εικόνα 4-4 Βιβλιοθήκες της OpenCV.....	30
Εικόνα 5-1 Χαρακτηριστικά τύπου Haar.....	34
Εικόνα 5-2 Τιμή της ακέραιας εικόνας στο σημείο (x,y).....	35
Εικόνα 5-3 Άθροισμα εικονοστοιχείων για ακέραια εικόνα.....	36
Εικόνα 5-4 Χαρακτηριστικά AdaBoost.....	38
Εικόνα 5-5 Καταρράκτης ανίχνευσης.....	40
Εικόνα 6-1 Λήψη πλαισίου από κάμερα.....	42
Εικόνα 6-2 Μετατροπή του πλαισίου σε κλίμακα του γκρι.....	42
Εικόνα 6-3 Εξισορρόπηση ιστογράμματος.....	43
Εικόνα 6-4 Εντοπισμός προσώπου.....	43
Εικόνα 6-5 Εντοπισμός ματιών.....	43
Εικόνα 6-6 Εξαγωγή περιοχής ματιών.....	44
Εικόνα 6-7 Το αποτέλεσμα από το άθροισμα απόλυτων διαφορών.....	44

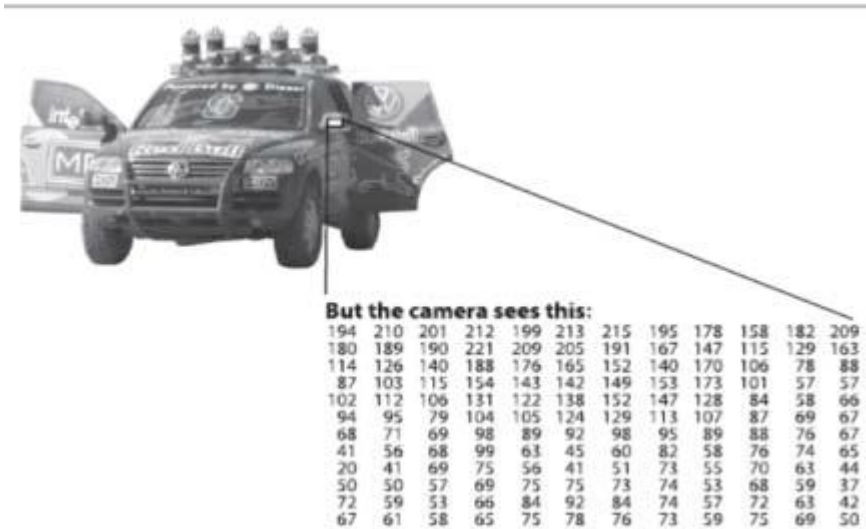
Εικόνα 6-8	Το αποτέλεσμα από την κατωφλίωση όταν υπάρχει ανοιγόκλεισμα ματιών	44
Εικόνα 6-9	Το τελικό αποτέλεσμα όταν υπάρχει ανίχνευση βλεφαρίσματος.....	45
Εικόνα 7-1	Το ορθογώνιο των ματιών σχηματικά.....	52
Εικόνα 8-1	Ανιχνευτής προσώπου και ματιών όταν τα μάτια είναι ανοιχτά.....	53
Εικόνα 8-2	Ανιχνευτής προσώπου και ματιών όταν τα μάτια είναι κλειστά.....	54
Εικόνα 8-3	Ανίχνευση περισσότερων από ένα ζευγάρι μάτια λόγω κακού φωτισμού...	55
Εικόνα 8-4	Μη ανίχνευση βλεφαρίσματος λόγω κακού φωτισμού.....	55
Εικόνα 8-5	Βλεφαρίσμα το οποίο ανιχνεύτηκε αλλά δεν εμφανίζεται στο παράθυρο...	56
Εικόνα 8-6	Ανίχνευση βλεφαρίσματος από την κίνηση της κόρης του ματιού.....	56
Εικόνα Π-1	Extract εκτελέσιμου αρχείου.....	68
Εικόνα Π-2	Μεταβλητές περιβάλλοντος.....	69
Εικόνα Π-3	Δημιουργία καινούργιου project.....	69
Εικόνα Π-4	Προσθήκη include directories.....	70
Εικόνα Π-5	Εισαγωγή βιβλιοθηκών.....	71

1. Τι είναι η τεχνητή όραση

Τεχνητή όραση είναι η μετατροπή των δεδομένων από μια φωτογραφική μηχανή ή μια κάμερα, σε ένα συμπέρασμα ή μια νέα αναπαράσταση. Οι μετατροπές αυτές γίνονται για την επίτευξη ορισμένων στόχων. Τα συμπεράσματα μπορεί να είναι “υπάρχει ένα πρόσωπο σε αυτή τη εικόνα” ή “υπάρχουν 14 καρκινικά κύτταρα σε αυτήν τη διαφάνεια”. Μια νέα αναπαράσταση μπορεί να είναι η μετατροπή μιας έγχρωμης εικόνας σε ασπρόμαυρη ή η αφαίρεση της κίνησης της κάμερας από μια ακολουθία εικόνων.

Επειδή είμαστε οπτικά πλάσματα, είναι εύκολο να μας δημιουργηθεί η λανθασμένη εντύπωση ότι η τεχνητή όραση είναι κάτι απλό. Πόσο δύσκολο είναι να βρεθεί, για παράδειγμα, ένα αυτοκίνητο σε μια εικόνα; Η αρχική σας διαίσθηση ίσως είναι αρκετά παραπλανητική. Ο ανθρώπινος εγκέφαλος διαχωρίζει το σήμα της όρασης σε πολλά κανάλια, τα οποία δίνουν διαφορετικά είδη πληροφοριών. Έχει ένα σύστημα αναγνώρισης το οποίο αναδεικνύει ποια είναι τα σημαντικά τμήματα της εικόνας για εξέταση και μειώνει την σημασία των άλλων τμημάτων. Αυτό που είναι δεδομένο είναι ότι η διαδικασία της όρασης αποτελείται από πολύ σύνθετες παραμέτρους οι οποίες, μέχρι στιγμής, είναι ελάχιστα κατανοητές. Υπάρχουν διάσπαρτες συνειρμικές εισαγωγές από τους αισθητήρες ελέγχου των μυών των ματιών και όλες τις άλλες ανθρώπινες αισθήσεις που επιτρέπουν στον εγκέφαλο να λειτουργεί βάση συσχετίσεων, οι οποίες δημιουργήθηκαν έπειτα από χιλιάδες χρόνια εξέλιξης του ανθρώπινου είδους. Τα συστήματα ανατροφοδότησης πληροφοριών στον εγκέφαλο επενεργούν σε όλα τα στάδια της επεξεργασίας συμπεριλαμβανομένου του hardware των αισθητήρων, δηλαδή τα μάτια, τα οποία ελέγχουν μηχανικά το φωτισμό μέσω της ίριδας και την λήψη της εικόνας στην επιφάνεια του αμφιβληστροειδή.

Σε ένα σύστημα τεχνητής όρασης όμως, ο ηλεκτρονικός υπολογιστής λαμβάνει ένα σύνολο από αριθμούς από μια κάμερα και αυτό είναι όλο. Ως επί το πλείστον, δεν υπάρχει ενσωματωμένη αναγνώριση προτύπων, δεν υπάρχει αυτόματος έλεγχος εστίασης και διαφράγματος, και δεν υπάρχουν συσχετίσεις από αιώνες εξέλιξης. Τα συστήματα όρασης βρίσκονται ακόμη σε νηπιακό στάδιο.



Εικόνα 1-1 Όραση υπολογιστή

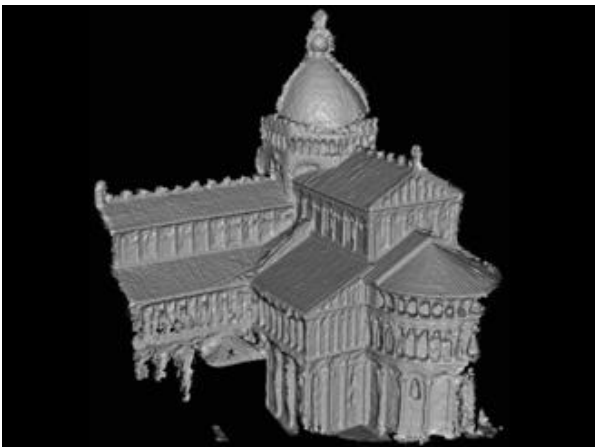
Στην εικόνα βλέπουμε ένα αυτοκίνητο και όπως μπορείτε να δείτε στην μεριά του οδηγού υπάρχει ένας καθρέπτης. Αυτό που βλέπει ένας υπολογιστής είναι ένας πίνακας από αριθμούς. Κάθε μεμονωμένος αριθμός του συγκεκριμένου πίνακα εμπεριέχει ένα αρκετά μεγάλο συστατικό θορύβου και έτσι από μόνος του μας δίνει λίγες πληροφορίες. Το έργο μας είναι να μετατραπεί αυτό το θορυβώδες σύνολο αριθμών στο συμπέρασμα «καθρέφτης».

Οι ερευνητές της μηχανικής όρασης έχουν αναπτύξει μαθηματικές τεχνικές για την ανάκτηση του τρισδιάστατου σχήματος και την εμφάνιση των αντικειμένων σε εικόνες. Πλέον έχουμε αξιόπιστες τεχνικές για τον ακριβή υπολογισμό τρισδιάστατων μοντέλων ενός περιβάλλοντος με χιλιάδες επικαλυπτόμενες φωτογραφίες (Εικόνα 1-2). Λαμβάνοντας υπόψη ένα αρκετά μεγάλο σύνολο απόψεων για ένα συγκεκριμένο αντικείμενο μπορούμε να δημιουργήσουμε ένα τρισδιάστατο μοντέλο επιφανειών χρησιμοποιώντας τη στεροσκοπική αντιστοίχιση (Εικόνα 1-3) Μπορούμε να εντοπίσουμε έναν άνθρωπο που κινείται σε ένα σύνθετο φόντο (Εικόνα 1-4). Μπορούμε ακόμη, με μέτρια επιτυχία, να προσπαθούμε να βρούμε και να ονομάσουμε όλους τους ανθρώπους σε μια φωτογραφία με το συνδυασμό προσώπου, ντυσίματος και μαλλιών (Εικόνα 1-5). Ωστόσο, παρ' όλες αυτές τις εξελίξεις, το όνειρο της κατοχής ενός υπολογιστή που ερμηνεύει μια εικόνα στο ίδιο επίπεδο με ένα δίχρονο παιδί (για παράδειγμα, να μετράει όλα τα ζώα σε μια εικόνα) παραμένει άπιαστο. Γιατί η όραση είναι τόσο δύσκολη; Εν

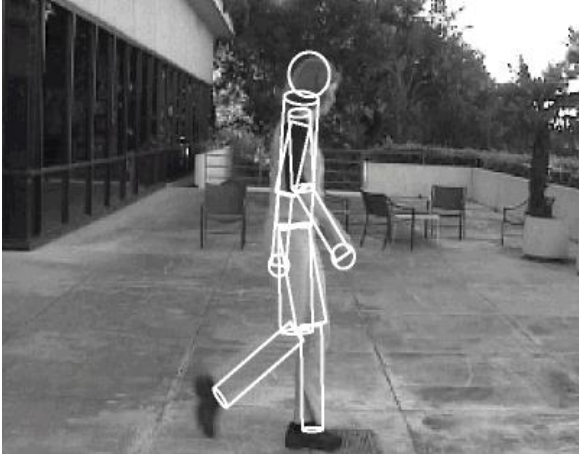
μέρει, αυτό συμβαίνει γιατί η όραση είναι ένα αντίστροφο πρόβλημα, στο οποίο προσπαθούμε να ανακτήσουμε κάποια άγνωστα πράγματα λόγω των μη επαρκών πληροφοριών για τον πλήρη προσδιορισμό της λύσης. Πρέπει λοιπόν να καταφύγουμε σε μοντέλα βασισμένα στη φυσική και στις πιθανότητες για να αποσαφηνιστούν ενδιάμεσες λύσεις.



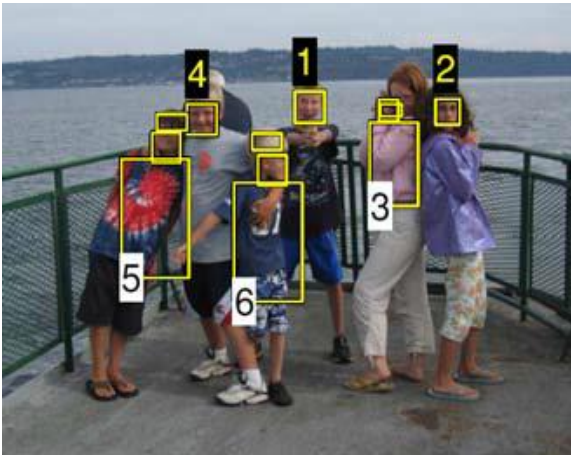
Εικόνα 1-2 Οι αλγόριθμοι ανασύστασης της χωρικής δομής από την κίνηση (structure from motion) μπορούν να ανακατασκευάσουν ένα τρισδιάστατο μοντέλο ενός μεγάλου πολύπλοκου σκηνικού από εκατοντάδες επικαλυπτόμενες φωτογραφίες.



Εικόνα 1-3 Αποτέλεσμα ανασύστασης της τρισδιάστατης εικόνας με χρήση στερεοσκοπικής αντιστοίχισης.



Εικόνα 1-4 Εντοπισμός Ανθρώπου



Εικόνα 1-5 Ανίχνευση Προσώπου

Τα μοντέλα που χρησιμοποιούνται στη τεχνητή όραση συνήθως αναπτύσσονται στη φυσική (ραδιομετρία, οπτικά και σχεδιασμός αισθητήρα) και στα γραφικά υπολογιστών. Και τα δύο αυτά πεδία μοντελοποιούν το πώς κινούνται και απεικονίζονται τα αντικείμενα, το πώς το φως αντανακλάται στις επιφάνειες τους, σκορπίζεται από την ατμόσφαιρα, διαθλάται μέσα από τους φακούς της κάμερας και τέλος προβάλλεται πάνω σε ένα επίπεδο (ή κοίλο) επίπεδο εικόνας. Στη τεχνητή όραση, προσπαθούν να κάνουν το αντίστροφο, δηλαδή, να περιγραφεί ο κόσμος που βλέπουμε σε μία ή περισσότερες εικόνες και να ανακατασκευάσουν τις ιδιότητες του, όπως το σχήμα, το φωτισμό και τη διανομή χρώματος. Είναι εκπληκτικό το ότι οι άνθρωποι και τα ζώα το κάνουν αυτό αβίαστα, ενώ οι αλγόριθμοι τεχνητής όρασης έχουν πιθανότητα σφάλματος.

Τα καλά νέα είναι ότι η τεχνητή όραση χρησιμοποιείται σήμερα σε μια ευρεία ποικιλία εφαρμογών οι οποίες περιλαμβάνουν:

- Τρισδιάστατο μοντέλο κτιρίων (φωτομετρία): πλήρως αυτοματοποιημένη κατασκευή τρισδιάστατων μοντέλων από αεροφωτογραφίες.
- Ιατρική απεικόνιση: ταξινόμηση κυττάρων, ανίχνευση όγκων, αναπαράσταση ανθρώπινων οργάνων, εγχείρηση με ρομπότ.
- Εφαρμογές αυτοκινήτου: αυτονομία οχημάτων, παρακολούθηση επαγρύπνησης του οδηγού, αποφυγή εμποδίων.
- Βιομηχανικοί αυτοματισμοί: βιομηχανική επιθεώρηση (ανίχνευση βλαβών και επιθεώρηση), διάβασμα barcode και ετικετών, ταξινόμηση εμπορευμάτων.
- Ρομποτική: αυτόματος καθορισμός θέσης, αποφυγή εμποδίων, έξυπνα ρομπότ αλληλεπιδρούν με τον άνθρωπο και τον εξυπηρετούν.
- Ασφάλεια: (ίριδα, δαχτυλικά αποτυπώματα, αναγνώριση προσώπου), επιτήρηση ύποπτων δραστηριοτήτων ή συμπεριφορών.
- Βάσεις δεδομένων εικόνας και βίντεο.

2. Τεχνικές ανίχνευσης προσώπου και χαρακτηριστικών

2.1 Εισαγωγή

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.

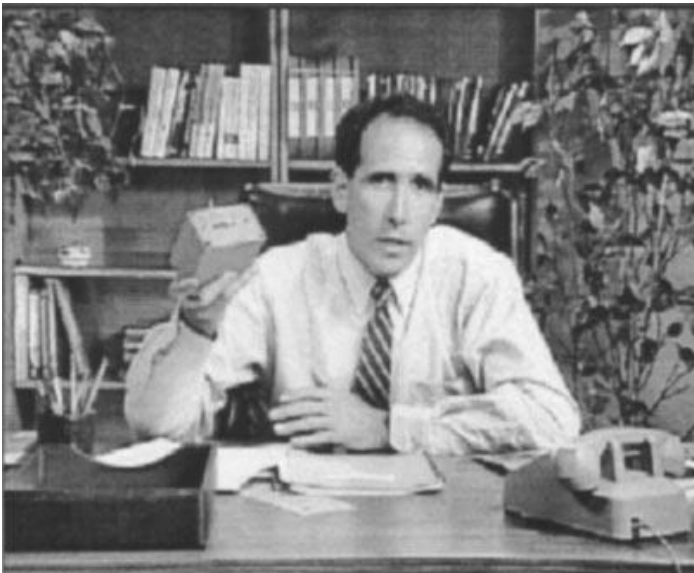
Πολλές από τις τρέχουσες τεχνικές αναγνώρισης προσώπου αναλαμβάνουν τη διαθεσιμότητα μετωπικών προσώπων ίδιου μεγέθους. Στην πραγματικότητα, αυτή η υπόθεση δεν μπορεί να ισχύει λόγω της ποικιλίας της εμφάνισης του προσώπου και των περιβαλλοντικών συνθηκών (Εικόνα 2-1). Αυτές οι εικόνες είναι τυπικές 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.



Εικόνα 2-1 Τυποποιημένες εικόνες εκπαίδευσης για αναγνώριση προσώπου

Η ανίχνευση προσώπου είναι μια από τις οπτικές διεργασίες όπου οι άνθρωποι μπορούν να κάνουν αβίαστα. Ωστόσο, σε όρους τεχνητής όρασης, αυτή η διεργασία δεν είναι εύκολη. Μια γενική τοποθέτηση του προβλήματος μπορεί να οριστεί ως εξής: Παίρνοντας μια ακίνητη εικόνα ή από βίντεο, να ανιχνευτεί και να εντοπιστεί ένας άγνωστος αριθμός προσώπων. Η λύση στο πρόβλημα περιλαμβάνει διαχωρισμό, εξαγωγή και επαλήθευση των προσώπων και ενδεχομένως τα χαρακτηριστικά του προσώπου από ένα μη ελεγχόμενο φόντο. Ένα σύστημα ανίχνευσης πρέπει επίσης να είναι σε θέση να επιτυγχάνει έργο ανεξάρτητα από τον φωτισμό, τον προσανατολισμό και την απόσταση της κάμερας.

Υπάρχουν δύο βασικές προσεγγίσεις για την ανίχνευση προσώπου: Η προσέγγιση με βάση τα χαρακτηριστικά και η προσέγγιση με βάση την εικόνα.



Εικόνα 2-2 Ένα ρεαλιστικό σενάριο ανίχνευσης προσώπου

2.2 Προσέγγιση βασισμένη στα χαρακτηριστικά

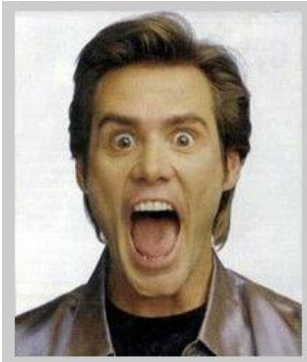
Αυτός ο τομέας περιλαμβάνει τεχνικές που έχουν χαρακτηριστεί ως ανάλυση χαμηλού επιπέδου. Είναι μέθοδοι που ασχολούνται με την κατάτμηση των οπτικών χαρακτηριστικών χρησιμοποιώντας ιδιότητες των εικονοστοιχείων όπως η κλίμακα του γκρι και το χρώμα. Τα χαρακτηριστικά που ανιχνεύουν αυτές οι μέθοδοι χαμηλού επιπέδου μπορεί να είναι διαφορεόμενα, αλλά είναι γρήγοροι και εύκολοι στο να εφαρμοστούν. Ένας άλλος τομέας από τεχνικές ονομάζεται ανάλυση χαρακτηριστικών, όπου η ανίχνευση βασίζεται σε χαρακτηριστικά του προσώπου με τη χρήση πληροφοριών της γεωμετρίας του προσώπου. Μέσα από την ανάλυση χαρακτηριστικών, οι ασάφειες των χαρακτηριστικών μειώνονται και προσδιορίζονται οι περιοχές του προσώπου και των χαρακτηριστικών. Αυτές οι τεχνικές έχουν αναπτυχθεί για την εξαγωγή μη άκαμπτων χαρακτηριστικών όπως τα μάτια και τα χείλη.

2.2.1 Ακμές

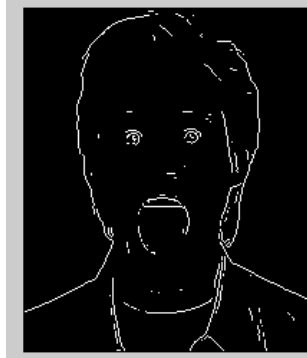
Είναι το πιο θεμελιώδες χαρακτηριστικό στις εφαρμογές ανίχνευσης προσώπου. Η περισσότερη δουλειά βασίστηκε στη βασική γραμμή των προσώπων από φωτογραφίες, με στόχο τον εντοπισμό των χαρακτηριστικών του προσώπου. Στη συνέχεια πραγματοποιήθηκαν εργασίες οι οποίες οδήγησαν στον εντοπισμό του εξωτερικού περιγράμματος του κεφαλιού. Αυτό ήταν η βάση του προγράμματος ανίχνευσης, όπου το επόμενο βήμα ήταν η ανάλυση των χαρακτηριστικών για να καθοριστεί αν το σχήμα που ανιχνεύθηκε είναι πράγματι το σχήμα του ανθρώπινου προσώπου. Η ανίχνευση ακμών εντοπίζει τις κύριες εξωτερικές γραμμές στην εικόνα (το κατώφλι μπορεί να ρυθμιστεί ώστε να ανιχνεύει τις κύριες γραμμές ή όλες τις γραμμές). Στη συνέχεια εκχωρεί κάθε εικονοστοιχείο της γραμμής σε ένα δυαδικό ψηφίο ώστε να τα ξεχωρίσει από το φόντο. Υπάρχουν πολλοί τύποι διαχείρισης ακμών. Όλοι λειτουργούν με το ίδιο σκεπτικό και δίνουν παρόμοια αποτελέσματα.

Σε μια ανίχνευση ακμών που βασίζεται στην προσέγγιση της ανίχνευσης προσώπου, οι ακμές (μετά από ταυτοποίηση) πρέπει να σημειωθούν και να ταιριάξουν σε ένα μοντέλο προσώπου προκειμένου να επαληθευθούν οι σωστές ανιχνεύσεις. Τα χαρακτηριστικά

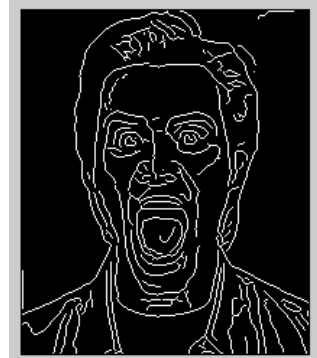
πρέπει να εντοπιστούν και να ταυτοποιηθούν όπως τα μάτια, η γραμμή των μαλλιών και η γραμμή του σαγονιού. Αν όλα αυτά φαίνεται να είναι σε αναλογία και στη θέση τους, τότε ανιχνεύεται ένα πρόσωπο. Αυτή η μέθοδος είναι ακριβής σε εικόνες χωρίς πολύπλοκο φόντο και το πρόσωπο πρέπει να είναι σε σαφή όψη κοιτώντας μπροστά. Αυτοί οι περιορισμοί μερικές φορές περιορίζουν την υλοποίηση ανίχνευσης ακμών ως ένα εργαλείο προ-επεξεργασίας για την αναγνώριση του σχήματος του προσώπου και στη συνέχεια αυτές οι φιγούρες παραδίδονται σε ένα σύστημα βασισμένο στα πρότυπα για μια διαδικασία ακριβότερης ανίχνευσης. Στις εικόνες 2-3, 2-4 και 2-5 φαίνονται μερικές ρουτίνες εύρεσης ακμών.



Εικόνα 2-3 Αρχική εικόνα



Εικόνα 2-4 Φίλτρο Sobel



Εικόνα 2-5 Φίλτρο Canny

2.2.2 Κατάτμηση χρώματος

Η ανίχνευση του χρώματος του δέρματος σε έγχρωμες εικόνες είναι μια πολύ δημοφιλής και χρήσιμη τεχνική για ανίχνευση προσώπου. Αυτή η ενότητα εξηγεί μια προσέγγιση για τον προσδιορισμό των μερών του χρωματισμένου δέρματος σε μια εικόνα. Έχουν αναφερθεί πολλές τεχνικές για τον εντοπισμό των περιοχών που υπάρχει δέρμα σε μια εικόνα. Ενώ το χρώμα εισόδου σε μια έγχρωμη εικόνα είναι στη μορφή RGB, το μοντέλο RGB δεν χρησιμοποιείται στη διαδικασία ανίχνευσης. Είναι γνωστό ότι το μοντέλο RGB δεν είναι αξιόπιστο για τον εντοπισμό του χρώματος του δέρματος. Αυτό συμβαίνει επειδή οι συνιστώσες του RGB υπόκεινται σε αλλαγή φωτεινότητας, σημαίνοντας ότι ίσως αποτύχει η ανίχνευση προσώπου αν οι συνθήκες φωτισμού αλλάζουν από εικόνα σε

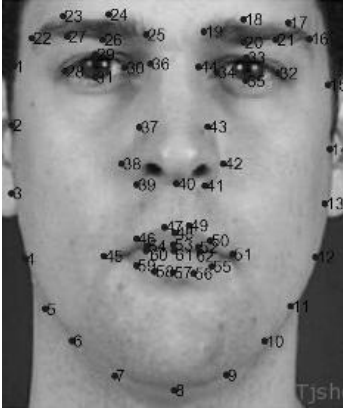
εικόνα. Η τεχνική συνήθως χρησιμοποιεί συνιστώσες χρώματος στο χρωματικό χώρο, όπως HSV ή YCbCr. Το μοντέλο χρώματος που χρησιμοποιείται είναι το μοντέλο YIQ, ένα καθολικό χρωματικό χώρο που χρησιμοποιείται για την έγχρωμη τηλεοπτική μετάδοση.

Σε αυτή την τεχνική, τοποθετείται η περιοχή του προσώπου σε ένα πιο σύνθετο πρόγραμμα αναγνώρισης προτύπων για τον προσδιορισμό του προσώπου. Η εικόνα δοκιμάζεται στο χώρο YIQ ώστε να εξεταστεί κάθε εικονοστοιχείο για τον καθορισμό και την ταξινόμηση του αν είναι δέρμα ή όχι. Χρησιμοποιείται ένας πίνακας αναζήτησης για να ταξινομήσει το «δέρμα» του κάθε εικονοστοιχείο, όπου εξετάζεται κάθε χρώμα για να φανεί αν βρίσκεται στην περιοχή του χρώματος του δέρματος και να συνδεθεί με το δυαδικό 1 αν είναι και με το 0 αν δεν είναι. Χρειάζεται ένα πλαίσιο για να καθοριστεί το εύρος και η θέση αυτών των τιμών που έχουν 1. Ο σκοπός της κατάτμησης χρώματος είναι να μειωθεί ο χώρος αναζήτησης των επακόλουθων τεχνικών, έτσι είναι σημαντικό να καθοριστεί τόσο σφιχτά το πλαίσιο χωρίς να κοπεί το πρόσωπο. Στη συνέχεια η έγχρωμη εικόνα χαρτογραφείται σε μια δυαδική εικόνα με 1 και 0 που αντιπροσωπεύουν τις περιοχές με δέρμα και χωρίς δέρμα. Είναι σύνηθες κατά τη διάρκεια της κατάτμησης χρώματος να επιστρέφονται τιμές που είναι κοντά στο δέρμα αλλά δεν είναι δέρμα ή άλλες περιοχές που έχουν χρώμα δέρματος και δεν είναι μέρος του προσώπου ή του σώματος. Αυτές οι λανθασμένες τιμές είναι γενικά απομονωμένα εικονοστοιχεία ή ομάδες εικονοστοιχείων που είναι δραματικά μικρότερα από τις συνολικές περιοχές του προσώπου. Συμπερίληψη αυτών των λάθος εικονοστοιχείων θα οδηγούσε σε ένα κουτί που είναι πολύ μεγαλύτερο απ' ό,τι προορίζεται και συντρίβει τον σκοπό της κατάτμησης. Περαιτέρω μορφολογικές βελτιώσεις εφαρμόζονται στην δυαδική έξοδο προκειμένου να μειωθούν ορισμένες από τις συνέπειες των θορυβωδών εικονοστοιχείων. Δεδομένου ότι τα ψευδή σφάλματα είναι γενικά πολύ μικρότερα απ' ό,τι η ίδια η περιοχή του προσώπου, μορφολογικές τεχνικές όπως η διάβρωση, το φιλτράρισμα και το κλείσιμο είναι καλά εργαλεία για την εξάλειψη αυτών των εικονοστοιχείων.

2.2.3 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 (labels). 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.

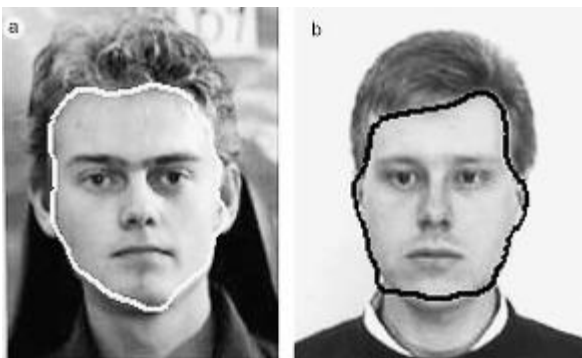
Το μοντέλο κατανεμημένων σημείων προσώπου αναπτύχθηκε ως ένα ευέλικτο μοντέλο. Αυτό το μοντέλο ορίζει ένα γενικό μοντέλο για ένα πρόσωπο το οποίο περιλαμβάνει χαρακτηριστικά προσώπου όπως φρύδια, μύτη και μάτια. Χρησιμοποιώντας 152 χειροκίνητα σημεία ελέγχου (x) και 160 εικόνες προσώπου εκπαίδευσης, αποκτήθηκε το μοντέλο κατανομής σημείου προσώπου. Σε σύγκριση με το σχηματικό μοντέλο μέσης τιμής το x τοποθετείται στην κορυφή (ή κοντά) στην περιοχή που δοκιμάζεται. Οι ετικέτες κάθε εικόνας συγκρίνονται μεταξύ τους. Κατά τη διάρκεια της σύγκρισης τα αντίστοιχα σημεία επιτρέπεται μόνο να διαφέρουν με έναν τρόπο που είναι σε συμφωνία με το σύνολο δεδομένων εκπαίδευσης. Τα γενικά χαρακτηριστικά του μοντέλου οδηγούν ώστε όλα τα χαρακτηριστικά μπορούν να ανιχνευθούν ταυτόχρονα και έτσι μειώνεται η ανάγκη για αναζήτηση χαρακτηριστικών, γλιτώνοντας χρόνο προ-επεξεργασίας. Άλλο πλεονέκτημα αυτής της τεχνικής είναι ότι μπορεί να ανιχνεύσει ένα πρόσωπο ακόμη κι αν ένα χαρακτηριστικό λείπει. Αυτό γίνεται λόγω του ότι άλλες συγκρίσεις χαρακτηριστικών μπορούν ακόμη να ανιχνεύσουν το πρόσωπο.



Εικόνα 2-6 Ανίχνευση με μοντέλα κατανεμημένων σημείων

2.2.4 Snakes

Τα snakes (ή active contours) συνήθως χρησιμοποιούνται για τον εντοπισμό του ορίου του κεφαλιού. Αυτό γίνεται κατά πρώτον με την αρχικοποίηση του snake στην εγγύτητα γύρω από το όριο του κεφαλιού. Το snake κλειδώνει πάνω στα κοντινά άκρα και στη συνέχεια παίρνει το σχήμα του κεφαλιού. Το μονοπάτι του snake προσδιορίζεται με την ελαχιστοποίηση μιας ενεργής συνάρτησης. E_{snake} , συμβολίζεται ως $E_{snake} = E_{internal} + E_{external}$, όπου $E_{internal}$, $E_{external}$ είναι οι εσωτερικές και εξωτερικές ενεργές συναρτήσεις. Η εσωτερική ενέργεια ορίζει τη φυσική εξέλιξη του snake και η εξωτερική εξουδετερώνει την εσωτερική ενέργεια για να μπορέσουν τα περιγράμματα να ξεχωρίζουν από τα κοντινά χαρακτηριστικά – ιδανικά το όριο της κεφαλής. Η ελαστική ενέργεια χρησιμοποιείται συνήθως ως εσωτερική ενέργεια – αυτό μπορεί να δώσει στο snake τον χαρακτηρισμό του ελαστικού δεσμού που προκαλεί την εξέλιξη του snake (μέσω της συρρίκνωσης και την επέκτασης). Η απαίτηση της εξωτερικής ενέργειας μπορεί να περικλείει μια συνάρτηση χρώματος δέρματος, η οποία προσελκύει την συνάρτηση χρώματος δέρματος στην περιοχή του προσώπου. Τα snakes είναι καλά εφοδιασμένα στην ανίχνευση ορίων χαρακτηριστικών αλλά και πάλι έχει προβλήματα. Τα περιγράμματα συχνά παγιδεύονται σε λάθος χαρακτηριστικά της εικόνας κι έτσι προκαλεί κόλλημα του προγράμματος. Τα snakes επίσης προσπαθούν να κρατήσουν την ελάχιστη καμπυλότητα και αυτό μπορεί να οδηγήσει σε προβλήματα όπως ότι το σχήμα μερικών προσώπων μπορεί να μην είναι εντελώς κυρτό.



Εικόνα 2-7 Παραδείγματα ανίχνευσης προσώπου με snakes

2.3 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.

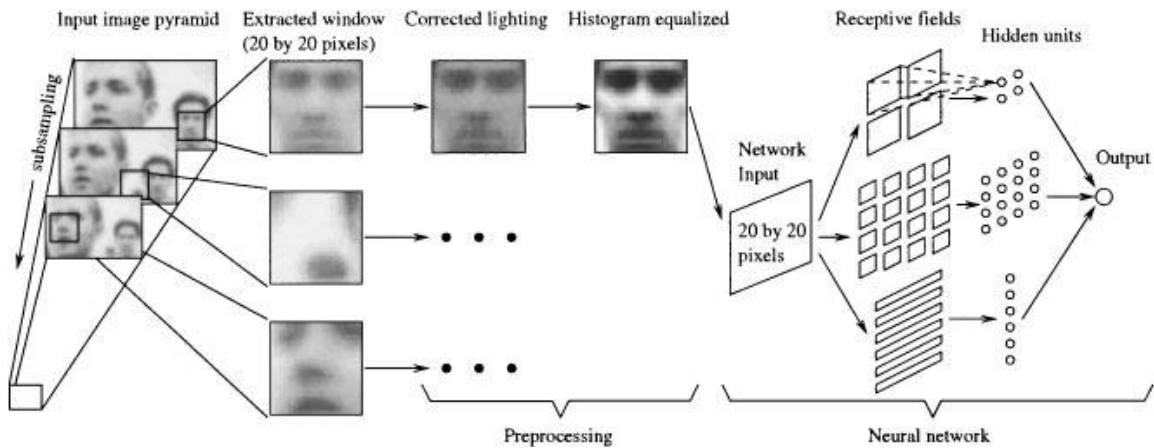
2.3.1 Νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα έχουν γίνει μια δημοφιλής τεχνική για την ανίχνευση προσώπου. Περιέχουν ένα στάδιο που αποτελείται από αισθητήρες πολλαπλών επιπέδων. Οι πρώτες νευρωνικές προσεγγίσεις βασίστηκαν σε πολύ-επίπεδους αισθητήρες που έδωσαν πολλά

υποσχόμενα αποτελέσματα με αρκετά απλά σύνολα δεδομένων. Η πρώτη προχωρημένη νευρωνική προσέγγιση η οποία αναφέρει αποτελέσματα σε ένα μεγάλο, δύσκολο σύνολο δεδομένων ήταν από το σύστημα Rowley. Το σύστημα Rowley [4] ενσωματώνει τη γνώση του προσώπου σε ένα αμφιβληστροειδές συνδεδεμένο νευρωνικό δίκτυο. Το νευρωνικό δίκτυο είναι σχεδιασμένο να κοιτάει σε παράθυρα των 20x20 εικονοστοιχείων. Υπάρχει ένα κρυμμένο επίπεδο με 26 μονάδες, όπου οι 4 μονάδες κοιτούν σε υποπεριοχές των 10x10 pixels, οι 16 κοιτούν σε υποπεριοχές των 5x5 εικονοστοιχείων, και οι 6 κοιτούν σε επικαλυπτόμενες οριζόντιες γραμμές των 20x5 εικονοστοιχείων. Το παράθυρο εισόδου είναι προ-επεξεργασμένο μέσω διόρθωσης φωτισμού και ιστόγραμμα εξισορρόπησης. Ένα πρόβλημα που προκύπτει με τις τεχνικές σάρωσης του παραθύρου είναι οι επικαλυπτόμενες ανιχνεύσεις. Ο De Silva [3] ασχολείται με το πρόβλημα μέσω δύο ευρετικών:

1. Κατωφλίωση: Μετράται ο αριθμός των ανιχνεύσεων σε μια μικρή περιοχή γύρω από την τρέχουσα θέση και αν είναι πάνω από ένα συγκεκριμένο κατώφλι, το πρόσωπο υπάρχει σ' αυτή την περιοχή.
2. Εξάλειψη επικαλύψεων: όταν μια περιοχή έχει ταξινομηθεί ως πρόσωπο σύμφωνα με την κατωφλίωση, τότε οι επικαλυπτόμενες ανιχνεύσεις είναι πιθανό να είναι ψευδώς θετικές και έτσι απορρίπτονται.

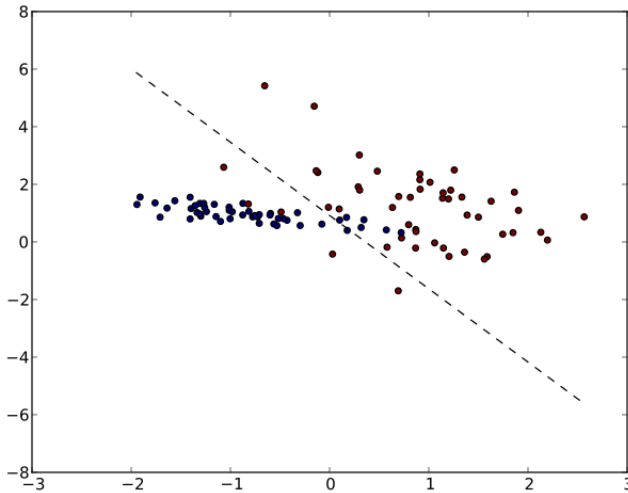
Κατά τη διάρκεια της εξάσκησης, ο στόχος για μια εικόνα προσώπου είναι η ανασυγκρότηση της ίδιας της εικόνας. Ο στόχος είναι να ρυθμιστεί η μέση τιμή των n κοντινότερων γειτόνων της εικόνας του προσώπου. Ένας αλγόριθμος εξάσκησης που βασίζεται στον αλγόριθμο εκκίνησης των Sung και Poggio [5]. Το σύστημα εκπαιδεύεται με έναν απλό κανόνα εκμάθησης το οποίο προωθεί και υποβιβάζει τα βάρη σε περιπτώσεις μη ταξινόμησης.



Εικόνα 2-8 Η μέθοδος του Rowley

2.3.2 Μηχανές Διανυσμάτων Στήριξης (Support Vector Machine)

Οι μηχανές διανυσμάτων στήριξης είναι εποπτικά μοντέλα μάθησης με σχετικούς αλγορίθμους μάθησης που αναλύουν δεδομένα και αναγνωρίζουν πρότυπα, που χρησιμοποιούνται για ταξινόμηση και ανάλυση παλινδρόμησης. Κατά βάση, η μηχανή διανυσμάτων στήριξης παίρνει ένα σύνολο από δεδομένα εισόδου και προβλέπει, για κάθε είσοδο, ποια από τις δύο πιθανές κλάσεις αποτελεί την έξοδο, καθιστώντας το έναν μη πιθανοτικό δυαδικό γραμμικό ταξινομητή. Δεδομένου ενός συνόλου παραδειγμάτων εκπαίδευσης, κάθε παράδειγμα χαρακτηρίζεται αν ανήκει σε μία από τις δύο κατηγορίες, ένας αλγόριθμος εκπαίδευσης χτίζει ένα μοντέλο που τοποθετεί τα νέα παραδείγματα είτε στη μία κατηγορία είτε στην άλλη. Ένα μοντέλο μηχανής διανυσμάτων στήριξης είναι η αναπαράσταση των παραδειγμάτων ως σημεία στο χώρο, χαρτογραφημένα ώστε τα παραδείγματα των επιμέρους κατηγοριών να χωρίζονται από ένα σαφές κενό. Στη συνέχεια τα καινούργια παραδείγματα χαρτογραφούνται στον ίδιο χώρο και προβλέπεται να ανήκουν σε μια κατηγορία με βάση σε ποια πλευρά του κενού θα πέσουν. Στην περίπτωση ανίχνευσης προσώπου, από τη μία πλευρά τοποθετούνται τα παραδείγματα που δεν έχουν πρόσωπο και στην άλλη πλευρά τα παραδείγματα που έχουν πρόσωπο (Εικόνα 2-9).



Εικόνα 2-9 Γραφική παράσταση Μηχανής Διανυσμάτων Στήριξης

2.3.3 Eigenfaces

Στα τέλη του '80 αναπτύχθηκε μια τεχνική που χρησιμοποιούσε ανάλυση σε κύριες συνιστώσες για την αποτελεσματική αναπαράσταση του ανθρώπινου προσώπου. Λαμβάνοντας υπόψη ένα σύνολο διαφορετικών εικόνων προσώπου, η τεχνική βρίσκει πρώτα τις κύριες συνιστώσες της κατανομής των προσώπων, εκφραζόμενη σε ιδιοδιανύσματα. Κάθε μεμονωμένο πρόσωπο στο σύνολο των προσώπων μπορεί να προσεγγιστεί από έναν γραμμικό συνδυασμό των μεγαλύτερων ιδιοδιανυσμάτων, πιο συχνά ονομάζονται eigenfaces, χρησιμοποιώντας κατάλληλα βάρη. Τα eigenfaces καθορίζονται από την εκτέλεση ενός κύριου συστατικού ανάλυσης σε ένα σύνολο παραδειγμάτων με κεντρικές όψεις στο ίδιο μέγεθος. Με τη μετακίνηση ενός παραθύρου καλύπτοντας ένα υπό παράθυρο για ολόκληρη την εικόνα τα πρόσωπα μπορούν να εντοπιστούν μέσα σε όλη την εικόνα. Ένα προ μεταγλωττισμένο σύνολο φωτογραφιών περιλαμβάνεται στο σύνολο εκπαίδευσης και από αυτό το σύνολο εκπαίδευσης εξάγονται τα eigenfaces. Οι φωτογραφίες στο σύνολο εκπαίδευσης αντιστοιχίζονται σε ένα άλλο σύνολο που είναι τα eigenfaces. Όπως και με οποιαδήποτε άλλη χαρτογράφηση στα μαθηματικά, μπορούμε να σκεφτούμε τα δεδομένα (τις φωτογραφίες και τα eigenfaces) σαν να υπάρχουν σε δύο τομείς. Οι φωτογραφίες στο σύνολο εκπαίδευσης είναι ένας

τομέας, και τα eigenfaces περιλαμβάνουν τον δεύτερο τομέα που συχνά αναφέρεται ως χώρος προσώπου. Στο σχήμα φαίνεται ένα παράδειγμα από eigenfaces που μπορούν να παραχθούν για να φτιάξουν στον χώρο προσώπου.

Τα eigenfaces που αποτελούν το χώρο προσώπου προσθέτονται μαζί με τα κατάλληλα βάρη για την επανασύνθεση μιας από τις φωτογραφίες στο σύνολο εκπαίδευσης. Μέσω αυτής της ανάλυσης αυτών των βαρών μπορεί να υλοποιηθεί η ανίχνευση προσώπου.



Εικόνα 2-10 Πιθανά πρόσωπα με eigenfaces

3. C++

3.1 Εισαγωγή

Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού H/Y. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών από γλώσσες υψηλού και χαμηλού επιπέδου. Είναι μια μεταγλωττιζόμενη γλώσσα, με τύπους. Υποστηρίζει δομημένο, αντικειμενοστρεφή και γενικό προγραμματισμό. Αναπτύχθηκε από τον Bjarne Stroustrup ξεκινώντας το 1979 στο Bell Labs, η C++ αρχικά ονομάστηκε C με κλάσεις, προσθέτοντας αντικειμενοστρεφή χαρακτηριστικά, όπως οι κλάσεις, και άλλες βελτιώσεις στη γλώσσα προγραμματισμού C. Η γλώσσα μετονομάστηκε σε C++ το 1983.

Η C++ είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού και εφαρμόζεται σε μια ευρεία ποικιλία από πλατφόρμες υλικού και λειτουργικών συστημάτων. Καθώς είναι ένας αποτελεσματικός compiler σε εγγενή κώδικα, οι τομείς εφαρμογής του περιλαμβάνουν συστήματα υλικού, συστήματα λογισμικού, οδηγούς συσκευών, ενσωματωμένο υλικό, server υψηλής απόδοσης και client εφαρμογές καθώς και λογισμικό ψυχαγωγίας όπως βιντεοπαιχνίδια. Αρκετές ομάδες παρέχουν ελεύθερο λογισμικό και ιδιοκτησιακό C++ compiler λογισμικό, συμπεριλαμβανομένου το GNU Project, LLVM, Microsoft και Intel. Η C++ έχει επηρεάσει σε μεγάλο βαθμό πολλές άλλες δημοφιλείς γλώσσες προγραμματισμού, κυρίως την C# και την Java.

Η γλώσσα ξεκίνησε ως βελτιώσεις της C, αρχικά προσθέτοντας κλάσεις, στη συνέχεια εικονικές συναρτήσεις, υπερφόρτωση φορέων, πολλαπλή κληρονομικότητα, πρότυπα και εξαίρεση χειρισμού, μεταξύ άλλων χαρακτηριστικών. Μετά από χρόνια ανάπτυξης, η γλώσσα προγραμματισμού C++ επικυρώθηκε ως πρότυπο το 1998 ISO/IEC 14882:1998. Το πρότυπο αυτό τροποποιήθηκε το 2003, ISO/IEC 14882:2003. Το τρέχον πρότυπο που επεκτάθηκε με ένα χαρακτηριστικά επικυρώθηκε και δημοσιεύτηκε από τον ISO το 2011 ως ISO/IEC 14882:2011.

3.2 Φιλοσοφία

Στο βιβλίο του "The Design and Evolution of C++ (1994), ο Bjarne Stroustrup περιγράφει κάποιους κανόνες που χρησιμοποιεί για το σχεδιασμό της C++:

- η C++ είναι σχεδιασμένη ως μια γενικής χρήσης γλώσσα με στατικούς τύπους, που είναι όσο αποτελεσματική και φορητή, όσο η C
- η C++ είναι σχεδιασμένη να υποστηρίζει άμεσα και σφαιρικά πολλά είδη προγραμματισμού (δομημένος προγραμματισμός, αντικειμενοστραφής προγραμματισμός, γενικός προγραμματισμός)
- η C++ είναι σχεδιασμένη να δίνει επιλογές στον προγραμματιστή, ακόμα κι αν του επιτρέπει να επιλέξει λανθασμένα
- η C++ είναι σχεδιασμένη να είναι όσο το δυνατόν συμβατή με τη C, ώστε να διευκολύνει τη μετάβαση από τη C
- η C++ αποφεύγει χαρακτηριστικά που αναφέρονται σε συγκεκριμένες πλατφόρμες ή δεν είναι γενικής χρήσης
- η C++ δεν έχει κόστος για χαρακτηριστικά της γλώσσας που δεν χρησιμοποιούνται
- η C++ είναι σχεδιασμένη να λειτουργεί χωρίς κάποιο εξελεγμένο προγραμματιστικό περιβάλλον

4. OpenCV

4.1 Εισαγωγή

Η OpenCV είναι μία βιβλιοθήκη ανοιχτού κώδικα που είναι ελεύθερα διαθέσιμη. Είναι γραμμένη σε C και C++ και τρέχει σε Linux, Windows και Mac OS X. Υπάρχει ενεργή ανάπτυξη για διεπαφές με Python, Ruby, Matlab και άλλες γλώσσες.

Η OpenCV σχεδιάστηκε για υπολογιστική αποτελεσματικότητα και με ιδιαίτερη έμφαση στις εφαρμογές πραγματικού χρόνου. Είναι γραμμένη σε βελτιστοποιημένη C και μπορεί να επωφεληθεί από πολυπύρηνους επεξεργαστές.

Ένας από τους στόχους της OpenCV είναι να παρέχει στο χρήστη απλές στη χρήση υποδομές στην τεχνητή όραση που βοηθάει τους ανθρώπους να χτίσουν γρήγορα, αρκετά εξελιγμένες εφαρμογές όρασης. Η βιβλιοθήκη περιέχει πάνω από 500 συναρτήσεις που εκτείνονται σε πολλούς τομείς της όρασης, συμπεριλαμβανομένων της εργοστασιακής επιθεώρησης προϊόντων, ιατρικής απεικόνισης, ασφάλειας, διεπαφή με χρήστη, βαθμονόμηση κάμερας, στερεοφωνική όραση και ρομποτική. Επειδή η τεχνητή όραση και η μηχανική μάθηση συχνά πηγαίνουν χέρι-χέρι, η OpenCV επίσης περιλαμβάνει μια πλήρη, γενικού σκοπού Βιβλιοθήκη Μηχανικής Μάθησης (MLL). Αυτή η υπό-βιβλιοθήκη επικεντρώνεται στη στατιστική αναγνώριση προτύπων και ομαδοποίηση. Η MLL είναι ιδιαίτερα χρήσιμη για οπτικές διεργασίες που είναι ο πυρήνας της αποστολής της OpenCV, αλλά μπορούν να χρησιμοποιηθούν για οποιοδήποτε πρόβλημα μηχανική εκπαίδευσης.

4.2 Ποιοι χρησιμοποιούν την OpenCV

Οι περισσότεροι επιστήμονες της πληροφορικής και προγραμματιστές γνωρίζουν κάποια πτυχή του ρόλου που παίζει η τεχνητή όραση. Αλλά λίγοι άνθρωποι γνωρίζουν όλους τους τρόπους με τους οποίους χρησιμοποιείται η τεχνητή όραση. Για παράδειγμα, οι περισσότεροι άνθρωποι κάπως γνωρίζουν για τη χρήση του σε επιτήρηση, και πολλοί επίσης γνωρίζουν ότι χρησιμοποιείται όλο και περισσότερο για εικόνες και βίντεο στο

διαδίκτυο. Λίγοι έχουν δει τη χρήση της τεχνητής όρασης σε παιχνίδια. Επίσης λίγοι άνθρωποι συνειδητοποιούν ότι οι περισσότερες εναέριες και street-map εικόνες (όπως το Google's Street της Google) κάνουν χρήση της προσαρμογής κάμερας και τεχνικές συρραφής εικόνων. Μερικοί γνωρίζουν εξειδικευμένες εφαρμογές για παρακολούθηση ασφαλείας, για επανδρωμένα ιπτάμενα οχήματα ή για αναλύσεις βιοϊατρικής.

Η άδεια ανοικτού κώδικα της OpenCV έχει δομηθεί έτσι ώστε μπορεί να φτιαχτεί ένα εμπορικό προϊόν με το σύνολο ή μέρος της OpenCV. Εν μέρει λόγω αυτών των φιλελεύθερων όρων αδειοδότησης, υπάρχει μια μεγάλη κοινότητα χρηστών που περιλαμβάνει ανθρώπους από μεγάλες εταιρείες όπως οι IBM, Microsoft, Intel, SONY, Siemens, και Google, και ερευνητικά κέντρα όπως τα Stanford, MIT, CMU, Cambridge, INRIA. Η OpenCV είναι δημοφιλής σε όλο τον κόσμο, με μεγάλες κοινότητες χρηστών στην Κίνα, την Ιαπωνία, την Ρωσία, και την Ευρώπη.

Από την alpha έκδοση του Ιανουαρίου του 1999, η OpenCV έχει χρησιμοποιηθεί σε πολλές εφαρμογές, προϊόντα, και έρευνες. Οι εφαρμογές περιλαμβάνουν συρραφή εικόνων από δορυφόρους και web χάρτες, ευθυγράμμιση σαρωμένων εικόνων, μείωση του θορύβου σε ιατρικές εικόνες, ανάλυση αντικειμένων, συστήματα ασφάλειας και ανίχνευσης διείσδυσης, αυτόματα συστήματα παρακολούθησης, συστήματα ελέγχου παραγωγής, βαθμονόμηση κάμερας, στρατιωτικές εφαρμογές, καθώς και σε μη επανδρωμένα εναέρια, επίγεια και υποβρύχια οχήματα. Έχει ακόμη χρησιμοποιηθεί στην αναγνώριση ήχων και μουσικής, όπου οι τεχνικές αναγνώρισης εικόνων εφαρμόζονται στο φασματογράφημα του ήχου. Η OpenCV αποτελούσε ένα βασικό μέρος του συστήματος όρασης στο ρομπότ του Stanford «Stanley», το οποίο κέρδισε το βραβείο των δυο εκατομμυρίων δολαρίων στον αγώνα DARPA Grand Challenge Desert Robot Race.

4.3 Προέλευση

Η OpenCV αναπτύχθηκε από ερευνητική πρωτοβουλία της Intel Research για την προώθηση των εφαρμογών CPU υψηλής έντασης. Προς το σκοπό αυτό, η Intel λανσάρισε πολλά project, συμπεριλαμβανομένων την ανίχνευση ακτινών σε πραγματικό χρόνο και την τρισδιάστατη εμφάνιση τοίχων. Ένας από τους συντάκτες που δούλευε

στην Intel εκείνη την εποχή επισκέπτονταν πανεπιστήμια και παρατήρησε ότι μερικές ομάδες κορυφαίων πανεπιστημίων, όπως το MIT Media Lab, είχαν αναπτύξει και εσωτερικά ανοιχτές υποδομές στην τεχνητή όραση- κώδικας που μεταφέρθηκε από φοιτητή σε φοιτητή και που έδωσε σε κάθε νέο φοιτητή ένα πολύτιμο προβάδισμα στο να αναπτύξει τη δική του εφαρμογή όρασης.

Ωστόσο, η OpenCV σχεδιάστηκε ως ένας τρόπος για να κάνει τις υποδομές της τεχνητής όρασης διαθέσιμες παγκοσμίως. Με τη βοήθεια της Intel Performance Library Team, η OpenCV ξεκίνησε με έναν πυρήνα εφαρμοσμένου κώδικα και αλγοριθμικών προδιαγραφών που στέλνονταν στη ρώσικη ομάδα βιβλιοθήκης της Intel. Αυτό είναι το «πού» της OpenCV: ξεκίνησε στα ερευνητικά εργαστήρια της Intel σε συνεργασία με την ομάδα Software Performance Libraries για την υλοποίηση και τη βελτιστοποίηση της τεχνογνωσίας στη Ρωσία.

Ο αρχηγός των μελών της ρώσικης ομάδας ήταν ο Vadim Pisarevsky, ο οποίος διηύθυνε, κωδικοποίησε και βελτιστοποίησε πολλά πράγματα της OpenCV και που ακόμη είναι στο κέντρο ενός μεγάλου μέρους της προσπάθειας της OpenCV. Μαζί του, ο Victor Eruhimov βοήθησε στην ανάπτυξη της πρόωρης υποδομής και ο Valery Kuriakin διαχειρίστηκε το ρώσικο εργαστήριο και υποστήριξε σε μεγάλο βαθμό την προσπάθεια. Υπήρχαν αρκετοί στόχοι για την OpenCV από την αρχή:

- Προηγμένη έρευνα όρασης παρέχοντας όχι μόνο ανοιχτό αλλά και βελτιστοποιημένο κώδικα για τις βασικές υποδομές όρασης.
- Διάδοση των γνώσεων όρασης παρέχοντας μια κοινή υποδομή όπου οι προγραμματιστές μπορούν να αξιοποιήσουν έτσι ώστε ο κώδικας να είναι πιο εύκολα αναγνώσιμος και μεταβιβάσιμος.
- Προηγμένες εμπορικές εφαρμογές με βάση την όραση κάνοντας τις φορητές, βελτιστοποιημένη απόδοση κώδικα με διάθεση δωρεάν-με άδεια όπου οι εμπορικές εφαρμογές δεν ανοίγουν μόνες τους.

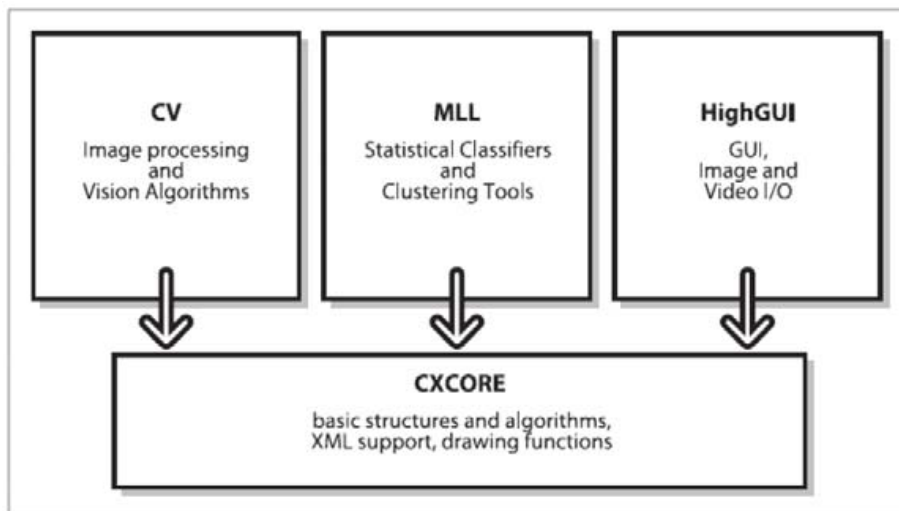
Αυτοί οι στόχοι αποτελούν το «γιατί» της OpenCV. Ενεργοποιώντας της εφαρμογές τεχνητής όρασης, αυξάνεται η ανάγκη για γρήγορους επεξεργαστές. Η ανάγκη αναβάθμισης σε γρηγορότερους επεξεργαστές θα δημιουργήσουν περισσότερα έσοδα στην Intel αντί την πώληση περισσότερου software. Ίσως γι' αυτό αυτός ο ανοιχτός και ελεύθερος κώδικας προέκυψε από έναν προμηθευτή hardware από μια εταιρεία software.

Κατά μία έννοια, υπάρχει περισσότερος χώρος για καινοτομίες για software μέσα σε μια εταιρεία hardware.

Σε κάθε προσπάθεια ανοιχτού κώδικα, είναι σημαντικό να επιτευχθεί μια σημαντική μάζα η οποία θα κάνει το project αυτοτροφοδοτούμενο. Τα downloads έχουν γίνει πάνω από 7 εκατομμύρια και ο αριθμός αυτός μεγαλώνει όλο και περισσότερο.

4.4 Δομή και περιεχόμενο

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).



Εικόνα 4-4 Βιβλιοθήκες της OpenCV. Το Σχήμα δεν περιλαμβάνει την CvAux, η οποία περιέχει τόσο περιοχές που δεν χρησιμοποιούνται (ενσωματωμένο HMM αναγνώρισης προσώπου) αλλά και πειραματικούς αλγόριθμους (κατάτμηση φόντου, πρόσοψης).

4.5 Εφαρμογές

Οι τομείς εφαρμογής της OpenCV περιλαμβάνουν:

- Σύστημα αναγνώρισης προσώπου
- Αλληλεπίδραση ανθρώπου-υπολογιστή
- Ρομποτική κίνηση
- Αναγνώριση αντικειμένων
- Κατασκευή από κίνηση
- Παρακολούθηση κίνησης
- Εργαλεία δισδιάστατων και τρισδιάστατων χαρακτηριστικών
- Στερεοσκοπική όραση: αντίληψη του βάθους από δύο κάμερες

Για την υποστήριξη μερικών εφαρμογών, η OpenCV περιλαμβάνει μια στατική βιβλιοθήκη μηχανικής μάθησης που περιλαμβάνει:

- Ενίσχυση
- Δέντρα απόφασης
- Αλγόριθμος μέγιστης πρόβλεψης
- Αλγόριθμος k-κοντινότερων γειτόνων
- Ταξινομητής Bayes
- Τεχνητά νευρωνικά δίκτυα
- Random Forest
- Μηχανές Διανυσμάτων Στήριξης

5. Ανίχνευση προσώπου με Haar χαρακτηριστικά

5.1 Εισαγωγή

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 [7] και αναπτύχθηκε από τον Rainer Lienhart [11]. 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.

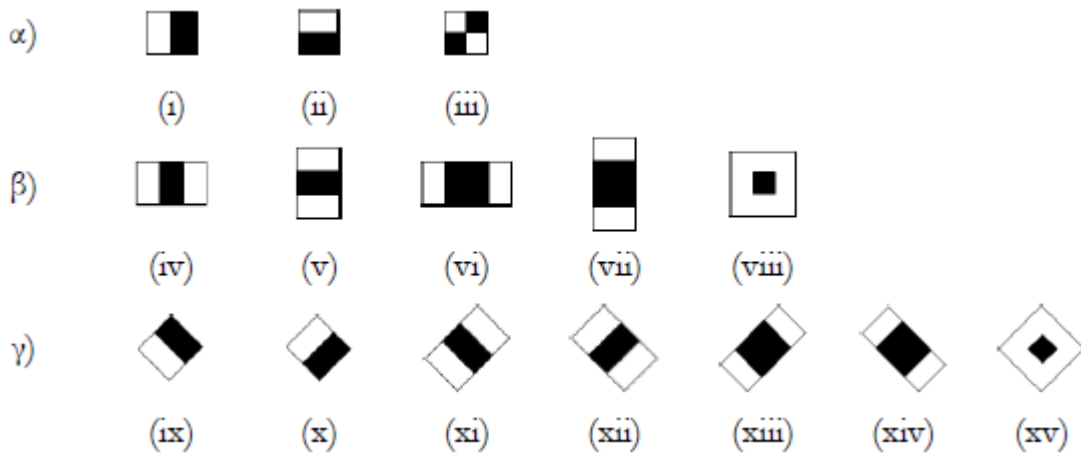
Η τρίτη σημαντική συμβολή είναι μια μέθοδος για τον επιτυχή συνδυασμό πιο πολύπλοκων ταξινομητών σε μια δομή καταρράκτη η οποία αυξάνει δραματικά την ταχύτητα του ανιχνευτή με την εστίαση της προσοχής σε υποσχόμενες περιοχές της εικόνας. Η έννοια πίσω από την εστίαση της προσοχής είναι ότι είναι συχνά 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.

5.2 Χαρακτηριστικά

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 κλασικά 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.

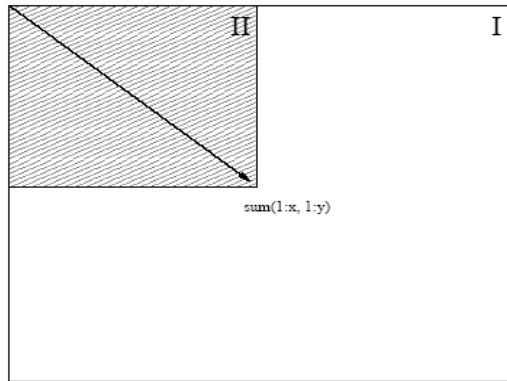


Εικόνα 5-1 Χαρακτηριστικά τύπου Haar. α) Κλασικά Haar χαρακτηριστικά, β) Χαρακτηριστικά για γραμμές και ράβδους, γ) Χαρακτηριστικά περιστραμμένα κατά 45°

5.3 Ακέραια εικόνα (integral image)

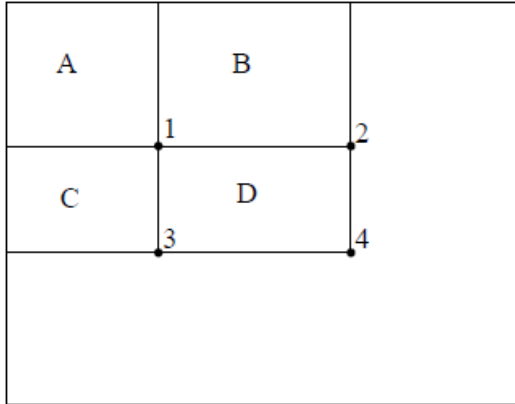
Για να προσδιοριστεί αποτελεσματικά η παρουσία ή η απουσία εκατοντάδων Haar χαρακτηριστικών σε κάθε περιοχή της εικόνας και σε διάφορες κλίμακες, οι Viola και Jones χρησιμοποίησαν μια τεχνική που ονομάζεται Ακέραια Εικόνα (Integral Image). Γενικά “integrating” σημαίνει προσθήκη μικρών μονάδων μαζί. Σ’ αυτή την περίπτωση, οι μικρές μονάδες είναι οι τιμές των εικονοστοιχείων. Η ακέραια τιμή για κάθε εικονοστοιχείο είναι το άθροισμα όλων των εικονοστοιχείων πάνω και αριστερά του.

Ξεκινώντας από πάνω αριστερά και πηγαίνοντας προς δεξιά και κάτω, όλη η εικόνα μπορεί να ενσωματωθεί με λίγες λειτουργίες ανά εικονοστοιχείο.



*Εικόνα 5-2 Η τιμή της ακέραιας εικόνας στο σημείο x,y είναι το άθροισμα των *pixel* πάνω και αριστερά. Όπου *I* η αρχική εικόνα και *II* η ακέραια εικόνα*

Χρησιμοποιώντας την ακέραια εικόνα μπορεί να υπολογιστεί κάθε ορθογώνιο άθροισμα τεσσάρων διατάξεων αναφορών (εικόνα 5-3). Είναι σαφές ότι η διαφορά μεταξύ του αθροίσματος των δύο ορθογωνίων μπορεί να υπολογιστεί σε οχτώ αναφορές. Αφού τα χαρακτηριστικά δύο ορθογωνίων που ορίζονται πάνω περιλαμβάνουν γειτονικά αθροίσματα ορθογωνίων μπορούν να υπολογιστούν σε διάταξη έξι αναφορών, οχτώ στην περίπτωση του χαρακτηριστικού τριών ορθογωνίων και εννιά στα χαρακτηριστικά τεσσάρων ορθογωνίων.



Εικόνα 5-3 Το άθροισμα των pixel μέσα στο ορθογώνιο D μπορεί να υπολογιστεί σε διάταξη τεσσάρων αναφορών. Η τιμή της ακέραιας εικόνας στην περιοχή 1 είναι το άθροισμα των pixel στο ορθογώνιο A . Η τιμή στην περιοχή 2 είναι $A+B$, στην περιοχή 3 είναι $A+C$ και στην περιοχή 4 είναι $A+B+C+D$. Το άθροισμα μέσα στο D μπορεί να υπολογιστεί $4+1-(2+3)$.

5.4 Αλγόριθμος Εκπαίδευσης AdaBoost

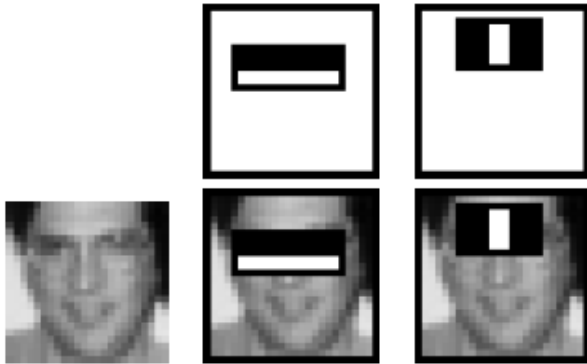
Στην μέθοδο ανίχνευσης που εξετάζουμε, χρησιμοποιήθηκε ο αλγόριθμος AdaBoost τόσο για την επιλογή των χαρακτηριστικών που θα χρησιμοποιηθούν, όσο και για την εκπαίδευση του ταξινομητή. Ο αλγόριθμος εκμάθησης AdaBoost ανήκει στην κατηγορία των αλγορίθμων ενδυνάμωσης (boosting) και χρησιμοποιείται για 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 ένας μικρός αριθμός 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).



Εικόνα 5-4 Τα πρώτα και δεύτερα χαρακτηριστικά διαλέχθηκαν από τον AdaBoost. Τα δύο χαρακτηριστικά φαίνονται στην πάνω σειρά και μετά επικαλύπτονται σε ένα τυπικό πρόσωπο εκμάθησης στην κάτω σειρά. Το πρώτο χαρακτηριστικό μετράει τη διαφορά της έντασης ανάμεσα στην περιοχή των ματιών και μια περιοχή κατά μήκος των άνω μαγουλών. Το χαρακτηριστικό εκμεταλλεύεται την παρατήρηση ότι η περιοχή των ματιών είναι συνήθως πιο σκούρα από τα μάγουλα. Το δεύτερο χαρακτηριστικό συγκρίνει την ένταση στην περιοχή των ματιών με αυτή κατά μήκος της γέφυρας της μύτης.

5.5 Καταρράκτης ταξινομητών

Η βασική αντίληψη είναι ότι μπορούν να κατασκευαστούν μικρότεροι, και ως εκ τούτου αποτελεσματικοί, ενισχυμένοι ταξινομητές οι οποίοι απορρίπτουν πολλά από τα αρνητικά υπό-παράθυρα ενώ ανιχνεύουν σχεδόν όλα τα θετικά παραδείγματα. Απλούστεροι ταξινομητές χρησιμοποιούνται για να απορρίψουν την πλειοψηφία των υπό-παραθύρων πριν καλεστούν πιο πολύπλοκοι ταξινομητές για να επιτύχουν χαμηλά ποσοστά λάθους.

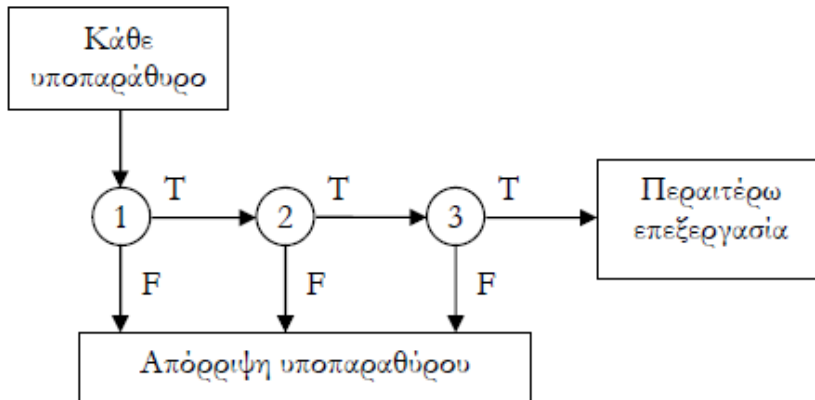
Τα στάδια του καταρράκτη κατασκευάστηκαν από εκπαιδευμένους ταξινομητές με τη χρήση AdaBoost. Ξεκινώντας με έναν δυνατό ταξινομητή δύο χαρακτηριστικών, μπορεί

να ληφθεί ένα αποτελεσματικό φίλτρο προσώπου με την προσαρμογή ενός κατώφλιου δυνατού ταξινομητή για να ελαχιστοποιήσει τα αρνητικά αποτελέσματα. Το αρχικό κατώφλι του Adaboost σχεδιάστηκε για να δώσει ένα χαμηλό ποσοστό σφάλματος στα δεδομένα εκπαίδευσης. Ένα χαμηλότερο κατώφλι αποδίδει υψηλότερα ποσοστά ανίχνευσης και μεγαλύτερα ποσοστά λάθος ανίχνευσης.

Η απόδοση της ανίχνευσης του ταξινομητή δύο χαρακτηριστικών είναι πολύ πιο αποδεκτή ως ένα σύστημα ανίχνευσης αντικειμένου. Παρ' όλα αυτά ο ταξινομητής μπορεί να μειώσει σημαντικά τον αριθμό των υπό-παραθύρων που χρειάζονται περαιτέρω επεξεργασία με πολύ λίγες λειτουργίες:

1. Αξιολόγηση των ορθογώνιων χαρακτηριστικών (απαιτεί 6 με 9 πίνακες αναφοράς ανά χαρακτηριστικό).
2. Υπολογισμός του αδύναμου ταξινομητή για κάθε χαρακτηριστικό (απαιτεί ένα κατώφλι ανά χαρακτηριστικό).
3. Συνδυασμός των αδύναμων ταξινομητών (απαιτεί έναν πολλαπλασιασμό ανά χαρακτηριστικό, μια πρόσθεση και τέλος ένα κατώφλι).

Η γενική μορφή της μεθόδου ανίχνευσης είναι από ένα εκφυλισμένο δέντρο απόφασης, που ονομάζεται «καταρράκτης» (cascade). (Εικόνα 5-5). Ένα θετικό αποτέλεσμα από τον πρώτο ταξινομητή ενεργοποιεί την αξιολόγηση του δεύτερου ταξινομητή ο οποίος έχει επίσης προσαρμοστεί ώστε να επιτυγχάνει πολύ μεγάλα ποσοστά ανίχνευσης. Ένα θετικό αποτέλεσμα από τον δεύτερο ταξινομητή ενεργοποιεί τον τρίτο ταξινομητή κτλ. Ένα αρνητικό αποτέλεσμα σε οποιοδήποτε σημείο οδηγεί στην άμεση απόρριψη του υπό-παραθύρου.



Εικόνα 5-5 Σχηματική αναπαράσταση ενός καταρράκτη ανίχνευσης. Μια σειρά από ταξινομητές εφαρμόζονται σε κάθε υπό-παράθυρο, Ο αρχικός ταξινομητής εξαλείφει ένα μεγάλο αριθμό από αρνητικά παραδείγματα με πολύ μικρή επεξεργασία. Μεταγενέστερα στρώματα εξαλείφουν επιπλέον αρνητικά αλλά απαιτούν πρόσθετο υπολογισμό. Μετά από διάφορα στάδια επεξεργασίας ο αριθμός των υπό-παραθύρων έχουν μειωθεί δραστικά. Η περαιτέρω επεξεργασία μπορεί να λάβει οποιαδήποτε μορφή όπως επιπλέον στάδια ενός καταρράκτη (όπως στο συγκεκριμένο σύστημα) ή ένα εναλλακτικό σύστημα ανίχνευσης.

Η δομή του καταρράκτη αντανακλά το γεγονός ότι σε κάθε μεμονωμένη εικόνα η συντριπτική πλειοψηφία των υπό-παραθύρων είναι αρνητική. Ως εκ τούτου, ο καταρράκτης επιχειρεί να απορρίψει όσο το δυνατό περισσότερα αρνητικά στο πιο πρώιμο στάδιο. Ενώ ένα θετικό παράδειγμα θα προκαλέσει την αξιολόγηση κάθε ταξινομητή στον καταρράκτη, που αυτό είναι εξαιρετικά σπάνιο συμβάν.

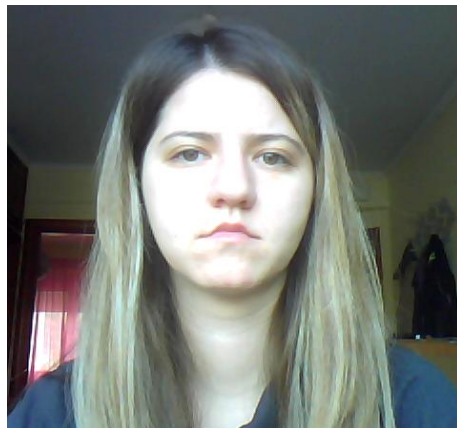
6. Περιγραφή εφαρμογής αναγνώρισης βλεφαρίσματος

Ο σκοπός της παρούσας εργασίας είναι η ανάπτυξη εφαρμογής για την ανίχνευση βλεφαρίσματος των ματιών. Υπάρχουν αρκετές μέθοδοι για την ανίχνευση κίνησης (γιατί στην ουσία το βλεφάρισμα είναι κάποιου είδους κίνησης). Οι μέθοδοι αυτοί χωρίζονται στις μεθόδους πραγματικού χρόνου, που εντοπίζουν την κίνηση μέσα από μια κάμερα την ώρα που αυτή κάνει την λήψη (κάμερες συστημάτων ασφαλείας) και σε αυτές που επεξεργάζονται αποθηκευμένα δεδομένα ετεροχρονισμένα, με σκοπό να βγάλουν διάφορα συμπεράσματα (κάμερες μετεωρολογικών δορυφόρων). Εμάς μας ενδιαφέρουν οι μέθοδοι πραγματικού χρόνου μιας και σκοπός μας είναι ο εντοπισμός του βλεφαρίσματος των ματιών σε μια λήψη από κάμερα, την ώρα που η κάμερα αυτή κάνει την λήψη, προκειμένου ο υπολογιστής να προβεί σε κάποιες ενέργειες όταν αντιλαμβάνεται ότι υπάρχει κίνηση. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++, με τη χρήση της βιβλιοθήκης OpenCV στο πρόγραμμα της Microsoft Visual Studio 2010 Express. Εφόσον θέλουμε να επικεντρωθούμε στα μάτια, το πρώτο βήμα είναι η ανίχνευση του προσώπου και κατά συνέπεια των ματιών. Η ανίχνευση του προσώπου γίνεται με έναν ανιχνευτή που υποστηρίζει η OpenCV, τον Cascade Classifier. Αφού γίνει η ανίχνευση, με βάση το πρόσωπο, γίνεται η ανίχνευση των ματιών. Σε ένα βίντεο ο εντοπισμός κίνησης (στην περίπτωση μας το βλεφάρισμα) γίνεται μεταξύ δύο συνεχόμενων εικόνων για όλες τις εικόνες του βίντεο. Αφού βρεθούν τα μάτια σε δύο διαδοχικά πλαίσια, επόμενο βήμα είναι ο υπολογισμός της διαφοράς μεταξύ των εικόνων των ματιών. Δηλαδή ο υπολογισμός ενός πίνακα ιδίων διαστάσεων με τις δύο εικόνες, όπου κάθε στοιχείο του είναι η διαφορά των αντίστοιχων στοιχείων των δύο εικόνων. Η διαφορά αυτή ουσιαστικά είναι η κίνηση μιας και ως κίνηση σε αυτήν την μέθοδο θεωρείται οτιδήποτε αλλάζει στην εικόνα. Η τεχνική που χρησιμοποιήθηκε γι' αυτό το βήμα είναι το άθροισμα απόλυτων διαφορών. Αν το αποτέλεσμα είναι 0, τότε δεν υπάρχει διαφορά ανάμεσα στα διαδοχικά πλαίσια. Σε αντίθετη περίπτωση υπάρχει βλεφάρισμα. Το τρίτο βήμα είναι η επιλογή ενός κατωφλίου και η κατωφλίωση με αυτό το κατώφλι του πίνακα της διαφοράς. Τέλος, εμφανίζεται θετικό μήνυμα σε περίπτωση βλεφαρίσματος.

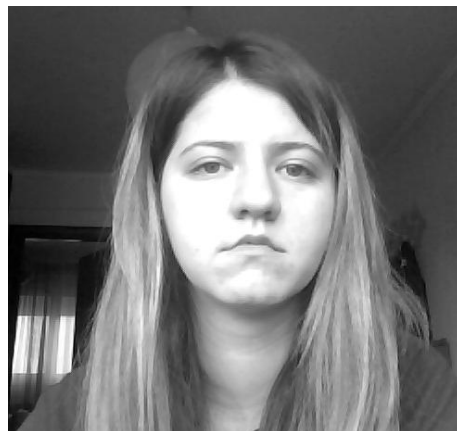
Η διαδικασία που ακολουθήθηκε χωρίζεται στα παρακάτω βήματα:

1. Λήψη πλαισίου από κάμερα
2. Εντοπισμός προσώπου
3. Εντοπισμός ματιών
4. Λήψη επόμενου πλαισίου από κάμερα
5. Εκ νέου εντοπισμός προσώπου
6. Εντοπισμός ματιών
7. Σύγκριση της περιοχής των ματιών από τα δύο διαδοχικά πλαίσια
8. Επιλογή κατωφλίου για αφαίρεση θορύβου
9. Εμφάνιση μηνύματος σε περίπτωση βλεφαρίσματος

Παρακάτω αναλύεται με εικόνες η διαδικασία της ανίχνευσης



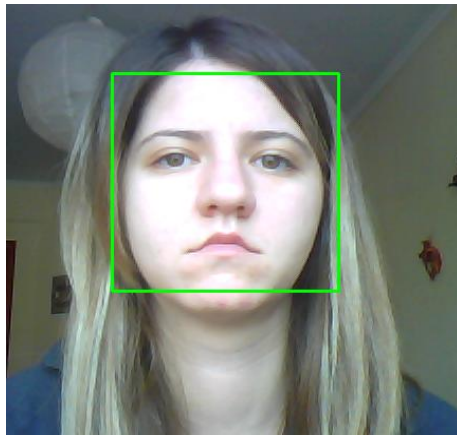
Εικόνα 6-1 Λήψη πλαισίου από κάμερα



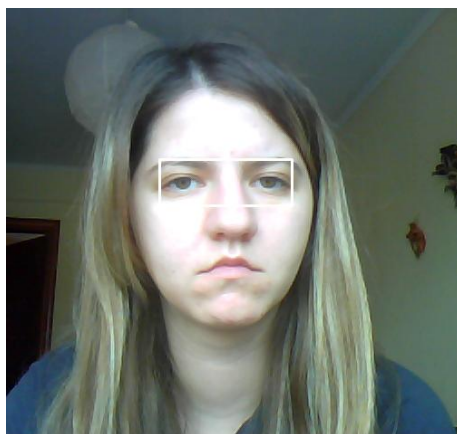
Εικόνα 6-2 Μετατροπή του πλαισίου σε κλίμακα του γκρι



Εικόνα 6-3 Εξισορρόπηση ιστογράμματος



Εικόνα 6-4 Εντοπισμός προσώπου



Εικόνα 6-5 Εντοπισμός ματιών



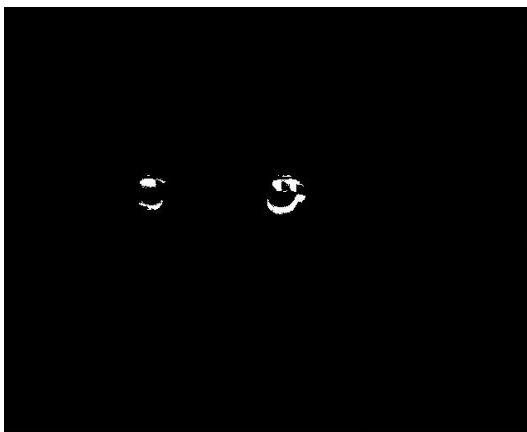
Εικόνα 6-6 Εξαγωγή της περιοχής των ματιών

Στη συνέχεια γίνεται λήψη ακόμη ενός πλαισίου και ακολουθείται η ίδια διαδικασία για το δεύτερο πλαίσιο.

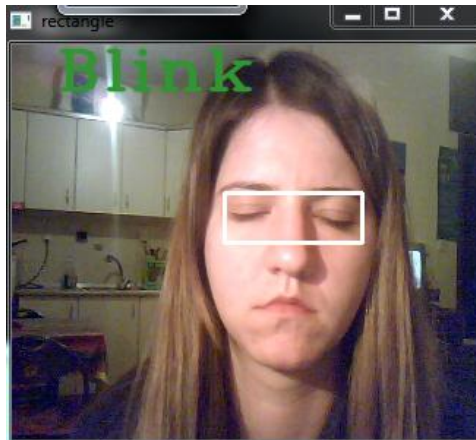
Για να γίνει η σύγκριση του ROI των ματιών γίνεται χρήση της τεχνικής του αθροίσματος απόλυτων διαφορών.



Εικόνα 6-7 Το αποτέλεσμα από το άθροισμα απόλυτων διαφορών



Εικόνα 6-8 Το αποτέλεσμα μετά την κατωφλίωση όταν υπάρχει ανοιγόκλεισμα ματιών.



Εικόνα 6-9 Το τελικό αποτέλεσμα όταν υπάρχει ανοιγόκλεισμα ματιών

7. Ανάλυση εφαρμογής

7.1 Άνοιγμα κάμερας και λήψη πλαισίου

Για το άνοιγμα της κάμερας χρησιμοποιείται η εντολή `VideoCapture cap(0)`.

Στη συνέχεια για τη λήψη πλαισίου χρησιμοποιείται η εντολή `cap >> cam_frame` όπου `cam_frame` είναι ένας πίνακας `Mat` που αποθηκεύονται οι τιμές των εικονοστοιχείων του πλαισίου. Αυτό μπαίνει μέσα σε ένα βρόχο `while` ώστε να παίρνει συνεχόμενα πλαίσια μέχρι να αποφασίσουμε εμείς πότε θα σταματήσει. Μετά γίνεται αντιγραφή του πλαισίου ώστε να μπορεί να ληφθεί το επόμενο πλαίσιο.

7.2 Διαδικασία εύρεσης βλεφαρίσματος

Για τη δημιουργία του ορθογωνίου των ματιών, δημιουργούμε ένα διάνυσμα από ορθογώνια όπου θα αποθηκεύονται τα ορθογώνια με τις περιοχές ματιών που ανιχνεύονται. Στη συνέχεια καλείται η συνάρτηση για τον εντοπισμό του προσώπου και ματιών (εξηγείται παρακάτω). Αφού εντοπιστούν τα μάτια, σχεδιάζετε ένα ορθογώνιο γύρω από την περιοχή αυτή και εξάγεται ο πίνακας `mat` με τις τιμές της εικόνας.

Εφόσον τελειώσει η διαδικασία, γίνεται λήψη του επόμενου πλαισίου και ακολουθείται ακριβώς η ίδια διαδικασία όπως με το προηγούμενο πλαίσιο.

7.3 Άθροισμα απόλυτων διαφορών

Στην επεξεργασία εικόνας, η μέθοδος του Άθροίσματος των Απόλυτων Διαφορών (ΑΑΔ) χρησιμοποιείται ευρέως για την ανίχνευση κίνησης και συμπίεσης βίντεο. Η μέθοδος ΑΑΔ ουσιαστικά εκτελεί μια σύγκριση `pixel-pixel` μεταξύ δύο πλαισίων με τον υπολογισμό των απόλυτων διαφορών μεταξύ των εικονοστοιχείων. Αυτή η διαδικασία επαναλαμβάνεται για πολλαπλά διαδοχικά πλαίσια και στη συνέχεια οι απόλυτες διαφορές προσθέτονται για να δημιουργήσουν τον τελικό διδιάστατο πίνακα ΑΑΔ. Σε μια τυπική εφαρμογή ανίχνευσης κίνησης, συγκρίνεται ένας αριθμός από διαδοχικά πλαίσια με ένα πλαίσιο παρασκήνιου με το κάθε πλαίσιο να διαιρείται σε έναν αριθμό

από μικρότερα μπλοκ, και η υπέρβαση ενός κατωφλίου σε μια ορισμένη θέση του πίνακα ΑΑΔ δείχνει κάποια κίνηση στο αντίστοιχο μπλοκ στο πλαίσιο παρασκηνίου.

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_{i,j} - T_{i,j}|$$

Στο πρόγραμμα που αναπτύχθηκε, χρησιμοποιήθηκε η τεχνική του ΑΑΔ για την ανίχνευση βλεφαρίσματος. Όταν γίνεται η εξαγωγή του ROI των ματιών, το αποτέλεσμα αποθηκεύεται σε ένα mat, δηλαδή έναν πίνακα που περιέχει τις τιμές των εικονοστοιχείων μιας εικόνας. Εφόσον έχει γίνει η εξαγωγή του ROI των ματιών από τα δύο διαδοχικά πλαίσια, εφαρμόζεται το άθροισμα των απόλυτων διαφορών. Αν το αποτέλεσμα του αθροίσματος είναι 0 τότε οι δύο εικόνες είναι ίδιες και προφανώς δεν υπάρχει βλεφάρισμα. Αν όμως το αποτέλεσμα προκύψει πάνω από 0, σημαίνει πως υπάρχει κάποια διαφορά στις εικόνες που εκλαμβάνεται ως βλεφάρισμα.

7.4 Κατωφλίωση

Το πρόβλημα στην ανίχνευση κίνησης είναι ότι ακόμη κι αν δεν υπάρχει καθόλου κίνηση, τα διαδοχικά πλαίσια δεν θα είναι ποτέ ακριβώς ίδια. Αυτό οφείλεται σε μια οπτική παραμόρφωση της εικόνας, που αποκαλείται «θόρυβος». Συνήθως έχει τη μορφή κόκκων και γίνεται εντονότερο όταν υπάρχει κακός φωτισμός. Για να εξαλειφθεί όσο είναι δυνατό ο θόρυβος στο ROI των ματιών χρησιμοποιούμε την κατωφλίωση. Μ' αυτό τον τρόπο η ευαισθησία της ανίχνευσης μειώνεται και ανιχνεύονται μόνο μεγάλες διαφορές ανάμεσα στα πλαίσια.

Η συνάρτηση threshold εφαρμόζει ένα κατώφλι σε κάθε στοιχείο της εικόνας.

```
double threshold ( InputArray src,  
                  OutputArray dst,  
                  double thresh,  
                  double maxval,  
                  int type )
```


src

Η εικόνα εισόδου (σε κλίμακα του γκρι).

dst

Η εικόνα εξόδου που έχει ίδιο μέγεθος και τύπο με την εικόνα εισόδου.

thresh

Η τιμή του ορίου (κατώφλι). Όσο μεγαλύτερη είναι η τιμή του ορίου, τόσο μικρότερη ευαισθησία υπάρχει ανάμεσα στα πλαίσια. Η τιμή που επιλέχτηκε είναι το 80.

maxval

Η μέγιστη τιμή που μπορεί να χρησιμοποιηθεί στην τελική εικόνα. Έχει οριστεί το 255 δηλαδή το χρώμα μαύρο.

type

Ο τύπος κατωφλίωσης. Χρησιμοποιήθηκε CV_THRESH_BINARY όπου μετατρέπει την εικόνα από αποχρώσεις του γκρι σε δυαδική. Αν η τιμή του εικονοστοιχείου στην εικόνα είναι μεγαλύτερη από το κατώφλι, τότε παίρνει τη μέγιστη τιμή που ορίσαμε στη συνάρτηση, το 255 (μαύρο). Σε αντίθετη περίπτωση παίρνει την τιμή 0 (άσπρο).

$$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

7.5 Μετατροπή αποτελέσματος σε ακέραιο

Μέχρι αυτό το σημείο οι διαφορές στα εικονοστοιχεία στα μάτια βρίσκονται σε μορφή δισδιάστου πίνακα. Το επόμενο βήμα είναι η μετατροπή τον πίνακα σε μονοδιάστατο με τη χρήση της συνάρτησης Scalar. Το Scalar αντιπροσωπεύει μια δομή τεσσάρων στοιχείων και χρησιμοποιείται ευρέως στην OpenCV για το πέρασμα των τιμών των

εικονοστοιχείων. πχ. `Scalar(a, b, c)`. Οι τιμές των εικονοστοιχείων αποθηκεύονται στις μεταβλητές `val[0]`, `val[1]` και `val[2]`. Στις δυαδικές εικόνες οι τιμές του γκρι αποθηκεύονται στο `val[0]` και στις έγχρωμες RGB αποθηκεύονται στα `val[0]`, `val[1]`, `val[2]`. Μ' αυτό τον τρόπο μπορούμε να πάρουμε τον τελικό αριθμό που περιέχει το άθροισμα των εικονοστοιχείων.

❖ Διαδικασίες για την εύρεση προσώπου και ματιών

Όταν καλείται η συνάρτηση `detectAndDisplay` γίνονται οι εξής διαδικασίες:

7.6 Εξίσωση ιστογράμματος

Το ιστόγραμμα μιας εικόνας είναι ένα είδος ιστογράμματος το οποίο προσφέρει μια γραφική αναπαράσταση της τονικής κατανομής των γκρι τιμών σε μια ψηφιακή εικόνα. Με το ιστόγραμμα της εικόνας, μπορούμε να αναλύσουμε τη συχνότητα της εμφάνισης των διαφορετικών επιπέδων του γκρι που περιλαμβάνονται στην εικόνα.

Ουσιαστικά η εξίσωση ιστογράμματος απλώνει τις τιμές έντασης κατά μήκος του συνόλου τιμών προκειμένου να επιτευχτεί μεγαλύτερη αντίθεση. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη όταν η εικόνα αντιπροσωπεύεται από μικρές τιμές αντίθεσης όπως εικόνες στις οποίες τόσο το φόντο όσο και το προσκήνιο είναι φωτεινά την ίδια στιγμή.

7.7 Συνάρτηση ανίχνευσης προσώπου

Η `OpenCV` έχει 4 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 του 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.

Η συνάρτηση δέχεται τις εξής παραμέτρους:

```
void CascadeClassifier::detectMultiScale ( const Mat& image,  
                                           vector<Rect>& objects,  
                                           double scaleFactor=1.1,  
                                           int minNeighbors=2,  
                                           int flags=0,  
                                           Size minSize=Size(),  
                                           Size maxSize=Size() );
```

Εξήγηση παραμέτρων:

image

Η εικόνα που χρησιμοποιείται για την ανίχνευση του προσώπου.

objects

Είναι ένα διάνυσμα ορθογωνίων όπου κάθε ορθογώνιο περιλαμβάνει το ανιχνευμένο πρόσωπο.

scaleFactor

Ο συντελεστής που καθορίζει κατά πόσο θα μεγεθύνεται η εικόνα σε κάθε σάρωση της εικόνας. Η προκαθορισμένη τιμή είναι 1.1 που σημαίνει αύξηση του παραθύρου κατά 10%.

minNeighbors

Ο ελάχιστος αριθμός των γειτονικών ορθογωνίων που αντιστοιχούν στο ίδιο αντικείμενο συγχωνεύονται σε ένα. Όλες οι ομάδες με μικρότερο αριθμό ανιχνευμένων ορθογωνίων απορρίπτονται ως λανθασμένες. Η προκαθορισμένη τιμή είναι 2.

flags

Παράμετρος που καθορίζει τον τρόπο λειτουργίας. Σ' αυτή την περίπτωση χρησιμοποιήθηκε η σημαία CV_HAAR_SCALE_IMAGE, όπου λέει στον αλγόριθμο να καθορίσει το μέγεθος της εικόνας αντί του παραθύρου ανίχνευσης.

minSize

Το ελάχιστο δυνατό μέγεθος του ανιχνευμένου αντικειμένου. Αντικείμενα με μικρότερο μέγεθος από αυτό το καθορισμένο, απορρίπτονται. Η προκαθορισμένη τιμή είναι 20x20.

maxSize

Το μέγιστο δυνατό μέγεθος του ανιχνευμένου αντικειμένου. Αντικείμενο με μεγαλύτερο μέγεθος από το καθορισμένο απορρίπτονται. Σ' αυτή την περίπτωση δεν υπάρχει κάποιο όριο.

7.8 Ανίχνευση περιοχής ματιών

Για την ανίχνευση των ματιών προτιμήθηκε η ανίχνευση και των δύο ματιών σε ένα ορθογώνιο αντί για τα δύο μάτια ξεχωριστά. Ο λόγος εξηγείται στο επόμενο κεφάλαιο.

Αρχικά υπολογίζεται το πάνω αριστερά σημείο όπου ξεκινάει το ορθογώνιο, με βάση το ορθογώνιο του προσώπου, που έχει υπολογιστεί ήδη.

Το σημείο $pt1(x,y)$ υπολογίζεται ως εξής:

το σημείο x προκύπτει από το αρχικό σημείο x του ορθογωνίου του προσώπου + το πλάτος του ορθογωνίου του προσώπου*0,18.

Το σημείο y προκύπτει από το αρχικό σημείο y του ορθογωνίου του προσώπου + το μήκος του ορθογωνίου του προσώπου*0,3.

Στη συνέχεια υπολογίζεται το κάτω δεξιά σημείο του ορθογωνίου με παρόμοιο τρόπο με το πάνω αριστερά σημείο. Πιο αναλυτικά το σημείο $pt2(x,y)$ υπολογίζεται ως εξής:

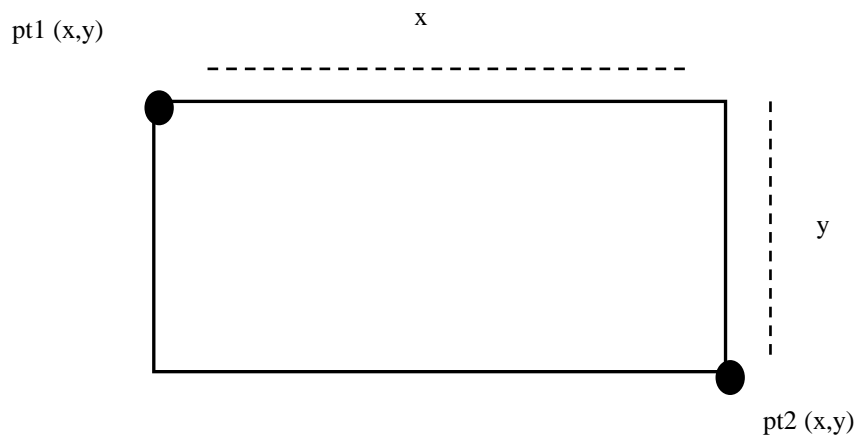
το σημείο x προκύπτει από το αρχικό σημείο x του ορθογωνίου του προσώπου + το πλάτος του ορθογωνίου του προσώπου*0,84

Το σημείο y προκύπτει από το αρχικό σημείο y του ορθογωνίου του προσώπου + το μήκος του ορθογωνίου του προσώπου*0,54

Τέλος για τον σχηματισμό του ορθογωνίου των ματιών χρησιμοποιείται η κλάση Rect.

Η Rect είναι μια πρότυπη κλάση για ορθογώνια δύο διαστάσεων και περιγράφεται από τις εξής παραμέτρους:

- Τις συντεταγμένες του πάνω αριστερού άκρου ($pt1(x,y)$). Δηλώνονται ως $pt1.x$ και $pt1.y$.
- Το μήκος και το πλάτος του ορθογωνίου. Το μήκος υπολογίζεται από την απόλυτη διαφορά των x των σημείων $pt1$ και $pt2$ και το πλάτος από την απόλυτη διαφορά των y των σημείων.

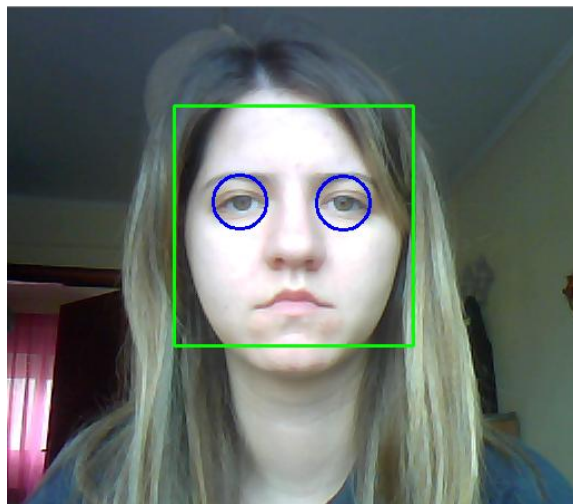


Εικόνα 7-1 Το ορθογώνιο των ματιών σχηματικά

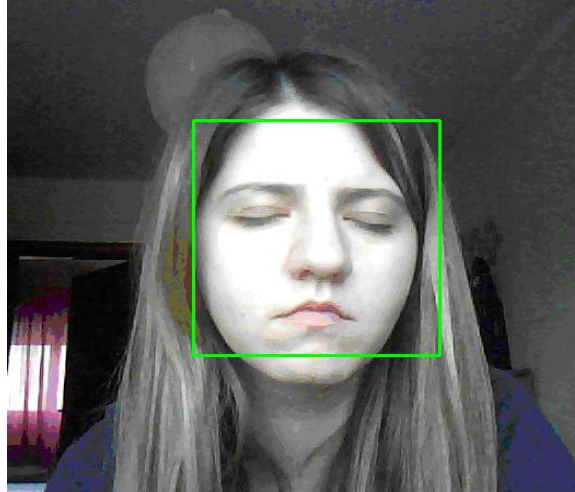
8. Πειράματα

8.1 Haar Ανιχνευτής ματιών

Στις αρχικές μας δοκιμές μαζί με τον ανιχνευτή προσώπου χρησιμοποιήθηκε κι ένας ανιχνευτής ματιών που σκοπός του είναι να ανιχνεύει και τα δύο μάτια ξεχωριστά. Η συνάρτηση ανίχνευσης που χρησιμοποιήθηκε ήταν η ίδια που χρησιμοποιήθηκε και για το πρόσωπο μόνο που σ' αυτή την περίπτωση η ανίχνευση έγινε με βάση το πρόσωπο. Η συνάρτηση είναι η εξής: `eyes_cascade.detectMultiScale(faceROI, eyes_obj, 1.1, 2, 0|CV_HAAR_SCALE_IMAGE, Size(20, 20))`. Το `faceROI` είναι ένα `mat` που περιέχει τα εικονοστοιχεία της περιοχής του προσώπου που ανιχνεύτηκε. `Eyes_obj` είναι ένα διάνυσμα ορθογωνίων όπου περιλαμβάνει τα ορθογώνια του κάθε ματιού ξεχωριστά. Και οι λοιπές παράμετροι που αναλύθηκαν στην ανίχνευση προσώπου.



Εικόνα 8-1 Ανιχνευτής προσώπου και ματιών όταν τα μάτια είναι ανοιχτά



Εικόνα 8-2 Ανιχνευτής προσώπου και ματιών όταν τα μάτια είναι κλειστά

Το αποτέλεσμα όμως με τη χρήση και των δύο ταξινομητών δεν ήταν το αναμενόμενο. Ο ρυθμός αναπαραγωγής των πλαισίων αργούσε πολύ με αποτέλεσμα όταν γίνεται μία κίνηση από το χρήστη, να φαίνεται στο παράθυρο του προγράμματος μετά από 2-3 δευτερόλεπτα κι επίσης πολλές φορές δε φαινόταν καν το βλεφάρισμα των ματιών. Και οι δύο ταξινομητές κάνουν πολλές ανιχνεύσεις το δευτερόλεπτο κι αυτό κάνει πολύ αργό το σύστημα. Εφόσον ο τρόπος που διαλέξαμε για να γίνει η σύγκριση των ματιών σε κάθε πλαίσιο δεν απαιτούσε την ανίχνευση των δύο ματιών ξεχωριστά, ο ανιχνευτής ματιών αφαιρέθηκε.

8.2 Λανθασμένες ανιχνεύσεις

Σε συνθήκες κακού φωτισμού υπάρχει μεγάλη περίπτωση να γίνει λάθος ανίχνευση προσώπου και ματιών. Αυτό γίνεται διότι δεν έχει γίνει επαρκής εκπαίδευση του ταξινομητή σε κακές συνθήκες φωτισμού. Για καλύτερη απόδοση του classifier χρειάζεται επανεκπαίδευση του με περισσότερες εικόνες κακού φωτισμού.

Επίσης ο classifier παρουσιάζει κακή απόδοση σε εικόνες υψηλής ανάλυσης, κι αυτό επειδή η εκπαίδευση έχει περιοριστεί σε εικόνες 640x480.



Εικόνα 8-3 Ανίχνευση περισσότερων από ένα ζευγάρι ματιών σε κακό φωτισμό. Ο ταξινομητής μπερδεύει κάποιο σημείο της εικόνας για πρόσωπο και κατά συνέπεια υπολογίζει ακόμη ένα ζευγάρι μάτια.

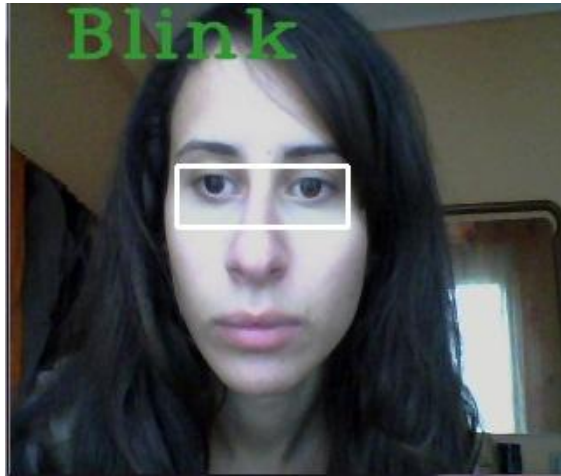
Υπάρχουν περιπτώσεις που δε γίνεται ανίχνευση βλεφαρίσματος διότι λόγω φωτισμού δε φαίνεται η διαφορά των εικονοστοιχείων στην περιοχή των ματιών σε δύο διαδοχικά πλαίσια.



Εικόνα 8-4 Μη ανίχνευση βλεφαρίσματος με φωτισμό λάμπας

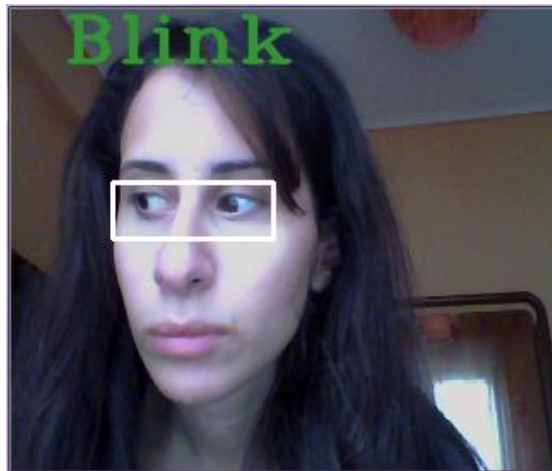
Υπάρχουν περιπτώσεις που μπορεί να υπάρχει βλεφάρισμα από την πλευρά του χρήστη, να ανιχνεύεται από το πρόγραμμα, όμως να μην το πιάνει η κάμερα του υπολογιστή.

Αυτό μπορεί να γίνει λόγω της μικρής καθυστέρησης που υπάρχει στην εναλλαγή των πλαισίων ή αν το βλεφάρισμα είναι γρήγορο.



Εικόνα 8-5 Βλεφάρισμα το οποίο ανιχνεύτηκε αλλά το ανοιγόκλεισμα των ματιών δεν εμφανίζεται στο παράθυρο.

Τέλος υπάρχει περίπτωση ο χρήστης να κουνήσει την κόρη των ματιών του ή να κουνήσει το σώμα του και αυτό αλλάζει τα εικονοστοιχεία της εικόνας. Ως αποτέλεσμα ανιχνεύεται βλεφάρισμα χωρίς να υπάρχει στην πραγματικότητα.



Εικόνα 8-6 Ανίχνευση βλεφαρίσματος από την κίνηση της κόρης του ματιού

Σύμφωνα με τα πειράματα βγήκαν τα εξής συμπεράσματα:

Σε φυσικό φωτισμό, η αναγνώριση του βλεφαρίσματος γίνεται αρκετά εύκολα και είναι σπάνιο να μην ανιχνευθεί το βλεφάρισμα. Το αρνητικό είναι πως οποιαδήποτε κίνηση του ματιού μπορεί να ερμηνευτεί ως βλεφάρισμα, γι' αυτό οι κινήσεις πρέπει να είναι περιορισμένες.

Σε φωτισμό με λάμπα, η κατάλληλη απόσταση για την ανίχνευση είναι περίπου 20 εκατοστά από την κάμερα. Αν η απόσταση είναι μεγαλύτερη τότε υπάρχει πιθανότητα να μην ανιχνεύεται κάθε φορά το βλεφάρισμα.

9. Συμπεράσματα

9.1 Σε σχέση με την OpenCV

Η OpenCV είναι μια πολύ χρήσιμη βιβλιοθήκη για επεξεργασία βίντεο και εικόνας με πολλές επιλογές και δυνατότητες. Υπάρχουν πληθώρα συναρτήσεων οι οποίες με μία γραμμή κώδικα μπορεί να καταφέρει αυτό που χρειάζεται παραπάνω γραμμές αν ήταν σε απλή C++. Για παράδειγμα στην C++ για να γίνει προσπέλαση των εικονοστοιχείων μιας εικόνας χρειάζονται δύο if, μία για την προσπέλαση των γραμμών του πίνακα και μία των στηλών, ενώ με την OpenCV αυτή η διαδικασία γίνεται με μόνο μία εντολή. Κατά συνέπεια μειώνεται πολύ ο όγκος του κώδικα και είναι πιο εύκολη η επεξεργασία του.

Οι δυσκολίες που αντιμετωπίσαμε με την OpenCV ήταν ότι δεν υπάρχει επαρκής τεκμηρίωση. Πέρα από τις οδηγίες που υπάρχουν στην επίσημη ιστοσελίδα της OpenCV, οι οποίες ανανεώνονται συνεχώς όποτε βγαίνει καινούργια έκδοση της βιβλιοθήκης, δεν υπάρχει επαρκές υλικό στο διαδίκτυο για τις συναρτήσεις και τη λειτουργία τους. Στην έκδοση 1.x της OpenCV χρησιμοποιούνταν εντολές για C interface, ενώ στις επόμενες εκδόσεις οι συναρτήσεις άλλαξαν σε C++ interface. Οι περισσότερες online πηγές, ειδικά τα βιβλία, αναφέρονται στην παλιά έκδοση της OpenCV κι έτσι είναι δύσκολο να βρεθεί υλικό για την εφαρμογή.

9.2 Σε σχέση με την εφαρμογή

Στόχος της εργασίας ήταν η δημιουργία μιας εφαρμογής πραγματικού χρόνου. Αυτός ο στόχος επετεύχθη εφόσον τα αποτελέσματα της εφαρμογής φαίνονται εκείνη τη στιγμή. Στην προσπάθεια εντοπισμού της κίνησης έχουμε τις εξής δυσκολίες: θόρυβος, κίνηση αντικειμένων στο χώρο, αλλαγή φωτισμού και κακής ποιότητας του μέσου λήψης εικόνας. Η οποιαδήποτε μορφή θορύβου μπορεί να θεωρηθεί ως κίνηση με αποτέλεσμα να μπερδέψει την όλη διαδικασία εντοπισμού του βλεφαρίσματος των ματιών. Μια μετακίνηση της κάμερας ή μια μετακίνηση του ατόμου που χειρονομεί μπορεί να προκαλέσει τεράστια προβλήματα στον εντοπισμό κίνησης. Η κίνηση αντικειμένων στο

χώρο είναι ένα άλλο πρόβλημα με σοβαρές επιπτώσεις στον εντοπισμό κίνησης. Εάν κατά την διάρκεια της λήψης του βίντεο περάσει κάποιο αντικείμενο ή κάποιος άνθρωπος μπροστά από την κάμερα, ή κάποιο αντικείμενο με κινητά μέρη κινηθεί, αυτό εντοπίζεται ως κίνηση από το πρόγραμμα και μπορεί να επηρεάσει σημαντικά τα αποτελέσματα. Ακόμα η αλλαγή φωτισμού έχει καταστροφικές συνέπειες. Σχεδόν όλα τα εικονοστοιχεία της εικόνας αλλάζουν κατά πολύ. Η μέθοδος εντοπισμού βλεφαρίσματος που χρησιμοποιείται θα μου δείξει ότι όλα τα εικονοστοιχεία κινήθηκαν. Έτσι δεν υπάρχει περίπτωση να εντοπιστεί το βλεφάρισμα, δεδομένου ότι άλλαξε ο φωτισμός. Επομένως ο φωτισμός πρέπει να παραμένει σταθερός συνεχώς προκειμένου να μην αλλοιώνονται τα αποτελέσματα. Τέλος ένα κακής ποιότητας μέσο λήψης εικόνας έχει ως αποτέλεσμα την εισαγωγή θορύβου και κατά συνέπεια την αλλοίωση των αποτελεσμάτων με την εισαγωγή ανύπαρκτης κίνησης.

Άρα για την καλύτερη απόδοση του συστήματος:

- Ο χρήστης πρέπει να βρίσκεται σε κοντινή απόσταση από την κάμερα του υπολογιστή
- Κατά προτίμηση να υπάρχει φυσικός φωτισμός ημέρας
- Η πηγή του φωτός να βρίσκεται απέναντι από το χρήστη
- Να μην υπάρχει κίνηση του σώματος ή μεγάλη κίνηση των ματιών δεξιά-αριστερά-πάνω-κάτω.

9.3 Μελλοντικές βελτιώσεις

Για τη βελτίωση της απόδοσης της ανίχνευσης μπορούν να γίνουν πολλά πράγματα, κάποια από τα οποία είναι εύκολο να εφαρμοστούν.

Κάποιες προσθήκες θα ήταν η βελτίωση της ποιότητας της εικόνας όταν ο φωτισμός δεν είναι καλός ώστε να ανιχνεύεται πιο εύκολα το βλεφάρισμα.

Θα μπορούσε να αναπτυχθεί μια εφαρμογή για λεπτομερέστερη ανίχνευση βλεφαρίσματος μέσω της ανίχνευσης της ίριδας του ματιού. Όσο θα ανιχνεύεται η ίριδα θα σημαίνει πως το μάτι είναι ανοιχτό, ενώ όταν δεν θα υπάρχει ανίχνευση θα σημαίνει πως θα είναι κλειστό.

Μια άλλη βελτίωση θα ήταν η ανάπτυξη της αντίστοιχης web εφαρμογής για αλληλεπίδραση με τον browser για άτομα με ειδικές ανάγκες. Αν για παράδειγμα χρειαζόταν registration σε μια ιστοσελίδα, να υπήρχε η ανίχνευση του βλεφαρίσματος για να γίνεται αυτή η διαδικασία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. K. Curran, X. Li and N. Mc Caughley, “The Use of Neural Networks in Real-time Face Detection”, Journal of Computer Sciences 1 (1): 47-62, 2005.
2. E. Hjelmas and B. Kee Low, “Face Detection: A survey”, Computer Vision and Image Understanding 83, 236–274, 2001.
3. L. C. De Silva, K. Aizawa and M. Hatori, “Detection and Tracking of Facial Features by using a Facial Feature Model and Deformable Circular Templates Inst. of Electronics, Information and Communication Engineers, 1995.
4. H. A. Rowley, S. Baluja, and T. Kanade, “Neural Network-Based Face Detection”, IEEE Trans. Pattern, 1998.
5. K. K. Sung and T. Poggio, “Example-based Learning for View-based Human Face Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995.
6. R. Krutsch and D. Tenorio, “Histogram Equalization”, Freescale Semiconductor, Document Number: AN4318, June 2011.
7. P. Viola and M. Jones, “Robust Real-time Object Detection”, Second international workshop on statistical and computational theories of vision – modeling, learning, computing and sampling, July 13 2001.
8. R. Szeliski, “Computer Vision: Algorithms and Applications”, September 2010.
9. G. Bradski and A. Kaehler, “Learning OpenCV”, O’Reilly Media Inc., 2008
10. P. I. Wilson and J. Fernandez, “Facial feature detection using haar classifiers”, CCSC: South Central Conference, 4 April 2006
11. R. Lienhart and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, Intel Labs, Intel Corporation, USA
12. OpenCV Library, www.OpenCV.org
13. Visual Studio 2010 Express
<http://www.microsoft.com/visualstudio/eng/downloads#d-2010-express>
14. G. Agam, “Introduction to programming with OpenCV”, Department of Computer Science, 2004
15. Wikipedia: C++ <http://el.wikipedia.org/wiki/C++>, 5 May 2013

16. Wikipedia: Support Vector Machine

http://en.wikipedia.org/wiki/Support_vector_machine , 16 November 2013

17. R. Hewitt, "Seeing with OpenCV", Servo Magazine, February 2007

A. Κώδικας

```
#include "stdafx.h"
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;

/** Function Headers */
void detectAndDisplay(Mat, std::vector<Rect>& );

/** Global variables */
String face_cascade_name = "haarcascade_frontalface_alt.xml"; //Load
the cascade
CascadeClassifier face_cascade;
RNG rng(12345);

/** @function main */
int main( int argc, const char** argv )
{
    Mat cam_frame;
    Mat cam_frame2;
    Mat img_gray;
    Mat img_prev;
    Mat difference;
    Mat difference2;
    Mat my_eyes;
    Mat eyes4;

    //Load the cascades
    if( !face_cascade.load( face_cascade_name ) )
    { printf("--(!)Error loading\n"); return -1; };

    VideoCapture cap(0);

    bool first_frame = true;

    while( true )
    {
```

```

cap >> cam_frame;
resize (cam_frame,cam_frame,Size(320,270));

if (first_frame)
{
    img_prev=cam_frame.clone();
    first_frame = false;
    continue;
}

//eyes for first frame
Mat Image=cam_frame;
int my_index;
Mat my_I = Image.clone();
std::vector<Rect> my_rect1;
detectAndDisplay(my_I,my_rect1); //return all eyes rectangles in
the vector my_rect1

for(my_index=0; my_index<my_rect1.size(); my_index++)
{
    rectangle(Image, my_rect1[my_index], Scalar( 255, 255, 255
), 2, 8, 0 );//Draw the eyes-rectangle on image
my_eyes=Image(my_rect1[my_index]); //Creates matrix out of
the eyes ROI for all eyes rectangles in the vector my_rect1
resize(my_eyes, eyes4, Size(150, 150));
namedWindow("my_eyes"+my_index, CV_WINDOW_FREERATIO);
imshow("my_eyes"+my_index, eyes4);
resize(my_eyes, eyes4, Size(140, 120));
}

namedWindow("rectangle", CV_WINDOW_AUTOSIZE);
imshow("rectangle",Image );

//eyes for the second frame

cap >> cam_frame2;
resize (cam_frame2,cam_frame2,Size(320,270));

Mat Image2=cam_frame2;
int my_index2;
Mat my_I2 = Image2.clone();
std::vector<Rect> my_rect2;
detectAndDisplay(my_I,my_rect2); //return all eyes rectangles in
the vector my_rect1
Mat my_eyes2;
Mat eyes5;

```

```

for(my_index2=0; my_index2<my_rect2.size(); my_index2++)
{
    rectangle(Image2, my_rect2[my_index2], Scalar( 255, 255,
    255 ), 2, 8, 0 );//Draw the eyes-rectangle on image
    my_eyes2=Image2(my_rect2[my_index2]); //Creates matrix out
    of the eyes ROI for all eyes rectangles in the vector
    my_rect1
    resize(my_eyes2, eyes5, Size(140, 120));
}

//calculate sum of absolute differences

absdiff(eyes4,eyes5,difference);
threshold(difference, difference2, 80, 255, CV_THRESH_BINARY);
Scalar res=sum(difference2);
int sum=res.val[0];
cout<<sum<<endl;

//if result bigger than 5000 then show "blink"
if (sum>5000)
{
    putText(Image, "Blink",
    cvPoint(30,30),FONT_HERSHEY_COMPLEX_SMALL,          2,
    cvScalar(50,150,50), 2, CV_AA);
    namedWindow("rectangle", CV_WINDOW_AUTOSIZE);
    imshow("rectangle", Image);
}

img_prev=img_gray.clone();

int k;                                //Press ESC to exit loop
k = waitKey(33);
if (k==27)
    break;
}
return 0;
}

void detectAndDisplay(Mat I, std::vector<Rect>& my_rect)
{

```

```

std::vector<Rect> face_obj;

Mat I_gray;

if(I.type() != 0)
{
cvtColor(I, I_gray, CV_BGR2GRAY);
equalizeHist( I_gray, I_gray);
}
else
{
equalizeHist( I, I_gray);
}

//Detect faces
face_cascade.detectMultiScale( I_gray, face_obj, 1.1, 2,
0|CV_HAAR_SCALE_IMAGE, Size(20, 20) );

for( int i=0; i<face_obj.size(); i++)//Run for each face
{

//In each face detect eyes
Point pt1(face_obj[i].x + face_obj[i].width*0.18 ,
face_obj[i].y + face_obj[i].height*0.3);
Point pt2(face_obj[i].x + face_obj[i].width*0.84 ,
face_obj[i].y + face_obj[i].height*0.54);
Rect myrect3(pt1.x, pt1.y, abs(pt1.x-pt2.x), abs(pt1.y-
pt2.y));
my_rect.push_back(myrect3);

}

}

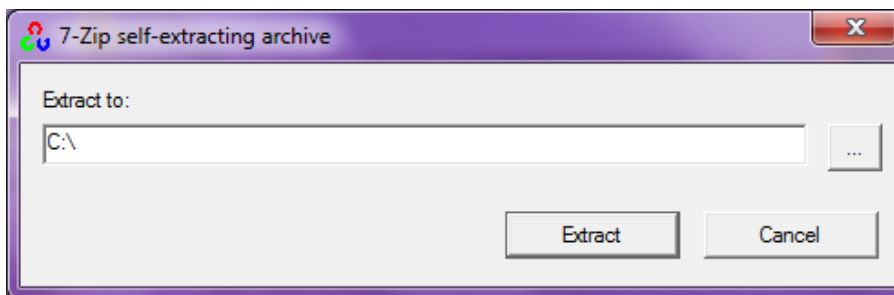
```

B. Οδηγίες εγκατάστασης OpenCV

Τα βασικά βήματα είναι να κατεβάσουμε και να εγκαταστήσουμε το Visual Studio 2010 Express από τη σελίδα <http://www.microsoft.com/visualstudio/eng/downloads#d-2010-express> καθώς και τη βιβλιοθήκη OpenCV που είναι διαθέσιμη στην επίσημη ιστοσελίδα <http://OpenCV.org/downloads.html>.

Εγκατάσταση της βιβλιοθήκης

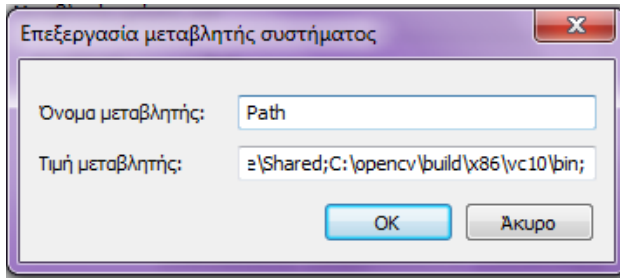
Αφού έχουμε κατεβάσει την τελευταία έκδοση της OpenCV (στη συγκεκριμένη περίπτωση την έκδοση 2.4.4 για Windows) ανοίγουμε το εκτελέσιμο αρχείο και κάνουμε extract στον φάκελο C:\.



Εικόνα Π-1 Extract εκτελέσιμου αρχείου

Στη συνέχεια χρειάζεται να προσθέσουμε τις μεταβλητές περιβάλλοντος. Πηγαίνουμε Πίνακας Ελέγχου → Σύστημα και Ασφάλεια → Σύστημα → Ρυθμίσεις Συστήματος για Προχωρημένους → Μεταβλητές περιβάλλοντος.

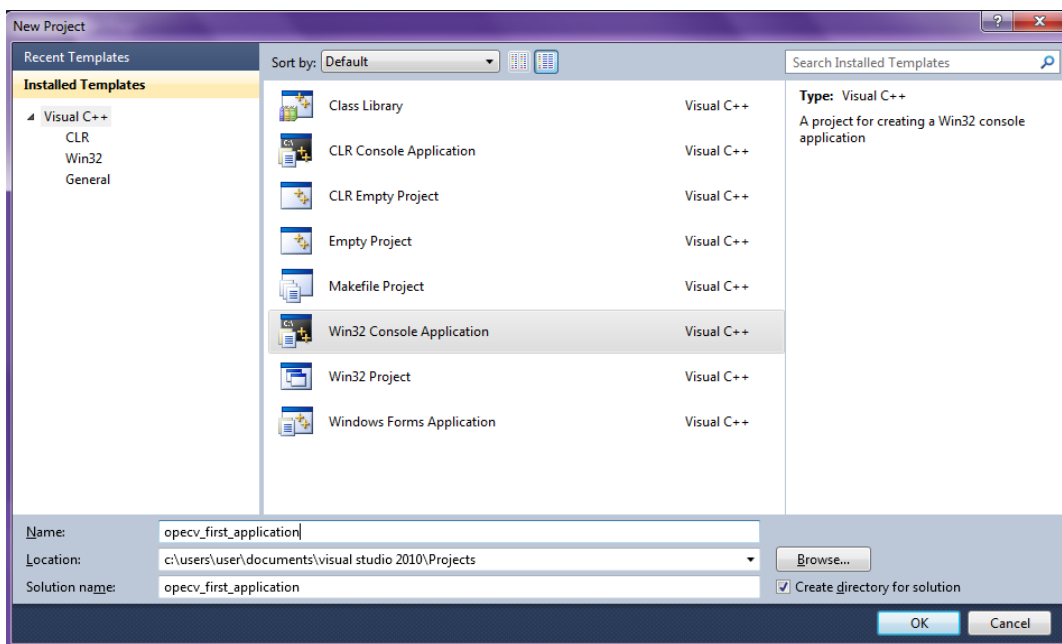
Στη μεταβλητή Path επιλέγουμε Επεξεργασία και προσθέτουμε C:\OpenCV\build\x86\vc10\bin\ για λειτουργικό σύστημα 32 bit και C:\OpenCV\build\x64\vc10\bin\ για λειτουργικό σύστημα 63 bit και OK.



Εικόνα Π-2 Μεταβλητές περιβάλλοντος

Δημιουργία project στη Visual Studio Express 2010

Για να γίνει η σύνδεση της OpenCV με τη Visual Studio κάνουμε τα παρακάτω βήματα: Ανοίγουμε το Visual Studio και επιλέγουμε File → New Project → Win32 Console Application. Δίνουμε το όνομα που θέλουμε στο project και OK.



Εικόνα Π-3 Δημιουργία καινούργιου project

Στο επόμενο παράθυρο πατάμε Finish.

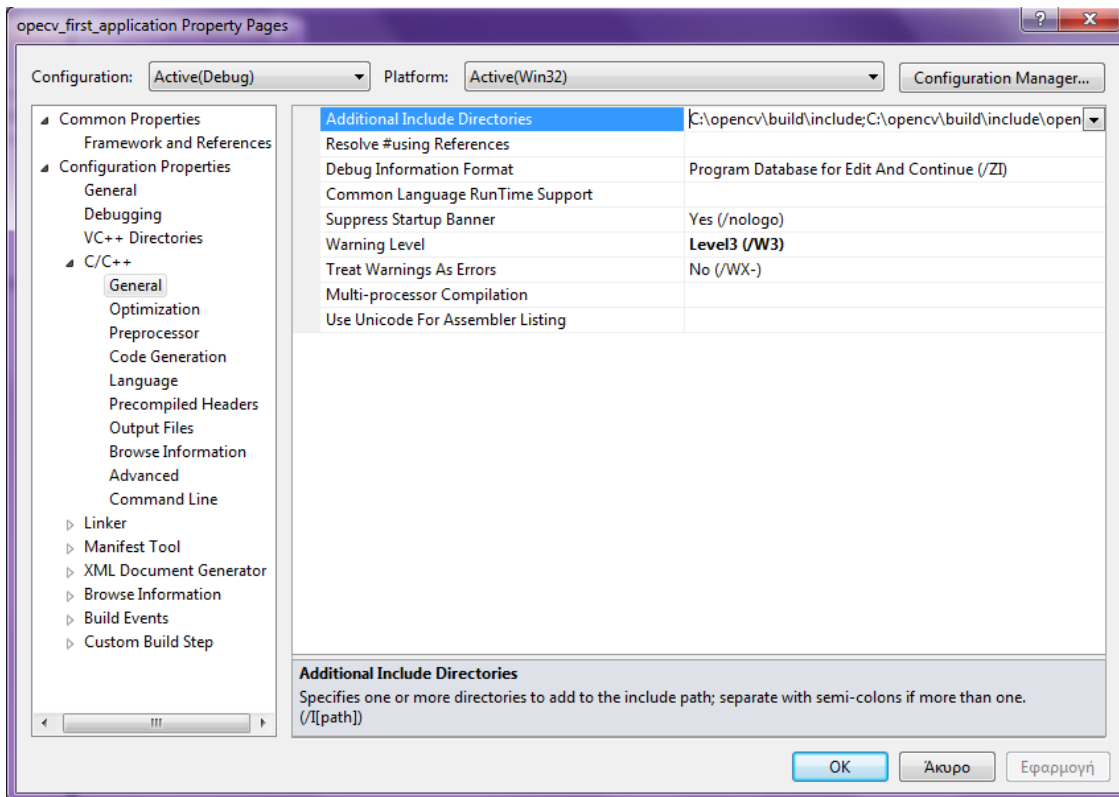
Project → Properties → Configuration Properties.

Στο C/C++ → General → Additional Include Directories εισάγουμε τους φακέλους Include.

Εφόσον ο φάκελος OpenCV βρίσκεται στον C:\ προσθέτουμε τις εξής διαδρομές:

C:\OpenCV\build\include

C:\OpenCV\build\include\OpenCV



Εικόνα Π-4 Προσθήκη include directories

Κατόπιν επιλέγουμε Linker->Input και στο Additional Dependencies βάζουμε τις εξής βιβλιοθήκες:

C:\OpenCV\build\x86\vc10\lib\OpenCV_core244d.lib

C:\OpenCV\build\x86\vc10\lib\OpenCV_highgui244d.lib

C:\OpenCV\build\x86\vc10\lib\OpenCV_video244d.lib

C:\OpenCV\build\x86\vc10\lib\OpenCV_ml244d.lib

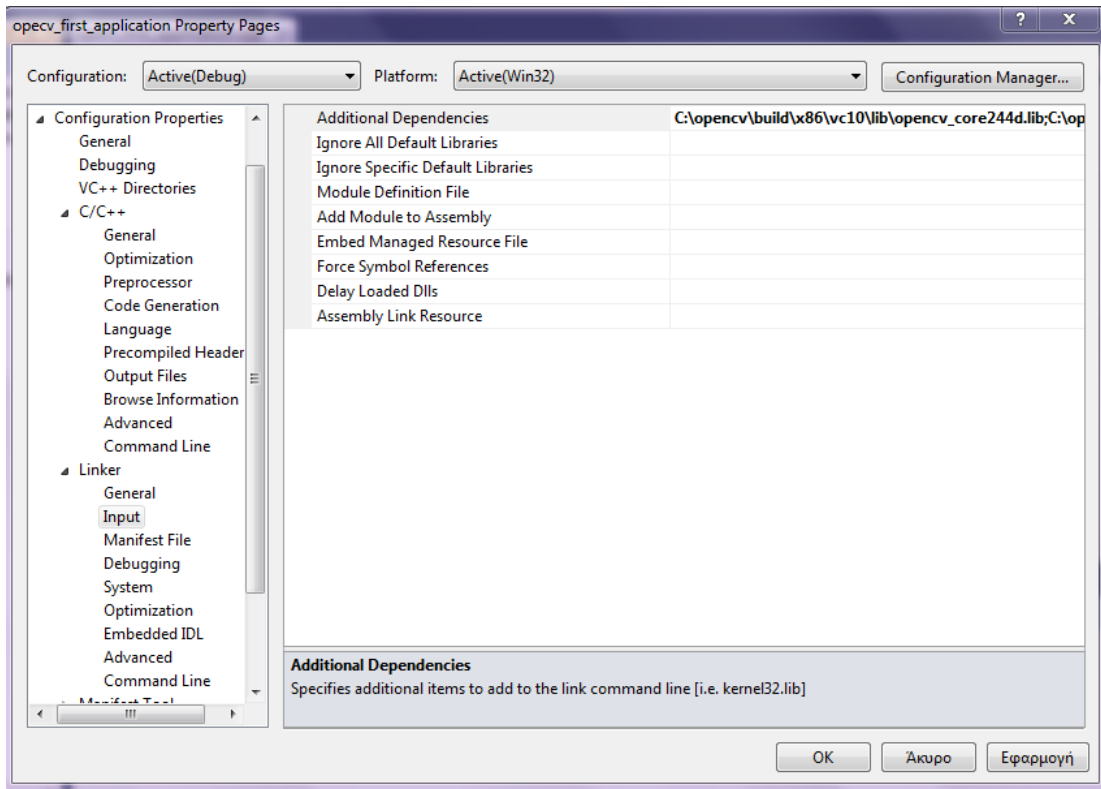
C:\OpenCV\build\x86\vc10\lib\OpenCV_legacy244d.lib

C:\OpenCV\build\x86\vc10\lib\OpenCV_imgproc244d.lib

C:\OpenCV\build\x86\vc10\lib\OpenCV_objdetect244d.lib

Σημείωση: το 244d είναι η έκδοση της OpenCV, δηλαδή 2.4.4. Αν είχαμε άλλη έκδοση, πχ 2.4.5 θα έγραφε 255d.

Τέλος επιλέγουμε Εφαρμογή και OK.



Εικόνα Π-5 Εισαγωγή βιβλιοθηκών

Και είμαστε έτοιμοι για την πρώτη μας εφαρμογή!

Σημείωση: Η εγκατάσταση έγινε σε λειτουργικό σύστημα windows 7, 32 bit.