



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Οπτική Αναγνώριση Χαρακτήρων (OCR). Περιγραφή
ενός νέου αλγορίθμου και κατασκευή εφαρμογής Η/Υ
που τον υλοποιεί για το σκοπό της Αυτόματης
Αναγνώρισης Πινακίδων Κυκλοφορίας (ANPR)**

Του φοιτητή

Πέτρου Χ. Δρακούλη

Αρ. Μητρώου: 052926

Επιβλέπων καθηγητής

Δρ. Κωνσταντίνος Γουλιάνας

Θεσσαλονίκη 2013

ΠΕΡΙΛΗΨΗ

Το πρόβλημα της οπτικής αναγνώρισης χαρακτήρων από τον ηλεκτρονικό υπολογιστή σχεδόν καθ' αποκλειστικότητα αντιμετωπίζεται με την εφαρμογή διατάξεων τεχνητών νευρωνικών δικτύων, με όλα πλεονεκτήματα και τα μειονεκτήματα που εγγενώς αυτά φέρουν. Κυρίαρχο σημείο της επίλυσης είναι η μετατροπή των μορφολογικών χαρακτηριστικών των χαρακτήρων σε πληροφορία εισόδου των δικτύων. Αυτό συνήθως γίνεται προβάλλοντας την εισερχόμενη μορφή σ' ένα πλέγμα τετραγώνων και κωδικοποιώντας τα τετράγωνα που ενεργοποιούνται. Απόρροια αυτής της διαδικασίας είναι η πλήρης εξάρτηση της διεργασίας από τη διατήρηση της στροφικής ευθυγράμμισης των εισαγόμενων στο σύστημα μορφών, μεταξύ των φάσεων της εκπαίδευσης και της ανάκλησης. Σε πραγματικές εφαρμογές αυτό συνήθως επιτυγχάνεται με τη χρήση αλγορίθμων ευθυγράμμισης των εισερχόμενων φωτογραφιών, στο στάδιο της προεπεξεργασίας. Αντ' αυτού, στα πλαίσια αυτής της πτυχιακής εργασίας επινοήθηκε και προτείνεται ένας νέος, μη-νευρωνικός και ανεξαρτήτου κατεύθυνσης εισόδου, αλγόριθμος οπτικής αναγνώρισης μορφών. Η λειτουργία αυτού του αλγορίθμου επαληθεύεται και επιδεικνύεται για το σκοπό της αυτόματης αναγνώρισης πινακίδων κυκλοφορίας με την χρήση εφαρμογής ηλεκτρονικού υπολογιστή που κατασκευάστηκε σε γλώσσα προγραμματισμού Java™, ειδικά γι' αυτή την εργασία.

ABSTRACT

The problem of Optical Character Recognition by computers is almost exclusively addressed with the application of set-ups of artificial neural networks with all the advantages and disadvantages they inherently bear. Ascendant point of solution is the transformation of the morphological characteristics of the characters into network inputs. This is usually done by projecting input figures to a grid and encoding activated squares. Aftereffect of this process is the total dependence of the task on conserving the rotational alignment of input figures, between the phases of training and recall. In real-life applications this is usually achieved by using image alignment algorithms in a pre-processing stage. Instead, as part of this thesis, was conceived and is proposed a new, non-neural based and input direction independent, algorithm of optical figure recognition. The function of this algorithm is being verified and demonstrated for the purpose of Automatic Number Plate Recognition with the use of a computer application written in Java™ programming language, specifically for this assignment.

ΕΥΧΑΡΙΣΤΙΕΣ

Αφιερωμένο στον κύριο Νίκο Δ. Μυλωνά, χωρίς τον οποίο η παιδική μου περιέργεια για τον προγραμματισμό ίσως να μην μεταμορφωνόταν ποτέ στο πάθος που με χαρακτηρίζει σήμερα. Σ' αυτόν που με έκανε να πιστέψω πως όπου υπάρχει ο Μαθητής εμφανίζεται κι ο Δάσκαλος και που, αν και γνωρίζω ελάχιστα, θεωρώ και θα θεωρώ πάντα μέντορά μου.

Στο πρόσωπό του θέλω επίσης να ευχαριστήσω, από καρδιάς και μυαλού, όλους τους πραγματικούς Δασκάλους που γνώρισα στη ζωή μου, όποιο πόστο κι αν κατείχαν. Είτε υπήρξαν γνωριμίες μιας ημέρας, είτε φίλοι που με ακολουθούν για μια ζωή. Είτε δάσκαλοι στο σχολείο και το ΤΕΙ, είτε στα βιβλία. Να ξέρουν πως ακόμη κι αν για τους σκοπούς αυτής της εργασίας δεν τους μνημονεύω προσωπικά, τους θυμάμαι καλά, έναν-έναν, και πως οι πράξεις τους καθορίζουν μέχρι και σήμερα ένα κομμάτι αυτού που είμαι.

Επίσης, θέλω να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου που με προστατεύει, πιστεύει και στηρίζει σε ό,τι κι αν κάνω. Στον πατέρα μου Χρήστο, την μητέρα μου Νίκη, τον αδερφό μου Σταύρο και τη γιαγιά μου Παρασκευούλα.

Τέλος, θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Κωνσταντίνο Γουλιάνα, για την ανεκτικότητα, το ανοιχτό μυαλό και την κατανόηση που επέδειξε απέναντι μου κατά την περάτωση αυτής της πτυχιακής εργασίας.

Εκ βαθέων, Πέτρος Δρακούλης

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	5
ΕΥΧΑΡΙΣΤΙΕΣ.....	7
ΠΕΡΙΕΧΟΜΕΝΑ.....	9
1 ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ	11
ΕΙΣΑΓΩΓΗ	11
1.1 ΙΣΤΟΡΙΑ.....	12
1.2 Η ΣΗΜΑΣΙΑ ΤΗΣ ΓΙΑ ΤΟΥΣ ΤΥΦΛΟΥΣ	15
1.3 Η ΣΗΜΑΣΙΑ ΤΗΣ ΓΙΑ ΤΟΥΣ ΟΡΓΑΝΙΣΜΟΥΣ ΚΑΙ ΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ	16
1.4 ΕΙΔΗ ΕΦΑΡΜΟΓΩΝ	18
1.5 ΔΥΝΑΤΟΤΗΤΕΣ ΕΦΑΡΜΟΓΩΝ	20
1.6 ΑΝΑΓΝΩΡΙΣΗ ΜΕ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	23
1.7 ΑΝΑΓΝΩΡΙΣΗ ΜΕ ΣΤΑΤΙΣΤΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ.....	30
2 ΠΕΡΙΓΡΑΦΗ ΕΝΟΣ ΝΕΟΥ ΑΛΓΟΡΙΘΜΟΥ	35
ΕΙΣΑΓΩΓΗ	35
2.1 Ο ΑΛΓΟΡΙΘΜΟΣ	38
2.2 ΒΕΛΤΙΩΣΕΙΣ.....	44
3 ΚΑΤΑΣΚΕΥΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΥΤΟΜΑΤΗΣ ΑΝΑΓΝΩΡΙΣΗΣ ΠΙΝΑΚΙΔΩΝ ΚΥΚΛΟΦΟΡΙΑΣ	51
ΕΙΣΑΓΩΓΗ	51
3.1 ΤΟ ΠΡΟΓΡΑΜΜΑ.....	53
3.2 ΒΕΛΤΙΩΣΕΙΣ.....	61
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	65

1 ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ

ΕΙΣΑΓΩΓΗ

Η Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition - OCR) είναι η μηχανική ή ηλεκτρονική μετατροπή σαρωμένων εικόνων χειρόγραφου ή τυποποιημένου κειμένου σε κωδικοποιημένη μορφή χαρακτήρων κειμένου ηλεκτρονικού υπολογιστή.

Χρησιμοποιείται ευρέως ως ένας τρόπος εξαγωγής δεδομένων από κάποιας μορφής έντυπο, όπως έγγραφα, αποδείξεις πωλήσεων, γράμματα αλληλογραφίας, βιβλία κ.τ.λ. Είναι δηλαδή μια μέθοδος ψηφιοποίησης (digitizing) έντυπων κειμένων έτσι ώστε να μπορούν να προσπελαστούν ηλεκτρονικά, να αποθηκευτούν σε μικρότερο χώρο, να είναι προσβάσιμα από τον παγκόσμιο ιστό και να μπορούν να χρησιμοποιηθούν περαιτέρω σε διαδικασίες όπως είναι η αυτόματη μετάφραση (machine translation), η εξόρυξη δεδομένων (data mining) ή η μετατροπή κειμένου σε φωνή από τον ηλεκτρονικό υπολογιστή (text-to-speech).

Η οπτική αναγνώριση χαρακτήρων αποτελεί πεδίο έρευνας για τους τομείς της αναγνώρισης προτύπων (pattern recognition), της τεχνητής νοημοσύνης (artificial intelligence) και της κατανόησης των εικόνων από τον ηλεκτρονικό υπολογιστή (computer vision).

Οι πρώιμες εφαρμογές του είδους βασίζονταν στην απομνημόνευση φωτογραφιών ενός-ενός των γραμμάτων, τα οποία και έψαχναν μεμονωμένα μέσα στα προς έλεγχο κείμενα. Με τον καιρό αναπτύχθηκαν πιο έξυπνα συστήματα τα οποία εξετάζουν τις λέξεις σαν σύνολα και τα οποία προχωρούν σε «συλλογιστικά άλματα» λαμβάνοντας υπ' όψιν τους και το νόημα της λέξης στόχου στην προσπάθειά τους να κάνουν τη σωστή αναγνώριση (Για παράδειγμα μπορεί να χρησιμοποιούν λεξικό για να μαντέψουν τη σωστή λέξη, ακόμη και αν έχει αναγνωριστεί λάθος κάποιο γράμμα της). Μερικά συστήματα είναι ικανά ακόμη και αναπαράγουν σε κάποιο βαθμό τη διαμόρφωση (print layout) της πρωτότυπης σελίδας (Δηλαδή τις εικόνες, τις στήλες ή άλλο γραφιστικό και στιλιστικό περιεχόμενο).

1.1 ΙΣΤΟΡΙΑ

Η εκκίνηση της ενασχόλησης με την οπτική αναγνώριση χαρακτήρων μπορεί να αναχθεί στην επέκταση της τηλεγραφίας και στις προσπάθειες ανάπτυξης συσκευών ανάγνωσης για τους τυφλούς.

Το 1914 ο Emanuel Goldberg ανέπτυξε μια μηχανή η οποία διάβαζε χαρακτήρες και τους μετέτρεπε σε κώδικα μορς. Την ίδια περίπου περίοδο, ο Edmund Fournier d'Albe ανέπτυξε το Optophone, έναν φορητό σαρωτή εικόνας ο οποίος όταν κινούταν κατά μήκος μια εκτυπωμένης γραμμής κειμένου παρήγαγε ηχητικούς τόνους που αντιστοιχούσαν στους χαρακτήρες που συναντούσε.

Ο Goldberg συνέχισε με την ανάπτυξη τεχνολογίας οπτικής αναγνώρισης χαρακτήρων για εισαγωγή δεδομένων. Αργότερα, πρότεινε μια μέθοδο φωτογράφισης δεδομένων τα οποία μετά, με τη χρήση φωτοκύτταρων, αντιστοιχίζονταν σε κάποια πρότυπα αναγνώρισης. Το 1929 ο Gustav Tauschek αποδείχθηκε ότι είχε παρόμοιες ιδέες και εξασφάλισε μια πατέντα οπτικής αναγνώρισης χαρακτήρων στη Γερμανία. Το ίδιο και ο Paul W. Handel, ο οποίος εξασφάλισε κι αυτός μια πατέντα ερασιτεχνίας που αφορούσε μια τεχνική αντιστοίχισης προτύπων στις Ηνωμένες Πολιτείες το 1933 (U.S. Patent 1,915,993). Το 1935 και ο Tauschek εξασφάλισε τελικά αμερικάνικη πατέντα για τη μέθοδό του (U.S. Patent 2,026,329).

Το 1949, ερευνητές της RCA (Radio Corporation of America) εργάστηκαν για πρώτη φορά πάνω σε μια πρώιμη μορφή ενός συστήματος οπτικής αναγνώρισης χαρακτήρων με τη βοήθεια ηλεκτρονικού υπολογιστή για λογαριασμό της Γενικής Γραμματείας Απόστρατων των Ηνωμένων Πολιτειών Αμερικής, το οποίο όμως αντί να μετατρέπει το κείμενο σε κείμενο υπολογιστή είχε ως στόχο να εκφωνεί τα γράμματα. Ήταν δηλαδή μία πρώιμη εφαρμογή text-to-speech, αλλά αποδείχθηκε πολύ ακριβή και ριζοσπαστική για την εποχή της ώστε να καθιερωθεί.

Το 1950 ο David H. Shepard, ένας εκ των κορυφαίων κρυπταναλυτών της Υπηρεσίας Ασφαλείας του Αμερικανικού Στρατού, ασχολήθηκε με το πρόβλημα της μετατροπής εκτυπωμένων μηνυμάτων σε γλώσσα μηχανής για την επεξεργασία των δεδομένων τους από ηλεκτρονικό υπολογιστή και κατασκεύασε μία μηχανή που την ονόμασε «Gismo». Έλαβε πατέντα για την κατασκευή του αυτή το 1953 (U.S. Patent 2,663,758). Ο «Gismo» μπορούσε να διαβάσει 23 από τα 26 γράμματα του λατινικού αλφαβήτου, να καταλάβει κώδικα μορς και μουσικά σύμβολα και να εκφωνήσει και να αντιγράψει εκτυπωμένες σελίδες. Ο Shepard προχώρησε ιδρύοντας την εταιρία IMR (Intelligent Machines Research Corporation) η οποία σε λίγο καιρό ανέπτυξε τα πρώτα συστήματα οπτικής αναγνώρισης χαρακτήρων για εμπορική χρήση.

Το 1955, το πρώτο εμπορικό σύστημα οπτικής αναγνώρισης χαρακτήρων εγκαταστάθηκε στα γραφεία του περιοδικού Reader's Digest και χρησιμοποιήθηκε για να εισάγει αναφορές πωλήσεων σε έναν ηλεκτρονικό υπολογιστή. Μετέτρεπε τις δακτυλογραφημένες αναφορές σε διάτρητες κάρτες τις οποίες αργότερα το προσωπικό του τμήματος συνδρομών έβαζε σ' έναν υπολογιστή που εκτελούσε την επεξεργασία των δεδομένων που αφορούσαν τις 15-20 εκατομμύρια αποστολές εντύπων που εκτελούσαν κάθε χρόνο. Το δεύτερο σύστημα πουλήθηκε στην Standard Oil και χρησιμοποιήθηκε για την ανάγνωση αποδείξεων συναλλαγών πιστωτικών καρτών. Άλλα συστήματα που δημιουργήθηκαν και πουλήθηκαν από την IMR περιελάμβαναν έναν αναγνώστη τιμολογίων για την τηλεφωνική εταιρία Bell του Ohio κι έναν σαρωτή σελίδων για την αμερικάνικη πολεμική αεροπορία ο οποίος χρησιμοποιήθηκε για την τηλεγράφιση δακτυλογραφημένων μηνυμάτων. Τα επόμενα χρόνια ο Shepard πούλησε της πατέντες του σε διάφορες εταιρίες, μεταξύ των οποίων και στην IBM.

Γύρω στο 1965, το Reader's Digest και η RCA συνεργάστηκαν για την κατασκευή ενός αναγνώστη εγγράφων που είχε ως στόχο την ψηφιοποίηση των σειριακών αριθμών κουπονιών του περιοδικού που επιστράφηκαν από διαφημίσεις. Τα γράμματα που χρησιμοποιούνταν στα έγγραφα αυτά είχαν εκτυπωθεί από έναν εκτυπωτή τυμπάνου (drum printer) της ίδιας εταιρίας με μια γραμματοσειρά ειδικά σχεδιασμένη για τους σκοπούς της οπτικής αναγνώρισης (OCR-A font). Μάλιστα ο αναγνώστης αυτός ήταν απ' ευθείας συνδεδεμένος με έναν υπολογιστή RCA 301, έναν από τους πρώτους ηλεκτρονικούς υπολογιστές τεχνολογίας ημιαγωγών (transistor technology). Ένας άλλος ειδικός αναγνώστης που επίσης αναπτύχθηκε σε συνεργασία, τοποθετήθηκε στα γραφεία της αεροπορικής εταιρίας TWA και χρησιμοποιήθηκε για την επεξεργασία του αποθέματος των καρτών επιβίβασης. Ο αναγνώστης αυτός διάβαζε έγγραφα με το ρυθμό των 1500 σελίδων το λεπτό απορρίπτοντας τις σελίδες που δεν μπορούσε να διαβάσει πλήρως. Το προϊόν αυτό μπήκε τελικά στην κανονική γραμμή παραγωγής της RCA όπου και προωθήθηκε εμπορικά τα επόμενα χρόνια για διάφορες εργασίες αντίστοιχου τύπου.

Το αμερικάνικο ταχυδρομείο χρησιμοποιούσε μηχανές οπτικής αναγνώρισης χαρακτήρων ήδη από το 1965 βασιζόμενο σε τεχνολογία του μεγάλου επιστήμονα κι εφευρέτη Jacob Rabinow. Η πρώτη χρήση τεχνολογίας οπτικής αναγνώρισης χαρακτήρων στην Ευρώπη έγινε από το βρετανικό ταχυδρομείο. Το 1965 ξεκίνησε το σχεδιασμό ενός ολόκληρου τραπεζικού συστήματος, το «National Giro» το οποίο θα χρησιμοποιούσε στον πυρήνα των συναλλαγών του τεχνολογία οπτικής αναγνώρισης χαρακτήρων, μια ιδέα ρηξικέλευθη η οποία άλλαξε από τότε τη μορφή των τραπεζικών συναλλαγών στη Βρετανία.

Το καναδικό ταχυδρομείο χρησιμοποιούσε τεχνολογία αυτόματης αναγνώρισης χαρακτήρων από το 1971. Το σύστημα διάβαζε το όνομα και τη

διεύθυνση του παραλήπτη και εκτύπωνε πάνω στο φάκελο έναν γραμμικό κωδικό (bar code) με οδηγίες δρομολόγησης ανάλογα με τον ταχυδρομικό κώδικα του προορισμού. Για να μην συγχέονται οι γραμμικοί κωδικοί με σύμβολα που μπορεί να σχεδίασε ο αποστολέας οπουδήποτε πάνω στο φάκελο η εκτύπωση αυτή γινόταν με ειδικό πορτοκαλί μελάνι το οποίο είχε πολύ υψηλά ανακλαστικά χαρακτηριστικά κάτω από υπεριώδες φως. Από αυτό το στάδιο και πέρα ήταν εύκολο τα γράμματα να ταξινομηθούν και να δρομολογηθούν μηχανικά από κάποιο σύστημα ελέγχου γραμμικών κωδικών.

1.2 Η ΣΗΜΑΣΙΑ ΤΗΣ ΓΙΑ ΤΟΥΣ ΤΥΦΛΟΥΣ

Το 1974 ο Ray Kurzweil ίδρυσε την εταιρία Kurzweil Computer Products Inc και συνέχισε την ανάπτυξη προϊόντων οπτικής αναγνώρισης χαρακτήρων τα οποία λειτουργούσαν ανεξαρτήτως γραμματοσειράς. Αποφάσισε πως το καλύτερο προϊόν αυτής της τεχνολογίας θα ήταν αυτό που θα βοηθούσε τους τυφλούς ανθρώπους να διαβάζουν. Δηλαδή ένα προϊόν με το οποίο ένας ηλεκτρονικός υπολογιστής θα μπορούσε να εκφωνεί κείμενα στους τυφλούς.

Για να υλοποιηθεί μια τέτοια συσκευή ήταν απαραίτητη η ανακάλυψη δύο ακόμα τεχνολογιών. Του επίπεδου σαρωτή CCD (CCD flatbed scanner) και του ηλεκτρονικού συνθέτη φωνής (voice synthesizer). Στις 13 Ιανουαρίου 1976, το τελειωμένο προϊόν αποκαλύφθηκε στο κοινό με μεγάλη δημοσιότητα κατά τη διάρκεια μιας κοινής συνέντευξης τύπου του Kurzweil και της αμερικανικής ένωσης τυφλών.

Το 1978 ξεκίνησε ανοικτά η διάθεση του προγράμματος στην αγορά. Ένας από τους πρώτους αγοραστές του προγράμματος ήταν η εταιρία LexisNexis η οποία το αγόρασε για να ανεβάσει επίσημα έγγραφα και αποκόμματα εφημερίδων στη εκκλαπτόμενη τότε online βάση δεδομένων τους.

Το 1980, ο Kurzweil πούλησε την εταιρία του στη Xerox την οποία ενδιέφερε έντονα εκείνη την περίοδο η περαιτέρω εμπορευματοποίηση της μετατροπής εντύπων σε αρχεία ηλεκτρονικού υπολογιστή. Τελικά η Xerox απεμπόλησε τα δικαιώματα της μεταβιβάζοντας τα στην εταιρία Scansoft, η οποία στη συνέχεια συγχωνεύτηκε με την εταιρία Nuance Communications όπου και παραμένουν μαζί μέχρι και σήμερα.



Τελευταίας τεχνολογίας φορητή συσκευή ανάγνωσης για τους τυφλούς

1.3 Η ΣΗΜΑΣΙΑ ΤΗΣ ΓΙΑ ΤΟΥΣ ΟΡΓΑΝΙΣΜΟΥΣ ΚΑΙ ΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ

Τα πλεονεκτήματα που προσφέρει η χρήση εφαρμογών οπτικής αναγνώρισης χαρακτήρων σε μια επιχείρηση ή έναν οργανισμό είναι πολλά. Οι διαδικασίες που καθίστανται δυνατές, ή που βελτιώνονται, από την ύπαρξη αυτής της τεχνολογίας αναμφισβήτητα αυξάνουν την παραγωγικότητα και την αποδοτικότητα της ανθρώπινης εργασίας, ανεξαρτήτως του πεδίου δραστηριοποίησης της εταιρίας ή του οργανισμού όπου εφαρμόζονται. Σε γενικές γραμμές, οι εφαρμογές οπτικής αναγνώρισης χαρακτήρων λειτουργούν επικουρικά βελτιώνοντας και σε κάποιες περιπτώσεις αποκλειστικά καθιστώντας την εφικτή, την ηλεκτρονική οργάνωση μιας επιχείρησης, κάποια από τα πλεονεκτήματα της οποίας είναι τα εξής:

- Μειωμένα έξοδα λειτουργίας - Ελαχιστοποιούνται τα έξοδα αρχειοθέτησης και μακρόχρονης αποθήκευσης εγγράφων, αφού καθίσταται δυνατό να διατηρούνται ή να παράγονται επιτόπου σε έντυπη μορφή μόνο τα απαραίτητα σε αυτή τη μορφή ζητούμενα από το κράτος και τους πελάτες έγγραφα. Εξαιτίας αυτής της τεχνολογίας έγινε πρακτικά εφικτή και οικονομικά βιώσιμη η μετατροπή ολόκληρου του αρχείου εγγράφων ενός οργανισμού σε ηλεκτρονική μορφή. Ο χειροκίνητος τρόπος εισαγωγής των εγγράφων στο ηλεκτρονικό σύστημα είναι εξ' αρχής αποτρεπτικός αφού για ένα ολόκληρο αρχείο θ' απαιτούσε τεράστια ανθρώπινη προσπάθεια η οποία μεταφράζεται άμεσα σε χρόνο και χρήμα. Υπολογίζεται ότι η ανθρώπινη αναζήτηση σ' ένα μεγάλο φυσικό αρχείο κοστίζει για μια εταιρία μέχρι και 20\$ ανά έγγραφο, εάν το έγγραφο ήταν αποθηκευμένο στη σωστή θέση και 120\$ αν είχε αποθηκευθεί σε λάθος θέση, ενώ κοστίζει περίπου 250\$ η αναπαραγωγή με τρίτους τρόπους, όπως διασταύρωση άλλων εγγράφων κ.τ.λ., ενός χαμένου εγγράφου.
- Αύξηση αποδοτικότητας – Μία από τις περισσότερο επαναλαμβανόμενες και λιγότερο αποδοτικές εργασίες γραφείου σήμερα είναι η εισαγωγή δεδομένων φορμών στα ηλεκτρονικά συστήματα των οργανισμών. Έχει υπολογιστεί στις Ηνωμένες Πολιτείες ότι ο μέσος εργαζόμενος χρειάζεται περίπου 12 λεπτά για να συμπληρώσει μία ηλεκτρονική φόρμα από δεδομένα που διαβάζει επιτόπου σε κάποιο έγγραφο.
- Βελτίωση της προσβασιμότητας των εγγράφων – Εφ' όσων υπάρχουν σε ηλεκτρονική μορφή, τα έγγραφα μπορούν να προσπελαστούν ταυτόχρονα από πολλούς εργαζόμενους, εξαλείφοντας την ανάγκη για φωτοτυπίες. Η ελαχιστοποίηση της χρήσης των φωτοτυπιών έχει και περιβαλλοντικό αντίκτυπο αφού μειώνει την κατανάλωση χαρτιού και μελάνης, όπως και τις εκπομπές διοξειδίου του άνθρακα εξαιτίας της χρήσης των φωτοτυπικών μηχανημάτων.

- Αύξηση παραγωγικότητας - Η ικανότητα άμεσης αναζήτησης στα περιεχόμενα ενός εγγράφου, ιδιαίτερα σε χώρους όπου γίνεται κατά συρροή χρήση εντύπων, μπορεί να είναι καθοριστική. Η ικανότητα ηλεκτρονικής τροποποίησης εγγράφων, αντιγραφής τμημάτων τους και γενικά όλα τα πλεονεκτήματα που πηγάζουν από την ύπαρξη σε ηλεκτρονική μορφή ολόκληρης της βάσης δεδομένων των εγγράφων ενός οργανισμού καθίστανται ικανά εξ' αιτίας της ύπαρξης αυτής της τεχνολογίας.
- Ανάκτηση πολύτιμου χώρου – Οι αναλογικά μεγάλοι χώροι αποθήκευσης εγγράφων καθίστανται πλέον αχρείαστοι, οπότε μπορεί να δημιουργηθεί άμεσα, χωρίς επιπλέον έξοδα, πολύτιμος εργασιακός χώρος. Ένα απλό DVD μπορεί να χωρέσει έγγραφα που σε έντυπη μορφή θα γέμιζαν 15 αρχειοθήκες.
- Αποφυγή καταστροφών – Εξαιτίας της ευκολίας τήρησης αρχείων ανάκτησης (back up), αυξάνονται σημαντικά οι πιθανότητες να μην χαθεί σημαντικό υλικό εξαιτίας μιας καταστροφής όπως πυρκαγιά, πλημμύρα κ.τ.λ. Πλέον με την ανάπτυξη των τεχνολογιών cloud καθίσταται δυνατό η φυσική αποθήκευση του αρχείου να μην γίνεται καν στο χώρο εργασίας αλλά οπουδήποτε μέσα στο internet, όπου και πρακτικά είναι άφθαρτο.
- Ασφάλεια – Εξαιτίας του ηλεκτρονικού τρόπου της πρόσβασης, μετά την αρχική τους είσοδο στο σύστημα και την καταστροφή του φυσικού εγγράφου, είναι δυνατό να ασφαλίσουμε τα ευαίσθητα αρχεία με κωδικούς, ή να επιτρέπουμε την πρόσβαση σε αυτά επιλεκτικά, μόνο στους χρήστες που επιθυμούμε, μειώνοντας τις πιθανότητες να πέσουν εταιρικές πληροφορίες σε λάθος χέρια.
- Συνέπεια – Υπολογίζεται ότι περίπου το 3% με 7,5% των εγγράφων ενός οργανισμού έχει χαθεί, δεν μπορεί να ανακτηθεί, ή είναι αρχειοθετημένο σε λάθος θέση. Εξαιτίας της άμεσης ανταπόκρισης που έχει μια τέτοια οργάνωση σε σχέση με τη χειροκίνητη εκδοχή της κάνει τις επιχειρήσεις που την χρησιμοποιούν να μοιάζουν σ' έναν εξωτερικό παρατηρητή πιο οργανωμένες και επαγγελματικές.

Βεβαίως, η ηλεκτρονική οργάνωση μια επιχείρησης ή ενός οργανισμού ανοίγει το δρόμο σε νέες, μοντέρνες προκλήσεις ασφαλείας, οι οποίες πάντως κατά γενική ομολογία είναι συγκρίσιμες, αν όχι μικρότερες, από τις αντίστοιχες της φυσικής.

1.4 ΕΙΔΗ ΕΦΑΡΜΟΓΩΝ

Η τεχνολογία οπτικής αναγνώρισης χαρακτήρων ενσωματώνεται σε προϊόντα με διάφορες μορφές.

- Desktop εφαρμογές – Τα συστήματα οπτικής αναγνώρισης χαρακτήρων είναι αναλυτικές μηχανές τεχνητής νοημοσύνης (artificial intelligence) που εξετάζουν σειρές χαρακτήρων αντί για ολόκληρες λέξεις ή φράσεις. Στηρίζονται στην ανάλυση του σχήματος των χαρακτήρων και βασίζονται σε λεξικά για να κάνουν τις καλύτερες δυνατές προβλέψεις. Είναι οι πρώτες εφαρμογές που κυκλοφόρησαν ευρέως και χρησιμοποιούνται εδώ και πολλά χρόνια για επαγγελματικές και καθημερινές εργασίες. Αρκετές εφαρμογές τέτοιου είδους συμπεριλαμβάνονται συχνά στο λογισμικό εγκατάστασης συσκευών σάρωσης, εκτύπωσης ή συνδυασμού.
- Online εφαρμογές – Με την ανάπτυξη της πληροφορικής, η χρήση της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων μεταφέρθηκε από έναν υπολογιστή σε μια πληθώρα από πλατφόρμες όπως ο παγκόσμιος ιστός, το διαδικτυακό νέφος (cloud computing) και οι φορητές συσκευές (mobile devices). Μετά από 30 χρόνια ανάπτυξης για επιτραπέζιους υπολογιστές, οι εφαρμογές οπτικής αναγνώρισης χαρακτήρων άρχισαν να προσαρμόζονται στη νέα πραγματικότητα. Η διαδικτυακή οπτική αναγνώριση χαρακτήρων (Web OCR, Online OCR, Web Based OCR) έχει γίνει η νέα τάση του χώρου για την κάλυψη των αναγκών μεγαλύτερων ομάδων χρηστών. Η ευρεία εξάπλωση των ευρυζωνικών συνδέσεων διαδικτύου κατέστησε δυνατή τη δημιουργία εφαρμογών οπτικής αναγνώρισης χαρακτήρων σε αποκλειστικά διαδικτυακή μορφή για επιχειρήσεις και απλούς χρήστες. Από το 2000 και μετά ένας σημαντικός αριθμός νέων εταιριών μπήκε στο παιχνίδι της προσφοράς διαδικτυακών εφαρμογών οπτικής αναγνώρισης χαρακτήρων, με πολλές από αυτές να τις προσφέρουν ακόμα και δωρεάν στους τελικούς χρήστες, αντλώντας τα έσοδα τους από διαφημίσεις.
- Εφαρμογές προσαρμοσμένες στην περίπτωση (application oriented OCR) – Από τη στιγμή που η τεχνολογία οπτικής αναγνώρισης χαρακτήρων απέκτησε ευρεία αποδοχή και άρχισε να χρησιμοποιείται σε πολλές διαφορετικές συνθήκες και έργα, προέκυψε η ανάγκη για συστήματα περισσότερο εξειδικευμένα στις ανάγκες κάποιων πεδίων. Για παράδειγμα, περίπλοκα φόντα σελίδων, φωτογραφίες φθαρμένες, τσαλακωμένες, λερωμένες, χαμηλής ανάλυσης, θολές, κείμενα με ορολογίες που δεν συναντώνται στο τυπικό λεξικό, ειδικές γραμματοσειρές, όλα αυτά είναι συνθήκες που συναντώνται συχνά σε πραγματικά περιβάλλοντα εργασίας και καθιστούν πολύ δύσκολη τη σωστή λειτουργία τυποποιημένων προγραμμάτων γενικής χρήσης.

Έτσι αναπτύχθηκαν εφαρμογές βελτιστοποιημένες για συγκεκριμένες εισόδους όπως τιμολόγια (invoice OCR), επαγγελματικές κάρτες (business-card OCR), στιγμιότυπα οθόνης (screenshot OCR), ταυτότητες (ID card OCR), διπλώματα οδήγησης (driver license OCR) κ.τ.λ.

1.5 ΔΥΝΑΤΟΤΗΤΕΣ ΕΦΑΡΜΟΓΩΝ

Ιδρυμένο από το υπουργείο ενέργειας των Ηνωμένων Πολιτειών Αμερικής (DOE), το ινστιτούτο έρευνας της επιστήμης της πληροφορίας (ISRI) είχε αναλάβει υπό τη σκέπη του την εξέλιξη της τεχνολογίας της αυτόματης κατανόησης εκτυπωμένων κειμένων από τον ηλεκτρονικό υπολογιστή και ήταν το καθ' ύλην αρμόδιο όργανο για τη διεξαγωγή μετρήσεων ακριβείας των λογισμικών οπτικής αναγνώρισης χαρακτήρων από το 1992 μέχρι το 1996.

Η αναγνώριση κειμένου γραμμένου με λατινικούς χαρακτήρες δεν είναι ακόμη και σήμερα 100% ακριβείς, από οποιοδήποτε σύστημα, ακόμη και όταν χρησιμοποιούνται καθαρές εικόνες. Μια μελέτη που χρησιμοποίησε ως δείγματα άρθρα από καλοδιατηρημένες εφημερίδες του 19^{ου} και των αρχών του 20^{ου} αιώνα αποφάνθηκε ότι η επιτυχία αναγνώρισης των εμπορικών προγραμμάτων οπτικής αναγνώρισης χαρακτήρων ποικίλει μεταξύ 71% και 98%. 100% αναγνώριση μπορεί να επιτευχθεί μόνο με τη συμμετοχή ανθρώπου στη διαδικασία. Τομείς όπως η αναγνώριση χειρόγραφων «ρέουσας» γραφής (μονοκονδυλιές - cursive handwriting) ή η αναγνώριση χαρακτήρων ανατολικοασιατικού τύπου (east asian), όπου το κάθε γράμμα μπορεί να αποτελείται από πολλά κομμάτια, αποτελούν ακόμη και σήμερα ενεργό πεδίο έρευνας.

Μετρήσεις ακρίβειας μπορούν να γίνουν με διαφορετικούς τρόπους οι οποίοι επηρεάζουν σημαντικά τα αποτελέσματα. Για παράδειγμα, εάν δεν επιτρέπεται στο προς μέτρηση πρόγραμμα να χρησιμοποιεί λεξικό για τη διόρθωση των λέξεων τις οποίες βρήκε σχεδόν σωστά, επιτυχία αναγνώρισης της τάξης του 95% θα μπορούσε σε πραγματικές συνθήκες να αντιστοιχεί σε επίδοση της τάξης του 99%. Ο λόγος που χρησιμοποιούνται και τέτοιες μετρήσεις είναι ώστε να βγάλουν οι ερευνητές συμπεράσματα για την ποιότητα των επιμέρους μηχανισμών που εμπλέκονται στη διαδικασία της αναγνώρισης. Δηλαδή να γνωρίζουν ποια εφαρμογή διαχειρίζεται πιο άρτια το λεξικό, αλλά να γνωρίζουν και ποια έχει τον ακριβέστερο μηχανισμό «ωμής αναγνώρισης» (raw recognition). Η γνώση από αυτήν την ανάλυση θα μπορούσε κάλλιστα να οδηγήσει στη συνδυαστική κατασκευή ενός ανώτερου προγράμματος το οποίο θα χρησιμοποιούσε τον ισχυρό μηχανισμό ωμής αναγνώρισης μιας εφαρμογής και τις έξυπνες μετά-διορθώσεις μιας άλλης.

Η τεχνολογία απευθείας αναγνώρισης χαρακτήρων (On-line Character Recognition – On-line CR) συχνά συγχέεται με την οπτική αναγνώριση χαρακτήρων, όμως δεν είναι έτσι. Η οπτική αναγνώριση χαρακτήρων, ακόμη και στη διαδικτυακή της μορφή (Online OCR), είναι μια μορφή «στατικής» αναγνώρισης (off-line character recognition), δηλαδή αναγνώρισης του σχήματος ενός στατικού στιγμιότυπου μιας μορφής (fixed static shape), ενώ η απευθείας αναγνώριση χαρακτήρων αναγνωρίζει δυναμικά τη μορφή που

σχηματίζεται, όχι τόσο από το σχήμα, αλλά από τις αλλαγές κατεύθυνσης του μέσου γραφής (dynamic motion during handwriting). Για παράδειγμα, η απευθείας αναγνώριση χαρακτήρων που χρησιμοποιείται στις «χειρονομίες» (gestures) των ταμπλετών (tablet PC) αναγνωρίζει εκτός της μορφής και την κατεύθυνση που τρέχει το στυλό, όπως για παράδειγμα από αριστερά προς τα δεξιά ή το αντίστροφο. Η απευθείας αναγνώριση χαρακτήρων συναντάται συχνά και με άλλες ονομασίες όπως δυναμική αναγνώριση χαρακτήρων (dynamic character recognition), αναγνώριση χαρακτήρων πραγματικού χρόνου (realtime character recognition) ή ευφυής αναγνώριση χαρακτήρων (intelligent character recognition - ICR).

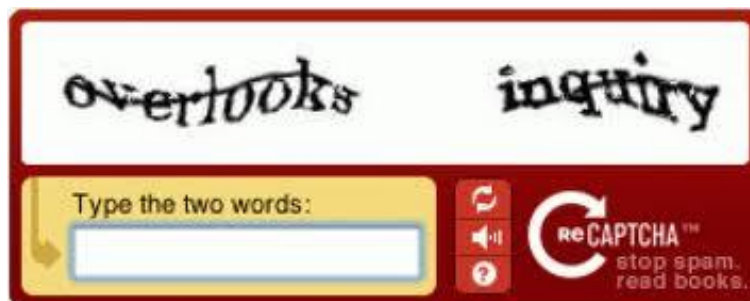
Προγράμματα απευθείας αναγνώρισης χαρακτήρων έχουν γίνει ευρέως γνωστά τα τελευταία χρόνια με την έκρηξη της αγοράς των έξυπνων κινητών τηλεφώνων (smartphones) και των ταμπλετών. Οι αλγόριθμοι που χρησιμοποιούνται σε αυτές τις συσκευές εκμεταλλεύονται το γεγονός ότι τη στιγμή της εισαγωγής των δεδομένων γίνονται γνωστά στη συσκευή η κίνηση, η ταχύτητα και η κατεύθυνση της γραφίδας σχεδίασης και ότι ο χρήστης μπορεί να εκπαιδευτεί να σχεδιάζει τα σύμβολα που περιμένει η εφαρμογή με τον τρόπο που βολεύει την αναγνώριση. Αυτά, δυστυχώς δεν μπορούν να εφαρμοστούν στην αναγνώριση σαρωμένων εγγράφων οπότε η οπτική αναγνώριση χειρόγραφων χαρακτήρων είναι ακόμη ένα αρκετά ανοικτό πρόβλημα. Ακρίβεια της τάξης του 80% με 90% είναι δυνατόν να επιτευχθεί για καθαρούς, καλούς, χειρόγραφους χαρακτήρες κάτι που όμως και πάλι μεταφράζεται σε δεκάδες σφάλματα ανά σελίδα, κάτι που καθιστά μια τέτοια εφαρμογή δύσχρηστη σε πραγματικές συνθήκες εργασίας.

Η αναγνώριση «ρέουσας» γραφής, δηλαδή γραφής όπου η κάθε λέξη γίνεται με μία κίνηση του μέσου γραφής (μονοκονδυλιά), είναι ένα πεδίο έρευνας που βρίσκεται αυτή τη στιγμή στην ακμή του με τυπικά ποσοστά αναγνώρισης αρκετά μικρότερα από την αναγνώριση σαρωμένου χειρόγραφου εγγράφου (το οποίο έχει γραφτεί με διακριτά τα γράμματα – όπως π.χ. όταν συμπληρώνουμε φόρμες με κουτάκια). Σε αυτό τον τύπο γραφής, προτιμάται να εξετάζεται σαν ενιαίο δείγμα ολόκληρη η λέξη, αντί να επιχειρείται η τμηματοποίηση της και εξαγωγή των γραμμάτων της ένα ένα. Επίσης είναι συχνή η χρήση λεξικού προσαρμοσμένου στις ανάγκες της συγκεκριμένης εφαρμογής.

Για παράδειγμα, στο πρόβλημα της ανάγνωσης των στοιχείων μιας επιταγής η βελτίωση της αναγνώρισης που συντελείται με τη χρήση λεξικού είναι πολύ μεγάλη, αφού είναι πολύ λίγες οι λέξεις που μπορεί να γράψει κάποιος πάνω σε μια επιταγή. Η γνώση γραμματικής και συντακτικού από τον αλγόριθμο βοηθά κατακόρυφα στην κατανόηση της «ρέουσας» γραφής γιατί έχει αποδειχθεί ότι αποκλειστικά και μόνο από τη χρήση σχεδιαστικής πληροφορίας δεν είναι δυνατόν να επιτευχθεί ακρίβεια μεγαλύτερη του 98% για τη γραφή ενός μέσου ανθρώπου. Κατά κάποιο τρόπο, μέσος όρος

ακρίβειας της τάξης του 98% είναι το ανώτατο θεωρητικό όριο που μπορεί να επιτύχει ένας τέλειος αλγόριθμος αναγνώρισης «ρέουσας» γραφής χωρίς να χρησιμοποιεί γραμματική και συντακτική πληροφορία. Αυτό συμβαίνει γιατί οι άνθρωποι έχουμε την τάση όταν γράφουμε με «ρέοντα» τρόπο μία λέξη, είτε να μη την ολοκληρώνουμε πλήρως, είτε να τη γράφουμε παραλείποντας γράμματα. Επίσης, είναι συχνό να προσθέτουμε κοσμητικά στοιχεία στους χαρακτήρες (π.χ. τραβηγμένες μεγάλες κυματιστές γραμμές στο τελικό 'ς', μεγάλες γυριστές ουρές που υπερκαλύπτουν τα υπόλοιπα γράμματα στο 'λ' κ.τ.λ.), όπως κι επίσης είναι γενικά αποδεκτό ότι ενώ τα «καλά», διακριτά, γράμματα μας μοιάζουν αρκετά με το πρότυπο, ο γρήγορος «ρέοντας» τρόπος γραφής μας απέχει σημαντικά από άνθρωπο σε άνθρωπο. Σε τέτοιες περιπτώσεις, εάν το πρόγραμμα δε ξέρει τι περίπου να περιμένει είναι αδύνατο να αναγνωρίσει τα πάντα.

Μια τεχνική η οποία έχει αξιοσημείωτη επιτυχία στην αναγνώριση δύσκολων λέξεων, από έγγραφα γενικά ακατάλληλα για αυτόματη αναγνώριση, είναι η αυτόματη αποστολή των μη αναγνωρισμένων από το πρόγραμμα λέξεων σε ανθρώπους για περεταίρω αναγνώριση, όπως γίνεται στο σύστημα reCAPTCHA.



To User Interface του γνωστού συστήματος reCaptcha

1.6 ΑΝΑΓΝΩΡΙΣΗ ΜΕ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Για την οπτική αναγνώριση χαρακτήρων χρησιμοποιούνται κατά κόρων αλγόριθμοι που υλοποιούν διατάξεις τεχνητών νευρωνικών δικτύων (artificial neural networks). Υπάρχουν πολλών ειδών τεχνητά νευρωνικά δίκτυα τα οποία στις πραγματικές εφαρμογές συνήθως συνδυάζονται έχοντας το καθένα δικαίωμα μιας «ζυγισμένης» ψήφου (weighted vote) στο τελικό αποτέλεσμα αλλά όλων η λογική είναι παρεμφερείς και παρουσιάζεται παρακάτω.

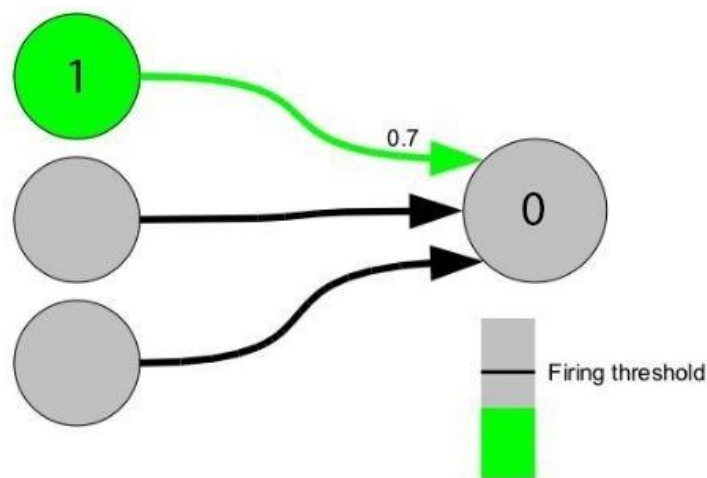
Τα τεχνητά νευρωνικά δίκτυα είναι μαθηματικοί ταξινομητές (classifiers) των οποίων η έμπνευση προήλθε από ένα μοντέλο λειτουργίας των βιολογικών νευρώνων του εγκεφάλου. Οι νευρώνες είναι βιολογικές οντότητες οι οποίες σε λογικό επίπεδο μπορούν να περιγραφούν σαν «πύλες» (logic gates) με από μία έως πολλές εισόδους και μία έξοδο. Οι συνδέσεις των νευρώνων λέγονται συνάψεις. Εκτός από απευθείας εισόδους κι έξοδο, οι νευρώνες μπορούν να έχουν σαν είσοδο την έξοδο ενός άλλου νευρώνα και αυτοί με τη σειρά τους να αποτελούν είσοδο για άλλους, σχηματίζοντας με αυτόν τον τρόπο δίκτυα.

Στα έμβια όντα αυτά τα δίκτυα μπορεί να αποτελούνται από δισεκατομμύρια νευρώνες συνδεδεμένους μεταξύ τους με πολλαπλάσιες συνάψεις. Υπάρχουν ενδείξεις ότι η πολυπλοκότητα του νευρωνικού δικτύου συνδέεται με το νοητικό επίπεδο του οργανισμού που το φέρει. Είναι άλωσε εύλογο αφού ο ανθρώπινος εγκέφαλος έχει το πολυπλοκότερο γνωστό νευρωνικό δίκτυο που αποτελείται κατά μέσο όρο από 100.000.000.000 νευρώνες. Τα σήματα μεταξύ των νευρώνων στον εγκέφαλο, σε αντίθεση με αυτό που πιστεύαμε παλιότερα επηρεασμένοι από τον τρόπο λειτουργίας των ηλεκτρικών κυκλωμάτων, δεν είναι ηλεκτρικά αλλά χημικά.

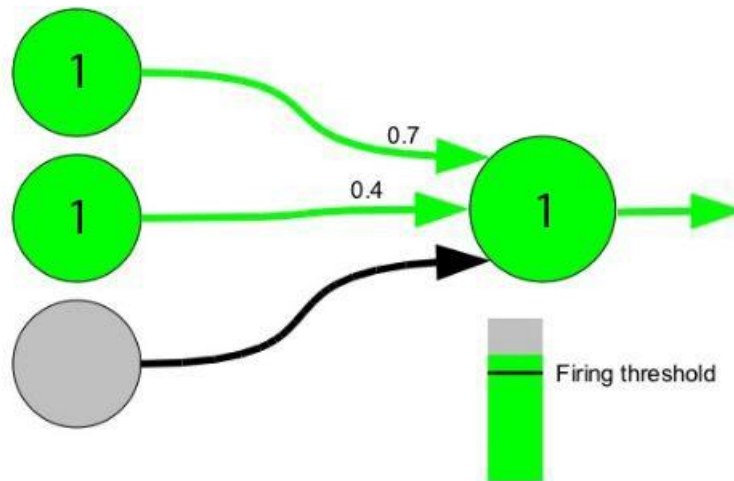
Οι χημικές ουσίες που διακινούνται μεταξύ των νευρώνων ονομάζονται νευροδιαβιβαστές (neurotransmitters). Βεβαίως, οι δυνάμεις που αναπτύσσονται και οι οποίες είναι υπεύθυνες για την εξαπόλυση, τη πρόσληψη, τη μετακίνηση και τη συνοχή αυτών των ουσιών είναι ηλεκτρικές αλλά το νόημα είναι ότι οι νευρώνες δε διακινούν ηλεκτρόνια, όπως τα ηλεκτρικά κυκλώματα, αλλά μόρια χημικών νευροδιαβιβαστών κάποιοι από τους οποίους είναι η σεροτονίνη (serotonin), η νορεπινεφρίνη (norepinephrine), η ντοπαμίνη (dopamine) και άλλοι. Μάλιστα εικάζεται ότι η ελαττωμένη σε σχέση με το φυσιολογικό ποσότητα αυτών των ουσιών στις συνάψεις των νευρώνων είναι η αιτία αρκετών ψυχικών διαταραχών. Αν και κανείς δεν έχει κάνει ως τώρα κάποια μέτρηση που να το αποδεικνύει σ' έναν εν λειτουργία εγκέφαλο έχει αναπτυχθεί ολόκληρη κατηγορία φαρμάκων που έχουν ως στόχο την αύξηση της ποσότητας των νευροδιαβιβαστών που πλέουν μεταξύ των συνάψεων. Η κατηγορία αυτών των φαρμάκων λέγεται «επιλεκτικοί αναστολείς επαναπρόσληψης νευροδιαβιβαστή» (συνήθως

σεροτονίνης ή νορεπινεφρίνης) (Selective Serotonin Reuptake Inhibitor – SSRI and Selective Norepinephrine Reuptake Inhibitor – SNRI) και θεωρούνται το παρόν της ψυχιατρικής καθώς είναι πιο αποτελεσματικά και πολύ πιο ελαφριά σε παρενέργειες από τα φάρμακα παλαιού τύπου που αντικαθιστούν.

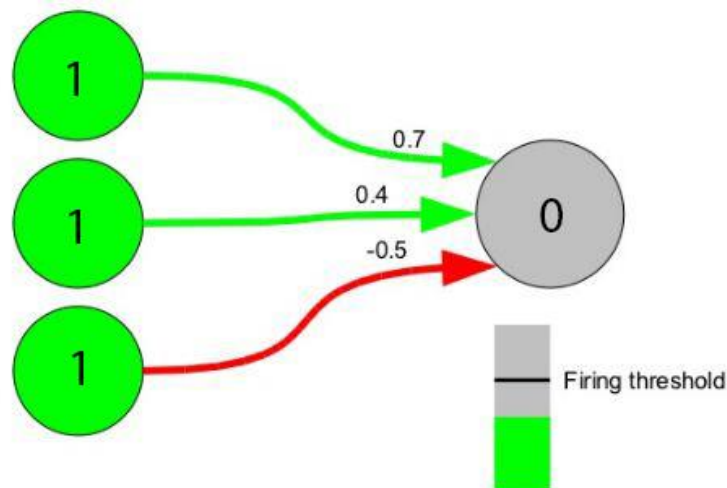
Η λογική λειτουργίας του βιολογικού νευρώνα είναι η εξής. Ο νευρώνας έχει έναν αριθμό εισόδων από τις οποίες δέχεται χημικά σήματα. Τα σήματα αυτά ανάλογα με το από ποια είσοδο (σύναψη) προέρχονται λαμβάνονται υπ' όψιν με διαφορετικό συντελεστή βαρύτητας. Δηλαδή μπορεί ο νευρώνας να είναι πιο ευαίσθητος σε σήματα που προέρχονται για παράδειγμα από τα μάτια και λιγότερο ευαίσθητος σε σήματα που προέρχονται από τη μύτη. Όταν το άθροισμα της έντασης των σημάτων, όπως αυτά υπολογίστηκαν ανάλογα με τα βάρη τους το καθένα, ξεπεράσει μια τιμή που λέγεται «κατώφλι πυροδότησης» (firing threshold) τότε ο νευρώνας ενεργοποιεί την έξοδό του, δηλαδή στέλνει το δικό του χημικό σήμα παρακάτω. Όσο το άθροισμα των σημάτων που λαμβάνει σαν είσοδο είναι κάτω από αυτό το όριο δεν στέλνει κανένα σήμα.



Η είσοδος (1) x το βάρος της (0,7) = 0,7 δεν ξεπερνάει το κατώφλι πυροδότησης



Οι είσοδοι επί το βάρους τους αθροιστικά ξεπερνάνε το κατώφλι πυροδότησης ($1 \times 0,7 + 1 \times 0,4 = 1,1$)



Οι είσοδοι μπορούν να έχουν και αρνητικό βάρος

Το πώς αυτή η διάταξη μπορεί να οικοδομεί λογική γίνεται λίγο πιο προφανές με το παρακάτω παράδειγμα. Έστω ένας νευρώνας που έχει δύο μόνο εισόδους, κατώφλι πυροδότησης 1 και τα σήματα που λαμβάνει στις εισόδους του μπορεί να είναι 0 ή 1. Αν τα βάρη των εισόδων του οριστούν στο 0,5 για να ενεργοποιηθεί η έξοδός του θα πρέπει να ισχύει η ανίσωση:

$$(\text{ΕίσοδοςA} \times 0,5) + (\text{ΕίσοδοςB} \times 0,5) \geq 1$$

Αυτό γίνεται μόνο αν και οι δύο εισόδοι του είναι ενεργοποιημένοι (1), οπότε ένας νευρώνας σε τέτοια «ρύθμιση» λειτουργεί σαν λογική πύλη AND (AND logic gate). Αν στον ίδιο νευρώνα τα βάρη ήταν 1 και 1 τότε ο νευρώνας θα λειτουργούσε σαν λογική πύλη OR (OR logic gate). Αν στη μία εισόδο του δίνουμε πάντα 1 με βάρος 1 και αλλάζουμε το βάρος της άλλης εισόδου σε -1 τότε ο νευρώνας θα λειτουργούσε σαν μια λογική πύλη NOT (NOT logic gate). Είναι αποδεδειγμένο ότι έχοντας σαν πρώτες ύλες αυτές τις τρεις πύλες μπορείς να φτιάξεις οποιαδήποτε διάταξη λογικής.

Φεύγοντας από το ανθρώπινο σώμα κι επιστρέφοντας στο προγραμματιστικό μας περιβάλλον, η ιδέα είναι πως μπορούμε να κατασκευάσουμε μία μαθηματική διάταξη η οποία χρησιμοποιώντας την παραπάνω λογική μπορεί να κατηγοριοποιεί αντικείμενα. Ο παραλληλισμός έχει ως εξής. Έστω ένας αριθμός από αντικείμενα τα οποία ανήκουν σε τρεις κατηγορίες. Στην κατηγορία A, στην κατηγορία B και στην κατηγορία Γ. Έστω ότι τα αντικείμενα αυτά έχουν από 5 χαρακτηριστικά το καθένα. Για παράδειγμα βάρος, ύψος, χρώμα, υφή και ηλικία. Οι συνδυασμοί των χαρακτηριστικών καθορίζουν σε ποια από τις 3 κατηγορίες ανήκει το κάθε αντικείμενο. Να σημειωθεί εδώ ότι προφανώς η ίδια λογική μπορεί να επεκταθεί σε οσοδήποτε κατηγορίες και σε οσαδήποτε χαρακτηριστικά εισόδου.

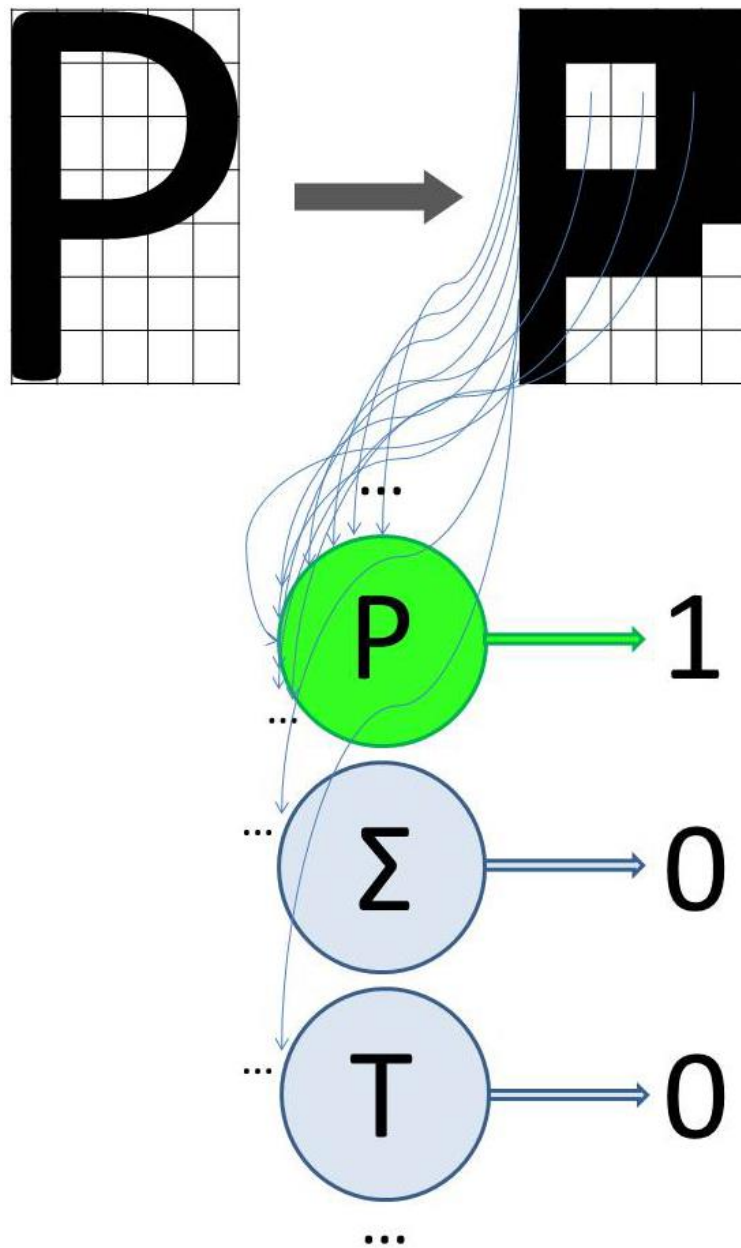
Φτιάχνουμε 3 νευρώνες, όσοι και οι κατηγορίες, και αναθέτουμε στον καθένα αφού εξετάσει όλα τα αντικείμενα να αποφασίσει ποια από αυτά ανήκουν στην κατηγορία που εποπτεύει. Δηλαδή να πάρει σαν εισόδους τις 5 τιμές του κάθε αντικειμένου, να τις «ζυγίσει» ανάλογα με το πόσο θεωρεί ότι συμβάλει η κάθε μία στο αποτέλεσμα και αν το άθροισμα ξεπερνάει την τιμή ενεργοποίησης του να πει, ναι, το αντικείμενο αυτό ανήκει στην κατηγορία που ελέγχω. Η λογική λειτουργεί, το πρόβλημα είναι πώς να γνωρίζει ο νευρώνας τι βαρύτητα έχει η κάθε εισόδος στο αποτέλεσμα. Δηλαδή, μπορεί να παίζει πολύ μεγάλο ρόλο το ύψος στο αν ένα αντικείμενο ανήκει σε μια κατηγορία, ενώ η υφή να μη μετράει καθόλου (για παράδειγμα αν έχει να εξετάσει αν τα αντικείμενα ανήκουν στην κατηγορία «μεγάλα αντικείμενα»).

Τη συμβολή της κάθε εισόδου στο αποτέλεσμα την μαθαίνουν οι νευρώνες με μία διαδικασία που λέγεται εκπαίδευση (training). Μαζεύουμε μια λίστα με αντικείμενα των οποίων ξέρουμε την κατηγορία στην οποία ανήκουν. Τα δίνουμε στο νευρώνα ένα ένα και του ζητάμε να αποφασίσει για όλα αν ανήκουν στην κατηγορία που ελέγχει. Προφανώς στην αρχή που δεν ξέρει τίποτα οι απαντήσεις του θα είναι τυχαίες. Στο τέλος κάθε γύρου ελέγχου όλων των προτύπων το οποίο ονομάζεται εποχή (training epoch), ανάλογα με τι λάθη έκανε στις αποφάσεις του μεταβάλλει λίγο τα βάρη των εισόδων του με τη προσδοκία στον επόμενο γύρο να αποφασίσει σωστά για περισσότερα αντικείμενα. Ξεκινάει ο επόμενος γύρος εκπαίδευσης στον οποίο γίνονται τα ίδια πράγματα. Εάν η μεταβολή των βαρών που είχε κάνει πριν ήταν προς τη σωστή κατεύθυνση και σε αυτό το γύρο βρίσκει σωστά περισσότερα

αντικείμενα τότε συνεχίζει τη μεταβολή των βαρών προς την ίδια κατεύθυνση, αν όχι αλλάζει. Αυτό γίνεται παράλληλα για όλους τους νευρώνες και τερματίζεται μόλις εκπληρωθεί κάποια συνθήκη η οποία συνήθως περιλαμβάνει ένα μέτρο ποιότητας (για παράδειγμα αν βρήκε σωστά πάνω από το 80% των προτύπων) κι ένα μέτρο χρόνου (αν η εκπαίδευση ξεπέρασε τις 1000 εποχές κ.τ.λ.). Αυτό συνιστά τη φάση της εκπαίδευσης. Αξίζει να σημειωθεί ότι υπάρχουν πολλοί διαφορετικοί αλγόριθμοι εκπαίδευσης, άλλοι πιο ακριβείς αλλά χρονοβόροι, άλλοι λιγότερο αλλά γρήγοροι, πάντως όλοι στηρίζονται στην ίδια λογική της «δοκιμής και σφάλματος» (trial-and-error).

Τώρα που οι νευρώνες ξέρουν σε αρκετά καλό βαθμό κατά πόσο να λαμβάνουν υπ' όψιν τους το κάθε χαρακτηριστικό ώστε ν' αποφασίσουν τι ανήκει και τι όχι στην κατηγορία που εξετάζουν, μπορούμε να τους τροφοδοτούμε με άγνωστα αντικείμενα των οποίων ξέρουμε μόνο τα χαρακτηριστικά και να μας λένε με μια καλή προσέγγιση σε ποια κατηγορία πρέπει να ανήκουν. Αυτή η φάση ονομάζεται «ανάκληση» (recall) και είναι στην ουσία ο λόγος για τον οποίο κατασκευάζουμε τέτοια συστήματα. Για να μας λένε τι πρέπει να είναι αυτό που δεν ξέρουμε τι είναι.

Επιστρέφοντας στο πρόβλημα της οπτικής αναγνώρισης χαρακτήρων η παραπάνω λογική εφαρμόζεται ως εξής. Μαζεύουμε φωτογραφίες από το κάθε γράμμα του αλφαβήτου, των ψηφίων ή γενικά όλων των συμβόλων που μας ενδιαφέρει να αναγνωρίζει η εφαρμογή. Η πρώτη επιλογή που έχουμε να κάνουμε είναι με ποιο τρόπο θα μετατρέψουμε μια εικόνα ενός συμβόλου σε χαρακτηριστικά κατάλληλα για είσοδο ενός νευρώνα, όπως αυτά που περιγράψαμε πριν. Συνήθως αυτό γίνεται με τον εξής τρόπο. Φτιάχνουμε ένα ορθογώνιο πλέγμα από κουτάκια, έστω 7x5, και προβάλουμε πάνω του το σύμβολο του οποίου τα χαρακτηριστικά θέλουμε να εξάγουμε. Αξίζει να σημειωθεί ότι σε όσο πιο πολλά κουτάκια επιλέξουμε να χωρίζουμε τον κάθε χαρακτήρα, τόσο πιο ακριβής είναι η ανάκληση αλλά και τόσο πιο πολύ καθυστερεί η διαδικασία της εκπαίδευσης, ειδικά για κάποιους αλγορίθμους. Αν ένα κουτάκι περιέχει μέσα του τμήμα του χαρακτήρα που βρίσκεται από κάτω, τότε λέμε ότι το κουτάκι αυτό έχει την τιμή 1. Αν δεν περιέχει, έχει την τιμή 0. Οι τιμές όλων των κουτακιών αυτού του πλέγματος συνιστούν τα χαρακτηριστικά αυτής της μορφής και εισάγονται ως είσοδοι στους νευρώνες. Οπότε για ένα πρόβλημα στο οποίο θα είχαμε να αναγνωρίσουμε μόνο ελληνικά κεφαλαία γράμματα θα χρειαζόμασταν ένα δίκτυο από 24 νευρώνες, ένας για κάθε γράμμα, 35 εισόδων ο καθένας.



Όλοι οι νευρώνες παίρνουν τα ίδια κουτάκια για είσοδο αλλά μόνο ο P πυροδοτεί

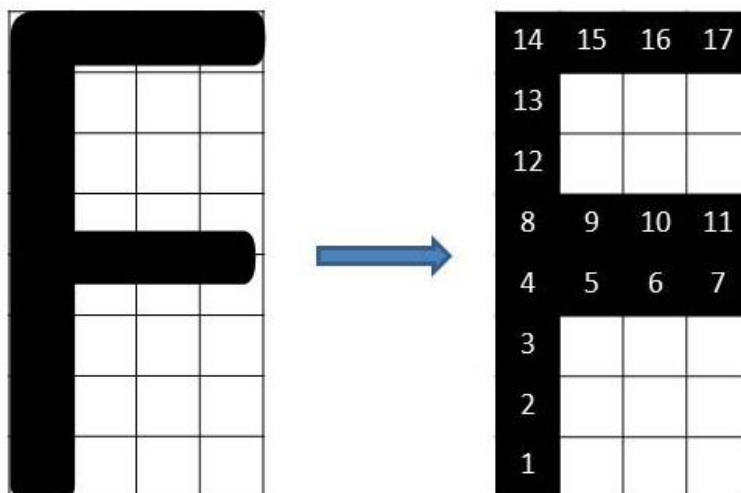
Η εκπαίδευση και η ανάκληση γίνεται με τον ίδιο τρόπο που αναφέραμε παραπάνω. Βέβαια, το παράδειγμα στο οποίο αναφερόμαστε είναι για εκπαιδευτικούς λόγους απλοποιημένο. Οι πραγματικές εφαρμογές όπως είπαμε και προηγουμένως μπορεί να χρησιμοποιούν κατά την ανάκληση λεξικά για να αυξάνουν την ακρίβεια των αποτελεσμάτων, μπορεί να χρησιμοποιούν πολλά παράλληλα δίκτυα με διαφορετική εκπαίδευση το καθένα και να συνδυάζουν τα αποτελέσματά τους, και γενικά να

χρησιμοποιούν πολλών ειδών προσαρμογές και βελτιστοποιήσεις οι οποίες διαφοροποιούν και χαρακτηρίζουν την κάθε εφαρμογή.

1.7 ΑΝΑΓΝΩΡΙΣΗ ΜΕ ΣΤΑΤΙΣΤΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ

Αν και όπως είπαμε η οπτική αναγνώριση χαρακτήρων σχεδόν μονοπωλείται από τα τεχνητά νευρωνικά δίκτυα, έχουν αναπτυχθεί κατά καιρούς και διαφορετικοί μέθοδοι αναγνώρισης, με από κακές έως και καλές επιδόσεις, οι οποίες στηρίζουν τη λειτουργία τους στην εξαγωγή κάποιων στατιστικών παρατηρήσεων πάνω στις μορφές που προσπαθούν να αναγνωρίσουν. Κάποιοι από αυτούς παρουσιάζονται παρακάτω.

Ένας πρώτος, απλοϊκός τρόπος, είναι αφού απομονώσουμε τα γράμματα μιας λέξης να τα βάλουμε ένα ένα σ' ένα ορθογώνιο πλέγμα προκαθορισμένου μεγέθους και να μετρήσουμε τον αριθμό των κουτακιών που καταλαμβάνουν. Η λογική είναι ότι για αρκετές γραμματοσειρές και γι' αρκετά μεγέθη, τυχαίνει κανένας χαρακτήρας να μην σχηματίζεται με τον ακριβώς ίδιο αριθμό εικονοστοιχείων μ' έναν άλλον. Για παράδειγμα, σ' ένα υποθετικό πλέγμα 15x10 το A μπορεί να καταλαμβάνει 63 κουτάκια, το B που είναι πιο «γεμάτο» 69 κουτάκια, ενώ το «λεπτό» I μόλις 24. Έχοντας κρατήσει το σύστημα κατά την εκπαίδευσή του μετρήσεις από όλα τα γράμματα επιχειρεί να αναγνωρίσει μια καινούρια μορφή με τον εξής τρόπο. Βάζει την προς εξέταση μορφή μέσα στο ίδιο πλέγμα που χρησιμοποιήθηκε στην εκπαίδευση και μετράει τον αριθμό των κουτακιών που καταλαμβάνει. Έπειτα, κοιτάζει τη βάση δεδομένων του και βλέπει πιο από τα αποθηκευμένα γράμματα έχει την πλησιέστερη τιμή σε αυτό που μέτρησε τώρα. Με αυτό τον τρόπο αποφασίζει σε ποιο γράμμα αντιστοιχεί η μορφή που μέτρησε. Στο παράδειγμά μας, έστω ότι μετράει μια μορφή και βλέπει ότι αποτελείται από 64 κουτάκια. Αυτή η τιμή, είναι πιο κοντά στην τιμή του A (63) απ' ότι στο αμέσως επόμενο B (96), οπότε αποφασίζει ότι είναι το A.

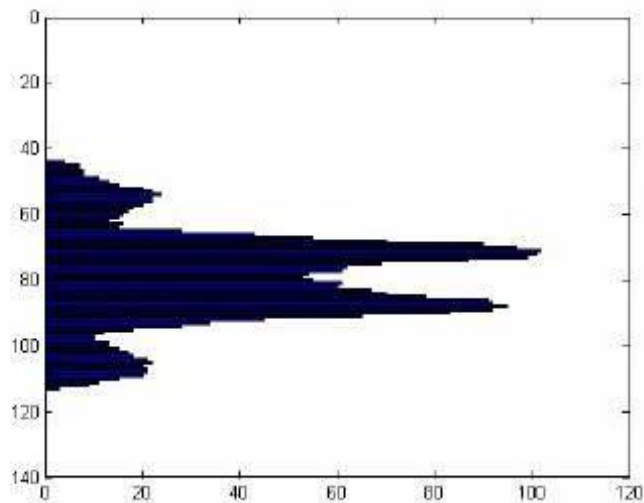
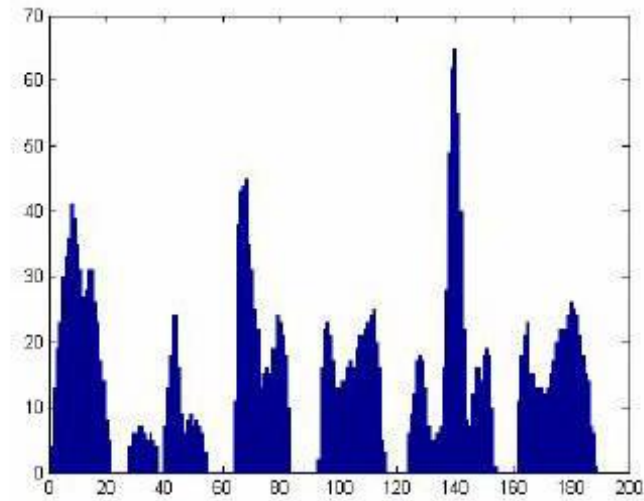
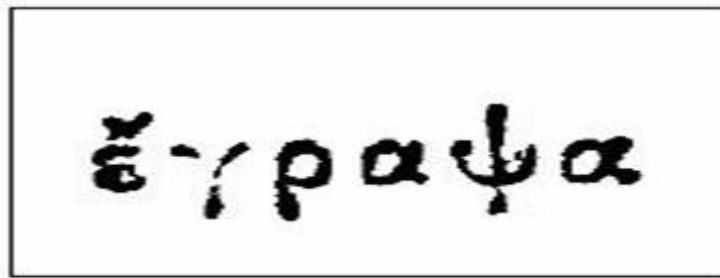


Το γράμμα F καταλαμβάνει 17 κουτάκια

Το πρόβλημα με την παραπάνω διαδικασία, είναι ότι στηρίζεται στην υπόθεση ότι δε θα τύχουν για οποιαδήποτε γραμματοσειρά χαρακτήρες που για το ίδιο μέγεθος θα αποτελούνται από τον ίδιο αριθμό εικονοστοιχείων. Αυτό σε πραγματικές συνθήκες και για μια ευρεία γκάμα γραμματοσειρών δεν συμβαίνει. Υπάρχουν αρκετές γραμματοσειρές για παράδειγμα που τα γράμματα M και W, ή τα 6 και 9 ή άλλα, αποτελούνται από τον ίδιο ακριβώς αριθμό εικονοστοιχείων. Είναι και λογικό, αφού αυτά τα ζευγάρια χαρακτήρων είναι ουσιαστικά αναστροφές της ίδια μορφής. Εμπλουτίζοντας την παραπάνω λογική, έχει προταθεί να χωρίζεται ο κάθε χαρακτήρας στη μέση και να παίρνονται 2 μετρήσεις. Μία για το άνω μισό και μία για το κάτω. Οπότε ο συνδυασμός των δύο αυτών μετρήσεων να αντιπροσωπεύει πιο μονοσήμαντα έναν χαρακτήρα. Ακόμη καλύτερα, χώρισε τον χαρακτήρα σε 4 μέρη και πάρε τέσσερις διαφορετικές μετρήσεις. Η λογική αυτή μπορεί να συνεχιστεί, πάντως έχει μάλλον αποφασιστεί ότι για γενικές συνθήκες δεν μπορεί να αποδώσει ικανοποιητικά κι έτσι εγκαταλείφτηκε για μόνο κάποιες περιπτώσεις αναγνώρισης ειδικών ομάδων χαρακτήρων, σχεδιασμένων ειδικά για την περίπτωση. Το πλεονέκτημα πάντως αυτού του αλγορίθμου είναι η ευκολία των υπολογισμών του και άρα η ταχύτητα εκτέλεσής του σε πιθανές συσκευές και γι' αυτό η λογική των γραμμικών κωδικών (bar codes) βρίσκεται πολύ κοντά σε αυτόν (εκεί μετράμε το πάχος των γραμμών).

Η μόνη (μέχρι σήμερα) βιώσιμη και εμπορικά εφαρμόσιμη λύση αυτής της κατηγορίας αναγνώρισεων είναι μια μέθοδος που βασίζεται στην κατασκευή ιστογραμμάτων (histograms). Μάλιστα, αυτές οι τεχνικές φαίνεται ότι λειτουργούν καλύτερα για ανατολικά σύνολα χαρακτήρων, όπως κινέζικα και γιαπωνέζικα ιδεογράμματα (εκεί που η δυτικού τύπου οπτική αναγνώριση χαρακτήρων δε λειτουργεί και πολύ καλά) και γι αυτό στα μέρη εκείνα κυκλοφορούν κάποια εμπορικά προϊόντα που βασίζονται σ' αυτές τις τεχνικές.

Η λογική του αλγορίθμου είναι η εξής. Παίρνουμε τον χαρακτήρα που θέλουμε να ελέγξουμε και «πατικώσουμε» τα pixels του από πάνω προς τα κάτω. Ανάλογα με το πλήθος των pixels που έχει ο κάθε χαρακτήρας σε κάθε νοητή κάθετο που περνάει από μέσα του σχηματίζεται στη βάση ένα ιστόγραμμα το οποίο είναι χαρακτηριστικό του. Έτσι, αφού έχουμε αποθηκεύσει ιστογράμματα από όλους τους χαρακτήρες που μας ενδιαφέρει να αναγνωρίζει το σύστημά μας, μπορούμε να εξετάζουμε άγνωστους χαρακτήρες κατασκευάζοντας και συγκρίνοντας το ιστόγραμμα τους με τα' αποθηκευμένα. Για μεγαλύτερη ακρίβεια, η μέθοδος αυτή μπορεί να επεκταθεί και στη λήψη δύο ιστογραμμάτων από τον κάθε χαρακτήρα, ένα στον οριζόντιο κι ένα στον κάθετο άξονα.



Κάθετο και οριζόντιο ιστόγραμμα της λέξης «έγραψα»

Ένα εν γένει πλεονέκτημα αυτής της μεθόδου, πέραν της απλότητας των υπολογισμών της, χαρακτηριστικό της οικογένειας των στατιστικών μεθόδων αναγνώρισης χαρακτήρων, είναι ότι παρέχει έναν ευθύ και άμεσο τρόπο,

χωρίς την ανάγκη περεταίρω υπολογισμών, διαχωρισμού των χαρακτήρων μέσα σε μια λέξη ή μια πρόταση. Λαμβάνοντας μία μέτρηση στο μήκος μιας ολόκληρης γραμμής, σε εκτυπωμένους μη επικαλυπτόμενους χαρακτήρες τουλάχιστον, είναι εύκολο να βρεις τα σημεία διαχωρισμού των χαρακτήρων ή των λέξεων αναζητώντας τα μηδενικά σημεία του ιστογράμματος. Γι αυτό το λόγο συνήθως η δειγματοληψία γίνεται μαζικά, σ' επίπεδο γραμμής, και όχι μεμονωμένα καθώς δεν υπάρχει κάποιο ιδιαίτερο πλεονέκτημα από αυτό (τουλάχιστον σε πρώτο στάδιο, σε περίπτωση που χρειάζεται περεταίρω έρευνα για την αποκάλυψη της ταυτότητας κάποιου χαρακτήρα μπορεί μεμονωμένα το πρόγραμμα να παίρνει και κάθετο ιστόγραμμα από τον αμφιλεγόμενο χαρακτήρα).

2 ΠΕΡΙΓΡΑΦΗ ΕΝΟΣ ΝΕΟΥ ΑΛΓΟΡΙΘΜΟΥ

ΕΙΣΑΓΩΓΗ

Τους πρώτους μήνες μετά την ανάθεση της πτυχιακής μου εργασίας, αναλώθηκα σε σκέψεις προσπαθώντας να σκεφτώ έναν αλγόριθμο ο οποίος θα μπορούσε ν' αναγνωρίζει τα γράμματα ενός κειμένου, όποια κατεύθυνση κι αν έχουν. Δεν είναι ότι το απαιτούσε η διαδικασία, έτσι κι αλλιώς οι πινακίδες των αυτοκινήτων είναι πάντα ίσιες, παρ' όλ' αυτά δεν ένοιωθα πως άξιζε η σε βάθος ενασχόληση με κάτι το οποίο δεν αποτελούσε, για τα δικά μου μέτρα, ολοκληρωμένη λύση.

Έχοντας παρακολουθήσει το μάθημα των τεχνητών νευρωνικών δικτύων που προσφέρεται στη σχολή, γνώριζα πως χρησιμοποιούνται τα τεχνητά νευρωνικά δίκτυα για την οπτική αναγνώριση χαρακτήρων. Αυτό που δεν γνώριζα, είναι η σχεδόν καθολική εφαρμογή που έχουν στο χώρο. Ήλπιζα, ότι με την έρευνα γύρω από το θέμα θα ερχόμουν κάποια στιγμή αντιμέτωπος με τον αλγόριθμο τον οποίο έψαχνα. Άρχισα να ψάχνω λεπτομέρειες γύρω από εμπορικές εφαρμογές οπτικής αναγνώρισης χαρακτήρων και αυτόματης αναγνώρισης πινακίδων κυκλοφορίας αλλά δεν έβρισκα κάτι το διαφορετικό. Όλων η λειτουργία, λίγο πολύ, βασιζόταν στα τεχνητά νευρωνικά δίκτυα.

Η αναγνώριση με χρήση τεχνητών νευρωνικών δικτύων έχει αποδειχτεί ότι μπορεί να είναι πολύ καλή, έχει όμως ένα βασικό μειονέκτημα. Αυτό, είναι ότι για να λειτουργήσει, πρέπει η μορφή προς αναγνώριση να έχει διεύθυνση ίδια με αυτήν του προτύπου, βάσει του οποίου εκπαιδεύτηκε το τεχνητό νευρωνικό δίκτυο. Με λίγα λόγια, ένα τεχνητό νευρωνικό δίκτυο δεν μπορεί να αναγνωρίσει έναν πλήρως γνωστό και σωστά κατασκευασμένο χαρακτήρα, εάν αυτός εισαχθεί στο σύστημα περιστρεμμένος. Για παράδειγμα, μπορεί ένα σύστημα ν' αναγνωρίζει τον χαρακτήρα 'Η' εάν αυτός εισαχθεί κανονικά, να μη τον αναγνωρίζει όμως εάν εισαχθεί περιστρεμμένος κατά 90 μοίρες. Αυτό συμβαίνει εξ' αιτίας του τρόπου μετατροπής του σχήματος ενός χαρακτήρα σε δεδομένα εισόδου για το νευρωνικό. Όπως είδαμε και στο κεφάλαιο 1.6, αυτό γίνεται τοποθετώντας τη μορφή εισόδου πάνω σ' ένα πλέγμα από τετράγωνα, και εξάγοντας τα κουτάκια τα οποία αυτή ενεργοποιεί. Όπως είναι λογικό, ο χαρακτήρας 'Η' εάν τοποθετηθεί ίσια ενεργοποιεί κάποια τετράγωνα, αν όμως τοποθετηθεί πλάγια, ακόμη και αν πρόκειται για τον ίδιο ακριβώς χαρακτήρα, ενεργοποιεί άλλα.

	1	2	3	4	5
A	1	0	0	0	1
B	1	0	0	0	1
Γ	1	1	1	1	1
Δ	1	0	0	0	1
E	1	0	0	0	1

	1	2	3	4	5
A	1	1	1	1	1
B	0	0	1	0	0
Γ	0	0	1	0	0
Δ	0	0	1	0	0
E	1	1	1	1	1

Αν και πρόκειται για τον ίδιο χαρακτήρα περιστρεμμένο, ενεργοποιούνται διαφορετικά τετράγωνα

Βεβαίως το πρόβλημα αυτό επιλύεται από τους μηχανικούς με διάφορους τρόπους, αλλά αυτοί είναι «πλάγιοι» με την έννοια ότι δεν είναι πως κατασκευάζεται το τεχνητό νευρωνικό δίκτυο με τέτοιο τρόπο ώστε να μην πέφτει σε αυτήν την παγίδα, απλά χρησιμοποιούνται διάφορες τεχνικές περιστροφής της εικόνας ώστε τελικά να δοθεί στο δίκτυο ο χαρακτήρας ευθυγραμμισμένος, με τη σωστή κατεύθυνση. Αναμφισβήτητα κάτι τέτοιο δουλεύει, όμως ήταν και κάτι με το οποίο δεν ήθελα να ασχοληθώ.

Μου αρέσει όταν σκέφτομαι προς την επίλυση ενός προβλήματος το οποίο έχει φυσικό ανάλογο, να αναλογίζομαι πως μπορεί να λύνεται αυτό ασυνείδητα από τον ανθρώπινο εγκέφαλο. Ένας άνθρωπος, και σίγουρα και κάποια άλλα ζώα, εφ' όσων τον είχαν διδαχθεί, θα αναγνώριζαν τον χαρακτήρα 'H' με όποια κατεύθυνση και αν τους δινόταν αυτός. Εφ' όσων γνώριζαν πως είναι η μορφή ενός αυτοκινήτου, θα αναγνώριζαν ένα, ακόμη και ανάποδα. Καθ' ότι κανένας δεν γνωρίζει επαρκώς το τι ακριβώς συμβαίνει μέσα στον εγκέφαλο δεν μπορώ ν' αποκλείσω το ενδεχόμενο πράγματι, ασυνείδητα, και οι άνθρωποι να ευθυγραμμίζουμε την εικόνα που βλέπουμε πρώτου την αναλογιστούμε. Σε αυτό συνηγορεί και το γεγονός ότι αναγνωρίζουμε με κάποια καθυστέρηση αντικείμενα τα οποία δεν έχουν την στατιστικά αναμενόμενη διεύθυνσή τους (για παράδειγμα μπορούμε να αναγνωρίσουμε ανάποδα γράμματα, αλλά αργούμε λίγο να τα συνειδητοποιήσουμε). Πεποίθησή μου πάντως είναι ότι δε συμβαίνει κάτι τέτοιο, και ότι στη φύση υπάρχει ένας αλγόριθμος ικανός να ταυτοποιεί τις μορφές ανεξαρτήτως κατεύθυνσης.



Θα αναγνωρίζατε το αυτοκίνητο ακόμη και ανάποδα

Πέρασαν αρκετοί άγονοι μήνες κατά τους οποίους όσο και να το σκεφτόμουν δεν μου ερχόταν καμία καλή ιδέα. Αυτό με οδήγησε, μη θέλοντας ν' ασχοληθώ με ένα ακόμη τεχνητό νευρωνικό δίκτυο, στη σκέψη να εγκαταλείψω την εργασία αυτή και να προσπαθήσω να αναλάβω, με ότι συνεπάγεται αυτό για την πορεία των σπουδών μου, μια άλλη το επόμενο εξάμηνο. Ευτυχώς όλα άλλαξαν μια καλοκαιρινή μέρα του 2012, όταν μου ήρθε μία ιδέα η οποία έμοιαζε να έχει προοπτικές. Ασφαλώς και δεν ισχυρίζομαι ότι ο αλγόριθμος που προτείνεται παρακάτω είναι αυτός που υλοποιείται στη φύση, για να είμαι και ειλικρινής νομίζω ότι είναι αρκετά «τεχνικός» για να είναι ο σωστός, πάντως είναι ένας αλγόριθμος αναγνώρισης μορφών ο οποίος είναι ικανός, από τον πυρήνα της λειτουργίας του, να εξάγει αποτελέσματα ανεξαρτήτου κατεύθυνσης.

Αξίζει να σημειωθεί ότι η παρούσα βιβλιογραφική έρευνα έγινε εκ των υστέρων, μετά τη σύλληψη του προτεινόμενου αλγορίθμου και μετά την ολοκλήρωση της εφαρμογής αυτόματης αναγνώρισης πινακίδων κυκλοφορίας που τον υλοποιεί, ως απαραίτητο προαπαιτούμενο της επιτυχούς ολοκλήρωσης της πτυχιακής μου εργασίας. Ως εκ τούτου, δεν γνώριζα κατά την ανάπτυξη του αλγορίθμου μου καμία άλλη μέθοδο οπτική αναγνώρισης χαρακτήρων πέραν αυτής που χρησιμοποιεί τεχνητά νευρωνικά δίκτυα. Πλέον, γνωρίζω πως υπάρχουν και οι στατιστικές μέθοδοι αναγνώρισης, στις οποίες θα κατέτασσα και η δική μου, πάντως ακόμη και μετά την ενασχόλησή μου με αυτές για της ανάγκες αυτού του κειμένου, δεν έχω συναντήσει κάποιον αλγόριθμο όμοιο με τον δικό μου σε τέτοιο βαθμό που να με κάνει να θεωρήσω ότι προϋπάρχει. Βέβαια, και μόνο το γεγονός ότι τον επινόησα εγώ και ότι δεν είναι και κάτι το εξαιρετικά ευφυές με κάνει να έχω αρκετές αμφιβολίες επί της πρωτοτυπίας της ιδέας, πάντως δεν έχω, μέχρι τη στιγμή που γράφονται αυτές οι γραμμές, ένδειξη για το αντίθετο.

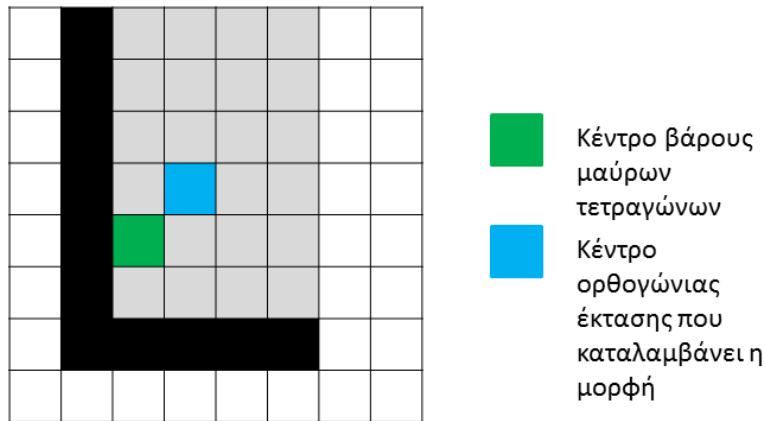
2.1 Ο ΑΛΓΟΡΙΘΜΟΣ

Έστω ως δεδομένο εισόδου μία μορφή η οποία απομονώθηκε με κάποιο τρόπο και παραδίδεται στον αλγόριθμο ως πλέγμα από ενεργά (μαύρα, 1) και μη ενεργά (άσπρα, 0) τετράγωνα κουτάκια.

Πυρήνας του αλγορίθμου είναι κάτι που ονομάζω «περιστροφική πολλαπλή δειγματοληψία». Δηλαδή η διαδοχική λήψη δειγμάτων υπό τη μορφή ευθύγραμμων τμημάτων που ξεκινούν από το κέντρο της αντικείμενης μορφής με κατεύθυνση προς τα έξω, σε περιστροφικό μοτίβο εύρους ενός πλήρους κύκλου.

Αυτό, σημαίνει ότι ο αλγόριθμος πραγματοποιεί την εξέταση σαν ένας διαβήτης. Όπως θα κάναμε και σ' έναν πραγματικό διαβήτη, πρέπει να επιλέξουμε τον άξονα περιστροφής και το άνοιγμα. Μπορεί κάποιος να σκεφτεί διάφορα πιθανά σημεία για την τοποθέτηση του άξονα περιστροφής, όμως στην παρούσα φάση προτείνω τρία. Το πρώτο είναι το κέντρο βάρους της μορφής, το δεύτερο είναι το κέντρο ενός υποθετικού τετραγώνου (ή οποιουδήποτε κανονικού πολυγώνου ή κύκλου) το οποίο την χωράει ακριβώς και το τρίτο είναι ο συνδυασμός αυτών των δύο.

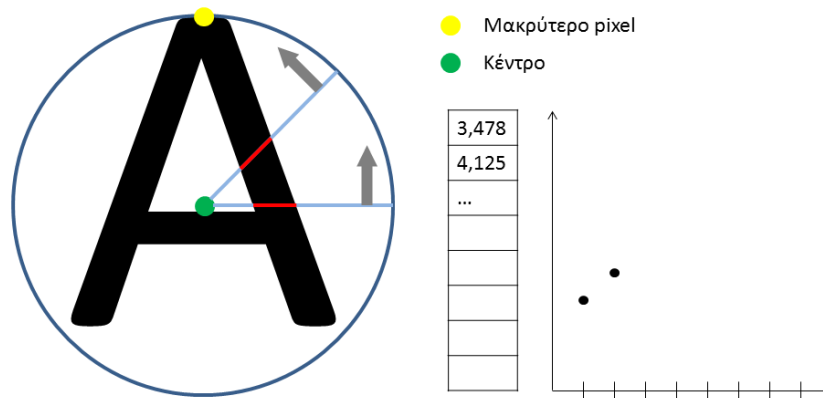
Η θέσπιση του κέντρου βάρους της μορφής ως άξονα περιστροφής καθιστά το μοντέλο αναγνώρισης περισσότερο ικανό στις περιπτώσεις μορφών οι οποίες αποκλίνουν από το πρότυπο ως προς την υπερέκταση των γραμμών τους (σύνηθες κοσμητικό στοιχείο), ενώ αυτή του κέντρου ενός υποθετικού τετραγώνου που την χωράει ακριβώς στις περιπτώσεις μορφών αποτελούμενων από γραμμές ανομοιόμορφου πάχους και σβησμένων σημείων (για παράδειγμα σε γράμματα τα οποία έχουμε ξαναπατήσει ένα κομμάτι τους από πάνω για να φαίνεται καλύτερα). Ο συνδυασμός και των δύο πιθανολογώ ότι λειτουργεί καλύτερα σε περιπτώσεις γενικής χρήσης, όπου δεν γνωρίζουμε εξ αρχής τι μορφής είσοδο να περιμένουμε.



Δύο από τα πιθανά κέντρα της μορφής

Αφού επιλέξουμε το τετράγωνο που θα λειτουργεί σαν άξονας περιστροφής, βρίσκουμε το τετράγωνο το οποίο απέχει πιο πολύ από αυτό και μετράμε την απόστασή του. Σαν άνοιγμα του διαβήτη ορίζουμε αυτή την απόσταση. Να σημειώσουμε ότι για χάριν γενικότητας οι αποστάσεις από εδώ και στο εξής θα εκφράζονται σε «πλευρές στοιχειώδους τετραγώνου», όπου στοιχειώδες τετράγωνο είναι το κάθε τετράγωνο του πλέγματος πάνω στο οποίο είναι αποτυπωμένη η μορφή.

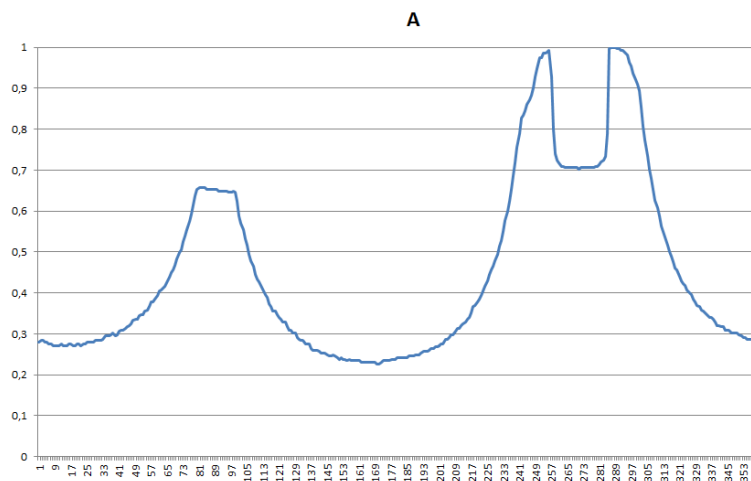
Τοποθετούμε τον διαβήτη στον άξονα περιστροφής και με άνοιγμα όσες πλευρές στοιχειωδών τετραγώνων ορίσαμε πριν, διαγράφουμε έναν πλήρη κύκλο. Οι ακτίνες αυτού του κύκλου ονομάζονται «δείγματα» και πάνω σε αυτές γίνονται όλες οι μετρήσεις της προς ανάγνωση μορφής. Ξεκινάμε από όποια ακτίνα του κύκλου θέλουμε (αυτό είναι το στοιχείο που καθιστά τον αλγόριθμο ανεξάρτητο κατεύθυνσης) και εκτελούμε πάνω της μία σειρά μετρήσεων. Στο παρόν παράδειγμα θα χρησιμοποιήσουμε τον υπολογισμό της μέσης απόστασης των ενεργών τετραγώνων (μαύρα, 1) του δείγματος από τον άξονα περιστροφής. Κρατάμε το αποτέλεσμα στην άκρη (σ' έναν πίνακα) και προχωρούμε με τη χάραξη νέου δείγματος. Η νέα ακτίνα στην οποία θα κάνουμε την επόμενη μέτρηση πρέπει να απέχει (με όποια φορά θέλουμε αρκεί να τηρείται το ίδιο σε όλο τον αλγόριθμο) τόσες μοίρες όσες απαιτούνται για να ληφθούν μέχρι το πέρασ ενός πλήρους κύκλου όσα ισαπέχοντα δείγματα θεωρούμε ότι είναι αρκετά.



Η περιστροφική πολλαπλή δειγματοληψία εν δράσει

Όσο πιο πολλά δείγματα επιλέξουμε να πάρουμε (σε όσες πιο πολλές φέτες επιλέξουμε να κόψουμε την κάθε μορφή), τόσο πιο «ανάγλυφα» αντιλαμβάνεται ο αλγόριθμος το σχήμα της μορφής που εξετάζουμε, κάτι που καθιστά τη λειτουργία του δυνητικά ανώτερη. Βέβαια, η λεπτομέρεια της δειγματοληψίας θα πρέπει να συμβαδίζει με αυτήν του πλέγματος πάνω στο οποίο είναι αποτυπωμένη η μορφή, καθώς η λήψη δειγμάτων πολύ κοντινών για την ανάλυση του πλέγματος αναφοράς, οδηγεί στο φαινόμενο διαδοχικών δειγμάτων μηδενικής ή σχεδόν μηδενικής διαφοροποίησης, κάτι που δε συνεισφέρει στο αποτέλεσμα.

Αφού πάρουμε μετρήσεις διαδοχικά από όλες τις επιλαχούσες ακτίνες, κανονικοποιούμε τα αποτελέσματα σε ένα προαποφασισμένο εύρος (για παράδειγμα $[0,1]$). Ο λόγος που γίνεται αυτή η κανονικοποίηση είναι ώστε να είναι συγκρίσιμα μεταξύ τους αποτελέσματα που προέρχονται από μορφές διαφορετικών μεγεθών και αναλύσεων πλεγμάτων αναφοράς. Αυτή η διαδικασία είναι που εξασφαλίζει ότι ένα μικρό 'Α' κι ένα μεγάλο 'Α' θα έχουν ακριβώς τις ίδιες μετρήσεις. Στη συνέχεια, κατασκευάζουμε ένα γράφημα προβάλλοντας τις μετρήσεις στο χρόνο. Η γραφική παράσταση που σχηματίζεται είναι η «ταυτότητα», κι ελπίζουμε ότι είναι «ικανά χαρακτηριστική», της μορφής που εξετάστηκε.



Η γραφική παράσταση του 'Α' κεντραρισμένου κατά κέντρο βάρους με τη λήψη 360 δειγμάτων

Περνώντας από αυτή την διαδικασία όλους τους χαρακτήρες που μας ενδιαφέρει να αναγνωρίζει ο αλγόριθμος, κατασκευάζουμε ένα αρχείο εκπαίδευσης το οποίο περιέχει την περιγραφή όλων των καμπύλων (τα σημεία) των μορφών με τις οποίες εκπαιδεύσαμε τον αλγόριθμο. Αυτό συνιστά τη διαδικασία της εκπαίδευσης.

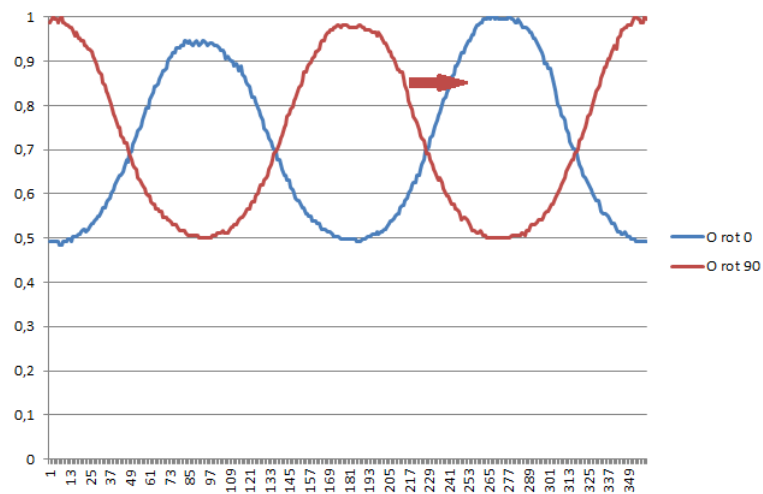
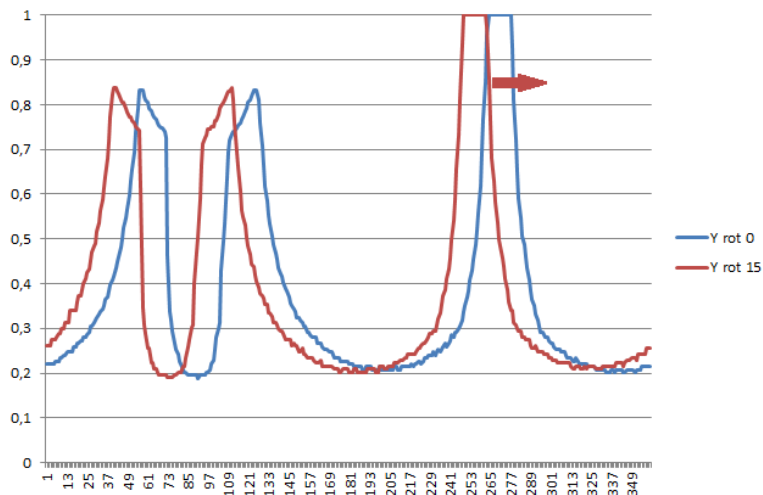
Η διαδικασία της ανάκλησης (αναγνώρισης) λειτουργεί με τη σύγκριση της καμπύλης της προς αναγνώριση μορφής με τις καμπύλες όλων των γνωστών μορφών. Ο τρόπος με τον οποίο γίνεται αυτή η σύγκριση είναι που δίνει στον αλγόριθμο τον χαρακτηρισμό «ανεξάρτητος κατεύθυνσης». Βεβαίως αυτό γίνεται με αλγοριθμικό τρόπο, η οπτική μεταφορά του οποίου είναι η εξής.

Παίρνουμε την καμπύλη της μορφής που θέλουμε να αναγνωρίσουμε και την προβάλουμε, σε αντιπαράβολή, πάνω από τη καμπύλη ενός γνωστού χαρακτήρα. Μετά, συγκρίνουμε τις δύο καμπύλες σημείο σημείο και αθροίζουμε τις διαφορές τους. Στο τέλος, διαιρούμε αυτό το άθροισμα με τον αριθμό των σημείων από τα οποία αποτελείται η κάθε καμπύλη (αριθμός δειγμάτων). Το αποτέλεσμα που προκύπτει είναι ένας αριθμός από το 0,0 μέχρι το 1,0 όπου το 0,0 σημαίνει απόλυτη ταύτιση των δύο καμπυλών ενώ το 1,0 πλήρη διαφωνία. Μαθηματικά αυτό μπορεί να συμβολιστεί ως εξής:

$$\text{Διαφοροποίηση}_{(AB)} = \frac{\sum_{i=1}^n |A_i - B_i|}{n}$$

Όπου A και B οι δύο καμπύλες, n ο αριθμός των σημείων από τα οποία αποτελείται η κάθε καμπύλη και i ο αύξον αριθμός του σημείου.

Κρατάμε αυτό το αποτέλεσμα και μετατοπίζουμε τη μία καμπύλη ένα σημείο πιο δίπλα (δεν παίζει ρόλο ποια καμπύλη θα είναι αυτή ή αν θα είναι αριστερά ή δεξιά, αρκεί να διατηρούμε το ίδιο για όλο τον αλγόριθμο), φέρνοντας την τιμή που περισσεύει ξανά μπροστά. Έπειτα υπολογίζουμε πάλι τη διαφοροποίηση των δύο καμπυλών. Κρατάμε το αποτέλεσμα και συνεχίζουμε τη μετάθεση της καμπύλης που μετακινήσαμε και πριν, ακόμα ένα σημείο (φέρνοντας πάλι ό,τι περισσεύει μπροστά). Όπως καταλάβατε, αυτό γίνεται μέχρι να έρθουν πάλι όλα τα σημεία στην αρχική τους θέση και είναι κάτι το οποίο αντιστοιχεί στην περιστροφική εξέταση του χαρακτήρα κατά 360 μοίρες. Η ελάχιστη μεταξύ των διαφοροποιήσεων που βρήκαμε είναι και η βέλτιστη ταύτιση μεταξύ της εξεταζόμενης μορφής και του γνωστού πρότυπου.



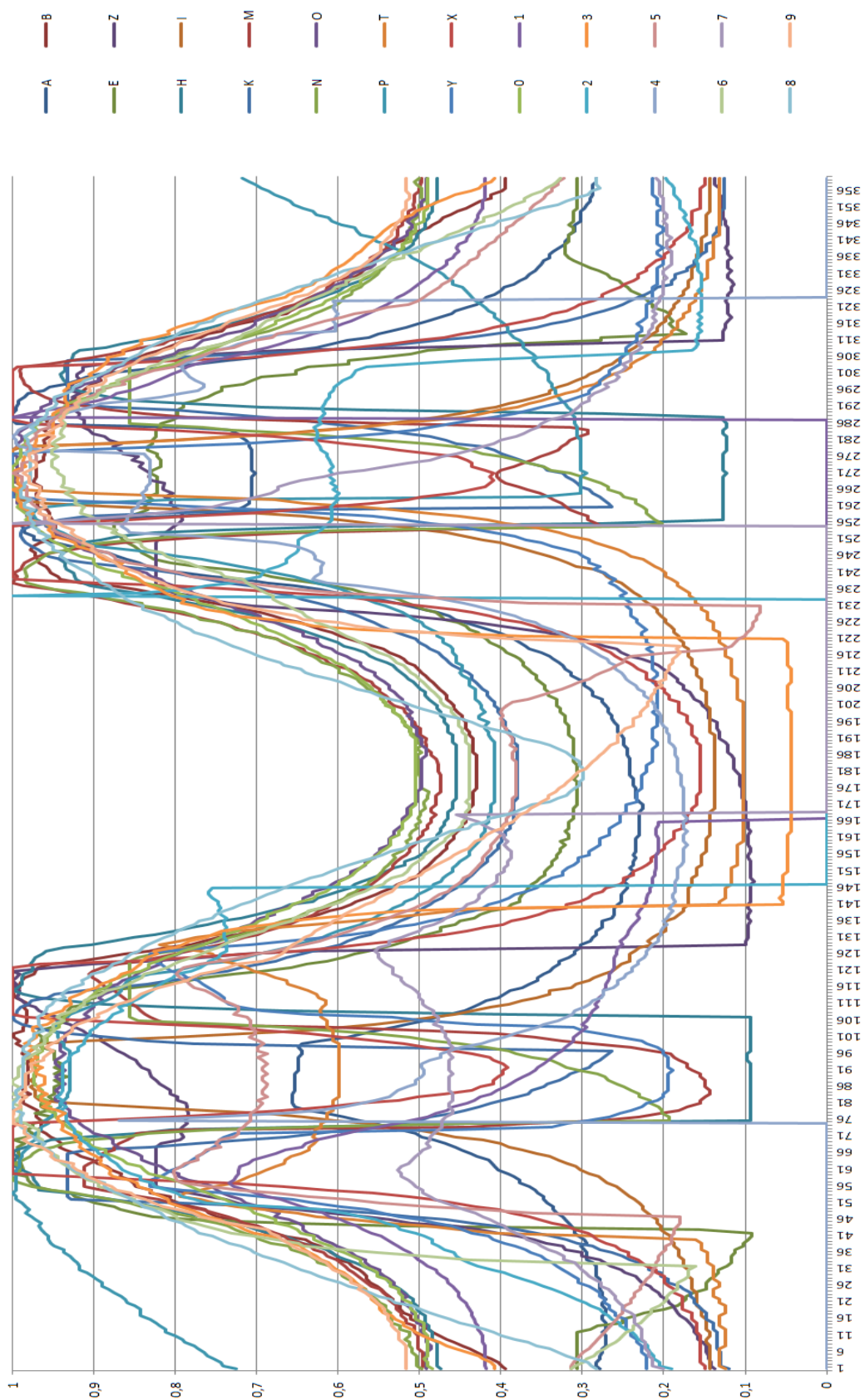
Μετακινούμε τη μία καμπύλη αριστερά-δεξιά μέχρι να βρούμε τη βέλτιστη ταύτιση

Αφού πραγματοποιήσουμε την ίδια διαδικασία, συγκρίνοντας την εξεταζόμενη μορφή με ένα ένα όλα τα πρότυπα που έχουμε αποθηκευμένα, επιλέγουμε την ελάχιστη διαφοροποίηση μεταξύ των επιμέρους διαφοροποιήσεων που έχει με κάθε πρότυπο και την ονοματίζουμε ως τέτοια. Επίσης, εάν κατά τη διαδικασία διατηρήσουμε και το μέτρο της μετατόπισης που εφαρμόσαμε στην καλύτερη ταυτοποίηση, μπορούμε να υπολογίσουμε την κλίση του αναγνωρισμένου χαρακτήρα σε σχέση με το πρότυπο. Αυτό, θα μπορούσε να φανεί χρήσιμο στην περίπτωση που θέλουμε να διακρίνουμε μέσα σ' ένα σύνολο από ανακατεμένα γράμματα ποια από αυτά ανήκουν στην ίδια γραμμή ή σε παράλληλες, έτσι ώστε και με την κοινή χρήση συντεταγμένων να μπορούμε να αποφασίσουμε με περισσότερη σαφήνεια για το ποια ανήκουν στο ίδιο κείμενο. Αυτό συνιστά τη διαδικασία της ανάκλησης (αναγνώρισης).

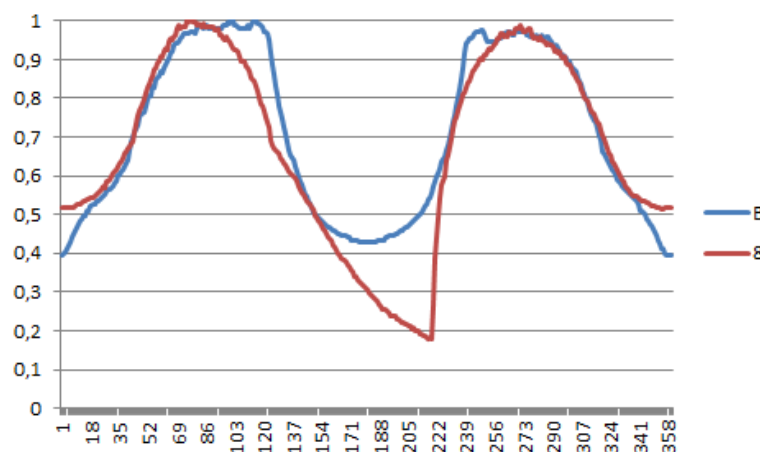
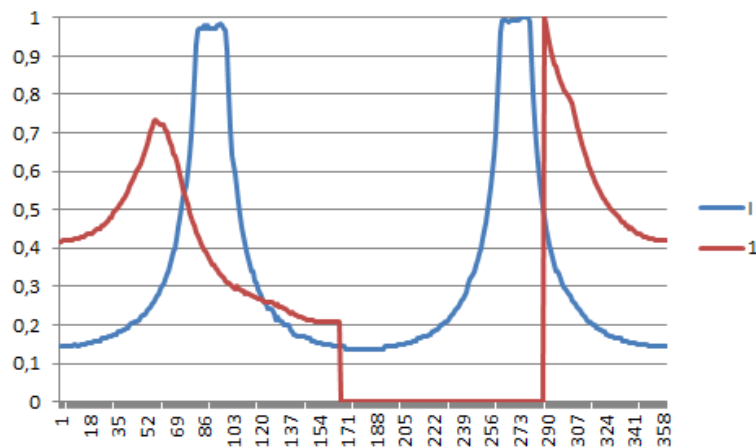
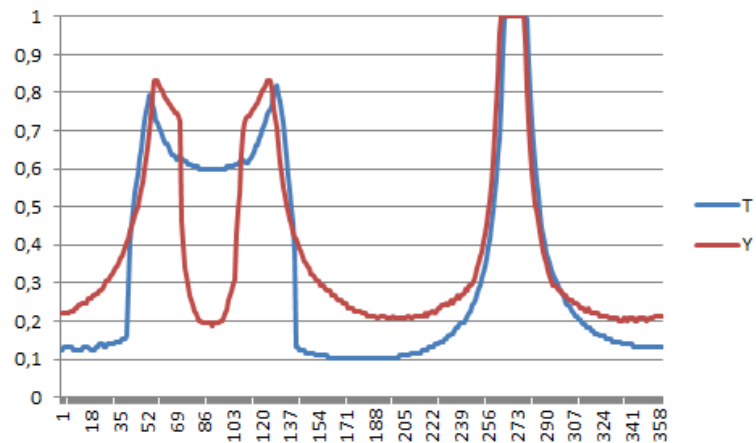
2.2 ΒΕΛΤΙΩΣΕΙΣ

Όπως αναφέρθηκε και στην προηγούμενη παράγραφο, θα μπορούσε κάποιος να σκεφτεί διάφορων ειδών μετρήσεις που θα μπορούσαν να γίνουν πάνω στο κάθε δείγμα. Σκοπός μας είναι να επιλέξουμε ένα μέτρο το οποίο και θα προσφέρει την επαρκή διαφοροποίηση μεταξύ των μορφών τις οποίες μας ενδιαφέρει να αναγνωρίζονται αλλά και θα καθιστά έγκυρη τη διάταξη μεταξύ διαφορετικών αποκλιόντων από τα πρότυπα στιγμιότυπων. Με λίγα λόγια, δεν μας ενδιαφέρει η εύρεση ενός μέτρου το οποίο απλά θα μας λέει αν δύο πρότυπα είναι όμοια, αλλά και πόσο διαφέρουν.

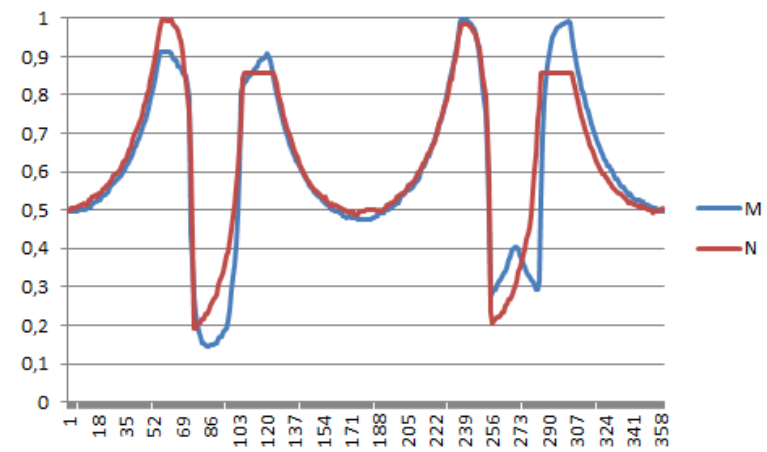
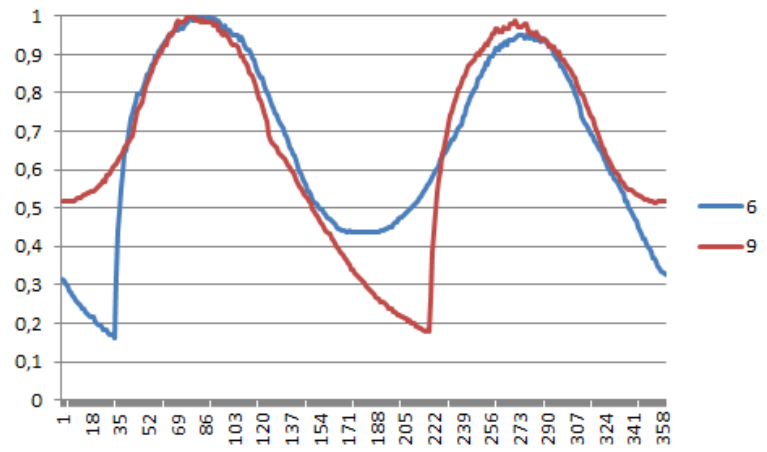
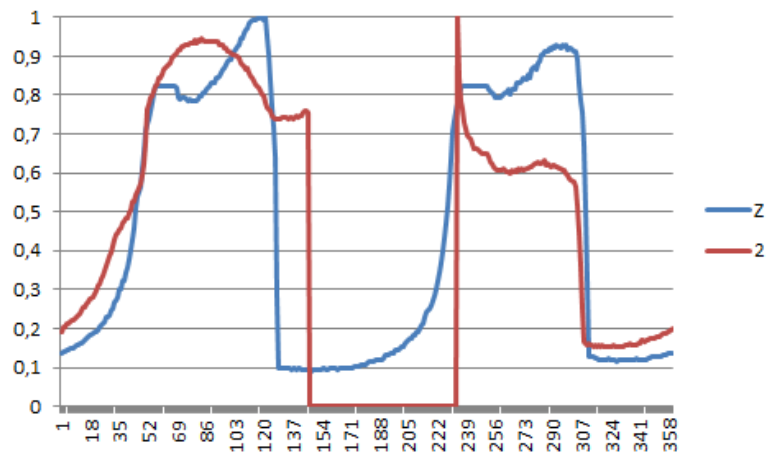
Το πρώτο μέτρο που σκέφτηκα με τη σύλληψη του αλγορίθμου ήταν αυτό που περιγράφηκε παραπάνω, με τον μέσο όρο της απόστασης των μαύρων τετραγώνων από τον άξονα περιστροφής. Όπως φαίνεται στο παρακάτω σχήμα, οι καμπύλες που εξάγονται με τη χρήση αυτού του μέτρου είναι διακριτές για τον κάθε χαρακτήρα. Επιπλέον, εφ' όσων ο υπολογισμός παράγει συνεχές αποτέλεσμα όμοιας αξιολόγησης καθ' όλη την έκταση του πεδίου τιμών του (δηλαδή το ότι το μικρότερο είναι καλύτερο ισχύει για ολόκληρο το $[0,1]$ και όχι μόνο τμηματικά), η σύγκριση μεταξύ δύο μη πλήρως όμοιων μορφών έχει νόημα.



Κάθε χαρακτήρας έχει διαφορετική καμπύλη

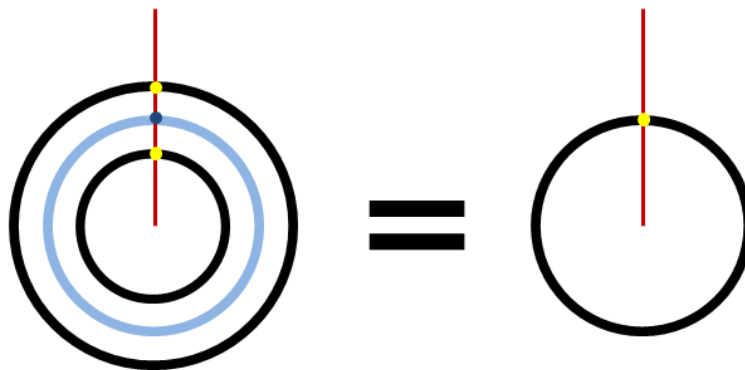


Απομονωμένες καμπύλες ζευγών χαρακτήρων που ομοιάζουν



Απομονωμένες καμπύλες ζευγών χαρακτήρων που ομοιάζουν

Παρατηρούμε ότι παρ' όλο που υπάρχει διαφοροποίηση μεταξύ των καμπυλών όλων των χαρακτήρων, για κάποιες ομάδες από αυτούς η διαφοροποίηση είναι σχετικά μικρή. Αυτό οδηγεί τον αλγόριθμο σε κάποιες λάθος αναγνωρίσεις μεταξύ χαρακτήρων που, για δεδομένη γραμματοσειρά, ανήκουν στην ίδια ομάδα. Επίσης, καθ' ότι η μέτρηση του μέσου όρου δε λαμβάνει υπ' όψιν της την κατανομή των στοιχείων, είναι δυνατόν να κατασκευάσουμε «τεχνητά σύμβολα» τα οποία, χωρίς να είναι, θα παρερμηνεύονται από τον αλγόριθμο ως γνωστά.



Ο πρώτος χαρακτήρας κατά την εξαγωγή του μέσου όρου της απόστασης των pixel του από το κέντρο δίνει για κάθε ακτίνα (δείγμα) το ίδιο αποτέλεσμα με τον δεύτερο αν και πρόκειται για διαφορετικούς χαρακτήρες

Το πρόβλημα του τεχνητού χαρακτήρα της παραπάνω εικόνας μπορεί να λυθεί προσθέτοντας στον τύπο του υπολογισμού ένα νέο στέλεχος, το οποίο θα λαμβάνει υπ' όψιν του και την τυπική απόκλιση των ενεργών τετραγώνων του δείγματος. Βέβαια, χωρίς να το έχω σκεφτεί ιδιαίτερα, μπορεί κάλλιστα να γίνεται και πάλι να κατασκευαστεί κάποιος νέος τεχνητός χαρακτήρας ο οποίος θα ξεγελάει και αυτή την μέτρηση.

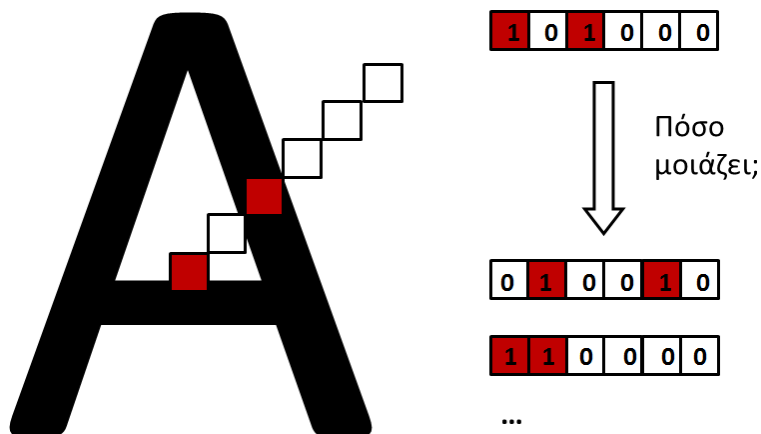
Είναι εφικτή η μέτρηση της μέσης διαφοροποίησης που μπορεί να προσφέρει ένας πιθανός υπολογισμός μεταξύ των χαρακτήρων εκπαίδευσης. Αρκεί κάποιος να εφαρμόσει τον υπολογισμό του οποίου την ποιότητα θέλει να εξακριβώσει πάνω σ' ένα δεδομένο σύνολο χαρακτήρων, να συγκρίνει με τον προαναφερθέντα στην περιγραφή της ανάκλησης τρόπο όλες τις καμπύλες μεταξύ τους και να εξάγει τη ολική μέση διαφοροποίηση. Ο υπολογισμός που προσφέρει τη μέγιστη δυνατή διαφοροποίηση μεταξύ των δοκιμαζόμενων, είναι ο πλέον κατάλληλος γι' αυτό το σύνολο χαρακτήρων.

Προσωπικά, υποστηρίζω ότι υπάρχει ένας συνδυασμός μετρήσεων τέτοιος ώστε να καθιστά ικανή τη διάκριση αλλά και τον αναλογικό υπολογισμό διαφοροποίησης, οποιωνδήποτε διαφορετικών δειγμάτων.

Ένας τέτοιος τρόπος θα μπορούσε να είναι ο εξής. Αντί να κάνουμε πάνω στο δείγμα την οποιαδήποτε αριθμητική μέτρηση, παίρνουμε τα τετράγωνα του ένα ένα με τη σειρά (δεν παίζει ρόλο η κατεύθυνση αρκεί να τηρείται η ίδια σε όλο τον αλγόριθμο) και σημειώνουμε εάν αυτά είναι ενεργά (μαύρα, 1) ή όχι (λευκά, 0). Έτσι, σχηματίζεται για κάθε δείγμα μια αλληλουχία αριθμών από 0 και 1, πλήθους ίσου με το άνοιγμα του διαβήτη (σε πλευρές στοιχειωδών τετραγώνων), το οποίο χαρακτηρίζει το κάθε δείγμα. Αφού συγκεντρώσουμε τις μετρήσεις όλων των δειγμάτων για εύρος ενός πλήρους κύκλου, τις αποθηκεύουμε σ' έναν πίνακα διαστάσεων «αριθμός δειγμάτων x άνοιγμα διαβήτη», ο οποίος αποτελεί τη συνολική μέτρηση της μορφής. Μπορούμε να θεωρήσουμε ότι το κάθε δείγμα είναι ένα σημείο σ' έναν υποθετικό χώρο διαστάσεων όσο και το άνοιγμα του διαβήτη, με συντεταγμένες αυτά τα μηδενικά και τους άσσους. Για παράδειγμα, το υποθετικό δείγμα 001101101000 (12 ψηφία) μπορεί ν' αντιστοιχεί σ' ένα σημείο του 12διάστατου χώρου με συντεταγμένες $a(0,0,1,1,0,1,1,0,1,0,0,0)$. Έτσι, όταν φτάσουμε στον αλγόριθμο στο σημείο που θα πρέπει να υπολογίσουμε τη διαφορά μεταξύ των δύο αντίστοιχων δειγμάτων των καμπυλών που συγκρίνουμε, θα υπολογίσουμε την ευκλείδεια απόσταση μεταξύ των δύο σημείων στα οποία αυτά αντιστοιχούν. Ευτυχώς, το πυθαγόρειο θεώρημα επεκτείνεται σε η διαστάσεις και μας βοηθάει να μετράμε, απτά, αποστάσεις σε χώρους με περισσότερες από 3 διαστάσεις (είναι αδύνατο να το οπτικοποιήσουμε αλλά λειτουργεί):

$$\text{Απόσταση}_{(AB)}^2 = \sum_{i=1}^n (A_i - B_i)^2$$

Όπου n ο αριθμός των διαστάσεων του χώρου και A,B τα δύο σημεία.



Δεν αρκεί να γνωρίζουμε ότι τα δείγματα δεν ταυτίζονται αλλά και πόσο διαφέρουν

3 ΚΑΤΑΣΚΕΥΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΥΤΟΜΑΤΗΣ ΑΝΑΓΝΩΡΙΣΗΣ ΠΙΝΑΚΙΔΩΝ ΚΥΚΛΟΦΟΡΙΑΣ

ΕΙΣΑΓΩΓΗ

Ως Αυτόματη Αναγνώριση Πινακίδων Κυκλοφορίας (Automatic Number Plate Recognition - ANPR) ορίζουμε το σύνολο των τεχνολογιών υλικού και λογισμικού που συνδυασμένα καθιστούν εφικτή την αυτόματη λήψη κι επεξεργασία φωτογραφιών με σκοπό την ταυτοποίηση οχημάτων. Στην καρδιά του λογισμικού ενός τέτοιου συστήματος υπάρχει πάντα ένας αλγόριθμος αναγνώρισης χαρακτήρων, συνήθως τεχνητού νευρωνικού δικτύου, καθιστώντας κατ' ουσίαν τη διαδικασία ειδική περίπτωση οπτικής αναγνώρισης χαρακτήρων.

Η αυτόματη αναγνώριση πινακίδων κυκλοφορίας αναπτύχθηκε για πρώτη φορά το 1976, από το επιστημονικό τμήμα της Βρετανικής Αστυνομίας, ενώ πρώτη φορά τέθηκαν συστήματα σε λειτουργία το 1979. Από τότε ξεκίνησε και η εμπορική διάθεση της σε τρίτες εταιρίες, ενώ η πρώτη αστυνομική σύλληψη που οφείλεται σε τέτοια εφαρμογή έγινε το 1981 στην Αγγλία. Πλέον, η τεχνολογία αυτόματης αναγνώρισης πινακίδων κυκλοφορίας επεκτείνεται σε μια ευρεία γκάμα εφαρμογών, πέραν του αστυνομικού ελέγχου, όπως τον έλεγχο διέλευσης διοδίων, την χρονοχρέωση ιδιωτικών χώρων στάθμευσης, την επιτήρηση πρατηρίων καυσίμων self-service, τον έλεγχο κίνησης ασφαλών εγκαταστάσεων κ.τ.λ.

Συζητήσεις περί ιδιοτικότητας των προσωπικών δεδομένων αναδύθηκαν από τη γένεση ακόμα της τεχνολογίας αυτόματης αναγνώρισης πινακίδων κυκλοφορίας, αλλά με διάφορες κατά τόπους προσαρμογές που συνήθως αφορούν την αποθήκευση και την εκμετάλλευση των πληροφορικών που συλλέγονται κατά την εφαρμογή της, η χρήση της εξακολουθεί να εμφανίζει, παγκοσμίως, θετική τάση.



Κάμερα συστήματος αυτόματης αναγνώρισης πινακίδων κυκλοφορίας ενσωματωμένη σε περιπολικό



Κάμερες κυκλοφορίας με δυνατότητα νυχτερινής λήψης

3.1 ΤΟ ΠΡΟΓΡΑΜΜΑ

Η εφαρμογή ηλεκτρονικού υπολογιστή που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας είχε ως στόχο να επιδείξει και να αποδείξει τη λειτουργία του νεοσυληφθέντα αλγόριθμου οπτικής αναγνώρισης χαρακτήρων που περιγράφηκε στο προηγούμενο κεφάλαιο. Ως εκ τούτου, αποτελεί ένα «κέλυφος» το οποίο επιτρέπει στον χρήστη να δοκιμάσει τον νέο αυτό αλγόριθμο, τροφοδοτώντας τον με ένα σύνολο από φωτογραφίες και δοκιμάζοντας τις διάφορες παραμέτρους του, εξερευνώντας τη λειτουργία του.

Κατά τη γνώμη μου, επίκεντρο και σημαντικότερο στοιχείο αυτής της εργασίας ήταν η ανάπτυξη του νέου αλγόριθμου οπτικής αναγνώρισης χαρακτήρων, και όχι η κατασκευή ενός προγράμματος αυτόματης αναγνώρισης πινακίδων κυκλοφορίας. Ως εκ τούτου, η εφαρμογή κατασκευάστηκε σε αρμονία με τη γενική φύση του αλγορίθμου, μη όντας συγκεκριμένα βελτιστοποιημένα για την περίπτωση, και λαμβάνοντας μόνο κατά τα τελευταία στάδια επεξεργασίας, προ της παρουσίασης του αποτελέσματος στον χρήστη, υπ' όψιν της το γεγονός ότι προορίζεται για την εξέταση φωτογραφιών πινακίδων κυκλοφορίας.

Συγκεκριμένα, η εφαρμογή δεν εκτελεί κάποια από τις εξειδικευμένες διεργασίες που συνήθως προηγούνται της αναγνώρισης σ' ένα πραγματικό πρόγραμμα αυτόματης αναγνώρισης πινακίδων κυκλοφορίας, όπως για παράδειγμα την εύρεση και την απομόνωση της πινακίδας από την υπόλοιπη φωτογραφία, αλλά επιχειρεί στηριζόμενη στο γεγονός ότι οι ελληνικές πινακίδες αποτελούνται μόνο από μονομερείς χαρακτήρες την απ ευθείας αναγνώριση της φωτογραφίας, ελπίζοντας ότι αυτή περιλαμβάνει μόνο χαρακτήρες οι οποίοι ανήκουν στην προς αναγνώριση πινακίδα. Δηλαδή, το πρόγραμμα σαρώνει ολόκληρη τη φωτογραφία από αριστερά προς τα δεξιά λαμβάνοντας υπ όψιν του όλους τους χαρακτήρες που συναντάει στην πορεία, και προσπαθεί με ότι συνέλεξε να συνθέσει στο τέλος μια έγκυρη για τα δεδομένα του ελληνικού νόμου πινακίδα κυκλοφορίας. Ασφαλώς και εάν επρόκειτο για εμπορική εφαρμογή, ο αλγόριθμος αναγνώρισης θα επωφελούνταν σημαντικά από την ύπαρξη τέτοιων διεργασιών προεπεξεργασίας των φωτογραφιών που εξετάζει, αλλά στα πλαίσια αυτής της εφαρμογής η οποία προορίζεται αποκλειστικά για την επίδειξη και τον πειραματισμό του αλγορίθμου που υλοποιεί, δεν κρίθηκε απαραίτητο.

Η εφαρμογή γράφτηκε εξ' ολοκλήρου και από λευκής χάρτου σε γλώσσα προγραμματισμού Java, καθώς δεδομένων των μαθημάτων που παρακολούθησα στο τμήμα πληροφορικής του Α.Τ.Ε.Ι. Θεσσαλονίκης είναι η γλώσσα με την οποία έχω την μεγαλύτερη ευχέρεια. Στον κώδικα της εφαρμογής δεν γίνεται κλήση καμιάς εξωτερικής βιβλιοθήκης, πέραν των κλάσεων που υπάρχουν ενσωματωμένες στη γλώσσα, ενώ ακόμα και το

γραφικό της περιβάλλον γράφτηκε και αυτό εξ' ολοκλήρου στο χέρι (Java swing), χωρίς τη χρήση κάποιου εργαλείου αυτόματης κατασκευής γραφικού περιβάλλοντος (GUI builder).

Στην παρούσα έκδοση, ο πηγαίος κώδικας συνίσταται σε 1350 μεικτές γραμμές κατανεμημένες σε 2 αρχεία. Ένα αρχείο MainGUI.java, 710 γραμμών, το οποίο περιέχει τον κώδικα του γραφικού περιβάλλοντος κι ένα αρχείο MainPipeline.java, 640 γραμμών, το οποίο περιέχει τον κώδικα του «πυρήνα σκέψης» της εφαρμογής, συμπεριλαμβανομένης και της υλοποίησης τμηματικά σε μεθόδους του αλγορίθμου που περιγράφηκε νωρίτερα. Για την συγγραφή των 14 κλάσεων και των 40 μεθόδων που απαρτίζουν την εφαρμογή αφιερώθηκαν λίγο παραπάνω από 3 αδιάλειπτοι μήνες καθημερινής πολύωρης εργασίας, ενώ αξίζει να σημειωθεί ότι καθ' όσων ο σχεδιασμός γινόταν εξ' ολοκλήρου από λευκής χάρτου, υπολογίζεται ότι για κάθε τελική γραμμή κώδικα που προέκυψε δημιουργήθηκε σε κάποιο σημείο κατ' ελάχιστον άλλη μία, η οποία αναλώθηκε σε δοκιμές και ελέγχους.

Η επεξεργασία της φωτογραφίας γίνεται μέσα στην εφαρμογή κατά στάδια. Σχηματικά, ο «σωλήνας επεξεργασίας» (pipeline) της εφαρμογής μπορεί να παρασταθεί ως εξής:



imageToActivePixels



getMorphes



formalizeMorphes



recognition

NHN62A4BEI

formalizeResults

NHN-6246

Ο «σωλήνας επεξεργασίας» της εφαρμογής

Ο αλγόριθμος επεξεργάζεται τις φωτογραφίες ως δισδιάστατους πίνακες ενεργών (μαύρων, 1) και ανενεργών (άσπρων, 0) τετραγώνων. Η μετατροπή μιας έγχρωμης φωτογραφίας σε πίνακα 2 καταστάσεων γίνεται εξάγοντας για κάθε τετράγωνο την φωτεινότητά του (luminosity) και επιλέγοντας την τιμή από την οποία και κάτω το τετράγωνο θα θεωρείται μαύρο (ενεργό). Η τιμή αυτή στον κώδικα ορίζεται ως «κατώφλι του μαύρου» (black threshold) ενώ ο τύπος με τον οποίο, δοθέντων των τιμών κόκκινου-πράσινου-μπλε ενός τετραγώνου (RGB), υπολογίζεται η φωτεινότητα του είναι ο εξής:

$$\text{Luminosity} = (R \times 0.2126) + (G \times 0.7152) + (B \times 0.0722)$$

Να σημειωθεί ότι υπάρχουν και άλλοι τύποι υπολογισμού της φωτεινότητας ενός rgb pixel, όμως αυτός είναι ο πλέον επικρατών καθώς λαμβάνει υπ' όψιν του τη διαφορετική ευαισθησία που έχει το ανθρώπινο μάτι στο κάθε ένα από τα τρία βασικά χρώματα (human color perception).

Έτσι, μια φωτογραφία μετά την είσοδό της στον σωλήνα επεξεργασίας του αλγορίθμου μετατρέπεται και διακινείται μεταξύ των διαφορετικών σταδίων του ως πίνακας ενεργών-ανενεργών pixel, μέχρι και την εξαγωγή του αποτελέσματος.

Επόμενο στάδιο της επεξεργασίας είναι ο διαχωρισμός των μορφών της εισαχθείσας φωτογραφίας. Αυτό γίνεται στη μέθοδο "getMorphes". Όπως είπαμε και προηγουμένως, δεν εφαρμόζεται κάποια ειδική τεχνική, απλώς στηριζόμενοι στο γεγονός ότι οι ελληνικές πινακίδες αποτελούνται αποκλειστικά από μονομερείς χαρακτήρες χωρίζουμε τις μορφές βάσει της συνέχειας των ενεργών pixel. Δηλαδή, ορίζουμε μία τιμή η οποία στον κώδικα αναφέρεται ως «ακτίνα γειτονιάς» (neighborhood radius) και ομαδοποιούμε στην ίδια μορφή τα pixels που απέχουν μεταξύ τους απόσταση ίση ή μικρότερη από αυτή την τιμή.

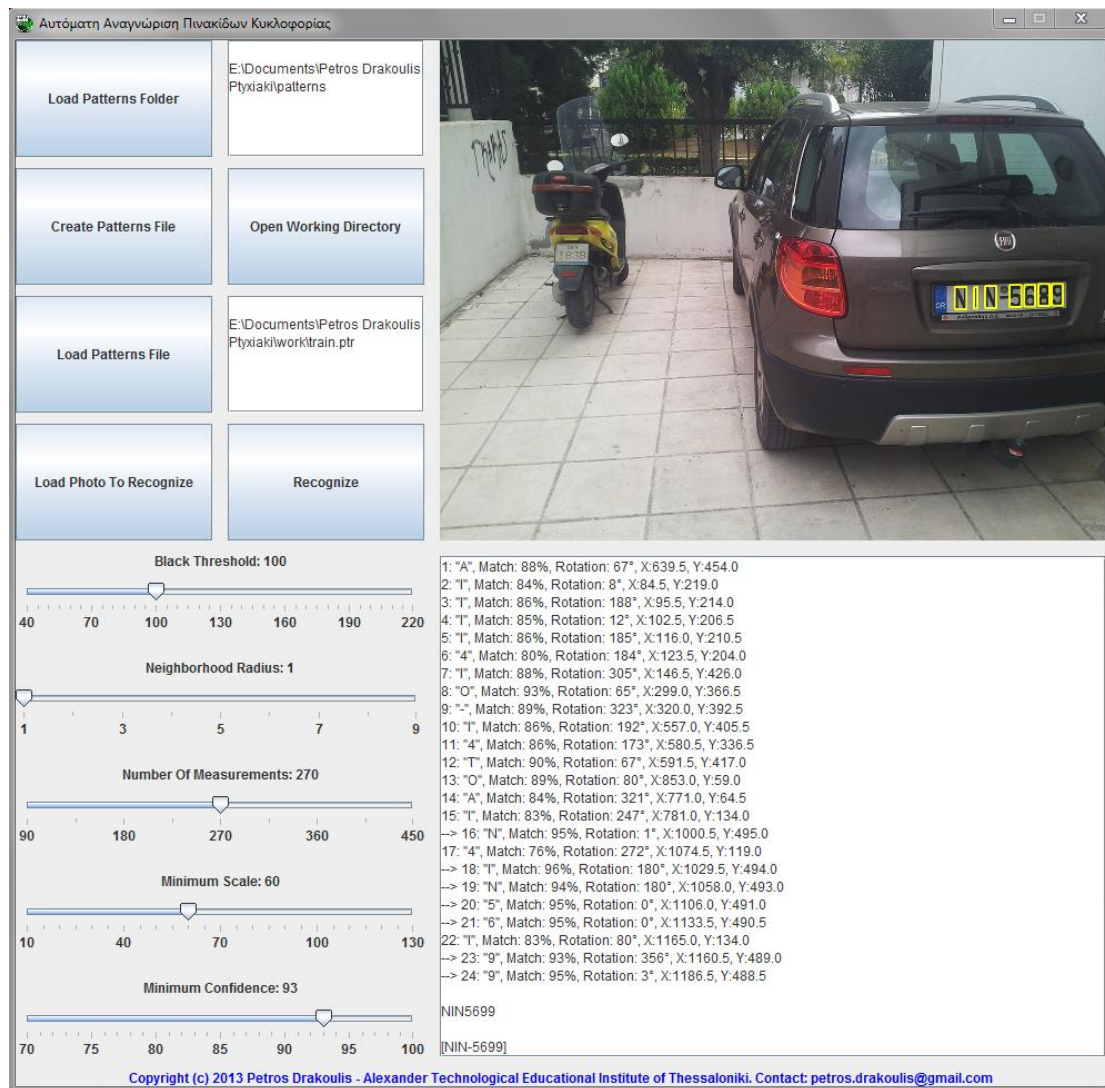
Στη συνέχεια, γίνεται μια πρώτη απόρριψη των πολύ μικρών μορφών βάσει της μεταβλητής «κλίμακα» (scale) και οι εναπομείνουσες μορφές κεντράρονται σε τετράγωνους πίνακες ώστε να είναι έτοιμες για αναγνώριση. Αυτή η διαδικασία εκτελείται στη μέθοδο "formalizeMorphes".

Ο «πυρήνας» της αναγνώρισης, δηλαδή η περιστροφική πολλαπλή δειγματοληψία που προτείνεται σε αυτήν την εργασία, κωδικοποιείται στη μέθοδο "recognition". Εκεί, λαμβάνεται από κάθε μορφή αριθμός δειγμάτων όσος ορίζεται στη μεταβλητή «αριθμός μετρήσεων» (number of patterns), ενώ αξίζει να σημειωθεί ότι ανεξαρτήτως της τιμής που επέλεξε ο χρήστης, το πρόγραμμα πάντα ταυτίζει την τιμή αυτή με την τιμή με την οποία μετρήθηκαν

τα πρότυπα, όπως αυτά περιγράφονται στο εκάστοτε ενεργό αρχείο εκπαίδευσης.

Το τελευταίο στάδιο της αναγνώρισης κωδικοποιείται στη μέθοδο “formalizeResults” και περιλαμβάνει την αποβολή των μορφών των οποίων η ταύτιση με κάποιο από τα πρότυπα είναι μικρότερη απ’ όση ορίζεται στη μεταβλητή «μέγιστη διαφορά ταύτισης» (maxMatchDifference), όπως και για πρώτη φορά στον αλγόριθμο την εφαρμογή κάποιων «έξυπνων» διορθώσεων, στοχευμένων ειδικά στη μορφή των ελληνικών πινακίδων. Συγκεκριμένα, ο αλγόριθμος γνωρίζει ότι σε μια ελληνική πινακίδα οι πρώτοι τρεις χαρακτήρες είναι γράμματα, ακολουθεί μια παύλα και μετά τέσσερις αριθμοί, αλλάζει κάποιους αναγνωρισμένους χαρακτήρες σε κάποιους πιο κατάλληλους για τη θέση που βρέθηκαν. Δηλαδή, εάν βρει το ‘2’ ανάμεσα στους τρεις πρώτους χαρακτήρες προφανώς πρόκειται για το ‘Ζ’, εάν βρει το ‘I’ στην τέταρτη θέση προφανώς πρόκειται για την παύλα ‘-’, ενώ αν βρει μεταξύ των τελευταίων τεσσάρων χαρακτήρων το ‘B’ προφανώς πρόκειται για το ‘8’. Ανάλογα χειρίζεται και την περίπτωση του ζευγαριού 0,0.

Στην παρούσα φάση, το πρόγραμμα μπορεί να διαχειριστεί μόνο μία πινακίδα ανά φωτογραφία.



Το γραφικό περιβάλλον της εφαρμογής

Το γραφικό περιβάλλον της εφαρμογής αποτελείται από ένα μόνο παράθυρο, χωρισμένο σε 4 περιοχές. Η πάνω αριστερή περιοχή περιέχει τα πλήκτρα ελέγχου, η κάτω αριστερή τους ρυθμιστές των παραμέτρων εκτέλεσης, η πάνω δεξιά τη φωτογραφία προς αναγνώριση και η κάτω δεξιά την έξοδο του προγράμματος. Αναλυτικότερα, το κάθε στοιχείο χειρισμού του γραφικού περιβάλλοντος κάνει το εξής:

- Load Patterns Folder – Επιλέγει ο χρήστης τον φάκελο μέσα στον οποίο βρίσκονται οι φωτογραφίες πρότυπα που θα χρησιμοποιηθούν για την εκπαίδευση του αλγόριθμου αναγνώρισης.

- **Create Patterns File** – Δημιουργείται στο φάκελο work το αρχείο train.ptf με τις μετρήσεις όλων των προτύπων που βρίσκονται στο φάκελο που φορτώθηκε παραπάνω και το οποίο τελικά χρησιμοποιεί ο αλγόριθμος αναγνώρισης για να ταυτοποιήσει τα ευρήματα. Η φάση αυτή αντιστοιχεί στη φάση εκπαίδευσης των Τεχνητών Νευρωνικών Δικτύων.
- **Open Working Directory** – Ανοίγει τον φάκελο μέσα στον οποίο το πρόγραμμα αποθηκεύει τα αποτελέσματα του. Είναι χρήσιμο να ελέγχουμε μετά από κάθε αποτυχημένη αναγνώριση τον φάκελο αυτό και να βλέπουμε τα τμήματα των φωτογραφιών τα οποία διαχώρισε και πάνω στα οποία επιχείρησε την αναγνώριση ώστε να προβούμε στις κατάλληλες διορθώσεις των παραμέτρων εκτέλεσης προσβλέποντας σε μια καλύτερη συνέχεια.
- **Load Patterns File** – Επιλέγει ο χρήστης το αρχείο μετρήσεων προτύπων που θα χρησιμοποιήσει ο αλγόριθμος κατά τη φάση της αναγνώρισης για την ταυτοποίηση των ευρημάτων.
- **Load Photo To Recognize** – Επιλέγει ο χρήστης τη φωτογραφία στην οποία επιθυμεί να προσπαθήσει ο αλγόριθμος να αναγνωρίσει τις πινακίδες κυκλοφορίας.
- **Recognize** – Εκκινεί τον αλγόριθμο αναγνώρισης της δοθείσας φωτογραφίας με τις τρέχουσες ρυθμίσεις παραμέτρων.
- **Black Threshold** – Ορίζει το επίπεδο φωτεινότητας από το οποίο και κάτω τα pixels θα θεωρούνται μαύρα. Στις σκοτεινές φωτογραφίες θα πρέπει να ορίζεται αρκετά υψηλό ώστε να μετρούν σαν μαύρα μόνο τα πραγματικά μαύρα pixels, ενώ στις φωτεινές φωτογραφίες όπου τα μαύρα γράμματα των πινακίδων μπορεί να πλησιάζουν περισσότερο το γκρι θα πρέπει να ορίζεται χαμηλό.
- **Neighborhood Radius** – Ορίζει τη μέγιστη απόσταση που μπορούν να έχουν γειτονικά μαύρα pixels ώστε να θεωρείται ότι ανήκουν στο ίδιο σχήμα. Είναι χρήσιμο σε κακές φωτογραφίες με θόρυβο όπου κάποιο γράμμα μπορεί να τεμαχίζεται από μια λευκή γραμμή. Σε αυτήν τη περίπτωση μια τιμή μεγαλύτερη του 1 μπορεί να βοηθήσει ώστε να αγνοηθεί η γραμμή. Ταυτόχρονα όμως δυσκολεύει το διαχωρισμό των κοντινών σχημάτων μέσα στη φωτογραφία καθώς τα κάνει να θεωρούνται ένα. Επίσης αυξάνει το χρόνο επεξεργασίας της φωτογραφίας και γι αυτό τιμές μεγαλύτερες του 1 θα πρέπει να χρησιμοποιούνται μόνο ως τελευταίες λύσεις ανάγκης.
- **Number Of Measurements** – Ορίζει τον αριθμό των δειγμάτων που θα πάρει περιστροφικά ο αλγόριθμος από το κάθε σχήμα. Η αύξηση του κάνει τον αλγόριθμο να αντιλαμβάνεται τη μορφή του κάθε σχήματος με μεγαλύτερη ακρίβεια, συμβάλλοντας με αυτόν τον τρόπο στη βελτίωση της ποιότητας αναγνώρισης, αλλά ταυτόχρονα επιβαρύνει ελαφρώς το χρόνο επεξεργασίας της φωτογραφίας χωρίς να έχει ιδιαίτερο νόημα σε

φωτογραφίες χαμηλής ανάλυσης όπου διαδοχικές μετρήσεις δίνουν ακόμη και το ίδιο αποτέλεσμα.

- **Minimum Scale** – Ορίζει την κλίμακα από την οποία και πάνω θα λαμβάνονται προς εξέταση τα διάφορα σχήματα της φωτογραφίας. Βρίσκοντας πειραματικά την τιμή η οποία επιτρέπει οριακά στα γράμματα της φωτογραφίας να περνούν προς εξέταση, αποκλείουμε σε μεγάλο βαθμό την εξέταση του θορύβου της φωτογραφίας. Για παράδειγμα, η ρύθμιση 40 σημαίνει ότι θα λαμβάνονται προς εξέταση μόνο τα σχήματα που είναι σε μέγεθος 40 φορές μικρότερα από τη φωτογραφία ή μεγαλύτερα. Μεγάλες τιμές καθιστούν δυνατό να αναγνωριστούν μικρά γράμματα αλλά παράλληλα, αφού αυξάνεται το πλήθος των σχημάτων που τελικά θα εξεταστούν, αυξάνουν και το πλήθος των εσφαλμένων αναγνωρίσεων.
- **Minimum Confidence** – Ορίζει το ελάχιστο ποσοστό ταύτισης που πρέπει να έχουν τα ευρήματα της φωτογραφίας με τα πρότυπα εκπαίδευσης ώστε να θεωρηθεί ότι αναγνωρίστηκαν. Μεγάλες τιμές καθιστούν δύσκολη τη λανθασμένη αναγνώριση, αλλά δυσχεραίνουν και τη σωστή σε φωτογραφίες όπου τα γράμματα είτε δε φαίνονται καλά είτε απέχουν αρκετά από τα πρότυπα εκπαίδευσης.

Αξίζει να σημειωθεί ότι οι περιγραφές των στοιχείων χειρισμού υπάρχουν αυτούσιες στο γραφικό περιβάλλον της εφαρμογής, υπό τη μορφή αναδυόμενων παραθύρων, αρκεί ο χρήστης ν' αφήσει το ποντίκι πάνω στο στοιχείο χειρισμού που τον ενδιαφέρει.

Ένα ακόμη ενδιαφέρον χαρακτηριστικό που ενσωματώθηκε στο γραφικό περιβάλλον της εφαρμογής είναι, όπως φαίνεται και από τη φωτογραφία, η δυνατότητα της να «κυκλώνει» τα ευρήματα της πάνω στη φωτογραφία. Με αυτό τον τρόπο γίνεται πιο κατανοητή στον τελικό χρήστη η λειτουργία του προγράμματος.

Επίσης, στην έξοδο κειμένου της εφαρμογής δεν προβάλλεται μόνο το τελικό αποτέλεσμα, αλλά και το ενδιάμεσο, που παραδίδεται από τη μέθοδο της περιστροφικής πολλαπλής δειγματοληψίας (recognition) στη μέθοδο της τελικής εξαγωγής συμπερασμάτων (formalizeResults). Δηλαδή, μια λίστα με όλες τις μορφές που επιχειρήθηκε ν' αναγνωριστούν, τη βέλτιστη ταύτιση τους με κάποιο από τα πρότυπα, το ποσοστό ταύτισης, την γωνία περιστροφής και τις συντεταγμένες των διάφορων μορφών πάνω στη φωτογραφία. Έτσι, ο χρήστης έχει τη δυνατότητα, αφού ανοίξει τον φάκελο εργασίας της εφαρμογής (working directory) από το ανάλογο πλήκτρο, να δει μία μία τις μορφές της αρχικής φωτογραφίας που απομονώθηκαν και, ψάχνοντας το ανάλογο όνομα αρχείου στη λίστα, να δει την απόφαση που πήρε ο αλγόριθμος αναγνώρισης γι' αυτές.

3.2 ΒΕΛΤΙΩΣΕΙΣ

Η εφαρμογή, αν και είναι λογικά ορθή στα πλαίσια ενός νοητού ηλεκτρονικού υπολογιστή, ενδέχεται να παρουσιάσει κατά την πραγματική της εκτέλεση κάποια προβλήματα. Βεβαίως, προνοήθηκε από σχεδιασμού της όλα τα σφάλματα που μπορεί να προκύψουν να είναι ελεγχόμενα, με την έννοια ότι εφ' όσων η εικονική μηχανή της java (java vm) βρίσκεται σε λειτουργία, η εφαρμογή δεν «κρεμάει» ποτέ. Εάν προκύψει σφάλμα, απλά εμφανίζεται το ανάλογο μήνυμα και ο χρήστης μπορεί να εκτελέσει και πάλι την «προβληματική» ενέργεια, δοκιμάζοντας κάποιες αλλαγές στις παραμέτρους.

Το πιο συχνό σφάλμα που μπορεί να προκύψει είναι η υπερχειλίση της στοίβας (stack overflow). Όπως περιγράψαμε και στο προηγούμενο κεφάλαιο, ο αλγόριθμος απομονώνει τις μορφές εξετάζοντας τη συνέχεια των pixel. Δηλαδή, εφ' όσων βρει ένα ενεργό (μαύρο, 1) pixel στη φωτογραφία το οποίο δεν ανήκει ήδη σε κάποια ομάδα (μορφή), ξεκινάει μια νέα ομάδα, κοιτάζει όλα τα «γειτονικά» του, δηλαδή αυτά που βρίσκονται γύρω του σε απόσταση το πολύ ίση με την τιμή που ορίζεται στη μεταβλητή «ακτίνα της γειτονιάς» (neighborhood radius) και εφ' όσων αυτά δεν ανήκουν ήδη σε κάποια ομάδα τα εγγράφει σε αυτήν. Έπειτα, συνεχίζουν αυτά πλέον να εγγράφουν τα δικά τους γειτονικά στην ομάδα και ούτω καθ' εξής. Ο τρόπος που κωδικοποιώ την παραπάνω συμπεριφορά είναι αναδρομικός (recursive) και είναι λογικά περατός (finite – ολοκληρώνεται μετά από συγκεκριμένο, μη άπειρο, αριθμό βημάτων). Το πρόβλημα είναι ότι από τη φύση της η Java έχει πολύ μικρή στοίβα (stack) για το σκοπό αυτό. Ανάλογα με την έκδοση του λειτουργικού συστήματος και της αρχιτεκτονικής του ηλεκτρονικού υπολογιστή, αυτή κυμαίνεται από 320KB μέχρι 1MB ανά νήμα εκτέλεσης (thread). Έτσι, εάν εισαχθεί προς αναγνώριση μια φωτογραφία υψηλής ανάλυσης η οποία περιέχει αρκετά παχιές, «αναδρομικά βαθιές» μορφές, η στοίβα γεμίζει προτού φτάσει ο αλγόριθμος στο μέγιστο αναδρομικό βάθος της δομής που εξετάζει, οπότε και προκύπτει εξαίρεση. Στην παρούσα έκδοση, αυτό αντιμετωπίζεται μερικώς δίνοντας κρυφά από το χρήστη στην εικονική μηχανή της java την εντολή (vm argument) αύξησης της στοίβας «-Xss128m», η οποία αυξάνει τη στοίβα στα 128MB, αλλά αυτό συνήθως δεν είναι αρκετό για «βαριές» φωτογραφίες ανάλυσης μεγαλύτερης από 1 με 2 megapixel (1.000.000 με 2.000.000 pixels). Μια ποιοτικότερη, αλλά περισσότερο περίπλοκη, λύση θα ήταν η αντικατάσταση του αναδρομικού κώδικα με επαναληπτικό και διατήρηση των δεδομένων σ' έναν πίνακα. Δεδομένου ότι η Java αποθηκεύει τους πίνακες στον σπορό (heap), ο οποίος είναι τάξεις μεγαλύτερος από τη στοίβα και επεκτείνεται ευκολότερα βάσει εντολών που περιλαμβάνουν ποσοστό χρήσης της μνήμης του ξενιστή ηλεκτρονικού υπολογιστή αντί σταθερής τιμής, το προσδοκώμενο αποτέλεσμα είναι ανώτερο.

Ένα άλλο σφάλμα που μπορεί να προκύψει κατά την εκτέλεση της εφαρμογής είναι η εξάντληση του σορού (heap exhaustion). Αυτό συμβαίνει γιατί ο αλγόριθμος είναι γραμμένος μη έχοντας στο νου τους περιορισμούς ενός πραγματικού ηλεκτρονικού υπολογιστή. Κατά τη θεωρητική ανάπτυξη του αλγορίθμου, φανταζόμουν τα διάφορα στάδια του «σωλήνα επεξεργασίας» (pipeline) να εκτελούνται κατά σειρά, και να μην ξεκινάει ποτέ ένα νέο στάδιο προτού τελειώσει με ολόκληρη τη φωτογραφία το προηγούμενο. Δηλαδή, οραματιζόμουν να γίνεται πρώτα η μετατροπή όλων των pixels της φωτογραφίας σε άσπρο-μαύρο, μετά η εξαγωγή όλων των μορφών από την αρχική φωτογραφία, στο επόμενο στάδιο παραλαβή όλων αυτών και κεντράρισμά τους ώστε να είναι κατάλληλες για αναγνώριση, μετά η αναγνώριση, πάλι όλων, και τέλος η διαλογή για τη διαμόρφωση του τελικού αποτελέσματος. Αυτός ο τρόπος σκέψης, έχει ως αποτέλεσμα τη διακίνηση τεράστιων πινάκων μεταξύ των σταδίων, οι οποίοι προωθούν με μιας όλα τ' αποτελέσματα του ενός σταδίου στο επόμενο. Λογικά είναι ορθό, αλλά αυτοί οι τεράστιοι πίνακες (για παράδειγμα αυτός που περιέχει όλες τις μορφές που διαχωρίστηκαν σε πλήρη ανάλυση – σε μια φωτογραφία 1000 x 1000 αν βρέθηκαν 100 μορφές ο πίνακας έχει μέγεθος $1000 \times 1000 \times 100 = 100.000.000 \text{ bytes} = 100\text{MB}$) μπορεί να εξαντλήσουν το μέγεθος του σορού που αποθηκεύονται. Στην πραγματικότητα, προφανώς πέραν του τελευταίου σταδίου της επεξεργασίας που είναι η απόφαση, όλα τα στάδια θα μπορούσαν να εκτελεστούν για κάθε μια από τις πιθανές μορφές ξεχωριστά. Δηλαδή, με το που θα έβρισκε ο αλγόριθμος μία μορφή, θα ξεκινούσε μόνο πάνω της το κεντράρισμα και την αναγνώριση. Θα κρατούσε στην άκρη τα στοιχεία της, όπως την απόφαση, το ποσοστό ταύτισης, την κλίση ή ότι άλλο μπορεί να υπολογίζει, και μετά θα επέστρεφε στην αναζήτηση της επόμενης μορφής. Μόλις την έβρισκε θα έκανε τα ίδια γι' αυτήν. Με αυτό τον τρόπο οι μετακινήσεις μεταξύ των διαφόρων σταδίων θα ήταν τάξεις μικρότερες και δε θα υπήρχε το πρόβλημα. Δυστυχώς, η σκέψη αυτή έγινε σε πολύ προχωρημένο στάδιο και οι αλλαγές που απαιτούσε στον κώδικα ώστε να πραγματοποιηθεί ήταν απαγορευτικές για τον χρόνο που μου έμενε για να ολοκληρώσω την εργασία.

Μία άλλη ατέλεια του προγράμματος είναι το σταθερό μέγεθος του παραθύρου της. Καθώς το γραφικό περιβάλλον δε γράφτηκε με τη βοήθεια κάποιου αυτόματου γραφικού εργαλείου (gui builder) αλλά στο χέρι, ο πολύπλοκος τρόπος με τον οποίο συντίθενται χειροκίνητα τα παράθυρα στη Swing δεν επιτρέπει πάντα την από σχεδιασμού προβλεπόμενη συμπεριφορά των διαφόρων στοιχείων του παραθύρου κατά την αλλαγή μεγέθους, τουλάχιστον από κάποιον που δεν έχει την ανάλογη εμπειρία. Εάν επέτρεπα στο παράθυρο της εφαρμογής ν' αλλάζει μέγεθος κατά βούληση του χρήστη, προέκυπταν κατά τις διάφορες επεκτάσεις και σμικρύνσεις «άσχημες» διατάξεις των στοιχείων και αλλοίωση της μορφής του γραφικού περιβάλλοντος που θα ήθελα να βλέπει ο χρήστης. Οπότε, εφ' όσων μετά

από πειράματα και κάποιες αλλαγές στον κώδικα μερικών στοιχείων δεν κατάφερα να κάνω όλες τις πιθανές αλλαγές του μεγέθους να δίνουν αποτέλεσμα δομής όμοιας με αυτήν που θα ήθελα να κοιτάζει ο χρήστης, αποφάσισα να ορίσω ένα σταθερό μέγεθος παραθύρου και να μην επιτρέπω στο χρήστη να το μεταβάλλει. Το πρόβλημα είναι ότι ήθελα τα διάφορα στοιχεία να είναι αρκετά μεγάλα, και ειδικά η φωτογραφία, ώστε να φαίνονται πάνω της οι λεπτομέρειες και οι ενδείξεις του αλγορίθμου, οπότε επέλεξα ένα μέγεθος παραθύρου περίπου 1000 x 1000 pixels. Δυστυχώς, σε αναλύσεις οθόνης μικρότερες από αυτό το παράθυρο της εφαρμογής δεν χωράει πλήρως στην οθόνη. Βέβαια, στις μέρες μας δεν χρησιμοποιούνται πλέον σε ηλεκτρονικό υπολογιστή αναλύσεις πλάτους μικρότερου από 1024 pixels, εξακολουθούν όμως να χρησιμοποιούνται μικρότερου ύψους (π.χ. hd ready 720p). Σε τέτοια περιβάλλοντα, δυστυχώς, η εφαρμογή στην παρούσα έκδοση δεν μπορεί να προβληθεί σωστά.

Μια άλλη βελτίωση την οποία θα μπορούσε να υποστεί η εφαρμογή είναι η παραλληλοποίηση (paralleling) και η πολυνηματοποίηση της (multi-threading). Για λόγους τους οποίους αναλύσαμε εν μέρει προηγουμένως, ο νέος αυτός αλγόριθμος αναγνώρισης είναι εν δυνάμει παράλληλος. Αυτό μπορεί να γίνει και σε επίπεδο μορφών, όπως είδαμε, δηλαδή η κάθε μορφή να εξετάζεται παράλληλα, αλλά και σε επίπεδο μίας μορφής, να λαμβάνονται και να υπολογίζονται τα δείγματα πάνω στον κύκλο της παράλληλα. Θεωρώ ότι υπάρχουν μεγάλα περιθώρια παραλληλισμού στον κώδικα, αρκεί κάποιος να δείξει την πειθαρχία να το κάνει σωστά και δομημένα.

Μία άλλη δυνατότητα που θα μπορούσε να εξερευνηθεί, είναι η εκμετάλλευση από πλευράς του αλγορίθμου των γνώσεων τοπολογίας και κατεύθυνσης που αναπτύσσει κατά την διαδικασία της αναγνώρισης, ώστε να ομαδοποιεί τους χαρακτήρες που βρίσκει με σκοπό να καταλάβει ποιοι από αυτούς ανήκουν στην ίδια πινακίδα, πόσες πινακίδες υπάρχουν στη φωτογραφία, ποια φορά ανάγνωσης είναι η σωστή κ.τ.λ. Εάν το κατάφερνε αυτό, θα μπορούσε το πρόγραμμα χωρίς καν να απομονώνει και να εστιάζει σε κάποια περιοχή (όπως γίνεται και τώρα) να αναγνωρίζει ταυτόχρονα πολλές πινακίδες (όπως δεν μπορεί να κάνει τώρα). Για παράδειγμα, χαρακτήρες που έχουν την ίδια κλίση, μέγεθος και απέχουν μεταξύ τους λίγο, έχουν πολλές πιθανότητες ν' ανήκουν στην ίδια πινακίδα. Επίσης, μπορούμε από την κλίση που έχουν συνολικά οι χαρακτήρες μιας ομάδας να καταλάβουμε και τη σωστή φορά ανάγνωσης της.

Βεβαίως, όπως τονίσαμε και σε προηγούμενο κεφάλαιο, ο αλγόριθμος αναγνώρισης θα επωφελούταν δεόντως από τις διάφορες τεχνικές προεπεξεργασίας των εικόνων που υπάρχουν, όπως την εύρεση και την απομόνωση της πινακίδας (κάτι που θα του επέτρεπε να αναγνωρίζει και πολλές πινακίδες ανά φωτογραφία), κ.τ.λ, αλλά όλα αυτά ξεφεύγουν από τους στόχους αυτής της εφαρμογής, ο οποίος είναι απλά και μόνο η επίδειξη της

βασικής λειτουργίας του αλγορίθμου περιστροφικής πολλαπλής δειγματοληψίας και αναγνώρισης.

Ακόμη, θα μπορούσε να προστεθεί στο πρόγραμμα ως ένα ακόμη στοιχείο ευφυΐας η δυνατότητα της αυτό-προσαρμογής (self-adjusting) των παραμέτρων εκτέλεσης (κατώφλι του μαύρου, ελάχιστη κλίμακα μορφών, ελάχιστη ταύτιση με κάποιο πρότυπο - ίσως και ακτίνα της γειτονιάς) κατά τη διάρκεια της εκτέλεσης, ανάλογα με το πόσο πετυχημένα βαίνει, κάνοντας χρήση κάποιου αλγορίθμου βελτιστοποίησης. Η έννοια της επιτυχίας θα μοντελοποιείται σε μια «συνάρτηση ενέργειας» (energy function) την οποία θα προσπαθεί ο αλγόριθμος να μεγιστοποιεί δυναμικά (mathematical optimization), μεταβάλλοντας τις παραμέτρους (fine-tuning) μεταξύ των εκτελέσεων, η μορφή πάντως της οποίας δεν διερευνάται στο στάδιο της έρευνας στο οποίο βρίσκομαι.

Τέλος, αν και ήδη εφαρμόζεται στο τελευταίο στάδιο μια πρώτη «έξυπνη» διόρθωση του αποτελέσματος στηριζόμενη στο γεγονός ότι οι ελληνικές πινακίδες έχουν πάντα στις πρώτες τρεις θέσεις γράμματα και στις επόμενες τέσσερις αριθμούς, ένα ακόμη ποιοτικότερο φίλτρο, προσαρμοσμένο ακόμη καλύτερα στις ιδιαιτερότητες των ελληνικών πινακίδων, όπως το ότι δεν υπάρχουν όλοι οι συνδυασμοί γραμμάτων ή ότι δεν υπάρχει ολόκληρο το εύρος των τετραψήφιων αριθμών για όλους τους κωδικούς πόλεων κ.τ.λ., θα λειτουργούσε ακόμη καλύτερα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Διαμαντάρας Κ. (2007), Τεχνητά νευρωνικά δίκτυα, Κλειδάριθμος, Αθήνα, Ελλάδα

Δουλγέρη Ν. (2007), Οπτική αναγνώριση λέξης, Διπλωματική εργασία του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου, Καρλόβασι, Ελλάδα

Χόρτης Σ.Μ. (2011), Σύστημα οπτικής αναγνώρισης χαρακτήρων με χρήση πολλαπλών τεχνικών μηχανικής μάθησης, Διπλωματική εργασία του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, Αθήνα, Ελλάδα

Li N. (1991), An Implementation of OCR System Based on Skeleton Matching, Computing Laboratory of University of Kent, Canterbury, United Kingdom

Mori S., Suen C.Y., Yamamoto K. (1992), Historical review of OCR research and development, Proceedings of the IEEE conference of 1992, Vol. 80, No. 7, pp. 1029-1058

Wikipedia (As seen in December 2012), “Optical character recognition”

Wikipedia (As seen in December 2012), “Automatic number plate recognition”