ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Automatic speech reading

ΓΙΑΝΝΗΣ ΜΥΛΟΓΙΑΝΝΑΚΗΣ
ΑΜ: 1998030500

ΤΡΙΜΕΛΕΣ ΕΠΙΤΡΟΠΗ: ΑΛΕΞΑΝΔΡΟΣ ΠΟΤΑΜΙΑΝΟΣ
ΒΑΣΙΛΕΙΟΣ ΔΙΓΑΛΑΚΗΣ
ΜΙΧΑΛΗΣ ΖΕΡΒΑΚΗΣ

Στους γονείς μου.

1ο κεφάλαιο: Εισαγωγή

2ο κεφάλαιο: Frond End

3ο κεφάλαιο: Αλγόριθμοι του ROI

Ευχαριστίες

Ευχαριστώ πολύ τον Καθηγητή Αλέξανδρο Ποταμιάνο για τη στήριξη που μου πρόσφερε με τις γνώσεις του και κυρίως την υπομονή του.

Ευχαριστώ και τους υπόλοιπους καθηγητές.

Επίσης ευχαριστώ τους φίλους μου και την οικογένειά μου για την προσφορά τους στον ψυχολογικό τομέα.

Συνοπτικά

Στο πλαίσιο της προσπάθειας ανάπτυξης της αυτόματης αναγνώρισης οπτικοακουστικής ομιλίας (Audio-Visual Automatic Speech Recognition, AVASR), 36 άτομα προφέρουν στην Αγγλική γλώσσα τα ψηφία από 0 έως 9 μπροστά σε μια κάμερα παίρνοντας παράλληλα διάφορες πόζες στο πρόσωπο. Κατόπιν αναλύονται τα δεδομένα του βίντεο σε καρέ-καρέ και διαχωρίζονται οι φωτογραφίες και ο ήχος σε κάθε ομιλητή με τη βοήθεια του MAT LAB. Στη συνέχεια το σύστημα της αυτόματης αναγνώρισης κάνει την προσπάθεια να αναγνωρίσει τις λέξεις με τη βοήθεια ενός προγράμματος που είχε δημιουργηθεί γι' αυτό το σκοπό, το **HTK TOOLKIT**, με βάση τον ήχο και την εικόνα του βίντεο και να φτάσει σε ικανοποιητικό επίπεδο την ευστοχία της αναγνώρισης.

Στην ακουστική αναγνώριση έγινε εξαγωγή 13 cepstral coefficients, που σε μορφή MFFC_D_A (χωρίς ενέργεια) γίνονταν συνολικά 39 τα coefficients και η ευστοχία είχε φτάσει στο 98% στο baseline πείραμα.

Στην οπτική αναγνώριση, όπου χρησιμοποιείται η Ευκλείδεια απόσταση για το ROI στα χείλια, έγινε εξαγωγή 35 πρώτων coefficients 2D-DCT από μάσκα 16x16, που σε μορφή MFFC_D (χωρίς ενέργεια) γίνονταν 70 συνολικά τα coefficients. Σημειώτεον πως τα features αναδειγματοποιούνταν από 29.97 fps στα 100 fps. Η ευστοχία κυμαινόταν στο 25%.

Η συγκεκριμένη διπλωματική εργασία αφορά στους αλγορίθμους της επεξεργασίας της μάσκας ROI (Region of Interest), η οποία βρίσκεται συγκεκριμένα στο στόμα του προσώπου των ομιλητών. Δηλαδή, οι αλγόριθμοι αφορούν στην αυτόματη αναγνώριση οπτικής ομιλίας (Automatic Recognition of Visual Speech, ARVS).

Αναπτύχθηκαν λοιπόν 3 αλγόριθμοι βασισμένοι στην ενέργεια των MFFC coefficients με σκοπό τη βελτίωση ευστοχίας της αναγνώρισης.

Στον πρώτο αλγόριθμο χρησιμοποιείται το DCT με βάση τη μεγαλύτερη ενέργεια. Η ευστοχία κυμαίνεται στο 25-26%.

Στο δεύτερο αλγόριθμο χρησιμοποιείται το DCT με βάση την ενέργεια των περιττών στηλών. Η ευστοχία κυμαίνεται στο 40%.

Στον τρίτο αλγόριθμο χρησιμοποιείται το DCT με βάση την ενέργεια των περιττών-αρτίων στηλών. Η ευστοχία κυμαίνεται στο 37%.

1ο Κεφάλαιο

ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Η ομιλία είναι τα δεδομένα της γλωσσικής συμπεριφοράς από συγκεκριμένους ομιλητές σε συγκεκριμένο τόπο και χρόνο. Η παραγωγή του ανθρώπινου λόγου ξεκινάει με την ιδέα ή τη σκέψη την οποία θέλει ο ομιλητής να μεταδώσει στον ακροατή. Ο ομιλητής μετατρέπει αυτή τη σκέψη σε μια σειρά από νευρολογικές επεξεργασίες για να παραγάγει ένα ηχητικό κύμα που θα ληφθεί από το ακουστικό σύστημα του ακροατή, το οποίο μετασχηματίζεται σε νευρολογικό σήμα. Πιο συγκεκριμένα, η ανθρώπινη ομιλία είναι ένα πολυδιάστατο σήμα με ακουστική και οπτική συνιστώσα. Η παρουσία και των δύο συμβάλλει στην καλύτερη ποιότητα επικοινωνίας. Μια πολύ σημαντική παράμετρος της οπτικής συνιστώσας είναι η κίνηση του στόματος, μιας και όλοι έχουμε την ικανότητα να διαπιστώνουμε ότι η κίνηση του στόματος του ανθρώπου που μιλά είναι συγχρονισμένη με την ομιλία.

Επικοινωνία είναι η διαδικασία ανταλλαγής πληροφοριών και μεταβίβασης μηνυμάτων από ένα άτομο σε άλλο (πομπός και δέκτης) μέσω συμβόλων, ήχων, αριθμών, χειρονομιών.

Υπάρχουν τρεις κύριες μορφές επικοινωνίας:

- Η λεκτική
- Η νοηματική (μη λεκτική)
- Η γραπτή

Η διαδικασία αυτή μπορεί να είναι μονοδρομική ή αμφιδρομική. Στη μονόδρομη επικοινωνία μεταβιβάζεται ένα μήνυμα προς κάποιο άτομο (δέκτης) χωρίς όμως να αναμένεται η ανταπόκριση, ενώ αντίθετα, στην αμφίδρομη επικοινωνία, όπου εμπλέκονται περισσότερα άτομα,

αποκωδικοποιούνται με σαφήνεια τα μηνύματα, υπάρχει ανατροφοδότηση και ανάδραση, με αποτέλεσμα τη γνήσια και αληθινή επικοινωνία. Η αποτελεσματικότητα της επικοινωνίας εξαρτάται από τις ιδιαίτερες δεξιότητες και ικανότητες του δέκτη να αποκωδικοποιεί τα μηνύματα που λαμβάνει.

Η εναλλαγή ρόλων (ο πομπός να γίνεται δέκτης και το αντίστροφο) κατά τη διάρκεια της επικοινωνίας, η αποτελεσματική αποκωδικοποίηση, η ανατροφοδότηση, η επανάδραση, είναι τα βασικά στοιχεία που εγγυώνται την εξασφάλιση μιας αληθινής και ορθής επικοινωνίας.

Η επικοινωνία που πραγματοποιείται μέσω της ομιλίας μπορεί να είναι λεκτική ή μη λεκτική. Η λεκτική επικοινωνία περιλαμβάνει φαινόμενα που αφορούν στον προφορικό και στο γραπτό λόγο. Με τον προφορικό λόγο μεταδίδονται περισσότερα και με μεγαλύτερη ταχύτητα μηνύματα από ό,τι με το γραπτό, επειδή ο προφορικός λόγος βοηθιέται από παραγλωσσικά γνωρίσματα, όπως οι παύσεις, ο τόνος της φωνής, ο επιτονισμός και τα γεμίσματα. Η μη λεκτική επικοινωνία περιλαμβάνει την έκφραση και την κίνηση του σώματος, τις αισθήσεις, τις εκφράσεις του προσώπου και των ματιών, τις χειρονομίες.

1.2 <u>Επικοινωνία μέσω λόγου</u>

Ο λόγος χρησιμοποιείται για να μεταδίδονται πληροφορίες από τον ομιλητή στον ακροατή. Η παραγωγή του ανθρώπινου λόγου, όπως είδαμε παραπάνω, ξεκινάει με την ιδέα ή τη σκέψη του ομιλητή και καταλήγει σε νευρολογικό σήμα στον εγκέφαλο του ακροατή. Για να το επιτύχει αυτό ο ομιλητής αναπτύσσει μια σκέψη, την οποία μετατρέπει σε λέξεις. Οι λέξεις αυτές είναι δομημένες σε προτάσεις και βασίζονται σε γραμματικούς κανόνες που ισχύουν στη γλώσσα που χρησιμοποιείται. Τέλος, συμπληρώνονται τα πρόσθετα τυπικά ή γενικά χαρακτηριστικά, όπως ο βαθμός τονισμού ή το άγχος για να δώσει έμφαση σε απόψεις σημαντικές για το συνολικό νόημα.

Ο εγκέφαλος παράγει μια σειρά εντολών που μετακινούν τους διάφορους μυς του φωνητικού συστήματος για να παράγουν το επιθυμητό ηχητικό κύμα. Αυτό το ακουστικό κύμα λαμβάνεται από το ακουστικό σύστημα του ομιλητή και μετασχηματίζεται σε μια αλληλουχία από νευρολογικούς παλμούς που εξασφαλίζουν την ανάδραση, η οποία είναι απαραίτητη για τη σωστή παραγωγή του λόγου. Το ακουστικό κύμα επίσης μεταβιβάζεται διαμέσου του αέρα στο ακουστικό σύστημα του ακροατή.

Η διαδικασία της αντίληψης του λόγου (της αναγνώρισης της ομιλίας) ξεκινάει όταν ο ακροατής εισπράττει τα ηχητικά κύματα με το εξωτερικό αυτί. Αυτά μετασχηματίζονται σε νευρολογικούς παλμούς στο μέσο και στο ενδότερο αυτί και αυτοί οι παλμοί διερμηνεύονται (κωδικοποιούνται με τη μορφή νευρικών σημάτων) μέσα στο ακουστικό κέντρο του εγκεφάλου για να προσδιοριστεί ποια ιδέα έχει λάβει. Μπορούμε να αντιληφθούμε ότι και στην παραγωγή λόγου αλλά και στην αντίληψη, το ανθρώπινο ακουστικό σύστημα παίζει σημαντικό ρόλο στην ικανότητα να επικοινωνεί αποτελεσματικά.

1.3 Μη λεκτική επικοινωνία

Η μη λεκτική επικοινωνία ορίζεται από τον τρόπο που κάποιος κινεί και χειρίζεται το σώμα του. Είναι δηλαδή, αυτό που συνήθως αποκαλείται «γλώσσα του σώματος». Περιλαμβάνει, αλλά δεν περιορίζεται μόνο σ' αυτά, τις κινήσεις των χεριών, των δακτύλων και τις γενικότερες θέσεις και κινήσεις του σώματος, το βλέμμα, τις κινήσεις και τις εκφράσεις του προσώπου, τις πρακτικές αγγίγματος και τους τυπικούς τρόπους χαιρετισμού.

Η αναγνώριση χειλιών ανήκει στην κατηγορία της μη λεκτικής επικοινωνίας. Πιο συγκεκριμένα, καθώς διατυπώνει ο ομιλητής την πρότασή του, για να σχηματίσει λέξεις χρησιμοποιεί το στόμα, τα δόντια και τη γλώσσα του. Αυτά τα στοιχεία (τα οποία συνθέτουν την αναγνώριση χειλιών) παρατηρεί ο δέκτης που 'λαμβάνει' το μήνυμα του πομπού για να μπορέσει να το καταλάβει. Γενικά, αρκετοί ομιλητές καταφεύγουν στην αναγνώριση χειλιών, συνήθως σε καταστάσεις όπου επικρατούν συνθήκες που εμποδίζουν τη λεκτική επικοινωνία.

Οι άνθρωποι με φυσιολογική όραση και ακοή ασυναίσθητα χρησιμοποιούν πληροφορίες από τα χείλια και το πρόσωπο για να βοηθηθούν στην ακουστική κατανόηση στην καθημερινή συζήτηση και κυρίως οι ευφραδείς ομιλητές μιας γλώσσας είναι ικανοί να διαβάζουν τα χείλια σε μεγάλη έκταση. Κάθε ήχος της ομιλίας (φώνημα) έχει συγκεκριμένη θέση στο πρόσωπο και στο στόμα. Όμως, πολλά φωνήματα μοιράζονται το ίδιο viseme και άρα είναι αδύνατον να αναγνωριστούν από εικαστικές πληροφορίες μόνο. Οι ήχοι που παράγονται μέσα στο στόμα ή στο φάρυγγα δεν είναι ανιχνεύσιμοι, όπως λόγου χάριν τα γλωσσικά σύμφωνα. Τα ηχηρά και μη-ηχηρά ζεύγη φαίνονται απαράλλακτα, για παράδειγμα τα [p] και [b], [k] και [g], [t] και [d], [f] και [v], [s] και [z]. Το ίδιο μπορούμε να παρατηρήσουμε και στην τροπή φθόγγου σε έρρινο.

Έχει εκτιμηθεί πως μόνο 30% - 40% από τους ήχους της Αγγλικής γλώσσας είναι ανιχνεύσιμοι μόνο με βάση την όραση. Ένα χαρακτηριστικό παράδειγμα: η φράση “where there’s life, there’s hope” μοιάζει με τη φράση “where’s the lavender soap” στις περισσότερες Αγγλικές διαλέκτους. Τότε ένα άτομο που αναγνωρίζει τα χείλια, πρέπει να χρησιμοποιήσει πρότυπα από το περιβάλλον και επίγνωση για τη φράση που εκστόμισε ο ομιλητής. Είναι αρκετά πιο εύκολο να διαβάσει τις συνηθισμένες φράσεις, όπως οι ευχές, από ό,τι τις φραστικές διατυπώσεις που εμφανίζονται σε απομόνωση και χωρίς υποστηρικτικές πληροφορίες, όπως το όνομα ενός ανθρώπου που δεν έχει συναντήσει ποτέ του στο παρελθόν.

Για την διπλωματική εργασία, θα μας απασχολήσει κυρίως η μη λεκτική επικοινωνία και πιο συγκεκριμένα η αναγνώριση χειλιών χωρίς τη βοήθεια των ήχων των λέξεων στην ομιλία.

1.4 ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ

Η αναγνώριση εικόνας (image recognition) ασχολείται με τεχνικές και μεθόδους που οδηγούν είτε στην ταξινόμηση (αντιστοίχηση) της εικόνας εισόδου σε μία συγκεκριμένη τάξη ή ομάδα εικόνων, είτε στην παραγωγή μιας περιγραφής της εικόνας.

Η αναγνώριση εικόνας ξεκινά από μία εικόνα και τη μετασχηματίζει σε μια συνοπτική περιγραφή, που μπορεί για παράδειγμα να οδηγήσει σε ένα σύνολο αριθμών, ένα σύνολο συμβόλων ή ένα γράφο. Η περαιτέρω επεξεργασία των ανωτέρω με βάση τις τεχνικές της αναγνώρισης προτύπων οδηγεί στην ταξινόμηση της αρχικής εικόνας. Μερικές εφαρμογές της αναγνώρισης εικόνας είναι η αυτόματη ιατρική διάγνωση που βασίζεται σε ιατρικές εικόνες, η αναγνώριση χαρακτήρων, η αναγνώριση ομιλίας κ.α. Στο συγκεκριμένο θέμα, το στόμα δίχως αμφιβολία μπορεί να χαρακτηριστεί ως σύμβολο ή χαρακτήρας για την οπτική αναγνώριση. Κάλλιστα μπορεί να χαρακτηριστεί και ως το πιο απαραίτητο και βασικό στοιχείο για την επίτευξη της αναγνώρισης.

1.4.1 Ιστορικό

Οι πρώτες προσπάθειες για την ανάπτυξη ενός συστήματος οπτικής αναγνώρισης χαρακτήρων έγιναν από τον Tausheck το 1929 και από τον Handel το 1933. Η μηχανή του Tausheck (όπως περιγράφεται στο δίπλωμα ευρεσιτεχνίας που απέκτησε) που αναγνώριζε τυπογραφικούς χαρακτήρες και νούμερα, βασιζόταν στη σύμπτωση ιχνών/μασκών (template/mask matching).

Η ιδέα αυτή προκύπτει από το αξίωμα της υπέρθεσης και περιγράφεται σαν το έβδομο αξίωμα στον πρώτο τόμο των "Στοιχείων'' του Ευκλείδη. Η μηχανή του Tausheck αποτελείται από μηχανικές μάσκες, οι οποίες έχουν τη μορφή συγκεκριμένων χαρακτήρων και μέσα από τις οποίες περνάει φως, το οποίο στη συνέχεια κατευθύνεται σε ένα φωτοανιχνευτή. Μεταξύ του φωτοανιχνευτή και των μηχανικών μασκών της συσκευής τοποθετείται το χαρτί. Όταν το φως δεν φθάνει στον φωτοανιχνευτή, τότε υπάρχει ακριβές ταίριασμα (matching) και η μηχανή αναγνωρίζει το χαρακτήρα εισόδου.

Ενώ για τους ανθρώπους τα σύμβολα A και ***A*** έχουν το ίδιο νόημα (αντιστοιχούν στην ίδια σύλληψη), η μηχανή του Tausheck δεν μπορεί να τα αναγνωρίσει, εκτός εάν στη μηχανή έχουν κατασκευασθεί και οι δύο μάσκες για το σύμβολο A. Ακόμη και σήμερα δεν έχει δοθεί γενική λύση στο πρόβλημα αυτό (δηλαδή στο πρόβλημα της αναγνώρισης διαφορετικών γραμματοσειρών με δεδομένο ότι το σύστημα έχει εκπαιδευτεί σε μία μόνο γραμματοσειρά). Αυτό αποτελεί και ένα από τα κεντρικά προβλήματα της έρευνας στην αναγνώριση χαρακτήρων. Έχουν προταθεί όμως αρκετές τεχνικές που επιλύουν μερικώς το πρόβλημα αυτό.

Με την κατασκευή των πρώτων υπολογιστών αρχίζουν να γίνονται περισσότερες προσπάθειες για την ανάπτυξη πιο αξιόπιστων συστημάτων οπτικής αναγνώρισης χαρακτήρων.

Η πρώτη γενιά τέτοιων συστημάτων εμφανίζεται στις αρχές του 1960 και χαρακτηρίζεται από αρκετούς περιορισμούς. Για παράδειγμα, τα συστήματα αυτά αναγνωρίζουν ένα μικρό σύνολο συμβόλων/χαρακτήρων που προέρχονται από μία γραμματοσειρά. Μερικά από αυτά είναι το NCR20 και διάφορα συστήματα της IBM, όπως το 1428, 1285, 1287, 1917, το N240D-1 της NEC και το H-852 της Hitachi.

Τα συστήματα οπτικής αναγνώρισης χαρακτήρων δεύτερης γενιάς εμφανίζονται στα μέσα της δεκαετίας του 1960 και έχουν δυνατότητες αναγνώρισης χειρόγραφων χαρακτήρων. Τέτοια συστήματα αναπτύχθηκαν από την IBM, την Toshiba και τη NEC.

Οι δύο τελευταίες εταιρείες κατασκευάζουν μηχανές που χρησιμοποιούνται από τα ταχυδρομεία για την αυτόματη διαλογή των γραμμάτων. Η διαλογή αυτή βασίζεται στην αναγνώριση του ταχυδρομικού κωδικού που είναι γραμμένος στους φακέλους.

Η τρίτη γενιά χαρακτηρίζεται από επιπλέον δυνατότητες, όπως η αναγνώριση τυπογραφικών χαρακτήρων πολλών γραμματοσειρών και χειρογράφων κειμένων, τα οποία περιέχουν μεγάλα σύνολα χαρακτήρων π.χ. κινέζικοι χαρακτήρες. Τέτοια συστήματα κατασκευάζονται από το 1975. Μερικά από αυτά είναι το ASPET/71 της Toshiba, το 1975 της IBM και το Katanaka της NTT που σήμερα ονομάζεται DT-OCR100C. Τα τελευταία χρόνια έχουν αναπτυχθεί OCR συστήματα υπό τη μορφή πακέτων λογισμικού χαμηλού κόστους, τα οποία λειτουργούν σε προσωπικούς υπολογιστές.

Τα τελευταία χρόνια όμως, η συγκεκριμένη επιστήμη έχει αναπτυχθεί ραγδαία τόσο που η αναγνώριση ήχου (φωνής) έχει φτάσει σε πολύ ικανοποιητικό επίπεδο, χάρη στην ανάπτυξη των γλωσσών προγραμματισμού αλλά και της τεχνολογίας, ενώ σε πολλά ερευνητικά ιδρύματα γίνονται ήδη προσπάθειες για τη βελτίωση της αποτελεσματικότητας της αναγνώρισης εικόνας.

1.4.2 <u>Εφαρμογές</u>

Για την αναγνώριση χαρακτήρων τα προηγούμενα χρόνια, όταν δεν υπήρχαν ακόμη ηλεκτρονικοί υπολογιστές αλλά και στις εποχές που αυτοί δεν χρησιμοποιούνταν ευρέως, μεγάλος όγκος πληροφοριών καταγράφονταν στο χαρτί. Έτσι, σήμερα υπάρχει τεράστιος όγκος πληροφοριών που είναι καταχωρημένες σε χαρτί και πολλές φορές παρουσιάζεται η ανάγκη από τον άνθρωπο για την εισαγωγή τέτοιων πληροφοριών στον Η/Υ. Η εισαγωγή αυτή κρίνεται απαραίτητη σε διάφορες εφαρμογές λόγω του χαμηλού κόστους διατήρησης, της δυνατότητας εύκολης αντιγραφής και επομένως αντιμετώπισης της φθοράς, αλλά και του μικρού χρόνου επεξεργασίας και ανάκτησης των πληροφοριών που επιτυγχάνεται με τη βοήθεια των Η/Υ. Με βάση τα παραπάνω, φαίνεται η ανάγκη ανάπτυξης και χρήσης των συστημάτων οπτικής αναγνώρισης χαρακτήρων.

Μία από τις βασικές εφαρμογές της αναγνώρισης χαρακτήρων είναι η αυτόματη εισαγωγή κειμένων σε ηλεκτρονική μορφή (π.χ. ASCII σύμβολα) στον ηλεκτρονικό υπολογιστή.

Ένα σύστημα οπτικής αναγνώρισης χαρακτήρων μπορεί να βοηθήσει στη γρήγορη εισαγωγή κειμένων (από βιβλία, περιοδικά, έγγραφα) μειώνοντας τις περισσότερες φορές το χρόνο και το κόστος δακτυλογράφησης. Έτσι, ένα σύστημα οπτικής αναγνώρισης χαρακτήρων μπορεί να θεωρηθεί σαν το πρώτο στάδιο εφαρμογών, όπως για παράδειγμα μίας βάσης δεδομένων, όπου η εισαγωγή πληροφοριών στα πεδία της βάσης γίνεται αυτόματα με τη χρήση δελτίων που περιέχουν χειρόγραφους ή τυπογραφικούς χαρακτήρες, ενός συστήματος αρχειοθέτησης εγγράφων ή ενός συστήματος ελεύθερης ανάκτησης κειμένων (full text retrieval).

Μερικές άλλες τυπικές εφαρμογές της αναγνώρισης χαρακτήρων είναι η εισαγωγή δελτίων οργανισμών (π.χ. Βιβλιοθηκών, Στατιστικών Υπηρεσιών, Οικονομικών Εφοριών κ.α.) στον Η/Υ, η αυτόματη ταξινόμηση φακέλων (π.χ. ταχυδρομικών φακέλων) και εγγράφων με βάση κωδικούς που είναι καταγεγραμμένοι σε αυτά, η εισαγωγή προσωπικών εγγράφων και σημειώσεων στον Η/Υ χωρίς τη χρήση πληκτρολογίου, η αναγνώριση των χαρακτήρων που είναι τοποθετημένοι στις πινακίδες των αυτοκινήτων με σκοπό την αυτόματη χρέωση των κατόχων των αυτοκινήτων σε χώρους στάθμευσης ή ακόμη για επιβολή προστίμου σε περιπτώσεις παράβασης του κώδικα οδικής κυκλοφορίας σε συγκεκριμένα σημεία των δρόμων κ.α.

Στην εποχή μας, που χαρακτηρίζεται ως μεταβατική από την εποχή του χαρτιού στην εποχή της ηλεκτρονικής αρχειοθέτησης, η αναγνώριση χαρακτήρων είναι μία σημαντική εφαρμογή της αναγνώρισης προτύπων, γιατί θα βοηθήσει στην ηλεκτρονική αρχειοθέτηση πληροφοριών που έχουν παραχθεί κατά τους προηγούμενους αιώνες.

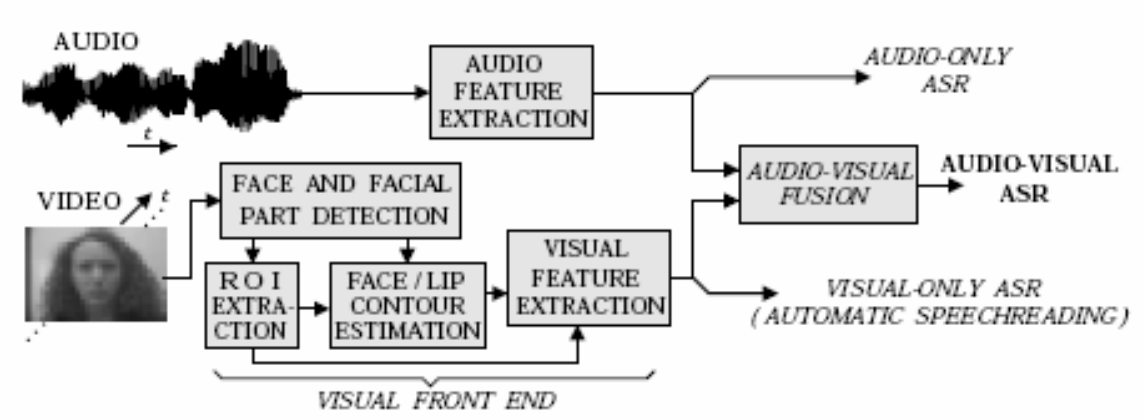
2ο Κεφάλαιο

FRONT END

2.1 Γενικά – Audio visual front end

Το Audio-visual front end είναι η διαδικασία στην οποία το σύστημα επεξεργάζεται ένα βίντεο (στην περίπτωση μας είναι ένα βίντεο με 36 ομιλητές) με τη βοήθεια του ήχου και της εικόνας, 'ακούει' και 'διαβάζει' τα λόγια των ομιλητών.

Το audio-visual front end χωρίζεται σε 2 μέρη: το audio front end και το visual front end.



ΣΧΗΜΑ 1

Όπως φαίνεται στο σχήμα 1, στο audio front end, το μικρόφωνο καταγράφει τον ήχο του βίντεο το οποίο είναι το ηχητικό μήνυμα του πομπού και αποθηκεύεται σε αρχείο με την μορφή του ήχου (wmv, mpeg). Στη συνεχεία αναλύεται συνήθως με τις μεθόδους της ψηφιακής επεξεργασίας σήματος, με διαφόρους μετασχηματισμούς. Στη συνέχεια εξάγει τα ηχητικά χαρακτηριστικά από τα αρχεία ήχου με τη βοήθεια των μεθόδων που αναφερθήκαν παραπάνω, και τέλος με τη βοήθεια της εκπαίδευσης ('εκμάθησης') του συστήματος συγκρίνει το άγνωστο προφίλ με τα προφίλ των ακουστικών μονάδων που έχει στη βάση δεδομένων του και παίρνει την τελική απόφαση για τα αποτελέσματα.

Έτσι, στο τέλος μπορεί να τα συνδυάσει με τα αποτελέσματα του visual front end για να προκύψει η αποτελεσματικότητα/ευστοχία όλου του audio-visual front end. Όσο για το visual front end, είναι λίγο διαφορετική η διαδικασία: αναλύει τα καρέ του βίντεο, εντοπίζει την 'ενδιαφέρουσα' περιοχή (Region of Interest, ROI), συγκεκριμένα το πρόσωπο, και με διάφορους αλγορίθμους-μετασχηματισμούς εξάγει τα εικονικά χαρακτηριστικά του ROI και τέλος με τη βοήθεια του σταδίου της εκμάθησης, κάνει συγκρίσεις των χαρακτηριστικών και παίρνει την τελική απόφαση για τα αποτελέσματα.
Σ' αυτό το κεφάλαιο θα μας απασχολήσει μόνο το visual front end.

2.2 Visual front end

Το visual front end όπως αναφέρθηκε παραπάνω χρησιμοποιείται για την οπτική αναγνώριση.

Η διαδικασία του visual front end με τη σειρά είναι το face detection και facial part detection (εντοπισμός προσώπου και τα μέρη του), ROI (Region of interest) και visual feature extraction (εξαγωγή χαρακτηριστικών από το ROI και η επεξεργασία των χαρακτηριστικών τους για να παρθεί στο τέλος η τελική απόφαση για την αναγνώρισή τους). Η διαδικασία περιγράφεται αναλυτικά παρακάτω.

2.2.1 Face detection – Region of Interest

Για την εντόπιση του προσώπου και των μερών του υπάρχουν πολλά συστήματα που χρησιμοποιούν παραδοσιακά τεχνάσματα για τον εντοπισμό προσώπου και την επεξεργασία εικόνας, όπως για παράδειγμα η κατάτμηση των χρωμάτων, η εντόπιση ακμών, το κατώφλι της εικόνας, το ταίριασμα ταμπλετών, ενώ άλλα συστήματα βασίζονται σε στατιστικά μοντέλα προσέγγισης, όπως λόγου χάριν προσλαμβάνοντας νευρικά κυκλώματα.

Γενικά, όλα τα συστήματα audio-visual ASR πρέπει καταρχήν να προσδιορίσουν μια **'ενδιαφέρουσα περιοχή'** (Region of Interest, ROI). Για παράδειγμα, μπορεί να είναι το πρόσωπο, όπου μια μάσκα ROI μπορεί να χρησιμοποιηθεί για να ταιριάξει με ακρίβεια το μέρος του προσώπου, ή μπορεί να είναι μόνο το στόμα όπου μπορεί να χρησιμοποιηθεί ένα μοντέλο χειλιών για να ταιριάξει τις καμπύλες των χειλιών. Σε μερικά συστήματα τα χρώματα του ομιλητή εντοπίζονται είτε με τη βοήθεια των μασκών είτε χωρίς και τονίζονται, οπού τα χαρακτηριστικά του στόματος εξάγονται με χρωματικές μεθόδους.

Υπάρχουν διάφοροι αλγόριθμοι που έχουν προταθεί σ' αυτό το θέμα τα 20 τελευταία χρόνια. Γενικά μπορούν να χωρισθούν σε τρεις κατηγορίες:

α) Video pixel
β) Καμπύλες
γ) Συνδυασμός των παραπάνω.

Η σύμπτωση Ιχνών (template matching) είναι σίγουρα η πιο απλοϊκή μορφή αναγνώρισης προτύπων. Στην τεχνική της σύμπτωσης ιχνών για κάθε τάξη αποθηκεύεται στη μνήμη της μηχανής ένα ίχνος ή μάσκα ή πρωτότυπο. Το πρότυπο εισόδου συγκρίνεται με το πρωτότυπο κάθε τάξης και η ταξινόμηση βασίζεται σε ένα προαποφασισμένο κριτήριο ομοιότητας (similarity or matching criterion). Η απόφαση λαμβάνεται για ταξινόμηση σε εκείνη την τάξη για την οποία η ομοιότητα είναι μεγαλύτερη. Δυστυχώς πολλές φορές είναι δύσκολο να διαλέξει κανείς "καλά πρωτότυπα" όπως επίσης και "καλά κριτήρια ομοιότητας". Μερικές τεχνικές "ελαστικής" σύγκρισης με πρωτότυπα έχουν προταθεί για να αντιμετωπισθεί κάποια μεταβλητότητα. Στα όρια αυτών των τεχνικών φθάνουμε στις κλασσικές μεθοδολογίες αναγνώρισης προτύπων που εργάζονται με τεμαχισμό του χώρου προτύπων.

Οι δυσκολίες των τεχνικών της αναγνώρισης προτύπων εμφανίζονται κυρίως όταν στην εικόνα το φόντο ή η πόζα του προσώπου και τα φώτα είναι πολύ διαφορετικά και έτσι τα χαρακτηριστικά γίνονται περίπλοκα και ασαφή με αποτέλεσμα να μην βγαίνουν ικανοποιητικά τα αποτελέσματα. Για να αυξηθεί η αποτελεσματικότητα τα πρόσωπα στήνονται μετωπικά μπροστά στην κάμερα με καθαρή πόζα και άριστη φωτεινότητα και προσπαθούν να ομιλούν καθαρά.

ΣΧΗΜΑ 2
Εντοπισμός του προσώπου και ο ορισμός της 'ενδιαφέρουσας περιοχής' (Region of Interest)

Στο πείραμά μας χρησιμοποιείται η ευκλείδεια απόσταση για την προσέγγιση του ROI στο πρόσωπο ώστε να εστιαστεί το σύστημα στο στόμα του ομιλητή.

Ευκλείδεια απόσταση

Η Ευκλείδεια απόσταση είναι η μέση απόσταση ανάμεσα σε 2 σημεία. Για 2 μονοδιάστατα σημεία, έστω $P=(p_x)$ και $Q=(q_x)$, η απόσταση υπολογίζεται με τον τύπο: $\sqrt{(p_x-q_x)^2}=|p_x-q_x|$.

Για δύο 2-D σημεία, $P=(p_x,p_y)$ και $Q=(q_x,q_y)$, η απόσταση υπολογίζεται με τον τύπο: $\sqrt{(p_x-q_x)^2+(p_y-q_y)^2}$.

Στο συγκεκριμένο πείραμα, για τη συγκεκριμένη ROI και την εικόνα, βρίσκει τα αρχικά x, y σημεία σε μια όμοια σε μέγεθος περιοχή της εικόνας που ελαχιστοποιεί την ευκλείδεια απόσταση των RGB μεταξύ του ζητούμενου και του δεδομένου ROI. Εφαρμόζονται οι 16x16, 20x20, 24x24, 28x28, και 32x32 μάσκες ROI.

2.2.2 Visual feature extraction

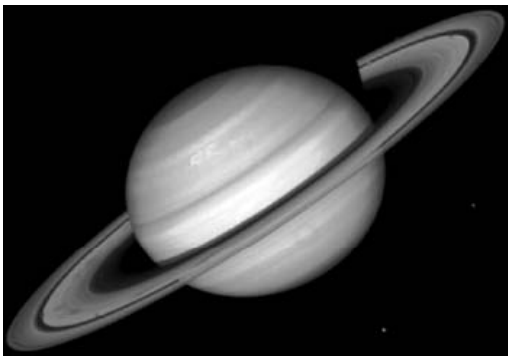
Τα εικονικά χαρακτηριστικά που εξάγονται από το ROI συμπιέζονται και μετασχηματίζονται στη συνέχεια με διάφορους μετασχηματισμούς. Οι πιο δημοφιλείς είναι το Principal components analysis **(**PCA), Discrete Cosine Transform (DCT), Discrete wavelet transform, Hadamard και Haar transforms και linear discriminant analysis (LDA). Θα ασχοληθούμε μόνο με το DCT που χρησιμοποιείται στο baseline πείραμα.

2.2.2.1 Διακριτός Συνημιτονικός Μετασχηματισμός

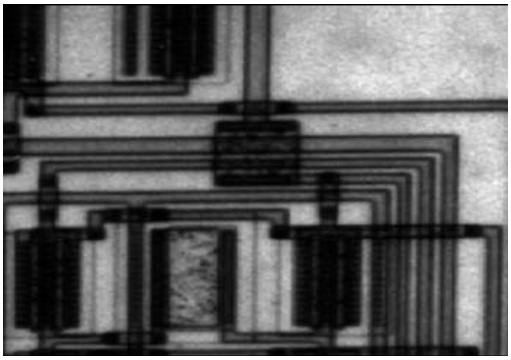
Ο **Διακριτός Συνημιτονικός Μετασχηματισμός** (Discrete Cosine Transform, DCT) είναι μία μέθοδος που βρίσκει μεγάλη εφαρμογή στην ψηφιακή συμπίεση γενικά, αλλά και στο MPEG ειδικότερα. Με το μετασχηματισμό DCT μπορούμε να μεταφέρουμε την πληροφορία που περικλείει η εικόνα από το πεδίο του χώρου στο πεδίο της συχνότητας (αφηρημένο πεδίο), δηλαδή κωδικοποιείται κάθε στοιχείο της εικόνας με μία μονή σάρωση από αριστερά προς τα δεξιά και από πάνω προς τα κάτω,

όπου η περιγραφή της μπορεί να γίνει με σημαντικά μικρότερο πλήθος bits (συμπίεση με απλά λόγια), για διάφορους λόγους.

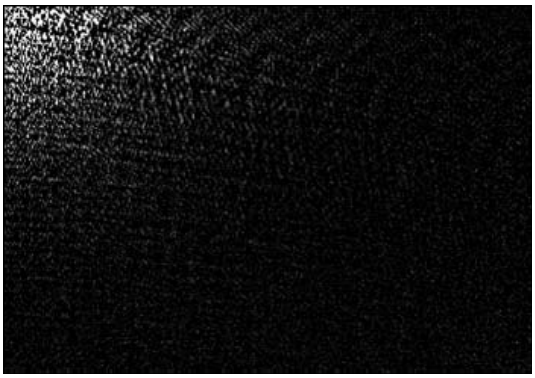
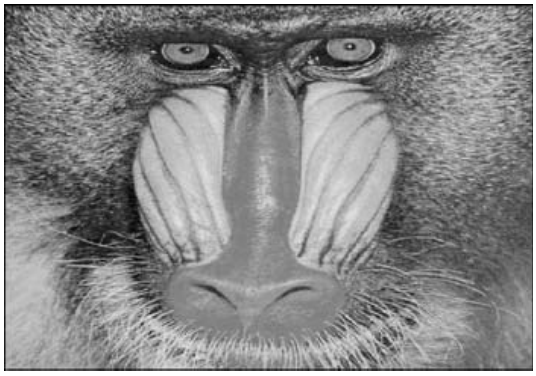
Δηλαδή, η βασική ιδέα είναι ο μετασχηματισμός δεδομένων σε κάποιον άλλο μαθηματικό χώρο, ο οποίος προσφέρεται καλύτερα για συμπίεση. Οι πρώτες συχνότητες της ενέργειας στο σύνολο έχουν τη μεγαλύτερη σπουδαιότητα, ενώ οι τελευταίες τη μικρότερη. Για παράδειγμα, δίνονται οι παρακάτω φωτογραφίες που μετασχηματίζονται με το μετασχηματισμό DCT (η πρώτη στήλη) και η ενέργεια των συχνοτήτων (η δεύτερη στήλη):

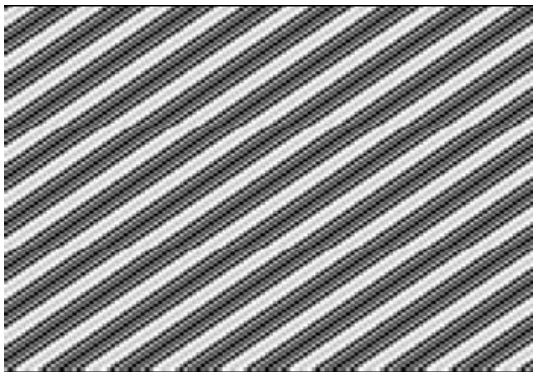
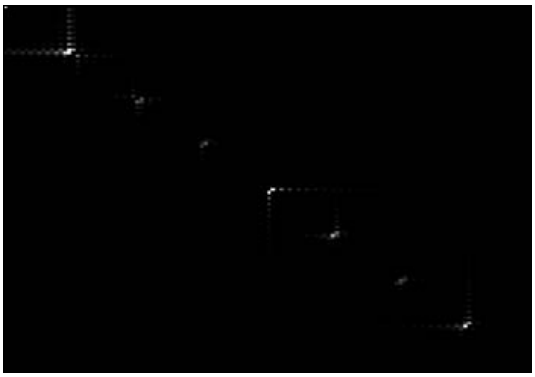








Όταν συμπιέζουμε ένα μέρος των τελευταίων αυτών συχνοτήτων, χάνεται, αναλόγως της ανοχής που έχουμε θέσει για την ποιότητα.

Ο μετασχηματισμός DCT ορίζεται ως εξής :

Για κάθε pixel (x,y) εφαρμόζοντας τον τύπο :

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{k-1} \sum_{y=0}^{l-1} pixel(x,y) \cos\left[\frac{2x+1}{2N}\right] \cos\left[\frac{2y+1}{2N}\right]$$

όπου C(x) = 0.7071, x =0

1 , x >0

παίρνουμε την τιμή DCT(i,j) που είναι η τιμή του συντελεστή του μετασχηματισμού στο πεδίο της συχνότητας. Έτσι αντιστοιχίζουμε τις τιμές των pixels στις αντίστοιχες τιμές συντελεστών.

Οι συντελεστές αυτοί μεταφέρουν ο καθένας ένα κομμάτι της αρχικής πληροφορίας (αυτό που αντιστοιχεί στο κομμάτι του φάσματος που περιγράφει). Επειδή όμως έχει παρατηρηθεί ότι η ανθρώπινη όραση αντιλαμβάνεται πολύ περισσότερο τα φαινόμενα που σχετίζονται με χαμηλές συχνότητες, όπως π.χ. χρώματα με μικρότερα μήκη κύματος, ενώ δείχνει κάποια ανοσία σε υψίσυχνες περιοχές του σήματος (π.χ. ακμές της εικόνας), οι συντελεστές του μετασχηματισμού που αντιστοιχούν σε χαμηλές συχνότητες έχουν μεγαλύτερη βαρύτητα από αυτούς που περιγράφουν τις υψηλές συχνότητες και για το λόγο αυτό οι πρώτοι περιγράφονται με τη μεγαλύτερη δυνατή ακρίβεια.

Κατά την αναπαραγωγή γίνεται η αντίστροφη διαδικασία με τη βοήθεια του μετασχηματισμού **IDCT** (Inverse Discrete Cosine Transform - Αντίστροφος Διακριτός Μετασχηματισμός Συνημίτονων) , που περιγράφεται από τον τύπο:

$$Pixel(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i) C(j) DCT(i,j) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

Το αποτέλεσμα είναι να πάρουμε πίσω σχεδόν ανέπαφη την αρχική πληροφορία (εκτός από κάποια αναπόφευκτα σφάλματα στρογγυλοποίησης).

3ο Κεφάλαιο

Υλοποίηση του Visual Front End

3.1 Γενικά

Η υλοποίηση του visual front end έγινε στο περιβάλλον του MAT LAB με τη βοήθεια μερικών script με την Perl. Το κεντρικό αρχείο είναι το visual_front_end.m και τα υπόλοιπα αρχεία είναι τα find_roi_match_euclidean_fixed.m, savehtk.m και roi_feature_extract1.m που εξυπηρετούν διάφορες εργασίες που είναι απαραίτητες για το visual front end.

3.2 Αρχεία

3.2.1 Visual_front_end.m

Αφού το βίντεο του κάθε ομιλητή έχει διαχωριστεί σε αρχεία .png που είναι και τα καρέ του βίντεο και έχει ρυθμιστεί το μέγεθος του ROI, παίρνει το πρώτο frame του ομιλητή. Τότε 'τοποθετεί' την ταμπλέτα του ROI στο πρώτο frame με την Ευκλείδεια απόσταση (find_roi_match_euclidean_fixed.m) και κάνει την εξαγωγή των εικονικών χαρακτηριστικών (coefficients) του ROI (roi_feature_extract1.m). Κατόπιν τα αποθηκεύει σε μορφή .mfcc (savehtk.m), προκειμένου να γίνει η επεξεργασία τους στη συνέχεια με το HTK toolkit, και προχωράει στο επόμενο frame του βίντεο. Όλα αυτά γίνονται με loop για όλα τα frames του ομιλητή. Αυτό επαναλαμβάνεται και για τους 36 ομιλητές του βίντεο.

3.2.2 Find_roi_match_euclidean_fixed.m

Για το ROI, βρίσκει τις συντεταγμένες του x και y της ταμπλέτας που θα ελαχιστοποιήσουν την ευκλείδεια RGB απόσταση του δεδομένου και του ζητούμενου ROI. Το μέγεθος του ROI καθορίζεται από το visual_front_end.m ανάλογα δηλαδή με τον αλγόριθμο, όλα τα μεγέθη για τη διπλωματική εργασία είναι 16x16, 20x20, 24x24, 28x28 και 32x32.

3.2.3 roi_feature_extract1.m

Αρχικά μετατρέπει το έγχρωμο RGB frame σε ασπρόμαυρο. Έπειτα με τα blocks του ROI κάνει εξαγωγή (extract) των τιμών των pixels. Τέλος, χρησιμοποιεί αλγόριθμο βασισμένο στο μετασχηματισμό Διακριτό Συνημιτονικό Μετασχηματισμό (οι αλγόριθμοι αναλύονται παρακάτω) για να μετασχηματίσει τις τιμές των pixels.

3.2.4 Savehtk.m

Μετά από την εκτέλεση του roi_feature_extract1.m, αποθηκεύει τις τιμές που έχουν μετασχηματιστεί με το Διακριτό Συνημιτονικό σε μορφή .MFCC για να γίνει επεξεργασία στη συνέχεια με το HTK Tool.

3.2.5 HTK Tool

Μετά από την εξαγωγή εικονικών χαρακτηριστικών, η επεξεργασία τους και η διάγνωσή τους γίνεται με τη βοήθεια του **HTK toolkit,** καθώς και τα scripts της προγραμματιστικής γλώσσας Perl**.**

Το HTK toolkit είναι ένα εργαλείο για να δημιουργούνται κρυφά μοντέλα Markov (HMM).

Κρυφά μοντέλα Markov

Στη μοντελοποίηση Markov υποθέτουμε ότι υπάρχει μια οικογένεια μοντέλων με συγκεκριμένη δομή που περιγράφει ικανοποιητικά τις κατηγορίες τάξεων, διαφοροποιώντας τις τιμές των παραμέτρων των μοντέλων αυτών. Η δομή των μοντέλων αυτών αποτελείται από ένα σύνολο καταστάσεων που συνδέονται μεταξύ τους με:

1) Έναν πίνακα πιθανότητας μετάβασης από κατάσταση σε κατάσταση
2) Ένα διάνυσμα πιθανότητας αρχικής κατάστασης
3) Ένα σύνολο συνεχών πυκνοτήτων πιθανοτήτων, που συνδέουν τις καταστάσεις με τις πειραματικά μετρημένες τιμές, ή έναν πίνακα πιθανοτήτων παρατηρούμενων τιμών (οπότε έχουμε διακριτές συναρτήσεις πυκνότητας πιθανότητας, συνήθως εξαιτίας του κβαντισμού).

Για την εκτίμηση των παραμέτρων αυτών, χρησιμοποιείται στο στάδιο εκμάθησης, ένα μεγάλο σύνολο εκφωνήσεων.

Στη διαδικασία της εκμάθησης υπολογίζεται το μοντέλο κάθε λέξης και υποτίθεται ότι με τον τρόπο αυτό, συλλαμβάνεται η ιδιαιτερότητά της. Το στάδιο της εκμάθησης έχει μεγάλο υπολογιστικό φορτίο σε σχέση με τις άλλες τεχνικές. Στο στάδιο της εξέτασης υπολογίζεται για κάθε μοντέλο αναφοράς η πιθανότητα να έχει "γεννήσει" την άγνωστη εκφώνηση.

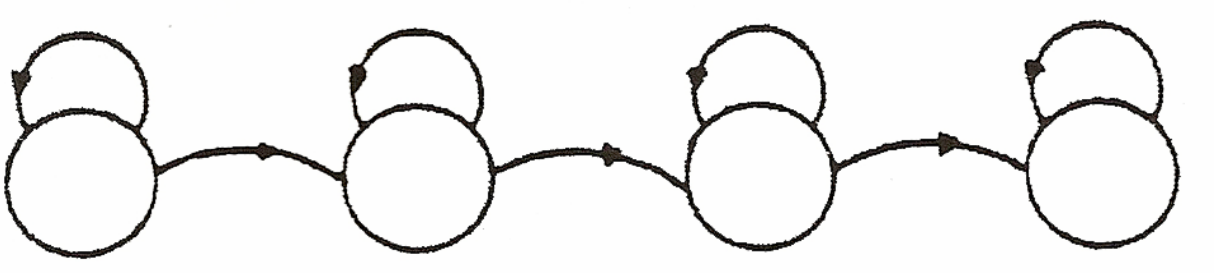
Ειδικότερα, η εκμάθηση συνδέεται άμεσα με το πρόβλημα προσαρμογής των μοντέλων Markov. Το πρόβλημα της προσαρμογής είναι το εξής: Δεδομένης μιας ακολουθίας Ω (ή ενός συνόλου από ακολουθίες παρατήρησης) και ενός τυχαίου κρυφού Μαρκοβιανού μοντέλου Μ, να επαναπροσδιορισθούν οι παράμετροι του Μ, ώστε να μεγιστοποιείται η πιθανότητα, τα δεδομένα εισόδου (η ακολουθία ή οι ακολουθίες) να έχουν γεννηθεί από αυτό το μοντέλο. Ένας από τους βασικούς αλγορίθμους προσαρμογής που χρησιμοποιείται στην αναγνώριση μεμονωμένων λέξεων, είναι ο αλγόριθμος Baum-Welch.

Από την άλλη πλευρά, η τεχνική σύγκρισης, συνδέεται με το πρόβλημα της εκτίμησης των κρυφών Μαρκοβιανών μοντέλων και ορίζεται ως εξής: Δεδομένης μιας ακολουθίας παρατηρούμενων συμβόλων $\Omega = o_1 o_2 ... o_\Gamma$ και ενός μοντέλου Μ, να υπολογιστεί η πιθανότητα $\left(\Pi(\Omega/\Mu)\right)$ (να έχει γεννηθεί η ακολουθία αυτή από το συγκεκριμένο μοντέλο).

Οι αλγόριθμοι εκτίμησης που χρησιμοποιούνται στο στάδιο της αναγνώρισης, είναι ο αλγόριθμος "μπρος – πίσω" (forward – backward) και ο αλγόριθμος Viterbi.

Για την εφαρμογή των κρυφών μοντέλων Markov, στην αναγνώριση μεμονωμένων λέξεων, ενδιαφέρον παρουσιάζει μια ειδική κατηγορία μοντέλων που τα ονομάζουμε "προσανατολισμένα", ή "από τα αριστερά προς τα δεξιά μοντέλα" (left to right). Τα μοντέλα αυτά έχουν τα εξής ιδιαίτερα χαρακτηριστικά:

α) Η αρχική κατάσταση είναι μοναδική και καθορισμένη (επιλέγεται η πρώτη από τις καταστάσεις του μοντέλου)
β) Η τελική κατάσταση είναι μοναδική και καθορισμένη (επιλέγεται η τελευταία κατάσταση του μοντέλου)
γ) Από τη στιγμή που το μοντέλο Markov φεύγει από μια κατάσταση δεν ξαναγυρίζει σε αυτήν.



Κρυφά Μοντέλα Markov.

Χρησιμοποιήθηκαν scripts της Perl για την εκτέλεση της δημιουργίας των HMM , και των εντολών HCopy, HInit, HCompv, HRest, HVite, HResults. Για το classification των παραμέτρων, χρησιμοποιήθηκαν τα HVite και HResults.

3.3 Αλγόριθμοι του ROI για το visual front end

3.3.1 Γενικά

Για την εξαγωγή εικονικών χαρακτηριστικών στο πείραμα χρησιμοποιήθηκαν διάφοροι αλγόριθμοι. Αυτοί οι αλγόριθμοι βασίστηκαν στο **Διακριτό Συνημιτονικό Μετασχηματισμό.** Όμως είναι διαφορετικοί στον τρόπο της εξαγωγής των coefficients. Παρακάτω παρουσιάζονται αναλυτικά οι αλγόριθμοι.

3.4 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση τη μεγαλύτερη ενέργεια

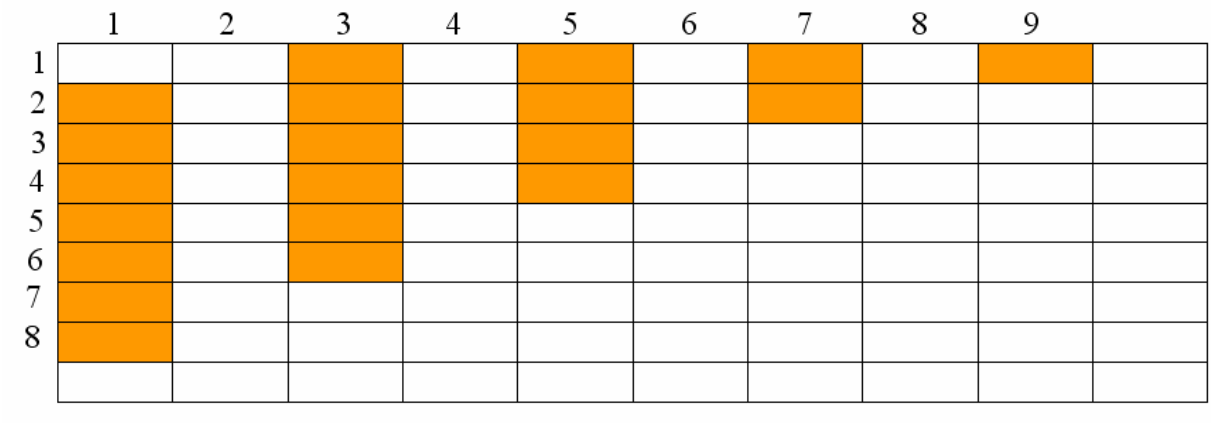
Ο Διακριτός Συνημιτονικός Μετασχηματισμός (DCT) όπως αναλύθηκε παραπάνω χρησιμοποιείται να συμπιέσει την εικόνα του στόματος του ομιλητή (δηλαδή το ROI) και οι μεγαλύτερες συχνότητες της ενέργειας στην εικόνα επιλέγονται σαν εικονικά χαρακτηριστικά για την επεξεργασία στη συνέχεια.

Για παράδειγμα, έστω υποθετικά είναι ένα ROI 20x20. Τότε θα υπάρχουν 20x20 pixels που σημαίνει ότι συνολικά τα pixels είναι 400. Κατόπιν επιλέγονται 20 (ή 30) coefficients που έχουν τη μεγαλύτερη ενέργεια των συχνοτήτων από τα 400 pixels.

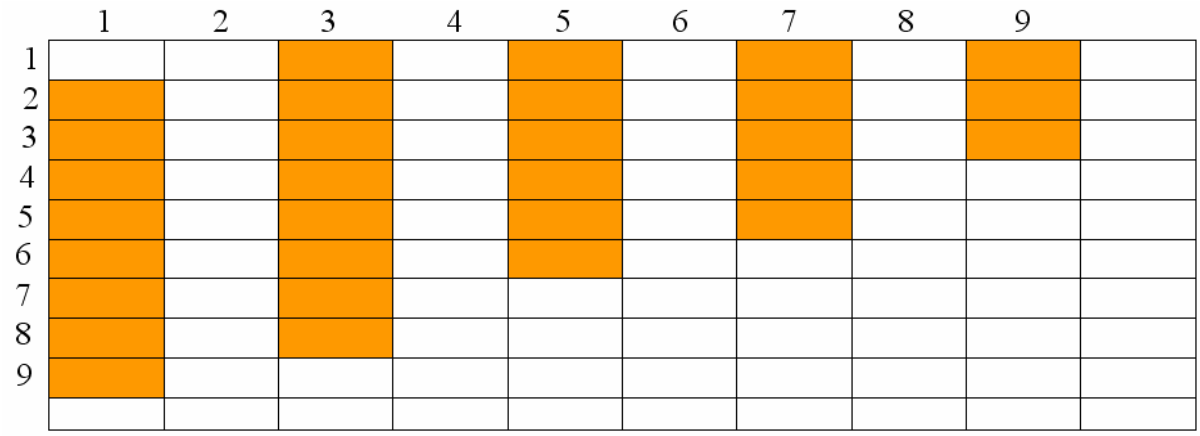
3.5 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση την ενέργεια των περιττών στηλών

Επιλέγουμε τις ενέργειες των περιττών στηλών του ROI. Τα coefficients των περιττών στηλών του ROI επιλέγονται με βάση τους παρακάτω πίνακες (τα έγχρωμα μέρη του ROI δείχνουν τα coefficients που επιλέγονται). Σημειώνεται πως το πάνω αριστερό block του ROI πάντα απορρίπτεται.

Για 20 features



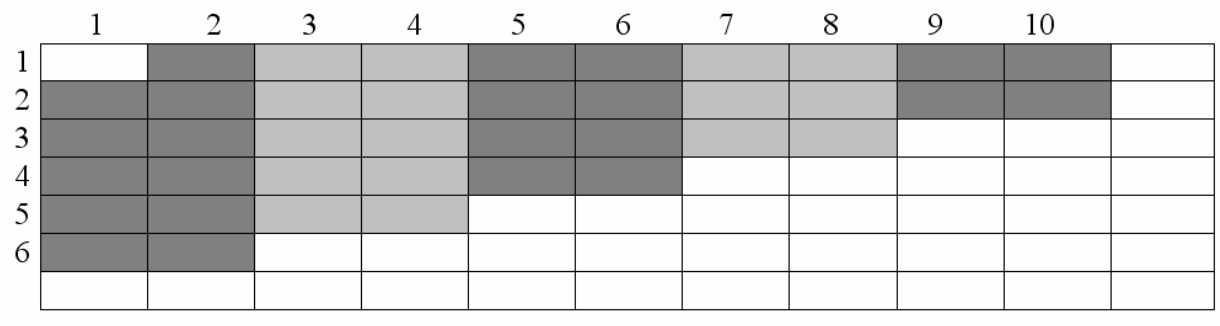
Για 30 features



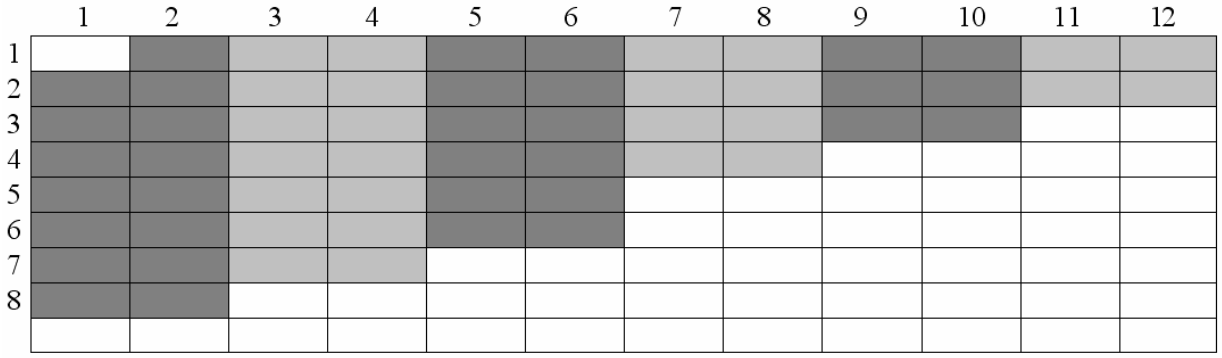
3.6 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση την ενέργεια των περιττών-άρτιων στηλών

Επιλέγουμε τη μεγαλύτερη ενέργεια από τη σύγκριση ανά γραμμή του πίνακα στα ζευγάρια περιττών-άρτιων στηλών του ROI. Τα ζευγάρια των περιττών-άρτιων στηλών του ROI επιλέγονται με βάση τους παρακάτω πίνακες (τα σκιαγραφημένα μέρη του ROI δείχνουν τα coefficients).

20 features



30 features



4ο Κεφάλαιο

Αποτελέσματα των Αλγόριθμων του ROI

Αρχικά έγινε εξαγωγή των 20 coefficients από τα ROI (16x16, 20x20, 24x24, 28x28, 32x32) που έγιναν 60 με τη δεύτερη παράγωγο των coefficients (1ος τρόπος εξαγωγής) και στη συνέχεια έγινε εξαγωγή των 30 coefficients που έγιναν 60 (2ος τρόπος εξαγωγής) και 90 coefficients (3ος τρόπος εξαγωγής) με την πρώτη παράγωγο και δεύτερη παράγωγο των coefficients αντίστοιχα.

4.1 Γενικά – Αναλυτικά Αποτελέσματα

Η αναλυτική μορφή των αποτελεσμάτων που εξάγονται από το πρόγραμμα HTK tool, σε έναν αλγόριθμο, δίνεται παρακάτω. Ουσιαστικά τα αποτελέσματα χωρίζονται σε 2 μέρη: το Training όπου έχει 30 ομιλητές και το Testing όπου προσπαθεί να αναγνωρίσει τα λόγια των 6 ‘αγνώστων’ ομιλητών. Η διαδικασία αυτή επαναλαμβάνεται δυο φορές, δηλαδή η πρώτη φορά είναι χωρίς classification στα features ενώ η δεύτερη φορά γίνεται με classification.

Το παράδειγμα είναι από τον αλγόριθμο Διακριτό Συνημιτονικό Μετασχηματισμό με βάση τη μεγαλύτερη ενέργεια με μέγεθος ROI στο 16x16.

Πριν από το classification (Training)

```
---------------------- Sentence Scores ----------------------
====================== HTK Results Analysis =======================
  Date: Wed Jan 17 14:15:03 2007
  Ref : labels/normalonlytrain
  Rec : results_training.mlf
------------------------ File Results --------------------------
s05f.rec:   24.00( 24.00)  [H=  12, D= 31, S=  7, I=  0, N= 50]
s06f.rec:   38.00( 30.00)  [H=  19, D= 14, S= 17, I=  4, N= 50]
s07m.rec:   32.00( 32.00)  [H=  16, D= 21, S= 13, I=  0, N= 50]
s08m.rec:   24.00( 24.00)  [H=  12, D= 33, S=  5, I=  0, N= 50]
s09m.rec:   38.00( 38.00)  [H=  19, D= 20, S= 11, I=  0, N= 50]
s10m.rec:   32.00( 26.00)  [H=  16, D= 21, S= 13, I=  3, N= 50]
s11f.rec:   14.00( 14.00)  [H=   7, D= 38, S=  5, I=  0, N= 50]
s12m.rec:   24.00( 24.00)  [H=  12, D= 31, S=  7, I=  0, N= 50]
s13m.rec:   14.00( 14.00)  [H=   7, D= 43, S=  0, I=  0, N= 50]
s14m.rec:   40.00( 26.00)  [H=  20, D= 20, S= 10, I=  7, N= 50]
s15f.rec:   40.00( 34.00)  [H=  20, D= 16, S= 14, I=  3, N= 50]
s16f.rec:   26.00( 26.00)  [H=  13, D= 19, S= 18, I=  0, N= 50]
s25f.rec:   50.00( 20.00)  [H=  25, D= 16, S=  9, I= 15, N= 50]
s19f.rec:   28.00( 28.00)  [H=  14, D= 26, S= 10, I=  0, N= 50]
s21f.rec:   24.00( 24.00)  [H=  12, D= 34, S=  4, I=  0, N= 50]
```

```
s03m.rec:   34.00( 30.00)  [H=  17, D= 12, S= 21, I=  2, N= 50]
s17m.rec:   32.00( 28.00)  [H=  16, D= 25, S=  9, I=  2, N= 50]
s22m.rec:   16.00( 16.00)  [H=   8, D= 42, S=  0, I=  0, N= 50]
s23f.rec:   32.00( 26.00)  [H=  16, D= 29, S=  5, I=  3, N= 50]
s24m.rec:   28.00( 28.00)  [H=  14, D= 20, S= 16, I=  0, N= 50]
s18f.rec:   28.00( 26.00)  [H=  14, D= 31, S=  5, I=  1, N= 50]
s26f.rec:   35.00( 25.00)  [H=  14, D= 17, S=  9, I=  4, N= 40]
s27m.rec:   28.00( 28.00)  [H=  14, D= 29, S=  7, I=  0, N= 50]
s28f.rec:   34.00( 24.00)  [H=  17, D= 25, S=  8, I=  5, N= 50]
s29m.rec:   32.00( 32.00)  [H=  16, D= 27, S=  7, I=  0, N= 50]
s30f.rec:   28.00( 28.00)  [H=  14, D= 31, S=  5, I=  0, N= 50]
s31m.rec:   30.00( 28.00)  [H=  15, D= 18, S= 17, I=  1, N= 50]
s32m.rec:   32.00( 24.00)  [H=  16, D= 22, S= 12, I=  4, N= 50]
s35m.rec:   44.00( 44.00)  [H=  22, D= 19, S=  9, I=  0, N= 50]
s36f.rec:   30.00( 26.00)  [H=  15, D= 28, S=  7, I=  2, N= 50]
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=30, N=30]
WORD: %Corr=30.34, Acc=26.58 [H=452, D=758, S=280, I=56, N=1490]
------------------------ Confusion Matrix -------------------------
       z   o   t   t   f   f   s   s   e   n
       e   n   w   h   o   i   i   e   i   i
       r   e   o   r   u   v   x   v   g   n
       o           e   r   e       e   h   e
                   e               n   t      Del [ %c / %e]
zero  51   5   2   4   6   1   2   0   1   1   76 [69.9/1.5]
 one   1  45   2   0   2   3   4   1   3   1   87 [72.6/1.1]
 two   3   1  52   1   6   1   7   2   0   1   75 [70.3/1.5]
thre   6   2   5  33   4   2   8   1   1   4   83 [50.0/2.2]
four   3   2   7   1  72   1   6   1   2   3   51 [73.5/1.7]
five   6   3   6   0   5  37   5   2   1   0   84 [56.9/1.9]
 six   3   1   6   1   5   1  54   2   0   2   74 [72.0/1.4]
seve   6   1   7   1   7   2   6  34   1   2   82 [50.7/2.2]
eigh   4   6   6   3  10   4   6   1  24   3   82 [35.8/2.9]
nine   3   6   7   6   8   2   2   1   0  50   64 [58.8/2.3]
 sil   0   0   0   0   0   0   0   0   0   0    0
Ins    7   4  10   3  18   4   4   1   3   2
```

Πριν από το classification (Testing):

```
------------------------ Sentence Scores --------------------------
====================== HTK Results Analysis =======================
  Date: Wed Jan 17 14:15:05 2007
  Ref : labels/normalonlytest
  Rec : results.mlf
------------------------- File Results ----------------------------
s01m.rec:   32.00( 28.00)  [H=  16, D= 17, S= 17, I=  2, N= 50]
s02m.rec:   38.00( 22.00)  [H=  19, D=  9, S= 22, I=  8, N= 50]
s04f.rec:   28.00( 28.00)  [H=  14, D= 32, S=  4, I=  0, N= 50]
s20f.rec:   18.00( 18.00)  [H=   9, D= 40, S=  1, I=  0, N= 50]
s33m.rec:   34.00( 24.00)  [H=  17, D= 28, S=  5, I=  5, N= 50]
s34f.rec:   30.00( 24.00)  [H=  15, D= 27, S=  8, I=  3, N= 50]
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=6, N=6]
WORD: %Corr=30.00, Acc=24.00 [H=90, D=153, S=57, I=18, N=300]
------------------------ Confusion Matrix -------------------------
       z   o   t   t   f   f   s   s   e   n
       e   n   w   h   o   i   i   e   i   i
       r   e   o   r   u   v   x   v   g   n
       o           e   r   e       e   h   e
```

```
                  e              n   t      Del [ %c / %e]
zero   8   1   1   0   0   1   4   0   0   0   15 [53.3/2.3]
 one   0   5   3   1   0   0   2   0   0   0   19 [45.5/2.0]
 two   0   1  10   0   0   0   2   0   0   0   17 [76.9/1.0]
thre   1   0   1  10   0   0   1   0   1   0   16 [71.4/1.3]
four   1   0   0   0  10   1   0   0   3   0   15 [66.7/1.7]
five   1   0   1   2   0  10   0   0   0   0   16 [71.4/1.3]
 six   0   0   1   0   0   0  14   0   0   0   15 [93.3/0.3]
seve   0   0   4   1   1   1   0   4   1   0   18 [33.3/2.7]
eigh   2   3   2   0   1   0   3   0   5   0   14 [31.2/3.7]
nine   1   1   1   4   0   0   1   0   0  14    8 [63.6/2.7]
 sil   0   0   0   0   0   0   0   0   0   0    0
Ins    0   0   5   2   2   0   7   1   1   0
===============================================================
```

Τα αποτελέσματα από το classification (training)

```
====================== HTK Results Analysis =======================
  Date: Wed Jan 17 14:16:36 2007
  Ref : labels/normalonlytrain
  Rec : results_classification_training.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=43.58 [H=693, S=897, N=1590]
WORD: %Corr=43.58, Acc=43.58 [H=693, D=1, S=896, I=0, N=1590]
------------------------ Confusion Matrix -------------------------
       z   o   t   t   f   f   s   s   e   n
       e   n   w   h   o   i   i   e   i   i
       r   e   o   r   u   v   x   v   g   n
       o           e   r   e       e   h   e
                   e               n   t      Del [ %c / %e]
zero  56   2  16  12  24   8  17  15   3   6    0 [35.2/6.5]
 one   3  69   5  16  30  15   8   3   5   5    0 [43.4/5.7]
 two  11   0  85   6  20   0  14  10   6   7    0 [53.5/4.7]
thre  11  20   7  40  30   9   9   7  13  13    0 [25.2/7.5]
four   4   8  11  12  94   9  10   2   5   4    0 [59.1/4.1]
five   4   7   3  10  19  88   7   5   6   9    1 [55.7/4.4]
 six  16   8  12   5  23   6  61  10  11   7    0 [38.4/6.2]
seve   8   8   3  10  20   3  13  78   7   9    0 [49.1/5.1]
eigh   7  11  17   3  20  10  11   9  57  14    0 [35.8/6.4]
nine   6   5   6   3  25   9   8  12  20  65    0 [40.9/5.9]
Ins    0   0   0   0   0   0   0   0   0   0
```

Τα αποτελέσματα από το classification (testing)

```
====================== HTK Results Analysis =======================
  Date: Wed Jan 17 14:16:38 2007
  Ref : labels/normalonlytest
  Rec : results_classification.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=26.33 [H=79, S=221, N=300]
WORD: %Corr=26.33, Acc=26.33 [H=79, D=0, S=221, I=0, N=300]
------------------------ Confusion Matrix -------------------------
       z   o   t   t   f   f   s   s   e   n
       e   n   w   h   o   i   i   e   i   i
```

	r o	e	o	r e e	u r	v e	x	v e n	g h t	n e	Del	[%c / %e]
zero	5	2	5	5	0	4	7	0	2	0	0	[16.7/8.3]
one	1	8	0	2	3	4	1	2	9	0	0	[26.7/7.3]
two	0	0	8	2	1	2	9	1	4	3	0	[26.7/7.3]
thre	2	3	2	6	3	3	2	3	4	2	0	[20.0/8.0]
four	0	3	5	1	6	2	2	2	2	7	0	[20.0/8.0]
five	3	1	1	2	0	11	0	4	3	5	0	[36.7/6.3]
six	2	2	1	6	2	2	10	1	3	1	0	[33.3/6.7]
seve	3	0	0	3	1	1	7	7	2	6	0	[23.3/7.7]
eigh	0	3	0	1	0	8	2	1	7	8	0	[23.3/7.7]
nine	5	0	0	2	2	2	3	2	3	11	0	[36.7/6.3]
Ins	0	0	0	0	0	0	0	0	0	0		

Το σύμβολο H είναι ο αριθμός των σωστών labels, το D είναι ο αριθμός των διαγραφών, το S είναι ο αριθμός των αντικαταστάσεων, το I είναι ο αριθμός των εισαγωγών και το N είναι ο συνολικός αριθμός των labels στα ορισμένα αντιγραφικά αρχεία. Το ποσοστό των labels που αναγνωρίζονται σωστά είναι

$$\%Correct = \frac{H}{N} \times 100\%$$

και η ευστοχία υπολογίζεται με το τύπο

$$Accuracy = \frac{H - I}{N} \times 100\% .$$

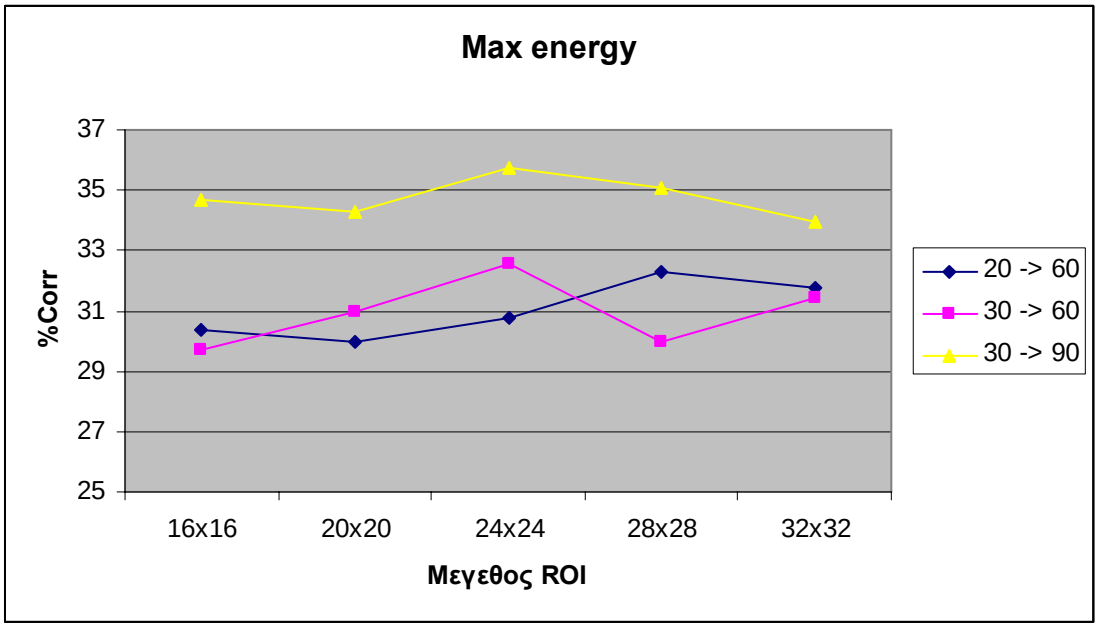
Με αυτή τη μορφή των αποτελεσμάτων λοιπόν εξαχθήκαν όλα τα αποτελέσματα όλων των πειραμάτων για το visual front end που παρουσιάζονται συνοπτικά στις επόμενες ενότητες.

4.2 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση τη μεγαλύτερη ενέργεια

Στο training (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

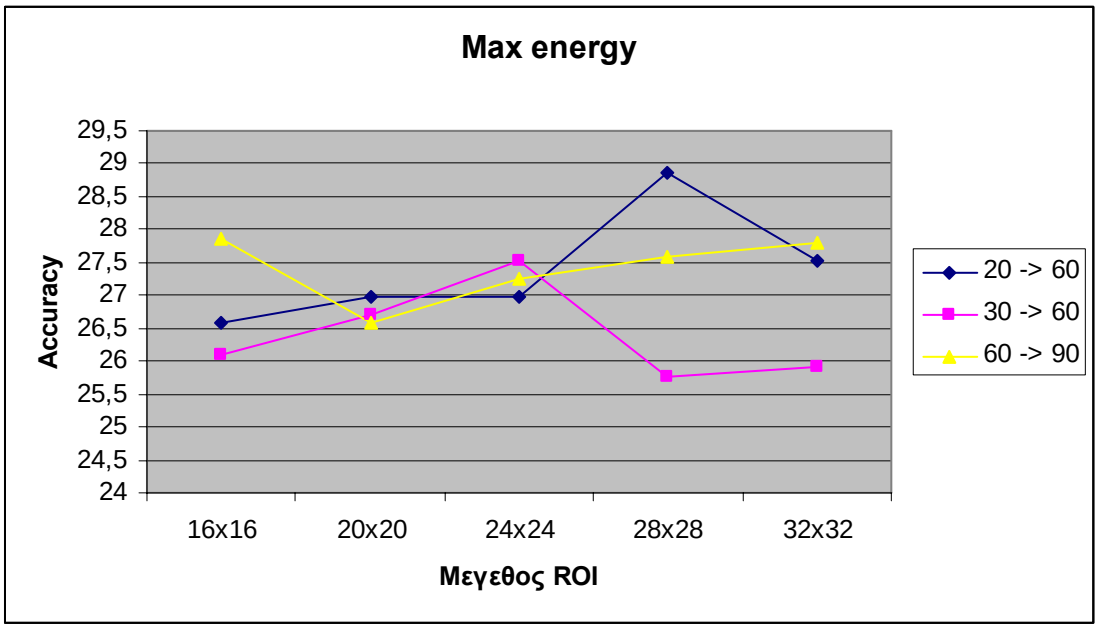
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	30,34%	30%	30,74%	32,28%	31,74%
30 → 60	29,73%	30,94%	32,55%	30%	31,41%
30 → 90	34,7%	34,3%	35,77%	35,1%	33,96%



Accuracy

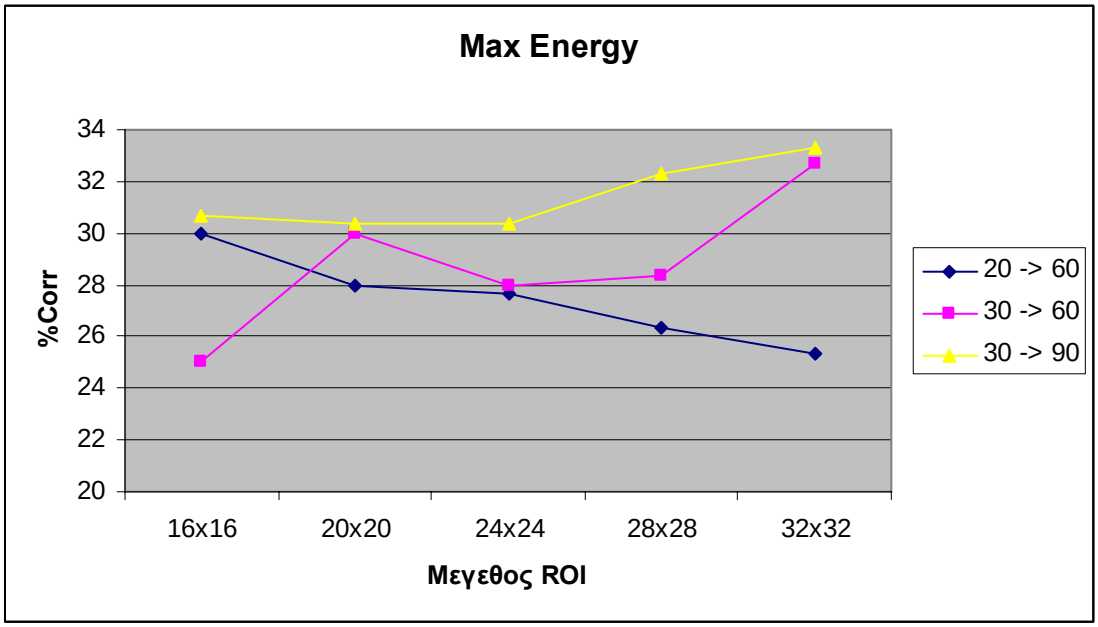
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	26,58%	26,98%	26,98%	28,86%	27,52%
30 → 60	26,11%	26,71%	27,52%	25,77%	25,91%
30 → 90	27,85%	26,58%	27,25%	27,58%	27,79%



Στο testing (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

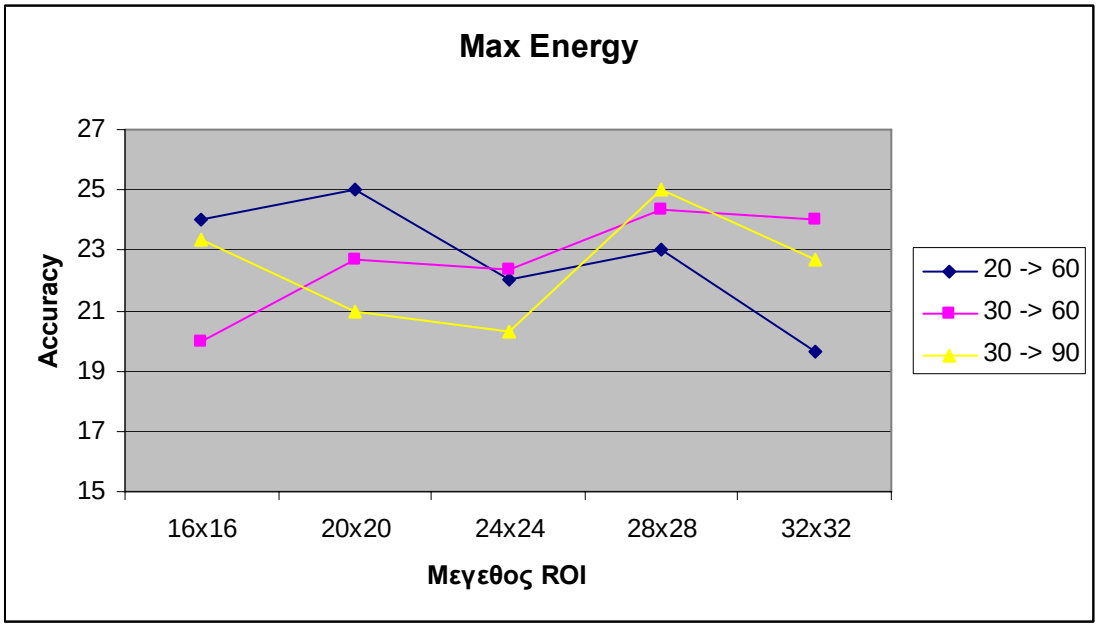
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	30%	28%	27,67%	26,33%	25,33%
30 → 60	25%	30%	28%	28,33%	32,67%
30 → 90	30,67%	30,33%	30,33%	32,33%	33,33%



Accuracy

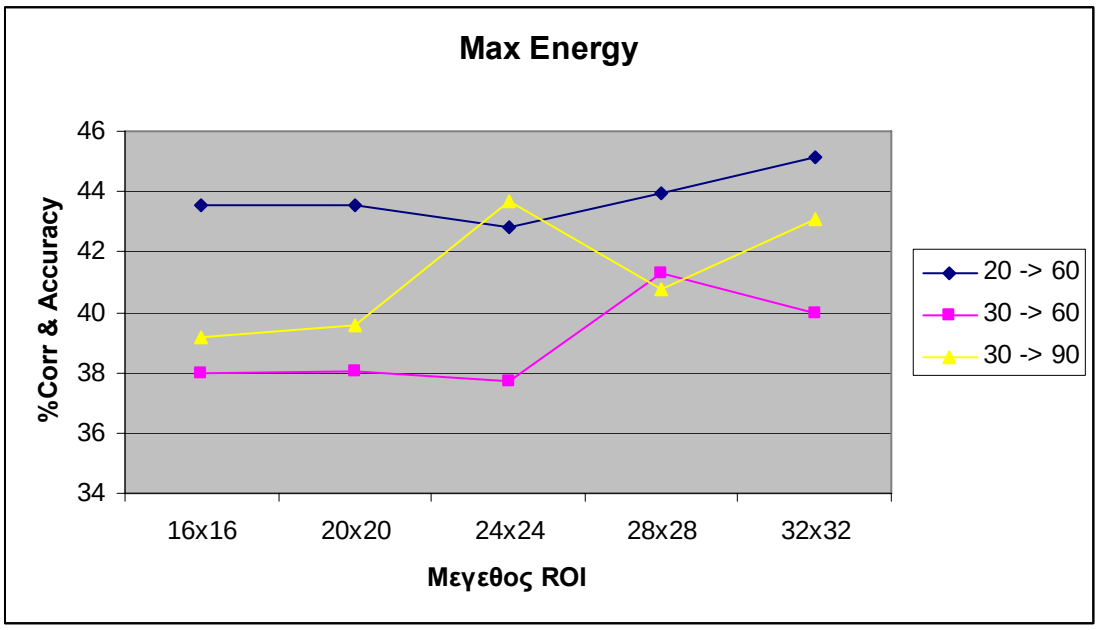
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	24%	25%	22%	23%	19,67%
30 → 60	20%	22,67%	22,33%	24,33%	24%
30 → 90	23,33%	21%	20,33%	25%	22,67%



Τώρα με classification τα αποτελέσματα είναι τα εξής:

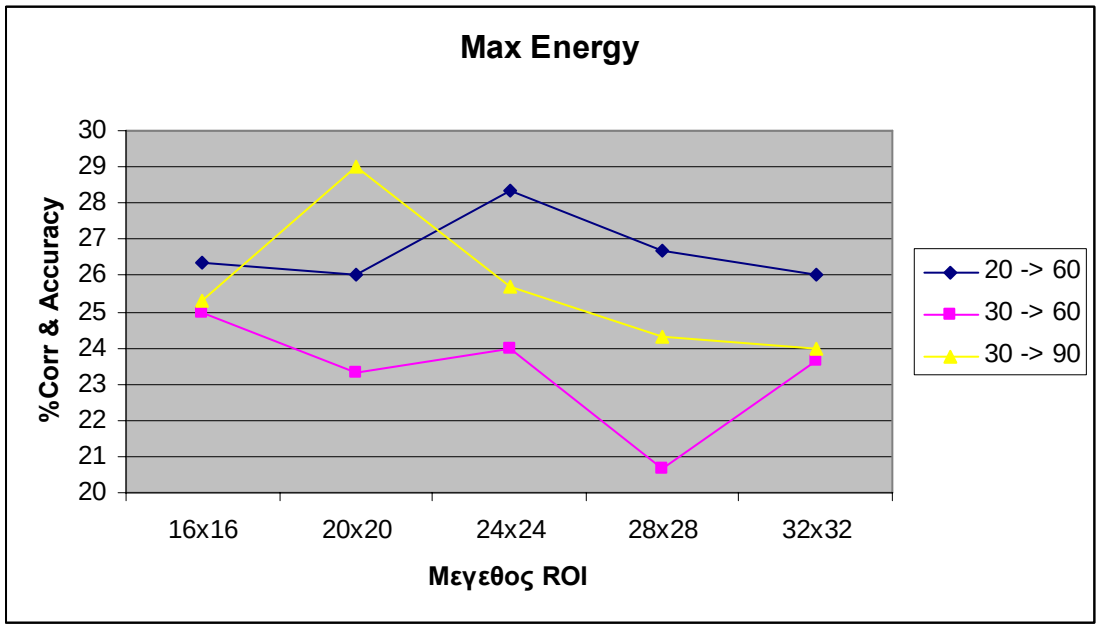
Training (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	43,58%	43,58%	42,83%	43,96%	45,16%
30 → 60	37,99%	38,05%	37,74%	41,32%	40%
30 → 90	39,18%	39,56%	43,65%	40,75%	43,08%



Testing (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	26,33%	26%	28,33%	26,67%	26%
30 → 60	25%	23,33%	24%	20,67%	23,67%
30 → 90	25,33%	29%	25,67%	24,33%	24%



Ανάλυση αποτελεσμάτων

Μπορούμε να παρατηρήσουμε πως η ευστοχία του αλγόριθμου κυμαίνεται γύρω στο 19-25% (κυρίως 22-24%) χωρίς classification και στα αποτελέσματα με classification, η ευστοχία βρίσκεται στο 21-28% (κυρίως 25-26%). Τα μικρότερα ROI έχουν την τάση να βγάζουν πιο εύστοχα τα αποτελέσματα από τα μεγαλύτερα ROI στα πειράματα.

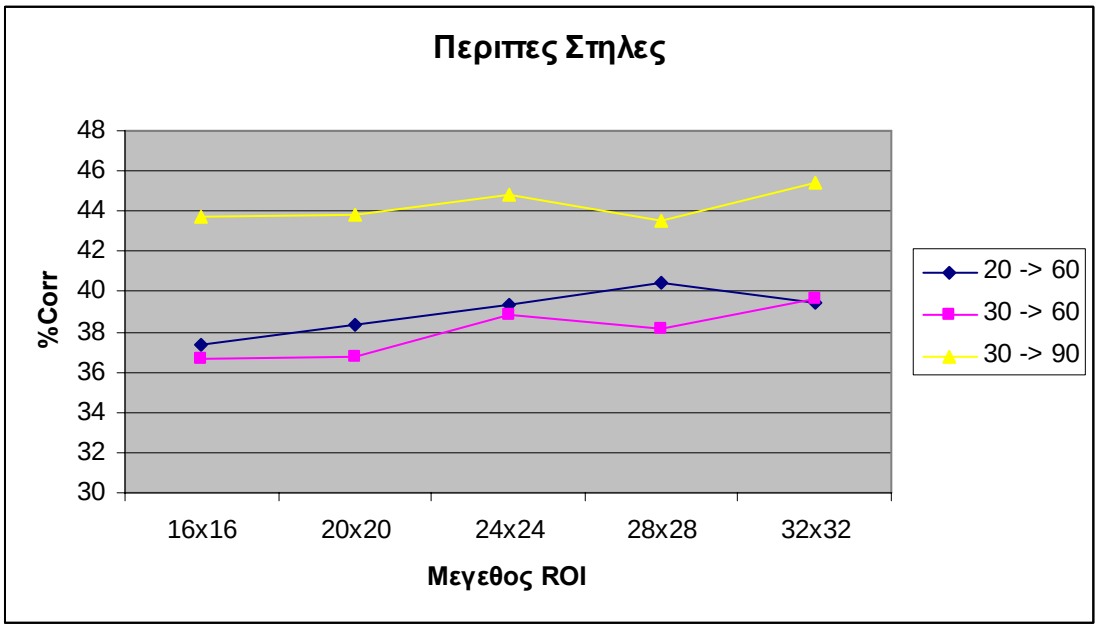
Ο πρώτος τρόπος εξαγωγής (20 → 60) δείχνει να είναι πιο αποτελεσματικός από τους υπόλοιπους τρόπους στο classification καθώς βγάζει 4-5% κατά μέσο όρο περισσότερο από τους υπόλοιπους αλγόριθμους.

4.3 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση την ενέργεια των περιττών στηλών

Στο training (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

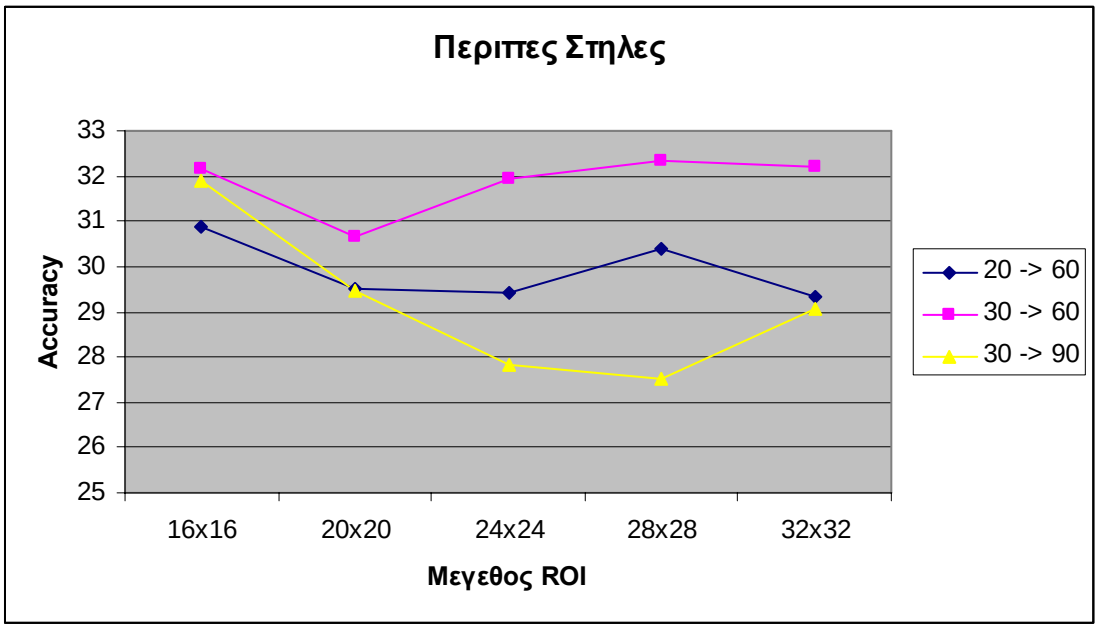
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	37,32%	38,32%	39,33%	40,47%	39,4%
30 → 60	36,64%	36,78%	38,86%	38,19%	39,66%
30 → 90	43,76%	43,83%	44,77%	43,56%	45,44%



Accuracy

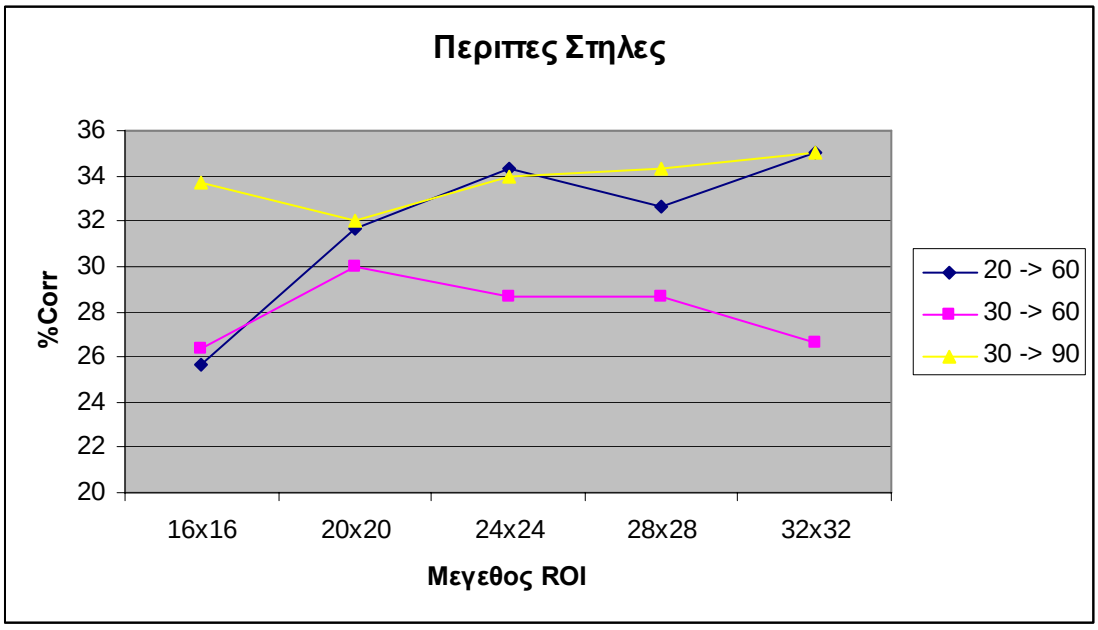
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	30,87%	29,53%	29,4%	30,4%	29,33%
30 → 60	32,15%	30,67%	31,95%	32,35%	32,21%
30 → 90	31,88%	29,46%	27,85%	27,52%	29,06%



Στο testing (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

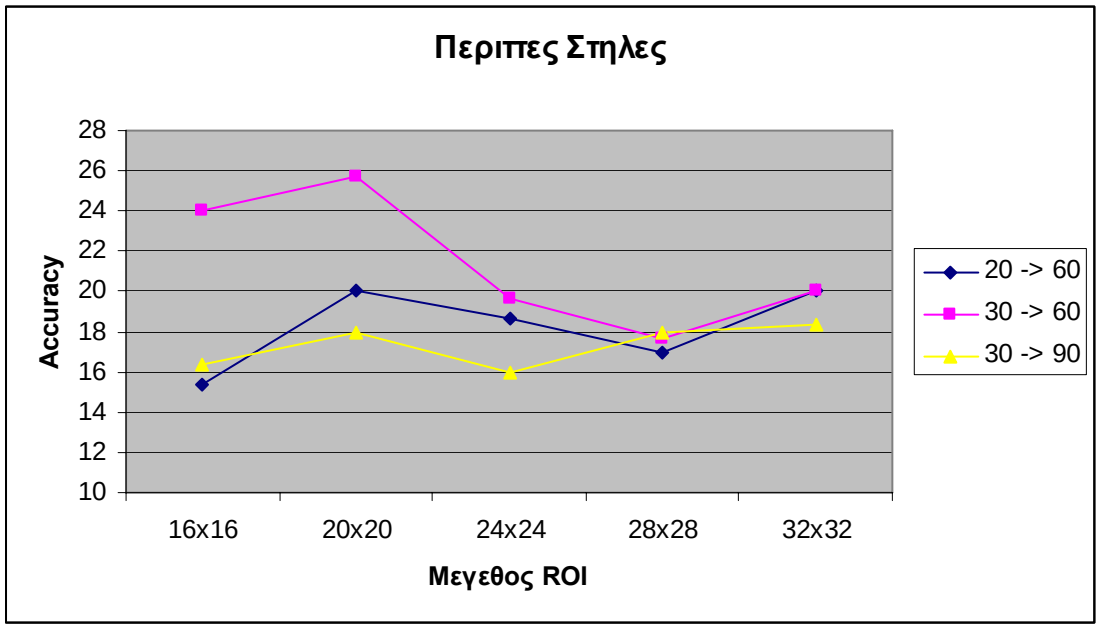
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	25,67%	31,67%	34,33%	32,67%	35%
30 → 60	26,33%	30%	28,67%	28,67%	26,67%
30 → 90	33,67%	32%	34%	34,33%	35%



Accuracy

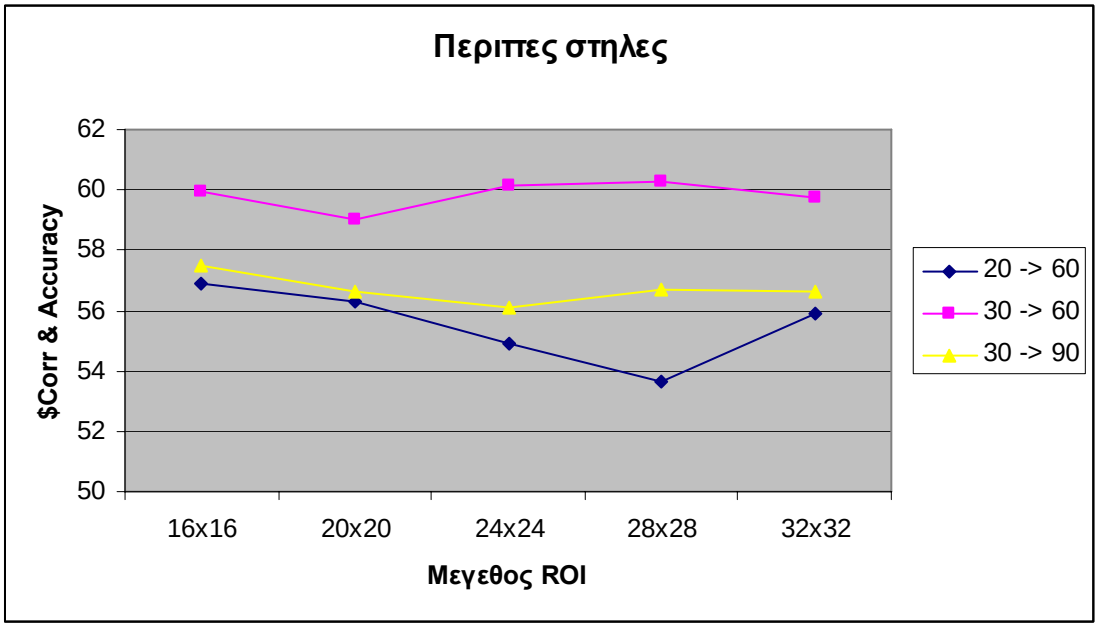
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	15,33%	20%	18,67%	17%	20%
30 → 60	24%	25,67%	19,67%	17,67%	20%
30 → 90	16,33%	18%	16%	18%	18,33%



Τώρα με classification τα αποτελέσματα είναι τα εξής:

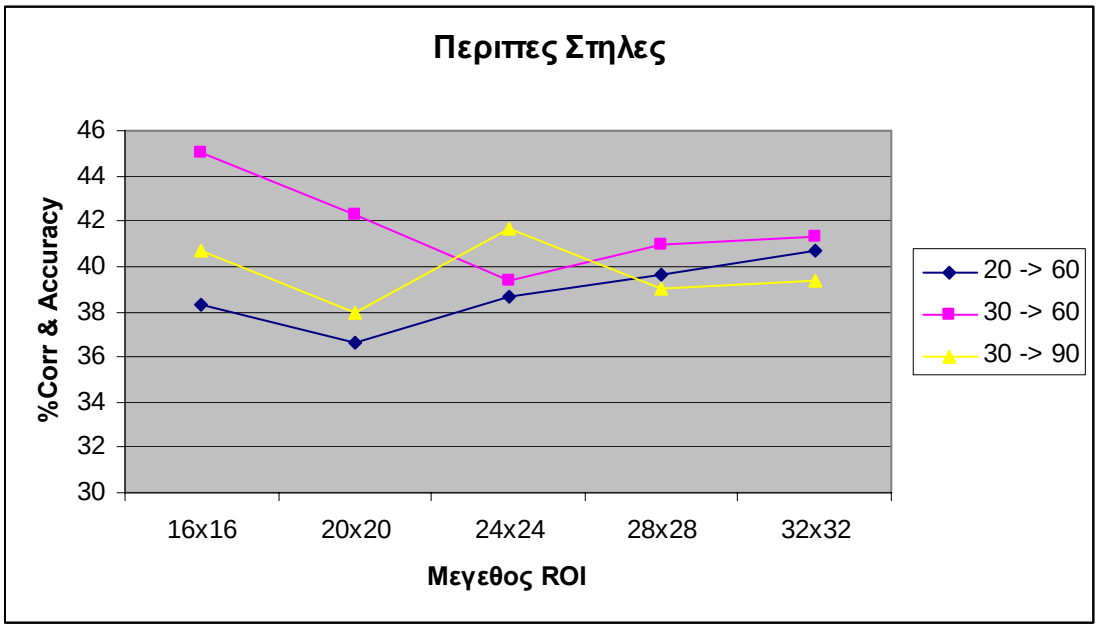
Training (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	56,92%	56,29%	54,91%	53,65%	55,91%
30 → 60	59,94%	58,99%	60,13%	60,25%	59,75%
30 → 90	57,48%	56,6%	56,1%	56,67%	56,6%



Testing (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	38,33%	36,67%	38,67%	39,67%	40,67%
30 → 60	45%	42,33%	39,33%	41%	41,33%
30 → 90	40,67%	38%	41,67%	39%	39,33%



Ανάλυση αποτελεσμάτων

Η ευστοχία του αλγόριθμου είναι στα ίδια επίπεδα με τον πρώτο αλγόριθμο, δηλαδή στο 17-24% (κυρίως 18-20%), όμως στο classification βλέπουμε μια βελτίωση από τον προηγούμενο αλγόριθμο, καθώς το ποσοστό της ευστοχίας κυμαίνεται στο 37-45% (κυρίως 39-41%).

Στο classification ο 2ος τρόπος εξαγωγής (30 → 60) φαίνεται να είναι ο καλύτερος από τους άλλους , ειδικά στα 'μικρά' ROI (16x16 και 20x20) όπου έχει 5-6% περισσότερο από τα αποτελέσματα των άλλων αλγορίθμων. Στα μεγάλα ROI (24x24, 28x28, 32x32) υπάρχει μια ελάχιστη διαφορά στους τρόπους εξαγωγής όπου πάλι ο 2^{ος} τρόπος είναι ελάχιστα καλύτερος.

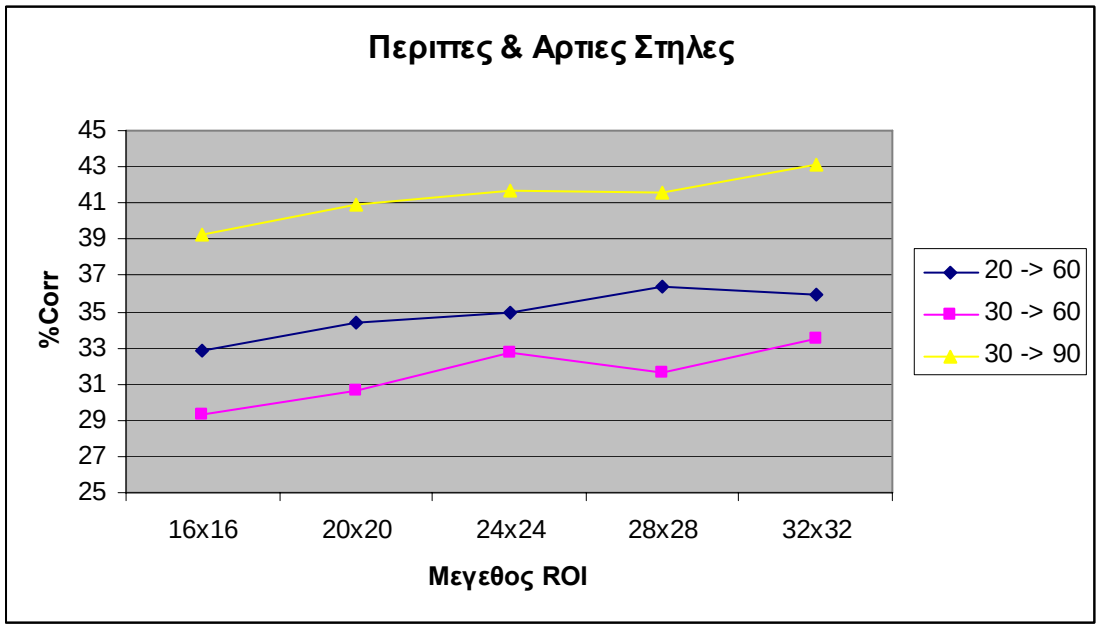
Στη γενική ανάλυση, ο συγκεκριμένος αλγόριθμος είναι καλύτερος από τον πρώτο αλγόριθμο.

4.4 Διακριτός Συνημιτονικός Μετασχηματισμός με βάση την ενέργεια των περιττών-άρτιων στηλών

Στο training (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

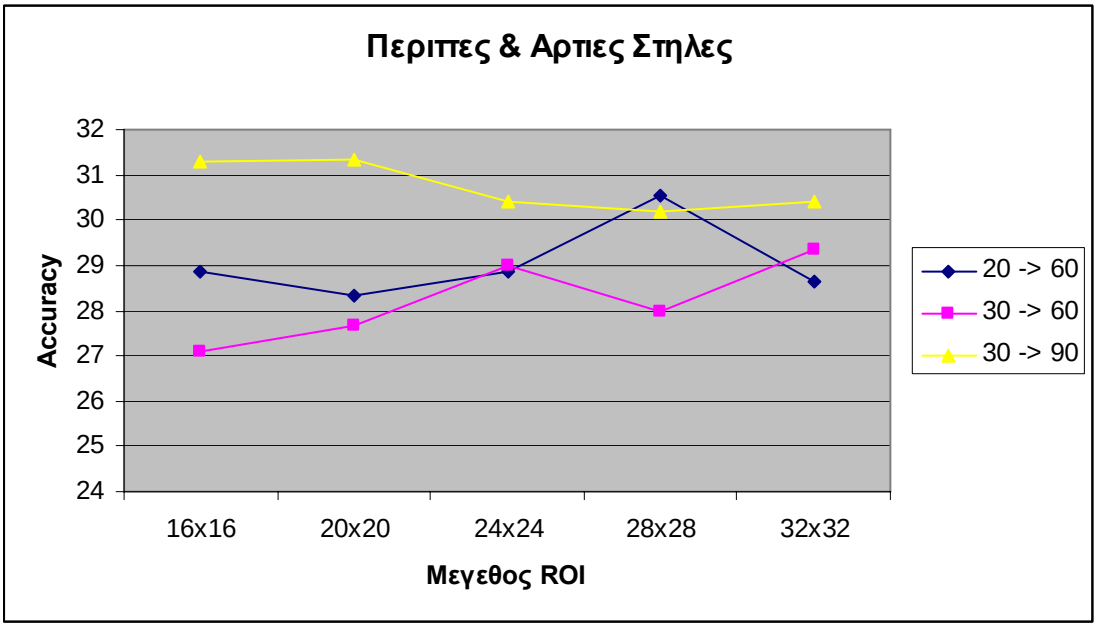
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	32,89%	34,43%	34,9%	36,38%	35,97%
30 → 60	29,26%	30,67%	32,75%	31,61%	33,56%
30 → 90	39,26%	40,94%	41,68%	41,54%	43,09%



Accuracy

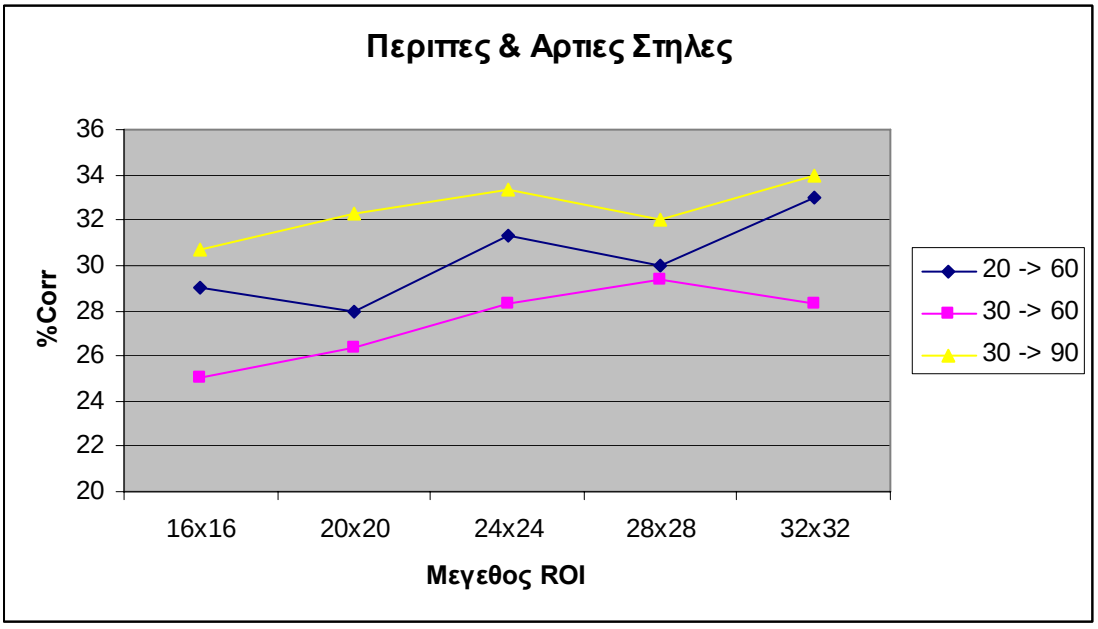
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	28,86%	28,32%	28,86%	30,54%	28,66%
30 → 60	27,11%	27,65%	28,99%	27,99%	29,33%
30 → 90	31,28%	31,34%	30,4%	30,2%	30,4%



Στο testing (χωρίς classification) είχαμε τα αποτελέσματα της ευστοχίας:

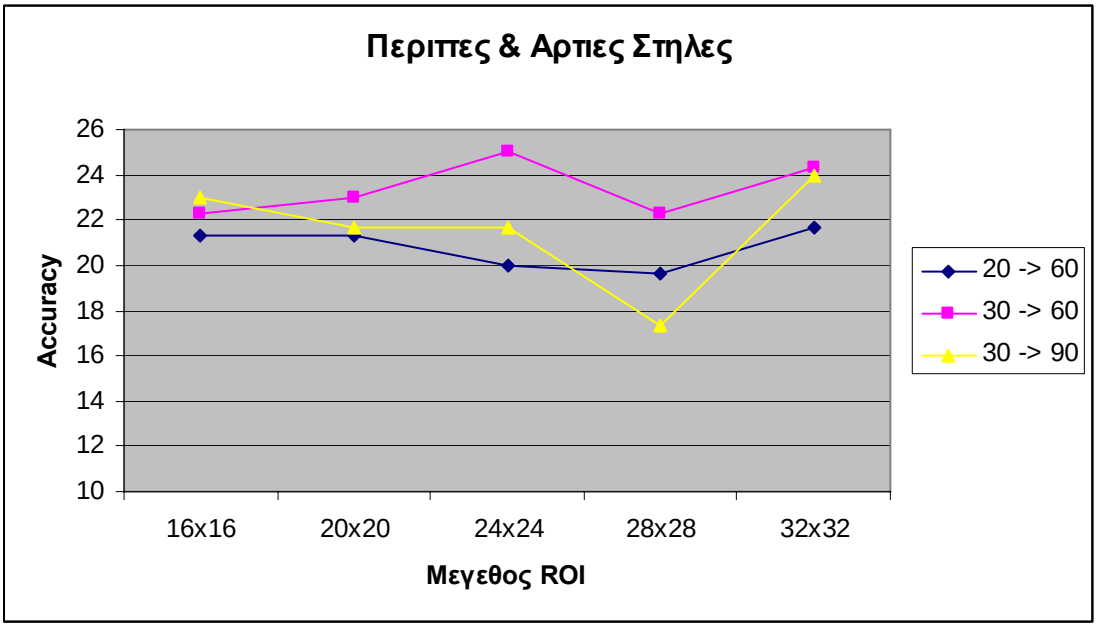
Correct%

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	29%	28%	31,33%	30%	33%
30 → 60	25%	26,33%	28,33%	29,33%	28,33%
30 → 90	30,67%	32,33%	33,33%	32%	34%



Accuracy

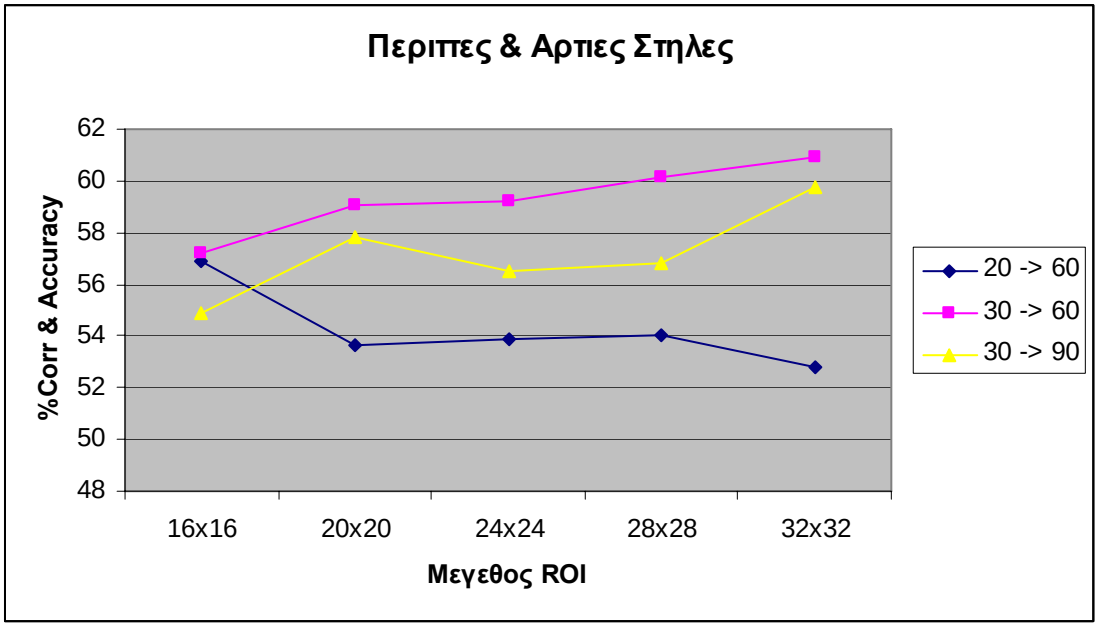
FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	21,33%	21,33%	20%	19,67%	21,67%
30 → 60	22,33%	23%	25%	22,33%	24,33%
30 → 90	23%	21,67%	21,67%	17,33%	24%



Τώρα με classification τα αποτελέσματα είναι τα εξής:

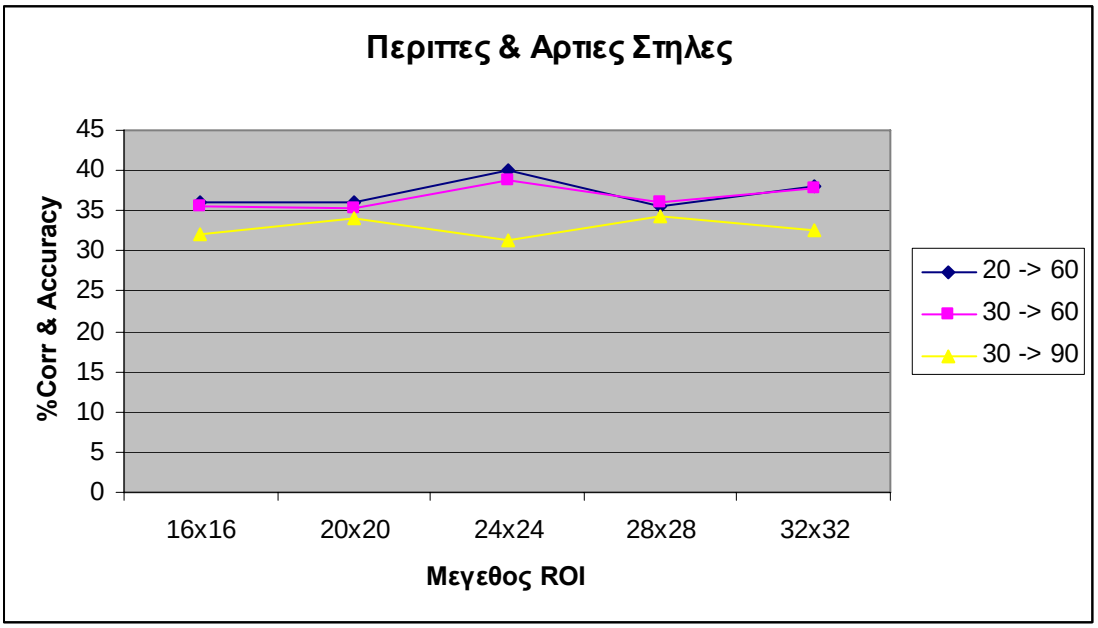
Training (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	56,92%	53,65%	53,9%	54,03%	52,77%
30 → 60	57,23%	59,06%	59,25%	60,13%	60,88%
30 → 90	54,91%	57,86%	56,48%	56,79%	59,75%



Testing (Correct% και Accuracy)

FEATURES	16x16	20x20	24x24	28x28	32x32
20 → 60	36%	36%	40%	35,67%	38%
30 → 60	35,67%	35,33%	38,67%	36%	37,67%
30 → 90	32%	34%	31,33%	34,33%	32,67%



Ανάλυση αποτελεσμάτων

Στον τελευταίο αλγόριθμο η ευστοχία έχει φτάσει 17-24% (κυρίως 23-24%) και στο classification έχει φτάσει στο 32-40% (κυρίως 36-38%). Στο classification ο πρώτος τρόπος εξαγωγής (20 → 60) είναι πιο αποτελεσματικός από τους άλλους.

Πάντως γενικά σε όλα τα μεγέθη του ROI, η ευστοχία είναι σταθερή στους τρόπους εξαγωγής.

4.5 Συνοπτικά

Με μια γενική ματιά μπορούμε να βγάλουμε το συμπέρασμα ότι ο αλγόριθμος των περιττών στηλών είναι πιο αποτελεσματικός από τους άλλους αλγορίθμους. Ειδικά στα μικρά ROI στο classification αγγίζει το 45% στην ευστοχία.

Παρόλα αυτά, γενικά η ευστοχία των αλγορίθμων είναι αρκετά χαμηλή για να θεωρηθεί αντικειμενικά ικανοποιητική η αναγνώριση των λέξεων στους ομιλητές με το συγκεκριμένο visual front end. Οπότε, πρέπει να υπάρξει βελτίωση γενικά στους αλγορίθμους, αλλά και στα άλλα μέρη του visual front end, για παράδειγμα το ROI detection.

Αναλύονται αυτά στο επόμενο κεφάλαιο.

5ο Κεφάλαιο

Future work

Το ROI detection θα μπορούσε να βελτιωθεί. Σ' αυτή την περίπτωση, θα μπορούσαν να χρησιμοποιηθούν το GMM (Gaussians mixture model) και το EM (Expectation-Maximization). Συνοπτικά, έστω ότι έχουμε k ομάδες. Γίνεται τότε η εκμάθηση ομάδων, όπου προσδιορίζονται οι παράμετροί τους, δηλαδή οι μέσες τιμές και οι τυπικές αποκλίσεις. Το κριτήριο απόδοσης είναι η πιθανότητα των δεδομένων εκπαίδευσης με γνωστές ομάδες. Χρησιμοποιείται έτσι μετά ο αλγόριθμος EM για την εύρεση τοπικού μέγιστου της πιθανότητας.

5.1 Gaussian Mixture Models

Η πυκνότητα των GMM είναι το άθροισμα των Γκαουσιανών πυκνοτήτων:

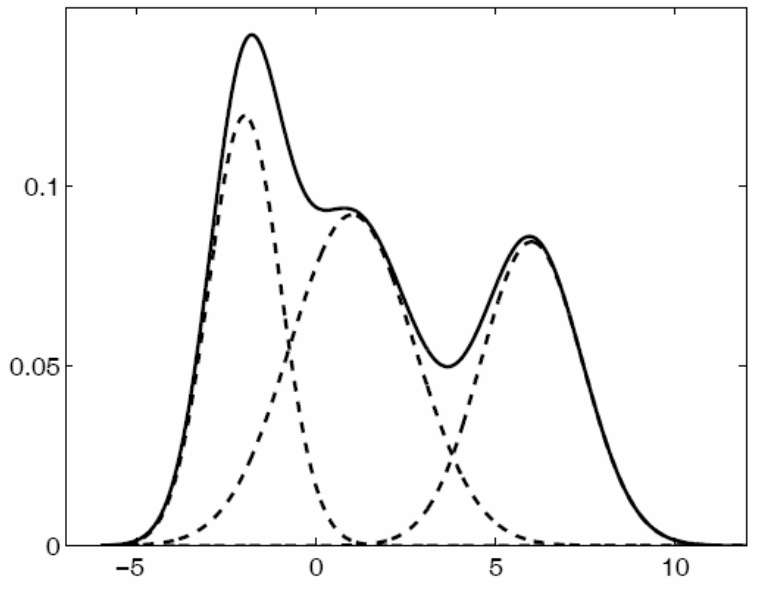
$$P_{GM}(y) = \sum_{k=1}^{K} c_k p_k(y)$$

όπου c_k είναι οι συστατικές πιθανότητες (όπως πιθανότητες που υπακούουν π.χ. $\Sigma c_k = 1$) , που λέγονται και βαρύτητες. Εξετάζουμε N-διαστατές πυκνότητες ώστε να προκύψει ένα διάνυσμα $y = (y_1, ..., y_N)$. Άρα η $p_k(y)$ είναι μια N-διαστατή Γκαουσιανή πιθανότητα πυκνότητας

$$p_k(y) = \frac{1}{(2\pi)^{N/2} (\det \Sigma_k)^{1/2}} e^{-\frac{1}{2}(y-\mu_k)^T \Sigma_k^{-1} (y-\mu_k)}$$

όπου μ_k είναι το διάνυσμα μέσης τιμής και Σ_k είναι ο πίνακας αποκλίσεων. Οι παράμετροι του GMM είναι $\{c_k\}_{k=1}^{K}, \{\mu_k\}_{k=1}^{K}, \{\Sigma_k\}_{k=1}^{K}$. Για παράδειγμα, έστω ότι μοντελοποιούμε ένα 10-διαστατό LP coefficient διανυσμάτων που έχουν εξαχθεί από μια ανάλυση LP των frames (καρέ) της ομιλίας.

Υποθέτουμε ότι έχουμε K = 32 στοιχεία του GMM μας. Σ' αυτή την περίπτωση , έχουμε συνολικά 32 (βαρύτητες) + 32 επί 10 (στοιχεία της μέσης τιμής) + 32 επί (10 επί 10/2 + 10/2) (πίνακας διασποράς, οι οποίοι είναι συμμετρικοί) = 2112 παράμετροι. Στο παρακάτω σχήμα παρουσιάζεται ένα παράδειγμα με n = 1 και K = 3.



Ένα GMM (συμπαγής γραμμή) με 3 στοιχεία πυκνότητας (διακεκομμένες γραμμές) με μέσες τιμές -2, 1 και 6 και με διασπορές 1, 3 και 2 αντίστοιχα. Οι βαρύτητες είναι 0.3, 0.4 και 0.3 αντίστοιχα. Στο σχήμα τα στοιχεία πυκνοτήτων πολλαπλασιάζονται με τις βαρύτητες.

5.2 EM (Expectation-Maximization)

Ο αλγόριθμος Εκτίμηση Μεγιστοποίησης χρησιμοποιείται για να παράγει την εκτίμηση μέγιστης πιθανοφάνειας από ελλιπή δεδομένα και γενικεύει τα k-μέσα σε πιθανοκρατικό πλαίσιο.

Είναι μια επαναληπτική διαδικασία με 2 βήματα: το βήμα Ε που λέγεται Εκτίμηση (Expectation) και το Βήμα Μ που λέγεται Μεγιστοποίηση (Maximization). Με περισσότερα λόγια, το βήμα Ε υπολογίζει την πιθανότητα ομάδας για κάθε υπόδειγμα (χρησιμοποιεί τις τρέχουσες τιμές των pixel), ενώ το βήμα Μ κάνει την Εκτίμηση των παραμέτρων κατανομών από τις πιθανότητες των ομάδων. Αποθηκεύει τις πιθανότητες των ομάδων βαρυτήτων υποδειγμάτων και σταματάει όταν η βελτίωση είναι αμελητέα (όταν κορεστεί δηλαδή το μέγιστο log-Likelihood).

Η εκτίμηση των παραμέτρων από τα υποδείγματα με βαρύτητες γίνεται με:

$$\mu_A = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}$$

$$\sigma_{\mathrm{A}}^2 = \frac{w_1 \left(x_1 - \mu\right)^2 + w_2 \left(x_2 - \mu\right)^2 + \cdots + w_n \left(x_n - \mu\right)^2}{w_1 + w_2 + \cdots + w_n}.$$

Βιβλιογραφία

● *Late Integration in audio-visual continuous speech recognition*: Ashish Verma, Tanveer Faruquie, Chalapathy Neti, Sankar Basu.

● *Improving Audio-Visual Speech Recognition with an Infrared Headset*: Jing Huang, Gerasimos Potamianos, Chalapathy Neti.

● *Joint Audio-Visual Speech Processing for Recognition and Enhancement*: Gerasimos Potamianos, Chalapathy Neti, and Sabine Deligne

● *AUDIO-VISUAL SPEECH RECOGNITION*: Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin, Iain Matthews (Carnegie Mellon University, Pittsburgh), Herve Glotin ,
Dimitra Vergyri, June Sison, Azad Mashari and Jie Zhou

● *AUDIO-VISUAL LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION IN THE BROADCAST DOMAIN*: S. Basu, C. Neti, N. Rajput_, A. Senior, L. Subramaniam, A. Verma.

● *Audio-Visual Speech Recognition in Challenging Environments*: Gerasimos Potamianos, Chalapathy Neti.

● *WEIGHTING SCHEMES FOR AUDIO-VISUAL FUSION IN SPEECH RECOGNITION*: Herv´e Glotin, Dimitra Vergyri, Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin.

● *Asynchrony modeling for audiovisual speech recognition*: Guillaume Gravier, Gerasimos Potamianos, Chalapathy Neti.

● *Large-Vocabulary Audio-Visual Speech Recognition by Machines and Humans*:
Gerasimos Potamianos, Chalapathy Neti, Giridharan Iyengar, Eric Helmuth.

● *Recent Advances in the Automatic Recognition of Audio-Visual Speech*: Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W.

● *Audio-Visual Automatic Speech Recognition: An Overview*: Gerasimos Potamianos, Chalapathy Neti, Juergen Luettin, Robert Bosch, Iain Matthews.

● *Clustering with Gaussian Mixtures*: Andrew W. Moore

● *The HTK Book*: Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland.

● *A study of Gaussian mixture models of colour and texture features for image classification and segmentation*: Haim Permuter, Joseph Francos, Ian Jermyn.

● *An Introduction to Statistical Machine Learning (EM for GMMs)*: Samy Bengio

● *ΜΑΘΗΣΗ ΜΗΧΑΝΩΝ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ:* Γ. Καραγιάννης, Γ. Σταινχάουερ.

APPENDIX

HTK TOOLKIT

Μετά από την εξαγωγή εικονικών χαρακτηριστικών, η επεξεργασία τους και η διάγνωσή τους γίνεται με τη βοήθεια του **HTK toolkit,** καθώς και τα scripts της προγραμματιστικής γλώσσας Perl**.**

Το HTK toolkit είναι ένα εργαλείο για να δημιουργούνται κρυφά μοντέλα Markov (HMMs).
Γενικά το βοήθημα είναι σχεδιασμένο για την αναγνώριση ομιλίας.
Παρακάτω αναλύεται η διαδικασία στο πείραμα.

Για να φτιαχτεί ένα HMM, απαραίτητα πρέπει πρώτα να δημιουργήσουμε ένα πρωτότυπο definition. Τα HMM μπορούν να αποθηκευτούν σε ένα κείμενο και ο πιο απλός τρόπος για να δημιουργηθεί ένα πρωτότυπο definition είναι να χρησιμοποιηθεί ένα text editor και φτιάχνουμε εικόνα σαν την παρακάτω:

```
~h "hmm1"
<BeginHMM>
    <VecSize> 4 <MFCC>
    <NumStates> 5
    <State> 2
       <Mean> 4
          0.2 0.1 0.1 0.9
       <Variance> 4
          1.0 1.0 1.0 1.0
    <State> 3
       <Mean> 4
          0.4 0.9 0.2 0.1
       <Variance> 4
          1.0 2.0 2.0 0.5
    <State> 4
       <Mean> 4
          1.2 3.1 0.5 0.9
       <Variance> 4
          5.0 5.0 5.0 5.0
    <TransP> 5
       0.0 0.5 0.5 0.0 0.0
       0.0 0.4 0.4 0.2 0.0
       0.0 0.0 0.6 0.4 0.0
       0.0 0.0 0.0 0.7 0.3
       0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Ο σκοπός του πρωτοτύπου definition είναι να περιγράφει τη φόρμα και την τοπολογία του HMM. Οι συγκεκριμένοι αριθμοί στο definition δεν είναι σημαντικοί στην αρχή. Συνεπώς το μέγεθος του πίνακα και οι παράμετροι μπορούν να προσδιοριστούν και να οριστεί ο αριθμός των καταστάσεων. Οι επιτρεπόμενες μεταβάσεις ανάμεσα στις καταστάσεις θα μπορούσαν να δηλωθούν με τις μη-μηδενικές τιμές στα αντίστοιχα στοιχεία του πίνακα μεταβάσεων και τα μηδενικά αλλού. Οι γραμμές του πίνακα μεταβάσεων πρέπει να έχουν άθροισμα ίσον με ένα εκτός από την τελευταία γραμμή στην οποία όλα τα στοιχεία πρέπει να είναι μηδενικά. Όλες οι μέσες τιμές είναι μηδενικές, αλλά οι διαγώνιες διασπορές είναι θετικές. Όλες οι καταστάσεις του definition είναι πανομοιότυπες.

A.1 HCopy

Η συνάρτηση HCopy θα αντιγράψει τα αρχεία δεδομένων σε ένα καθορισμένο αρχείο εξόδου, μετατρέποντας τα δεδομένα σε μια παραμετροποιημένη φόρμα, σε μορφή MFFC (Mel Frequency Cepstral Coefficients). Τα αρχεία μπορούν να έχουν οποιαδήποτε επέκταση, αλλά η επέκταση του αρχείου εξόδου είναι πάντα HTK. Έτσι λοιπόν, με τη βοήθεια της εντολής HCopy του HTK toolkit, αντιγράφουμε τα δεδομένα (training και testing) και στο νέο αρχείο υπολογίζουμε τις πρώτες και δεύτερες παραγώγους των coefficients. Έτσι στα νέα αρχεία, θα υπάρχουν τα coefficients, οι $1^{ες}$ και οι $2^{ες}$ παράγωγοί τους.

Οι παράγωγοι αυτές βελτιώνουν την ποιότητα της πληροφορίας των cepstral coefficients. Οι πρώτες παράγωγοι των coefficients, που λέγονται και Δέλτα coefficients, υπολογίζονται με το τύπο:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta \left(c_{t+\theta} - c_{t-\theta} \right)}{2 \sum_{\theta=1}^{\Theta} \theta^2}$$

όπου d_t είναι ένα Δέλτα coefficient στη χρονική στιγμή t υπολογίζοντας τα αντίστοιχα στατικά coefficients $c_{t+\theta}$ και $c_{t-\theta}$. Η τιμή του Θ χρησιμοποιείται για την παράμετρο DELTAWINDOW της εντολής. Εφαρμόζοντας τον τύπο στους Δέλτα coefficients βρίσκουμε τις δεύτερες παραγώγους. Έτσι μετατρέπουμε τα MFCC σε MFCC_D (μόνο Δέλτα coefficients) ή σε MFCC_D_A (Δέλτα και δεύτερες παράγωγοι των coefficients).

A.2 HCreate

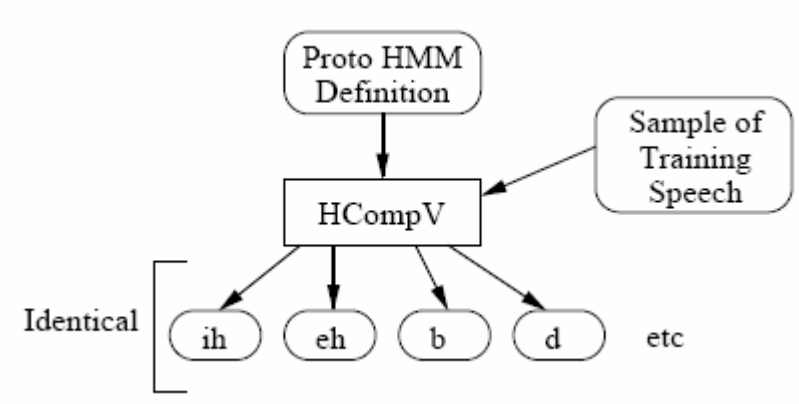
Είναι ένα script της perl που δημιουργεί μοντέλα HMM για τα coefficients. Συγκεκριμένα, δημιουργεί 4 σετ των μοντέλων για τα ψηφία 0-9 και το silence:

1. Τα αρχικά μοντέλα HMM
2. Τα μοντέλα που θα χρησιμοποιηθούν από τη συνάρτηση HCompV
3. Τα μοντέλα που θα χρησιμοποιηθούν από τη συνάρτηση Hinit
4. Τα μοντέλα που θα χρησιμοποιηθούν από τη συνάρτηση HRest

Η μέση τιμή και η διασπορά όλων των μοντέλων ορίζονται αρχικά μηδέν.

A.3 HCompV

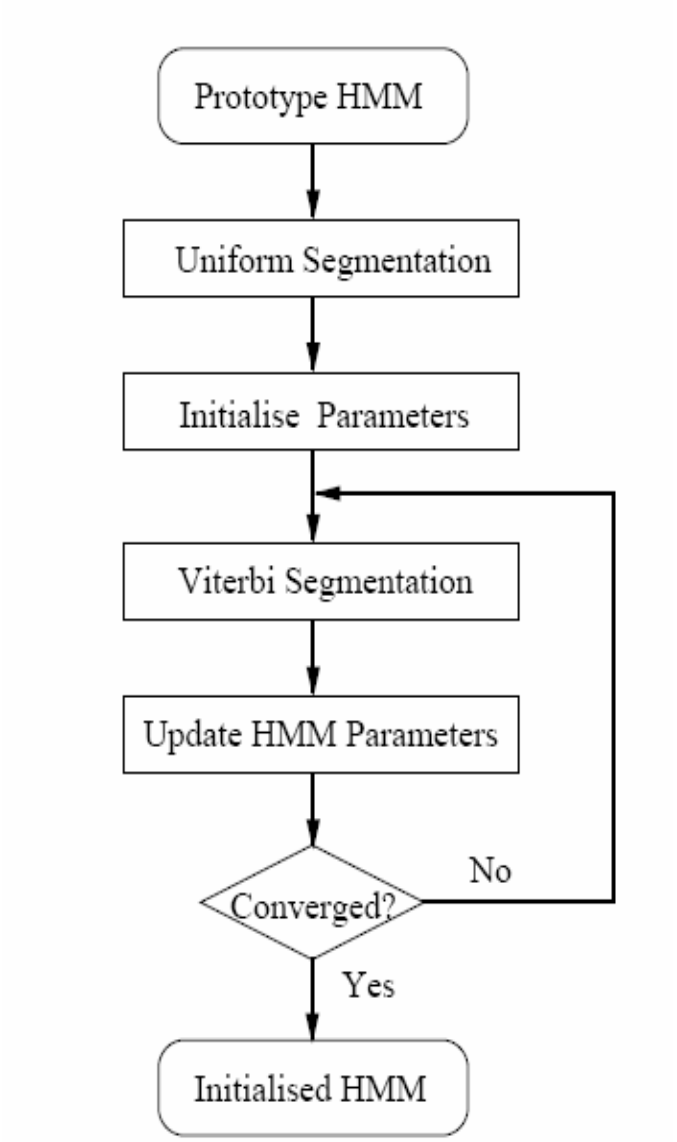
Η συνάρτηση HCompV συντελεί στην αρχικοποίηση των μοντέλων. Συγκεκριμένα, σκανάρει τα αρχεία δεδομένων εκπαίδευσης, παίρνει τα αρχικά μοντέλα HMM και υπολογίζει τη μέση τιμή και τη διασπορά των μοντέλων HMM και ορίζει όλα τα μοντέλα των ψηφίων να έχουν την ιδία μέση τιμή και διασπορά.



A.4 HInit

Η συνάρτηση Hinit χρησιμοποιείται για να δίνει τις αρχικές εκτιμήσεις για τις παραμέτρους ενός απλού HMM χρησιμοποιώντας ένα σετ των συνεχών παρατηρήσεων. Χρησιμοποιεί τον αλγόριθμο Viterbi να τμηματοποιήσει τις εκπαιδευτικές παρατηρήσεις.

Η συνάρτηση HInit παίρνει σαν είσοδο το πρωτότυπο HMM definition που είχε δημιουργηθεί προηγουμένως με το HCreate και ορίζει τη ζητούμενη HMM τοπολογία (έχει δηλαδή τη φόρμα του ζητούμενου HMM εκτός από τις μέσες τιμές και διασπορές). Ο πίνακας μεταβάσεων του πρωτοτύπου καθορίζει ταυτόχρονα τις επιτρεπόμενες μεταβάσεις και τις αρχικές πιθανότητές τους. Οι μεταβάσεις, οι οποίες έχουν μηδενικές πιθανότητες, θα παραμείνουν μηδενικές και συνεπώς ορίζονται σαν μη-επιτρεπόμενες μεταβάσεις. Το HInit υπολογίζει τις πιθανότητες των μεταβάσεων μετρώντας τον αριθμό επισκέψεων σε κάθε κατάσταση στη διαδικασία. Η έξοδος του HInit είναι ουσιαστικά η είσοδος του HRest.



Α.4.1 Ο ΑΛΓΟΡΙΘΜΟΣ VITERBI

Ο αλγόριθμος Viterbi είναι ένας reverse dynamic programming αλγόριθμος και η χρήση του ενδείκνυται για την εύρεση της πιο πιθανής σειράς κωδικοποιημένων συμβόλων (maximum likelihood), που έχουν σταλεί σε ένα ψηφιακό κανάλι επικοινωνίας. Το maximum-likelihood sequence είναι όρος γνωστός και ως Viterbi path.

Ο αλγόριθμος ανιχνεύει την πιο πιθανή ακολουθία που ο κωδικοποιητής ακολούθησε κατά την κωδικοποίηση του μηνύματος και την χρησιμοποιεί για να ανακατασκευάσει το αρχικό μήνυμα. Αντί του υπολογισμού ενός μηνύματος με βάση κάθε μεμονωμένο δείγμα λήψης, στην συνελικτική κωδικοποίηση και τη διαδικασία αποκωδικοποίησης Viterbi, ένα μήνυμα κωδικοποιείται ως μία ακολουθία λέξης (codeword), με κάποιο βαθμό συσχέτισης μεταξύ κάθε δείγματος μέσα στο σήμα. Οι αποκωδικοποιητές Viterbi εφαρμόζονται συνήθως χρησιμοποιώντας έναν DSP (digital signal processor) ή με κάποιο εξειδικευμένο hardware.

Ο αλγόριθμος προτάθηκε από τον Andrew Viterbi το 1967 σαν ένα σύστημα διόρθωσης λαθών (error-correction coding) στις ψηφιακές επικοινωνιακές ζεύξεις με θόρυβο. Έχει χρησιμοποιηθεί κυρίως στην αποκωδικοποίηση των συνελικτικών κωδίκων (convolutional encoders) σε συστήματα τηλεπικοινωνιών, όπως CDMA και GSM digital cellular, dial-up modems, σε δορυφορικές επικοινωνίες deep-space communications, και 802.11 wireless LANs. Η βασική αρχή αυτού του αλγορίθμου προκύπτει από την παρατήρηση ότι εάν η βέλτιστη διαδρομή μεταξύ του σημείου Α και του σημείου Γ περνά από ένα ενδιάμεσο σημείο Β, τότε το τμήμα αυτής της διαδρομής μεταξύ του σημείου Α και του σημείου Β είναι η βέλτιστη διαδρομή μεταξύ αυτών των δύο σημείων.

Γενικά ο αλγόριθμος Viterbi υπολογίζει το βέλτιστο μονοπάτι σε ένα trellis, χωρίς να είναι γνωστό εκ των προτέρων ποιοι κόμβοι θα αποτελούν το τελικό βέλτιστο μονοπάτι. Για το λόγο αυτό υπολογίζεται σε κάθε στιγμή το optimum path για κάθε κόμβο. Έτσι σε κάθε χρονική στιγμή και για κάθε κόμβο διατηρείται ένα survivor path. Θα πρέπει να σημειωθεί ότι ο Viterbi συνεισφέρει στη μείωση του υπολογιστικού φόρτου, γεγονός που απορρέει από τη χρήση των trellises. Στο σημείο αυτό μπορούμε να δώσουμε την περιγραφή του αλγόριθμου που αποτελείται από τέσσερα βήματα:

Βήμα 1: Ο αλγόριθμος υπολογίζει το path metric για κάθε path σε όλους του κόμβους την χρονική στιγμή k, προσθέτοντας το metric του survivor path προς τους κόμβους αυτούς τη χρονική στιγμή k-1 στο branch weight που έχει ο κάθε κόμβος τη χρονική στιγμή k.

Βήμα 2: Έπειτα για κάθε path προς τους κόμβους τη χρονική στιγμή k, κρατάει το path με το καλύτερο metric και το ανάγει στο survivor path.

Βήμα 3: Στη συνέχεια αποθηκεύει το survivor path και το metric του για κάθε κόμβο στη χρονική στιγμή k.

Βήμα 4: Τέλος αυξάνει το k και επαναλαμβάνει το βήμα 1.

Σε πολλές περιπτώσεις εύρεσης του shortest path στα trellises ο αριθμός των branches που χρειάζονται μπορεί να είναι απεριόριστα μεγάλος. Στην εφαρμογή του αλγορίθμου, καταφεύγουμε σε κάποιες τεχνικές ώστε να παραχθούν τα outputs πριν από το τέλος της λαμβανόμενης ακολουθίας, δεδομένου ότι η ακολουθία που λαμβάνεται μπορεί να συνεχιστεί κατά τρόπο αόριστο. Μία συνηθισμένη προσέγγιση στο πρόβλημα αυτό είναι να διατηρούνται όλα τα paths του trellis σε κάποιο παράθυρο με περιορισμένο μήκος. Όταν το μήκος του path γίνει ίσο με το μήκος του παραθύρου, επιλέγεται το state με το μικρότερο κόστος. Το path που οδηγεί σ' αυτό το state ανιχνεύεται και το πρώτο branch στο παράθυρο επιλέγεται ως έξοδος του αλγόριθμου. Στην επόμενη χρονική στιγμή το παράθυρο ολισθαίνει κατά ένα και τα paths στο τέλος του παραθύρου επεκτείνονται και επιλέγεται ξανά το path με το μικρότερο κόστος για να παραχθεί πάλι η επόμενη έξοδος. Η ίδια διαδικασία συνεχίζεται μέχρι να παραχθεί όλη η ακολουθία των outputs. Αν το παράθυρο έχει επιλεχθεί να είναι αρκετά μεγάλο, αυτή η προσέγγιση σχεδόν πάντα παράγει το optimum path. Ωστόσο, εφόσον το optimum path καθορίζεται από την αρχή μέχρι το τέλος της ακολουθίας, είναι πιθανόν η μέθοδος του Viterbi με τη χρήση παραθύρων να οδηγήσει σε λάθος αποτέλεσμα. Αυτό όμως συμβαίνει περιστασιακά.

A.5 HRest

Η συνάρτηση HRest χρησιμοποιείται για τον επαναληπτικό υπολογισμό των παραμέτρων ενός σετ των HMM χρησιμοποιώντας τον αλγόριθμο Baum – Welch. Για την είσοδο δέχεται το αρχείο που προκύπτει μετά από το HInit.

A.5.1 Ο Αλγόριθμος Baum –Welch για Γκαουσιανά Μείγματα

Όπως είπαμε, η πιο ευρέως χρησιμοποιούμενη συνεχής συνάρτηση πυκνότητας πιθανότητας, είναι το μείγμα Γκαουσιανών κατανομών της μορφής

$$\mathrm{b}_{\mathrm{j}}(o_t) = \sum_{k=1}^{M} w_{jk} b_{jk}(o_t), j = 1,...,N$$

όπου o_t είναι το διάνυσμα (ακολουθία παρατήρησης) και w_{im} είναι ο συντελεστής βάρους του m μείγματος στην i κατάσταση. Οι συντελεστές βαρών των μειγμάτων υπόκεινται στους ακόλουθους περιορισμούς:

$$\sum_{k=1}^{M} w_{jk} = 1, j = 1,...,N$$
$$w_{jk} \geq 0, \quad j = 1,...,N, \quad k = 1,...,M$$

Έστω $\mathrm{N}(\mathrm{O}, \mu_{im}, \Sigma_{im})$ μια πολυμεταβλητή Γκαουσιανή κατανομή με διάνυσμα μέσης τιμής μ΄ και πίνακα συμμεταβλητότητας Σ΄ :

$$\mathrm{N}(\mathrm{O}, \mu_{im}, \Sigma_{im}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma'|}} \exp\left(-\frac{1}{2}(o-\mu')\Sigma'^{-1}(o-\mu') \right)$$

Στην περίπτωση αυτή, η εκτίμηση των παραμέτρων των HMM συνεπάγεται την εκτίμηση των παραμέτρων (της μέσης τιμής και της μεταβλητότητας) κάθε
Γκαουσιανού μείγματος. Η ενδιάμεση μεταβλητή δ ορίζεται ως εξής:

$$\delta_i(t) = \sum_{m=1}^{M} \delta_{im}(t) = \sum_{m=1}^{M} \frac{1}{p} a_j(t-1) a_{ji} w_{im} b_{im}(o_t) \beta_i(t)$$

όπως προαναφέραμε Μ είναι ο συνολικός αριθμός των κατανομών που χρησιμοποιούνται για το σχηματισμό του μείγματος στην i κατάσταση και N είναι ο συνολικός αριθμός των καταστάσεων. Η διαφορά με την προηγούμενη περίπτωση είναι ότι κάθε παρατήρηση στην ακολουθία εξόδου, δεν σχετίζεται με μία μόνο συνάρτηση πυκνότητας πιθανότητας, αλλά στη δημιουργία της συνεισφέρουν όλες οι κατανομές που αποτελούν το γκαουσσιανό μείγμα, η κάθε μια με το δικό της συντελεστή βάρους.

Μια ακόμα μεταβλητή, που είναι απαραίτητη για τον πλήρη ορισμό της μεθόδου προσδιορισμού της ακολουθίας εξόδου με γκαουσιανά μείγματα, είναι η μεταβλητή **γ** , η οποία εκφράζει την πιθανότητα το μοντέλο να βρίσκεται στην κατάσταση j τη χρονική στιγμή t και η παρατήρηση o_t να δημιουργείται με συνεισφορά του k μείγματος. Η μεταβλητή **γ** ορίζεται ως :

$$\gamma_{j,k(t)} = \frac{a_j(t)\beta_j(t)}{\sum_{j=1}^{N} a_j(t)\beta_j(t)} \cdot \frac{w_{jk} N(o_t, \mu_{jk}, \Sigma_{jk})}{\sum_{\kappa=1}^{M} w_{jk} \mathrm{N}(o_t, \mu_{jk}, \Sigma_{jk})}$$

Με χρήση της ενδιάμεσης μεταβλητής δ, οι τύποι που χρησιμοποιούνται για τον επαναληπτικό υπολογισμό των παραμέτρων του μείγματος, παίρνουν την ακόλουθη μορφή:

$$\mu'_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t) o_t}{\sum_{t=1}^{T} \delta_{im}(t)}$$

$$\sigma'_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t)(o_t - \mu'_{im})(o_t - \mu'_{im})'}{\sum_{t=1}^{T} \delta_{im}(t)}$$

$$w_{im} = -\frac{\sum_{t=1}^{T} \delta_{im}(t)}{\sum_{t=1}^{T} \delta_i(t)}$$

$$\alpha'_{ij} = \frac{\sum_{t=1}^{T} a_i(t) a_{ij} b_j(o_t + 1)\beta_j(t+1)}{\sum_{t=1}^{T} a_i(t)\beta_i(t)}.$$

A.6 HVite

Η συνάρτηση HVite χρησιμοποιείται για αναγνώριση ομιλίας και βασίζεται στον αλγόριθμο Viterbi beam search. Θα πάρει τα μοντέλα HMM που έχουν δημιουργηθεί από τη συνάρτηση HRest και κάνει τη διάγνωση, δηλαδή επιλέγει το πιο πιθανό μονοπάτι στον πίνακα καταστάσεων.

A.7 HResults

Η συνάρτηση HResults είναι το εργαλείο το οποίο αναλύει την απόδοση του HTK. Διαβάζει σε ένα ζευγάρι των αρχείων label (ουσιαστικά οι έξοδοι από ένα αναγνωριστικό εργαλείο όπως το HVite) και τα συγκρίνει με αντίστοιχα βοηθητικά αντιγραφικά αρχεία.

Όταν υπολογίσει την ευστοχία της πρότασης χρησιμοποιώντας το δυναμικό προγραμματισμό στη βασική έξοδο , τα στατιστικά θα αποθηκευτούν σε ένα κείμενο της μορφής:

```
---------------------------- Overall Results -------------------
SENT:  %Correct=13.00 [H=13, S=87, N=100]
WORD:  %Corr=53.36, Acc=44.90 [H=460,D=49,S=353,I=73,N=862]
================================================================
```

Η πρώτη γραμμή δηλώνει την ευστοχία με βάση το συνολικό αριθμό των αρχείων label, τα οποία είναι πανομοιότυπα με τα αντιγραφικά αρχεία.
Η δεύτερη γραμμή είναι η ευστοχία λέξεων με βάση το δυναμικό προγραμματισμό ανάμεσα στα αρχεία labels και στα αντιγραφικά. Το σύμβολο H είναι ο αριθμός των σωστών labels, το D είναι ο αριθμός των διαγραφών, το S είναι ο αριθμός των αντικαταστάσεων, το I είναι ο αριθμός των εισαγωγών και το N είναι ο συνολικός αριθμός των labels στα ορισμένα αντιγραφικά αρχεία. Το ποσοστό των labels που αναγνωρίζονται σωστά είναι

$$\%Correct = \frac{H}{N} \times 100\%$$

και η ευστοχία υπολογίζεται με το τύπο

$$Accuracy = \frac{H - I}{N} \times 100\%$$

A.8 HVite classification

Η λειτουργία εδώ είναι ακριβώς ίδια με τη συνάρτηση HVite με τη διαφορά πως τα αρχεία MFCC τμηματοποιούνται σε 50 τμήματα για κάθε αρχείο, όπου ένα τμήμα αντιστοιχεί σε μια μόνο λέξη (ψηφίο συγκεκριμένα ή silence) και εκτελείται η συνάρτηση HVite σ' αυτά τα τμήματα.

A.9 HResults classification

Η λειτουργία εδώ είναι ακριβώς ίδια με τη συνάρτηση HResults που εκτελείται στα 50 τμήματα του κάθε αρχείου MFCC.