Επεξεργασία Σημάτων Φωνής και Ήχου

Απαλλακτική Εργασία

*Ακαδημαϊκό Έτος 2020 - 2021*

Παναγιώτης Αποστολόπουλος, Π17007

Δημήτρης Ματσαγγάνης, Π17068

Παύλος Ρουμελιώτης, Π17112

Σκαρπέλος Αλέξανδρος, Π17122



Περιεχόμενα

[1. Εκφώνηση Εργασίας 2](#_Toc77022017)

[2. Εισαγωγή 3](#_Toc77022018)

[3. Υλοποίηση 5](#_Toc77022019)

[3.1 Προεπεξεργασία 5](#_Toc77022020)

[3.1.1 Εισαγωγή αρχείων & Ρυθμός δειγματοληψίας 5](#_Toc77022021)

[3.1.2 FIR Band Pass Filter 5](#_Toc77022022)

[3.2 Dataset 6](#_Toc77022023)

[3.3 Ενέργεια Σήματος 7](#_Toc77022024)

[3.4 Zero Crossing Rate 7](#_Toc77022025)

[3.5 Background vs Foreground Classifier – Κατάτμηση Σήματος 8](#_Toc77022026)

[3.6 Εξαγωγή MFCC Χαρακτηριστικών 9](#_Toc77022027)

[3.7 Αναγνώριση ψηφίων 10](#_Toc77022028)

[4. Αποτελέσματα 11](#_Toc77022029)

[5. Συμπεράσματα 11](#_Toc77022030)

[6. Απαιτήσεις συστήματος και εκτέλεση 12](#_Toc77022031)

[7. Βιβλιογραφία 14](#_Toc77022032)

# Εκφώνηση Εργασίας

Θέμα 1 (8 βαθμοί): Καλείστε να υλοποιήσετε ένα ASR σύστημα, που δέχεται είσοδο μία ηχογράφηση κάθε φορά, η οποία συνιστά πρόταση αποτελούμενη από 5-10 ψηφία της Αγγλικής γλώσσας που έχουν ειπωθεί με αρκούντως μεγάλα διαστήματα παύσης.

1) Το σύστημα προχωρά στην κατάτμηση της πρότασης χρησιμοποιώντας υποχρεωτικά έναν ταξινομητή background vs foreground της επιλογής σας.

2) Στη συνέχεια αναγνωρίζει κάθε λέξη χρησιμοποιώντας ως φασματική αναπαράσταση μόνο το mel-spectrogram. Αν χρειαστείτε δεδομένα εκπαίδευσης, χρησιμοποιήστε μόνο σύνολο(α) δεδομένων από το site OpenSLR.

3) Στην έξοδο παράγεται κείμενo με τα ψηφία που αναγνωρίστηκαν. • Δώστε έμφαση στην επεξεργασία του σήματος, προτού αρχίσουν τα στάδια κατάτμησης/αναγνώρισης (π.χ., με κατάλληλα φίλτρα, αλλαγή ρυθμού δειγματοληψίας, κ.λ.π). • Είναι σημαντικό να περιγράψετε το σύστημα αλγοριθμικά (εξαγωγή χαρακτηριστικών, αλγόριθμος αναγνώρισης) και να εξηγήσετε τις επιδόσεις του χρησιμοποιώντας τις κατάλληλες μετρικές. • Πρέπει να εξηγήσετε ποια δεδομένα χρησιμοποιήσατε κατά τον έλεγχο και την εκπαίδευση του συστήματος. Αν είναι δικά σας, πώς τα δημιουργήσατε. • Προσπαθήστε να μην εξαρτάται το σύστημα από τα χαρακτηριστικά της φωνής του ομιλητή, αλλά να είναι όσο το δυνατόν ανεξάρτητο ομιλητή.

**Θέμα 2 (2 βαθμοί):** Άσκηση επισημείωσης δεδομένων ήχου. Θα δοθούν λεπτομέρειες σε ξεχωριστή ανακοίνωση.

# Εισαγωγή

Ζητούμενο της συγκεκριμένης εργασίας είναι η υλοποίηση ενός Automatic Speech Recognition συστήματος, το οποίο θα δέχεται μια ηχογράφηση μιας πρότασης αποτελούμενη από 5-10 ψηφία της Αγγλικής και το οποίο με τις κατάλληλες αλγοριθμικές διαδικασίες θα παράγει ένα κείμενο με τα ψηφία που λέχθηκαν.

Αρχικά, για το αρχείο εισόδου θα πρέπει να δοθεί ιδιαίτερη έμφαση στην επεξεργασία του σήματος, πριν την διαδικασία κατάτμησης και αναγνώρισης, χρησιμοποιώντας κατάλληλα φίλτρα αλλά και αλλάζοντας τον ρυθμό δειγματοληψίας. Mε την χρήση ενός FIR Pass φίλτρου επιτυγχάνεται η ποιοτική βελτίωση του σήματος, ενώ με την αλλαγή του ρυθμού δειγματοληψίας μπορούμε να μειώσουμε τον αριθμό των δειγμάτων ανά δευτερόλεπτο, καθιστώντας τους υπολογισμούς πιο γρήγορους. Στη συνέχεια, με την βοήθεια ενός ταξινομητή **Background vs Foreground**, ταξινομούνται στο background οτιδήποτε δεν είναι σήμα ομιλίας και στο foreground το σήμα ομιλίας με τη βοήθεια του short term processing, και την εύρεση του Root Mean Square Energy (RMSE) και του Zero Crossing Rate για κάθε παράθυρο. Πιο αναλυτικά, θα σπάει το σήμα σε κομμάτια (windows), τα οποία θα είναι επικαλυπτόμενα κατά 50%, δηλαδή με HOP\_SIZE = ½(FRAME\_SIZE) και για κάθε κομμάτι σήματος θα παίρνει μια απόφαση κατά πόσο είναι σήμα ομιλίας ή σήμα υποβάθρου (background), αξιοποιώντας τα παραπάνω χαρακτηριστικά.

Στη συνέχεια, κάθε λέξη θα αναγνωρίζεται χρησιμοποιώντας τη φασματική αναπαράσταση μόνο του mel-spectogram. Αυτό θα αξιοποιηθεί καταλλήλως, προκειμένου να εξαχθούν τα MFCC features, τα οποία ταξινομούνται στο SVM model. Για τη δημιουργία του SVM χρησιμοποιήθηκαν δεδομένα τα οποία κατασκευάσαμε εμείς οι ίδιοι. Τέλος, χρησιμοποιείται μια μετρική που μετράει το ποσοστό επιτυχούς αναγνώρισης των ψηφίων.

Σαν υποθέσεις εργασίας κάναμε τις εξής:

1. Για τη δημιουργία του Dataset ηχογραφήσαμε όλους τους αριθμούς από δέκα φορές τον έκαστο, μεριμνώντας να περιορίσουμε όσο το δυνατόν περισσότερο το θόρυβο.
2. To FRAME\_LENGTH είναι ίσο με και το HOP\_SIZE ίσο με, δηλαδή υπάρχει επικάλυψη
3. Ο ρυθμός δειγματοληψίας καθορίστηκε στα 8000 ms.
4. Για το αρχείο εισόδου για το οποίο καλούμαστε να εφαρμόσουμε την παραπάνω διαδικασία, επιλέχθηκε η χρήση κενού μεταξύ των ψηφίων, διαστήματος 500 ms.

# Υλοποίηση

Για την υλοποίηση και την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού *Python 3.7.9*, οι απαιτήσεις για την εκτέλεση παρατίθενται στην **ενότητα 6**. Τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος θα αναλυθούν διεξοδικά παρακάτω.

## Προεπεξεργασία

Στη συγκεκριμένη ενότητα θα αναφερθούμε στις απαραίτητες ενέργειες που γίνονται από την εφαρμογή μας με σκοπό την βέλτιστη επεξεργασία του, τόσο του συνόλου εκπαίδευσης, όσο και του εισαγόμενου αρχείου προς αναγνώριση.

### Εισαγωγή αρχείων & Ρυθμός δειγματοληψίας

Αρχικά, για κάθε αρχείο που βρίσκεται στο σύνολο εκπαίδευσης (***dataset***), ορίζουμε τον ρυθμό δειγματοληψίας, *sample rate*, ίσο με 8000 ms, αφού μετά από έρευνα, αυτός ο ρυθμός επιτρέπει την καλύτερη καταγραφή και ποιοτικότερη επεξεργασία του σήματος.

Ομοίως και για το σήμα εισόδου, ορίζουμε τον ρυθμό δειγματοληψίας ίσο με 8000 ms, ενώ για τον σκοπό αυτό υλοποιήθηκε μια συνάρτηση, η ***preprocessing()***.

### FIR Band Pass Filter

To πρώτο βήμα για την επεξεργασία τόσο των αρχείων που βρίσκονται στο σύνολο εκπαίδευσης, όσο και του ηχογραφημένου σήματος, είναι η διαδικασία φιλτραρίσματος μέσω ενός FIR filter band (*Finite Impulse Response*). Ένα band pass filter, είναι μία διαδικασία μέσω της οποίας γίνονται δεκτές οι συχνότητες ενός συγκεκριμένου πεδίου ορισμού, ενώ ταυτόχρονα απορρίπτονται οι συχνότητες εκτός αυτού του πεδίου.

Στην υλοποίησή μας χρησιμοποιούμε ένα δικό μας band pass filter, το οποίο όπως προαναφέρθηκε χαρακτηρίζει ως αποδεκτά τα φίλτρα μεταξύ δύο συχνοτήτων *fL (frequency Low)* και *fH (frequency High)* και απορρίπτει τις συχνότητες εκτός του παραπάνω πεδίου. Οι πλέον κατάλληλες συχνότητες για το βέλτιστο φιλτράρισμα είναι τα 200 Hz για το *fL* και τα 4000 Hz για το *fH.*

Στη συνέχεια, υπολογίζουμε τα άνω και κάτω φράγματα με βάση τις συχνότητες *fL* και *fH,* συνελίσσουμε τα δύο αποτελέσματα και τα εφαρμόζουμε στο εισερχόμενο σήμα. Η παραπάνω διαδικασία υλοποιείται με τη βοήθεια της βιβλιοθήκης ***numpy*** και των μεθόδων που αυτή μας παρέχει.

## Dataset

Στον φάκελο *training* υπάρχουν ξεχωριστοί φάκελοι για κάθε ψηφίο από το 0 έως το 9, όπου ο καθ΄ ένας από αυτούς περιέχει 10 ηχογραφημένα παραδείγματα για τον κάθε ομιλητή. Το πρόγραμμα εκτελεί αναζήτηση για το κάθε αρχείο και το ταυτίζει με το κάθε ψηφίο. Μετά το αρχείο υπόκειται στη προεπεξεργασία, όπως αυτή περιεγράφηκε στην παραπάνω ενότητα και έπειτα γίνεται η εξαγωγή των *MFCC* χαρακτηριστικών, που θα εξηγηθεί εκτενώς στην ***Ενότητα 3.6.*** Αφού ολοκληρωθεί η εξαγωγή αυτών, με τη βοήθεια της βιβλιοθήκης ***numpy*** υπολογίζουμε την μέση τιμή αυτών χρησιμοποιώντας την σχετική μέθοδο.

Ακολούθως, τα συμπεράσματα που προέκυψαν για το κάθε αρχείο, καθώς και τα ψηφία που αντιπροσωπεύουν εισάγονται σε ένα *Random Forest Classifier*. Ο συγκεκριμένος ταξινομητής υλοποιείται με τη βοήθεια της βιβλιοθήκης ***sklearn*** και για τον σκοπό αυτό δημιουργήθηκε η συνάρτηση ***build\_dataset***, η οποία επιστρέφει τον classifier.

## Ενέργεια Σήματος

Σε αυτή την ενότητα θα αναφερθούμε στην ενέργεια του σήματος και τη χρησιμότητα της στην υλοποίησή μας.

Αρχικά, αξίζει να αναφερθούμε στον ορισμό της ενέργειας του σήματος. Πιο συγκεκριμένα, ως ενέργεια σήματος ορίζουμε άθροισμα των τετραγώνων των φωνημάτων (των κυματομορφών - magnitudes).

Ωστόσο, ως παράμετρο της εφαρμογής μας δεν χρησιμοποιούμε την ενέργεια, αλλά υπολογίζουμε τη μέση τετραγωνική ρίζα αυτής (*Root Mean Square Energy – RMSE*) για κάθε παράθυρο (frame) των ηχητικών σημάτων. Αυτό γίνεται με τη βοήθεια της βιβλιοθήκης της Python, ***librosa*** και συγκεκριμένα τη μέθοδο *librosa.feature.rms*, η οποία υπολογίζει με αρκετά γρήγορο τρόπο τη μέση τετραγωνική ρίζα της ενέργειας, καθώς δεν απαιτεί τον υπολογισμό του STFT (*Short-Time Fourier Transform*).

Συνεπώς, ορίζοντας τις παραμέτρους που απαιτεί η μέθοδος (το ηχητικό σήμα, το μήκος παραθύρου και το μέγεθος του βήματος), συνδυαστικά με την χρήση της ***numpy*** λαμβάνουμε ως αποτέλεσμα έναν πίνακα που εμπεριέχει την ενέργεια των ακουστικών σημάτων. Αυτός ο πίνακας παίζει καθοριστικό ρόλο στην εύρεση των ουσιώδη φωνημάτων (*thresholds*) μέσα στο κάθε ηχητικό αρχείο. Προκειμένου να εξάγουμε την ενέργεια του σήματος δημιουργήσαμε τη συνάρτηση ***rmse***.

## Zero Crossing Rate

Σημαντικό βήμα για την υλοποίηση του ταξινομητή Background vs Foreground είναι η εύρεση του ρυθμού διέλευσης από το μηδέν για το αρχείο εισόδου. Ο ρυθμός αυτός προκύπτει βρίσκοντας τον αριθμό των φορών που διέρχεται ένα σήμα του μηδενός από αρνητικές σε θετικές τιμές και αντιστρόφως διαιρώντας το με το μήκος του παραθύρου.

Η σημασία του εν λόγω ρυθμού έγκειται στο γεγονός ότι αξιοποιείται προκειμένου να μπορέσουμε να έχουμε μια άποψη για τη θορυβώδη ή όχι φύση του ήχου κάθε στιγμή. Όταν έχουμε ένα σήμα, στο οποίο υπάρχουν τμήματα τα οποία οφείλονται στο σήμα του περιβάλλοντος υποθέτουμε ότι σε εκείνα τα σημεία το Zero Crossing Rate (ZCR) είναι μεγαλύτερο και η ενέργεια μικρότερη. Συνεπώς, ο συνδυασμός των δύο αυτών παρατηρήσεων μας επιτρέπει να κόψουμε το χρήσιμο σήμα και να αποκόψουμε τα τμήματα του σήματος που οφείλονται στο περιβάλλον. Για την εύρεση του ZCR υλοποιήθηκε η συνάρτηση ***zero\_crossing\_rate***, στης οποίας το σώμα αξιοποιήθηκε η μέθοδος της βιβλιοθήκης ***librosa***, ***feauture.zero\_crossing\_rate***, με ορίσματα το σήμα εισόδου, το μήκος του παραθύρου και το *hop\_size*.

## Background vs Foreground Classifier – Κατάτμηση Σήματος

Στη παρούσα ενότητα θα αναφερθούμε στον ταξινομητή *Background vs Foreground*, ο οποίος αξιοποιεί τόσο την ενέργεια του σήματος, όσο και το ρυθμό διέλευσης από το μηδέν (*Zero Crossing Rate*).

Πιο αναλυτικά, με τη βοήθεια της συνάρτησης ***b\_vs\_f***, θα διακρίνουμε τα φωνήματα των ψηφίων από τον ήχο του περιβάλλοντος. Σαν κατώφλι (threshold) της ενέργειας θεωρούμε την μέση τιμή αυτής, ενώ σαν κατώφλι του *ZCR* την μέση αυτού. Για την εύρεση ενός φωνήματος ελέγχουμε εάν το *Zero Crossing Rate* του εκάστοτε παραθύρουείναι χαμηλότερο ή ίσο από το αντίστοιχο κατώφλι και ταυτόχρονα η ενέργεια αυτού υψηλότερη ή ίση του κατωφλιού.

Το επόμενο βήμα, είναι η κατάτμηση του σήματος αξιοποιώντας τα αποτελέσματα της προηγούμενης διαδικασίας. Ειδικότερα, μετά από μια σειρά ελέγχων επιστρέφεται μια λίστα με τα φωνήματα για τα ψηφία που αναγνωρίστηκαν.

## Εξαγωγή MFCC Χαρακτηριστικών

Το επόμενο στάδιο στην υλοποίησή μας είναι η εξαγωγή των MFCC (Mel-Frequency Cepstral Coefficients) χαρακτηριστικών, τα οποία όπως θα δούμε και στη συνέχεια θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου αναγνώρισης των ψηφίων.

Η τεχνική εξαγωγής χαρακτηριστικών MFCC περιλαμβάνει τη κατάτμηση του σήματος σε παράθυρα, την εφαρμογή του DFT (Discrete Fourier Transform), την αντιστοίχιση των συχνοτήτων στην κλίμακα Mel, αλλά και την εφαρμογή του αντίστροφου DCT. Η λεπτομερής περιγραφή των επιμέρους βημάτων που εμπλέκονται στην εξαγωγή χαρακτηριστικών MFCC εξηγείται παρακάτω:

1. Frame blocking and windowing:

Το σήμα ομιλίας είναι ένα αργό στο χρόνο ή σχεδόν στατικό σήμα. Για σταθερά ακουστικά χαρακτηριστικά, η ομιλία πρέπει να εξεταστεί για αρκετά σύντομο χρονικό διάστημα. Επομένως, η ανάλυση ομιλίας πρέπει να πραγματοποιείται πάντα σε μικρά τμήματα στα οποία θεωρείται το σήμα ομιλίας στάσιμο. Ο σκοπός της αλληλεπικαλυπτόμενης ανάλυσης είναι ότι κάθε ήχος ομιλίας της ακολουθίας εισόδου θα είναι περίπου στο κέντρο κάθε frame.

1. DFT spectrum:

Εφαρμόζοντας τον DFT μετατρέπεται σε φάσμα μεγέθους κάθε windowed frame.

1. Mel spectrum:

H αναπαράσταση του βραχυπρόθεσμου φάσματος ισχύος ενός ήχου, που βασίζεται σε έναν γραμμικό μετασχηματισμό ενός λογαριθμικού φάσματος ισχύος αναπαριστώμενο σε μια μη γραμμική κλίμακα mel.

1. Discrete Cosine Transform (DCT):

Το DCT εφαρμόζεται στους μετασχηματισμένους συντελεστές συχνότητας Mel παράγοντας ένα σύνολο συντελεστών cepstral. Ως εκ τούτου, οι κορυφές των χαμηλών συχνοτήτων αντιπροσωπεύονται από ένα σήμα στον cepstral τομέα με την αντίστοιχη κορυφή αιχμής στο βήμα του σήματος και από έναν αριθμό παραγόντων.

1. Dynamic MFCC features:

Οι συντελεστές cepstral αναφέρονται συνήθως ως στατικά χαρακτηριστικά, καθώς περιέχουν μόνο πληροφορίες από ένα δεδομένο πλαίσιο.

Προκειμένου να εξαχθούν τα συγκεκριμένα χαρακτηριστικά, αξιοποιήθηκε η μέθοδος της βιβλιοθήκης ***librosa***, ***feauture.mfcc***. Για το σύνολο εκπαίδευσης η μέθοδος δέχεται σαν όρισμα το εκάστοτε αρχείο, τον ρυθμό δειγματοληψίας, το *FRAME\_SIZE* και το *HOP\_SIZE*. Αντίστοιχα, για το αρχείο εισόδου τα ορίσματα είναι τα φωνήματα που έχουν προκύψει από την κατάτμηση, αλλά και ο ρυθμός δειγματοληψίας, το *FRAME\_SIZE* και το *HOP\_SIZE*. Έχοντας εξάγει τα χαρακτηριστικά αυτά, υπολογίζουμε την μέση τιμή αυτών χρησιμοποιώντας την σχετική μέθοδο της βιβλιοθήκης ***numpy***.

## Αναγνώριση ψηφίων

Τελευταίο βήμα στην εν λόγω υλοποίηση, είναι η αξιοποίηση της μέσης τιμής των *MFCC* χαρακτηριστικών από τον *Random Forest Classifier*, ώστε να γίνει η πρόβλεψη των ψηφίων με βάση το εκπαιδευμένο μοντέλο, όπως αυτό έχει προκύψει από το σύνολο εκπαίδευσης. Με τη συνάρτηση ***recognition*** ζητάμε από το μοντέλο να προβλέψει το ψηφίο που αντιστοιχεί στα *MFCC* χαρακτηριστικά του κάθε φωνήματος, όπως αυτά είχαν προκύψει από τη διαδικασία της κατάτμησης και εκτυπώνεται το αποτέλεσμα.

# Αποτελέσματα

Στη συγκεκριμένη ενότητα θα παρουσιαστούν τα αποτελέσματα από την εκτέλεση του προγράμματος.

## Μετρικές

Οι μετρικές διαδραματίζουν σημαντικό ρόλο στην παρούσα εφαρμογή, καθώς παρέχουν τα ποσοστά επιτυχίας των ταξινομητών.

### Μετρική Background vs Foreground Classifier

Η μετρική του ταξινομητή *Background vs Foreground* χρησιμοποιεί όλα τα αρχεία του φακέλου *testing* και βρίσκει το ποσοστό επιτυχίας του. Ως επιτυχία ορίζεται η εύρεση του σωστού αριθμού ψηφίων για το εκάστοτε αρχείο.

### Μετρική Random Forest Classifier

Η μετρική του ταξινομητή *Random Forest* χρησιμοποιεί όλα τα αρχεία του φακέλου *testing*, για τα οποία βρέθηκε με επιτυχία το πλήθος των ψηφίων τους. Για το κάθε αρχείο που πληρεί τις προϋποθέσεις, προβλέπει μέσω του *Random Forest Classifier* ([*βλ. Ενότητα 3.*7](#_Αναγνώριση_ψηφίων)) τα ψηφία.

Έπειτα, υπολογίζει το ποσοστό επιτυχίας του εν λόγο ταξινομητή, όπου ως επιτυχία ορίζεται η εύρεση του σωστού ψηφίου.

Τέλος, αξίζει να αναφερθεί ότι στην παρούσα μετρική γίνεται χρήση ενός *Confusion Matrix*, o οποίος περιγράφει την απόδοση του συγκεκριμένου μοντέλου ταξινόμησης σε ένα σύνολο δοκιμαστικών αρχείων (test data), όπου συγκρίνονται οι πραγματικές με τις προβλεπόμενες τιμές.

Μετρικες εικονες

ΕΙΚΟΝΑ ΜΕ ΤΑ ΑΡΧΙΔΙΑ ΤΟΥ ΑΛΕΞ (1)

## Διαγράμματα

Τα διαγράμματα αποτελούν σημαντικό κομμάτι της οπτικοποίησης της εργασίας μας και δεν θα μπορούσαμε παρά να μην αναφερθούμε σε αυτά εκτενώς.

Αξίζει να σημειωθεί το γεγονός ότι τα διαγράμματα Κυματομορφής – Ενέργειας

### Διάγραμμα Κυματομορφής (Waveplot)

Με το διάγραμμα κυματομορφής - waveplot μπορούμε να δούμε την κυματορφή του εισαγόμενου σήματος, καθώς και τη διάρκεια αυτού.

Στην πράξη όμως η υλοποίηση γίνεται με την βοήθεια της βιβλιοθήκης της ***librosa*** και συγκεκριμένα της μεθόδου *librosa.display.waveplot,* η οποία σχεδιάζει τη κυματομορφή του σήματος.

### Διάγραμμα Ενέργειας (Root Mean Square Energy)

Το διάγραμμα της μέσης τετραγωνικής ρίζας της ενέργειας μας παρέχει χρήσιμες πληροφορίες, οι οποίες αξιοποιούνται από τον ταξινομητή Background vs Foreground ([*βλ. Ενότητα 3.5*](#_Background_vs_Foreground)) για την κατάτμηση του σήματος.

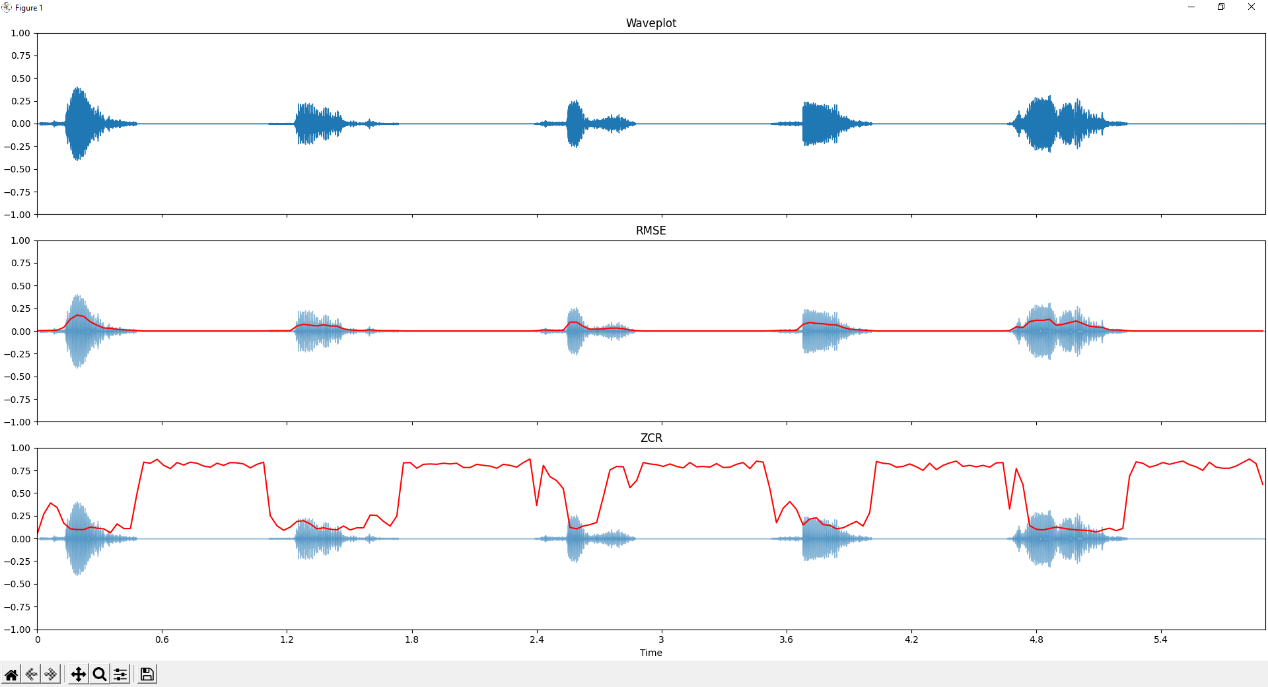
Η υλοποίηση του γίνεται με την βοήθεια της βιβλιοθήκης της ***librosa*** και συγκεκριμένα της μεθόδου *librosa.display.waveplot,* για την απεικόνιση της κυματομορφής του σήματος καθώς και της μεθόδου *librosa.frames\_to\_time*, για την για τη μετατροπή τους σε μονάδες χρόνου (δευτερόλεπτα).

### Διάγραμμα Zero Crossing Rate

Έπειτα, από το διάγραμμα του Zero Crossing Rate οπτικοποιούμε τη διαδικασία της εύρεσης του ρυθμού διέλευσης από το μηδέν και εξάγουμε χρήσιμες πληροφορίες, οι οποίες συνδυαστικά με την μέση τετραγωνική ρίζας της ενέργειας μας βοηθάνε στην εύρεση των φωνημάτων σε ένα ηχητικό σήμα.

Ομοίως με το αμέσως παραπάνω διάγραμμα, η υλοποίηση του παρόντος γίνεται με την βοήθεια της βιβλιοθήκης της ***librosa*** και συγκεκριμένα της μεθόδου *librosa.display.waveplot*, για την απεικόνιση της κυματομορφής του σήματος καθώς και της μεθόδου *librosa.frames\_to\_time*, για τη μετατροπή τους σε μονάδες χρόνου (δευτερόλεπτα).

Ακολουθεί μία ενδεικτική εικόνα από τα τρία παραπάνω διαγράμματα:

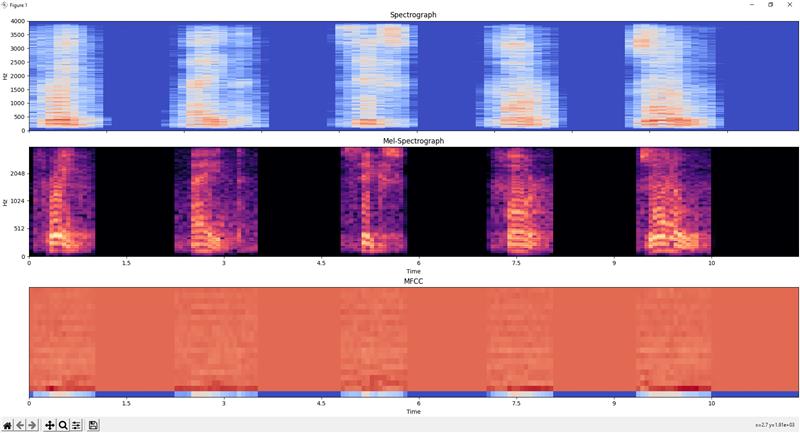


*Εικόνα 1: Διαγράμματα Waveplot - RMSE - ZCR*

### Διάγραμμα Spectrograph

### Διάγραμμα Mel-Spectrograph

### Διάγραμμα MFCC Χαρακτηριστικών



*Εικόνα 2: Διαγράμματα Spectrograph - Mel Spectrograph - MFCC*

# Συμπεράσματα

Μετά την εκτέλεση του προγράμματος καταλήγουμε στα εξής συμπεράσματα:

# Απαιτήσεις συστήματος και εκτέλεση

Η παραπάνω άσκηση υλοποιήθηκε στην γλώσσα προγραμματισμού ***Python, version 3.8****.* Για να μπορεί να γίνει η εκτέλεση του προγράμματος είναι απαραίτητο να είναι εγκατεστημένες οι βιβλιοθήκες:

* **librosa**
* **numpy**
* **sklearn**
* **matplotlib**
* **os**
* **time**

Εάν στον υπολογιστή σας δεν έχει γίνει εγκατάσταση των βιβλιοθηκών **librosa**, **numpy** προηγουμένως, τρέξτε τις εντολές ***«****pip install librosa****»***, **«***pip install numpy***»** στο command line.

Σε περίπτωση που η βιβλιοθήκη **sklearn** δεν είναι εγκατεστημένη, τότε τρέχουμε την εντολή **«***pip install scikit-sklearn***»** στο command line.

Επίσης, ο χρήστης θα πρέπει να έχει προβλέψει τα αρχεία του συνόλου εκπαίδευσης και το αρχείο εισόδου να είναι στον φάκελο **Dataset** και ffff, αντίστοιχα.

Oι συναρτήσεις που υλοποιήθηκαν βρίσκονται στο αρχείο **functions.py**. Το αρχείο που καλεί τις συγκεκριμένες συναρτήσεις και είναι υπεύθυνο για την εκτέλεση του προγράμματος είναι το αρχείο **main.py**.

Για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

1. Ανοίγουμε τη γραμμή εντολών και μεταβαίνουμε στον αντίστοιχο φάκελο**: \Ανάλυση Εικόνας**. (***Εικόνα 13*** )
2. Πληκτρολογούμε το όνομα του αρχείου **main.py**, ώστε να ανοίξει το εκτελέσιμο.(***Εικόνα 14*** )

To αρχείο εκτελείται και τα αποτελέσματα εμφανίζονται όπως έχει αναφερθεί στην **ενότητα 4**.

# Βιβλιογραφία

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.504.7782&rep=rep1&type=pdf

<https://www.asee.org/documents/zones/zone1/2008/student/ASEE12008_0044_paper.pdf>

etoima

[FIR Band System Theory & Design Examples](https://web.ece.ucsb.edu/~yoga/courses/DSP/P11_FIR_Design_Examples.pdf)

*(τελευταία προσπέλαση 02/07/2021)*

[Digital Signal Processing](https://www.berndporr.me.uk/teaching/dsp_handout45.pdf)

*(τελευταία προσπέλαση 02/07/2021)*

[Digital Audio Basics: Audio Sample Rate and Bit Depth](https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html)

*(τελευταία προσπέλαση 02/07/2021)*

[Sampling Rates, Sample Depths, and Bit Rates: Basic Audio Concepts](https://www.vocitec.com/docs-tools/blog/sampling-rates-sample-depths-and-bit-rates-basic-audio-concepts)

*(τελευταία προσπέλαση 03/07/2021)*

[NumPy v1.21 Documentation](https://numpy.org/doc/1.21/)

*(τελευταία προσπέλαση 08/07/2021)*

[Librosa v0.8.1 Documentation](https://librosa.org/doc/latest/index.html)

*(τελευταία προσπέλαση 14/07/2021)*

[Scikit-Learn v0.24.2](https://scikit-learn.org/0.21/documentation.html)

*(τελευταία προσπέλαση 12/07/2021)*

[Matplotlib v3.4.2](https://matplotlib.org/stable/contents.html)

*(τελευταία προσπέλαση 12/07/2021)*

[MFCC Features](https://link.springer.com/content/pdf/bbm%3A978-3-319-49220-9%2F1.pdf)

*(τελευταία προσπέλαση 11/07/2021)*

[Semantic Scholar - A free, AI-powered research tool for scientific literature](https://www.semanticscholar.org/)

*(τελευταία προσπέλαση 05/07/2021)*

[A tutorial on signal energy and its applications – Rodrigo Capobianco Guido (via Semantic Scholar)](https://www.semanticscholar.org/paper/A-tutorial-on-signal-energy-and-its-applications-Guido/0db0b876da9aa7227d81ff62edd5d12fd90913ea)

*(τελευταία προσπέλαση 05/07/2021)*

[Detection and Recognition Threshold of Sound Sources in Noise - Tjeerd Andringa, Carina Pal (via Semantic Scholar)](https://www.semanticscholar.org/paper/Detection-and-Recognition-Threshold-of-Sound-in-Andringa/a9a50597f8053fae3b2fe416ab63632c10f168c0)

*(τελευταία προσπέλαση 06/07/2021)*

[BF-Classifier: Background/Foreground Classification and Segmentation of Soundscape Recordings - Miles Thorogood, Jianyu Fan, Philippe Pasquier (via Semantic Scholar)](https://www.semanticscholar.org/paper/BF-Classifier%3A-Background%2FForeground-Classification-Thorogood-Fan/575cc8aee5f8b32e44ff2b48eb59265b0c9d8b82)

*(τελευταία προσπέλαση 10/07/2021)*

[Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals - Madiha Jalil, F. A. Butt, Ahmed Malik (via Semantic Scholar)](https://www.semanticscholar.org/paper/Short-time-energy%2C-magnitude%2C-zero-crossing-rate-of-Jalil-Butt/d4c67bff56f24392577eebfc5027f3927cd47f5d)

*(τελευταία προσπέλαση 08/07/2021)*

[Separation of Voiced and Unvoiced using Zero crossing rate and](https://www.semanticscholar.org/paper/1-Separation-of-Voiced-and-Unvoiced-using-Zero-rate-Bachur/9aaaa022fc58a2791904a15dd4c2658793c39c37)

[Energy of the Speech Signal - Bachu R.G., Kopparthi S., Adapa B., Barkana B.D. (via Semantic Scholar)](https://www.semanticscholar.org/paper/1-Separation-of-Voiced-and-Unvoiced-using-Zero-rate-Bachur/9aaaa022fc58a2791904a15dd4c2658793c39c37)

*(τελευταία προσπέλαση 09/07/2021)*