



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΩΝ ΦΩΝΗΣ ΚΑΙ ΗΧΟΥ

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

ΟΜΑΔΑ ΑΝΑΠΤΥΞΗΣ:

- **ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ – Π17172**
- **ΑΝΔΡΕΑΣ ΘΕΟΔΩΡΙΔΗΣ – Π17164**
- **ΒΑΣΙΛΕΙΟΣ ΖΑΡΤΗΛΑΣ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ – Π17168**

Πειραιάς, Ιούλιος 2021

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	2
1. ΕΙΣΑΓΩΓΗ.....	3
2. ΘΕΜΑ ΠΡΩΤΟ.....	5
2.1. ΥΠΟΕΡΩΤΗΜΑ ΠΡΩΤΟ.....	5
ΕΚΦΩΝΗΣΗ	5
ΥΛΟΠΟΙΗΣΗ	5
ΑΠΟΤΕΛΕΣΜΑ	9
2.2. ΥΠΟΕΡΩΤΗΜΑ ΔΕΥΤΕΡΟ.....	14
ΕΚΦΩΝΗΣΗ	15
ΥΛΟΠΟΙΗΣΗ	15
ΑΠΟΤΕΛΕΣΜΑ	17
2.3. ΥΠΟΕΡΩΤΗΜΑ ΤΡΙΤΟ.....	19
ΕΚΦΩΝΗΣΗ	19
ΥΛΟΠΟΙΗΣΗ	19
ΑΠΟΤΕΛΕΣΜΑ	20
3. ΘΕΜΑ ΔΕΥΤΕΡΟ.....	21
3.1. ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΠΙΣΜΟΥ ΣΥΜΒΑΝΤΩΝ ΗΧΟΥ.....	21
4. ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΠΗΓΕΣ	24
5. ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ	25
6. ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ.....	27

1. ΕΙΣΑΓΩΓΗ

Ένα **ASR** σύστημα είναι η διαδικασία εξαγωγής της μεταγραφής (ακολουθίας λέξεων) μιας πρότασης, δεδομένης της κυματομορφής της ομιλίας.

Αρχικά, έγινε ανάλυση της διαδικασίας που ακολουθεί το ASR για να μπορέσουμε να καταλάβουμε πως λειτουργεί. Όπως φαίνεται στο πιο κάτω διάγραμμα ροής δεδομένων, αυτή είναι η διαδικασία που ακολουθεί το σύστημα μας.

Εφόσον πάρουμε το σήμα μας γίνεται η προ-διεργασία του σήματος «**pre-processing**», δηλαδή, θα τροποποιήσουμε το σήμα, έτσι ώστε να είναι πιο αποδεκτό για εξαγωγή χαρακτηριστικών. Κάποια χαρακτηριστικά τα οποία πρέπει να ληφθούν υπόψη πριν την εκπαίδευση είναι οι **Mel-Frequency Cepstral Coefficients**, το **Zero-Crossing Rate** και η **ενέργεια** (Ενέργεια Παραθύρου).

Πρώτα, υπολογίσουμε την Ενέργεια Παραθύρου – Short-Time Energy, που σε ένα σήμα παρέχει μια βολική αναπαράσταση που αντανακλά τη διακύμανση του εύρους (Amplitude) και μπορεί να οριστεί ως,

$$E_n = \sum_{m=-\infty}^{\infty} [x(m) \cdot W(n-m)]^2$$

όπου $x(m)$ είναι το σήμα εισόδου και $W(n-m)$ το παράθυρο που θα απομονώσει το μέρος του σήματος.

Κατόπιν, υπολογίζουμε την μηδενική διασταύρωση – Zero-Crossing Rate. Ειδικότερα, το ZCR καταμετρά τις φορές που το εύρος των σημάτων ομιλίας σε ένα δεδομένο χρονικό διάστημα/frame διασχίζει ένα σιωπηλό κατώφλι (silent threshold) το οποίο αντικατοπτρίζεται με την τιμή μηδέν. Επίσης, αποτελεί βασικό χαρακτηριστικό για την ταξινόμηση των ήχων και όταν υπάρχει ένας μεγάλος αριθμός μηδενικών διασταυρώσεων συνεπάγεται ότι δεν υπάρχει κυρίαρχη ταλάντωση χαμηλής συχνότητας.

Το ZCR ορίζεται σαν,

$$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} 1_{\mathbb{R}<0}(s_t s_{t-1})$$

όπου s είναι ένα σήμα μήκους T και το $1_{\mathbb{R}<0}$ είναι μια συνάρτηση ένδειξης.

Μετά από αυτό, ο μέσος ρυθμός μηδενικής διασταύρωσης με τον οποίο εμφανίζονται μηδενικές διασταυρώσεις είναι ένα απλό μέτρο του περιεχομένου συχνότητας ενός σήματος και ορίζεται σαν,

$$Z_n = \sum_{m=-\infty}^{\infty} |sgn(x[n]) - sgn(x[n-1])| \cdot W[n-m]$$

όπου, για την συνάρτηση πρόσημου,

$$sgn(x[n]) = \begin{cases} 1, & x[n] \geq 0 \\ -1, & x[n] < 0 \end{cases}$$

και

$$W[n] = \begin{cases} \frac{1}{2N}, & 0 \leq n \leq N-1 \\ 0, & \text{αλλού} \end{cases}$$

2. ΘΕΜΑ ΠΡΩΤΟ

2.1. ΥΠΟΕΡΩΤΗΜΑ ΠΡΩΤΟ

ΕΚΦΩΝΗΣΗ

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.

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

Βασικά Χαρακτηριστικά:

- Δώστε έμφαση στην επεξεργασία του σήματος, προτού αρχίσουν τα στάδια κατάτμησης/αναγνώρισης (π.χ., με κατάλληλα φίλτρα, αλλαγή ρυθμού δειγματοληψίας, κλπ).
- 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.

ΥΛΟΠΟΙΗΣΗ

Πρώτα δημιουργήσαμε τα αρχεία `main.py` και `constants.py`, όπου το `main` θα λειτουργεί ως το αρχείο που θα καλεί τις κλάσεις από τα υπόλοιπα και θα εκτελεί την διαδικασία μέχρι το τέλος. Στο αρχείο `constants.py` αποθηκεύονται σταθερές μεταβλητές έτσι ώστε να είναι πιο οργανωμένος ο κώδικας και ταυτόχρονα για να μπορούμε να χρησιμοποιήσουμε αυτές τις σταθερές σε όλα τα αρχεία του προγράμματος. Επίσης στο αρχείο `utils.py` βρίσκονται όλες οι μέθοδοι για την επεξεργασία του σήματος, ο ταξινομητής, τα φίλτρα και γενικότερα όλα όσα χρειαστήκαμε για την υλοποίηση της εργασίας. Τέλος στο αρχείο `plots.py` βρίσκεται ο κώδικας για την εμφάνιση των γραφικών μας.

Στην συνέχεια, όταν γίνεται εκτέλεση του αρχείου `main` τότε θα εμφανιστεί μήνυμα στο console του χρήστη για να δώσει το μονοπάτι που βρίσκεται το αρχείο ήχου. Εάν το μονοπάτι που βρίσκεται το αρχείο ήχου που θα δοθεί δεν είναι έγκυρο και παράλληλα το αρχείο δεν είναι της μορφής ήχου .wav (Wave Audio File) τότε το πρόγραμμα θα τερματίσει εμφανίζοντας το ανάλογο μήνυμα λάθους.

Εφόσον δεν εμφανιστεί κάποιο λάθος στον χρήστη, το πρόγραμμα θα προχωρήσει με την βιβλιοθήκη `librosa`, συγκεκριμένα με την εντολή,

```
librosa.load(μονοπάτι_αρχείου, δειγματοληψία=16000)
```

Όπου για την παράμετρο της **δειγματοληψίας**: R_* , θα μετατρέψουμε την δειγματοληψία του αρχείου δια τον λόγο ότι η ομιλία έχει σχετικά χαμηλό εύρος ζώνης (κυρίως μεταξύ 100Hz-8kHz), π.χ. τα **8,000** δείγματα/δευτ. (8kHz) αρκούν για τα περισσότερα βασικά **ASR**. Ωστόσο, εμείς θα προτιμήσουμε **16,000** δείγματα/δευτ. (16kHz) επειδή παρέχουν πιο ακριβής πληροφορία υψηλής συχνότητας.

Συνεχίζοντας, παίρνουμε το **σήμα**: `nparray`, από το αρχείο που είναι ένας σταθερού μεγέθους πολυδιάστατος πίνακας αντικειμένων του ίδιου τύπου (float-32) και μεγέθους και καλούμε την συνάρτηση `pre_processing/2` που παίρνει σαν ορίσματα το σήμα και το όνομα του αρχείου που δόθηκε.

Αναλυτικότερα, στο σημείο της προ-διεργασίας εκτελούνται οι διαδικασίες που αναφέραμε πιο πάνω (βλέπε [Εισαγωγή](#)) με την μόνη διαφορά ότι βρίσκουμε και την προ-έμφαση του σήματος με την πιο κάτω εντολή με την βοήθεια της βιβλιοθήκης `librosa`.

```
librosa.effects.preemphasis(σήμα)
```

που εξάγετε με ένα φίλτρο αυτόματης παλινδρόμησης πρώτης τάξης.

Η προ-έμφαση πραγματοποιείται για την ισοπέδωση του φασματικού μεγέθους και την εξισορρόπηση των στοιχείων υψηλής και χαμηλής συχνότητας. Αυξάνει τη συνιστώσα υψηλών συχνοτήτων, βελτιώνοντας έτσι την αναλογία σήματος προς θόρυβο, προτού μεταδοθούν ή καταγραφούν σε μέσο αποθήκευσης και ορίζεται με,

$$x'[n] = x[n] - k \cdot x[n - 1]$$

όπου k είναι ο συντελεστής της προ-έμφασης που πρέπει να κυμαίνεται από $0 \leq k \leq 1$, η προκαθορισμένη τιμή είναι $k = 0,97$.

Μετά από αυτό, χρησιμοποιούμε την βιβλιοθήκη «[noisereduce](#)» για να αφαιρέσουμε τα σημεία που περιέχουν θόρυβο στο σήμα μας. Ο αλγόριθμος της βιβλιοθήκης λειτουργεί ως εξής:

- Υπολογίζεται ένας «Ταχύς Μετασχηματισμός Fourier (FFT)» πάνω στο σήμα.
- Τα στατιστικά στοιχεία υπολογίζονται σε FFT του θορύβου (σε συχνότητα).
- Το κατώτατο όριο (threshold) υπολογίζεται με βάση τα στατιστικά στοιχεία του θορύβου (και την επιθυμητή ευαισθησία του αλγορίθμου).
- Η μάσκα προσδιορίζεται συγκρίνοντας το σήμα FFT με το threshold.
- Η μάσκα λειαιίνει με ένα φίλτρο σε συχνότητα και χρόνο.
- Η μάσκα εφαρμόζεται στο FFT του σήματος, και είναι ανεστραμμένη.

Στην ουσία, ο FFT είναι ένας αλγόριθμος που υπολογίζει τον Διακριτό Μετασχηματισμό Fourier (**DFT**) μιας ακολουθίας ή τον αντίστροφο με τον αλγόριθμο FFT για να αποκτήσουμε πληροφορία σχετικά με την συχνότητα του σήματος. Ο DFT ορίζεται ως,

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} \cdot kn}$$

όπου $e^{-\frac{i2\pi}{N} \cdot kn}$ είναι μια πρώτη νιοστή ρίζα του 1.

Εκτός από αυτό, θα πραγματοποιήσουμε περικοπή των σημείων του αρχείου με την εντολή,

```
librosa.effects.trim(db=40, σήμα)
```

Δεδομένου του ρυθμού δειγματοληψίας (sample rate) 16,000, θα αφαιρούνται τα σημεία που ανιχνεύθηκε ήχος χαμηλότερος από 40dB για περίοδο 1 δευτερολέπτου.

Εφόσον γίνουν τα πιο πάνω, η προ-διεργασία έχει ολοκληρωθεί και συνεχίζουμε στην κατάτμηση του σήματος. Μέσω της μεθόδου `digits_segmentation()` όπου βρίσκεται στο αρχείο `utils` γίνεται η κατάτμηση. Αρχικά αντιστρέφουμε το σήμα για να μπορέσουμε να λάβουμε το τέλος του κάθε συμβάντος. Στην συνέχεια υπολογίζουμε το μήκος πλαισίου πολλαπλασιάζοντας τα δείγματα (L) με την ολίσθηση πλαισίου (R). Στο πρόγραμμα μας οι μεταβλητές αυτές βρίσκονται στο αρχείο `constants` ως **WINDOW_LENGTH** = 0,03 για την τιμή L και **WINDOW_HOP** = 0,01 για την τιμή R. Οι τιμές έχουν δηλωθεί και έχουν πάρει τις αντίστοιχες τιμές κατά αυτόν το τρόπο βάση της βιβλιογραφίας μας.

Συγκεκριμένα το βιβλίο[1] στην **σελίδα 792** αναφέρει ρητά:

«Στην συνέχεια, το φιλτραρισμένο σήμα τμηματοποιείται σε πλαίσια των L δειγμάτων, όπου τα διαδοχικά πλαίσια απέχουν R δείγματα. Οι τυπικές τιμές για τις παραμέτρους L και R αντιστοιχούν σε πλαίσια διάρκειας 15-40 msec, όπου η ολίσθηση πλαισίου είναι συνήθως 10 msec».

Μετάπειτα, μέσω της βιβλιοθήκης librosa κάνουμε κάποιες μετατροπές η οποίες μα βοηθούν στην κατάτμηση του σήματος. Αρχικά, μέσω της μεθόδου onset_detect εντοπίζουμε τα συμβάντα έναρξης κάθε ήχου. Επίσης, υπολογίζονται τα συμβάντα έναρξης κάθε ήχου και για το αντεστραμμένο σήμα. Επίσης, μετατρέπουμε και την ποσότητα των frames που έχουμε σε χρόνο (δευτερόλεπτα), το ίδιο κάνουμε και στο αντεστραμμένο σήμα. Ακολουθως, μετατρέπουμε τους δείκτες των frames σε δείκτες ήχου.

Κατά συνέπεια, έχουμε τρεις(3) επαναληπτικούς βρόγχους. Ο πρώτος επαναληπτικός βρόγχος χρησιμοποιείται για τον υπολογισμό κάθε αντίστροφου frame. Αφαιρούμε τον χρόνο από το μήκος του παράθυρου για να το συγκρίνουμε αργότερα με τον κανονικό χρόνο του σήματος για να γίνει η επιτυχής κατάτμηση των ψηφίων.

Ο δεύτερος και τρίτος βρόγχος βοηθούν στην διαγραφή τυχών background θορύβων (πλαίσια που χρονικά δεν επαρκούν για να θεωρηθούν ψηφία) για να γίνει πιο σωστή η κατάτμηση των ψηφίων. Ο δεύτερος βρόγχος κάνει τον έλεγχο στο αντίστροφο σήμα και ο τρίτος βρόγχος για το κανονικό σήμα.

Όπως βλέπουμε στην παρακάτω φωτογραφία ο αλγόριθμος ελέγχει το αντίστροφο σήμα, την αρχή και το τέλος κάθε πλαισίου. Αν το τέλος του πλαισίου όταν το αφαιρέσουμε από την αρχή του πλαισίου είναι μικρότερο από ένα τότε δεν λαμβάνεται ως πλαίσιο όπου είναι άξιο χρονικά για να θεωρηθεί ψηφίο. Μόνο στην Επανάληψη 9 βλέπουμε ότι είναι μεγαλύτερο άρα αναγνωρίζεται ως ψηφίο.

Να σημειώσουμε ότι στο τέλος αναφέρει ότι τα συνολικά ψηφία είναι 3 επειδή τα άλλα 2 ψηφία αναγνωρίζονται από τον έλεγχο του κανονικού σήματος όπως και εξηγούμε πιο κάτω.


```

Ελεγχος αν ένα παράθυρο πλαισίου έχει επαρκή διάρκεια για να θεωρηθεί ψηφίο:
Χρόνοι πριν τον έλεγχο: [-4.35, -3.66, -3.5100000000000002, -2.8200000000000003, -2.7, -1.74, -1.5, -1.08, -0.96, -0.6599999999999999, 4.65]
Επανάληψη 1
Χρόνοι μετά τον έλεγχο: [-3.66 -3.51 -2.82 -2.7 -1.74 -1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 2
Χρόνοι μετά τον έλεγχο: [-3.51 -2.82 -2.7 -1.74 -1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 3
Χρόνοι μετά τον έλεγχο: [-2.82 -2.7 -1.74 -1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 4
Χρόνοι μετά τον έλεγχο: [-2.7 -1.74 -1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 5
Χρόνοι μετά τον έλεγχο: [-1.74 -1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 6
Χρόνοι μετά τον έλεγχο: [-1.5 -1.08 -0.96 -0.66 4.65]
Επανάληψη 7
Χρόνοι μετά τον έλεγχο: [-1.08 -0.96 -0.66 4.65]
Επανάληψη 8
Χρόνοι μετά τον έλεγχο: [-0.96 -0.66 4.65]
Επανάληψη 9
Χρόνοι μετά τον έλεγχο: [-0.66 4.65]
Επανάληψη 10
[!] Total Digits Found: 3

```

Εικόνα 2.1

Στην παρακάτω φωτογραφία βλέπουμε την διαδικασία με το κανονικό σήμα. Αναγνωρίζουμε λοιπόν ότι ηχογραφήθηκαν ακόμα 2 ψηφία, για αυτό δεν υπάρχουν κι οι επαναλήψεις 3 και 5 εφόσον βλέπουμε στα πράσινα κουτάκια ότι η χρονική διάρκεια του σήματος επαρκεί για να θεωρηθεί ψηφίο.

```

Ελεγχος αν ένα παράθυρο πλαισίου έχει επαρκή διάρκεια για να θεωρηθεί ψηφίο:
Χρόνοι πριν τον έλεγχο: [0.06 0.57 0.66 1.71 2.4 3.93 4.35 4.53 5.31]
Επανάληψη 1
Χρόνοι μετά τον έλεγχο: [0.06 0.66 1.71 2.4 3.93 4.35 4.53 5.31]
Επανάληψη 2
Χρόνοι μετά τον έλεγχο: [0.06 1.71 2.4 3.93 4.35 4.53 5.31]
Επανάληψη 3
Επανάληψη 4
Χρόνοι μετά τον έλεγχο: [0.06 1.71 3.93 4.35 4.53 5.31]
Επανάληψη 5
Επανάληψη 6
Χρόνοι μετά τον έλεγχο: [0.06 1.71 3.93 4.53 5.31]
Επανάληψη 7
Χρόνοι μετά τον έλεγχο: [0.06 1.71 3.93 5.31]

```

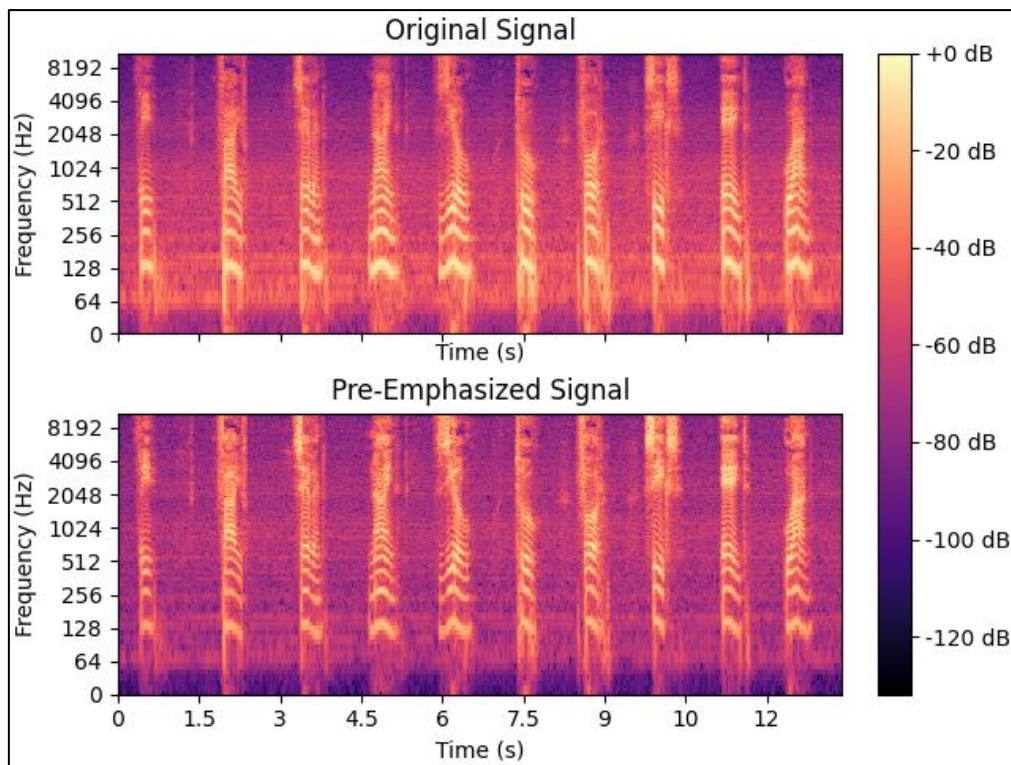
Εικόνα 2.2

Στην μέθοδο `valid_digits()` βρίσκουμε τα ψηφία από το σήμα.

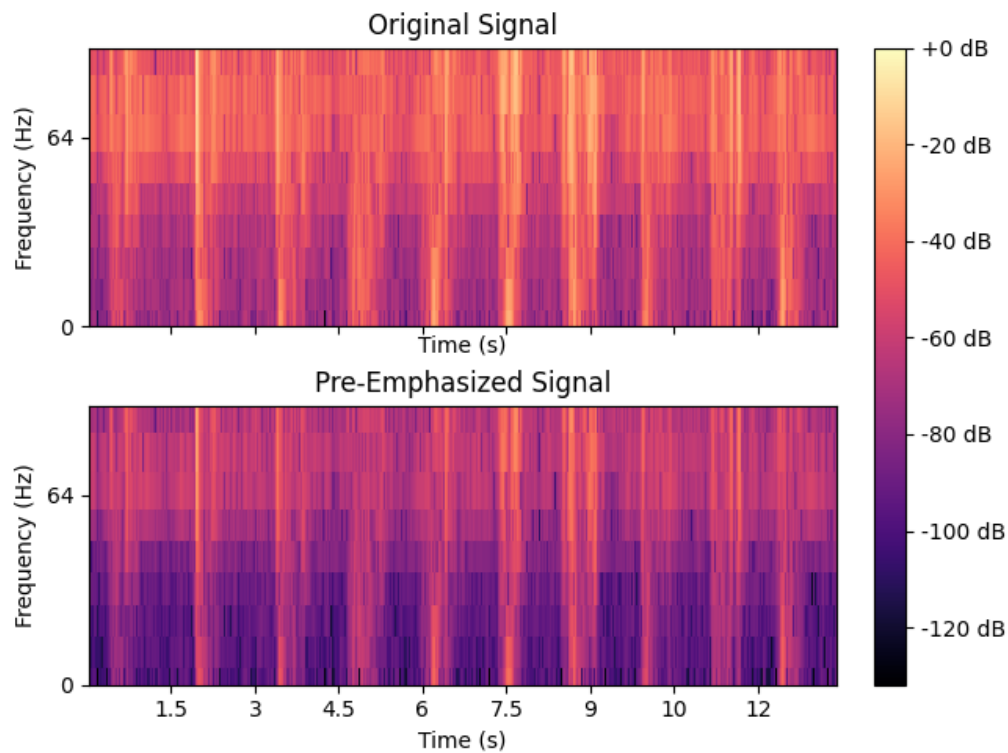
Στην αρχή δηλώνουμε ένα μετρητή για να μετράμε τα ψηφία που αναγνωρίζουμε. Στον επαναληπτικό βρόγχο που έχουμε δηλώσαμε την μεταβλητή `i` να αυξάνεται ανά δύο δια τον λόγο ότι παίρνουμε τον συνολικό αριθμό δειγμάτων και κάθε πλαίσιο έχει μια αρχή και ένα τέλος, άρα σύνολο κάθε πλαίσιο (ψηφίο) θα αποτελείτε από δύο δείγματα.

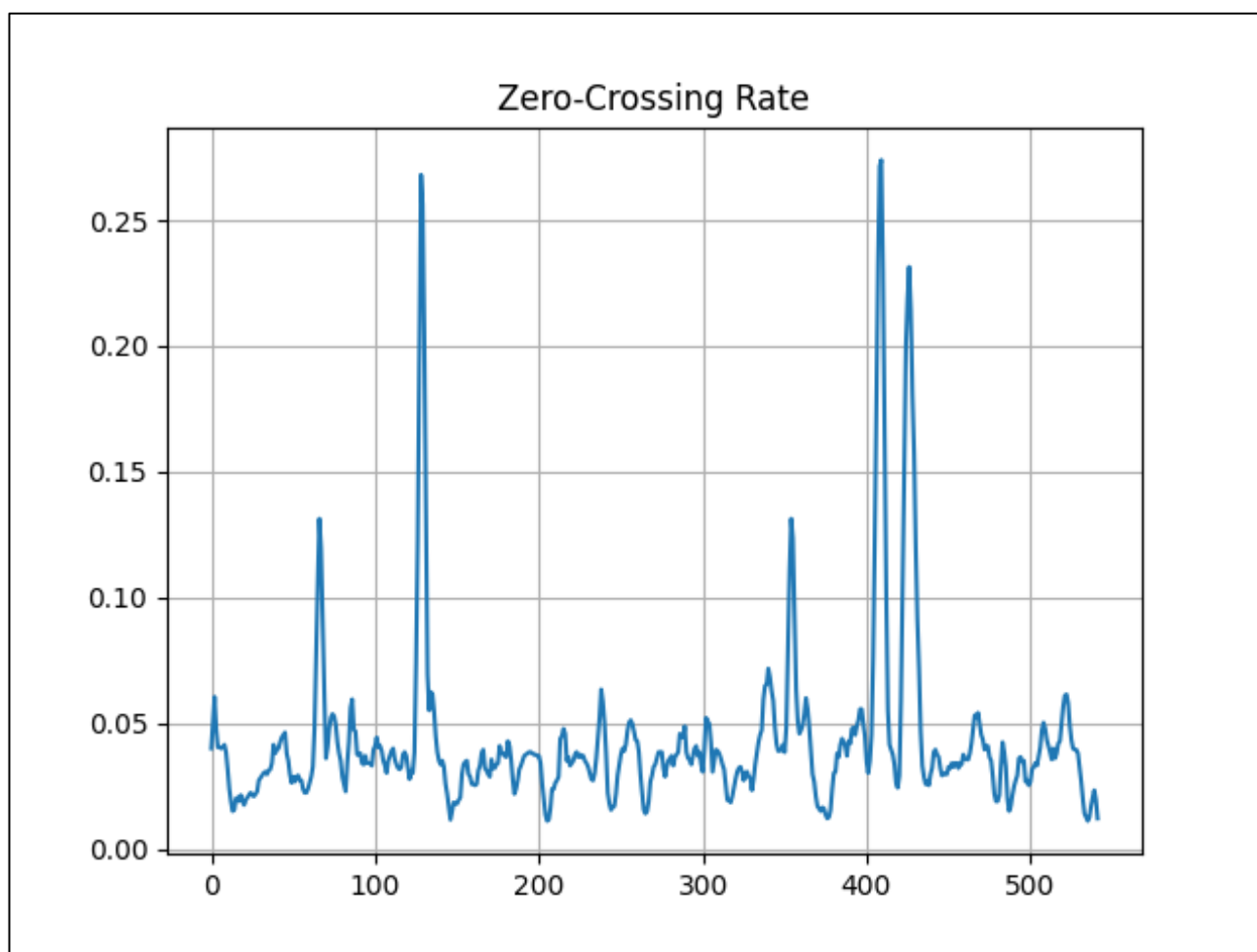
Για κάθε συνδυασμό δειγμάτων, γίνεται έλεγχος για την περίπτωση που έχουμε μονό αριθμό συνολικών δειγμάτων. Στην περίπτωση αυτή δεχόμαστε το πλαίσιο που βρίσκεται ανάμεσα στο δείγμα που έχουμε και το προηγούμενο. Αυτή η περίπτωση μπορεί να συμβεί μόνο στο τελευταίο δείγμα. Στην περίπτωση που ο αριθμός των συνολικών δειγμάτων είναι ζυγός δεχόμαστε το πλαίσιο που βρίσκεται ανάμεσα από το δείγμα που είμαστε και το επόμενο όπου αντικατοπτρίζει την αρχή και το τέλος ενός ψηφίου. Έτσι καταμετρούμε κάθε ψηφίο που λαμβάνουμε από το σήμα.

ΑΠΟΤΕΛΕΣΜΑ

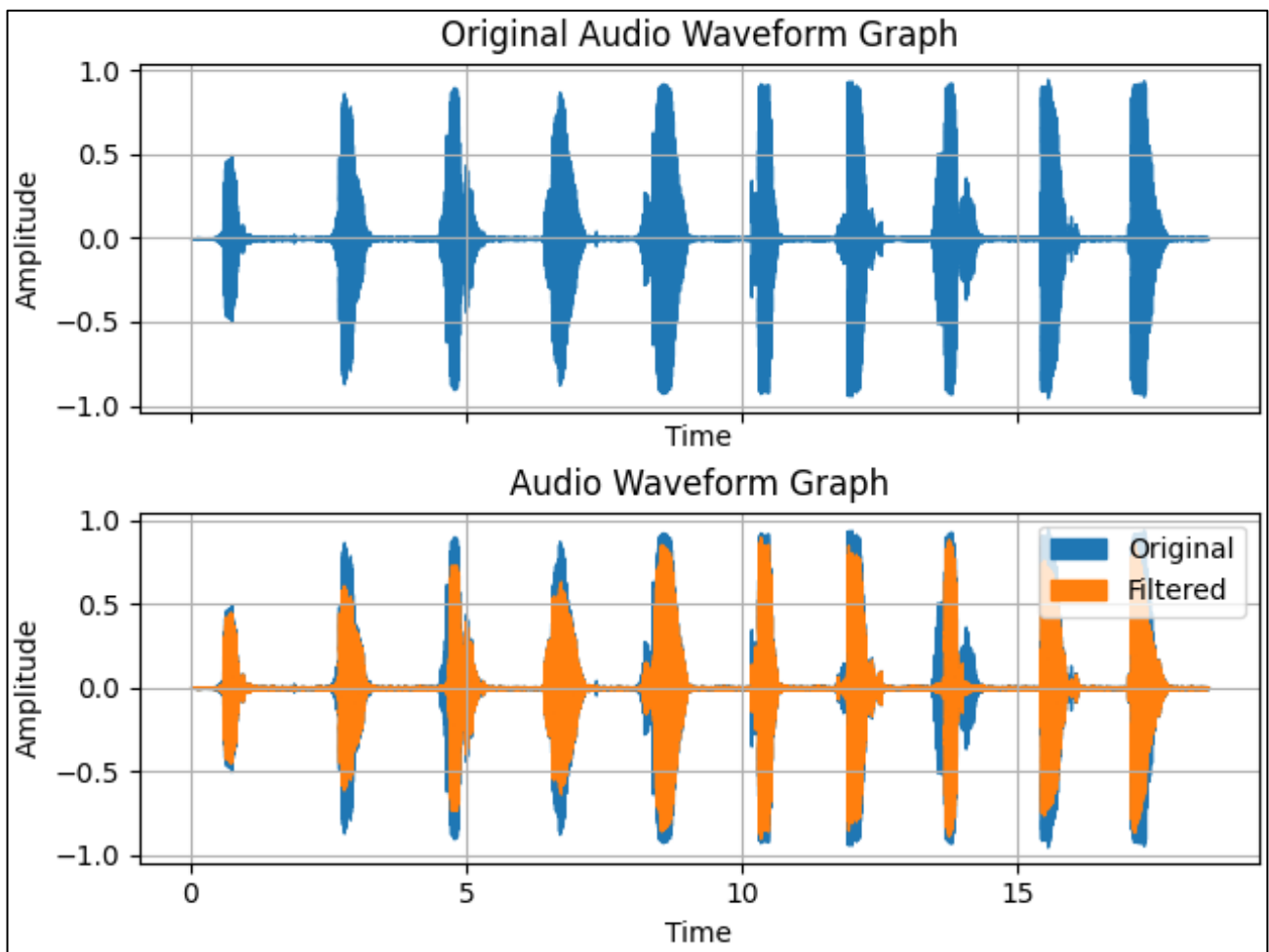


Σχήμα 2.1

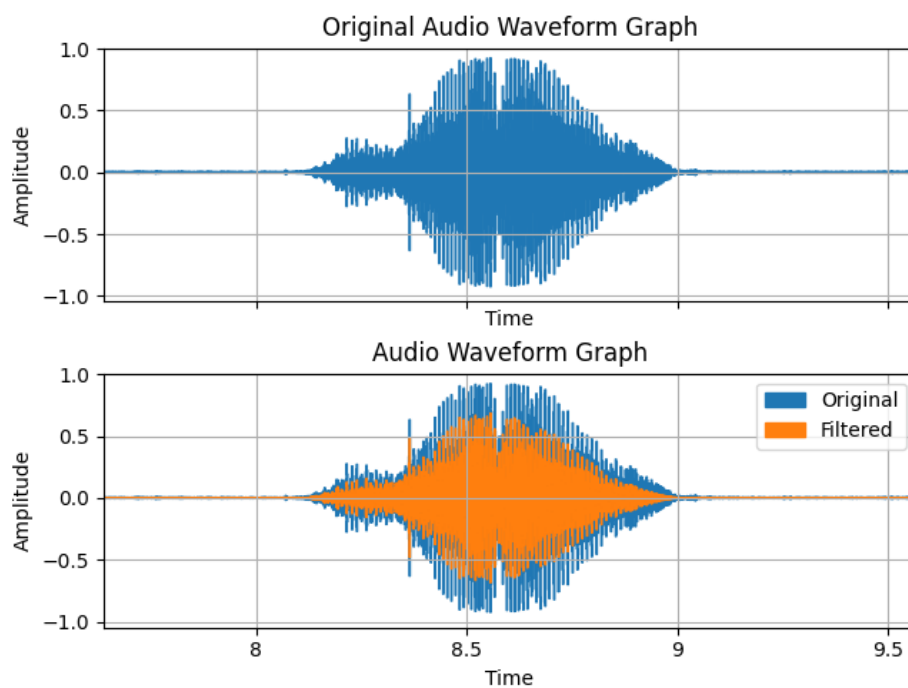


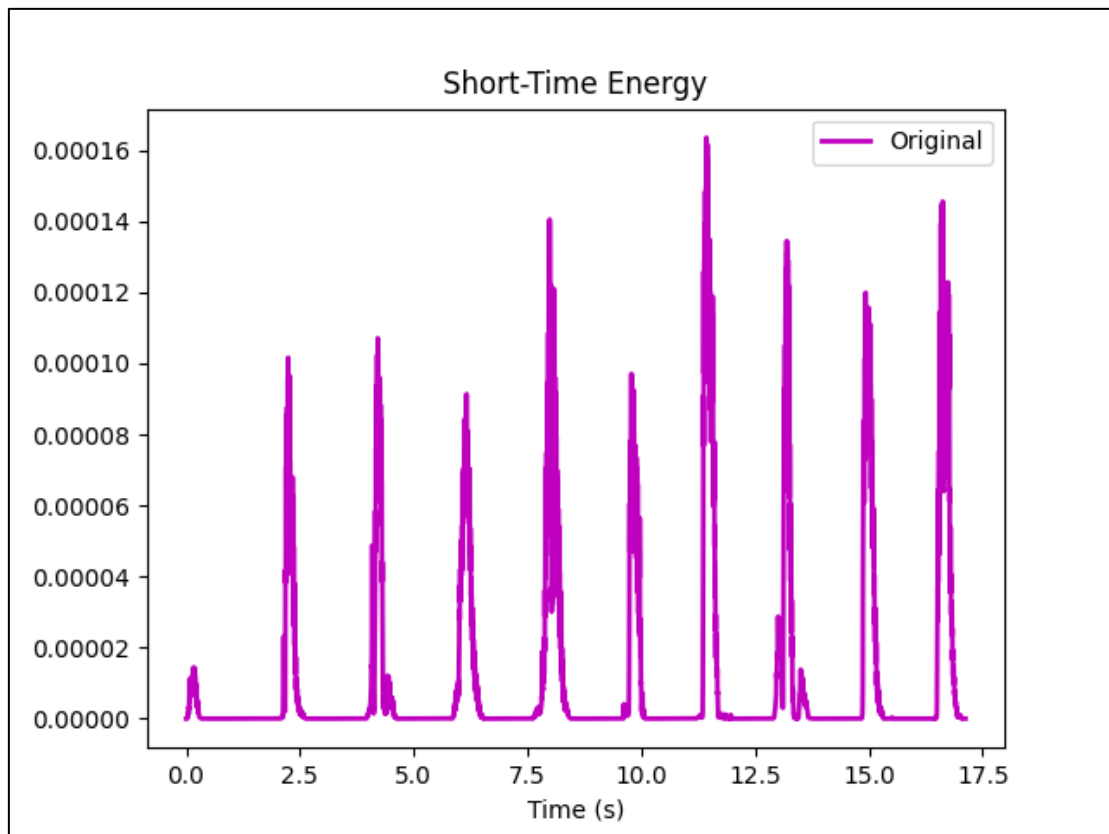


Σχήμα 2.2



Σχήμα 2.3





Σχήμα 2.4

```
Run - Automatic-Speech-Recognition
Run: main x
"G:\Downloads (HDD)\University\8th Semester\SPEECH AND AUDIO PROCES
Enter sound file path:
data/samples/sample-1.wav
=====
STATISTICS:
- Sample Rate: 16000
- Original Audio Duration: 18.45 s
- Audio (Filtered) Duration: 17.12 s
- Average ZCR: 0.09001683477145522
=====
Process finished with exit code 0
```

Εικόνα 2.3

2.2. ΥΠΟΕΡΩΤΗΜΑ ΔΕΥΤΕΡΟ

ΕΚΦΩΝΗΣΗ

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.

ΥΛΟΠΟΙΗΣΗ

Για την εκπαίδευση του αλγορίθμου ηχογραφήσαμε κάθε ψηφίο από τρεις (3) φορές σε διαφορετικά αρχεία ήχου. Δηλαδή, έχουμε συνολικά τριάντα (30) αρχεία τύπου WAV.

Μέσω της μεθόδου `recognition()` γίνεται η αναγνώριση των ψηφίων στο σήμα που έχει δώσει σαν είσοδο ο χρήστης. Δημιουργούμε ένα πίνακα για να αποθηκεύσουμε τα ψηφία που έχουν αναγνωριστεί. Μετά δημιουργούμε ένα βρόγχο για κάθε ψηφίο και χρησιμοποιούμε μέσω της βιβλιοθήκης `librosa`, το χαρακτηριστικό MFCC (Mel-frequency cepstral coefficients). Το MFCC αντιπροσωπεύει ξεχωριστές μονάδες ήχου καθώς το σχήμα του φωνητικού συστήματος το οποίο είναι υπεύθυνο για την παραγωγή ήχου.

Στο πλαίσιο αυτό, θα υπολογίσουμε το Mel-Spectrogram. Το Mel-Spectrogram λοιπόν, αποδίδει λογαριθμικά συχνότητες πάνω από ένα συγκεκριμένο όριο (η γωνιακή συχνότητα). Ένα Mel-Spectrogram κάνει δύο σημαντικές αλλαγές σε σχέση με ένα κανονικό Spectrogram που απεικονίζει τη συχνότητα έναντι του χρόνου.

- Χρησιμοποιεί την κλίμακα Mel αντί της συχνότητας στον άξονα y.
- Χρησιμοποιεί την κλίμακα dB αντί για το Amplitude (Πλάτος) για να υποδείξει τα χρώματα.

Ένα φασματογράφημα κόβει τη διάρκεια του ηχητικού σήματος σε μικρότερα χρονικά τμήματα και στη συνέχεια εφαρμόζει τον μετασχηματισμό Fourier σε κάθε τμήμα, για να προσδιορίσει τις συχνότητες που περιέχονται σε αυτό το τμήμα. Στη συνέχεια συνδυάζει τους μετασχηματισμούς Fourier για όλα αυτά τα τμήματα σε ένα ενιαίο διάγραμμα

Σχεδιάζει τη Συχνότητα (άξονας y) έναντι του χρόνου (άξονας x) και χρησιμοποιεί διαφορετικά χρώματα για να δείξει το πλάτος κάθε συχνότητας. Όσο πιο φωτεινό είναι το χρώμα τόσο μεγαλύτερη είναι η ενέργεια του σήματος. Δυστυχώς, όταν προβάλλουμε αυτό το φασματογράφημα, δεν προβάλλει πολλές πληροφορίες για να δούμε. Συμπερασματικά, τροποποιήσουμε το φασματογράφημα μας για να χρησιμοποιήσουμε την κλίμακα Mel αντί για τη συχνότητα και με την πιο κάτω εντολή, διαχωρίζουμε ένα πολύπλοκο φασματογράφημα D στα συστατικά μεγέθους (S) και φάσης (P), έτσι ώστε $D = S \cdot P$.

```
librosa.magphase(D)
```

Και ακολούθως,

```
librosa.feature.melspectrogram(S=φασματογράφημα, sr=δειγματοληψία)
```

υπολογίζουμε ένα φασματογράφημα με κλίμακα mel Όπου το φασματογράφημα S , χαρτογραφείται απευθείας στη βάση mel. Το φασματογράφημα μεγέθους S υπολογίζεται πρώτα και, στη συνέχεια, χαρτογραφείται στην κλίμακα .

Το φασματογράφημα μας είναι καλύτερο από πριν, αλλά το μεγαλύτερο μέρος του φασματογράφου είναι ακόμα σκοτεινό και δεν έχει αρκετές χρήσιμες πληροφορίες άρα το τροποποιούμε για να χρησιμοποιήσουμε την κλίμακα dB αντί για το Amplitude (Πλάτος) με την πιο κάτω εντολή.

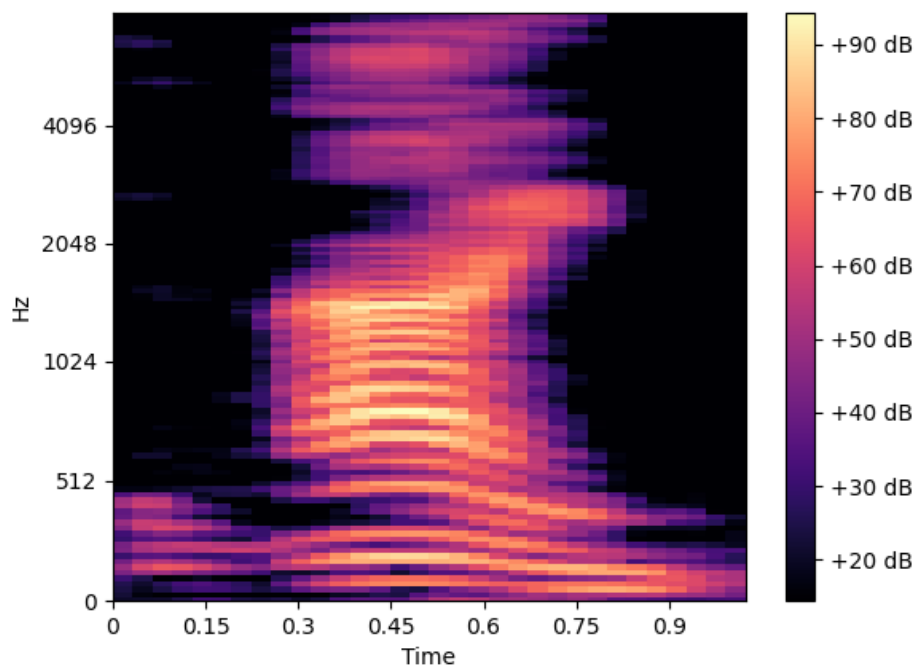
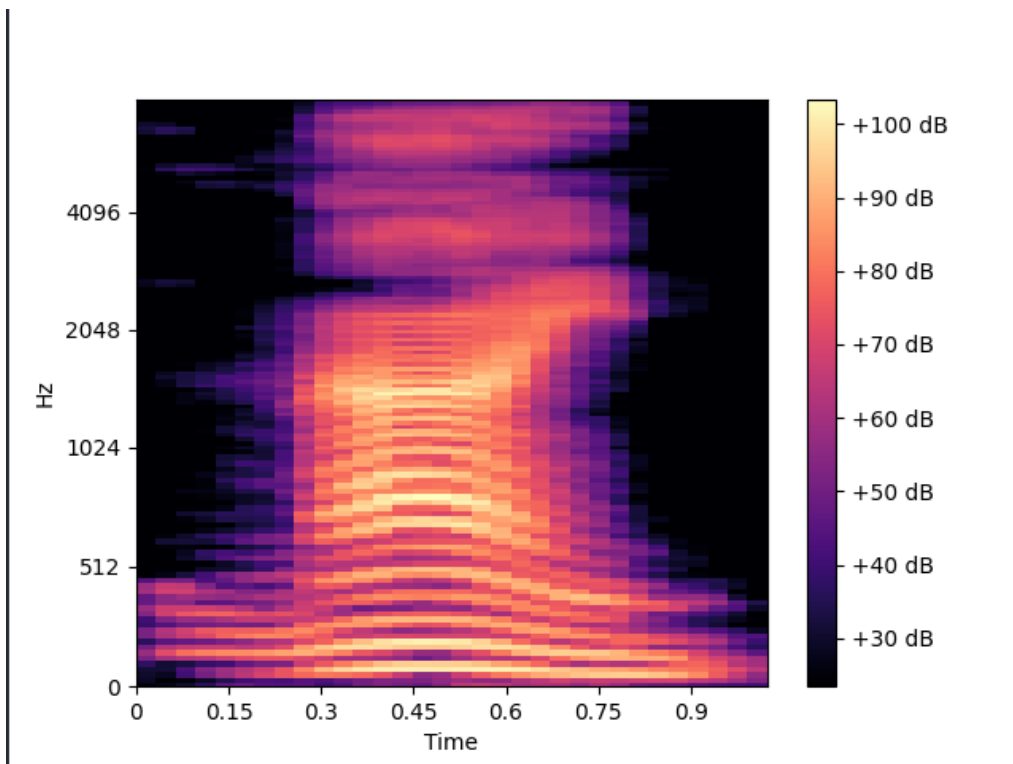
```
librosa.amplitude_to_db(S=Mel-Spectrogram, ref=np.min)
```

όπου, το S είναι το εύρος εισόδου και το ref , εάν είναι κλιμακωτό, το πλάτος $|S|$ κλιμακώνεται σε σχέση με το $ref = 20 \cdot \log_{10} \left(\frac{S}{ref} \right)$. Τα μηδενικά στην έξοδο αντιστοιχούν σε θέσεις όπου $S = ref$. Εάν καλείται, η τιμή αναφοράς υπολογίζεται ως $ref(S)$.

Για την εκπαίδευση των δεδομένων φορτώνουμε τα αρχεία ήχου μας μέσω της μεθόδου `get_training_samples_signal()`. Αυτό που γίνεται στην πραγματικότητα είναι ότι φορτώσαμε το κάθε σήμα κάθε ηχογράφησης μέσα στο `training_samples_signals`. Την μεταβλητή που επιστρέφουμε από την μέθοδο την χρησιμοποιούμε σαν όρισμα στην μέθοδο `recognition()` για να προχωρήσουμε στην αναγνώριση των ψηφίων.

Η μέθοδος `filtered_dataset_signal()` υλοποιήθηκε για να αφαιρούμε τον θόρυβο του `background` από το σήμα και τα σιωπηλά κομμάτια που είναι χαμηλότερα από 40 ντεσιμπέλς, στην ουσία φιλτράρουμε το σήμα μας. Για να αφαιρέσουμε τον θόρυβο στο `background` καλέσαμε την μέθοδο `remove_noise()` όπου κατασκευάσαμε για αυτόν το σκοπό. Η μέθοδος `remove_noise` δέχεται σαν όρισμα τι σήμα μας και μέσω την βιβλιοθήκης `noisereducer` αφαιρέσαμε τον θόρυβο στο `background`.

ΑΠΟΤΕΛΕΣΜΑ



2.3. ΥΠΟΕΡΩΤΗΜΑ ΤΡΙΤΟ

ΕΚΦΩΝΗΣΗ

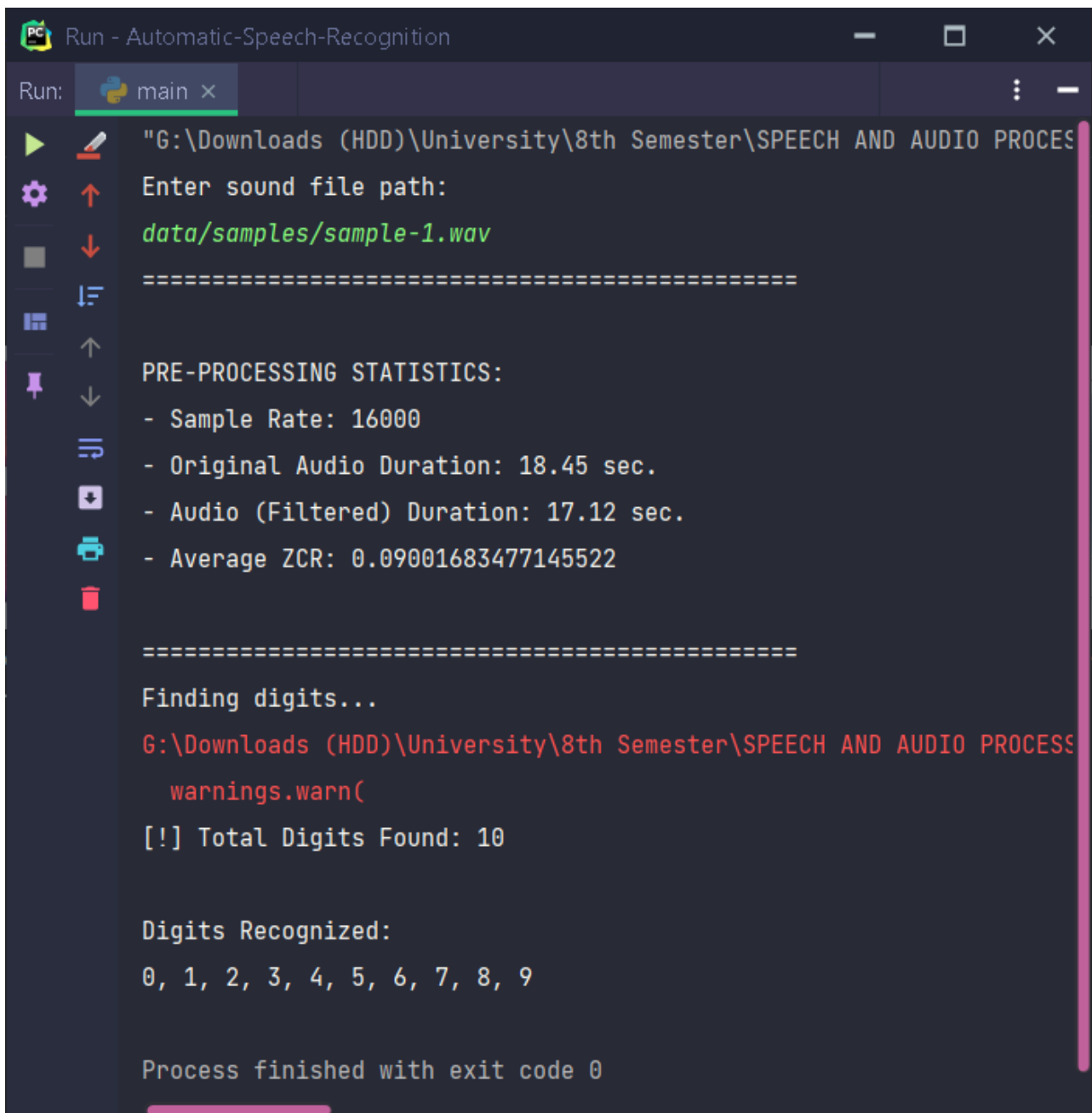
Στην έξοδο παράγεται κείμενο με τα ψηφία που αναγνωρίστηκαν.

ΥΛΟΠΟΙΗΣΗ

Τέλος, για την υλοποίηση του συγκεκριμένου υποερωτήματος, όταν ολοκληρωθούν οι υπόλοιπες διαδικασίες και γίνει πρόβλεψη των ψηφίων τότε με την πιο κάτω συνάρτηση θα επιστραφούν τα ψηφία σε ένα array το οποίο τυπώνουμε μετά διαχωρίζοντας τις τιμές του πίνακα με ένα κόμμα.

`recognition`(πίνακας με των αριθμό το συνολικό αρ. των ψηφίων,
σήμα μετά την προ-διεργασία,
σύνολο δεδομένων εκπαίδευσης)

ΑΠΟΤΕΛΕΣΜΑ



```
Run - Automatic-Speech-Recognition
Run: main x
"G:\Downloads (HDD)\University\8th Semester\SPEECH AND AUDIO PROCES
Enter sound file path:
data/samples/sample-1.wav
=====
PRE-PROCESSING STATISTICS:
- Sample Rate: 16000
- Original Audio Duration: 18.45 sec.
- Audio (Filtered) Duration: 17.12 sec.
- Average ZCR: 0.09001683477145522
=====
Finding digits...
G:\Downloads (HDD)\University\8th Semester\SPEECH AND AUDIO PROCES
warnings.warn(
[!] Total Digits Found: 10
Digits Recognized:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Process finished with exit code 0
```

Εικόνα 2.4

3. ΘΕΜΑ ΔΕΥΤΕΡΟ

Για την υλοποίηση του δεύτερου θέματος «Open source audio annotation study», έγινε χρήση του προγράμματος **Ocenaudio** για να συμπεράνουμε τα συμβάντα. Τα αποτελέσματα της ανάλυσης βρίσκονται στο αρχείο **thema-2.csv**.

3.1. ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΠΙΣΜΟΥ ΣΥΜΒΑΝΤΩΝ ΗΧΟΥ

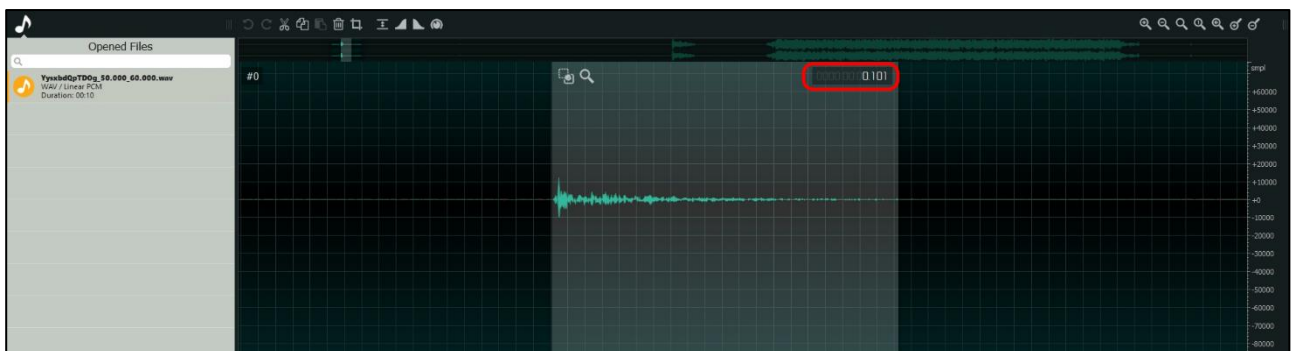
Θα χρησιμοποιήσουμε για παράδειγμα το πιο κάτω αρχείο ήχου: **YysxbdQpTDOg_50.000_60.000.wav**

- Αυτούσιο το αρχείο έχει την μορφή:



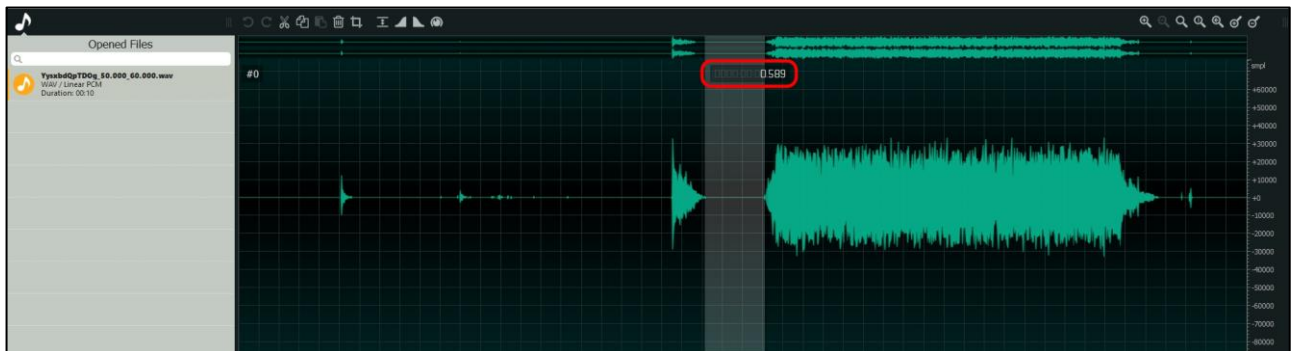
- Έλεγχος συμβάντος αν είναι μεγαλύτερο από **250 ms**:

Αν επιλέξουμε το πρώτο συμβάν που βλέπουμε στη προηγούμενη φωτογραφία μέσω του προγράμματος βλέπουμε ότι η διάρκεια του συμβάντος δεν ξεπερνά τα **250 ms** και το αγνοούμε επειδή δεν θεωρείται άξιο λόγου.



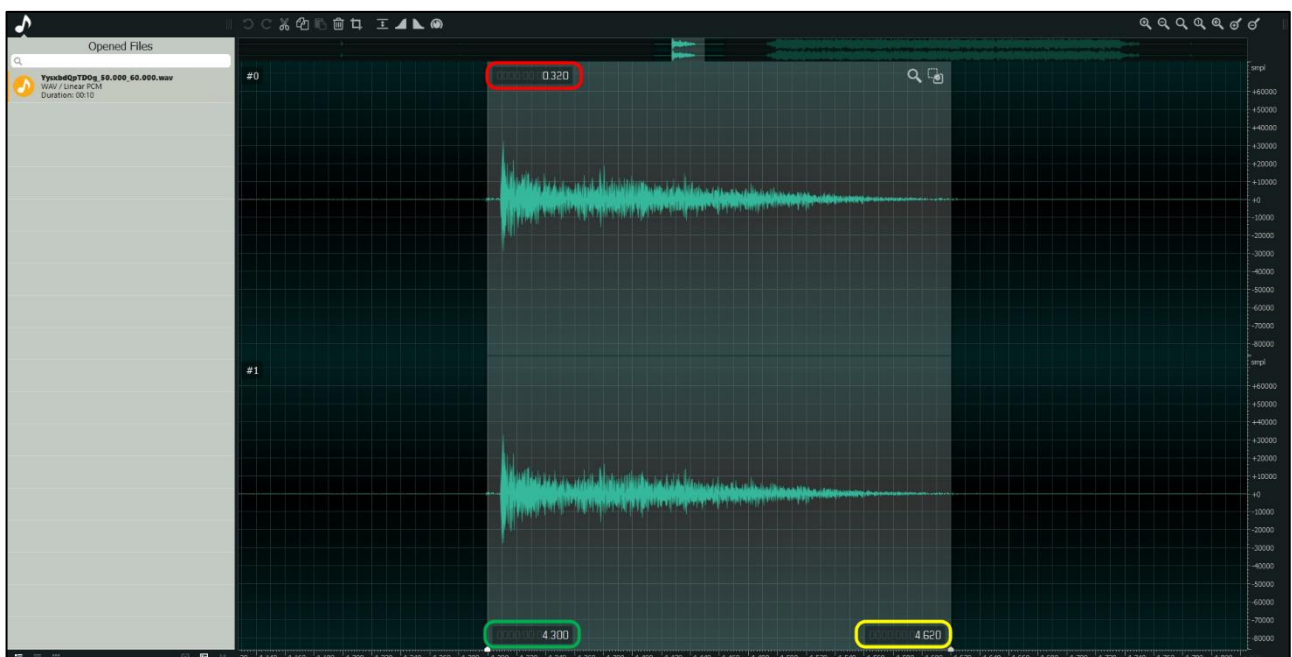
- Έλεγχος αν δύο συμβάντα του ίδιου είδους θεωρούνται ξεχωριστά:

Το συγκεκριμένο αρχείο ήχου κατατάσσεται στο είδος «blender» (δεν υπάρχει ούτε επικάλυψη ήχου αλλά ούτε άλλο είδος ήχου). Όπως βλέπουμε απέχουν μεταξύ τους με ένα «κενό» μεγαλύτερο από **150 ms**, άρα θεωρούνται δύο διαφορετικά συμβάντα.

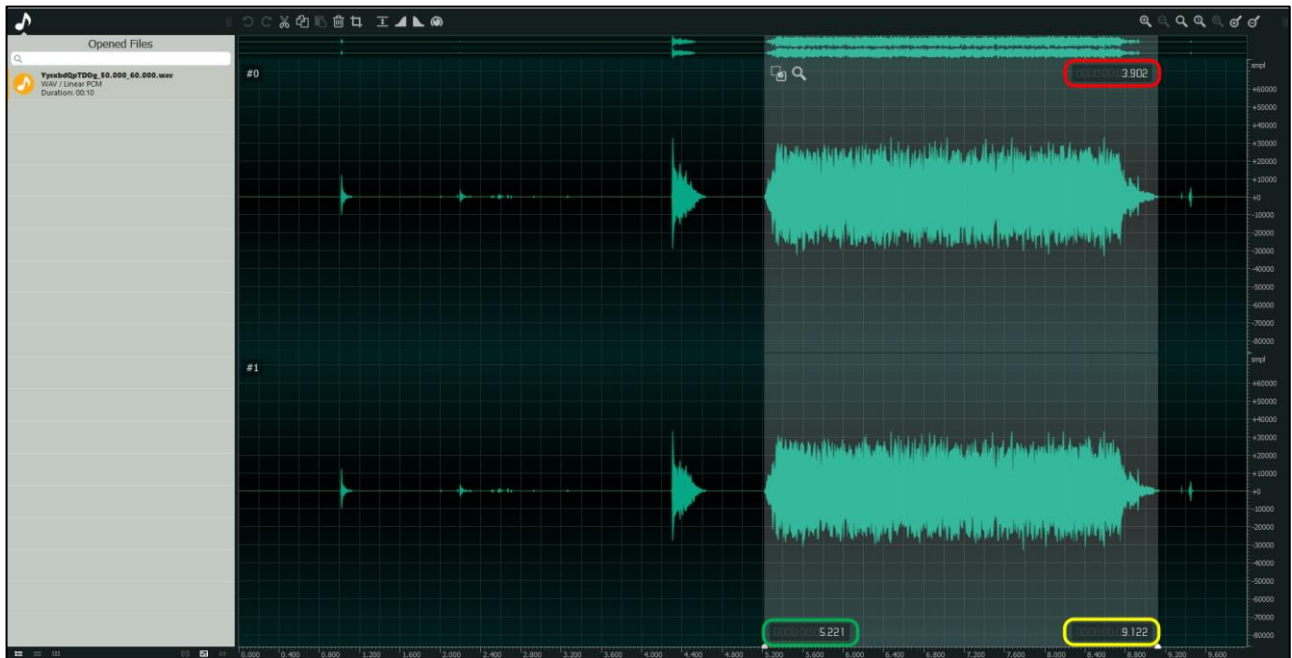


- Η αρχή και το τέλος των συμβάντων:

Αρχικά ελέγχουμε ότι η διάρκεια του συμβάν μας είναι μεγαλύτερη από **150 ms** (βλέπε κόκκινο κουτάκι). Εφόσον είναι βλέπουμε ότι η αρχή του συμβάντος μας (βλέπε πράσινο κουτάκι) είναι στα **4,300** δευτερόλεπτα και το τέλος του συμβάν είναι στα **4,620** δευτερόλεπτα.



Στο δεύτερο συμβάν βλέπου ξεκάθαρα ότι διαρκεί περισσότερο από **150 ms** και συνεχίζουμε την διαδικασία όπως προηγουμένως.



- Τέλος σημειώνουμε τα αποτελέσματα μας στο csv αρχείο:

Σημειώνουμε στην πρώτη στήλη (**A**) το όνομα του αρχείου, στην δεύτερη στήλη (**B**) την αρχή του συμβάν και στην τρίτη στήλη (**C**) το τέλος του συμβάν. Στην τέταρτη (**D**) και τελευταία στήλη σημειώνουμε το είδος του συμβάν.

A	B	C	D
YysxbdQpTDOg_50.000_60.000.wav	5,221	9,122	blender
YysxbdQpTDOg_50.000_60.000.wav	4,300	4,620	blender

Αυτή ήταν η διαδικασία που ακολουθήσαμε και για τα 100 αρχεία ήχου.

4. ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΠΗΓΕΣ

1. Ψηφιακή Επεξεργασία Φωνής: Θεωρία και Εφαρμογές: Rabiner L. (Θεωρία)
2. Automatic Speech Recognition – A Deep Learning Approach: Dong Yu, Li Deng
3. Σημειώσεις Εργαστηρίου.
4. <https://realpython.com/python-speech-recognition/>
5. <https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html>
6. <http://www.iitg.ac.in/samudravijaya/tutorials/asrTutorial.pdf>
7. <https://towardsdatascience.com/how-i-understood-what-features-to-consider-while-training-audio-files-eedfb6e9002b>
8. <https://www.sciencedirect.com/topics/engineering/short-time-energy>
9. <https://vlab.amrita.edu/?sub=3&brch=164&sim=857&cnt=1>
10. <https://anale-informatica.tibiscus.ro/download/lucrari/15-1-23-Ibrahim.pdf>
11. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.294.8172&rep=rep1&type=pdf>
12. <https://librosa.org/doc/main/generated/librosa.effects.preemphasis.html>
13. https://opencourses.uoc.gr/courses/pluginfile.php/15269/mod_resource/content/1/set1.pdf
14. http://cvsp.cs.ntua.gr/~nassos/resources/speech_course_2004/OnlineSpeechDemos/speechDemo_2004_Part1.html#2
15. <http://wantee.github.io/2015/03/14/feature-extraction-for-asr-preprocessing/>
16. <https://opencourses.uoa.gr/modules/document/file.php/DI36/Διδακτικό%20πακέτο/Παρουσιάσεις/PDF/speech%20processing%20and%20NLP-5.pdf>
17. <https://medium.com/@anonymomous.ut.grad.student/building-an-audio-classifier-f7c4603aa989>
18. [https://towardsdatascience.com/fast-fourier-transform-937926e591cb?gi=5d672a6fee18#:~:text=As%20the%20name%20implies%2C%20the%20to%20O\(NlogN\)%20](https://towardsdatascience.com/fast-fourier-transform-937926e591cb?gi=5d672a6fee18#:~:text=As%20the%20name%20implies%2C%20the%20to%20O(NlogN)%20)
19. <https://stackoverflow.com/questions/6771428/most-efficient-way-to-reverse-a-numpy-array>
20. <https://superkogito.github.io/blog/SignalFraming.html>
21. <https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft>

22. <https://machinelearningmastery.com/k-fold-cross-validation/>
23. <https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>
24. <https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>

5. ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ

Όνομα	Έκδοση	Τύπος
Ocenaudio	3.10.9	Εργαλείο
Python	3.9.6	Εργαλείο
PyCharm Professional	2021.1.3	Εργαλείο
librosa	0.8.1	Βιβλιοθήκη
numpy	1.21.0	Βιβλιοθήκη
scikit-learn	0.24.2	Βιβλιοθήκη
scipy	1.7.0	Βιβλιοθήκη
noisereducer	1.1.0	Βιβλιοθήκη
ffmpeg	1.4	Βιβλιοθήκη
matplotlib	3.4.2	Βιβλιοθήκη
termcolor	1.1.0	Βιβλιοθήκη

Συμπληρωματικά, το αρχείο `requirements.txt` που βρίσκεται στο directory μας, δημιουργείται αυτόματα από το PyCharm IDE με όλα τα εργαλεία που χρησιμοποιούμε συμπεριλαμβάνοντας τις εκδόσεις αυτών.

6. ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

Συντομογραφία	Λέξη
ASR	Automatic Speech Recognition
dB	Decibel
MFCC	Mel Frequency Cepstral Coefficient
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
ZCR	Zero-Crossing Rate
STE	Short-Time Energy
WAV	Wave Audio File
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
Δευτ.	Δευτερόλεπτο(α)
Κλπ.	Και τα λοιπά