

Συστήματα Πολυμέσων Υλοποίηση GSM 06.10

Χρήστος Μάριος Περδίκης 10075 cperdikis@ece.auth.gr
Γιώργος RootCrittter ????? ?????????@ece.auth.gr

Το παρόν έγγραφο είναι η αναφορά της εργασίας του μαθήματος Συστήματα Πολυμέσων τη χρονιά 2024-2025. Κληθήκαμε να υλοποιήσουμε κωδικοποίηση φωνής σύμφωνα με το πρότυπο GSM 06.10. Από τα τρία επίπεδα υλοποιήσαμε μέχρι το 1ο.

1 Short Term Analysis

1.1 Audio Wrapper και main.py

Αρχικά δημιουργήσαμε ένα πολύ απλό audio wrapper πρόγραμμα ώστε να μπορούμε να μετατρέπουμε τα ηχητικά δεδομένα ενός wav αρχείου σε μορφή την οποία μπορεί να επεξεργαστεί από κώδικα, και μετά πάλι πίσω σε wav μορφή. Ο κώδικας του προγράμματος αυτού βρίσκεται στο αρχείο *audio_wrapper.py*.

Υπάρχουν δύο συναρτήσεις για την ανάγνωση wav αρχείων. Αρχικά κατασκευάσαμε τη συνάρτηση *read_data*, η οποία έκανε χρήση του wave module και διαβάζαμε 160 frames του αρχείου ήχου τη φορά. Παρουσιάστηκε πρόβλημα όμως όταν θέλαμε να μετατρέψουμε το byte buffer που επιστρέφει η συνάρτηση *readframes* σε numpy array. Συγκεκριμένα, έπρεπε να γνωρίζουμε πόσα bit υπήρχαν σε κάθε sample, και δεν γνωρίζαμε αν έπρεπε να λάβουμε υπόψιν το endianness των bytes και το αν ήταν signed ή unsigned ακέραιοι. Για αυτό χρειάστηκε να κατασκευάσουμε τη συνάρτηση *scipy_read_data*, η οποία έχει υλοποίηση με το module *scipy*. Η συνάρτηση *io.wavfile.read* οποία επιστρέφει απευθείας ένα έτοιμο numpy array με τα samples ολόκληρου του αρχείου, λύνοντας τα προβλήματά μας. Κρατάμε και τις δύο συναρτήσεις για ιστορικούς λόγους, θεωρητικά δουλεύει και η *read_data* για 8-bits- και 16-bits-per-sample wav αρχεία, αλλά δοκιμάστηκε μόνο για να διαβάσει τα πρώτα 160 samples ενός wav αρχείου, όχι ολόκληρα αρχεία.

(να πω και για main.py εδώ; Καλό σημείο θα ήταν)

(audio wrapper write function)

1.2 Κωδικοποιητής

1.3 Αποκωδικοποιητής