|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Συστήματα Πολυμέσων  [Εμβόλιμη εξεταστική - 2024] |
| |  |  |  | | --- | --- | --- | | ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΠΟΥΛΟΣ | ΕΞΑΜΗΝΟ 9ο | Π20206 | |

**ΠΕΡΙΕΧΟΜΕΝΑ**

[preparation](#prepetation)

[ΘΕΜΑ i](#i)

[ΘΕΜΑ ii](#ii)

ΘΕΜΑ iii

ΘΕΜΑ iv

[Για το auxiliary](#giatoauxiliary)

**Preparation:**

Εγκατέστησα την **PyCharm**.

Πρόσθεσα τις βιβλιοθήκες **numPy** & **opencv-python**

Βρήκα ένα βίντεο με κατάληξη **.avi** με διάρκεια **10 seconds**

**i)**

* Σε πρώτη φάση, δημιουργούμε το αρχείο **coder.py** και στον ίδιο φάκελο βάζουμε και το **video** μας.
* ανοίγουμε το βίντεο και αποθηκεύουμε όλα τα καρέ στον πίνακα frames:

A computer screen shot of a computer code

Description automatically generated

**Ενδεικτικά(**το **output** υπάρχει στο αρχείο **output1.txt):**

Εάν εκτυπώσουμε την μεταβλητή **frame\_count** θα έχουμε **output**: Σύνολο καρέ: 300 (Η διάρκεια του video είναι 10 seconds 🡪 30 frames per second)

Εάν εκτυπώσουμε τον πίνακα **frames** μπορούμε να δούμε πληροφορίες όπως διαστάσεις και ένταση φωτεινότητας.

* Σε δεύτερη φάση, θα υπολογίσουμε την **εικόνα σφάλματος** για κάθε **P-Frame** ως εξής: εικόνα σφάλματος = P-frame – προηγούμενο frame.Ο τύπος αυτο ισχύει επειδη η εκφώνηση αναφέρει οτι δεν πρέπει να χρησιμοποιηθεί **αντιστάθμιση κίνησης**.
* Να αναφέρουμε εν τάχει πώς τα **I–frames** κωδικοποιούνται ανεξάρτητα, ως πλήρεις εικόνες, ενώ τα **P-frames** κωδικοποιούνται με βάση την πρόβλεψη από προηγούμενα frames. Αποθηκεύονται μόνο οι διαφορές σε σχέση με το προηγούμενο frame.
* Το **GOP** είναι **12**, αφού η εκφώνηση λέει *υποθέστε ότι ανά 12 πλαίσια το πρώτο είναι πάντα τύπου I και τα υπόλοιπα τύπου P.*
* Χρησιμοποιείται η μέθοδος **clip()** (από την **numpy**) για την αφαίρεση των δύο καρέ(line29). Συγκεκριμένα χρησιμοποιούμε την **astype()** για μετατροπή σε ακεραίους, για να αποφύγουμε την υπερχείλιση κατά την **αφαίρεση**.

A computer screen shot of text

Description automatically generated

* Στο αρχείο **auxiliary** μπορούμε να δούμε όλα **error frames** (από τα **P-frames** εννοείται).
* Σε τρίτη φάση, θα φτιάξουμε και την λειτουργία της **αποκωδικοποίησης**.
* Τα **I-frames** αποθηκεύονται όπως είναι, ενώ τα **P-frames** ανακατασκευάζονται από το προηγούμενο αποκωδικοποιημένο καρέ και την αντίστοιχη **εικόνα σφάλματος**.
* Η **ανακατασκευή** γίνεται με τη χρήση της συνάρτησης **clip()** από τη βιβλιοθήκη **NumPy**, η οποία **προσθέτει** τα δεδομένα του προηγούμενου καρέ με την **εικόνα σφάλματος** και εξασφαλίζει ότι οι τιμές παραμένουν εντός του εύρους [0, 255].
* Χρησιμοποιήσαμε τη συνάρτηση **clip()** επειδή αυτή **εξασφαλίζει** ότι οι υπολογισμένες τιμές παραμένουν εντός του επιτρεπτού **εύρους [0, 255].** Χωρίς αυτή τη συνάρτηση, οι τιμές των pixel ενδέχεται να γίνουν αρνητικές ή μεγαλύτερες από 255, κάτι που δεν είναι αποδεκτό σε εικόνες. A computer screen shot of a black screen

  Description automatically generated
* Αναλυτικά παραδείγματα εκτέλεσης στο **auxiliary2024.zip**

**ii)**

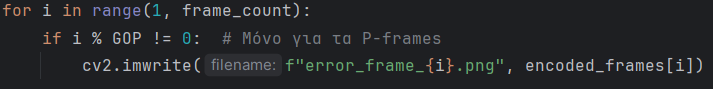
**Για το auxiliary2024.zip[**Η μέθοδος **imwrite()** ανήκει στην κατηγορία αποθήκευσης αρχείων, αποθηκεύει frame σε μορφή αρχείου (π.χ., PNG, JPG)**]:**

* **Ερώτημα i**

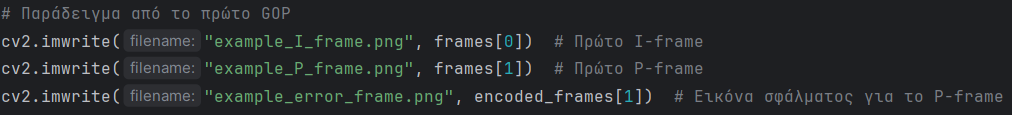
File : “ 300 original frames” [Ολα τα frames(θα χρησιμοποιηθούν στο ii ερώτημα)]



File : “ Error Frames(Only P-frames)” [Όλες οι **εικόνες σφάλματος**]



File : “Example - first GOP”[ Ένα παράδειγμα **I-frame** και το αντίστοιχο **P-frame**, καθώς και την εικόνα σφάλματος του **P-frame**]



* **Ερώτημα ii**

**ii)**