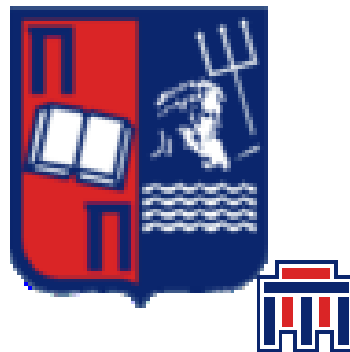


ΕΡΓΑΣΙΑ ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020



ΑΠΟΣΤΟΛΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ(Π17007)-
ΡΟΥΜΕΛΙΩΤΗΣ ΠΑΥΛΟΣ(Π17122)

Περιεχόμενα

A' Μέρος.....	3
Άσκηση 8.17	3
1.Εκφώνηση Άσκησης	3
2.Σύντομη παρουσίαση της κεντρικής ιδέας	5
3.Υλοποίηση	7
3.1 Ερώτημα Α'	7
3.1.1 Ανάγνωση Βίντεο.....	7
3.1.2 Κωδικοποίηση Βίντεο.....	7
3.1.3 Αρχείο για την εγγραφή του βίντεο	7
3.1.4 Διαχείριση πλαισίου I.....	8
3.1.5 Επαναληπτική διαδικασία για τα πλαίσια P	8
3.1.6 Προβολή πλαισίων σφαλμάτων	9
3.1.7 Πληροφοριακό περιεχόμενο πλαισίων σφαλμάτων	9
3.2 Ερώτημα Β'	10
3.2.1 Ανάγνωση βίντεο.....	10
3.2.2 Κωδικοποίηση Βίντεο.....	10
3.2.3. Αρχείο για την εγγραφή του βίντεο	10
3.2.4 Διαχείριση πλαισίου I.....	10
3.2.5 Επαναληπτική διαδικασία για τα πλαίσια P	11
3.2.6 Συνάρτηση συμπλήρωσης πλαισίων με μαύρα εικονοστοιχεία.....	11
3.2.7 Διάρθρωση P πλαισίων σε macroblock 16X16	11
3.2.8 Υπολογισμός διανυσμάτων κίνησης και λογαριθμική αναζήτηση του καλύτερου διανύσματος.....	12
3.2.9 Υπολογισμός μετρικής SAD	13
3.2.10 Υπολογισμός μπλοκ σφαλμάτων	13
3.2.11 Ανακατασκευή πλαισίων.....	13
3.2.12 Προβολή πλαισίων σφαλμάτων.....	13
3.2.13 Πληροφοριακό περιεχόμενο πλαισίων σφαλμάτων	14
4.Αποτελέσματα	15
4.1 Ερώτημα Α'	15
4.2 Ερώτημα Β'	16
5.Συμπεράσματα	18
6.Απαιτήσεις συστήματος και εκτέλεση	19

Άσκηση 8.18	25
1.Εκφώνηση Άσκησης	25
2.Σύντομη παρουσίαση της κεντρικής ιδέας	27
3.Υλοποίηση	28
3.1.1 Ανάγνωση Βίντεο.....	28
3.1.2 Κωδικοποίηση Βίντεο.....	28
3.1.3 Αρχείο για την εγγραφή του βίντεο	28
3.1.4 Διαχείριση πρώτου πλαισίου-παρασκηνίου.....	29
3.1.5 Επαναληπτική διαδικασία για τα πλαίσια P	29
3.1.6 Συνάρτηση συμπλήρωσης πλαισίων με μαύρα εικονοστοιχεία.....	29
3.1.7 Διαίρεση P πλαισίων σε macroblock	30
3.1.8 Υπολογισμός διανυσμάτων κίνησης και λογαριθμική αναζήτηση του καλύτερου διανύσματος.....	30
3.1.9 Υπολογισμός μετρικής SAD	31
3.1.10 Εύρεση κίνησης και αντικατάσταση μπλοκ με το αντίστοιχο μπλοκ παρασκηνίου	31
3.1.11 Ανακατασκευή πλαισίων.....	32
3.1.12 Προβολή πλαισίων σφαλμάτων.....	32
4. Αποτελέσματα	33
5.Συμπεράσματα	35
6.Απαιτήσεις συστήματος και εκτέλεση	36
B' Μέρος.....	40
1.Εκφώνηση Άσκησης	40
2.Σύντομη παρουσίαση της κεντρικής ιδέας	41
3.Υλοποίηση	42
3.1.1 Είσοδος πενταψήφιων αριθμών.....	42
3.1.2 Αντικατάσταση τυχαίου ψηφίου με το 5	42
3.1.3 Κατασκευή εικόνας A	42
3.1.4 Κατασκευή εικόνας B	43
3.1.5 Zigzag Scan.....	44
3.1.6 Κωδικοποίηση Huffman	44
3.1.7 Υπολογισμός μέσου λόγου συμπίεσης των εικόνων A και B.....	45
4. Αποτελέσματα	46
5.Συμπεράσματα	47
6.Απαιτήσεις συστήματος και εκτέλεση	48
Βιβλιογραφία	52

Α' Μέρος

Άσκηση 8.17

1.Εκφώνηση Άσκησης

Στην άσκηση αυτή θα υλοποιήσετε την τεχνική της αντιστάθμισης κίνησης και θα μελετήσετε πώς επηρεάζει τα σφάλματα πρόβλεψης. Δείγματα αρχείων βίντεο, μαζί με τη σχετική πληροφορία διαμόρφωσης, παρέχονται στην ενότητα αρχείων προς λήψη του ιστοτόπου www.cengage.com. Εκεί θα βρείτε και τον κώδικα για να διαβάσει και να προβάλλει πλαίσια βίντεο. Υποθέστε ότι το πρώτο πλαίσιο θα είναι πάντα ένα πλαίσιο I και ότι τα υπόλοιπα πλαίσια θα είναι τύπου P. Η υπόθεση αυτή ευσταθεί στην περίπτωση των σύντομων ακολουθιών βίντεο που επεξεργάζεστε, που έχουν μήκος το πολύ 100 πλαισίων.

- a. Στο πρώτο μέρος της άσκησης αυτής, υποθέστε ότι θέλετε να προβλέπετε ολόκληρα P πλαίσια και όχι κατά τμήματα. Η πρόβλεψη κάθε ολόκληρου πλαισίου γίνεται με βάση το προηγούμενο πλαίσιο. Υλοποιήστε μία διαδικασία που να δέχεται είσοδο δύο πλαίσια, υπολογίζει τη διαφορά τους και επιστρέφει ένα πλαίσιο σφαλμάτων. Δεν υπολογίζετε διάνυσμα κίνησης. Να προβάλετε τα πλαίσια σφαλμάτων. Σημειώστε ότι το πληροφοριακό περιεχόμενο του πλαισίου σφαλμάτων θα πρέπει να είναι μικρότερο, συγκρινόμενο με αυτό των πλαισίων.
- b. Στο δεύτερο βήμα θα υλοποιήσετε τεχνική πρόβλεψης κίνησης, η οποία θα υπολογίζει διανύσματα κίνησης ανά μπλοκ. Κάθε μπλοκ θα έχει το τυπικό MPEG μέγεθος 16×16 . Υλοποιήστε μια συνάρτηση που θα δέχεται είσοδο δύο πλαίσια: ένα πλαίσιο αναφοράς, το οποίο θα χρησιμοποιηθεί κατά την αναζήτηση των διανυσμάτων κίνησης, και ένα πλαίσιο στόχος, το οποίο θα προβλεφθεί. Διαιρέστε το πλαίσιο-στόχο σε μακρομπλόκ μεγέθους 16×16 . Εάν το πλάτος και ύψος του πλαισίου δεν είναι πολλαπλάσιο του 16, συμπληρώστε κατάλληλα το πλαίσιο με μαύρα εικονοστοιχεία. Για κάθε μπλοκ στο πλαίσιο-στόχο, ανατρέξτε στην αντίστοιχη θέση στο πλαίσιο αναφοράς και βρείτε την περιοχή που δίνει το καλύτερο ταίριασμα, όπως έχει εξηγηθεί στο

κείμενο του κεφαλαίου. Χρησιμοποιήστε τη μετρική SAD σε περιοχή αναζήτησης που προκύπτει για $k = 16$, έτσι ώστε τα διανύσματα κίνησης να έχουν μέγεθος το πολύ 16 εικονοστοιχείων ως προς κάθε κατεύθυνση. Με βάση το μπλοκ πρόβλεψης, υπολογίστε το μπλοκ σφαλμάτων ως τη διαφορά μεταξύ του αρχικού μπλοκ και του προβλεφθέντος. Αφού αυτή η διαδικασία ολοκληρωθεί για όλα τα μπλοκ, θα προκύψει ένα πλαίσιο σφαλμάτων. Να γίνει η πρόβλεψη όλων των πλαισίων σφαλμάτων. Θα διαπιστώσετε ότι τα πλαίσια σφαλμάτων εμφανίζουν σημαντικά μικρότερη εντροπία σε σύγκριση με τη προηγούμενη περίπτωση, μολονότι απαιτείται περισσότερος χρόνος για τον υπολογισμό τους.

2.Σύντομη παρουσίαση της κεντρικής ιδέας

Ζητούμενο στο **πρώτο ερώτημα** είναι η υλοποίηση μιας διαδικασίας, η οποία να δέχεται σαν είσοδο δύο πλαίσια και υπολογίζει τη διαφορά τους, επιστρέφοντας το πλαίσιο σφαλμάτων. Για την προβολή των πλαισίων σφαλμάτων επιλέχθηκε η δημιουργία ενός βίντεο. Η κεντρική ιδέα για την επίτευξη του στόχου, ήταν για κάθε δύο frames(πλαίσια) που διαβάζονται, να αποθηκεύονται σε πίνακες οι τιμές των pixels (εικονοστοιχείων) και στη συνέχεια αφαιρούμε αυτούς τους δύο πίνακες, ώστε να υπολογίσουμε τα πλαίσια σφαλμάτων, τα οποία και τα γράφουμε σε ένα καινούργιο βίντεο. Σαν υποθέσεις εργασίας κάναμε τις εξής:

1. Επιλέχθηκε ένα βίντεο 3 δευτερολέπτων, στο οποίο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου, δηλαδή η πτώση του μπουκαλιού.
2. Το βίντεο, κάθε frame αυτού, μετατράπηκε από έγχρωμο, σε βίντεο με αποχρώσεις του γκρι(Grayscale).

Στο **δεύτερο ερώτημα** ζητούμενο είναι η υλοποίηση μιας διαδικασίας, η οποία να δέχεται σαν είσοδο δύο πλαίσια, τα οποία αν δεν έχουν πλάτος και ύψος πολλαπλάσιο του 16 θα συμπληρώνονται με μαύρα εικονοστοιχεία. Στη συνέχεια το πλαίσιο-στόχος θα διαιρείται σε μακρομπλόκ μεγέθους 16X16 και με τον υπολογισμό των διανυσμάτων κίνησης και της μετρικής συνάρτησης SAD,θα υπολογίζεται το μπλοκ σφαλμάτων αλλά και η προβολή των πλαισίων σφαλμάτων. Για την προβολή των πλαισίων σφαλμάτων, και σε αυτό το ερώτημα, επιλέχθηκε η δημιουργία ενός βίντεο. Η κεντρική ιδέα για την επίλυση του προβλήματος ήταν για κάθε δύο frames που διαβάζονται να μετατρέπονται σε πλαίσια διαστάσεων 16X16 ,εφόσον δεν είναι, και στη συνέχεια αφού έχει γίνει η διαίρεση του πλαισίου- στόχου σε μακρομπλόκ μεγέθους 16X16 και στη συνέχεια να υπολογίσουμε τα διανύσματα κίνησης για κάθε ένα από τα μπλοκ. Επόμενο, βήμα στη διαδικασία υλοποίησης της διαδικασίας είναι η χρήση της μετρικής συνάρτησης SAD,για τον υπολογισμό της απόλυτης διαφοράς μεταξύ της υποψήφιας περιοχής και του τμήματος στόχου. Η αναζήτηση και η εύρεση του διανύσματος κίνησης η οποία ελαχιστοποιεί την ποσότητα SAD, θα γίνει με την βοήθεια της λογαριθμικής αναζήτησης. Ενώ για την προβολή των πλαισίων θα πρέπει αρχικά να

ανακατασκευάσουμε τα πλαίσια επαναφέροντάς αυτά στις αρχική τους διάσταση. Σαν υποθέσεις εργασίας κάναμε τις παρακάτω:

1. Επιλέχθηκε ένα βίντεο 3 δευτερολέπτων, στο οποίο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου, δηλαδή η πτώση του μπουκαλιού.
2. Το βίντεο, κάθε frame αυτού, μετατράπηκε από έγχρωμο, σε βίντεο με αποχρώσεις του γκρι(Grayscale).

3.Υλοποίηση

Για την υλοποίηση και την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού *Python 3*, οι απαιτήσεις για την εκτέλεση παρατίθενται στην ενότητα 6. Τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος θα αναλυθούν διεξοδικά παρακάτω.

3.1 Ερώτημα Α΄

3.1.1 Ανάγνωση Βίντεο

Το αρχικό βήμα στην υλοποίηση του ερωτήματος είναι μέσω της βιβλιοθήκης **opencv-python**, και πιο συγκεκριμένα με την μέθοδο **cv2.VideoCapture**, να μπορέσουμε να αναγνώσουμε το video με το οποίο θα εργαστούμε στη συνέχεια.

3.1.2 Κωδικοποίηση Βίντεο

Επόμενο βήμα στην υλοποίηση του συστήματος είναι η κωδικοποίηση του video, για το οποίο θα επιστρέψουμε τα πλαίσια σφαλμάτων. Πιο συγκεκριμένα, χρησιμοποιώντας την μέθοδο **cv2.VideoWriter_fourcc()**, και την κωδικοποίηση **'XVID'**.

3.1.3 Αρχείο για την εγγραφή του βίντεο

Στη συνέχεια, καθορίζονται και δίνονται οι παράμετροι για την εγγραφή του βίντεο σε κάποιο αρχείο. Απαραίτητη μέθοδος είναι η **cv2.VideoWriter**, η οποία παίρνει τις απαραίτητες παραμέτρους για:

- ❖ Το όνομα του αρχείου του βίντεο
- ❖ Την κωδικοποίηση (βλ. Ενότητα 3.1.2)

- ❖ Τα frames per second(fps)
- ❖ Το ύψος και το πλάτος του κάθε πλαισίου(size of frame)

3.1.4 Διαχείριση πλαισίου I

Λαμβάνοντας υπ' όψιν, ότι το πρώτο πλαίσιο θα είναι πάντα ένα **πλαίσιο I**, αφού αποθηκευτεί μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**.

3.1.5 Επαναληπτική διαδικασία για τα πλαίσια P

Αρχικά, βασική συνθήκη για τον υπολογισμό της διαφοράς δύο πλαισίων και η προβολή των πλαισίων σφαλμάτων είναι να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα. Εφόσον, η συνθήκη είναι αληθής το τρέχων πλαίσιο είναι το επόμενο πλαίσιο P. Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**. Η διαφορά δύο πλαισίων P, η οποία ορίζεται ως την διαφορά μεταξύ του τωρινού πλαισίου και του προηγούμενου πλαισίου, υπολογίζεται με την συνάρτηση **np.subtract**. Για την αναδρομικότητα της διαδικασίας, αποθηκεύουμε το τρέχων πλαίσιο ως προηγούμενο. Το κάθε πλαίσιο διαφορών αποθηκεύεται σε έναν πίνακα, τον οποίο θα τον αξιοποιήσουμε για τις παρατηρήσεις σχετικά με το πληροφοριακό περιεχόμενο του πλαισίου σφαλμάτων. Τέλος, στην περίπτωση που δεν υπάρχει επόμενο πλαίσιο, η επαναληπτική διαδικασία τερματίζει.

3.1.6 Προβολή πλαισίων σφαλμάτων

Όπως αναφέρθηκε στην Ενότητα 2, για την προβολή των πλαισίων σφαλμάτων επιλέχθηκε η δημιουργία ενός βίντεο. Για κάθε δύο πλαίσια, για τα οποία υπολογίζεται η διαφορά τους και επιστέφει το πλαίσιο σφαλμάτων, όπως αναλύθηκε και στην ενότητα 3.1.5, το οποίο γράφεται στο βίντεο και μετά το πέρας της επαναληπτικής διαδικασίας αποθηκεύεται στον φάκελο με το αρχείο του κώδικα ή δύναται να προβληθεί, εάν αυτό επιλεγεί από τον χρήστη. Για την αποθήκευση του βίντεο χρησιμοποιήθηκε η μέθοδος **.write()**, ενώ για την αναπαραγωγή αυτού η συνάρτηση **cv2.VideoCapture(filename)**.

3.1.7 Πληροφοριακό περιεχόμενο πλαισίων σφαλμάτων

Προκειμένου να βγάλουμε κάποιο συμπέρασμα σχετικά με το πληροφοριακό περιεχόμενο των πλαισίων σφαλμάτων, επιλέχθηκε, εκτός από τον υπολογισμό του χρόνου εκτέλεσης της διαδικασίας, και ο υπολογισμός της εντροπίας του τελικού βίντεο. Για τον υπολογισμό της, υλοποιήθηκε η συνάρτηση **calculate_entropy** και χρησιμοποιήθηκε η βιβλιοθήκη **entropy** της **scipy.stats**.

3.2 Ερώτημα Β'

3.2.1 Ανάγνωση βίντεο

Όπως και στο πρώτο ερώτημα το αρχικό βήμα για την επίλυση του ερωτήματος είναι η ανάγνωση του αρχείου βίντεο με το οποίο θα εργαστούμε στη συνέχεια, χρησιμοποιώντας τη βιβλιοθήκης **opencv-python**, και πιο συγκεκριμένα την μέθοδο **cv2.VideoCapture**.

3.2.2 Κωδικοποίηση Βίντεο

Η κωδικοποίηση του αρχείου προς επεξεργασία θα γίνει με την μέθοδο **cv2.VideoWriter_fourcc()**, κωδικοποιώντας το ως **'XVID'**.

3.2.3. Αρχείο για την εγγραφή του βίντεο

Επόμενο στάδιο είναι ο καθορισμός των παραμέτρων για την εγγραφή του βίντεο σε κάποιο αρχείο(Οι παράμετροι έχουν αναφερθεί εκτενώς στην ενότητα 3.1.3). Η μέθοδος που χρησιμοποιούμε είναι η **cv2.VideoWriter**.

3.2.4 Διαχείριση πλαισίου I

Λαμβάνοντας υπ' όψιν, ότι το πρώτο πλαίσιο θα είναι πάντα ένα **πλαίσιο I**, αφού αποθηκευτεί μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**. Στη συνέχεια θα κληθεί η συνάρτηση **black_pixels**, για το πλαίσιο αυτό. Ο σκοπός της συγκεκριμένης συνάρτησης θα αναλυθεί στην ενότητα 3.2.6.

3.2.5 Επαναληπτική διαδικασία για τα πλαίσια P

Η βασική συνθήκη για τον υπολογισμό της διαφοράς δύο πλαισίων και η προβολή των πλαισίων σφαλμάτων είναι να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα. Εφόσον, η συνθήκη είναι αληθής το τρέχων πλαίσιο είναι το επόμενο πλαίσιο P. Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**. Στη συνέχεια θα κληθεί η συνάρτηση **black_pixels**, για το πλαίσιο αυτό. Ο σκοπός της συγκεκριμένης συνάρτησης θα αναλυθεί στην παρακάτω ενότητα (3.2.6).

3.2.6 Συνάρτηση συμπλήρωσης πλαισίων με μαύρα εικονοστοιχεία

Η συνάρτηση **black_pixels** με ορίσματα ένα πλαίσιο(frame) και έναν ακέραιο αριθμό(η), ο οποίος είναι υπεύθυνος για την συνθήκη συμπλήρωσης με μαύρα εικονοστοιχεία ενός πλαισίου. Να υπενθυμιστεί σε αυτό το σημείο ότι πλαίσιο του οποίου το ύψος και το πλάτος δεν είναι πολλαπλάσιο του 16 υπόκειται στην συγκεκριμένη διαδικασία. Πιο αναλυτικά, για κάθε πλαίσιο θα γίνεται η διαίρεση του ύψους και του πλάτους με το 16, η οποία τιμή πολλαπλασιάζεται με το 16 και η τιμή αυτή αφαιρείται από την αρχική τιμή του ύψους και του πλάτους αντίστοιχα. Για την τιμή που θα προκύψει θα γίνει η αντίστοιχη επέκταση του πίνακα των pixel, γεμίζοντας της επιπλέον θέσεις με μαύρα εικονοστοιχεία. Η συγκεκριμένη συνάρτηση υλοποιήθηκε με την βοήθεια των συναρτήσεων της βιβλιοθήκης **numpy**, και πιο συγκεκριμένα: **np.pad & np.ceil**.

3.2.7 Διαίρεση P πλαισίων σε macroblock 16X16

Για την διαίρεση των P πλαισίου σε macroblock διαστάσεων 16X16, δεν χρησιμοποιήθηκε κάποια συνάρτηση αλλά ένα **for-loop**. Η κεντρική ιδέα είναι ότι για κάθε γραμμή και στήλη το macroblock ορίζεται ως έναν πίνακα της εξής μορφής: [i:i+16,j:j+16]. Δηλαδή, σε κάθε επανάληψη ο πίνακας του macroblock έχει πλήθος γραμμών από την τιμή του i έως και την τιμή του i+16, ενώ πλήθος στηλών από την τιμή του j έως και την τιμή του j+16.

3.2.8 Υπολογισμός διανυσμάτων κίνησης και λογαριθμική αναζήτηση του καλύτερου διανύσματος

Για τον υπολογισμό των διανυσμάτων κίνησης καθώς και της λογαριθμικής αναζήτησης για την εύρεση του καλύτερου διανύσματος, υλοποιήθηκε η συνάρτηση **compute_motion_vector**. Η περιοχή αναζήτησης καθορίζεται από την παράμετρο αναζήτησης k , η οποία είναι ίση με την τιμή 16 ($k=16$). Στόχος της αναζήτησης του διανύσματος κίνησης είναι να βρει την περιοχή εκείνη του πλαισίου αναφοράς που ταιριάζει καλύτερα με το μακρομπλόκ στόχο, ως προς όλα τα υποψήφια τμήματα στην περιοχή αναζήτησης. Ο τύπος αναζήτησης που χρησιμοποιήθηκε στην επίλυση της συγκεκριμένης άσκησης είναι η λογαριθμική, καθώς στο συγκεκριμένο σενάριο της εφαρμογής θα πρέπει οι υπολογισμοί να είναι όσο το δυνατόν πιο σύντομοι. Αρχικά, η μετρική υπολογίζεται επαναληπτικά σε εννέα θέσεις $((k/2, k/2), (0, k/2), (-k/2, k/2), (-k/2, 0), (-k/2, -k/2), (0, -k/2), (k/2, -k/2), (k/2, 0))$. Το πρώτο βήμα είναι ο υπολογισμός της μετρικής για το κέντρο της περιοχής και εν συνεχεία για τις παραπάνω περιοχές. Στην επόμενη επανάληψη, το κέντρο της περιοχής αναζήτησης μετακινείται στην καλύτερη θέση που υπολογίστηκε στο προηγούμενο βήμα, ενώ το βήμα αναζήτησης μειώνεται στο μισό. Η παραπάνω επαναληπτική διαδικασία συνεχίζεται έως ότου το μέγεθος του βήματος να γίνει ίσο με 1. Στο σημείο αυτό, από το καλύτερο ταίριασμα προκύπτει το καλύτερο διάνυσμα κίνησης.

3.2.9 Υπολογισμός μετρικής SAD

Η συνάρτηση **sad** υπολογίζει το άθροισμα της απόλυτης τιμής μεταξύ δύο μακρομπλόκ. Η συγκεκριμένη συνάρτηση χρησιμοποιείται από την συνάρτηση **compute_motion_vector**.

3.2.10 Υπολογισμός μπλοκ σφαλμάτων

Το μπλοκ σφαλμάτων ορίζεται ως τη διαφορά μεταξύ του αρχικού μπλοκ και του προβλεφθέντος. Συνεπώς, μετά από την εύρεση του καλύτερου διανύσματος κίνησης, και με άξονα τον ορισμό του μπλοκ σφαλμάτων θα το υπολογίσουμε για όλα τα macroblocks.

3.2.11 Ανακατασκευή πλαισίων

Προκειμένου, να δημιουργήσουμε το βίντεο με τα πλαίσια σφαλμάτων, θα πρέπει να ακολουθήσουμε την αντίστροφη διαδικασία της διαίρεσης των πλαισίων σε macroblocks. Η διαδικασία αυτή θα μπορούσε να περιγραφεί σαν την «ένωση» όλων των 16X16 macroblocks, επιστρέφοντας το αρχικό πλαίσιο, δηλαδή το πλαίσιο σφαλμάτων.

3.2.12 Προβολή πλαισίων σφαλμάτων

Όπως προαναφέρθηκε(Ενότητα 2), για την προβολή των πλαισίων σφαλμάτων επιλέχθηκε η δημιουργία ενός βίντεο. Για κάθε δύο πλαίσια, για τα οποία υπολογίζεται εκτελείται ο χωρισμός τους σε μακρομπλόκ διαστάσεων 16X16, υπολογίζονται τα διανύσματα κίνησης και βρίσκοντας το

καλύτερο με βάση τη λογαριθμική αναζήτηση, και στη συνέχεια τον υπολογισμό του μπλοκ σφαλμάτων και για τα 16 blocks του πλαισίου, προκύπτει το πλαίσιο σφαλμάτων, το οποίο γράφεται στο βίντεο και μετά το πέρας της επαναληπτικής διαδικασίας αποθηκεύεται στον φάκελο με το αρχείο του κώδικα ή δύναται να προβληθεί, εάν αυτό επιλεγεί από τον χρήστη. Για την αποθήκευση του βίντεο χρησιμοποιήθηκε η μέθοδος **.write()**, ενώ για την αναπαραγωγή αυτού η συνάρτηση **cv2.VideoCapture(filename)**.

3.2.13 Πληροφοριακό περιεχόμενο πλαισίων σφαλμάτων

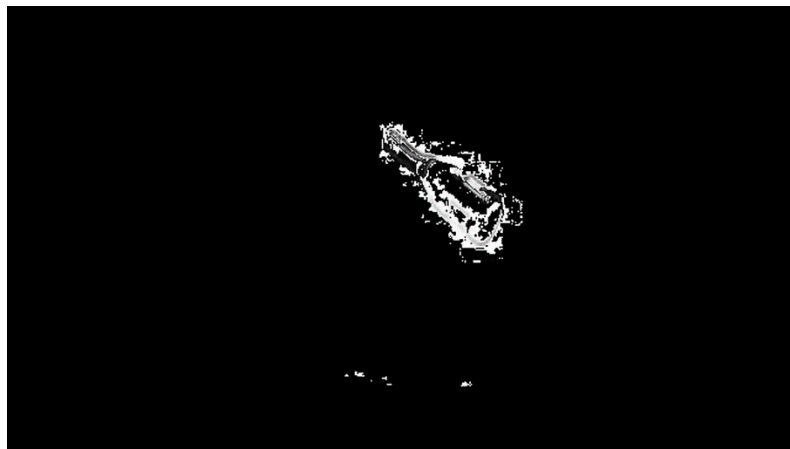
Προκειμένου να βγάλουμε κάποιο συμπέρασμα σχετικά με το πληροφοριακό περιεχόμενο των πλαισίων σφαλμάτων, επιλέχθηκε, εκτός από τον υπολογισμό του χρόνου εκτέλεσης της διαδικασίας, και ο υπολογισμός της εντροπίας του τελικού βίντεο. Για τον υπολογισμό της, υλοποιήθηκε η συνάρτηση **calculate_entropy** και χρησιμοποιήθηκε η βιβλιοθήκη **entropy** της **scipy.stats**.

4.Αποτελέσματα

4.1 Ερώτημα Α΄



Εικόνα 1- Πλαίσιο από το αρχικό βίντεο



Εικόνα 2- Πλαίσιο σφαλμάτων

Creating video with name "8_17a"...

Time elapsed: 0.2343583106994629s

Entropy: 0.2553997916055034

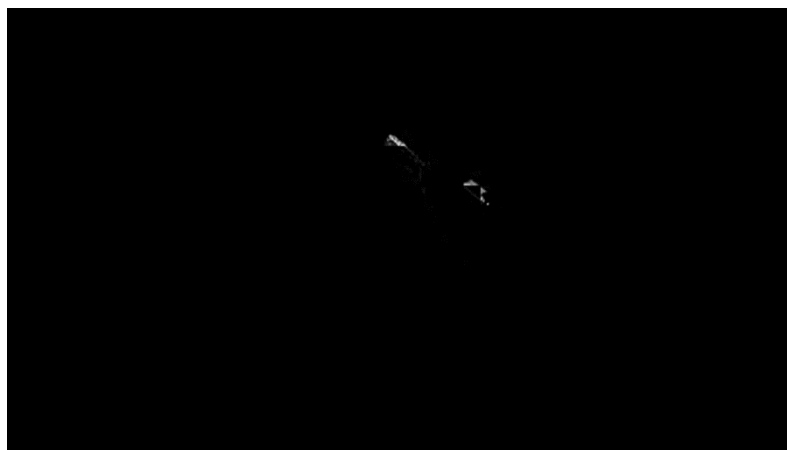
Finished!

Εικόνα 3- Πληροφοριακό περιεχόμενο τελικού βίντεο(χρόνος εκτέλεσης και εντροπία)

4.2 Ερώτημα Β'



Εικόνα 4- Πλαίσιο από το αρχικό βίντεο



Εικόνα 5- Πλαίσιο σφαλμάτων

```
Creating video with name "817b"...  
Frame #1 of 88 Completed.  
Frame #2 of 88 Completed.  
Frame #3 of 88 Completed.
```

Εικόνα 6-Έναρξη εκτέλεσης προγράμματος

```
Frame #83 of 88 Completed.  
Frame #84 of 88 Completed.  
Frame #85 of 88 Completed.  
Frame #86 of 88 Completed.  
Frame #87 of 88 Completed.  
Frame #88 of 88 Completed.  
  
Finished!  
The video has been added in your program's folder.  
  
Time elapsed: 361.75140380859375s  
  
Entropy: 0.10966190441230804
```

Εικόνα 7- Πληροφοριακό περιεχόμενο τελικού βίντεο(χρόνος εκτέλεσης και εντροπία)

5.Συμπεράσματα

Μετά την εκτέλεση και των δύο ερωτημάτων καταλήγουμε στα εξής συμπεράσματα:

1. Με την τεχνική της πρόβλεψης πλαισίων χωρίς την τεχνική της πρόβλεψης κίνησης παρατηρούμε ότι ο χρόνος εκτέλεσης του προγράμματος είναι πολύ μικρότερος σε σχέση με την τεχνική που χρησιμοποιήθηκε στο δεύτερο ερώτημα.
2. Στο πρώτο ερώτημα παρατηρείται ότι η εντροπία είναι μεγαλύτερη από αυτή του δεύτερου ερωτήματος και συγκεκριμένα ο λόγος των δύο εντροπιών είναι ο εξής:

$$\text{entropy_a/entropy_b} \approx 2,328$$

3. Συνεπώς, τεχνική της πρόβλεψης κίνησης με την χρήση της μετρικής SAD και την τεχνική της λογαριθμικής αναζήτησης, αν και χρονοβόρα, δίνει καλύτερα αποτελέσματα, επηρεάζοντας θετικά τα σφάλματα πρόβλεψης.

6.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:

- ❖ **numpy**
- ❖ **scipy**
- ❖ **opencv-python(cv2)**
- ❖ **time**
- ❖ **os**

Επίσης, ο χρήστης θα πρέπει να έχει προβλέψει το αρχείο-βίντεο που επιθυμεί να επεξεργαστεί να είναι στον ίδιο φάκελο με το αρχείο του κώδικα, δηλαδή στον φάκελο **8-17**.

Η επίλυση της άσκησης βρίσκεται στον φάκελο **8-17** και συγκεκριμένα στα αρχείο **8-17.py**.

Το συγκεκριμένο αρχείο υλοποιεί και τα δύο ερωτήματα της άσκησης, ωστόσο ο χρήστης μπορεί να επιλέξει πληκτρολογώντας το αντίστοιχο γράμμα στο command line. Πιο συγκεκριμένα, για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

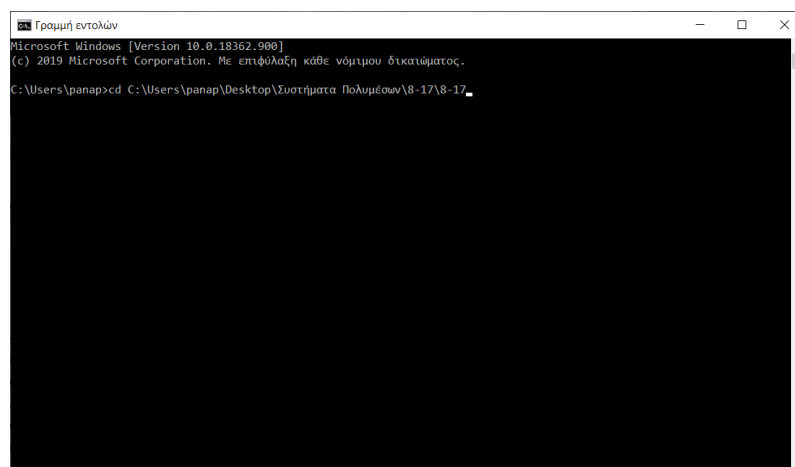
1. Ανοίγουμε τη γραμμή εντολών και μεταβαίνουμε στον αντίστοιχο φάκελο: **\Συστήματα Πολυμέσων\8-17\8-17.(Εικόνα 8)**
2. Πληκτρολογούμε το όνομα του αρχείου **8-17.py**, ώστε να ανοίξει το εκτελέσιμο.(**Εικόνα 9**)

Όταν ανοίξει το πρόγραμμα εμφανίζεται το menu του προγράμματος, όπου ο χρήστης πρέπει να επιλέξει ποιο ερώτημα επιθυμεί να εκτελέσει.(**Εικόνα 10**)

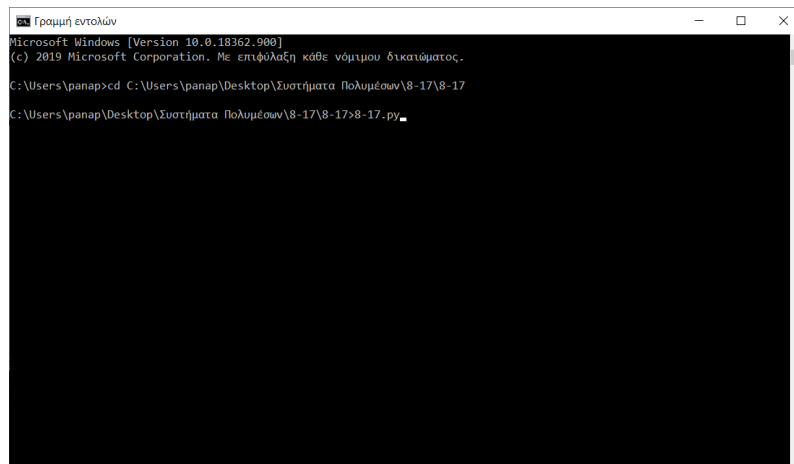
Σε περίπτωση που επιθυμεί ο χρήστης να εκτελέσει το πρώτο ερώτημα τότε πληκτρολογεί το: **a**, ενώ πληκτρολογεί το: **b**, εάν επιθυμεί την εκτέλεση του δεύτερου ερωτήματος. Το **c** πληκτρολογείται για την έξοδο από το πρόγραμμα.

Στην περίπτωση που επιλεγθεί είτε το **a**, είτε το **b**, τότε για να δημιουργηθεί το βίντεο για την προβολή των πλαισίων σφαλμάτων ή για την προβολή των πλαισίων σφαλμάτων με την τεχνική της πρόβλεψης κίνησης διαδικασία, αντίστοιχα, θα πρέπει να:

1. Εισάγει το όνομα του αρχείου, μαζί με την κατάληξη του, για το βίντεο για το οποίο θα εκτελεστεί η διαδικασία(π.χ.:**movie.m4v**, **Εικόνα 11**).
2. Εισάγει το όνομα του αρχείου , χωρίς την κατάληξη του, στο οποίο θα αποθηκεύεται το βίντεο με την ακολουθία πλαισίων σφαλμάτων.(**Εικόνα 12**)
3. Εάν ο χρήστης επιθυμεί την προβολή του παραπάνω αρχείου, τότε στο σχετικό μήνυμα θα πρέπει να πληκτρολογήσει τον χαρακτήρα: **y**, σε διαφορετική περίπτωση τον χαρακτήρα: **n**. (**Εικόνα 13,14 για το α' ερώτημα& Εικόνα 15,16 για το β' ερώτημα**)



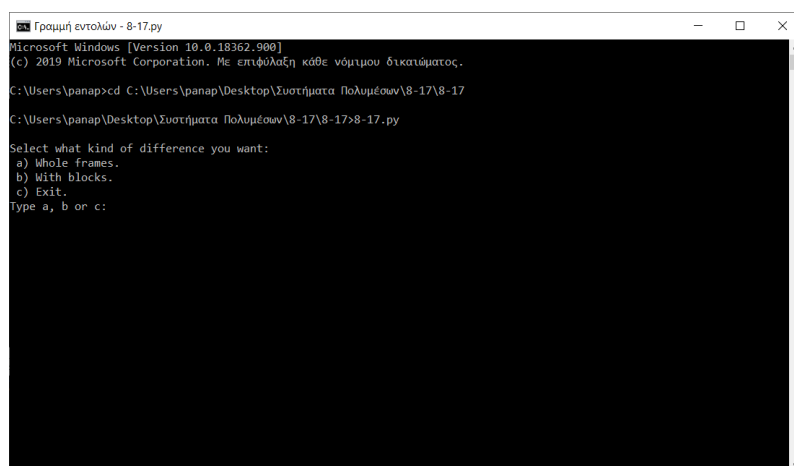
Εικόνα 8-Μετάβαση στο φάκελο, στον οποίο βρίσκεται το αρχείο του προγράμματος



```
Γραμμή εντολών
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17
C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17>8-17.py
```

Εικόνα 9-Εισαγωγή του ονόματος αρχείου προς εκτέλεση

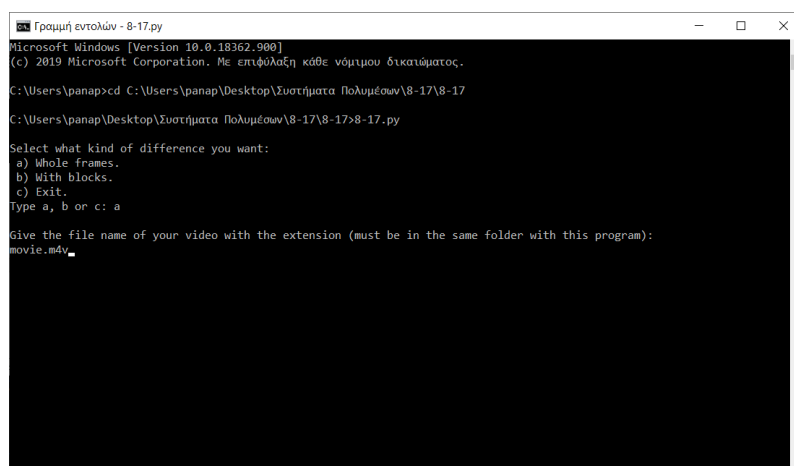


```
Γραμμή εντολών - 8-17.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17
C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17>8-17.py

Select what kind of difference you want:
a) Whole frames.
b) With blocks.
c) Exit.
Type a, b or c:
```

Εικόνα 10-«Menu» προγράμματος



```
Γραμμή εντολών - 8-17.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17
C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17>8-17.py

Select what kind of difference you want:
a) Whole frames.
b) With blocks.
c) Exit.
Type a, b or c: a

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v
```

Εικόνα 11-Εισαγωγή του ονόματος του αρχείου προς επεξεργασία

```
Γραμμή εντολών - 8-17.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17>8-17.py

Select what kind of difference you want:
a) Whole frames.
b) With blocks.
c) Exit.
Type a, b or c: a

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v

Give the file name of your output video: 8.17
```

Εικόνα 12- Εισαγωγή του ονόματος του αρχείου στο οποίο θα αποθηκευτεί το βίντεο των πλαισίων σφαλμάτων

```
Γραμμή εντολών - 8-17.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-17\8-17>8-17.py

Select what kind of difference you want:
a) Whole frames.
b) With blocks.
c) Exit.
Type a, b or c: a

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v

Give the file name of your output video: 8.17

Creating video with name "8.17"...

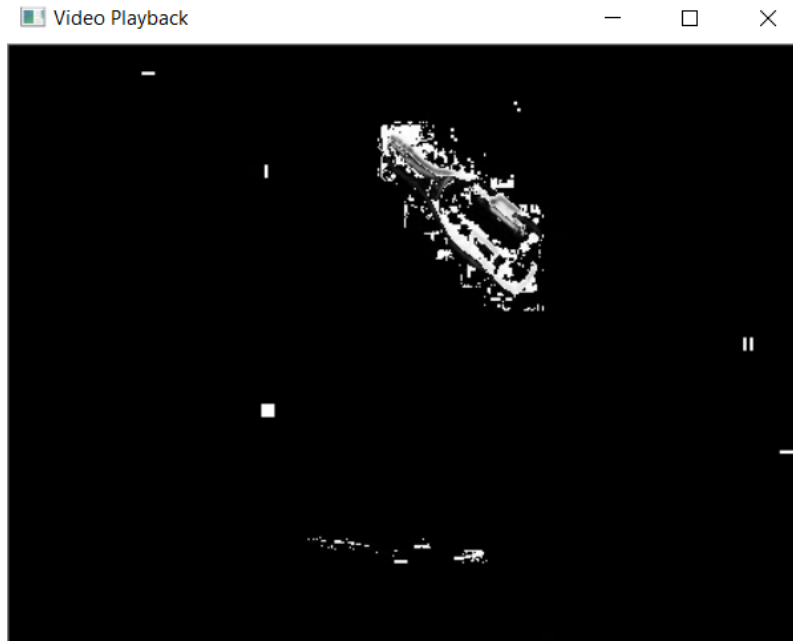
Time elapsed: 0.31914639472961426s

Entropy: 0.17702964546695227

Finished!
The video has been added in your program's folder.

Do you want to play the video? (y/n): y
```

Εικόνα 13- Επιλογή για την αναπαραγωγή του βίντεο των πλαισίων σφαλμάτων για το α' ερώτημα



*Εικόνα 14- Αναπαραγωγή του βίντεο των πλαισίων σφαλμάτων για το **α'** ερώτημα*

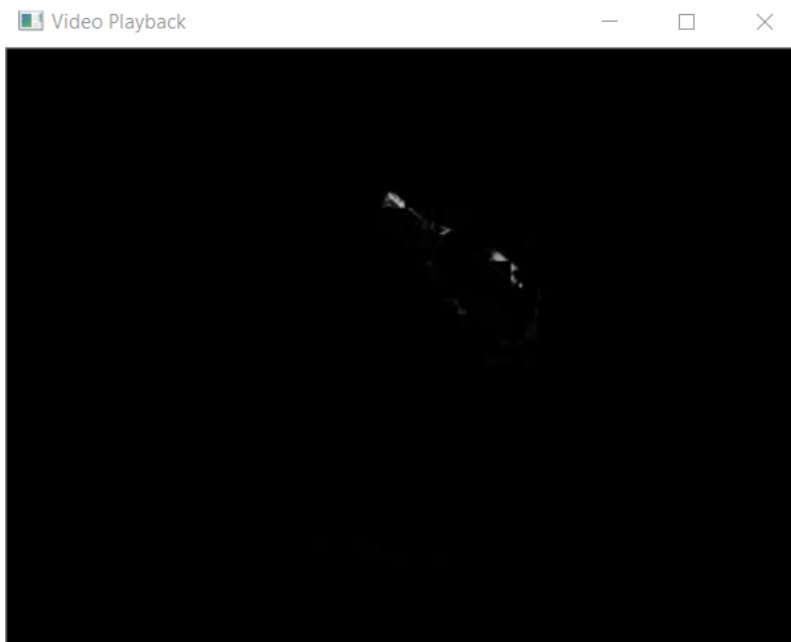
```
Γραμμή εντολών - 8-17.py
Frame #74 of 88 Completed.
Frame #75 of 88 Completed.
Frame #76 of 88 Completed.
Frame #77 of 88 Completed.
Frame #78 of 88 Completed.
Frame #79 of 88 Completed.
Frame #80 of 88 Completed.
Frame #81 of 88 Completed.
Frame #82 of 88 Completed.
Frame #83 of 88 Completed.
Frame #84 of 88 Completed.
Frame #85 of 88 Completed.
Frame #86 of 88 Completed.
Frame #87 of 88 Completed.
Frame #88 of 88 Completed.

Finished!
The video has been added in your program's folder.

Time elapsed: 389.41680788993835s
Entropy: 0.07601183985822554

Do you want to play the video? (y/n): y_
```

*Εικόνα 15- Επιλογή για την αναπαραγωγή του βίντεο των πλαισίων σφαλμάτων για το **β'** ερώτημα*



Εικόνα 16- Αναπαραγωγή του βίντεο των πλαισίων σφαλμάτων για το **6'**
ερώτημα

Άσκηση 8.18

1.Εκφώνηση Άσκησης

Σε αυτήν την άσκηση θα δείτε ότι η τεχνική της τμηματικής πρόβλεψης με βάση την αντιστάθμιση κίνησης, μπορεί επίσης να χρησιμοποιηθεί σε εφαρμογές εκτός συμπίεσης. Μία τέτοια ενδιαφέρουσα εφαρμογή είναι η απομάκρυνση αντικειμένων ή προσώπων από τη ροή του βίντεο. Για παράδειγμα, έστω ένα βίντεο στο οποίο η κάμερα δεν έχει κινηθεί και το παρασκήνιο είναι σχετικά στατικό, αλλά κινούνται ορισμένα αντικείμενα στο προσκήνιο. Στόχος σας είναι να προσεγγίσετε το αντικείμενο χρησιμοποιώντας μπλοκ και στη συνέχεια να αντικαταστήσετε αυτά τα μπλοκ με παρασκήνιο, σαν να μην ήταν ποτέ παρόν το αντικείμενο. Στη γενική περίπτωση, η λύση είναι πολύ δύσκολη, αλλά στο πλαίσιο αυτής της άσκησης θα επεξεργαστείτε ορισμένες απλούστερες ιδέες. Κατά την υλοποίηση, βεβαιωθείτε ότι μπορείτε να χειρίζεστε το μέγεθος του μπλοκ ως παράμετρο, προκειμένου να ελέγξετε πόσο καλά λειτουργεί ο αλγόριθμος απομάκρυνσης αντικειμένων για διάφορα μεγέθη μακροπλός.

- ❖ Κατ' αρχήν, φορτώστε ένα σύνολο πλαισίων βίντεο. Υποθέστε ότι το πρώτο πλαίσιο είναι αποκλειστικά πλαίσιο παρασκηνίου και δεν περιέχει αντικείμενα σε κίνηση. Δοθέντος ενός πλαισίου, η, προχωρήστε στη διαίρεσή του σε μπλοκ. Υπολογίστε ένα διάνυσμα κίνησης ανά μπλοκ με βάση το προηγούμενο πλαίσιο (αναφοράς). Για τα μπλοκ παρασκηνίου, θα πρέπει να προκύψουν μη μηδενικά διανύσματα κίνησης. Παρακολουθήστε όλα αυτά τα διανύσματα κίνησης. Μπορείτε ακόμη και να τα απεικονίσετε για κάθε μπλοκ, καθώς προβάλετε το βίντεο.
- ❖ Στη συνέχεια, βρείτε τα μπλοκ που αντιστοιχούν σε μη μηδενικά διανύσματα κίνησης. Αντικαταστήστε κάθε τέτοιο μπλοκ με το αντίστοιχο μπλοκ παρασκηνίου. Τα εικονοστοιχεία παρασκηνίου θα πρέπει να προέρχονται από προηγούμενο πλαίσιο, στο οποίο δεν υπήρχε αντικείμενο σε κίνηση. Η αντικατάσταση όλων αυτών των μπλοκ θα οδηγήσει στην απομάκρυνση των αντικειμένων, διότι τα

αντικείμενα αντικαθίστανται από το παρασκήνιο. Επαναλάβετε τη διαδικασία, χρησιμοποιώντας μακρομπλόκ διαφορετικού μεγέθους.

- ❖ Είναι πιθανό να αντιμετωπίσετε ασυνέχειες και παρενέργειες στα όρια των μπλοκ που αντικαθίστανται. Πως θα ελαχιστοποιήσετε αυτά τα φαινόμενα; Η προτεινόμενη λύση θα λειτουργήσει ικανοποιητικά, μόνο όταν υποθέστε ότι η κάμερα δεν κινείται, όταν τα αντικείμενα που κινούνται υπόκεινται σε λεία κίνηση και ότι δεν παρατηρούνται αλλαγές κατεύθυνσης. Όμως στη γενική περίπτωση, η κάμερα κινείται, τα αντικείμενα κινούνται με όχι αυστηρό τρόπο και επιπλέον, παρατηρούνται αλλαγές στον φωτισμό, καθώς η κάμερα κινείται.
- ❖ Αλλά ας υποθέσουμε ότι μπορείτε να εντοπίσετε όλα τα μακρομπλόκ ενός πλαισίου που αντιστοιχούν στο παρασκήνιο. Πως μπορείτε να αξιοποιήσετε αυτό το γεγονός, πλέον της χρήσης αντιστάθμισης κίνησης σε επίπεδο μακρομπλόκ;

2.Σύντομη παρουσίαση της κεντρικής ιδέας

Ζητούμενο στην παραπάνω άσκηση είναι η υλοποίηση μιας διαδικασίας, η οποία θα απομακρύνει αντικείμενα ή πρόσωπα από κάποιο βίντεο. Πιο συγκεκριμένα για κάποιο αντικείμενο ή πρόσωπο στο βίντεο το οποίο θα κινείται θα αντικαθίσταται από το παρασκήνιο. Η κίνηση θα εντοπίζεται και θα υπολογίζεται με την τεχνική της εύρεσης διανυσμάτων κίνησης. Ένα επιπλέον ζητούμενο της εκφώνησης είναι ο χειρισμός του μεγέθους του μπλοκ ως παράμετρο, ώστε να παρατηρήσουμε την αποδοτικότητα του αλγορίθμου. Σαν υποθέσεις εργασίας κάναμε τις παρακάτω:

1. Επιλέχθηκε ένα βίντεο 3 δευτερολέπτων, στο οποίο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου, δηλαδή η πτώση του μπουκαλιού. Συνεπώς, δεν παρατηρείται κάποια κίνηση στο παρασκήνιο. Η συγκεκριμένη υπόθεση συμβάλει στην καλύτερη προσέγγιση του κινούμενου αντικειμένου, άρα και στην αντικατάστασή του.
2. Το βίντεο, κάθε frame αυτού, μετατράπηκε από έγχρωμο, σε βίντεο με αποχρώσεις του γκρι(Grayscale).

3.Υλοποίηση

Για την υλοποίηση και την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού *Python 3*, οι απαιτήσεις για την εκτέλεση παρατίθενται στην **ενότητα 6**. Τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος θα αναλυθούν διεξοδικά παρακάτω.

3.1.1 Ανάγνωση Βίντεο

Το αρχικό βήμα στην υλοποίηση του ερωτήματος είναι μέσω της βιβλιοθήκης **opencv-python**, και πιο συγκεκριμένα με την μέθοδο **cv2.VideoCapture**, να μπορέσουμε να αναγνώσουμε το video με το οποίο θα εργαστούμε στη συνέχεια.

3.1.2 Κωδικοποίηση Βίντεο

Επόμενο βήμα στην υλοποίηση του συστήματος είναι η κωδικοποίηση του video, για το οποίο θα επιστρέψουμε τα πλαίσια σφαλμάτων. Πιο συγκεκριμένα, χρησιμοποιώντας την μέθοδο **cv2.VideoWriter_fourcc()**, και την κωδικοποίηση **'XVID'**.

3.1.3 Αρχείο για την εγγραφή του βίντεο

Στη συνέχεια, καθορίζονται και δίνονται οι παράμετροι για την εγγραφή του βίντεο σε κάποιο αρχείο. Απαραίτητη μέθοδος είναι η **cv2.VideoWriter**, η οποία παίρνει τις απαραίτητες παραμέτρους για:

- ❖ Το όνομα του αρχείου του βίντεο
- ❖ Την κωδικοποίηση (βλ. Ενότητα 3.1.2)
- ❖ Τα frames per second(fps)
- ❖ Το ύψος και το πλάτος του κάθε πλαισίου(size of frame)

3.1.4 Διαχείριση πρώτου πλαισίου-παρασκηνίου

Για το πρώτο πλαίσιο του video, λήφθηκε υπόψη ότι είναι αποκλειστικά πλαίσιο παρασκηνίου. Για αυτό το λόγο, αφού αποθηκευτεί μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**. Στη συνέχεια θα κληθεί η συνάρτηση **black_pixels**, για το πλαίσιο αυτό. Ο σκοπός της συγκεκριμένης συνάρτησης θα αναλυθεί στην ενότητα 3.1.6.

3.1.5 Επαναληπτική διαδικασία για τα πλαίσια P

Η βασική συνθήκη για τον υπολογισμό και την εύρεση των διανυσμάτων κίνησης και την αντικατάσταση των πλαισίων αυτών με τα αντίστοιχα του παρασκηνίου είναι να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα. Εφόσον, η συνθήκη είναι αληθής το τρέχων πλαίσιο είναι το επόμενο πλαίσιο P. Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε ασπρόμαυρο με την βοήθεια της συνάρτησης **cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)**. Στη συνέχεια, για κάθε ένα από αυτά τα πλαίσια καλείται η συνάρτηση **black_pixels**, της οποίας ο σκοπός θα αναλυθεί στην αμέσως επόμενη ενότητα.

3.1.6 Συνάρτηση συμπλήρωσης πλαισίων με μαύρα εικονοστοιχεία

Η συνάρτηση **black_pixels** με ορίσματα ένα πλαίσιο(frame) και έναν ακέραιο αριθμό(n), ο οποίος είναι υπεύθυνος για την συνθήκη συμπλήρωσης με μαύρα εικονοστοιχεία ενός πλαισίου. Να υπενθυμιστεί σε αυτό το σημείο ότι πλαίσιο του οποίου το ύψος και το πλάτος δεν είναι πολλαπλάσιο του 16 υπόκειται στη συγκεκριμένη διαδικασία. Πιο αναλυτικά, για κάθε πλαίσιο θα γίνεται η διαίρεση του ύψους και του πλάτους με το 16, η οποία τιμή πολλαπλασιάζεται με το 16 και η τιμή αυτή αφαιρείται από την αρχική τιμή του ύψους και του πλάτους αντίστοιχα. Για την τιμή που θα προκύψει θα γίνει η αντίστοιχη επέκταση του πίνακα των pixel, γεμίζοντας της επιπλέον θέσεις με μαύρα εικονοστοιχεία. Η συγκεκριμένη συνάρτηση υλοποιήθηκε με την

βοήθεια των συναρτήσεων της βιβλιοθήκης **numpy**, και πιο συγκεκριμένα: **np.pad** & **np.ceil**.

3.1.7 Διαίρεση P πλαισίων σε macroblock

Ζητούμενο σε αυτή την άσκηση είναι ο χειρισμός του μεγέθους του μπλοκ ως παράμετρο, προκειμένου να ελεγχθεί το πόσο καλά λειτουργεί ο αλγόριθμος. Συνεπώς, κατά την έναρξη του προγράμματος ζητείται από τον χρήστη να εισάγει το μέγεθος των μπλοκ. Για την διαίρεση των πλαισίων σε macroblock συγκεκριμένων διαστάσεων, δεν χρησιμοποιήθηκε κάποια συνάρτηση αλλά ένα **for-loop**. Η κεντρική ιδέα είναι ότι για κάθε γραμμή και στήλη το macroblock ορίζεται ως έναν πίνακα της εξής μορφής: $[i:i+X, j:j+X]$, όπου X είναι η τιμή που έχει εισάγει ο χρήστης. Δηλαδή, σε κάθε επανάληψη ο πίνακας του macroblock έχει πλήθος γραμμών από την τιμή του i έως και την τιμή του $i+X$, ενώ πλήθος στηλών από την τιμή του j έως και την τιμή του $j+X$.

3.1.8 Υπολογισμός διανυσμάτων κίνησης και λογαριθμική αναζήτηση του καλύτερου διανύσματος

Για τον υπολογισμό των διανυσμάτων κίνησης, συνεπώς και του εντοπισμού κίνησης, καθώς και της λογαριθμικής αναζήτησης για την εύρεση του καλύτερου διανύσματος, υλοποιήθηκε η συνάρτηση **compute_motion_vector**. Η περιοχή αναζήτησης καθορίζεται από την παράμετρο αναζήτησης k , η οποία είναι ίση με την τιμή 16 ($k=16$). Στόχος της αναζήτησης του διανύσματος κίνησης είναι να βρει την περιοχή εκείνη του πλαισίου αναφοράς που ταιριάζει καλύτερα με το μακρομπλόκ στόχο, ως προς όλα τα υποψήφια τμήματα στην περιοχή αναζήτησης. Ο τύπος αναζήτησης που χρησιμοποιήθηκε στην επίλυση της συγκεκριμένης άσκησης είναι η λογαριθμική, καθώς στο συγκεκριμένο σενάριο της εφαρμογής θα πρέπει οι υπολογισμοί να είναι όσο το δυνατόν πιο σύντομοι. Αρχικά, η μετρική υπολογίζεται επαναληπτικά σε εννέα θέσεις $((k/2, k/2), (0, k/2), (-k/2, k/2), (-k/2, 0), (-k/2, -k/2), (0, -k/2), (k/2, -k/2), (k/2, 0))$. Το πρώτο βήμα είναι

ο υπολογισμός της μετρικής για το κέντρο της περιοχής και εν συνεχεία για τις παραπάνω περιοχές. Στην επόμενη επανάληψη, το κέντρο της περιοχής αναζήτησης μετακινείται στην καλύτερη θέση που υπολογίστηκε στο προηγούμενο βήμα, ενώ το βήμα αναζήτησης μειώνεται στο μισό. Η παραπάνω επαναληπτική διαδικασία συνεχίζεται έως ότου το μέγεθος του βήματος να γίνει ίσο με 1. Στο σημείο αυτό, από το καλύτερο ταίριασμα προκύπτει το καλύτερο διάνυσμα κίνησης.

3.1.9 Υπολογισμός μετρικής SAD

Η συνάρτηση **sad** υπολογίζει το άθροισμα της απόλυτης τιμής μεταξύ δύο μακρομπλόκ. Η συγκεκριμένη συνάρτηση χρησιμοποιείται από την συνάρτηση **compute_motion_vector**.

3.1.10 Εύρεση κίνησης και αντικατάσταση μπλοκ με το αντίστοιχο μπλοκ παρασκηνίου

Μετά τον υπολογισμό των διανυσμάτων, επιθυμούμε να εντοπίσουμε τα μπλοκ με μη μηδενικά διανύσματα κίνησης ώστε αυτά να αντικατασταθούν με τα αντίστοιχα μπλοκ παρασκηνίου. Η συνθήκη για την παραπάνω διαδικασία είναι η εξής:

$$\text{motion_v}[0] + \text{motion_v}[1] \neq 0$$

Στη συνέχεια, και εφόσον η συνθήκη είναι αληθής, το μπλοκ αντικαθίσταται από το αντίστοιχο μπλοκ παρασκηνίου του προηγούμενου πλαίσιο(αναφοράς).

3.1.11 Ανακατασκευή πλαισίων

Προκειμένου, να δημιουργήσουμε το βίντεο με τα ανακατασκευασμένα πλαίσια, μετά την απομάκρυνση θα πρέπει να ακολουθήσουμε την αντίστροφη διαδικασία της διαίρεσης των πλαισίων σε macroblocks. Η διαδικασία αυτή θα μπορούσε να περιγραφεί σαν την «ένωση» όλων των macroblocks. Η διαδικασία περιλαμβάνει την αφαίρεση των μαύρων εικονοστοιχείων που προστέθηκαν σε προηγούμενο στάδιο. Για την διαδικασία αυτή χρησιμοποιήθηκε η συνάρτηση της βιβλιοθήκης **numpy**, την **np.delete**.

3.1.12 Προβολή πλαισίων σφαλμάτων

Όπως προαναφέρθηκε(Ενότητα 2), για την προβολή των πλαισίων σφαλμάτων επιλέχθηκε η δημιουργία ενός βίντεο. Για κάθε πλαίσιο το οποίο υπόκειται στην παραπάνω προβλέπεται η εγγραφή αυτού στο βίντεο και μετά το πέρας της επαναληπτικής διαδικασίας, το τελικό βίντεο, αποθηκεύεται στον φάκελο με το αρχείο του κώδικα ή δύναται να προβληθεί, εάν αυτό επιλεγεί από τον χρήστη. Για την αποθήκευση του βίντεο χρησιμοποιήθηκε η μέθοδος **.write()**, ενώ για την αναπαραγωγή αυτού η συνάρτηση **cv2.VideoCapture(filename)**.

4. Αποτελέσματα



Εικόνα 17- Πλαίσιο από το αρχικό βίντεο



Εικόνα 18- Πλαίσιο από το τελικό βίντεο, με μέγεθος μπλοκ 8.



Εικόνα 19- Πλαίσιο από το τελικό βίντεο, με μέγεθος μπλοκ 16.



Εικόνα 20- Πλαίσιο από το τελικό βίντεο, με μέγεθος μπλοκ 32.

5.Συμπεράσματα

Μετά την εκτέλεση και των δύο ερωτημάτων καταλήγουμε στα εξής συμπεράσματα:

1. Παρατηρούμε ότι με την συγκεκριμένη υλοποίηση δεν καταφέραμε να απομακρύνουμε το αντικείμενο με 100% επιτυχία, καθώς είναι εύκολο να διακρίνουμε ορισμένα pixels, τα οποία δεν αντικαταστάθηκαν. Αυτό οφείλεται στις ενδεχόμενες αλλαγές που μπορεί να υπάρχουν στο παρασκήνιο, στη φωτεινότητα ακόμα και στη σκιά του μπουκαλιού.
2. Η τεχνική αυτή έχει κάποιες «παρενέργειες» στο τελικό βίντεο, αφού κάποιος πολύ προσεκτικός παρατηρητής μπορεί να διαπιστώσει την απότομη και μη κλιμακούμενη αλλαγή στα εικονοστοιχεία στο πρώτο μισό, δηλαδή από τη μέση και πάνω, του βίντεο. Αυτό που διακρίνεται, είναι η θόλωση ορισμένων εικονοστοιχείων και κατά κύριο λόγο αυτών που αντικαταστάθηκαν.
3. Να σημειωθεί ότι για μεγαλύτερες τιμές της παραμέτρου για το μπλοκ, το αποτέλεσμα είναι χειρότερο, καθώς η αντικατάσταση δεν είναι σε μεγάλο βαθμός επιτυχής.

Προκειμένου να αντιμετωπιστούν οι παρενέργειες που παρατηρήθηκαν προτείνονται οι παρακάτω λύσεις:

1. Λαμβάνοντας υπ' όψη τον φωτισμό και άλλα χαρακτηριστικά του παρασκηνίου, τα οποία επηρεάζουν την διαδικασία και δεν είναι διακριτά, ο συνδυασμός μακρομπλόκ τόσο από το ίδιο, όσο και από τα προηγούμενα πλαίσια, θα μπορούσε να βελτιστοποιήσει το αποτέλεσμα.
2. Επειδή το μπουκάλι(ή στη γενική περίπτωση κάποιο αντικείμενο/ χαρακτήρας)δεν καλύπτει όλο το παρασκήνιο θα μπορούσαμε να εφαρμόσουμε την αντικατάσταση μακρομπλόκ από το ίδιο πλαίσιο(τρέχων) και όχι από το προηγούμενο.

6.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:

- ❖ **numpy**
- ❖ **opencv-python(cv2)**
- ❖ **time**
- ❖ **os**

Επίσης, ο χρήστης θα πρέπει να έχει προβλέψει το αρχείο-βίντεο που επιθυμεί να επεξεργαστεί να είναι στον ίδιο φάκελο με το αρχείο του κώδικα, δηλαδή στον φάκελο **8-18**.

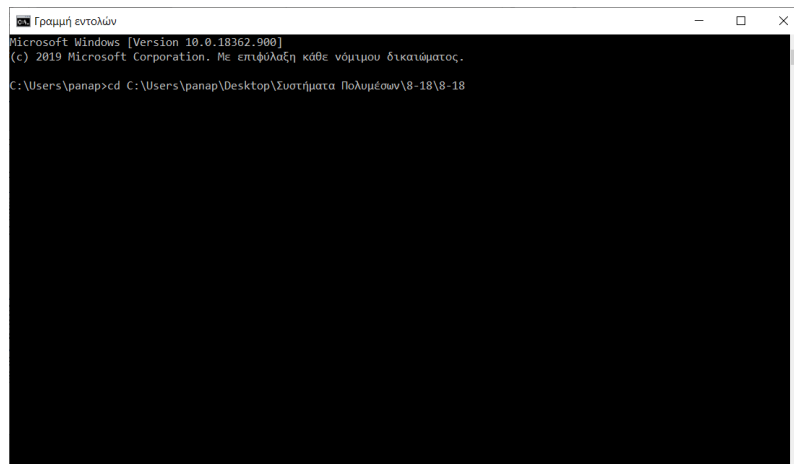
Η επίλυση της άσκησης βρίσκεται στον φάκελο **8-18** και συγκεκριμένα στα αρχείο **8-18.py**.

Για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

1. Ανοίγουμε τη γραμμή εντολών και μεταβαίνουμε στον αντίστοιχο φάκελο: **\Συστήματα Πολυμέσων\8-18\8-18.(Εικόνα 21)**
2. Πληκτρολογούμε το όνομα του αρχείου **8-18.py**, ώστε να ανοίξει το εκτελέσιμο.(**Εικόνα 22**)

Προκειμένου να δημιουργηθεί το βίντεο για την προβολή των πλαισίων μετά την απομάκρυνση του αντικειμένου/ ή προσώπου, θα πρέπει να:

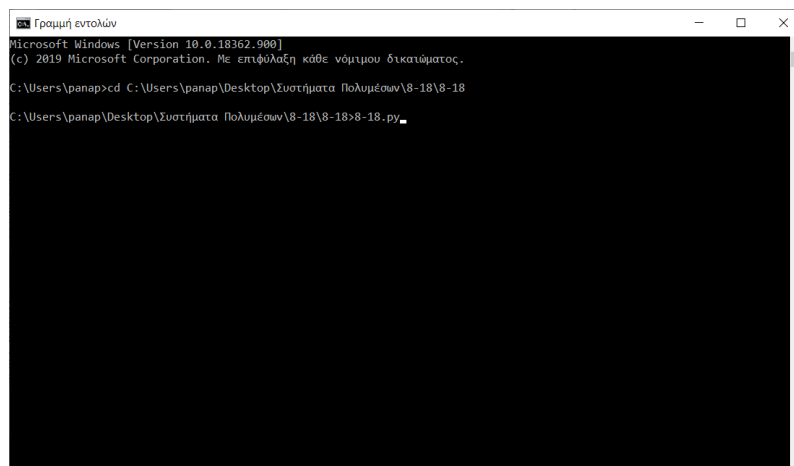
1. Εισάγει το όνομα του αρχείου, μαζί με την κατάληξη του, για το βίντεο για το οποίο θα εκτελεστεί η διαδικασία(π.χ.:**movie.m4v, Εικόνα 23**).
2. Εισάγει το όνομα του αρχείου , χωρίς την κατάληξη του, στο οποίο θα αποθηκεύεται το βίντεο με την ακολουθία πλαισίων σφαλμάτων.(**Εικόνα 24**)
3. Εισάγει την τιμή της παραμέτρου, για το μέγεθος του μπλοκ. (**Εικόνα 25**)
4. Εάν ο χρήστης επιθυμεί την προβολή του παραπάνω αρχείου, τότε στο σχετικό μήνυμα θα πρέπει να πληκτρολογήσει τον χαρακτήρα: **y**, σε διαφορετική περίπτωση τον χαρακτήρα: **n**. (**Εικόνα 26,27**)



```
Γραμμή εντολών
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18
```

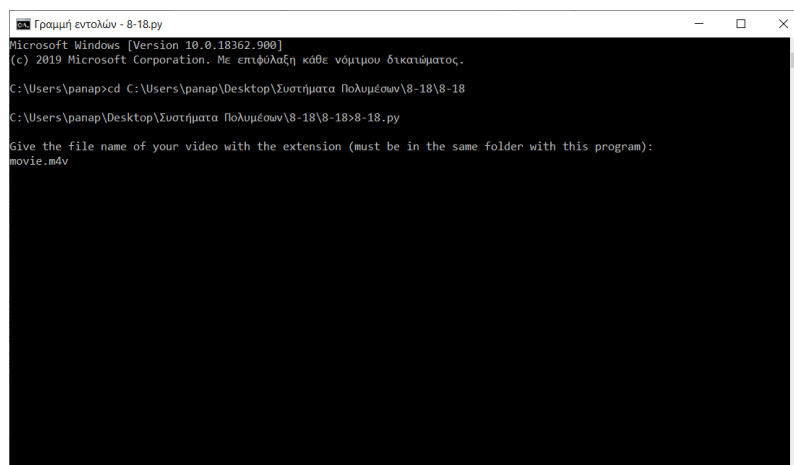
Εικόνα 21-Μετάβαση στο φάκελο, στον οποίο βρίσκεται το αρχείο του προγράμματος



```
Γραμμή εντολών
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18
C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18>8-18.py
```

Εικόνα 22-Εισαγωγή του ονόματος αρχείου προς εκτέλεση

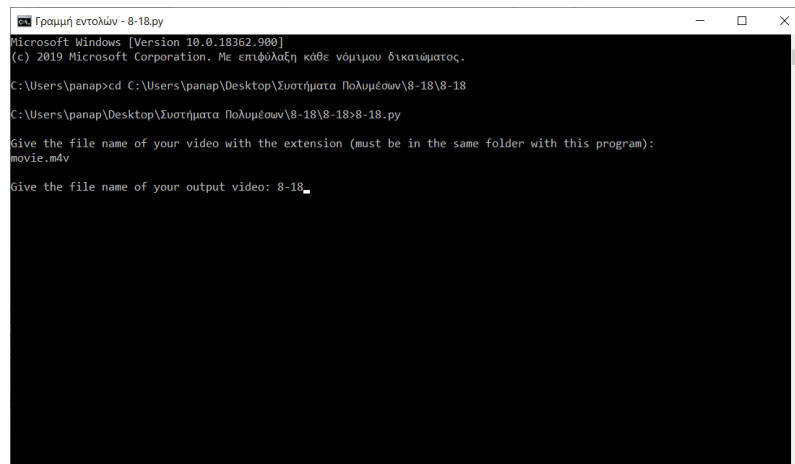


```
Γραμμή εντολών - 8-18.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18
C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18>8-18.py

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v
```

Εικόνα 23- Εισαγωγή του ονόματος του αρχείου προς επεξεργασία



```
Γραμμή εντολών - 8-18.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

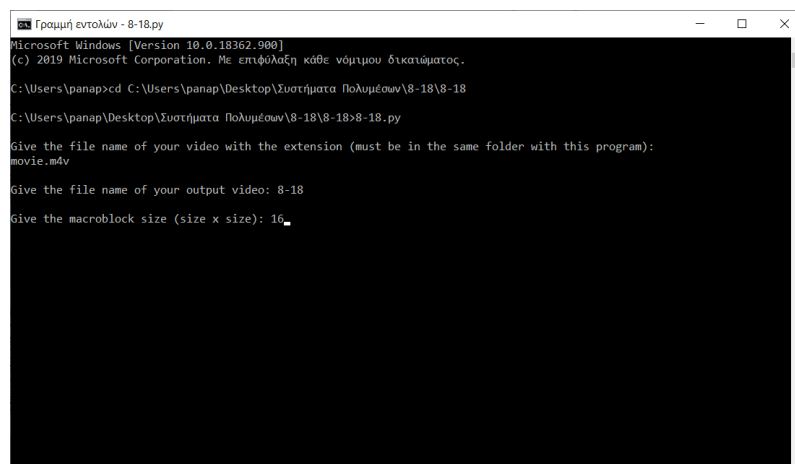
C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18>8-18.py

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v

Give the file name of your output video: 8-18_
```

Εικόνα 24- Εισαγωγή του ονόματος του αρχείου στο οποίο θα αποθηκευτεί το βίντεο των πλαισίων σφαλμάτων



```
Γραμμή εντολών - 8-18.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\8-18\8-18>8-18.py

Give the file name of your video with the extension (must be in the same folder with this program):
movie.m4v

Give the file name of your output video: 8-18

Give the macroblock size (size x size): 16_
```

Εικόνα 25-Εισαγωγή της τιμής της παραμέτρου, για το μέγεθος του μπλοκ

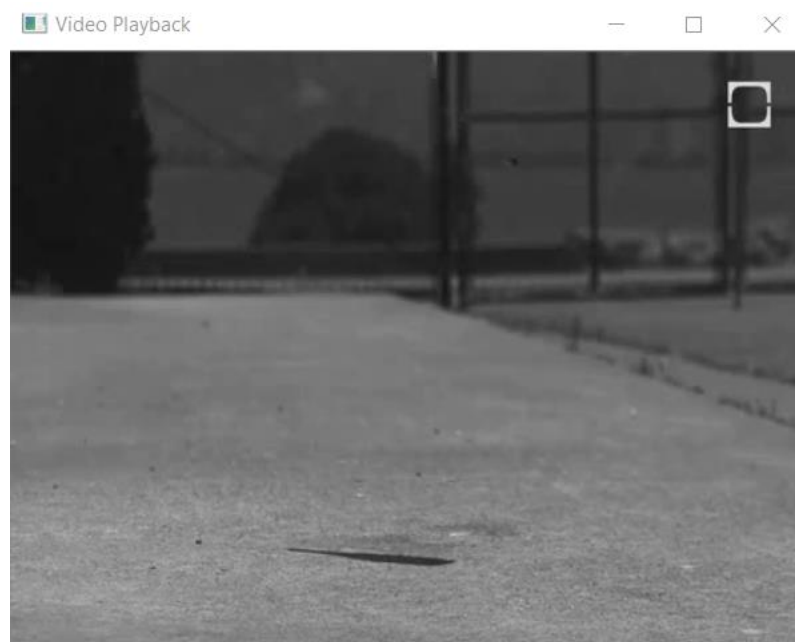
```
Γραμμή εντολών - 8-18.py
Frame #72 of 88 Completed.
Frame #73 of 88 Completed.
Frame #74 of 88 Completed.
Frame #75 of 88 Completed.
Frame #76 of 88 Completed.
Frame #77 of 88 Completed.
Frame #78 of 88 Completed.
Frame #79 of 88 Completed.
Frame #80 of 88 Completed.
Frame #81 of 88 Completed.
Frame #82 of 88 Completed.
Frame #83 of 88 Completed.
Frame #84 of 88 Completed.
Frame #85 of 88 Completed.
Frame #86 of 88 Completed.
Frame #87 of 88 Completed.
Frame #88 of 88 Completed.

Time elapsed: 453.12768721580505s

Finished!
The video has been added in your program's folder.

Do you want to play the video? (y/n): y_
```

Εικόνα 26-Επιλογή για την αναπαραγωγή του νέου βίντεο



Εικόνα 27-Αναπαραγωγή βίντεο

Β' Μέρος

1. Εκφώνηση Άσκησης

- Έστω ομάδα N φοιτητών που έχει δηλωθεί στο μάθημα ($1 \leq N \leq 3$).
- Έστω ότι A_i οι αντίστοιχοι πενταψήφιοι αριθμοί μητρώου των φοιτητών αυτών. - Κατασκευάστε συνθετική εικόνα διαστάσεων 104 γραμμών x 200 στηλών ως εξής: Κάθε γραμμή σχηματίζεται από πεντάδες αριθμών και κάθε πεντάδα είναι ένας A_i που επιλέγεται κάθε φορά τυχαία. Επίσης, σε κάθε πεντάδα, προτού αυτή ενσωματωθεί στην εικόνα, επιλέγεται τυχαία ένα ψηφίο και αντικαθίσταται με το 5.
- Αφού κατασκευάσετε την εικόνα, διαιρέστε τη σε macroblock 8x8 και σε κάθε macroblock εφαρμόστε τη διαδικασία που περιγράφεται στο Σχήμα 7-7 (σελ. 226) του βιβλίου.

Ερώτημα 1

Ποιος είναι ο μέσος λόγος συμπίεσης της εικόνας που προκύπτει συνενώνοντας τα macroblocks των κβαντισμένων DCT συντελεστών (έστω εικόνα B) σε σχέση με την αρχική εικόνα (έστω εικόνα A), αν η διαδικασία κατασκευής της εικόνας A επαναληφθεί 100 φορές. Χρησιμοποιήστε μόνο κωδικοποίηση Huffman για τις εικόνες A και B και όχι άλλες μεθόδους όπως, π.χ., Κωδικοποίηση Μήκους Διαδρομής.

2.Σύντομη παρουσίαση της κεντρικής ιδέας

Ένα από τα ζητούμενα του συγκεκριμένου προβλήματος ήταν η είσοδος και η διαχείριση πενταψήφιων αριθμών, δηλαδή αριθμοί μητρώων οι οποίοι να είναι σε πλήθος από 1 έως και 3. Αυτή ή αυτές οι πεντάδες πρέπει κάθε φορά να διαχειρίζονται με τον κατάλληλο τρόπο, ώστε δημιουργείται μια εικόνα 104 γραμμών και 200 στηλών. Αρχικά, για κάθε πεντάδα αριθμών που εισάγεται θα επιλέγεται τυχαία ένα ψηφίο και θα αντικαθίσταται από το ψηφίο 5. Η διαδικασία της κατασκευής εικόνας θα γίνεται με επαναληπτικό τρόπο ως εξής: Κάθε τροποποιημένος πενταψήφιος αριθμός που θα βρίσκεται στον πίνακα, θα επιλέγεται με τυχαίο τρόπο ώστε να συμπληρώνει συνεχόμενες θέσεις στον πίνακα της εικόνας. Η διαδικασία επαναλαμβάνεται τόσες φορές όσες είναι απαραίτητο για να δημιουργηθεί η απαιτούμενη εικόνα. Η ανωτέρα διαδικασία θα αναλυθεί παρακάτω. Να σημειωθεί σε αυτό το σημείο ότι οι εικόνες που θα δημιουργούνται κάθε φορά που θα εκτελείται ο κώδικας θα είναι 100. Προκειμένου να απαντηθεί το πρώτο ερώτημα θα κατασκευαστεί μια εικόνα B, η οποία θα είναι «προϊόν» της εφαρμογής της διαδικασίας που περιγράφεται στο Σχήμα 7-7, στη σελίδα 226, του βιβλίου, δηλαδή της κωδικοποίησης εικόνας με βάση τον μετασχηματισμό DCT¹. Όπως θα δούμε και στην ενότητα 3, για να οδηγηθούμε στο τελικό μας συμπέρασμα, δηλαδή την εύρεση του μέσου λόγου συμπίεσης, απαιτείται οι εικόνες να κωδικοποιηθούν με τον αλγόριθμο Huffman.

¹ Να σημειωθεί ότι στο αρχείο με τον κώδικα, σε σχόλια, είναι υλοποιημένη η διαδικασία μέχρι τέλους, δηλαδή και ο αντίστροφος μετασχηματισμός DCT.

3.Υλοποίηση

Για την υλοποίηση και την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού *Python 3*, οι απαιτήσεις για την εκτέλεση παρατίθενται στην **ενότητα 6**. Τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος θα αναλυθούν διεξοδικά παρακάτω.

3.1.1 Είσοδος πενταψήφιων αριθμών

Πρώτο βήμα για την επίλυση της άσκησης είναι η δημιουργία ενός πίνακα με πενταψήφιους αριθμούς, των οποίων το πλήθος θα ανήκει στο κλειστό σύνολο $[1,3]$. Ο χρήστης έχει τη δυνατότητα να εισάγει αριθμούς της επιθυμίας του, πάντα όμως εναρμονιζόμενος με τους περιορισμούς. Οι αριθμοί αυτοί αποθηκεύονται σε έναν πίνακα *A*.

3.1.2 Αντικατάσταση τυχαίου ψηφίου με το 5

Επόμενο βήμα στην υλοποίηση της άσκησης είναι η αλλαγή ενός τυχαίου ψηφίου σε κάθε πενταψήφιο αριθμό. Η διαδικασία αυτή υλοποιήθηκε ως εξής: Για κάθε αριθμό που είναι αποθηκευμένος στον πίνακα θα επιλέγεται ένας τυχαίο ψηφίο, το οποίο και θα αντικαθίσταται από το 5. Για την επίτευξη του στόχου μας έγινε η χρήση της συνάρτησης **random.randint()**, με την οποία το ψηφίο προς αντικατάσταση θα επιλέγεται.

3.1.3 Κατασκευή εικόνας A

Η εικόνα *A* θεωρήθηκε ότι είναι η αρχική εικόνα, δηλαδή αυτή η οποία θα δημιουργείται με βάση τους πενταψήφιους αριθμούς μητρώου των φοιτητών. Όπως αναφέρθηκε και στην ενότητα 2, η διαδικασία κατασκευής θα είναι επαναληπτική (100 επαναλήψεις-κατασκευές εικόνων), ωστόσο σε αυτή την υποενότητα θα αναλυθεί η διαδικασία για την πρώτη επανάληψη. Αρχικά, με την βοήθεια της συνάρτησης **np.zeros** θα δημιουργηθεί ο πίνακας της εικόνας, ο οποίος θα έχει διαστάσεις 104 γραμμών και 200 στηλών και θα είναι γεμάτος με μηδενικές τιμές. Ουσιαστικά η διαδικασία συμπλήρωσης του πίνακα με τα ψηφία των αριθμών του πίνακα *A* είναι μια επαναληπτική

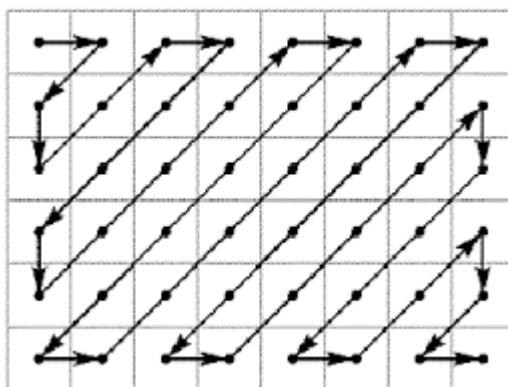
διαδικασία. Λαμβάνοντας υπόψη ότι το μήκος των αριθμών είναι 5 και ο αριθμός των στηλών 200, οδήγησε στο συμπέρασμα ότι ο αριθμός των επαναλήψεων για το πλάτος της εικόνας θα είναι $40(200/5)$. Έτσι, αφού επιλεγθεί τυχαία ένας αριθμός από τον αρχικό πίνακα αυτός «τοποθετείται» πέντε θέσεις μετά από το αντίστοιχο ψηφίο του προηγούμενου αριθμού. Όπως και προηγουμένως για την επιλογή τυχαίου αριθμού χρησιμοποιήθηκε η συνάρτηση **random.randint()**.

3.1.4 Κατασκευή εικόνας B

Η εικόνα B θεωρήθηκε ότι είναι κατασκευασμένη εικόνα μετά από την συνένωση των macroblocks των κβαντισμένων DCT συντελεστών. Για την κατασκευή της εικόνας B ακολουθήθηκε η διαδικασία της εικόνας 7-7 της σελίδας 226 του βιβλίου. Αρχικά δημιουργήθηκε ένας πίνακας διαστάσεων διαστάσεις 104 γραμμών και 200 στηλών και θα είναι γεμάτος με μηδενικές, ωστόσο θα μπορεί να 'δέχεται και δεκαδικούς αριθμούς. Για την αρχικοποίηση του χρησιμοποιήθηκε η συνάρτηση **np.zeros**. Η εικόνα B θα χωριστεί σε macroblock 8x8, ώστε να μπορέσει να πραγματοποιηθεί η ζητούμενη διαδικασία για κάθε macroblock. Επόμενο βήμα είναι η εισαγωγή των dct συντελεστών, όπως αυτοί προκύπτουν από την αρχική εικόνα A, οποία επίσης έχει χωριστεί σε macroblock 8x8. Η «αρμόδια» συνάρτηση είναι η **cv2.dct**. Για το επόμενο στάδιο απαραίτητη προϋπόθεση ήταν η ύπαρξη ενός πίνακα με τους συντελεστές κβάντισης. Αυτός ο πίνακας αποτελείται από τους ίδιους συντελεστές που υπάρχουν στον αντίστοιχο πίνακα στο βιβλίο. Έτσι, θα προκύψει η εισαγωγή των κβαντισμένων dct συντελεστών στον πίνακα της εικόνας B. Πιο συγκεκριμένα ο συγκεκριμένος πίνακας προκύπτει από την διαίρεση του πίνακα με τους dct συντελεστές με τους συντελεστές κβάντισης, κρατώντας από το αποτέλεσμα αυτής της πράξης μόνο το ακέραιο μέρος. Οι συναρτήσεις που χρησιμοποιήθηκαν είναι η **np.divide** και **np.ceil**, αντίστοιχα.

3.1.5 Zigzag Scan

Η συγκεκριμένη διαδικασία εφαρμόζεται τόσο για τα macroblock της εικόνας A, μετά από τον χωρισμό αυτής σε macroblock 8x8, όσο και για αυτά της εικόνας B, της οποίας ο πίνακας είναι το αποτέλεσμα της εισαγωγής των κβαντισμένων dct συντελεστών(δηλαδή είναι η συμπιεσμένη εικόνα A). Για την υλοποίησή της δημιουργήθηκε η συνάρτηση **zigzag**. Σκοπός αυτής είναι το σκανάρισμα των macroblock της εικόνας με κατάλληλο τρόπο, ώστε να δημιουργηθεί ένας μονοδιάστατος πίνακας, το οποίο θα βοηθήσει όπως θα δούμε και παρακάτω στην κωδικοποίηση Huffman. Η οπτικοποίηση της διαδικασίας αυτής αποτυπώνεται στην Εικόνα 28.



Εικόνα 28- Zigzag Scan

3.1.6 Κωδικοποίηση Huffman

Όπως περιεγράφηκε στην προηγούμενη υποενότητα, το σκανάρισμα Zigzag αποτελεί σημαντικό προαπαιτούμενο για την κωδικοποίηση Huffman. Η κωδικοποίηση Huffman θα γίνει τόσο για την εικόνα A, όσο και για την B. Για την συγκεκριμένη διαδικασία υλοποιήθηκε η συνάρτηση **huffman_encoding**. Μετά την κλήση της συνάρτησης **zigzag** θα βρεθούν οι κωδικοί huffman για το κάθε macroblock. Η διαδικασία αυτή γίνεται με την μέθοδο **huffman.codebook(collection.Counter(X).items())**. Περιληπτικά, στη θέση του **X** είναι τα στοιχεία του μακρομπλόκ, έτσι όπως έχουν προκύψει από το zigzag scanning, και το αποτέλεσμα της παραπάνω μεθόδου είναι η δημιουργία ενός dictionary της μορφής :

{symbol: code, ...}

,δηλαδή σε κάθε σύμβολο αποδίδεται ένας κωδικός, ο οποίος αντικατοπτρίζει και την συχνότητα εμφάνισής του. Για παράδειγμα ένας αριθμός που εμφανίζεται 6 φορές σε σχέση με έναν άλλον που εμφανίζεται μόνο μια φορά θα έχει κωδικό μικρότερου μήκους. Αφότου έχει γίνει η διαδικασία για όλα τα macroblock, τελικό βήμα είναι η απόδοση των παραπάνω κωδικών στα στοιχεία του μονοδιάστατου πίνακα που δημιουργήθηκε σύμφωνα με την διαδικασία που περιεγράφηκε στην προηγούμενη υποενότητα. Το αποτέλεσμα της κωδικοποίησης θα είναι αποθηκευμένο στον πίνακα **output**.

3.1.7 Υπολογισμός μέσου λόγου συμπίεσης των εικόνων A και B

Για τον υπολογισμό και την αποθήκευση του μέσου λόγου συμπίεσης των δύο εικόνων, απαραίτητη προϋπόθεση είναι η καταμέτρηση των bits της κάθε εικόνας, δηλαδή να δούμε το σύνολο των 0 και 1 σε κάθε μια από αυτές. Τα αποτελέσματα θα αποθηκεύονται σε έναν πίνακα κάθε φορά που μια εικόνα A και B υπόκειται στις παραπάνω διαδικασίες. Ο λόγος συμπίεσης είναι το αποτέλεσμα της διαίρεσης του πλήθους των bits της εικόνας A προς τα αντίστοιχο άθροισμα της εικόνας B. Ο μέσος λόγος, ο οποίος είναι και το ζητούμενο της άσκησης, υπολογίζεται ως το αποτέλεσμα της διαίρεσης μεταξύ του αθροίσματος των λόγων συμπίεσης προς το μήκος αυτού, δηλαδή: **sum(logos_simpiesis)/len(logos_simpiesis)**.

4. Αποτελέσματα

Παρακάτω παρατίθενται τα αποτελέσματα της εκτέλεσης του κώδικα για τους πενταψήφιους αριθμούς μητρώου: 17007 και 17112.

```
Give ID of students in a team (min:1 and max:3 students per team)
ID of first student: 17007
Do you want to stop? (y/n): n
ID of second student: 17112
Do you want to stop? (y/n): y
Students' ID for this group: ['17007', '17112']
Students' ID for this group after the change of one digit with the number 5: ['17005', '15112']
```

Εικόνα 29- Αποτέλεσμα μετά την αντικατάσταση ενός τυχαίου ψηφίου με το 5

```
Average compression ratio: 1.5402747198637063
```

Εικόνα 30- Μέσος λόγος συμπίεσης των εικόνων Α και Β

5.Συμπεράσματα

Στα παραπάνω αποτελέσματα παρατηρούμε ότι ο μέσος λόγος συμπίεσης είναι 1.5402747198637063 . Το αποτέλεσμα αυτό μεταφράζεται ότι στις 100 εικόνες που δημιουργήθηκαν οι Β εικόνες, στις οποίες χρησιμοποιήθηκε η τεχνική που περιγράφεται αναλυτικά στη σελίδα 226 του βιβλίου είναι πιο αποτελεσματική από τη συμπίεση μιας εικόνας μόνο με την τεχνική του Huffman.

6.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:

- ❖ **numpy**
- ❖ **opencv-python(cv2)**
- ❖ **random**
- ❖ **huffman**
- ❖ **collections**

Ο χρήστης κατά την εισαγωγή των πενταψήφιων αριθμών μητρώων θα πρέπει να είναι προσεχτικός σε δύο πράγματα. Το πρώτο είναι να εισάγει μόνο αριθμούς και όχι κάποιον άλλον χαρακτήρα ή τελεστή και το δεύτερο να είναι προσεχτικός με το πλήθος των αριθμών αυτών, οι οποίοι θα πρέπει να είναι από 1 έως και 3.

Η επίλυση της άσκησης βρίσκεται στον φάκελο **Β' Μέρος** και συγκεκριμένα στα αρχείο **merosb.py**.

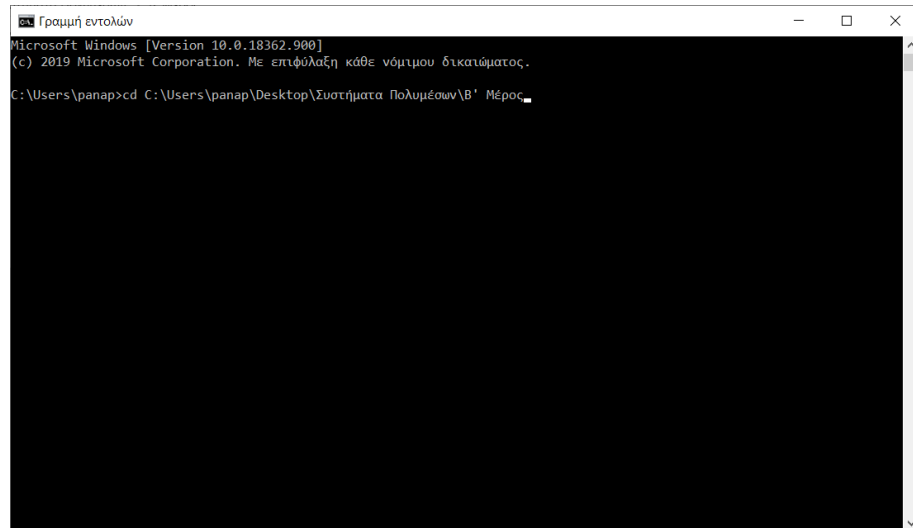
Για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

1. Ανοίγουμε τη γραμμή εντολών και μεταβαίνουμε στον αντίστοιχο φάκελο: **\Συστήματα Πολυμέσων\Β' Μέρος.(Εικόνα 31)**
2. Πληκτρολογούμε το όνομα του αρχείου **merosb.py**, ώστε να ανοίξει το εκτελέσιμο.(**Εικόνα 32**)

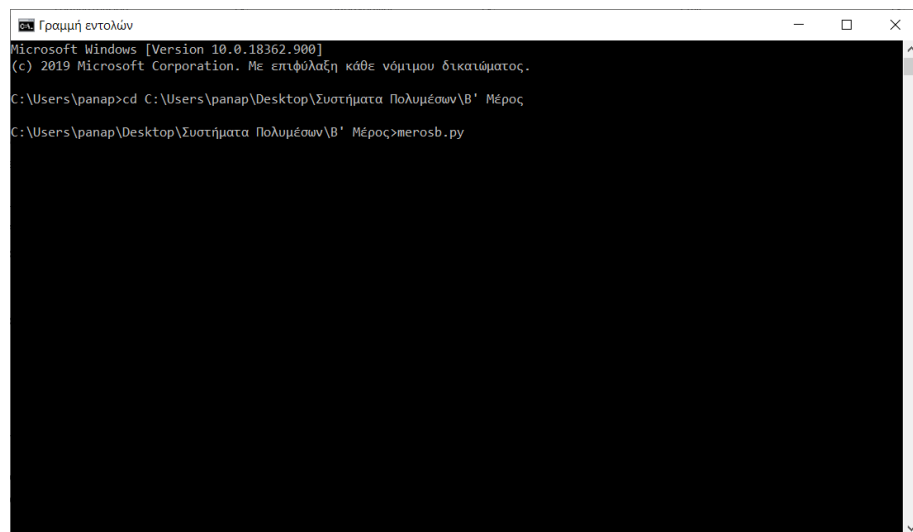
Προκειμένου να δημιουργηθούν οι δύο εικόνες και να υπολογιστεί ο μέσος λόγος συμπίεσης θα πρέπει ο χρήστης να ακολουθήσει τα παρακάτω βήματα:

1. Εισάγει τον πενταψήφιο αριθμό σύμφωνα με τους περιορισμούς που αναφέρθηκαν παραπάνω(**Εικόνα 33**).
2. Αν επιθυμεί την εισαγωγή και δεύτερου ή τρίτου θα πρέπει στην σχετική ερώτηση να πληκτρολογήσει τον χαρακτήρα **n** (**Εικόνα 34**). Σε αντίθετη περίπτωση, θα πρέπει να εισάγει τον χαρακτήρα **y** (**Εικόνα 35**).

3. Μόλις τελειώσει ολόκληρη διαδικασία για να κλείσει το πρόγραμμα θα πρέπει να «πατήσει» οποιοδήποτε πλήκτρο(**Εικόνα 36**)



Εικόνα 31- Μετάβαση στο φάκελο, στον οποίο βρίσκεται το αρχείο του προγράμματος



Εικόνα 32-Εισαγωγή του ονόματος αρχείου προς εκτέλεση

```
Γραμμή εντολών - merosb.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος>merosb.py

Give ID of students in a team (min:1 and max:3 students per team)

ID of first student: 17007_
```

Εικόνα 33- Εισαγωγή του πρώτου πενταψήφιου

```
Γραμμή εντολών - merosb.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος>merosb.py

Give ID of students in a team (min:1 and max:3 students per team)

ID of first student: 17007
Do you want to stop? (y/n): n
ID of second student: 17112
```

Εικόνα 34- Επιλογή για εισαγωγή νέου αριθμού

```
Γραμμή εντολών - merosb.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος>merosb.py

Give ID of students in a team (min:1 and max:3 students per team)

ID of first student: 17007

Do you want to stop? (y/n): n

ID of second student: 17112

Do you want to stop? (y/n): y
```

Εικόνα 35- Επιλογή για τερματισμό της διαδικασίας εισαγωγής πενταψήφιου αριθμού

```
Γραμμή εντολών - merosb.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\panap>cd C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος

C:\Users\panap\Desktop\Συστήματα Πολυμέσων\B' Μέρος>merosb.py

Give ID of students in a team (min:1 and max:3 students per team)

ID of first student: 17007

Do you want to stop? (y/n): n

ID of second student: 17112

Do you want to stop? (y/n): y

Students' ID for this group: ['17007', '17112']

Students' ID for this group after the change of one digit with the number 5: ['17507', '17512']

Average compression ratio: 1.4934634871511714

Press any button to exit the program...■
```

Εικόνα 36- Έξοδος από το πρόγραμμα με την εισαγωγή κάποιου χαρακτήρα

Βιβλιογραφία

1. «Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές», Parag Havaladar, Gerard Medioni - Επιμέλεια Ελληνικής έκδοσης Άγγελος Πικράκης, Ph.D
2. Σημειώσεις μαθήματος «Συστήματα Πολυμέσων», Διδάσκων Άγγελος Πικράκης, Ph.D
3. <https://docs.scipy.org/doc/> (τελευταία προσπέλαση 22/06/2020)
4. <https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/> (τελευταία προσπέλαση 23/06/2020)
5. <https://www.pyimagesearch.com/2017/01/09/count-the-total-number-of-frames-in-a-video-with-opencv-and-python/> (τελευταία προσπέλαση 29/06/2020)
6. <https://pypi.org/project/huffman/> (τελευταία προσπέλαση 06/07/2020)