Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ακαδημαϊκό Έτος: 2019-2020

ΟΡΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

2η Εργαστηριακή Άσκηση

Θέμα: Εξαγωγή Χαρακτηριστικών σε Βίντεο για Αναγνώριση Δράσεων

Τζε Χριστίνα-Ουρανία | 03116079
Ψαρουδάκης Ανδρέας | 03116001

30 Μαΐου 2020

Περιεχόμενα

2.1 Χωρο-χρονικά Σημεία Ενδιαφέροντος 2
2.1.1 2
2.1.2 3
2.1.3 4
2.2 Χωρο-χρονικοί Ιστογραφικοί Περιγραφητές 7
2.2.1 8
2.2.2 8
2.2.3 8
2.3 Κατασκευή Bag of Visual Words και χρήση Support Vector Machines για την ταξινόμηση δράσεων 9
2.3.1 9
2.3.2 10
2.3.3 10
2.3.4 10
2.3.5 11

Εισαγωγή

Σκοπός του δεύτερου μέρος της παρούσας εργαστηριακής άσκησης είναι η εξαγωγή χωρο-χρονικών χαρακτηριστικών με στόχο την εφαρμογή τους στο πρόβλημα κατηγοριοποίησης βίντεο που περιέχουν ανθρώπινες δράσεις. Αρχικά, γίνεται με την χρήση ανιχνευτών τοπικών χαρακτηριστικών αναζήτηση χωρο-χρονικών σημείων και κλιμάκων ενδιαφέροντος, τα οποία αντιστοιχούν σε περιοχές που χαρακτηρίζονται από σύνθετη κίνηση ή απότομες μεταβολές στην εμφάνιση του video εισόδου. Στη συνέχεια, πραγματοποιείται εξαγωγή χωρο-χρονικών ιστογραφικών περιγραφητών γύρω από τα ανιχνευθέντα σημεία ενδιαφέροντος και κατηγοριοποίηση των βίντεο σε κατηγορίες με βάση BoVW αναπαραστάσεις.

Μέρος 2: Εντοπισμός Χωρο-χρονικών Σημείων Ενδιαφέροντος και Εξαγωγή Χαρακτηριστικών σε Βίντεο Ανθρωπίνων Δράσεων

2.1 Χωρο-χρονικά Σημεία Ενδιαφέροντος

Πρώτο βήμα για την κατηγοριοποίηση των βίντεο αποτελεί η ανίχνευση κρίσιμων σημείων. Ως κρίσιμα σημεία θεωρούμε εκείνα τα οποία μεγιστοποιούν κάποιο κριτήριο 'οπτικής σημαντικότητας'.

Αρχικά, ξεκινάμε διαβάζοντας το βίντεο. Αυτό γίνεται με χρήση της συνάρτησης **read_video** που μας δίνεται. Συγκεκριμένα, διαβάζουμε τα πρώτα 200 frames τα οποία αποθηκεύουμε σε έναν τρισδιάστατο πίνακα του οποίου η 3η διάσταση αντιστοιχεί στον χρόνο και αποτελεί την ακολουθία των frames, τα οποία είναι grayscale εικόνες. Αφού τα διαβάσουμε μετατρέπουμε τις τιμές τους σε float και τα κανονικοποιούμε διαιρώντας με το 255.

2.1.1 Υλοποιούμε στην συνέχεια τον ανιχνευτή Harris Detector ο οποίος αποτελεί μία επέκταση στις 3 διαστάσεις του ανιχνευτή γωνιών Harris-Stephens.

Θεωρητικό υπόβαθρο

Για κάθε voxel (x,y,t) του βίντεο υπολογίζουμε τον 3×3 πίνακα $\mathrm{M}(x,y,t)$ ο οποίος δίνεται από την σχέση:

$$M(x, y, t; \sigma, \tau) = g(x, y, t; s\sigma, s\tau) * (\nabla L(x, y, t; \sigma, \tau)(\nabla L(x, y, t; \sigma, \tau))^T)$$

ή σε μορφή πινάκων:

$$M(x, y, t; \sigma, \tau) = g(x, y, t; s\sigma, s\tau) * \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix} \quad (1)$$

όπου $g(x, y, t; s\sigma, s\tau)$ ένας 3Δ γκαουσιανός πυρήνας ομαλοποίησης και $\nabla L(x, y, t; \sigma, \tau)$ οι χωρο-χρονικές παράγωγοι της ομαλοποιημένης εικόνας $L(x, y, t)$. Η εικόνα $L(x, y, t)$ προκύπτει από συνέλιξη των frames του βίντεο με μία 3Δ γκαουσιανή χωρικής και χρονικής απόκλισης σ και τ αντίστοιχα. Το κριτήριο 'οπτικής σημαντικότητας' δίνεται από την ακόλουθη σχέση:

$$H(x, y, t) = det(M(x, y, t)) - k \cdot trace^3(M(x, y, t))$$

Υλοποίηση αλγορίθμου

Η υλοποίηση του ανιχνευτή στις 3 διαστάσεις βρίσκεται εντός της συνάρτησης με όνομα **HarrisDetector** η οποία δέχεται για ορίσματα τα frames του βίντεο, τις παραμέτρους σ, τ της γκαουσιανής, την κλίμακα s και την παράμετρο k. Για την συνέλιξη των frames με την 3Δ γκαουσιανή πραγματοποιούμε τρείς διαδοχικές συνελίξεις με μονοδιάστατους γκαουσιανούς πυρήνες, έναν σε κάθε διάσταση. Αυτό γίνεται με την συνάρτηση **convolve1d** της scipy.ndimage δίνοντας κάθε φορά το κατάλληλο axis προκειμένου να πραγματοποιηθεί η συνέλιξη στην σωστή διάσταση. Προηγείται ο ορισμός με την **getGaussianKernel** της cv2 των δύο μονοδιάστατων γκαουσιανών πυρήνων μεγέθους $n \times n$, όπου $n = ceil(3 \cdot \sigma) \cdot 2 + 1$ για την χωρική διάσταση και $n = ceil(3 \cdot \tau) \cdot 2 + 1$ για την χρονική. Στη συνέχεια υπολογίζουμε τις μερικές παραγώγους ως προς x,y και t της ομαλοποιημένης ακολουθίας των frames, $L(x, y, t)$. Για τον υπολογισμό των παραγώγων (χωρικών και χρονικών) εφαρμόζουμε συνέλιξη (convolve1d) με τον πυρήνα κεντρικών διαφορών $[-1, 0, 1]$ προσαρμοσμένο στην κατάλληλη διάσταση. Όπως φαίνεται και από την σχέση (1) για τον υπολογισμό του πίνακα M πραγματοποιούμε συνέλιξη των επιμέρους στοιχείων του πίνακα των μερικών παραγώγων με την 3Δ γκαουσιανή $g(x, y, t; s\sigma, s\tau)$. Ορίζουμε πάλι δύο μονοδιάστατους γκαουσιανούς πυρήνες (getGaussianKernel) μεγέθους $n \times n$, όπου τώρα $n = ceil(3 \cdot s\sigma) \cdot 2 + 1$ για την χωρική διάσταση και $n = ceil(3 \cdot s\tau) \cdot 2 + 1$ για την χρονική. Οι συνελίξεις με τον 3Δ γκαουσιανό πυρήνα αντικαθιστώνται πάλι, για λόγους μείωσης της υπολογιστικής πολυπλοκότητας, με τρείς διαδοχικές συνελίξεις με μονοδιάστατους πυρήνες στην κατάλληλη διάσταση. Έχοντας πλέον ορίσει τα στοιχεία του πίνακα M υπολογίζουμε αναλυτικά την ορίζουσά του καθώς επίσης και το ίχνος του. Το κριτήριο 'οπτικής σημαντικότητας' δίνεται τώρα από την σχέση:

$$H(x, y, t) = det(M(x, y, t)) - k \cdot trace^3(M(x, y, t))$$

2.1.2 Στην συνέχεια υλοποιούμε τον ανιχνευτή Gabor. Αυτός βασίζεται στο χρονικό φιλτράρισμα του βίντεο με ένα ζεύγος Gabor φίλτρων αφού πρώτα αυτό έχει υποστεί εξομάλυνση στις χωρικές διαστάσεις μέσω ενός 2Δ γκαουσιανού πυρήνα $g(x, y; \sigma)$ με τυπική απόκλιση σ.

Θεωρητικό υπόβαθρο

Τα Gabor φίλτρα ορίζονται ως εξής:

$$h_{ev}(t;\tau,w) = cos(2\pi tw)exp(-t^2/2\tau^2) \text{ και } h_{od}(t;\tau,w) = sin(2\pi tw)exp(-t^2/2\tau^2)$$

Η συχνότητα ω του Gabor φίλτρου συνδέεται με την χρονική κλίμακα τ (απόκλιση της γκαουσιανής συνιστώσας του) μέσω της σχέσης: $\omega = 4/\tau$. Το κριτήριο σημαντικότητας προκύπτει παίρνοντας την τετραγωνική ενέργεια της εξόδου για το ζεύγος Gabor φίλτρων:

$$H(x,y,t) = (I(x,y,t) * g * h_{ev})^2 + (I(x,y,t) * g * h_{od}^2)$$

Υλοποίηση αλγορίθμου

Η υλοποίηση του ανιχνευτή Gabor βρίσκεται στην συνάρτηση **GaborDetector** η οποία δέχεται για ορίσματα τα frames του βίντεο, τις παραμέτρους σ και τ. Αρχικά, ορίζουμε μέσω της **getGaussianKernel** έναν μονοδιάστατο γκαουσιανό πυρήνα τυπικής απόκλισης σ. Στην συνέχεια πραγματοποιούμε με την convolve1d δύο διαδοχικές συνελίξεις των frames του βίντεο, μία στο επίπεδο x (axis=0) και μία στο y (axis=1), με τον μονοδιάστατο πυρήνα που ορίσαμε προηγουμένως. Για τον υπολογισμό της κρουστικής απόκρισης των Gabor θεωρούμε μέγεθος παραθύρου [−2τ,2τ] το οποίο ορίζουμε με χρήση της συνάρτησης **linspace** της βιβλιοθήκης numpy. Αφού βρούμε από τον αναλυτικό τύπο ορισμού τους τα φίλτρα h_{ev} και h_{od} τα κανονικοποιούμε διαιρώντας με την L1 νόρμα τους. Η τελευταία υπολογίζεται με την συνάρτηση **norm** της np.linalg. Τέλος, πραγματοποιούμε συνέλιξη προσαρμοσμένη στην κατάλληλη διάσταση (convolve1d και axis=2) των εξομαλυμένων στις χωρικές διαστάσεις frames του βίντεο με τα Gabor φίλτρα, οπότε προκύπτει τελικά το κριτήριο οπτικής σημαντικότητας ως εξής:

$$H(x,y,t) = (I(x,y,t) * g * h_{ev})^2 + (I(x,y,t) * g * h_{od}^2)$$

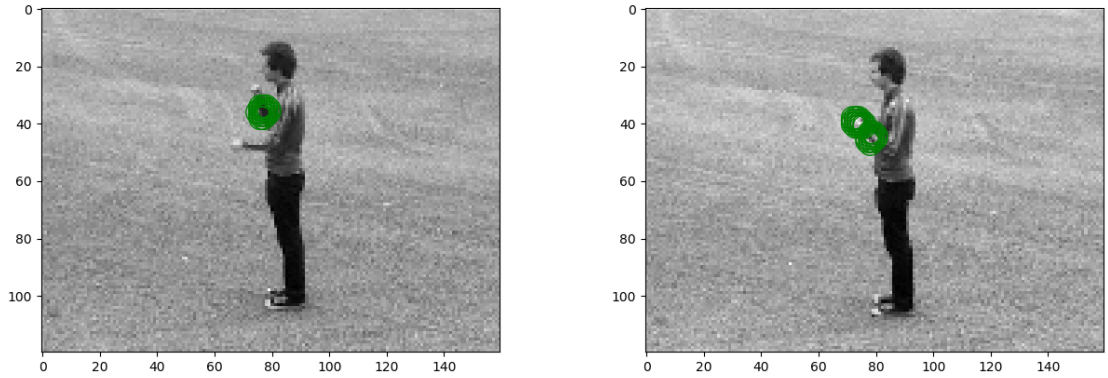
2.1.3 Για κάθε ανιχνευτή υπολογίζουμε τώρα τα σημεία ενδιαφέροντος σαν τα τοπικά μέγιστα του κριτηρίου σημαντικότητας. Για λόγους απλότητας επιστρέφουμε τα πρώτα 500 με τις μεγαλύτερες τιμές του κριτηρίου. Προκειμένου να μπορέσουμε να απεικονίσουμε τα ανιχνευθέντα σημεία με την έτοιμη συνάρτηση **show_detection**, αυτά πρέπει να είναι στην μορφή ενός πίνακα $N \times 4$ του οποίου οι τρείς πρώτες στήλες αντιστοιχούν στις συντεταγμένες τους (x,y,t), όπου t το frame στο οποίο ανιχνεύθηκαν, και η τέταρτη στην κλίμακα σ στην οποία ανιχνεύθηκαν. Όλα τα παραπάνω υλοποιούνται στις παρακάτω γραμμές κώδικα:

```
sorted_indx = H.flatten().argsort()[::-1][:500]
dim_idx = np.unravel_index(sorted_indx, H.shape)
x_coord = dim_idx[1].reshape(1,500)
y_coord = dim_idx[0].reshape(1,500)
```

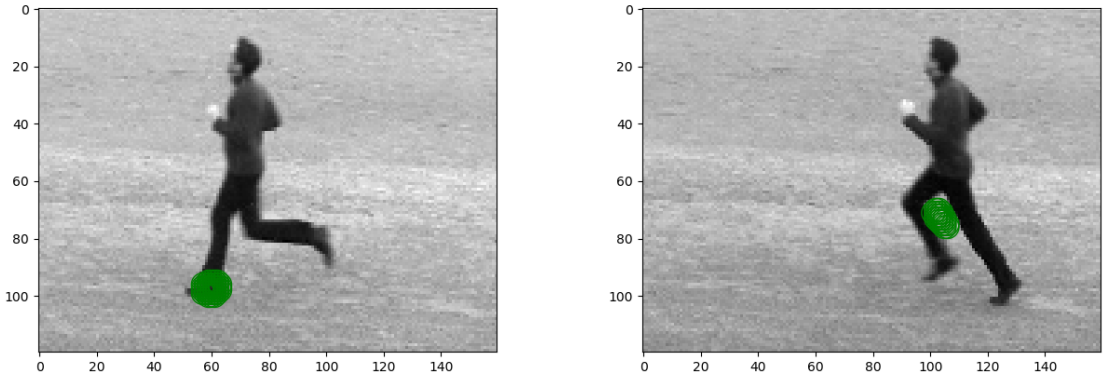
```
t_coord = dim_idx[2].reshape(1,500)
xyt_coord = np.concatenate((x_coord,y_coord,t_coord),axis=0)
xyt_coord = xyt_coord.T

scale = np.ones((xyt_coord.shape[0],1))*sigma
voxels_scale = np.concatenate((xyt_coord, scale), axis=1)
```

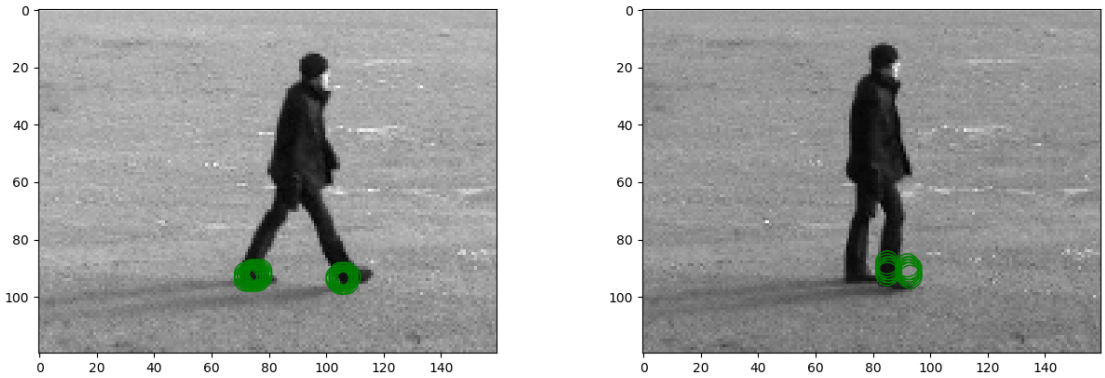
Αρχικά, μετατρέπουμε τον πίνακα H μεγέθους (n,m) (όπου n ο αριθμός των γραμμών και m ο αριθμός των στηλών των frames του βίντεο) του κριτηρίου οπτικής σημαντικότητας σε διάνυσμα μεγέθους $(nm,)$. Αυτό γίνεται με χρήση της συνάρτησης **flatten**. Στην συνέχεια μέσω της **argsort** βρίσκουμε σε μορφή λίστας τα indexes τα οποία αντιστοιχούν με την σειρά στις τιμές της flattened εκδοχής του κριτηρίου H από τις μικρότερες στις μεγαλύτερες. Επειδή εμάς μας ενδιαφέρουν οι πρώτες N (πχ. 500) μεγαλύτερες τιμές του H, αντιστρέφουμε την λίστα ([::-1]) και κρατάμε τα πρώτα 500 στοιχεία της. Για να βρούμε τα αντίστοιχα indexes για τον πίνακα H χρησιμοποιούμε την συνάρτηση **unravel_index** με ορίσματα τα flattened indexes και το μέγεθος του H. Επειδή τα *dim_idx* που επιστέφει η unravel_index είναι 'array like' ενώ η show_detection δέχεται τα ορίσματά της σε μορφή 'axis like', δηλαδή το x αντιστοιχεί στην στήλη και το y στην γραμμή του εκάστοτε σημείου, οι συντεταγμένες x, y, t προκύπτουν αντίστοιχα από τα dim_idx[1], dim_idx[0] και dim_idx[2]. Η εντολή **reshape**(1,500) γίνεται προκειμένου να μετατρέψουμε x_coord, y_coord και t_coord από διανύσματα μεγέθους (500,) σε πίνακες διαστάσεων (1, 500). Για να δημιουργήσουμε τον $N \times 3$ πίνακα με τις συντεταγμένες (x,y,t) των ανιχνευθέντων σημείων συνενώνουμε τους επιμέρους πίνακες με χρήση της συνάρτησης **concatenate** της numpy και παίρνουμε τον ανίστροφο του πίνακα που προκύπτει. Προκειμένου να προσθέσουμε σαν τέταρτη στήλη στον $N \times 3$ πίνακα την κλίμακα σ, κατασκευάζουμε τον πίνακα scale μίας στήλης και γραμμών όσο ο αριθμός των ανιχνευθέντων σημείων. Ο $N \times 4$ πίνακας προκύπτει τελικά από συνένωση των δύο επιμέρους πινάκων, του $N \times 3$ πίνακα των συντεταγμένων και του $N \times 1$ πίνακα των κλιμάκων.



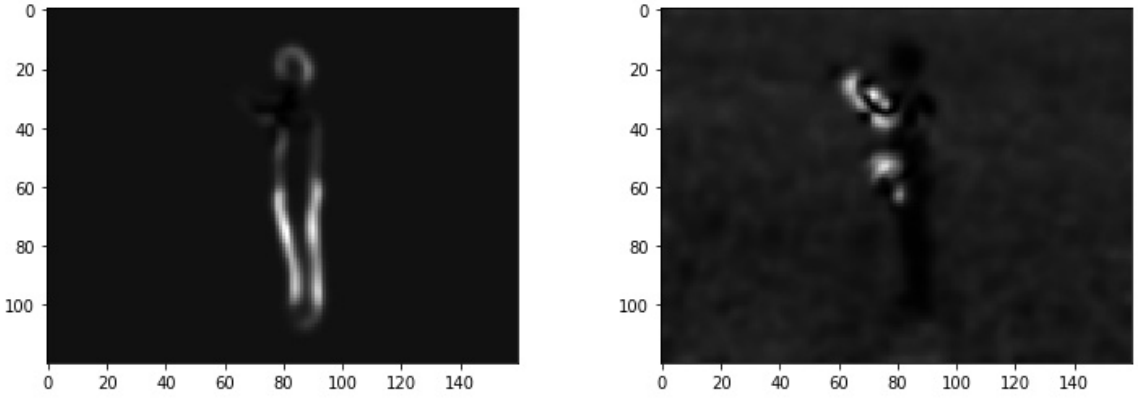
Σχήμα 1: Σημεία ενδιαφέροντος για την την κατηγορία 'boxing' με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)



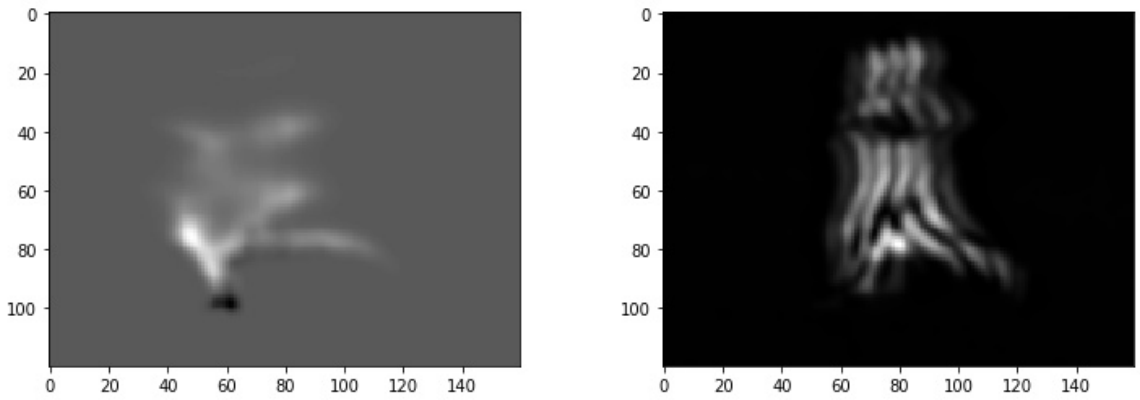
Σχήμα 2: Σημεία ενδιαφέροντος για την την κατηγορία ΄running΄ με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)



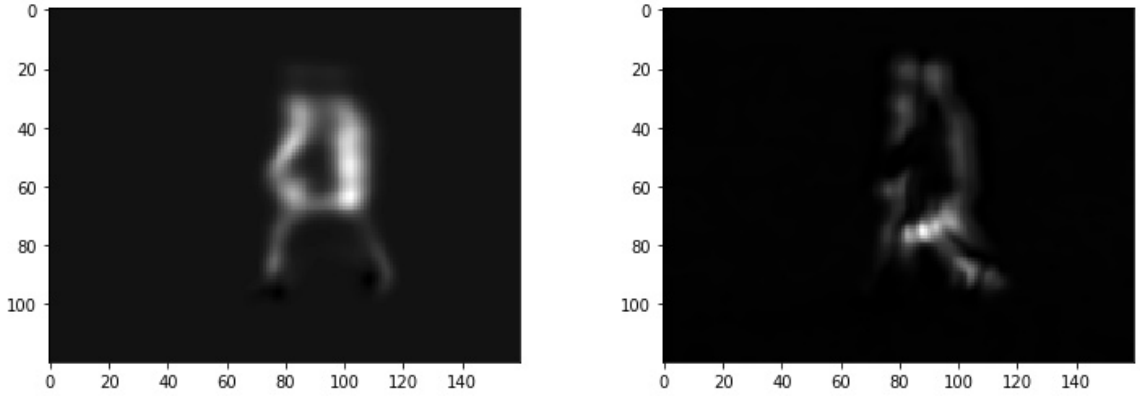
Σχήμα 3: Σημεία ενδιαφέροντος για την την κατηγορία ΄walking΄ με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)



Σχήμα 4: Κριτήριο H για την την κατηγορία ΄boxing΄ με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)



Σχήμα 5: Κριτήριο H για την κατηγορία ΄running΄ με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)



Σχήμα 6: Κριτήριο H για την κατηγορία ΄walking΄ με βάση τον ανιχνευτή Harris (αριστερά) και Gabor (δεξιά)

Γενικά και για τις δύο μεθόδους λαμβάνουμε ικανοποιητικά αποτελέσματα καθώς τα σημεία που ανιχνεύουν είναι κάθε φορά αντιπροσωπευτικά της δράσης που απεικονίζεται. Παραδείγματος χάριν στην περίπτωση του running και walking τα σημεία που λαμβάνουμε είναι κοντά στα πόδια ενώ στην περίπτωση του boxing είναι κοντά στα χέρια. Αναφορικά με τον ανιχνευτή Gabor αυτός φαίνεται να παράγει καλύτερα και πιο σωστά αποτελέσματα για αργές δράσεις. Αντίθετα για γρήγορες δράσεις όπως πχ. το τρέξιμο ο ανιχνευτής Gabor βρίσκει πολλά και όχι πάντα σωστά σημεία ενδιαφέροντος. Το γεγονός αυτό σχετίζεται με το κριτήριο οπτικής σημαντικότητας που χρησιμοποιεί ο εν λόγω ανιχνευτής το οποίο βασίζεται στην ενέργεια των κινήσεων. Τέλος, όπως φαίνεται και στις παραπάνω εικόνες και για τους δύο ανιχνευτές συμβαίνει να συσσωρεύονται τα σημεία που εντοπίζονται χωρικά και χρονικά, χωρίς ωστόσο αυτό να μειώνει την αποτελεσματικότητά τους.

2.2 Χωρο-χρονικοί Ιστογραφικοί Περιγραφητές

Για την κατηγοριοποίηση των βίντεο με βάση τα ανιχνευθέντα σημεία υλοποιούμε ιστογραφικούς περιγραφητές βασισμένους στον υπολογισμό ιστογραμμάτων της κατευθυντικής παραγώγου (HOG) και της οπτικής ροής (HOF - Histogram of Oriented Flow).

2.2.1 Για τον περιγραφητή HOG ξεκινάμε υπολογίζοντας για κάθε frame του βίντεο στο οποίο ανιχνεύθηκαν σημεία ενδιαφέροντος τις κατευθυντικές παραγώγους. Αυτές υπολογίζονται εφαρμόζοντας συνέλιξη με τον πυρήνα κεντρικών διαφορών [-1, 0, 1] προσαρμοσμένο στην κατάλληλη διάσταση. Για τον σκοπό αυτό χρησιμοποιούμε την συνάρτηση **convolve1d** με το κατάλληλο κάθε φορά axis έτσι ώστε να πραγματοποιηθεί η συνέλιξη στην σωστή διάσταση. Στην περίπτωση του περιγραφητή HOF υπολογίζουμε με βάση τον αλγόριθμο του Lucas-Kanade που υλοποίησαμε στο πρώτο μέρος της δεύτερης εργαστηριακής άσκησης το διάνυσμα της οπτικής ροής για κάθε frame του βίντεο στο οποίο εντοπίστηκε σημείο ενδιαφέροντος και του επόμενού του. Αν ένα σημείο εντοπιστεί στο τελευταίο πλαίσιο της αρχικής ακολουθίας των εικόνων του βίντεο τότε το αγνοούμε και συνεχίζουμε με τα υπόλοιπα. Αφού βρούμε τα διανύσματα κλίσης (gradient για τον HOG και οπτικής ροής για τον HOF) εξάγουμε τα διανυσματικά πεδία για μία τετραγωνική περιοχή $4\times\sigma$ εκατέρωθεν του εκάστοτε σημείου ενδιαφέροντος. Δίνουμε ιδιαίτερη προσοχή στα όρια της εικόνας έτσι ώστε να περιορίζουμε πάντοτε το bounding box εντός αυτής. Όλα τα παραπάνω υλοποιούνται εντός των συναρτήσεων **HOG** και **HOF** οι οποίες δέχονται για ορίσματα τα frames του βίντεο και τον $N \times 4$ πίνακα του προηγούμενου βήματος. Επιπλέον δημιουργούμε μια συνάρτηση με όνομα **CreateDescriptors** η οποία δέχεται τα ίδια ορίσματα με τις δύο προηγούμενες και μία επιπλέον συμβολοσειρά η οποία καθορίζει ποιόν από τους 3 περιγραφητές (HOG,HOF,HOG/HOF) θα υλοποιήσουμε. Σημειώνουμε πως για τη δημιουργία του HOG/HOF περιγραφητή συνενώνουμε απλά τους δύο επιμέρους περιγραφητές.

2.2.2 Αφού εξάγουμε το διανυσματικό πεδίο G_x, G_y χρησιμοποιούμε την συνάρτηση **orientation_histogram** που μας δίνεται προκειμένου να υπολογίσουμε τους δύο ιστογραφικούς περιγραφητές. Η συνάρτηση αυτή δέχεται ως είσοδο το διανυσματικό πεδίο (κατευθυντικές παραγώγους είτε κατεύθυνση ροής), το μέγεθος $n \times m$ του grid και το πλήθος, $nbins$, των bins. Επιπλέον επιστρέφει ένα ιστόγραμμα για την περιοχή του διανυσματικού πεδίου εισόδου το οποίο αποθηκεύουμε κάθε φορά σε έναν πίνακα desc μεγέθους $N \times (n \times m \times nbins)$, όπου N το πλήθος των σημείων ενδιαφέροντος.

2.2.3 Κατασκευή ιστογραμμάτων (Bonus Ερώτημα)
Στο βήμα αυτό κατασκευάζουμε μόνοι μας τα ιστογράμματα των περιγραφητών που μόλις υλοποιήσαμε. Η διαδικασία αυτή πραγματοποιείται μέσω της συνάρτησης **myHistogram**, η οποία δέχεται ως ορίσματα τα προσανατολισμένα διανυσματικά πεδία F_x και F_y, στην οριζόντια και κάθετη διεύθυνση αντίστοιχα, υπολογισμένα σε μια περιοχή γύρω από το εκάστοτε σημείο ενδιαφέροντος. Οι πίνακες F_x και F_y μπορεί να είναι είτε τετραγωνικοί, διαστάσεων $(8\sigma + 1) \times (8\sigma + 1)$, είτε και μη τετραγωνικοί στην περίπτωση που το σημείο ενδιαφέροντος βρίσκεται κοντά στα άκρα του αντίστοιχου frame. Στην περίπτωση που αυτό συμβεί, για λόγους απλότητας και ευκολίας, τους μετατρέπουμε εμείς σε τετραγωνικούς $(8\sigma + 1) \times (8\sigma + 1)$ επεκτείντονας τους με όσες μηδενικές γραμμές ή/και στήλες απαιτούνται. Στη συνέχεια, υπολογίζουμε τόσο το μέτρο όσο και τη γωνία του διανυσματικού πεδίου αξιοποιώντας την

συνάρτηση **cartToPolar** της βιβλιοθήκης cv2. Λαμβάνουμε λοιπόν ένα πίνακα mangitude και ένα πίνακα phase. Επειδή θέλουμε οι γωνίες μας να ανήκουν στο εύρος $[0, 180]$ και όχι στο $[0, 360]$ (μη-προσημασμένα gradients) , όπως τις επιστρέφει η συνάρτηση, εκτελούμε το υπόλοιπο της ακέραιας διαίρεσης του πίνακα phase με το 180. Στη συνέχεια, πρέπει να χωρίσουμε τον τετραγωνικό πίνακα $(8\sigma + 1) \times (8\sigma + 1)$ σε υποπεριοχές που ονομάζονται κελιά. Κάθε κελί επιλέγουμε να έχει μέγεθος 5×5. Επειδή δεν είναι εφικτός ακέραιος αριθμός κελιών χωρίς επικάλυψη, επιλέγουμε να έχουμε τα πρώτα 6 κελιά κάθε γραμμής/στήλης χωρίς overlap ενώ το τελευταίο να επικαλύπτεται κατά 2 με το αμέσως προηγούμενό του. Με βάση τον παραπάνω διαχωρισμό προκύπτουν συνολικά $7 \times 7 = 49$ κελιά για $\sigma = 4$. Τώρα πρέπει να υπολογίσουμε το τοπικό ιστόγραμμα που αντιστοιχεί σε κάθε κελί. Για το σκοπό αυτό το εύρος γωνιών $0° - 180°$ χωρίζεται ομοιόμορφα σε 9 τμήματα, ώστε κάθε ένα από αυτά να έχει εύρος $20°$. Κάθε τμήμα αντιστοιχεί σε μία ράβδο (bin) του τοπικού ιστογράμματος που κατασκευάζουμε. Κάθε pixel εντός του κελιού 'ψηφίζει' για την κατασκευή του ιστογράμματος ως εξής:

- Η 'ψήφος' δίνεται στην ράβδο στην οποία αντιστοιχεί η γωνία του gradient.
- Η τιμή της 'ψήφου' ισούται με το μέτρο του gradient.

Έπειτα διαιρούμε κάθε ιστόγραμμα που κατασκευάζουμε με την l_2 νόρμα του, ώστε να το κανονικοποιήσουμε. Αυτό είναι σημαντικό ώστε ο περιγραφητής μας να είναι ανεξάρτητος από μεταβολές στην φωτεινότητα. Σαν τελικό βήμα, συνενώνουμε όλους τους περιγραφητές που έχουμε φτιάξει σε ένα ενιαίο καθολικό ιστόγραμμα.

2.3 Κατασκευή Bag of Visual Words και χρήση Support Vector Machines για την ταξινόμηση δράσεων

Στο βήμα αυτό γίνεται η κατηγοριοποίηση των βίντεο με τις ανθρώπινες δράσεις σε 3 κατηγορίες/κλάσεις (που η κάθε μία θα αντιπροσωπεύει ένα διαφορετικό είδος δράσης) με χρήση BoVW αναπαραστάσεων βασισμένων στα HOG / HOF χαρακτηριστικά. Το τελικό αποτέλεσμα είναι το ποσοστό επιτυχούς ταξινόμησης δράσεων, χρησιμοποιώντας SVM (Support Vector Machine) ταξινομητή.

2.3.1 Ξεκινάμε αρχικά διαχωρίζοντας το σύνολο των βίντεο σε σύνολο εκπάιδευσης (train set) και σύνολο δοκιμής (test set). Για τον διαχωρισμό αυτό συμβουλευόμαστε το αρχείο training_videos.txt που μας δίνεται στο συμπληρωματικό υλικό της άσκησης το οποίο περιέχει τα ονόματα των βίντεο που ανήκουν στο train set. Βάση αυτού δημιουργούμε το αντίστοιχο αρχείο test_videos.txt με τα ονόματα των βίντεο του test_set. Στη συνέχεια διαβάζουμε με χρήση της συνάρτησης **read_video** τα 200 πρώτα frames για κάθε training_video και τα κανονικοποιούμε διαιρώντας με το 255. Ανάλογα με την κατηγορία στην οποία ανήκουν τους αποδίδουμε το αντίστοιχο label, 0 για 'running', 1 για boxing και 2 για 'walking'. Τα labels όλων των training_video αποθηκεύονται στην λίστα train_labels. Επιπλέον για κάθε βίντεο και

για έναν δεδομένο συνδυασμό detector (Harris ή Gabor) και descriptor (HOG, HOF, HOG/HOF) βρίσκουμε το ιστόγραμμα (περιγραφητή) μεγέθους $n \times m \times nbins$ και το αποθηκεύουμε στη λίστα desc_train μήκους N_{train}. Ομοίως ακολουθούμε την ίδια ακριβώς διαδικασία και για τα βίντεο του test set.

2.3.2 Υπολογίζουμε τώρα την τελική αναπαράσταση (global representation) για κάθε βίντεο με την bag of visual words (BoVW) τεχνική χρησιμοποιώντας μόνο τα βίντεο εκπαίδευσης. Για τον υπολογισμό των BoVW ιστογραμμάτων χρησιμοποιούμε την έτοιμη συνάρτηση **bag_of_words** δίνοντας ως ορίσματα τις λίστες desc_train, desc_test των περιγραφητών των βίντεο εκπαίδευσης και δοκιμής αντίστοιχα και τον αριθμό D (πχ. 500-1000) των κεντροειδών του K-means και άρα των οπτικών λέξεων.

2.3.3 Το τελευταίο στάδιο συνίσταται στην τελική κατηγοριοποίηση των εικόνων με βάση την BoVW αναπαράσταση. Για την κατηγοριοποίηση χρησιμοποιείται ενάς SVM ταξινομητής κατάλληλα προσαρμοσμένος για πολλαπλές κλάσεις τον οποίο και εκπαιδεύουμε με χρήση της έτοιμης συνάρτησης **svm_train_test**. Η συνάρτηση αυτή επιστρέφει το αποτέλεσμα της αναγνώρισης (pred) καθώς και το συνολικό ποσοστό επιτυχίας (accuracy).

2.3.4 Πραγματοποιούμε μια διαδικασία μη-γραμμικής κατηγοριοποίησης χρησιμοποιώντας όλους τους διαφορετικούς συνδυασμούς ανιχνευτών/περιγραφητών. Τα ποσοστά επιτυχίας δεν παραμένουν σταθερά για κάθε χρήση του ταξινομητή svm, καθώς η αρχικοποίηση των κέντρων στον αλγόριθμο συσταδοποίησης k-means μεταβάλλεται. Για να έχουμε λοιπόν μια ενδεικτική εικόνα της ακρίβειας που παρέχει κάθε συνδυασμός ανιχνευτή και περιγραφητή, παρουσιάζουμε τη μέση τιμή των ποσοστών επιτυχίας για 5 διαδοχικές εκτελέσεις:

	Detector	
Descriptor	**Harris**	**Gabor**
HOG	81,667 %	86,667 %
HOF	88,333 %	88,333 %
HOG/HOF	**91,667 %**	81,667 %

Παρατηρούμε ότι το μεγαλύτερο ποσοστό επιτυχούς κατηγοριοποίησης προκύπτει για τον συνδυασμό του ανιχνευτή Harris με τον περιγραφητή HOG/HOF, όπου έχουμε **91,667 %** μέση ακρίβεια. Μάλιστα, για τον συνδυασμό αυτό, σε ορισμένες εκτελέσεις, λαμβάνουμε πλήρως επιτυχημένη ταξινόμηση, δηλαδή πραγματοποιείται σωστή πρόβλεψη για τις κατηγορίες και των 12 βίντεο του συνόλου δοκιμής test.

2.3.5 Σαν τελευταίο βήμα πειραματιζόμαστε με διαφορετικούς διαμερισμούς των δεδομένων σε train και test set και παρατηρούμε την επίδραση που έχουν στα αποτελέσματά μας. Αρχικά, αποθηκεύουμε στην λίστα video_list τα ονόματα όλων των βίντεο της βάσης μας. Ομοίως αποθηκεύουμε στις λίστες running_video_list, boxing_video_list και walking_video_list τα ονόματα των βίντεο των αντίστοιχων δράσεων. Στην συνέχεια, πραγματοποιούμε τον διαχωρισμό τους σε σύνολο εκπαίδευσης και σύνολο δοκιμής επιλέγοντας τυχαία, με χρήση της συνάρτησης **random.sample**, 36 βίντεο για το train set (12 από κάθε κατηγορία). Τα testing_videos προκύπτουν αφαιρώντας από την λίστα των 48 συνολικά βίντεο τα 36 του συνόλου εκπαίδευσης. Από εκεί και πέρα η διαδικασία που ακολουθείται για την εξαγωγή των αποτελεσμάτων κατηγοριοποίησης είναι η ίδια με αυτήν των προηγούμενων βημάτων. Το τελικό ποσοστό επιτυχούς ταξινόμησης προκύπτει από τον μέσο όρο των επιμέρους accuracies για 5 διαφορετικούς διαμερισμούς των δεδομένων.

	Detector	
Descriptor	**Harris**	**Gabor**
HOG	80 %	**86,667 %**
HOF	71,667 %	85 %
HOG/HOF	83,333 %	85 %

Για τον δικό μας διαμερισμό των δεδομένων σε train set και test set παρατηρούμε πως το καλύτερο αποτέλεσμα προκύπτει για τον συνδυασμό Gabor ανιχνευτή με HOG περιγραφητή, όπου έχουμε **86,667 %** μέση ακρίβεια. Εξίσου υψηλά είναι και τα ποσοστά για τους συνδυασμούς Gabor-HOF και Gabor-HOG/HOF. Συνεπώς στην περίπτωση αυτή ισχύει πως ο Gabor ανιχνευτής, συνδυασμένος με οποιονδήποτε περιγραφητή, δίνει καλύτερα αποτελέσματα από ότι ο ανιχνευτής Harris.