Κεφάλαιο 1

Αλγόριθμος σύνδεσης των action tubes

Στο προηγούμενο κεφάλαιο περιγράψαμε μεθόδους για την παραγωγή υποψήφιων ToIs, δεδομένου ενός μικρού τμήματος βίντεο που διαρκεί 8 ή 16 καρέ. Ωστόσο, τα πραγματικά βίντεο και πραγματικές ανθρώπινες ενέργειες, σε εξωτερικές συνθήκες, διαρκούν πάνω από 16 καρέ τις περισσότερες φορές. Τα τρέχοντα δίκτυα δεν είναι σε θέση να επεξεργαστούν ένα ολόκληρο βίντεο με την μία, προκειμένου να προτείνει υποψήφια ToIs, λόγω προβλημάτων μνήμης και υπολογιστικής ενέργειας. Πολλές προσεγγίσεις για εντοπισμό δράσης λύνουν αυτό το πρόβλημα, δεδομένου ενός βίντεο, είτε προτείνουν υποψήφιες περιοχές σε επίπεδο καρέ και, στη συνέχεια, τις συνδέουν με σκοπό τη δημιουργία υποψήφιων action tubes, είτε το διαχωρίζουν σε τμήματα βίντεο, προτείνοντας ακολουθίες από δυσδιάστα πλαίσια τα οποία στην συνέχεια συνδέουν για να δημιουργήσουν action proposals. Και οι δύο προαναφερθείσες προσεγγίσεις καθιστούν την κατάλληλη επιλογή της μεθόδου σύνδεσης σημαντικό παράγοντα για την απόδοση του δικτύου. Αυτό συμβαίνει επειδή, παρόλο που στο επίπεδο καρέ ή στο επίπεδο τμήματος βίντεο οι προτάσεις μπορεί να είναι πολύ καλές, αν ο προτεινόμενος αλγόριθμος σύνδεσης δεν λειτουργεί καλά, οι τελικές προτάσεις action tubes δεν θα είναι αποτελεσματικές, οπότε το τελικό μοντέλο δεν θα είναι σε θέση να επιτύχει υψηλή απόδοση ταξινόμησης. Με άλλα λόγια, αν ο αλγόριθμος σύνδεσης δεν δημιουργεί προτάσεις δράσης με μεγάλο recall και καλή απόδοση MABO, ο ταξινομητής του μοντέλου δεν θα είναι σε θέση να εκτελέσει την κατάλληλη ταξινόμηση, επειδή πιθανώς θα του έχουν δοθεί action tubes χωρίς κανένα περιεχόμενο. Σε αυτό το κεφάλαιο, παρουσιάζουμε 3 διαφορετικές προσεγγίσεις που χρησιμοποιούνται για τη σύνδεση των προτεινόμενων ToIς που παράγονται από το TPN του προηγούμενου κεφαλαίου.

1.1 Πρώτη προσέγγιση: συνδυασμός επικάλυψης και πιθανότας δράσης

Ο αλγόριθμος μας εμπνέεται από την προσέγγιση των Hou, Chen, and Shah 2017, η οποία υπολογίζει όλες τις πιθανές ακολουθίες των ToIs. Για να βρει τα καλύτερα υποψηφία action tubes, χρησιμοποιεί μια βαθμολογία που μας λέει πόσο πιθανό μια ακολουθία του TOIς είναι να περιέχει μια ενέργεια. Αυτή η βαθμολογία είναι ένας συνδυασμός 2 μετρικών:

Πιθανότητα δράσης ή Δραστικότητα(Actioness), που είναι η πιθανότητα ενός ToI να περιέχει μια δράση. Αυτό το σκορ παράγεται από τα scoring layers του TPN.

Σκορ επικάλυψης μεταξύ των ToIs, το οποίο είναι το IoU των τελευταίων πλασίων του πρώτου ToI και των πρώτων πλαισίων του δεύτερου ToIs.

Η παραπάνω πολιτική βαθμολόγησης μπορεί να περιγραφεί από τον ακόλουθο τύπο:

$$S = \frac{1}{m}\sum_{i=1}^{m} Actioness_i + \frac{1}{m-1}\sum_{j=1}^{m-1} Overlap_{j,j+1}$$

Για κάθε πιθανό συνδυασμό ToIs, υπολογίζουμε το σκορ του όπως φαίνεται στην εικόνα 1.1.

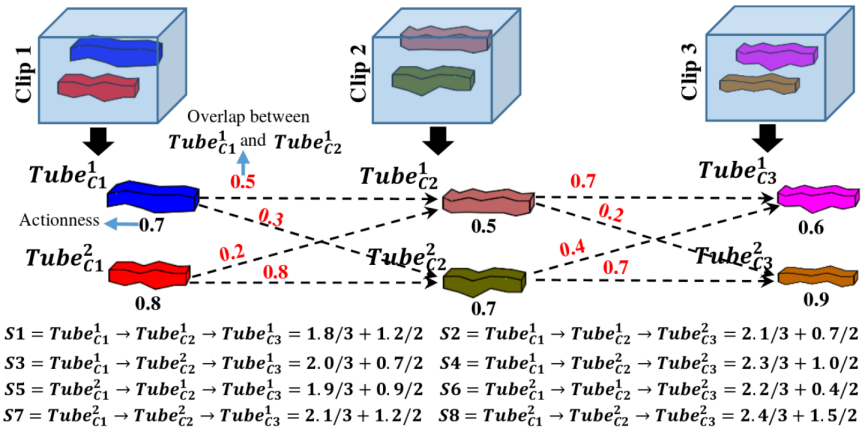

Figure 1.1: An example of calculating connection score for 3 random TOIs taken from Hou, Chen, and Shah 2017

Η παραπάνω προσέγγιση, όμως, χρειάζεται υπερβολικά πολύ μνήμη για την πραγματοποίηση όλων αυτών των υπολογισμων, έτσι ένα πρόβλημα μνήμης εμφανίζεται. Ο λόγος είναι πως για κάθε νέο βίντεο κλιπ, εμείς προτείνουμε k *ToIs* (16 κατά την διάρκεια της εκπαίδευσης και 150 κατά την διάρκεια του validation. Σαν

αποτέλεσμα, για ένα μικρό βίντεο χωρισμένο σε **10** μέρη, χρειάζεται να υπολογίσουμε 150^{10} σκορ κατά την διάρκεια της επικύρωσης. Αυτό οδηγεί το σύστημα μας να χρειάζεται υπερβολικά πολύ χρόνο για να το πραγματοποιήσει.

Για να αντιμετωπίσουμε αυτό το πρόβλημα, δημιουργούμε έναν άπληστο αλγόριθμο για να βρούμε τα υποψήφια action tubes. Ο αλγόριθμος αυτός, για κάθε νέο τμήμα βίντεο, κρατά τα tubes με βαθμολογία υψηλότερη από ένα κατώφλι και διαγράφει τα υπόλοιπα. Έτσι, δεν χρειάζεται να υπολογίσουμε συνδυασμούς με πολύ χαμηλό σκορ. Γράψαμε κώδικα για τον υπολογισμό των βαθμολογιών των tubes στη γλώσσα CUDA, η οποία έχει ως δυνατότητα την παράλληλη επεξεργασία του ίδιου κώδικα χρησιμοποιώντας διαφορετικά δεδομένα. Ο αλγόριθμος μας περιγράφεται παρακάτω:

1. Πρώτον, αρχικοποιούμε κενές λίστες για τα τελικά tubes,την διάρκεια τους, τις βαθμολογίες τους, τα ενεργά tubes, τη διάρκειά τους, το άθροισμα των σκορ επικάλυψης και δραστηκότητας τους όταν:

   - Η λίστα με τα τελικά tubes περιέχει όλα τα tubes που είναι πιθανότερο να περιέχουν μια ενέργεια και η λίστα βαθμολογίας τους περιέχει τις αντίστοιχες βαθμολογίες τους. Αναφερόμαστε σε κάθε tube από τον δείκτη του, ο οποίος σχετίζεται με ένα τένσορα, στον οποίο σώσαμε όλα τα ToIs που προτείνονται από το TPN για κάθε τμήμα βίντεο.
   - Η λίστα ενεργών tubes περιέχει όλα τα tubes που θα συνδιαστούν με τα νέα TOIs. Η λίστα άθροισης των επικαλυπτώμενων σκορ και η λίστα άθροισης δραστικότητας περιέχουν τα αντίστοιχα αθροίσματα τους, προκειμένου να αποφεύγεται ο υπολογισμός τους για κάθε βρόχο.

   Επίσης, προετοιμάζουμε το όριο σύνδεσης ίσο με 0,5.

2. Για το πρώτο τμήμα βίντεο, προσθέτουμε όλα τα ToIs τόσο στα ενεργά tubes όσο και στα τελικά tubes. Οι βαθμολογίες τους είναι μόνο η δική τους δραστηκότητας επειδή δεν υπάρχουν tubes για τον υπολογισμό της μεταξύ τους επικαλυπτόμενης βαθμολογίας. Έτσι, έτσι ορίζουμε το άθροισμα επικάλυψης ίσο με 0.

3. Για κάθε επόμενο βίντεο, μετά την λήψη των προτεινόμενων ToIs, πρώτα υπολογίζουμε το σκορ επικάλυψης τους με κάθε ενεργό tube. Μετά, αδειάζουμε την λίστα με τα ενεργά tubes, με την διάρκεια τους, το άθροισμα επικάλυψης και το άθροισμα πιθανοτήτων δράσης. Για κάθε νέο tube που έχει βαθμολογία υψηλότερη από το κατώφλι σύνδεσης προσθέτουμε τόσο στα τελικά action tubes όσο και στα ενεργά, στις αντίστοιχες λίστες και αυξάνουμε τη διάρκειά τους.

4. Εάν ο αριθμός των ενεργών tubes είναι μεγαλύτερος από ένα κατώτατο όριο, ορίζουμε το όριο σύνδεσης ίσο με τη βαθμολογία του 100ου καλύτερου tube. Πέραν αυτού, ενημερώνουμε την τελική λίστα των tubes, αφαιρώντας όλα τα tubes που έχουν σκορ χαμηλότερο από το κατώφλι σύνδεσης.

5. Μετά από αυτό, προσθέτουμε στα ενεργά tubes, τα προτεινόμενα ToIs απ΄ το τρέχον τμήμα, μαζί με τα σκορ δραστηκότητας στην λίστα με τα αθροίσματα δραστηκότητας και μηδενικές τιμές στις αντίστοιχες θέσεις στην λίστα με τα σκορ επικάλυψης.

6. Επαναλαμβάνουμε τα προηγούμενα 3 βήματα μέχρι να μην έχει μείνει κανένα τμήμα βίντεο.

7. Τέλος, όπως αναφέραμε προηγουμένως, έχουμε μια λίστα που περιέχει τα ευρετήρια των αποθηκευμένων tubes. Έτσι, τα τροποποιούμε για να έχουμε τα αντίστοιχα δυσδιάστα πλαίσια. Ωστόσο, οι 2 διαδοχικά ToIs δεν έχουν, πάντα, τα ίδια δυσδιάστα πλαίσια στα καρέ που επικαλύπτονται. Για παράδειγμα, τα ToIs από το τμήμα βίντεο 1^{st} ξεκινούν από το 1ο καρέ έως το 16ο καρέ. Εάν έχουμε βήμα βίντεο ίσο με 8, αυτά τα ToIs επικαλύπτονται χρονικά με τα ToIs από το επόμενο τμήμα βίντεο στα καρέ 8-16. Σε αυτά τα πλαίσια, στον τελικό action tube, επιλέγουμε την περιοχή που περιέχει και τα δύο πλαίσια οριοθέτησης που συμβολίζονται ως $min(x_1, x_1'), min(y_1, y_1'), max(x_2, x_2'), max(y_2, y_2'))$ για τα δυσδιάστατα πλαίσια (x_1, y_1, x_2, y_2) και (x_1, y_1, x_2, y_2).

1.1.1 JHMDB Dataset

Ξεκινώντας, θα ασχοληθούμε αρχικά μόνο με το JHMDB datasetπροκειμένου να καθορίσουμε την πολιτική που ακολουθούμε για να υπολογίσουμε το σκορ επικάλυψης. Κι αυτό γιατί τα βίντεο που περιέχει αυτό το σύνολο δεδομένων είναι πιο μικρά σε διάρκεια και λιγότερα στον αριθμό, οπότε θα μπορέσουμε να βγάλουμε συμπεράσματα πιο γρήγορα απ΄ το να εξετάζαμε και τα δύο σύνολα δεδομένων ταυτόχρονα.

Διάρκεια δείγματος ίση με 16 καρέ Ξεκινάμε ορίζοντας ως διάρκεια δείγματος ίση με 16 καρέ ανά βίντεο κλιπ. Αφού πραγματοποιήσαμε κάποια πρώτα πειράματα με βήμα βίντεο ίσο με 8 και 12 καρέ, στα οποία δεν είχαμε καλές επιδόσεις σε recall, αποφασίσαμε να εξετάσουμε την περίπτωση του βηματος βίντεο ίσο με 14, 15 και 16 τα οποία παρουσιάζονται στον πίνακα 1.1. Για κάθε διαφορετικό βήμα βίντεο έχουμε και διαφορετικές περιπτώσεις στις οποίες εξετάζουμε το σκορ επικάλυψης. Στις περιπτώσεις όπου έχουμε πάνω από 1 καρέ, λαμβάνουμε ως σκορ επικάλυψης την μέση τιμή όλων των καρέ. Στον 1.1 αναφερόμαστε με πιο έντονο χρώμα στα καρέ, για τα οποία εξετάζουμε την επικάλυψη τους.

combination	overlap thresh		
	0.5	0.4	0.3
0,1,...,13**{14,15}** **{14,15}**,16,...,28,29	0.3731	0.5336	0.6493

0,1,...,13,**{14,}**15, **{14,}**15,...,28,29	0.3694	0.5299	0.6455
0,1,...,14,**{15}** 14,**{15,}**16,...,28,29	0.3731	0.5187	0.6381
0,1,...,14,**{15}** **{15}**,16,...,30	0.3918	0.5187	0.6381
0,1,...,14,**{15}** **{16}**,17,...,31	0.4067	0.7313	0.8731

Table 1.1: Recall results for steps = 14, 15 and 16

Παρατηρούμε ότι έχουμε την καλύτερη επίδοση recall για βήμα βίντεο ίσο με 16 καρέ όταν συγκρίνουμε χωρικά την επικάλυψη του τελευταίου πλαίσιου με την επικάλυψη του πρώτου.

Διάρκεια δείγματος ίση με 8 Θέλωντας να επιβεβαιώσουμε ότι έχουμε τα καλύτερα αποτελέσματα όταν έχουμε βήμα βίντεο ίσο με την διάρκεια του δείγματος, εξετάσαμε και την περίπτωση να έχουμε διάρκεια δείγματος ίση με 8. Τα αποτελέσματα παρουσιάζονται στον πίνακα 1.2 και περιλαμβάνει τις περιπτώσεις όπου έχουμε βήμα βίντεο ίσο με 6, 7 και 8 καρέ.

combination	overlap thresh		
	0.5	0.4	0.3
0,1,2,3,4,5**{6,7}** **{6,7}**,8,9,10,11,12,13	0.3134	0.7015	0.8619
0,1,2,3,4,5,**{6,}**7 **{6,}**7,8,9,10,11,12,13	0.3209	0.6679	0.847
0,1,2,3,4,5,6,**{7}** 6,**{7}**8,9,10,11,12,13	0.3172	0.6567	0.8507
0,1,2,3,4,5,6**{7}** **{7,}**8,9,10,11,12,13,14	0.5597	0.7687	0.903
0,1,2,3,4,5,6**{7}** **{8}**9,10,11,12,13,14,15	0.653	0.8396	0.9179

Table 1.2: Recall results for steps = 6, 7 and 8

Με βάση και τα αποτελέσματα του πίνακα 1.2 είναι πλέον ξεκάθαρο ότι πετυχαίνομε καλύτερα αποτελέσματα όταν θέτουμε το βήμα βίντεο ίσο με την διάρκεια του δείγματος και το σκορ επικάλυψης υπολογίζεται από το πλαίσιο του τελευταίου καρέ του πρώτου ToI με το πλαίσιο του πρώτο καρέ του δεύτερου ToI.

1.1.2 UCF Dataset

Σε προηγούμενα βήματα, προσπαθήσαμε να βρούμε την καλύτερη πολιτική επικάλυψης για τον αλγόριθμο μας στο σύνολο δεδομένων JHMDB. Μετά από αυτό, είναι καιρός να εφαρμόσουμε τον αλγόριθμο μας στο σύνολο δεδομένων UCF χρησιμοποιώντας την καλύτερη βαθμολογική πολιτική επικάλυψης. Κάναμε κάποιες τροποποιήσεις στον κώδικα, για να χρησιμοποιούμε λιγότερη μνήμη, και μετακινήσαμε τα περισσότερα μέρη του κώδικα σε GPU. Αυτό συνέβη με τη χρήση τένσορων αντί για λιστες με βαθμολογίες ενώ οι περισσότερες πράξεις είναι, από τώρα και στο εξής, πράξεις πινάκων. Πάνω σ΄ αυτό, το τελευταίο βήμα του αλγόριθμου, η οποία είναι η τροποποίηση από δείκτες σε πραγματικές ακολουθίες από πλαίσια,είναι γραμμένο πλέον σε CUDA κώδικα έτσι λαμβάνει χώρα και στη GPU. Έτσι, τώρα μπορούμε να αυξήσουμε τον αριθμό των ToIs που επιστρέφονται από το TPN, τον μέγιστο αριθμό των ενεργών tubes πριν από την ενημέρωση του ορίου και τον μέγιστο αριθμό τελικών tubes.

Τα πρώτα πειράματα που διενεργήσαμε σχετίζονταν με τον αριθμό των τελικών σωλήνων, τα οποία το δίκτυο μας προτείνει, παράλληλα με τον αριθμό των προτεινόμενων ToIs από το TPN. Πειραματιζόμαστε για υποθέσεις, στις οποίες το TPN προτείνει 30, 100 και 150 ToIs, το τελικό δίκτυό μας προτείνει 500, 2000 και 4000 υποψήφια action tubes για διάρκεια δείγματος ίσο με 8 και 16 καρέ. Για διάρκεια δείγματος ίσο με 8 επιστρέφουμε 100 ToIs επειδή, όταν προσπαθήσαμε να επιστρέψουμε 150 ToIs, λαμβάνουμε OutOfMemory σφάλμα. Ο πίνακας 1.3 δείχνει τις αποδόσεις των χωροχρονικών recall και MABO, αυτών των προσεγγίσεων. Ο πίνακας **;;** δείχνει την απόδοση των χρονικών recall και MABO. Ενδιαφερόμαστε για τη χρονική απόδοση, επειδή το UCF αποτελείται από ατριμάριστα βίντεο, σε αντίθεση με το JHMDB που έχει μόνο τριμαρισμένα βίντεο. Έτσι, θέλουμε να γνωρίζουμε πόσο καλά το δίκτυό μας είναι σε θέση να προτείνει action tubes που επικαλύπτονται με τα πραγματικά action tubes πάνω από ένα «μεγάλο» όριο. Για χρονικό εντοπισμό, δεν χρησιμοποιούμε τα 0,5, 0,4 και 0,3 ως επικαλυπτόμενο όριο, αλλά αντ΄ αυτού, χρησιμοποιούμε 0,9, 0,8 και 0,7, επειδή είναι πολύ σημαντικό το δίκτυό μας να είναι σε θέση να προτείνει action tubes που περιέχουν μια ενέργεια, τουλάχιστον από χρονικής απόψεως. Για να υπολογίσουμε τη χρονική επικάλυψη, χρησιμοποιούμε το IoU για μια μόνο διάσταση.

combination	TPN tubes	Final tubes	0.5	0.4	0.3	MABO
0,1,...,6,**{7,}** **{8,}**9,...,14,15	30	500	0.2829	0.4395	0.5817	0.3501
		2000	0.3567	0.4996	0.6289	0.3815
		4000	0.3749	0.5316	0.6487	0.3934
	100	500	0.2966	0.451	0.5947	0.356
		2000	0.3757	0.5163	0.6471	0.3902
		4000	0.3977	0.5506	0.6624	0.4029
0,1,...,14,**{15,}** **{16,}**17,18,...,23	30	500	0.362	0.5042	0.6243	0.3866
		2000	0.416	0.5468	0.6631	0.4108
		4000	0.4281	0.5589	0.6779	0.4182

	150	500	0.3589	0.4981	0.6198	0.3845
		2000	0.4129	0.5392	0.6563	0.4085
		4000	0.4266	0.5521	0.6722	0.4162

Table 1.3: Recall results for UCF dataset

combination	TPN tubes	Final tubes	overlap thresh 0.9	0.8	0.7	MABO
0,1,...,6,**{7,}** **{8,}**9,...,15	30	500	0.4464	0.581	0.6844	0.7787
		2000	0.635	0.7665	0.8403	0.8693
		4000	0.7034	0.8228	0.8875	0.8973
	100	500	0.454	0.5924	0.692	0.783
		2000	0.651	0.7696	0.8441	0.8734
		4000	0.7209	0.8312	0.8913	0.9026
0,1,...,14,**{15,}** **{16,}**17,18,...,23	30	500	0.6844	0.8327	0.9027	0.8992
		2000	0.7475	0.8684	0.9217	0.9175
		4000	0.7567	0.8745	0.9255	0.9211
	150	500	0.7498	0.8707	0.9171	0.9125
		2000	0.8243	0.911	0.9392	0.9342
		4000	0.8403	0.9179	0.9437	0.9389

Table 1.4: Temporal Recall results for UCF dataset

Όπως φαίνεται και από τους πίνακες 1.3 και 1.4, για διάρκεια δείγματος ίση με 8 λαμβάνουμε την καλύτερη επίδοση όταν επιστρέφει το TPN 100 ToIs και συνολικά το ActionNet 4000 action tubes, ενώ για διάρκεια δείγματος ίση με 16 καρέ όταν επιστρέφει το TPN, 30 ToIs και το ActionNet 4000 action tubes.

Προτεινόμενη τροποποίηση του αλγορίθμου

Στην προηγούμενη προσέγγιση, το κατώφλι σύνδεσης ανανεώνεται και αυξάνεται κάθε φορά ο αριθμός από «ενεργά» tubes ξεπερνούν ένα συγκεκριμένο αριθμό. Ωστόσο, παρατηρήσαμε ότι με αυτόν τον τρόπο το σύστημα μας αδυνατεί να προτείνει action tubes τα οποία ξεκινούν μετά από ορισμένα καρέ. Κι αυτό γιατί μέχρι τότε το κατώφλι σύνδεσης έχει αυξηθεί τόσο που δεν επιτρέπει να δημιουργηθούν νέα tubes . Για τον λόγο αυτό τροποποιήσαμε τον αλγόριθμο μας έτσι ώστε να μην ανανεώνται το κατώφλι σύνδεσης. Παράλληλα, προσθέσαμε τον αλγόριθμο NMS προκειμένου να απορίπτει action tubes που επικαλύπτονται αρκετά με κάποια ήδη προτεινόμενα action tubes. Οι πίνακες 1.5 και 1.6 περιλαμβάνουν τα χωροχρονικά και χρονικά αποτελέσματα για το recall και το MABO, ενώ πειραματιζόμαστε με κατώφλι σύνδεσης του NMS ίσο με 0.7, 0.9 και χωρίς καθόλου NMS.

combination	NMS thresh	PreNMS tubes	overlap thresh 0.9	0.8	0.7	MABO
0,1,...,6,**{7,}** **{8,}**9,...,15	-		0.3779	0.5316	0.6471	0.393082961
	0.7	20000	0.3483	0.5194	0.6471	0.3783524086
	0.9		0.416	0.5605	0.6722	0.4074053106
0,1,...,14,**{15,}** **{16,}**17,...,23	-		0.438	0.5635	0.6829	0.4231788
	0.7	20000	0.4525	0.5848	0.7034	0.429747438
	0.9		0.3802	0.5133	0.6068	0.3862278851848662

Table 1.5: Spatio-temporal Recall results for UCF dataset

combination	NMS thresh	PreNMS tubes	overlap thresh 0.9	0.8	0.7	MABO
0,1,...,6,**{7,}** **{8,}**9,...,15	-	-	0.7087	0.8281	0.8913	0.899210587
	0.7	20000	0.6586	0.854	0.9278	0.903373468
	0.9		0.8137	0.8973	0.9361	0.9333068498
0,1,...,14,**{15,}** **{16,}**17,...,23	-		0.8327	0.9156	0.9399	0.940143272
	0.7	20000	0.8646	0.9369	0.9567	0.946701832
	0.9		0.6183	0.7696	0.8388	0.8628507037919737

Table 1.6: Temporal Recall results for UCF dataset

Συγκρίνοτας τις επιδόσεις των recall και MABO που παρουσιάζονται στον Πίνακας 1.5 μαζί με αυτές του Πίνακα 1.3, συμπεραίνουμε πως για διάρκεια δείγματος ίση με 8, η νέα τροποποίηση οδηγεί σε χειρότερα αποτελέσματα όταν το κατώφλι σύνδεσης είναι 0.7 αλλά καλύτερα για κατώφλι ίσο με 0.9 . Απ΄ την άλλη, για διάρκεια δείγματος ίση με 16, παρατηρούμε πως έχουμε καλύτερα αποτελέσματα για κατώφλι σύνδεσης του NMS αλγορίθμου ίσο με 0.7 .

1.2 Δεύτερη προσέγγιση

Όπως είδαμε και προηγουμένος, ο αλγόριθμος μας δεν έχει πάρα πολύ καλές recall επιδόσεις. Έτσι, δημιουργήσαμε έναν άλλο αλγόριθμο ο οποίος βασίζεται σε αυτόν που πρότειναν οι Hu et al. 2019. Αυτός ο αλγόριθμος εισάγει δύο νέες μετρικές σύμφωνα με τους Hu et al. 2019.

Πρόοδος που περιγράφει την πιθανότητα μιας συγκεκριμένης δράσης να εκτελείται στο ToI. Προσθέτουμε αυτόν τον παράγοντα επειδή παρατηρήσαμε ότι

η δραστικότητα είναι ανεκτική σε ψευδώς θετικά. Η πρόοδος είναι ένα μηχανισμός επαναβαθμολόγης για κάθε κατηγορία (όπως αναφέρονται οι Hu κ.ά. 2019)

Ρυθμός προόδου που ορίζεται ως η αναλογία προόδου κατά την οποία κάθε κατηγορία δράσης έχει πραγματοιποιηθεί.

Έτσι, κάθε action tube περιγράφεται ως ένα σύνολο TOIs

$$T = \{\mathbf{t}_i^{(k)} | \mathbf{t}_i^{(k)} = (t_i^{(k)}, s_i^{(k)}, r_i^{(k)})\}_{i=1:n^{(k)}, k=1:K}$$

όπου το $t_i^{(k)}$ περιέχει τις χωροχρονικές πληροφορίες των TOI , το $s_i^{(k)}$ το σκορ σιγουριάς του και το $r_i^{(k)}$ τον ρυθμό προόδου.

Σε αυτή την προσέγγιση, κάθε κλάση αντιμετωπίζεται ξεχωριστά, συνεπώς για την υπόλοιπη ενότητα συζητάμε για την παραγωγή action tubes μόνο για μία κλάση. Για τη σύνδεση 2 ToIs, για ένα βίντεο με N τμήματα βίντεο , εφαρμόζονται τα ακόλουθα βήματα:

1. Για το πρώτο τμήμα βίντεο (k = 1), προετοιμάζουμε έναν πίνακα με τα M καλύτερα ToIs, τα οποία θα θεωρούνται ως ενεργά tubes(AT). Αντιστοιχικά, προετοιμάζουμε έναν πίνακα με M ρυθμούς προόδου και M βαθμολογίες εμπιστοσύνης.

2. Για k = 2:N, εκτελούμε τα βήματα (α′) - (γ′):

   (α′) Υπολογίζουμε τις επικαλύψεις μεταξύ $AT^{(k)}$ και $TOIs^{(k)}$.

   (β′) Συνδέουμε όλα τα tubes που ικανοποιούν τα ακόλουθα κριτήρια:

      i. $overlapscore(at_i^{(k)}, t_j^{(k)}) < \theta, at \varepsilon AT^{(k)}, t \varepsilon TOIs^{(k)}$

      ii. $r(at_i^{(k)}) < r(t_j^{(k)})$ ή $r(t_i^{(k)}) - r(at_i(k)) < \lambda$

   (γ′) Για όλα τα νέα tubes ενημερώνουμε το σκορ εμπιστοσύνης και τον ρυθμό προόδου ως εξής:

   Το νέο σκορ εμπιστοσύνης είναι η μέση βαθμολογία όλων των συνδεδεμένων TOIs:

   $$s_z^{(k+1)} = \frac{1}{n} \sum_{n=0}^{k} s_i^{(n)}$$

   Ο νέος βαθμός προόδου είναι ο υψηλότερος βαθμός προόδου:

   $$r(at_z^{(k+1)} = max(r(at_i^{(k)}), r(t_j^{(k)}))$$

   (δ′) Διατηρούμε τα M-καλύτερα action tubes ως ενεργά tubes που προορίζονται τελκώς για ταξινόμηση.

Αυτή η προσέγγιση έχει το πλεονέκτημα ότι δεν χρειάζεται να εκτελέσουμε ξανά την ταξινόμηση, επειδή γνωρίζουμε ήδη την κατηγορία του κάθε τελικού action tube. Για να επικυρώσουμε τα αποτελέσματά μας, τώρα, υπολογίζουμε την επίδοση του ρεςαλλ μόνο για τα tubes που έχουν την ίδια κλάση με το προαγματικό tube. Και πάλι θεωρούμε ένα πραγματικό tube ότι είναι θετικό αν υπάρχει τουλάχιοστον ένα tube που επικαλύπτεται με το πραγματικό περισσότερα από ένα προκαθορισμένο όριο.

combination		**overlap thresh**		
sample dur	step	0.5	0.4	0.3
8	6	0.3284	0.5	0.6082
8	7	0.209	0.459	0.6119
8	8	0.3060	0.5672	0.6866
16	8	0.194	0.4366	0.7164
16	12	0.3358	0.5336	0.7537
16	16	0.2649	0.4664	0.709

Table 1.7: Recall results for second approach with step = 8, 16 and their corresponding steps