Κεφάλαιο 1

Classification stage

Στα προηγούμενα 2 κεφάλαια παρουσιάσαμε την διαδικασία που χρησιμοποιούμε για να δημιουργήσουμε υποψήφια action tubes, τα οποία πιθανώς να περιέχουν κάποια πραγματοποιούμενη δράση ή μπορεί όχι. Τις περισσότερες φορές τα προτεινόμενα action tubes ανήκουν στο φόντο, και γι΄ αυτό, όπως αναφέρθηκε και στον προηγούμενο κεφάλαιο, είναι σημαντικό να επιλέξουμε έναν καλό αλγόριθμο που προτείνει καλές ακολουθίες από πλαίσια. Ωστόσο, είναι αρκετά σημαντικό να επιλέξουμε και τον κατάλληλο ταξινομητή ο οποίος θα είναι σε θέση με μεγάλη ακρίβεια να προβλέψει αν ένα υποψήφιο action tube ανήκει σε μια γνωστή κατηγορία από δράσεις ή ανήκει στο φόντο. Κι αυτό γιατί μπορεί να παράγουμε καλές προτάσεις για υποψήφιες δράσεις, αλλά αν ο ταξινομητής μας δεν λειτουργεί στο έπακρο, το σύστημα μας πάλι θα αποτυγχάνει να αναγνωρίσει τις δράσεις.

Η σωστή επιλογή ενός ταξινομητή είναι μια μεγάλη απόφαση που καλούμαστε να πάρουμε. Ωστόσο, αυτός ο ταξινομητής θα δεχθεί ορισμένους χάρτες ενεργοποίησης τους οποίους θα κληθεί να ταξινομήσει. Συνεπώς, εκτός από την καλή επιλογή ταξινομητή, εξίσου σημαντική είναι η καλή επιλογή χαρακτηριστικών. Τέλος, μεγάλο ρόλο παίζει και η διαδικασία εκπαίδευσης του ταξινομητή προκειμένου να είναι σε θέση να γενικεύει και καταστάσεις overfitting να αποφεύγονται.

Σε αυτό το κεφάλαιο παρουσιάζουμε διάφορες μεθόδους που χρησιμοποιήσαμε οι οποίες περιλαμβάνουν ένα Γραμμικό ταξινομητή, ένα Recursive Neural Network (RNN), ένα Support Vector Machine (SVM)και ένα Multilayer Perceptron(MLP). Επίσης, πειραματιζόμαστε χρησιμοποιώντας χάρτες χαρακτηριστικών που εξηχθησαν μέσω του 3D RoiAlign χρησιμοποιώντας παράλληλα avg ή max pooling. Τελευταίο αλλά εξίσου σημαντικό είναι το γεγονός ότι προσπαθήσαμε να βρούμε το καλύτερο ποσοστό μεταξύ actio tubes προσκηνίου και φόντο αλλά και τον συνολικό αριθμό τους που είναι απαραίτητα κατά την διάρκεια της εκπαίδευσης προκειμένου ο ταξινομητής να λειτουργεί αποδοτικά.

Η όλη διαδικασία ταξινόμησης αποτελείται από τα ακόλουθα βήματα:

1. Διαχωρίζουμε το βίντεο σε μικρά βίντεο κλιπ και τροφοδοτούμε το δίκτυο TPN με αυτά τα βίντεο κλιπ και παίρνουμε ως αποτέλεσμα k-προτεινόμενα ToIs και τα αντίστοιχα χαρακτηριστικά τους για κάθε κλιπ βίντεο.

2. Συνδέουμε τα προτεινόμενα ToIs για να πάρουμε action tubes που μπορεί να περιέχουν μια ενέργεια.

3. Για κάθε υποψήφιο action tube, η οποία είναι μια ακολουθία του ToIς, τροφοδοτούμε τους χάρτης ενεργοποίησης του στον ταξινομητή για ταξινόμηση.

Στα πρώτα βήματα του σταδίου ταξινόμησης αναφερόμαστε μόνο στο σύνολο δεδομένων JHMDB, επειδή έχει μικρότερο αριθμό βίντεο από το σύνολο δεδομένων UCF το οποίο μας βοήθησε να εξοικονομήσουμε πολύ χρόνο και πόρους. Αυτό συμβαίνει επειδή κάναμε τα περισσότερα πειράματα μόνο JHMDB και αφού βρήκαμε τη βέλτιστη υλοποίηση, την υλοποιήσαμε για το UCF, επίσης.

1.1 JHDMB dataset

1.1.1 Ταξινομητές Linear, SVM και RNN

Training Για να εκπαιδεύσουμε τον ταξινομητή μας, πρέπει να εκτελέσουμε τα προηγούμενα βήματα, για κάθε βίντεο. Ωστόσο, κάθε βίντεο έχει διαφορετικό αριθμό καρέ και καταλαμβάνει μεγάλη ποσότητα μνήμης στη ΓΠΥ. Για να αντιμετωπίσουμε αυτή την κατάσταση και έχοντας 4 διαθέσιμες GPU, δίνουμε ως είσοδο ένα βίντεο ανά GPU. Έτσι μπορούμε να χειριστούμε 4 βίντεο ταυτόχρονα. Αυτό σημαίνει ότι ένα κλασσικό training παίρνει πάρα πολύ χρόνο για μόλις 1 εποχή. Η λύση με την οποία ήρθαμε, είναι να προυπολογίσουμε τους χάρτες χαρακτηριστικών τόσο για action tubes προσκηνίου όσο και φόντου και στη συνέχεια να τροφοδοτήσουμε αυτούς τους χάρτες στον ταξινομητή μας για να τον εκπαιδεύσουμε.Αυτή η λύση περιλαμβάνει τα ακόλουθα βήματα:

1. Αρχικά, εξαγουμε τους χάρτες χαρακτηριστικών από τα πραγματικά action tubes . Ακόμα εξάγουμε τα χαρακτηριστικά από action tubes φόντου τα οποία είναι διπλάσια στον αριθμό από αυτά του φόντου. Επιλέξαμε αυτή την αναλογία μεταξύ του αριθμού των θετικών και αρνητικών action tubes εμπνευσμένοι από τους Yang et al. 2017, των οποίων η μέθοδος χρησιμοποιεί ποσοστό 25% μεταξύ των περιοχών ενδιαφέροντος προσκηνίου και των συνολικών περιοχών, και συνολικά επιλέγει 128 τέτοιες περιοχές. Αντίστοιχα, επιλέγουμε ένα λίγο μεγαλύτερο ποσοστό επειδή έχουμε μόνο ένα πραγματικό action tube σε κάθε βίντεο. Έτσι, για κάθε βίντεο λαμβάνουμε 3 action tubes συνολικά, 1 προσκηνίου και 2 φόντου. Θεωρούμε ως background action tubes εκείνα που το σκορ επικάλυψης τους με οποιοδήποτε action tube είναι μεγαλύτερο απο 0.1 αλλά μικρότερο από 0.3 . Φυσικά,προκειμενου να εξάγουμε αυτά τα action tubes, χρησιμοποιούμε ένα προεκπαιδευμένο TPN, για να μας προτείνει ToIs για κάθε τμήμα βίντεο και τον προτεινόμενο αλγόριθμο σύνδεσης για να συνδέσουμε αυτά τα ToIs. Τελικώς, για κάθε action tube λαμβάνουμε τους αντίστοιχους χάρτες ενεργοποίησης χρησιμοποιώντας 3D RoiAlign.

2. Αφού εξαγουμε αυτά τα χαρακτηριστικά, εκπαιδεύουμε τους ταξινομητές μας. Ο Γραμμικός ταξινομητής χρειάζεται ένα σταθερό μέγεθος εισόδου,

συνεπώς χρησιμοποιούμε μια συνάρτηση pooling στην διάσταση του αριθμού των βίντεο. Έτσι, αρχικά έχουμε ένα χάρτη χαρακτηριστικών μεγέθους *3,512,16* και μετά λαμβάνουμε ως έξοδο έναν χάρτη χαρακτηριστικών μεγέθους *512,16*. Πειραματιζόμαστε χρησιμοποιώντας αμφότερα max και avg pooling όπως φαίνεται στον Πίνακα χρησιμοποιώντας 1.1. Για τον ταξινομητή RNN δεν χρειαζόμαστε καμία pooling διαδικασία ενώ για τον ταξινομητή SVM πειραματιζόμαστε ξανά χρησιμοποιώντας και τις δύο αυτές συναρτήσεις τα αποτελέσματα του οποίου φαίνονται στον Πίνακα 1.2.

Validation Το στάδιο επικύρωσης περιλαμβάνει τη χρήση τόσο προεκπαιδευμένου TPN όσο και του ταξινομητή. Έτσι, για κάθε βίντεο λαμβάνουμε σκορ ταξινόμησης για τα προτεινόμενα action tubes. Οι περισσότερες προσεγγίσεις συνήθως θεωρούν ένα κατώφλι σκορ εμπιστοσύνης πάνω από το οποίο θεωρούν ένα action tube ως προσκήνιο. Ωστόσο, εμείς δεν χρησιμοποιούμε κανένα σκορ εμπιστοσύνης. Αντιθέτως, επειδή γνωρίζουμε ότι JHMDB έχει κομμένα βίντεο με μόνο 1 εκτελούμενη δράση ανά βίντεο, εμείς απλά θεωρούμε το καλύτερο ως προς το σκορ action tube ως πρόβλεψη.

Classifier	**Pooling**	**mAP**		
		0.5	0.4	0.3
Linear	mean	14.18	19.81	20.02
	max	13.67	16.46	17.02
RNN	-	11.3	14.14	14.84

Table 1.1: First classification results using Linear and RNN classifiers

Dimensions		**Pooling**	**mAP precision**		
before	after		0.5	0.4	0.3
(k,64,8,7,7)	(1,64,8,7,7)	mean	3.16	4.2	4.4
(k,64,8,7,7)	(1,64,8,7,7)	max	1.11	2.35	2.71
(k,256,8,7,7)	(1,256,8,7,7)	mean	11.41	11.73	11.73
(k,256,8,7,7)	(1,256,8,7,7)	max	**22.07**	**24.4**	**25.77**

Table 1.2: Our architecture's performance using 5 different policies and 2 different feature maps while pooling in tubes' dimension. With bold is the best scoring case

1.1.2 Temporal pooling

Μετά τη λήψη των πρώτων αποτελεσμάτων, εφαρμόζουμε μια συνάρτηση χρονικής ομαδοποίησης (temporal pooling) εμπνευσμένη από το Hou, ἤεν και Σηαη

2017. Χρειαζόμαστε ένα σταθερό μέγεθος εισόδου για το Σ΄Μ. Ωστόσο, το χρονικό stride των action tube μας ποικίλλει από 2 έως 5, αφού ένα βίντεο με 15 καρέ αποτελείεται από 2 συνεχόμενες ToIs ενώ ένα βίντεο με 40 καρέ αποτελείται απο 5. Έτσι χρησιμοποιούμε ως σταθερή χρονική διάσταση ίσον με 2. Ως λειτουργία pooling χρησιμοποιούμε 3D max poolign, για κάθε φίλτρο του χάρτη χαρακτηριστικών ξεχωριστά. Για παράδειγμα, για ένα action tube με 4 συνεχόμενες ToIs, έχουμε $(4, 256, 8, 7, 7)$ ως μέγεθος του χάρτη χαρακτηριστικών. Διαχωρίσουμε το feature map σε 2 ομάδες χρησιμοποιώντας την συνάρτηση *linspace* και αναδιαμορφώνουμε το χάρτη χαρακτηριστικών σε $(256, k, 8, 7, 7)$ όπου k είναι το μέγεθος της κάθε ομάδας. Αφού κάνουμε χρήση 3D max pooling, θα πάρουμε ένα χαρακτηριστικό χάρτη διαστάσεων $(256, 8, 7, 7)$, ακολούθως τους ενώνουμε και τελικά λαμβάνουμε χαρακτηριστικών μεγέθους $(2, 256, 8, 7, 7)$. Σε αυτή την περίπτωση δεν πειραματιζόμαστε με χάρτες χαρακτηριστικών μεγέθους (64, 8, 7, 7) επειδή με βάση τα παραπάνω αποτελέσματα, δεν θα έχουμε καλύτερη επίδοση απ΄ τα χαρακτηριστικών μεγέθους $(256, 8, 7, 7)$. Τα αποτελέσματα παρουσιάζονται στον πίνακα 1.3, όπου περιλαμβάνεται η καλύτερη προηγούμενη μέθοδος η οποία χρησιμοποιεί max pooling αντί για temporal pooling.

Dimensions		**Temp Pooling**	**mAP precision**		
before	after		0.5	0.4	0.3
k,256,8,7,7	1,256,8,7,7	-	22.07	24.4	25.77
k,256,8,7,7	2,256,8,7,7	Yes	25.07	26.91	29.11

Table 1.3: mAP results using temporal pooling for both RoiAlign approaches