EIT Digital Masters - M1 DSC

Research Report
26/06/2019

Summarization of Sports Videos With A Fixed Duration

Supervisors:
Pr. Frederic Precioso, Melissa Sanabria Rosas

Author:
Sherly

Abstract

There has been a monumental growth in the amount of user-generated video data. These videos are extremely diverse in their content, and can vary in length from a few seconds to a few hours. It is therefore becoming increasingly important to automatically extract a brief yet informative summary of these videos in order to enable a more efficient and engaging viewing experience. Video summarization is defined as generating a shorter video clip which includes only the important scenes in the original video streams. In this project, we proposed a Multitask learning based methodology to perform video summarization with a constraint on the duration of the generated video summary.

Key words: Sports, Video, Summarization, Neural Networks, Multitask Learning

Contents

1 Introduction

1.1 Context

There has been many variations of works in the field of video summarization in recent years [1, 2, 3, 4]. Many of these approaches works around the visual features to produce the summaries. [1, 3, 4] extracts features from individual frames of the video as input and [2] uses C3D and extract features of every 16 frames as input. In this project, we utilise a multimodal approach to automatically generate summaries of soccer match videos that consider both event and audio features. The event features get a shorter and better representation of the match, and the audio helps detect the excitement generated by the game. This project is a further work from our previous work on multimodal approach [Section 3.1]. This method consists of three consecutive stages: Proposals, Summarization and Content Refinement. The first one generates summary proposals, using Multiple Instance Learning to deal with the similarity between the events inside the summary and the rest of the match. The Summarization stage uses event and audio features as input of a hierarchical Recurrent Neural Network to decide which proposals should indeed be in the summary. The last stage, takes advantage of the visual content to create the final summary. In this follow-up project, we would like to generate a summary for the video with a given **time constraint** for the summarized video.

1.2 Motivations

A visual analysis is performed on the results of the Summarization and Content Refinement stage in the original method [see **Section 3.1**]. We observed that the predictions of the probability of a proposal being positive are on the extremes instead of a smooth distribution in the Summarization stage, Figure 14 in Appendix. In the Content Refinement stage, we observed irregular distribution of the predictions which results in a video summary that is made of many small segments of even just a second or less. The resulting video summary may not be one with smooth transitions from one important segment to another. Hence, we proposed to change the use of GoogleNet on individual frames to using C3D to the sets of 16 frames to generate the input features to this stage. Combining the two observations, we would like to explore alternatives that could improve the entire task as a whole. In this project, we proposed to explore Multitask Learning framework which jointly learns the two tasks at the same time.

Finally, in order to reach the goal of the project, a multi-task loss given by the two tasks is proposed and a soft constraint is imposed on it in order to limit the duration of the generated video. The constraint is based on the target duration for the generated video summary.

1.3 Goals

The ultimate goal of the project is to generate a summary for the video with a given constraint of time for the output video. Various multitask learning methodology and constraint methods will be studied in this project in order to achieve that.

2 Background

This section describes the concepts and theory that is necessary to understand the work that is done in this project. It will first describe the concept of Multitask Learning, the application of multitask learning in the context of video summarization for this project and the evaluation metrics used.

2.1 Multitask Learning

Multi-Task Learning (MTL) is a learning paradigm in machine learning and its aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks [5].

There has been many works involving multitask learning in the fields of computer vision, bioinformatics, health informatics, speech, natural language processing, web applications and ubiquitous computing. Among these learning tasks, all of them or at least a subset of them are assumed to be related to each other. It is found that learning these tasks jointly can lead to much performance improvement compared with learning them individually.

(Multi-Task Learning)

> Given m learning tasks $\{T_i\}_m$ i=1 where all the tasks or a subset of them are related, this method aims to help improve the learning of a model for T_i by using the knowledge contained in all or some of the m tasks.

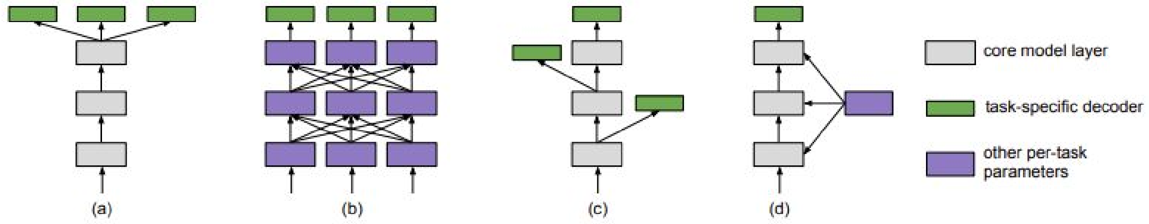


Figure 1: Existing deep multitask learning architectures

Many multitask learning architectures has been proposed over the years. Meyerson et al [6] has summarized the existing deep multitask learning architectures as shown in the figure above. Describing the figure above, (a) Classical approaches add a task-specific decoder to the output of the core single-task model for each task; (b) Column-based approaches include a network column for each task, and define a mechanism for sharing between columns; (c) Supervision at custom depths adds output decoders at depths based on a task hierarchy; (d) Universal representations adapts each layer with a small number of task-specific scaling parameters.

2.2 Multitask Learning for Video Summarization

There are two learning task in the model that we have proposed for this project on video summarization task. Firstly, this project is a further experimentation from the initial work based on the architecture shown in **Section 3.1.** As such, we will begin the experiments from that step and use the results obtained as our baseline. There are three stages to the original architecture: 1) Proposals, 2) Summarization and 3) Content Refinement. The Proposal stage remains unchanged and in this project, we will be modifying the Summarization and Content Refinement stage to a Multitask Learning architecture. As mentioned in **Section 2.1**, there are multiple architectures to Multitask Learning and in this case, we will experiment with the multi-input and multi-output architecture where a joint loss from the different tasks are optimized.

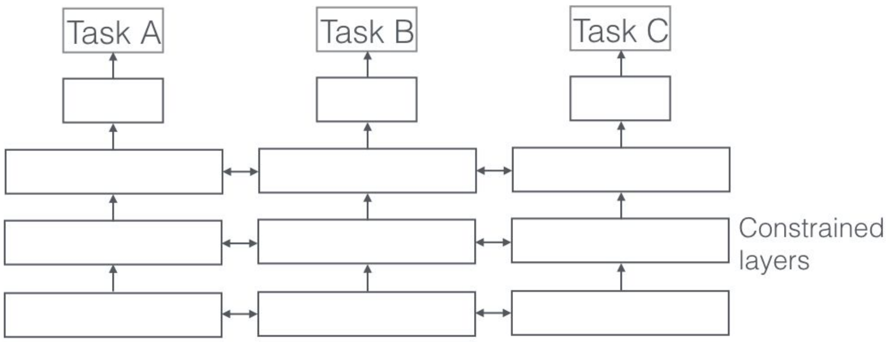


Figure 2: Multi-input, multi-output multitask learning architecture

Section 3.2 describes the details of the experiments conducted in this project.

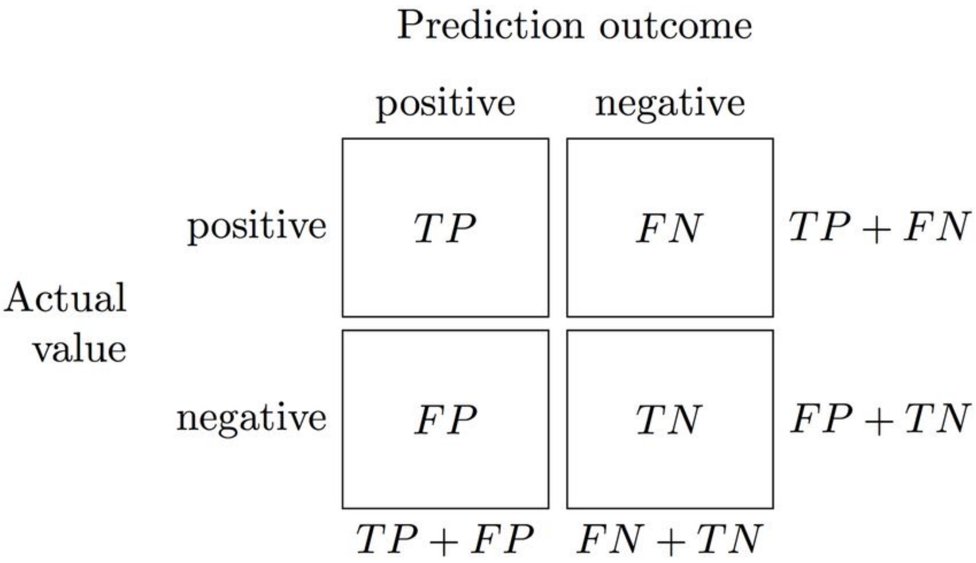
2.3 Constraint Methodologies

The goal of the project is to generate video summaries of a fixed duration. Many of the work on video summarization today focuses on generating good summaries that represents the original video. However, little work has been done in recent years on constraining the output duration of the video summary. There are three ways to approach this problem: 1) thresholding to reduce or increase the number of proposals in the intermediate or final stage to achieve the target duration, 2) post-processing of the final proposals to restrict the duration or 3) performing video summarization as a constraint satisfaction problem. The first two approaches relies mainly on manual intervention or heuristics. Constraint satisfaction programming (CSP) for video summarization has been explored previously. Sid et al [7], proposed a method that separates the generation rules and constraint rules by using a CSP solver. Shi et al [8] proposed a Greedy method in Constraint Satisfaction framework. There are two general ways to constraint the results: 1) soft constraint and 2) hard constraint. Soft constraints is performed by adding new terms to the loss function that is minimized during training. The alternative, hard constraints, imposes constraints on the parameters themselves.

2.4 Evaluation Metrics

In a machine learning framework, the next step after performing a model training is to evaluate our model based on the metrics that we have defined. The context in which we are modeling is the classification of sports actions and hence, metrics are built on the comparison between true classes and the predicted classes. Some metrics that are common for classification problems are accuracy, precision, recall and F1-scores [see below]. These metrics can be easily calculated based on the confusion matrix which defines the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN):

- True positive (TP) is defined as the number of correctly predicted positive values.
- True negative (TN) is defined as the number of correctly predicted negative values.
- False positive (FP) is defined as the number of positively predicted values which are actually not positive. It is also called the Type 1 error.
- False negative (FN) is defined as the number of negatively predicted values which are actually not negative. It is also called the Type 2 error.



		Prediction outcome		
		positive	negative	
Actual value	positive	TP	FN	$TP+FN$
	negative	FP	TN	$FP+TN$
		$TP+FP$	$FN+TN$	

Figure 3: Confusion Matrix

Based on these four values, we calculate the following evaluation metrics:

- **Accuracy** is the ratio between the number of classes correctly predicted and the total number of classes to predict.

$$Accuracy \ = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations

$$Precision \ = \frac{TP}{TP + FP}$$

- **Recall** is the ratio of correctly predicted positive observations to the all observation in actual class.

$$Recall = \frac{TP}{TP + FN}$$

- The **F-beta score** is the weighted harmonic mean of precision and recall, reaching its optimal value at 1 and its worst value at 0.

$$F_\beta - score = (1 + \beta^2) \frac{precision \times recall}{(\beta^2 \times precision) + recall}$$

In the context of Video Summarization, we evaluate our generated summary U by computing the Precision, Recall and F-score against V , the summary created by the editors:

$$Precision = \frac{overlapped\ duration\ of\ U\ and\ V}{duration\ of\ U}$$

$$Recall = \frac{overlapped\ duration\ of\ U\ and\ V}{duration\ of\ V}$$

$$Fscore = 2 \cdot \frac{precision \times recall}{precision + recall}$$

3 Methodologies

3.1 Original Architecture

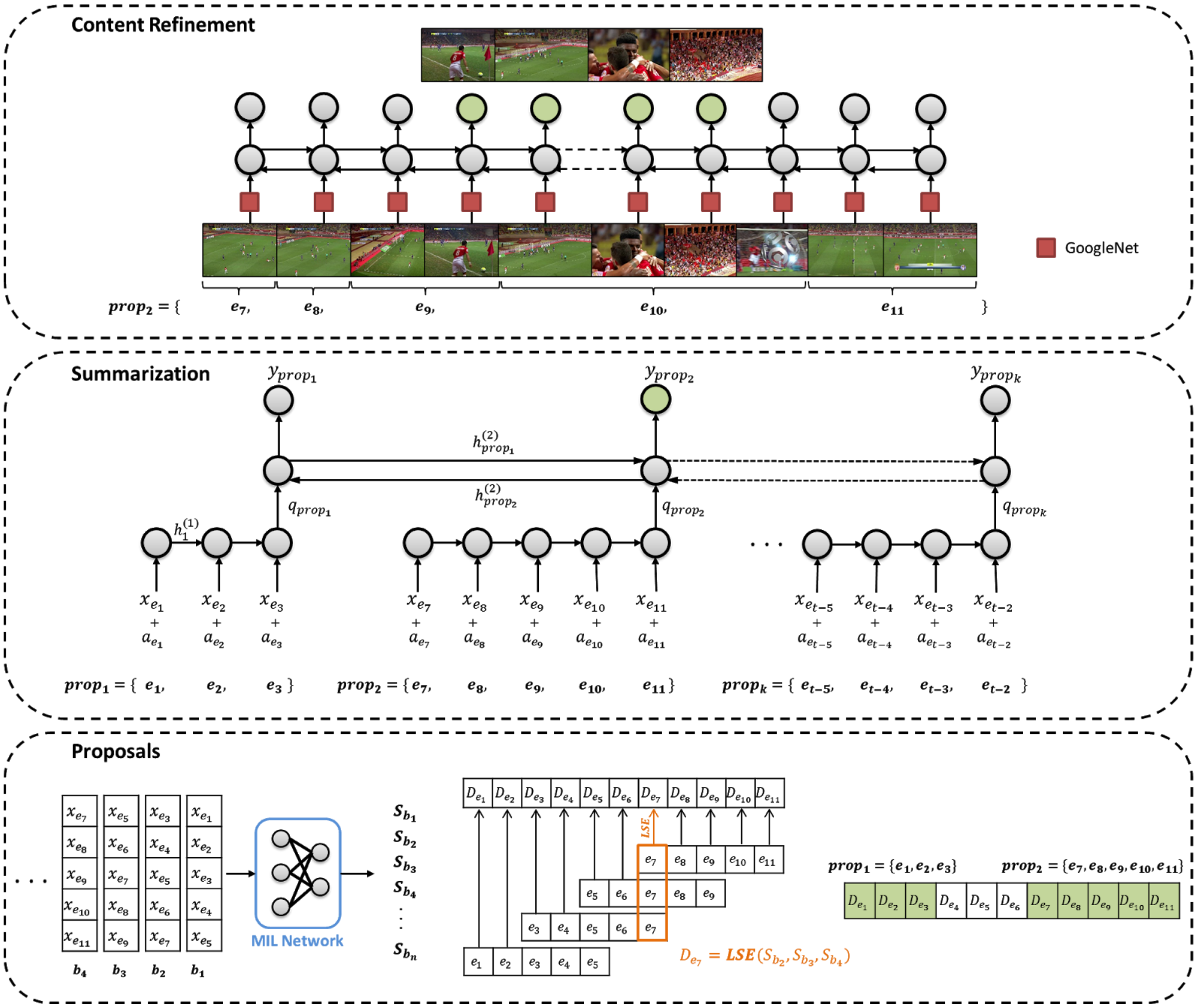

Content Refinement
GoogleNet
$prop_2 = \{ e_7, e_8, e_9, e_{10}, e_{11} \}$
Summarization
y_{prop_1}
y_{prop_2}
y_{prop_k}
$h^{(2)}_{prop_1}$
$h^{(2)}_{prop_2}$
q_{prop_1}
q_{prop_2}
q_{prop_k}
$h^{(1)}_1$
$x_{e_1} + a_{e_1}$, $x_{e_2} + a_{e_2}$, $x_{e_3} + a_{e_3}$
$x_{e_7} + a_{e_7}$, $x_{e_8} + a_{e_8}$, $x_{e_9} + a_{e_9}$, $x_{e_{10}} + a_{e_{10}}$, $x_{e_{11}} + a_{e_{11}}$
$x_{e_{t-5}} + a_{e_{t-5}}$, $x_{e_{t-4}} + a_{e_{t-4}}$, $x_{e_{t-3}} + a_{e_{t-3}}$, $x_{e_{t-2}} + a_{e_{t-2}}$
$prop_1 = \{ e_1, e_2, e_3 \}$
$prop_2 = \{ e_7, e_8, e_9, e_{10}, e_{11} \}$
$prop_k = \{ e_{t-5}, e_{t-4}, e_{t-3}, e_{t-2} \}$
Proposals
b_4 b_3 b_2 b_1
MIL Network
S_{b_1} S_{b_2} S_{b_3} S_{b_4} S_{b_n}
D_{e_1} … $D_{e_{11}}$
LSE
$D_{e_7} = LSE(S_{b_2}, S_{b_3}, S_{b_4})$
$prop_1 = \{e_1, e_2, e_3\}$
$prop_2 = \{e_7, e_8, e_9, e_{10}, e_{11}\}$

Figure 4: Multimodal architecture to Video Summarization

The original architecture trains a model that combines the precision and relevance of event metadata with the expressiveness of multimedia content. There are three stages in this architecture:

1. Proposals stage deals with the similarity of inter-categorical actions. Two very similar sets of events pass, tackle, pass can be part of two different actions goal-opportunity and corner, the former being in the summary while the latter not. This issue is addressed

by a Multiple Instance Learning (MIL) network providing a score for each event, further concatenated to end up with consecutive positive events as proposals. $E = \{e_1, e_2, ..., e_t\}$. These events are "atomic" soccer actions like: pass, tackle, out, etc. If there are three consecutive passes in the game, you will have three similar events "pass" in a row. $X = \{X_{e_1,} X_{e_2}, .., X_{e_t}\}$ represents the set of instances for a bag in MIL, where X_{e_t} is the feature vector characterizing the t-th event of the match.

2. Summarization stage consists of a multimodal Hierarchical LSTM. The first level LSTM accumulate in each proposal from previous stage, the emotion and excitement information of every concerned event using metadata-based feature vectors concatenated with audio features. The second level is a bi-LSTM capturing the forward-backward temporal dependencies among proposals in order to predict the probability of each proposal to be selected into the summary. The input to the first level is $\{X_{e_t} + a_{e_t}\}$, the multimodal feature vector for the t −th event. X_{e_t} is the feature vector characterizing the t-th event of the match generated from the event metadata and a_{e_t} is the feature vector representing the audio feature of the t-th event.
3. Content refinement stage exploits the visual information of each frame to refine the boundaries of the clips predicted as being part of the summary so that the resulting clips are not anymore restricted to start and/or end on event boundaries. A final bi-LSTM network hence predicts which frames among the ones belonging to the pre-selected proposals should be preserved in the final summary. This last stage allows also to focus on visually salient frames. Each input sample of the LSTM corresponds to the feature vectors of the frames inside the interval [F_{beg} , F_{end}] which is given by the timestamp of the first and last events of a proposal.

3.2 Proposed Experiments

3.2.1 Visual features with C3D

The original architecture shown in **Section 3.1** generates the visual features of the frames for positive events proposed by the Summarization stage. The visual features of each frame are generated with GoogleNet architecture and is used as the input to the Content Refinement stage. From our observation, utilizing individual frame based features generates a highly fluctuating distribution on the continuous frames as shown in Figure 5.

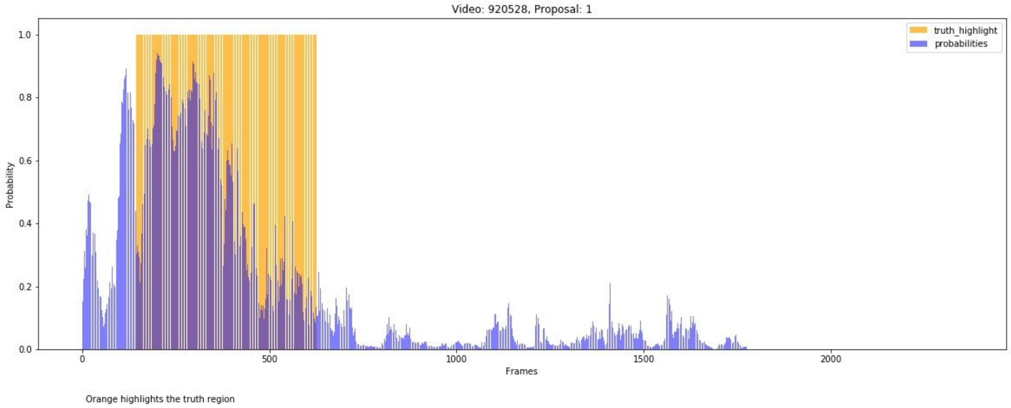


Figure 5: The distribution of probability over continuous frames in time

Therefore, we proposed to experiment with Convolution 3D (C3D) [12] network on the frames to observe the distribution as C3D preserves the spatio temporal features. Instead of using 2D ConvNets, this architecture uses a 3 x 3 x 3 convolutional kernels in all layers.

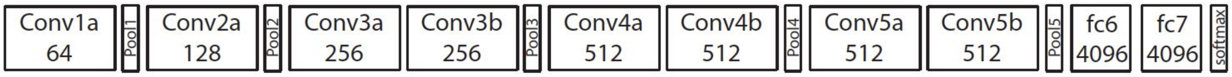


Figure 6: C3D Architecture

C3D takes a video clip of length 16 frames to extract the feature. The input proposals are marked with a beginning and end time according to the corresponding events in the proposals and the video is then broken down into sets of 16 frames to generate the visual features as the input to the Content Refinement stage.

3.2.2 Multi-task Learning

In this project, we proposed to view the Summarization and Content Refinement stages as a Multitask Learning model rather than a hierarchical model. The input proposals to both the Summarization and Content Refinement stage are the output proposals from the MIL network, Proposal stage. We would like to optimize the model and the tasks of each stage jointly via the multitask learning approach. At the end, the final label for the proposals are defined by the positivity of the jointly trained Summarization and Content Refinement stage. Figure 7 shows the proposed Multitask framework.

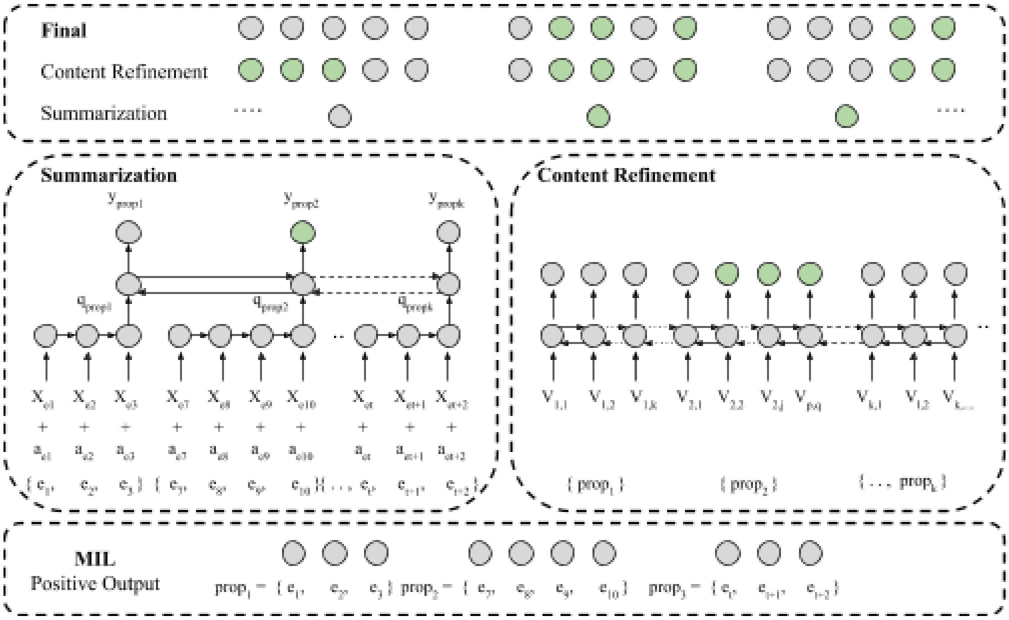

Final
Content Refinement
Summarization
Summarization
y_prop1
y_prop2
y_propk
q_prop1
q_prop2
q_propk
X_e1 X_e2 X_e3 X_e7 X_e8 X_e9 X_e10 X_et X_et+1 X_et+2
a_e1 a_e2 a_e3 a_e7 a_e8 a_e9 a_e10 a_et a_et+1 a_et+2
{ e_1, e_2, e_3 } { e_7, e_8, e_9, e_10 }{ . . . e_t, e_t+1, e_t+2 }
Content Refinement
V_1,1 V_1,2 V_1,k V_2,1 V_2,2 V_2,j V_p,q V_k,1 V_1,2 V_k,...
{ prop_1 }
{ prop_2 }
{ . . . prop_k }
MIL
Positive Output
prop_1 = { e_1, e_2, e_3 } prop_2 = { e_7, e_8, e_9, e_10 } prop_3 = { e_t, e_t+1, e_t+2 }

Figure 7: Multitask Learning architecture for Multimodal Video Summarization

The goal of this approach is to achieve better predictions with the joint training and to impose soft constraints on the joint-loss function in order to generate videos of fixed duration. The subsections below describes the different approaches and exploration in the field of Multi task learning for this project.

3.2.2.1 Multi-task Learning with individual task loss

The first step to approaching Multitask learning was to change the network to multi-task network and in this case, the optimization for each task is trained separately based on the loss of each task.

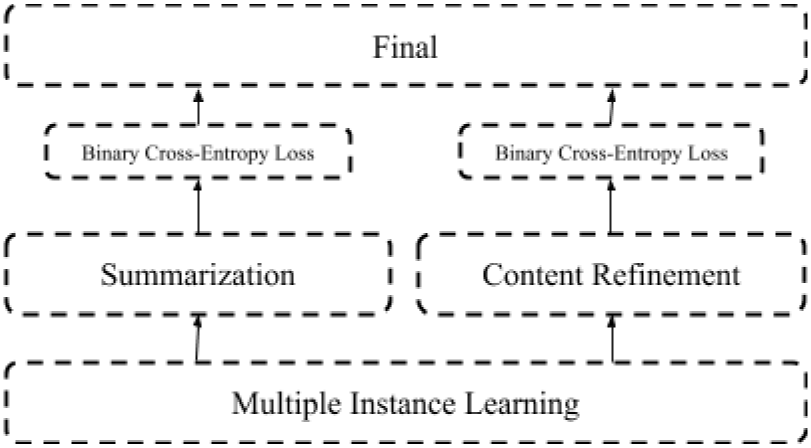

Final
Binary Cross-Entropy Loss
Binary Cross-Entropy Loss
Summarization
Content Refinement
Multiple Instance Learning

Figure 8: Multitask Learning with Individual Task Loss

3.2.2.2 Multi-task Learning with Joint Loss

In order to study the impact of a joint training of both tasks in order to achieve the final objective, the network is modified to optimize based on the joint loss of the two tasks. In this case, the joint loss is defined as the sum of the loss of the Summarization and the Content Refinement stage.

$$loss = summarization\ loss + content\ refinement\ loss$$

The figure below shows the network architecture.

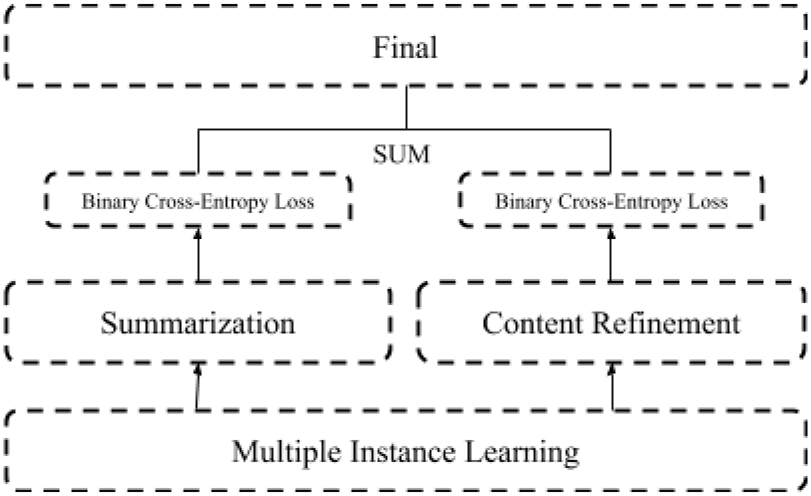

Final
SUM
Binary Cross-Entropy Loss
Binary Cross-Entropy Loss
Summarization
Content Refinement
Multiple Instance Learning

Figure 9: Multitask Learning with Joint Loss

3.2.2.3 Multi-task Learning with Custom Loss

In order to better fit Multitask Learning to our context, we proposed a new loss with a soft constraint. The soft constraint adds a new term to the loss function in order to optimize the model to summarize for a given fixed duration. The loss function is defined as:

$$loss = \sum_{i=0}^{q}\sum_{j=0}^{p} loss^{i}_{summarization} \times loss^{i,j}_{content} - [\,N_s \times \tfrac{25}{16}\,]$$

It is the sum of the product of losses of the proposals in summarization stage against its corresponding loss in content refinement stage. The term $[\,N_s \times \tfrac{25}{16}\,]$ is the constraint term where N_s is the number of seconds for the video summary and 25 being the number of frames per second and 16 the number of frames in one set of feature for C3D. The network diagram is as shown below.

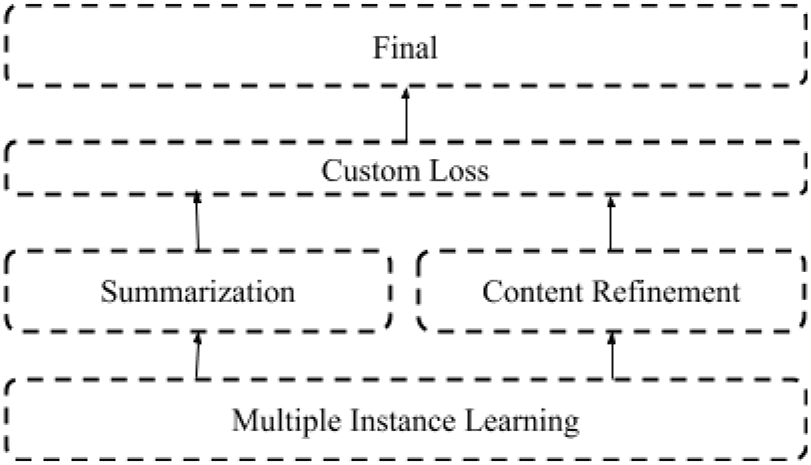


Figure 10: Multitask Learning with Custom Joint Loss

4 Dataset

There are two main dataset that is being utilised in this project namely Opta data and the actual football videos from WildMoka. The sections below will elaborate on the datasets as well as the feature extraction and utilisation on these datasets.

4.1 Opta Data

Opta Sports is an international sports analytics company based in the United Kingdom. It provides data for 30 sports in 70 countries, with clients ranging from leagues to broadcasters and betting websites. In this project, Wildmoka provides us with the Opta labelled data with metadata features for the football videos that we are performing action detections on.

4.1.1 Feature Extraction

Potentially positive events are first proposed by the Proposal stage. In the Summarization stage, the same set of metadata features in the Proposal stage are also utilized. Let the set of proposals be $props = \{prop_1, prop_2, .., prop_k\}$ where $prop_k = \{e_m, e_{m+1}, ...\}$ the set of positive proposals by the Proposal stage. We have X_{e_t} the feature vector characterizing the t-th event of the match. The table below lists the features utilised in the generation of the feature vector.

Feature	Description
Type Array	Encoded array that signifies the type of action for an instance

Qualifiers Array	Encoded array that signifies the qualifiers for an instance
Outcome	Outcome of the action, as per listed in F24 Appendices
x	X position of the action
y	Y position of the action

Table 1: Description of metadata features from Opta

In addition to the metadata utilized in the Proposal stage, the Summarization stage incorporates audio features as well. Sub-band short-time energies are used as proposed by As proposed by Rui et al. [9]. These sub-bands are 0- 630Hz (En_1), 630-1720Hz (En_2), 1720-4400Hz (En_3), and 4400Hz and above (En_4). Since each event has a timestamp corresponding to the video time when the event occurs, the energy of the event e_t is the energy of the second *s* corresponding to its timestamp. We set audio features into a vector a_{e_t} concatenating En^s , En^s_1, En^s_2, En^s_3, and En^s_4. An event is then represented by the concatenated feature vector $\{X_{e_t} + a_{e_t}\}$.

4.2 Video data

4.2.1 Feature Extraction

The content refinement stage uses the visual features from the video data. In the original proposed methodology, GoogleNet is used to generate the visual features of each frame of the positive events that has been classified by the Summarization stage. Through visual inspection of the results of the Content Refinement stage in the original methodology, we observed a highly irregular refinement of the summary. Hence, a new experiment by using Convolutional 3D (C3D) to generate the visual features are conducted, **Section 3.2.1**.

The proposals from the Proposal Stage are translated into the video data by picking up the respective start and end of proposal through its event metadata. Video segments are then split into non-overlapped 16-frame clips which are then used as input to the C3D networks in order to generate the visual features.

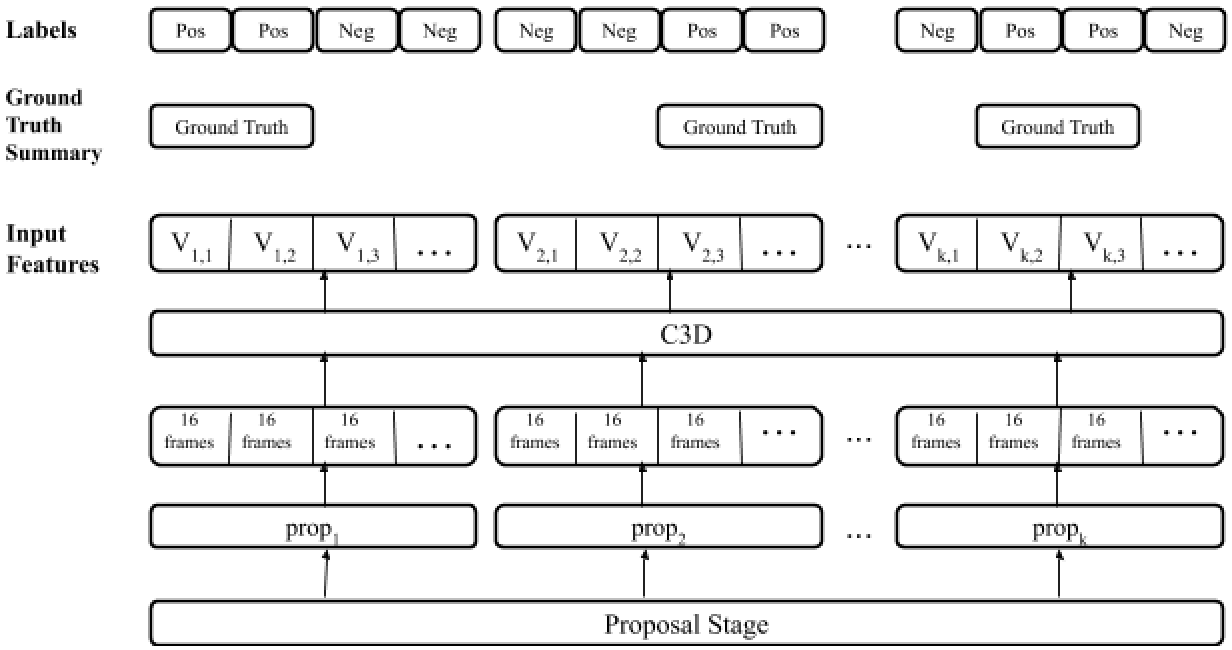


Figure 11: Generation of visual features for Content Refinement stage.

As compared to the original methodology, each feature, represents 16 frames within the event. Let the set of proposals be $props = \{prop_1, prop_2, .., prop_k\}$. For each proposal, $prop_k$, we generate the visual features $V_k = \{ v_{k,1}, v_{k,2}, ..., v_{k,q}\}$ where each $v_{k,q}$ are the visual features generated by C3D network with an input of 16-frames.

5 Experimental Results

5.1 Original Architecture

The table below summarizes the results that have been obtained in the original architecture.

Method	Precision	Recall	F-Score
event-vsLSTM (with audio)	0.414	0.389	0.384
event-H-RNN (with audio)	0.257	0.594	0.355
Multimodal	0.470	0.457	0.459

Table 2: Experimental results of the original architecture

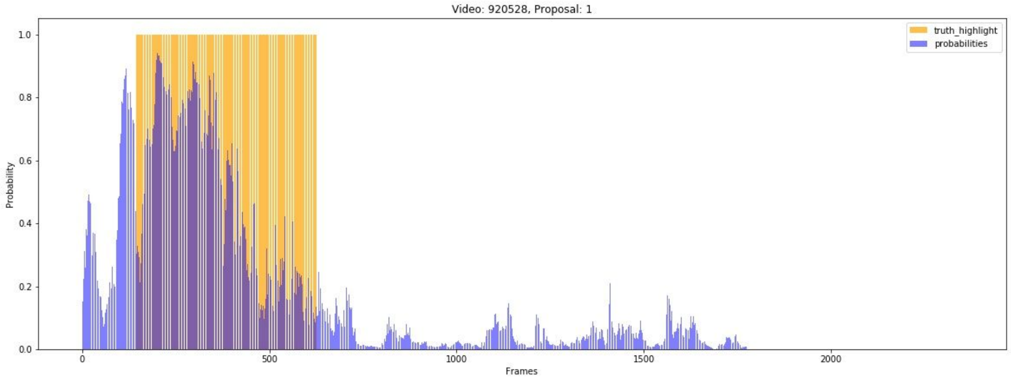
5.2 Proposed Experiments

The table below summarizes all the results from the experiments that we have proposed.

Exp	Method	Precision	Recall	F-Score
1	Multimodal (GoogleNet)	0.470	0.457	0.459
2	Multimodal (C3D)	0.532	0.416	0.464
3	C3D Content Refinement on Proposal Stage	0.306	0.379	0.336
4	MultiModal (GoogleNet) Summarization only	0.414	0.541	0.465
5	Multitask, Individual Loss Summarization only	0.421	0.659	0.514
6	Multitask, Joint Loss (Sum) Summarization only	0.478	0.632	0.545
7	Multitask, Individual Loss Content Refinement only	0.232	0.465	0.310
8	Multitask, Joint Loss (Sum) Content Refinement only	0.245	0.439	0.314
9	Multitask, Joint Loss (Sum) Summarization Threshold 0.5 Content Ref. Threshold 0.35	0.228	0.411	0.287
10	Multitask, Joint Loss (Sum) Summarization Threshold & Content Ref. Threshold 0.2	0.212	0.489	0.296

Table 3: Experimental results of the new proposed experiments

Comparing the original Multimodal approach with GoogleNet (Experiment 1) based features and the proposed Multimodal approach with C3D features (Experiment 2), we observe that there is a notable precision gain. In addition, based on our earlier studies on the distribution of probabilities, we have seen a better distribution of the probability for the output of the content refinement stage as shown in the comparison between Figure 12 and Figure 13 below.



Orange highlights the truth region

Figure 12: Probability distribution of Content Refinement stage of GoogleNet based features

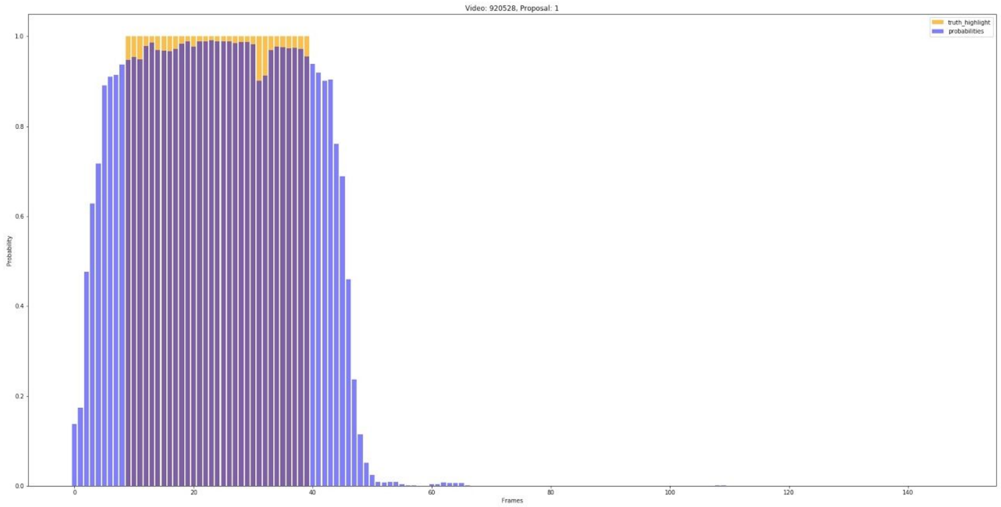


Figure 13: Probability distribution of Content Refinement stage of C3D based features

Experiment 3 is a modification to the input to the Content Refinement stage where we now generate the input features on the proposals from the Proposal stage as compared to Experiment 1 and 2 which takes the input proposals from the Summarization stage. In this case, as there are more input proposals fed into this network, we observe that both the precision and recall has decreased. In this case, for precision, there are more false positives and for the recall there are fewer true positives or more false negatives.

Experiment 5 to 10 is conducted in the Multitask setting. Experiment [5, 7] and [6, 8] are conducted in accordance to the proposed experiment in **Section 3.2.2.1** and **3.2.2.2** respectively.

Experiment 5 and 6 are evaluated solely with the predictions for the Summarization stage to determine the positivity. The precision and recall for both experiments has both performed much better than the original architecture which is evaluated in Experiment 4.

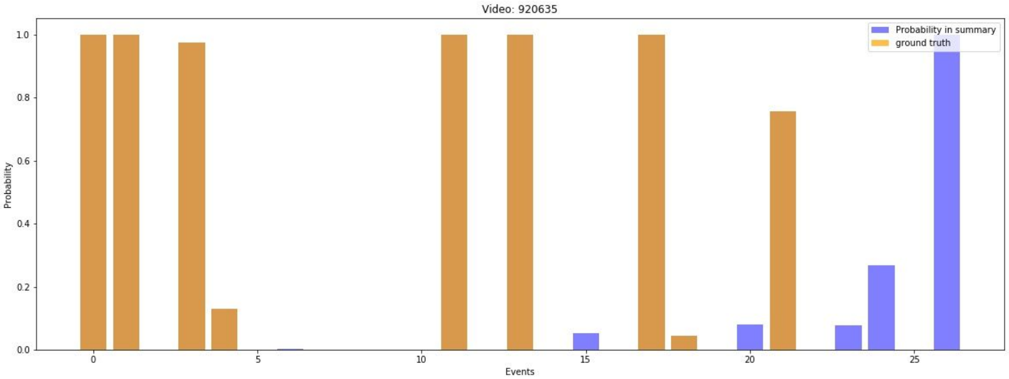
Experiment 7 and 8 are evaluated solely with the predictions of the Content Refinement stage to determine the positivity. The precision for both experiments has performed worse against Experiment 3 but we have gained an improvement in recall.

Experiment 9 and 10 are conducted in accordance to the proposed experiment in **Section 3.2.2.2**. It has the same network as Experiment 6 and 8 but it is evaluated differently. Experiment 9 and 10 takes into account both the output of the Summarization and Content Refinement stage to determine the positivity. Experiment 9 imposes a threshold of 0.5 on Summarization stage to determine positivity and 0.35 for the Content Refinement stage. This threshold is the same threshold used in evaluating for Experiment 1 and 2. We observe a decrease in both the precision and recall. Experiment 10 sets a fixed threshold of 0.2 across both tasks and has obtained better recall as compared to the original experiments.

6 Conclusion and future work

There are more to Multitask Learning architectures than the one experimented in this project. We have yet to explore architectures that involves shared layers [10], cross-stitch networks[11], and etc. There are also many more exploration that can be done on the constraint satisfaction factor of this project for the fixed time duration.

Appendix



Dark Orange highlights the ground truth events
Purple are the probabilities for the proposals to be positive

Figure 14: Plot of probability distribution of Summarization stage for the temporal events.

Bibliography

[1] Zhao, Bin & Liu, Wei & Lu, Xiaoqiang. (2018). HSA-RNN: Hierarchical Structure-Adaptive RNN for Video Summarization. 7405-7414. 10.1109/CVPR.2018.00773.

[2] Kanehira, Atsushi & Van Gool, Luc & Ushiku, Yoshitaka & Harada, Tatsuya. (2018). Viewpoint-Aware Video Summarization. 7435-7444. 10.1109/CVPR.2018.00776.

[3] Chu, Wen-Sheng & Song, Yale & Jaimes, Alejandro. (2015). Video Co-summarization: Video Summarization by Visual Co-occurrence. 10.1109/CVPR.2015.7298981.

[4] Fu, Tsu-Jui & Tai, Shao-Heng & Chen, Hwann-Tzong. (2019). Attentive and Adversarial Learning for Video Summarization. 1579-1587. 10.1109/WACV.2019.00173.

[5] Zhang, Yu & Yang, Qiang. (2017). A Survey on Multi-Task Learning.

[6] Meyerson, Elliot & Miikkulainen, Risto. (2017). Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering.

[7] Berrani, Sid Ahmed & Boukadida, Haykel & Gros, Patrick. (2013). Constraint Satisfaction Programming for Video Summarization. Proceedings - 2013 IEEE International Symposium on Multimedia, ISM 2013. 195-202. 10.1109/ISM.2013.38.

[8] Shi, Lu & King, Irwin & Lyu, Michael. (2019). Video Summarization Using Greedy Method in a Constraint Satisfaction Framework.

[9] Yong Rui, Anoop Gupta, and Alex Acero. 2000. Automatically extracting highlights for TV baseball programs. In Proceedings of the eighth ACM international conference on Multimedia. ACM, 105–115.

[10] Ruder, S., Bingel, J., Augenstein, I., & Søgaard, A. (2018). Latent Multitask Architecture Learning.

[11] Misra, I., Shrivastava, A., Gupta, A., & Hebert, M. (2016). Cross-Stitch Networks for Multi-task Learning. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3994-4003.

[12] Tran, D & Bourdev, L & Fergus, Rob & Torresani, L & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. IEEE Int. Conf. Comput. Vis. 4489-4497.