CS285 Extended Abstract: (Waka) FLOCKAS: Summarization Network
Aatif Jiwani, Dhruv Jhamb, ilian Herzi

When it comes to the domain of videos, an essential yet often understated problem is working with large video datasets. As the amount of collected visual media increases, video datasets continue to grow in size. Processing these videos becomes a computational bottleneck. Video summarization aims to address this problem by representing videos in a more efficient manner under the constraint of minimizing the loss of semantic information. However, finding which frames best summarize the semantic information in a video is difficult to formalize in an objective and difficult to solve once an objective is chosen. In this paper, we attempt to design said objective and a training procedure that captures more general semantic information of videos by training a single deep summarization model on two different but related problems; one explicitly for video summarization framed as an unsupervised sequential decision-making problem and another for a supervised video captioning problem.

In this project, we break the overall objective into two tasks, a summarization task and a captioning task, both training the same deep summarization network (DSN). For summarization, we modify the pipeline designed by Zhou et al. that uses a deep reinforcement learning training procedure, in particular a variant of the REINFORCE algorithm. We also modify video summary reward functions designed to capture video diversity and representation and introduce new video summary reward functions that aim to capture time deltas and objectness. Additionally, we create a new architecture that produces different frame representations. For the video captioning task, we train a captioning model that uses the summaries from the DSN to predict BERT pre-trained embeddings of the video reference sentences and the reference words themselves, capturing both the high-level and low-level semantic language features. We combine these contributions into an end-to-end training pipeline called FLOCKAS (Fast LSTM Object Captioning top-K Actions Summarization), which uses a policy to predict probabilities of selecting a given frame and then takes actions to form caption summaries based on those probabilities. To reiterate, these summaries are then rewarded in the RL training task and used as inputs to a captioning model in the caption prediction task.

Our contributions are as follows:

1. Addition of a novel supervised component for training the DSN that utilizes a new captioning model and the MSR-VTT dataset.
2. Improved video frame features for the datasets used as we switched from using GoogLeNet to ResNeXt-50.
3. A new training pipeline that simultaneously different objectives, specifically by oscillating between the video summarization task and video captioning task.
4. Evaluated our method on the summarization datasets to compare to existing approaches and achieve state of the art performance

FLOCKAS achieved state of the art performance on both the SumMe and TVSum datasets. FLOCKAS achieved an F-score of 45.5% on SumMe, which is over 4 percentage points better than the next best ***unsupervised*** summarization approach and over 3 percentage points better than the next best ***supervised*** summarization approach. FLOCKAS achieved an F-score of 58.9% on TVSum, which is 1.3 percentage points better than the next best ***unsupervised*** summarization approach and 0.8 percentage points better than the next best ***supervised*** summarization approach.

(Waka) FLOCKAS: Summarization Network

Aatif Jiwani[1], Dhruv Jhamb[1], and ilian Herzi[2]
University of California, Berkeley
{aatifjiwani, dhruvjhamb, ilianherzi}@berkeley.edu

Abstract— **Finding which frames best summarize the semantic information in a video is difficult to formalize in an objective and difficult to solve once an objective is chosen. In this paper, we attempt to design said objective and a training procedure that captures more general semantic information of videos by training a single summarization model on two different but related problems; one explicitly for video summarization framed as an unsupervised sequential decision problem and another for video captioning framed as a supervised caption prediction problem. For the summarization task, we introduce a training pipeline using deep reinforcement learning and rewards designed to capture both video diversity, representation, and objectness. For the video captioning task, we train a captioning model to predict BERT pretrained embeddings of reference sentences and the references themselves capturing both the high and low level semantic language features. We combine these contributions into an end-to-end training pipeline called FLOCKAS (Fast LSTM Object Captioning top-K Actions Summarization), which uses a policy to predict probabilities of selecting a given frame and then takes actions to form caption summaries based on those probabilities. These summaries are then rewarded in the RL training task and used as inputs to a captioning model in the caption prediction task. Using FLOCKAS on SumMe, TVSum and MSR-VTT, we achieve state of the art F-scores on the unsupervised video description task for SumMe and TVSum.**

I. INTRODUCTION

When it comes to the domain of videos, an essential yet often understated problem is working with large video datasets. As the amount of collected visual media increases, video datasets continue to grow in size. Processing these videos becomes a computational bottleneck [1]. Video summarization aims to address this problem by representing videos in a more efficient manner under the constraint of minimizing the loss of semantic information. Video summarization is defined as selecting a sequence of key frames from the original video such that those key frames “summarize” the video. By improving the quality of video summaries, these summaries can be used in place of the original videos, resulting in efficient data storage, processing, and faster training for video-based models. Video summarization will result in an improvement in the usability of large video datasets.

One particular application of video summarization that we are interested in is applying it to the video captioning task. MSR-VTT is comprised of 10,000 videos with more than 40 hours of total video runtime. Large datasets like MSR-VTT serve as a computational barrier and creating a methodology for representing those videos in a more efficient format would address this barrier.

Reinforcement Learning can be applied to the task of unsupervised video summarization by training a policy to predict whether or not to include the ith frame given its actions for frames $[0, i-1]$. Zhou et al. [2] create an unsupervised training pipeline for video summaries using a deep reinforcement learning training framework that trains a deep summarization network (DSN). We note that with this framework we can make adjustments to the reward function and architecture as a straightforward improvement of this baseline video summarization problem. However, another extension is to combine this pipeline with similar video-related problems that should intuitively should help the DSN model capture more general information. One such problem is to also train the DSN on a video captioning task, where we hypothesize that a DSN that does well in video captioning should have features that are also useful to the unsupervised video summary RL objective.

In this paper, we modify the pipeline of the unsupervised video summarization RL training procedure, in particular a variant of the REINFORCE algorithm, by modifying existing video summary reward functions, introducing new video summary reward functions, and changing the architecture to produce different frame representations. We also introduce a video captioning supervised pipeline that incorporates the video summary extracted from the DSN network. Both the supervised and unsupervised training tasks share the same DSN and are trained in parallel. We call the additional captioning network ontop of the DSN network the *captioning model*. Specifically, we train the DSN on the video summary datasets SumMe and TVSum, and in parallel train the DSN and captioning model on the MSR-VTT dataset using a modified CIDEr score and cross entropy loss.

Overall the contributions of this project are:

1) Addition of a novel supervised component for training the DSN that utilizes a new captioning model and the MSR-VTT dataset.
2) Improved video frame features for the datasets used as we switched from using GoogLeNet to ResNeXt50.
3) A new training pipeline that simultaneously different objectives, specifically by oscillating between the video summarization task and video captioning task.
4) Evaluated our method on the summarization datasets to compare to existing approaches and achieve state of the art performance

[1] Undergraduate Researcher in Professor John Canny’s Group at UC Berkeley
[2] Graduate Master’s Researcher in Professor John Canny’s Group at UC Berkeley

II. Related Work

A. Video Summarization:

Zhou et al. create an unsupervised training pipeline for video summarization using deep reinforcement learning [2]. They define a good summary as one that is both diverse in its selected frames and contains frames representative of their respective surrounding frames. Building on their code, we used the existing deep summarization network (DSN) for video summarization. The DSN is comprised of an encoder (CNN) and a decoder (LSTM). The encoder is used for feature extraction on the frames of a video and the decoder produces probabilities that are used for selecting key frames for the final video summary.

Both unsupervised and supervised techniques have been used to approach the problem of video summarization. In terms of unsupervised approaches, the use of reinforcement learning is common [2, 3]. Zhou et al. created an end-to-end, RL-based framework for training the DSN. They used an encoder-decoder model for the DSN and constructed a novel diversity-representativeness reward to create a fully unsupervised learning method. Chen et al. created a similar Encoder-Decoder framework with a reward based on diversity and textual discrepancy. Recent supervised approaches use the concept of attention to select important frames [4, 5]. Lee at al. used important objects to aid them in summarizing videos [6]. Zhang et al. developed an approach to transfer structures of known video summaries to new videos with similar topics [7]. Song et al. used images related to the video title to learn important visual concepts for summarizing videos [8]. Zhao et al. used group sparse coding to represent a video as a dictionary and then use the learned dictionary to generate the video summary [9].

B. Video Captioning:

Video captioning is one of the most popular applications using video datasets. Video captioning is a field that lies at the intersection of Computer Vision (CV) and Natural Language Processing (NLP). Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been successful when used in an Encoder-Decoder architecture. Beyond this, Encoder-Decoder frameworks with two LSTMs have also been built [10]. Just as attention has been used for video summarization, it has also been used in video captioning. Kelvin Xu et al. first introduced an attention based model for generating captions [11]. Yao et al. created a model that incorporates a temporal attention mechanism [12]. Jun Xu et al. released MSR-VTT as a large-scale video captioning dataset [13].

C. Reinforcement Learning:

Reinforcement learning, specifically deep reinforcement learning, has gained popularity recently. As students in CS 285, we have learned various methodology and techniques for deep reinforcement learning. Mnih et al. used a deep CNN to approximate Q-functions on the task of playing Atari games [14]. RL has also been used for computer vision tasks [11, 15]. For the task of video summarization specifically, reinforcement learning has been used by approaches that utilize policy gradients to minimize a reward [2, 3, 8].

III. Details of our approach

A. Problem Setup

Policy gradient is a model-free version of RL that attempts to find a policy given a reward with unknown dynamics. In this problem, each video represents some underlying Partially observable Markov decision process (POMDP) related to the unknown dynamics of the world captured by the video where the sequence of images represents a sequence of observations.

The RL problem that we actually seek to model however is a bit different from the dynamics that generated the video. Our goal is to frame the problem in a way that captures the "goodness" of the semantic information captured.

We model the states as a summarized video where each set of actions on all the frames can be considered a single action that moves us from our video to a video summary with a horizon of 1. We then obtain a reward that measures the "goodness" of this new video summarization and reset our episode with the initialized state as the original video. Note that "goodness" in this problem is explicitly measured by the designed reward functions where we measure the temporal difference between frames, a representativeness reward (selected frames should be similar to their surrounding frames), and the objectness in the frames (object detection).

The dynamics of our MDP are as follows: for a general video $\mathcal{V}$, with observed frames $\{v_t\}^T$, we seek to

$$\max_{a_1,a_2,\ldots,a_T} E_{p_\theta}[r(\{v_0, a_0, ..., v_t, a_T\})]$$

$$s.t.\ \mathcal{V}_i^{(t+1)} = f(\mathcal{V}_i^{(t)}, a_0, ..., a_T)$$

where we try to learn a policy, π_θ the DSN network,

$$\pi_\theta : v_i \in \mathcal{V} \to [0,1]^T$$

that maximizes the reward for transitioning to another summary.

This problem could in theory be solved using dynamic programming; however, it is a combinatorially complex problem so instead we decide to use RL to solve it. We then simplify this video summarization problem by grouping frames into intervals using the K-level task splitting (KTS) [2, 16] algorithm to generate change points, which are frame indices that indicate a significant change in the video (i.e. changing scenes). Using the trained policy from the above objective, we approximate the probability of picking an interval by taking the mean of the predicted frame probabilities of all frames in that interval. This sufficiently reduces the complexity of our problem so we can solve the rest via dynamic programming. In particular, we model this as a $0-1$ knapsack problem, where the weight is the temporal length of the change point and the value is the probability predicted by the DSN network.

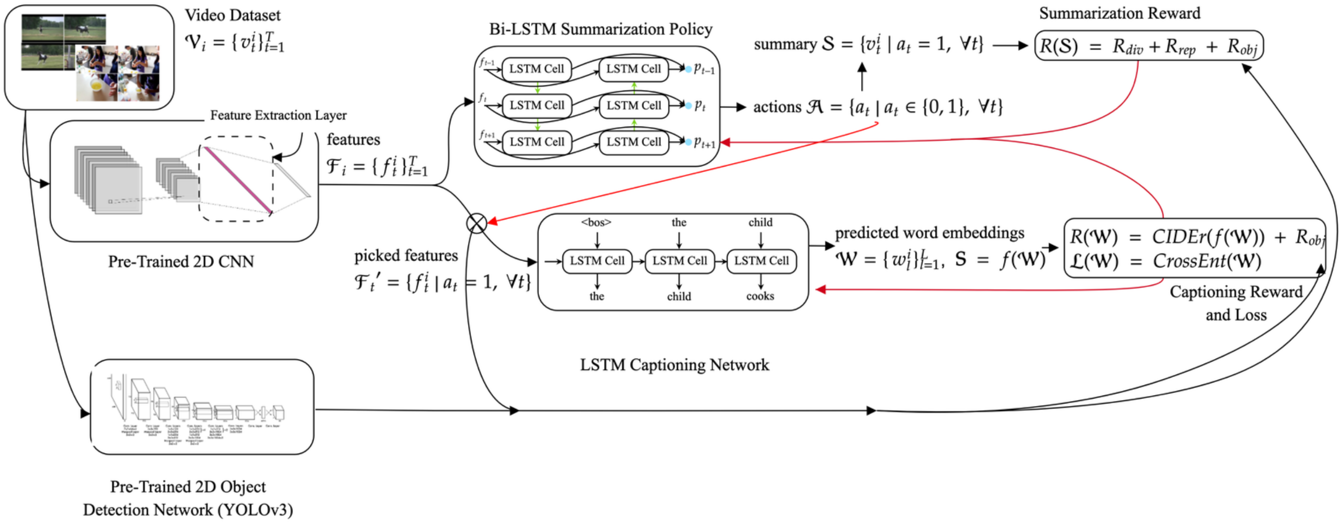
Video Dataset
$\mathcal{V}_i = \{v_t^i\}_{t=1}^T$
Feature Extraction Layer
features
$\mathcal{F}_i = \{f_t^i\}_{t=1}^T$
Pre-Trained 2D CNN
Bi-LSTM Summarization Policy
f_{t-1}
f_t
f_{t+1}
LSTM Cell
p_{t-1}
p_t
p_{t+1}
summary $\mathcal{S} = \{v_t^i \mid a_t = 1, \forall t\}$
actions $\mathcal{A} = \{a_t \mid a_t \in \{0,1\}, \forall t\}$
Summarization Reward
$R(\mathcal{S}) = R_{div} + R_{rep} + R_{obj}$
picked features
$\mathcal{F}_t' = \{f_t^i \mid a_t = 1, \forall t\}$
<bos>
the
child
cooks
LSTM Captioning Network
predicted word embeddings
$\mathcal{W} = \{w_l^i\}_{l=1}^L,\ \mathrm{S} = f(\mathcal{W})$
$R(\mathcal{W}) = CIDEr(f(\mathcal{W})) + R_{obj}$
$\mathcal{L}(\mathcal{W}) = CrossEnt(\mathcal{W})$
Captioning Reward and Loss
Pre-Trained 2D Object Detection Network (YOLOv3)

Fig. 1: Overview of our FLOCKAS pipeline where we combine the summarization network, captioning model, and object detection network

Using this setup, we can attempt to improve video summarization by tackling three different areas of this sequential decision making objective: 1. changing the frame-level features, v_i, 2. sculpting the reward function to better capture the "significant semantic information", and 3. changing the interval creation and value labeling procedure in the dynamic programming part of this problem. In this paper we tackle the first two points and hope to pursue the third in future work.

Additionally we train a video captioning problem using the DSN network, π_θ, as input to our captioning model,

$$g_\phi : \mathbb{R}^{k_{frame\ embedding}} \rightarrow \mathbb{R}^{k_{captioning\ embedding}}$$

The supervised video captioning objective is to maximize similarity between the captioning model's predicted embedding and pretrained BERT embeddings for the reference sentences for video $\mathcal{V}$ which we'll call $B(\cdot)$. $B(\cdot)$ is a function that takes a reference sentence and produces a sentence embedding through a pre-trained BERT model.

$$\max_{\phi,\theta}\ CIDEr(g_\phi(\pi_\theta(\mathcal{V})), B(\mathcal{V}_{ref}))$$

We also attempt to balance this high-level semantic representation with a prediction task of the words in the reference sentence by calculating a cross entropy loss that's subtracted from the above reward.

B. Feature Extractor

[2] use GoogLeNet as the encoder in the DSN. GoogLeNet introduced the inception module, which increased the width of the neural network by having multi-sized filters operating at the same level. GoogLeNet is 22 layers with 9 inception modules. In 2016, ResNeXt-152 outperformed GoogLeNet in ILSVRC 2016 and is based upon the highly regarded ResNet which solved the notorious vanishing gradient problem by introducing residual / skip connections. We chose to utilize a ResNeXt model to replace the GoogLeNet as the frame encoder for the DSN. We chose to use a smaller version of ResNeXt, ResNeXt-50, for extracting features from input videos, mainly for efficiency.

C. Captioning

We additionally introduce a captioning task as part of the training pipeline for the DSN. We encode ground truth captions for videos and the outputs of our captioning model as embeddings so we chose to utilize a pre-trained BERT to generate embeddings of the reference sentences. Our targets for training are these reference sentence embeddings.

In our pipeline, we want to use the predicted video summary to effectively caption the video. The intuition is that a good video summary that captures the essence of the video should be able to predict good captions in the BERT pre-trained sentence embedding space. We use the DSN to select key frames that summarize the video. Then we use those indices to mask the video frame features (from ResNeXt-50) such that we only pass in the features for the frames selected to be part of the summary into our captioning model.

For our captioning model, we use an LSTM to capture the temporal structure between BERT word embeddings. Using the word embeddings, we then construct sentence embeddings by taking the mean of the word embeddings. These are then compared to BERT embedded ground truth captions using a modified version of the CIDEr score.

CIDEr scores have become a popular metric in measuring how similar a predicted sentence is to the set of reference sentences for a single video, or commonly denoted as a single image in literature. Typically, CIDEr scores are computed with vector representations of a set of n-grams which is a collection of 1 to 4 words, but this range can vary. Vedantam et al. defines the CIDEr score for a particular set of n-grams to be:

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i)^T g^n(s_{ij})}{\|g^n(c_i)\|\|g^n(s_{ij})\|}$$

where c_i is the predicted sentence and S_i is the set of all reference sentences for image I_i. $g^n(c_i)$ denotes the vector representation of all the n-grams in c_i and the same follows for $g^n(s_{ij})$.

However, in this paper we use a modified version of the CIDEr score. The original CIDEr metric uses n-gram representations, but because we can in theory use any statistically-motivated vector representation to measure the similarity between sentences, we decided to implement CIDEr as follows:

$$CIDEr(c_i, S_i) = \frac{1}{m}\sum_j \frac{B(c_i)^T B(s_{ij})}{\|B(c_i)\|\|B(s_{ij})\|}$$

where again $B(\cdot)$ is a function that takes a sentence as an argument and produces a sentence embedding through a pre-trained BERT model. A slight caveat to this is that although we compute $B(s_{ij})$ using a BERT model, the sentence embedding for the predicted sentence, $B(c_i)$ is learned through the second head of the captioning network.

In addition to this modified CIDEr score we also compute a mean cross entropy loss on each word in the reference sentences and the distribution over the vocabulary resulting from an inverse transform that maps pretrained BERT embeddings to words. Combining the CIDEr score with the cross entropy loss intuitively tries to balance rewards for good high level semantic representation and good low level detail representations. We define the captioning reward function as the sum of those two components.

With the introduction of the captioning prediction task we've essentially added another training task on top of our model-free RL problem.

Additionally, because these two objectives are fundamentally different the summarization dataset and captioning dataset are also fundamentally different, we could not find a single dataset with both video sumaries and video captions. The datasets used by [2] for the video summarization task (SumMe, TVSum) contain key frames as ground truths and the captioning task (MSR-VTT) uses reference captions as the ground truths.

We note however that while the objectives are different, the model should be able to learn a representation that does well in both tasks. To reiterate, we introduced a new training scheme to oscillate between training both summarization and video captioning using the same DSN model and additional task related model components. Since we found each task equally important we decided to alternate every iteration between the two tasks but in theory depending on the importance of the objective we could alternate unfairly.

D. Reward Function Design: Diversity, Representativeness, Time Delta, Object Detection

For our experiments we combined three functions that rewarded video summaries for three different types of semantic representations: diversity between frames in the summary, similarity between frames in the summary, and distributions with large support over object classes that could be found in the video.

1) Diversity: We model the diversity reward as follows

$$R_{div} = \frac{1}{|\mathcal{V}|(|\mathcal{V}|-1)}\sum_{t\in\mathcal{V}}\sum_{t'\in\mathcal{V}} d(v_t, v_{t'})$$

where we encourage the model to reward frames that are different.

2) Representativeness:

$$R_{rep} = \exp(-\frac{1}{T}\sum_t^T \min_{t'\in\mathcal{V}} d(v_t, v_{t'}))$$

Simultaneously we try to balance the diversity reward score with a function that rewards similarity between frames in the summary and frames in the original video; The better the selected frames capture all the information in all frames of the video the larger the reward.

Note that for both the diversity and representativeness rewards $d(v_t, v_{t'})$ is the L_2 norm of the embedded vector frame representations.

3) Time Delta: We modified the aggregate reward function by adding the negative average time delta between selected key frames. This encourages the DSN to learn the clusters of frames that best represent the action in the frame.
$\mathcal{V} :=$ set of selected frame indices
$T :=$ constraint on the total number of frames

$$R_{\text{new}} = R_{\text{div}} + R_{\text{rep}} + R_{\text{time}}$$

$$R_{\text{time}} = \frac{1}{|\mathcal{V}|}\sum_{t,t'\in zip(\mathcal{V}[:-1],\mathcal{V}[1:])} t - t'$$

where $R_{\text{time}} \leq 0$.

4) Object Detection: Defining a "good" summary is often difficult because such a question only makes sense in the context of what the summary is being used for: classification, video description etc. Intuitively though, good summaries for videos are ones that contain nearly all of the semantic information of the original video with significantly less stored information compared to the original video. We decided that another signal that could help the model attune to good semantic information was the number and type of objects in a frame. For each frame, we label encode (similar to a variant of positional encoding) the type of objects scaled by the number of objects in that frame and add this as a reward to the both the summarization task and captioning task. More formally, where each object in a particular is labeled by $i \in \{1, ..., N_{objs}\}$ and there are k_i count of that object then the object encoded frame would be

$$encoding_i = [\cos(i)/N, \sin(i)/N]^T * k_i$$

Refer to Figure 2 for an example of the kinds of objects detected in the MSR-VTT captioning dataset.

The objects extracted from each frame come from a COCO-pretrained YOLOv3 [17] model. We selected this

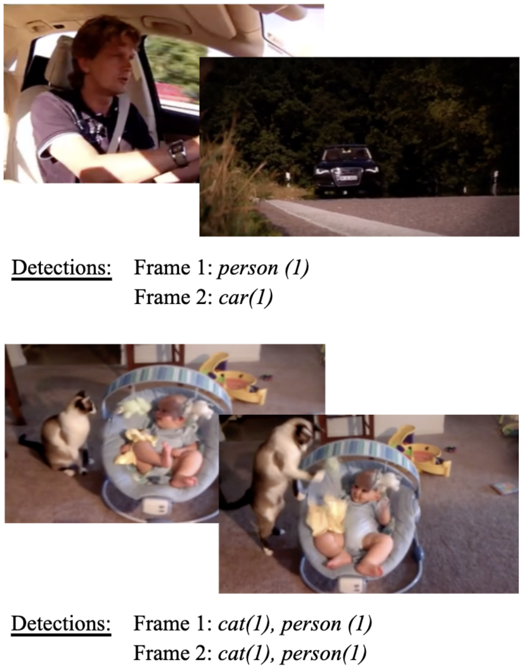

Fig. 2: Visual example of objects detected with YOLOv3 on sample MSR-VTT videos

```
Algorithm 1: Object detection encoding procedure
Result: Encodings per frame
initialize encoding array;
for each frame $v_i$ in $\mathcal{V}_i$ do
    for object, count in $v_i$ do
        embedding = $[\frac{\cos(obj)}{N_{obj}}, \frac{\sin(obj)}{N_{obj}}]k_i$;
        append embedding to encoded array;
    end
end
```

model over other pretrained models because it's SSD architecture allows for efficient and fast object detection while achieving good performance on object detection datasets, like COCO.

E. Top K Frames

One issue that we encountered is that sometimes the model was overly conservative in its probabilities during the early stages of training since each frame was assigned a small probability, which resulted in a predicted video summarization with no frames. To deal with this, whenever the model predicted a summarization with no frames, instead of selecting frames randomly based on their predicted probability used to a top-k function that picks k frames with the top-k assigned probabilities.

IV. Results

A. Datasets

In order to evaluate our additions to Zhou et al.'s training pipeline, we chose to use two datasets commonly used for the video summarization task: SumMe [18] and TVSum [8]. SumMe consists of 25 videos that range from 1 to 6 minutes, each annotated with at least 15 human summaries. TVSum contains 50 videos of various genres and 20 annotations of shot-level importance scores obtained from crowdsourcing. The videos were collected from YouTube and annotated using Amazon Mechanical Turk. Following the approach of other papers, we convert importance scores to shot-based summaries for evaluation [2, 7, 8]. Since we introduced training the video captioning task to the pipeline, we needed a captioning dataset. We chose to use MSR-VTT because it is a large-scale benchmark for translating video to text [13]. MSR-VTT provides 10,000 video clips, resulting in 41.2 total hours of video and 200,000 clip-sentence pairs in total. Each clip contains around 20 sentence annotations.

B. Metrics

For comparison with other approaches, we followed the protocol from Zhang et al. that was also used in Zhou et al. to use F-score to evaluate generated summaries with ground truth summaries [2, 7]. We used the diversity-representativeness reward from Zhou et al. for training on the video summarization task. This reward factors in diversity by measuring dissimilarity among selected frames in the video summary and representativeness by measuring how similar selected frames are from their surrounding frames from the original video. This reward does not utilize the ground truth labels. For training the video captioning task, we compute a novel reward comprised of two terms: cross-entropy loss and a CIDEr score. In order to gauge captioning performance on MSR-VTT, we refer to both metrics individually. The cross-entropy loss follows the standard definition. We define a modified CIDEr score metric. Traditionally, a CIDEr score is defined as the average cosine similarity between the TF-IDF of candidate sentences and reference sentences [19]. This accounts for both precision and recall. Then the scores from n-grams are combines by a weighted sum. Again, in this paper, we modify the CIDEr score by instead computing the average cosine similarity between the BERT embeddings of candidate and reference sentences. This modifies the scale of the CIDEr score.

C. Experiments

As a note, one modification we made was selecting the highest F-Score during training after 10 epochs have past.

DR-DSN (Baseline) : To create a baseline for our proposed approach, we ran an experiment that trained the DSN solely on the video summarization task using the diversity-representativeness reward (Kaiyang Zhou et al.) on both SumMe and TVSum. For this experiment, we kept everything the same as Zhou et al. and are using it as a reference to see the effect our additions have on performance. We replicated Zhou et al.'s approach and achieved an F-Score of 42.99% on the SumMe dataset and 57.77% on the TVSum dataset. However, we decided to compare our results to the published results of the paper as seen in table I.

Experiment	SumMe	TVSum
DR-DSN (Baseline)	41.4	57.6
DR-DSN (Baseline) with ResNeXt features	42.9	56.7
DR-DSN (Baseline) with time delta reward	**46.1**	59.1
FLOCKAS$_{w/o\ object\ detection}$	42.5	**59.7**
FLOCKAS$_{topk}$	43.6	58.1
FLOCKAS	45.5	58.9

TABLE I: Results of experiments. Each entry is an F-Score. DR-DSN (Baseline) results are taken from Zhou et al.

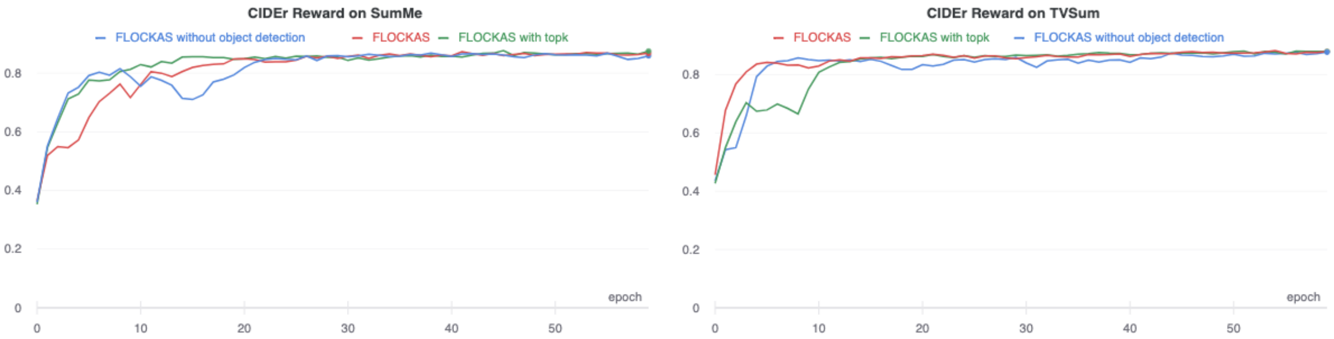

Fig. 3: CIDEr Score on both the SumMe and TVSum datasets

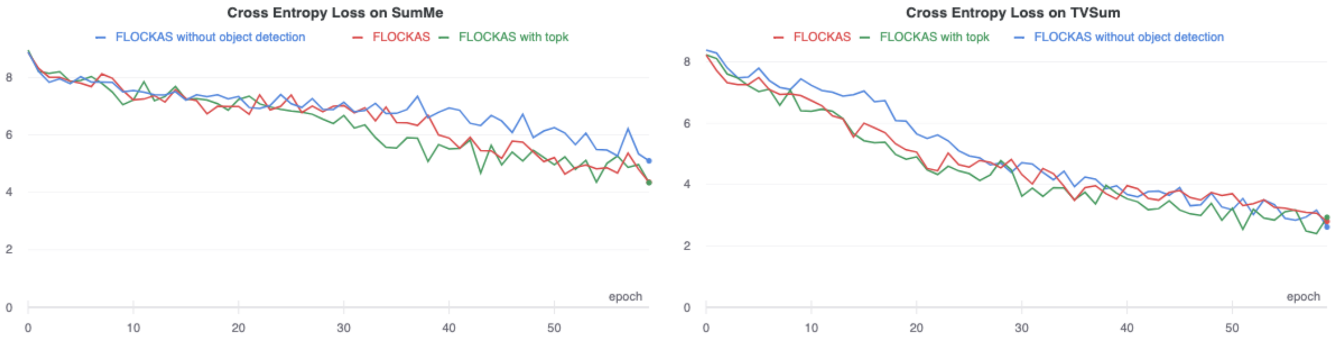

Fig. 4: Cross Entropy Loss on both the SumMe and TVSum datasets

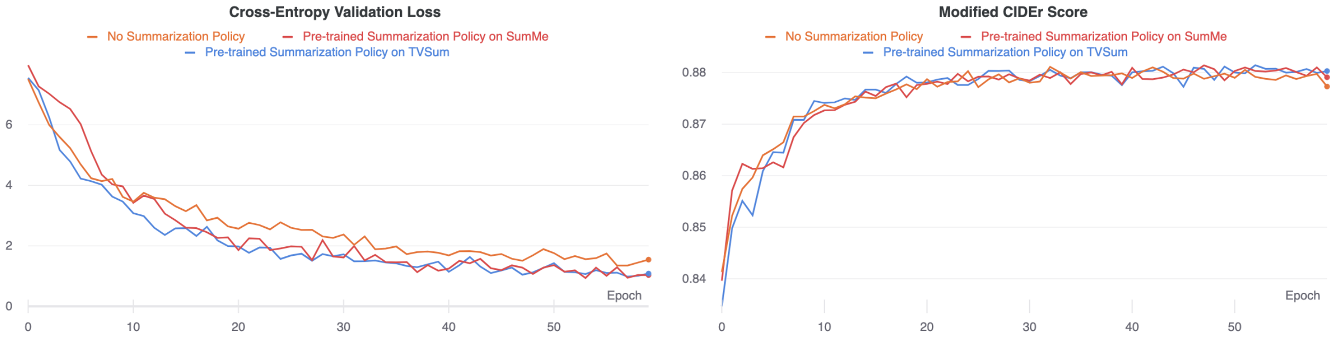

Fig. 5: Validation Loss and Modified CIDEr Score for training only a captioning network

DR-DSN (Baseline) with ResNeXt features: One of the modifications we made was to the CNN portion of the DSN that is used for extracting features from video frames. We switched from Zhou et al.'s use of GoogLeNet to ResNeXt for feature extraction, hypothesizing that ResNeXt would result in high quality features that would lead to performance gains. To test this, we ran an experiment training only the video summarization task with a ResNeXt model instead of GoogLeNet as the feature extractor network. When using ResNeXt and the same pipeline from Zhou et al., we achieved an F-Score of 42.85% on SumMe and 56.65% on the TVSum dataset. These were slightly lower than when GoogLeNet was utilized as the CNN for feature extraction.

DR-DSN (Baseline) with added time delta reward: We propose a subtle modification to the reward function from Zhou et al. where we append a time delta reward

Method [2]	SumMe F-Score	TVSum F-Score
Video-MMR	26.6	-
Uniform sampling	29.3	15.5
K-medoids	33.4	28.8
Vsumm	33.7	-
Web image	-	36.0
Dictionary selection	37.8	42.0
Online sparse coding	-	46.0
Co-archetypal	-	50.0
GAN$_{dpp}$	39.1	51.7
DR-DSN	41.4	57.6
FLOCKAS	**45.5**	**58.9**

TABLE II: Comparison of unsupervised approaches, results taken from Zhou et al.

Method	SumMe F-Score	TVSum F-Score
Interestingness	39.4	-
Submodularity	39.7	-
Summary transfer	40.9	-
Bi-LSTM	37.6	54.2
DPP-LSTM	38.6	54.7
GAN$_{sup}$	41.7	56.3
DR-DSN$_{sup}$	42.1	58.1
FLOCKAS	**45.5**	**58.9**

TABLE III: Comparison of supervised approaches, results taken from Zhou et al.

that incentivizes picking frames closer together temporally. We ran the pipeline with this modified reward function on both SumMe and TVSum. When adding the described novel time delta reward from above, we achieved significant performance improvements on both video summarization datasets. With the modifications being the added time delta reward term and the use of ResNeXt as the CNN used for feature extraction, we achieved 46.06% on the SumMe dataset and 59.12% on the TVSum dataset.

Training only captioning task: In order to evaluate the performance of the captioning model we added, we trained our captioning model (LSTM) on a set of 20 MSR-VTT videos.

FLOCKAS$_{w/o\ object\ detection}$ (Training both tasks, sampling MSR-VTT videos): We have our proposed pipeline with all our modifications except for the object detection reward. Here, the MSR-VTT dataset is sampled when training the video captioning task. When training using our proposed oscillation method that alternates between training the video summarization and video captioning task, we achieved an F-Score of 42.5% on the SumMe dataset and 59.7% on the TVSum dataset. While adding oscillation slightly worsened performance on SumMe, it improved performance on TVSum by a significant amount.

FLOCKAS (Training both tasks, sampling MSR-VTT videos, object detection reward): Extending beyond the previous experiment, we ran our finalized pipeline, which included the object detection reward specified in the approach section. This resulted in an F-Score of 45.5% on the SumMe dataset and 58.9% on the TVSum dataset. We found that adding in object detection significantly improved performance on SumMe, while performance on TVSum slightly worsened.

FLOCKAS$_{topk}$ (Always selecting top-k and sampling MSR-VTT videos, object detection reward): Frames are chosen to be part of a summary by sampling in accordance with the probabilities outputted by the DSN. If zero frames are selected for a summary, we added a modification to sample the top k frames, where k is 15% of the total number of frames in the video. We wanted to see how always picking the top k frames compared to the the sampling of frames. We found that always selecting the top k frames instead of sampling resulted in worse performance on both SumMe and TVSum.

D. Comparison with unsupervised approaches

FLOCKAS refers to our final pipeline, which includes all of our proposed modifications. To summarize and reiterate, it includes the use of ResNeXt features, training oscillation between summarization and captioning, and an object reward. When compared to the unsupervised approaches listed in Table 2 (from Zhou et al.), we can see that FLOCKAS outperforms other unsupervised approaches by a significant margin on both the SumMe dataset and TVSum dataset. Specifically for SumMe, we outperform the next best model, DR-DSN (Zhou et al.), by over 4 percentage points. For TVSum, FLOCKAS outperforms the next best model by 1.3 percentage points and beats other approaches by at least than 7 percentage points.

E. Comparison with supervised approaches

When compared to the supervised approaches listed in Table 3 (from Zhou et al.), we can see that FLOCKAS outperforms other supervised approaches by a significant margin on both the SumMe dataset and TVSum dataset. On SumMe, FLOCKAS achieves 3.4 percentage points above the next best supervised approach, which is more than double the improvement the DR-DSN$_{sup}$ made when compared to GAN$_{sup}$. With regards to TVSum, the improvement is not as drastic, but FLOCKAS still outperforms the next best model, DR-DSN$_{sup}$, by 0.8 percentage points.

F. Training Captioning Network using Pre-Trained Summarization Policy

In many video captioning tasks, especially tasks that concern lengthy videos, processing all the video features in a captioning network can serve as a severe bottleneck. For example, although attention networks like Transformers have the ability to actively ignore certain features, it still must accept and process all of the features. With most video captioning datasets, this can prove to be an issue with computational resources as DRAM and VRAM can fill up fast.

Here, we show that the use of a captioning network *and* a summarization policy can reach the same performance as a standalone captioning network, while simultaneously reducing the number of features for the network to process.

We ran 3 experiments: (1) Training only a captioning network standalone, (2) Training a captioning network with the assistance of the summarization policy pre-trained on SumMe, and (3) Training a captioning network with the assistance of the summarization policy pre-trained on TVSum. The primary metrics for this experiment are shown in 5.

V. Analysis

As per 5, we highlight two primary observations:

1) All experiments are able to reach a maxima Modified CIDEr reward of around 0.88. This observation implies that the summarization policy is able to filter unwanted features and yet still maintain the same performance. This widens the computation bottleneck as the captioning network need not all the features to achieve certain performance.
2) Experiment 1, the captioning network without the support of a summarization policy, fails to reach the same minimum loss achieved by Experiments 2 & 3. Although Cross-Entropy loss is often not a primary measure of caption generation quality, this is indeed an indicator that removing useless features enhances the ability to optimize the supervised captioning objective.

Finally, based on our general FLOCKAS experiments, because TVSum is a collection of YouTube video and not a hand-picked, curated set of videos, these represent videos ”in the wild” where there is a large variance in the distribution of different objects. We think this large variation coupled with the existent difficulties in object detection (occlusion, scale variance, etc.) is we think that this makes the relative objectness computed by the object detection reward somewhat difficult to capture any useful semantic signal at all.

VI. Conclusion

A. Future Work

For future work, we would be interested in exploring more benchmarks for evaluating our performance. When working with small datasets like SumMe and TVSum, it is difficult to determine how much of an improvement in F-Score is statistically significant. We also would want to compare our approach with the existing state of the art approaches for both video summarization and video captioning, specifically looking at approaches that use attention-based models.

B. Contributions

As a group, we met regularly and had weekly work sessions. We found that being organized and having a timeline helped us work on the project. Specific contributions are detailed below.

Aatif Jiwani:
Added support for the MSR-VTT captioning dataset and the SumMe & TVSum summarization datasets, switched to ResNeXt for feature extraction, and ran experiments for final paper.

Dhruv Jhamb:
Set up training oscillation, added LSTM for captioning model, created time delta reward, and ran experiments for the milestone and final report.

Ilian Herzi:
Worked on ResNeXt feature extraction pipeline, BERT feature extraction, object detection using YOLO, and adding the object detection reward, CIDEr score & modified CIDEr score.

VII. Acknowledgements

We would like to thank the phenomenal course staff of UC Berkeley’s CS 285: Deep Reinforcement Learning. All three of us thoroughly enjoyed Professor Sergey Levine’s lectures and appreciate the GSIs help during office hours. We would also like to thank Ph.D. student David Chan who advises us in Professor John Canny’s group at UC Berkeley.

VIII. Source Code

We plan on continuing work on this project, ultimately leading to a publication. For this reason, our source code is private and will be made publicly available after publishing.

References

[1] M. Ajmal, M. Ashraf, M. Shakir, Y. Abbas, and F. Shah, “Video summarization: Techniques and classification,” pp. 1–13, 09 2012.

[2] K. Zhou and Y. Qiao, “Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,” *CoRR*, vol. abs/1801.00054, 2018.

[3] Y. Chen, S. Wang, W. Zhang, and Q. Huang, “Less is more: Picking informative frames for video captioning,” *CoRR*, vol. abs/1803.01457, 2018.

[4] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[5] N. Ejaz, I. Mehmood, and S. Wook Baik, “Efficient visual attention based framework for extracting key frames from videos,” *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34 – 44, 2013.

[6] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, 2012.

[7] K. Zhang, W. Chao, F. Sha, and K. Grauman, “Video summarization with long short-term memory,” *CoRR*, vol. abs/1605.08110, 2016.

[8] Yale Song, J. Vallmitjana, A. Stent, and A. Jaimes, “Tvsum: Summarizing web videos using titles,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187, 2015.

[9] B. Zhao and E. Xing, “Quasi real-time summarization for consumer videos,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2513–2520, 2014.

[10] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence - video to text,” *CoRR*, vol. abs/1505.00487, 2015.

[11] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” *CoRR*, vol. abs/1502.03044, 2015.

[12] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” 2015.

[13] J. Xu, T. Mei, T. Yao, and Y. Rui, “Msr-vtt: A large video description dataset for bridging video and language,” June 2016.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, “Playing atari with deep reinforcement learning,” *CoRR*, vol. abs/1312.5602, 2013.

[15] X. Lan, H. Wang, S. Gong, and X. Zhu, “Identity alignment by noisy pixel removal,” *CoRR*, vol. abs/1707.02785, 2017.

[16] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, “Category-specific video summarization,” in *ECCV 2014 - European Conference on Computer Vision*, 2014.

[17] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.

[18] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in *ECCV*, 2014.

[19] R. Vedantam, C. L. Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” *CoRR*, vol. abs/1411.5726, 2014.