# Demystifying Multi-Faceted Video Summarization: Tradeoff Between Diversity, Representation, Coverage and Importance

Vishal Kaushal
IIT Bombay
vkaushal@cse.ittb.ac.in

Rishabh Iyer
Microsoft Corporation
rishi@microsoft.com

Khoshrav Doctor
University of Massachusetts, Amherst
kdoctor@cs.umass.edu

Anurag Sahoo
AitoeLabs
anurag@aitoelabs.com

Pratik Dubal
AitoeLabs
pratik@aitoelabs.com

Suraj Kothawade
IIT Bombay
surajkothawade@cse.iitb.ac.in

Rohan Mahadev
AitoeLabs
rohan@aitoelabs.com

Kunal Dargan
AitoeLabs
kunal@aitoelabs.com

Ganesh Ramakrishnan
IIT Bombay
ganesh@cse.ittb.ac.in

## Abstract

*This paper addresses automatic summarization of videos in a unified manner. In particular, we propose a framework for multi-faceted summarization for extractive, query base and entity summarization (summarization at the level of entities like objects, scenes, humans and faces in the video). We investigate several summarization models which capture notions of diversity, coverage, representation and importance, and argue the utility of these different models depending on the application. While most of the prior work on submodular summarization approaches has focused on combining several models and learning weighted mixtures, we focus on the explainability of different models and featurizations, and how they apply to different domains. We also provide implementation details on summarization systems and the different modalities involved. We hope that the study from this paper will give insights into practitioners to appropriately choose the right summarization models for the problems at hand.*

## 1. Introduction

Visual Data in the form of images, videos and live streams have been growing at an unprecedented rate in the last few years. While this massive data is a blessing to data science by helping improve predictive accuracy, it is also a curse since humans are unable to consume this large amount of data. Moreover, today, machine generated videos (via Drones, Dash-cams, Body-cams, Security cameras, Go-pro etc.) are being generated at a rate higher than what we as humans can process. Moreover, majority of this data is plagued with redundancy. Given this data explosion, machine learning techniques which automatically understand, organize and categorize this data are of utmost importance. Video summarization attempts to provide a highlight of the most critical and important events in the video, giving the viewer a quick glimpse of the entire video so they can decide which parts of the video is important.

What comprises of the most critical aspect of a video depends largely on the domain. What is important in a surveillance video is very different from the highlights of a soccer game. This work attempts to provide a better understanding of different summarization models in different domains. We try to make a case that the choice of the summarization model really depends on the application and domain at hand. This paper investigates several choices of summarization models – models which capture diversity, representation, importance or relevance, and coverage. We quantitatively and qualitatively study this pattern in many different domains, including surveillance footages, dashcam, bodycams and gopro footages, movies and TV shows and sports events like soccer. We argue how different characteristics are important for these domains, and through extensive experimentation establish the benefit of using the corresponding summarization models. For example, we show that for surveillance footages, diversity is more important compared to representation or coverage, while in a movie, representation and coverage form a better fit compared to diversity. Similarly, in a sports event like soccer, importance and relevance signals are important aspects of summarization. This paper also analyzes several choices of feature representations and concepts, including faces, scenes, humans, color information, objects etc.

We study three variants of summarization: one is extractive summarization, the second is query focused summarization and the third is Entity based summarization (which we also call Concept based summarization). Entity based summarization focuses on entities, like objects, scenes, humans, faces to provide a representative yet diverse subset of these entities. This answers questions like who are the different people or what are the diverse objects and scenes in the video. Finally, we discuss several implementational

details on how to create a video summarization system, including the preprocessing of features, different segmentations of shots and tricks for speeding up the optimization for near real time response times.

### 1.1. Existing Work

Several papers in the past have investigated the problems of video and image collection summarization. Video Summarization techniques differ in the way they generate the output summary. Some of these [38, 16] extract a set of keyframes from the video, while others focus on extracting video summaries or skims from the long video [8, 40]. Other forms of video summarization include creating GIF summaries from videos [9], Montages [31], Visual Storyboards from videos [4], video synopses [26] and time lapses and hyperlapse summaries [13]. Similarly, image collection summarization involves choosing a subset of representative images from the collection [35]. Another line of approach, which is similar to what we call Entity based Summarization, was proposed in [21], wherein the authors select representative summaries of all objects in a video. They do this by modeling the problem as that of sparse dictionary selection. Most video summarization techniques can be categorized into methods trying to model one of three properties of summaries (i) interestingness (how good is a given snippet as a summary), (ii) representativeness (how well the summary represents the entire video or image collection), and (iii) diversity (how non-redundant and diverse is the summary).

Examples of methods which model interestingness of snippets include [38] that find summary snippets through motion analysis and optical flow, [16] which uses humans and objects to determine interesting snippets and finally, [6] which models interestingness through a super-frame segmentation. [2] summarizes multiple videos collectively by looking at inter-video-frame similarity and posing a maximal bi-clique finding algorithm for finding summaries. Methods which only model the quality of the snippets, or equivalently the interestingness of the summaries and do not model the diversity often achieve redundant frames and snippets within their summary.

Hence a lot of recent work has focused on diversity models for video and image collection summarization. [28] used the Facility Location function with a diversity penalty for image collection summarization, while [29] defined a coverage function and a disparity function as a diversity model. [20] attempted to find the candidate chain of sub shots that has the maximum score composed of measures of story progress between sub shots, importance of individual sub shots and diversity among sub shot transitions. [35] was among the first to use a mixture of submodular functions learnt via human image summaries for this problem. For video summarization, [17] proposed the Maximum Marginal Relevance (MMR) as a diversity model, while [40, 5] used a Determinantal Point Process based approach for selecting diverse summaries. [41] proposed an approach for video summarization based on dictionary based sparse coding, and [8] proposed using mixtures of submodular functions and supervised learning of these mixtures via max-margin training, an approach used for several other tasks including document summarization [19] and image collection summarization [35].

### 1.2. Our Contributions

The goal of this work is not to achieve the best results on Video and Image summarization tasks and datasets like TVSum [30] and Summe [7]. Rather, we attempt to provide insights into what it takes to build a real world video summarization system. In particular, we try to understand the role of different submodular functions in different domains, and how to implement a video summarization system in practice.

As observed in prior work [8, 19, 35] several models for diversity, representation, coverage and uniformity can be unified within the class of Submodular Optimization. We build upon this work as follows.

1. This paper studies the role and characteristics of different summarization models. What constitutes a good summary depends on the particular domain at hand.

2. We investigate several diversity, coverage and representation models, and demonstrate how different models are applicable in different kinds of video summarization tasks.

3. We validate our claims by empirically showing the behavior of these functions on different kinds of videos, and quantitatively prove this on several videos in each domain. For example, we show that *Diversity* models focus on getting outliers in the video, which is important in domains like surveillance. On the other hand, *Representation* models capture the centroids and important scenes, which is useful in Movies. We also argue how coverage functions focus on achieving a good coverage of concepts. Similarly, we show that in domains like soccer, *Importance* or *Relevance* plays the most important role in the summary.

4. We also discuss the computational scalability of the optimization algorithms, and point out some computational tricks including lazy evaluations and memoization, which enable optimized implementations for various submodular functions. As a result, we show that once the important visual features have been extracted (via a pre-processing step), we can obtain the summary subset of the video (or frames) in a few seconds. This allows the user to interactively obtain summaries of various lengths, types and queries in real time. We empirically demonstrate the benefit of memoization and lazy greedy implementations for various video summarization problems.

Most past work on Video and Image collection summarization, either use a subset of hand-tuned submodular functions [28, 19, 17] or a learnt mixture of submodular functions [8, 35, 19]. This work addresses the orthagonal aspect how do different subclasses of submodular functions model

summarization and their performance in different video domains. We believe the insights gathered from this work, will help practitioners in choosing appropriate models for several real world video and image summarization tasks.

## 2. Background and Main Ideas

This section describes the building blocks of our framework, namely the Submodular Summarization Framework and the basics of Convolutional Neural Networks for Image recognitions (to extract all the objects, scenes, faces, humans etc.)

### 2.1. Submodular Summarization Framework

We assume we are given a set $V = \{1, 2, 3, \cdots, n\}$ of items which we also call the *Ground Set*. Also define a utility function $f : 2^V \rightarrow \mathbf{R}$, which measures how good of a summary a set $X \subseteq V$ is. Let $c : 2^V \rightarrow \mathbf{R}$ be a cost function, which describes the cost of the set (for example, the size of the subset). The goal is then to have a summary set $X$ which maximizes $f$ while simultaneously minimizes the cost function $c$. In this paper, we study a special class of set functions called *Submodular Functions*. Given two subsets $X \subseteq Y \subseteq V$, a set function $f$ is submodular, if $f(X \cup j) - f(X) \geq f(Y \cup j) - f(j)$, for $j \notin Y$. This is also called the diminishing returns property. Several Diversity and Coverage functions are submodular, since they satisfy this diminishing returns property. We also call a function *Monotone Submodular* if $f(X) \leq f(Y)$, if $X \subseteq Y \subseteq V$. The ground-set $V$ and the items $\{1, 2, \cdots, n\}$ depend on the choice of the task at hand. We now define a few relevant optimization problems which shall come up in our problem formulations:

$$\text{Problem 1: } \max_{X \subseteq V, s(X) \leq b} f(X) \tag{1}$$

Problem 1 is knapsack constrained submodular maximization [32]. The goal here is to find a summary with a fixed cost, and $s_1, s_2, \cdots, s_n$ denotes the cost of each element in the ground-set. A special case is cardinality constrained submodular maximization, when the individual costs are 1 [23]. This a natural model for extracting fixed length summary videos (or a fixed number of keyframes).

$$\text{Problem 2: } \min_{f(X) \geq c} s(X) \tag{2}$$

This problem is called the *Submodular Cover Problem* [39, 11]. $s(X)$ is the modular cost function, and $c$ is the coverage constraint. The goal here is to find a minimum cost subset $X$ such that the submodular coverage or representation function covers *information* from the ground set. A special case of this is the set cover problem. Moreover, Problem 2 can be seen as a Dual version of Problem 1 [11].

Submodular Functions have been used for several summarization tasks including Image summarization [35], video summarization [8], document summarization [19], training data summarization and active learning [37] etc. Using a greedy algorithm to optimize a submodular function (for selecting a subset) gives a lower-bound performance guarantee of around 63% of optimal and in practice these greedy solutions are often within 90% of optimal [14]. This makes it advantageous to formulate (or approximate) the objective function for data selection as a submodular function.

### 2.2. CNNs for Image Feature Extraction

Convolutional Neural Networks are critical to feature extraction in our summarization framework. We pre-process the video to extract key visual features including objects, scenes, faces, humans, etc. Convolutional Neural Networks have recently provided state of the art results for several recognition tasks including object recognition [15, 34, 10], Scene recognition [42], Face Recognition [24] and Object Detection and Localization [27]. We next describe the end to end system in detail.

## 3. Method

The input to our system is a video. Our system then extracts all important features from the video and generates an analysis database. The user can then interact with the system in several ways. User can generate a video summary of a given length, or extract a set of key frames or a montage describing the video. Similarly the user can search for a query and extract video snippets of frames which are relevant to the query. Finally the user can also view a summary of all objects, scenes, humans and faces in the video along with their statistics. All these interactions are enabled on the fly (in a few seconds). The user can also define the summarization model of their choice. We investigate and compare different submodular models, and argue the utility of different models based on the use case.

### 3.1. Problem Formulation for the Multi-Faceted Visual Summarization

We now formulate problem statements across the different summarization views. Extractive summarization considers the entire video. We can generate a summary either in terms of key frames (represent the video as a set of frames sampled at a frame-rate), or video snippets. In either case, we extract a ground set $V$, with each individual element either being a key frame or a video snippet. We then solve Problems 1 or 2 depending on the use case. Problem 1 is the right formulation if we are interested in obtaining a summary of a fixed budget. Problem 2 is useful if we don't care about the size of the video, but we are interested in the summary capturing all the *information* of the video. In the case of query based summarization, we first extract the set of frames or snippets relevant to that query $q$. Denote this by $V_q$. We then solve the submodular optimization problem on $V_q$. Finally, in the case of entity based summarization, we extract all the entities in the video, and denote the set of entities as $V_e$. $V_e$ represents, for example, all the faces of people in the video. We can then run our summarization with $V_e$ as the groundset.

In the case of Extractive or Query based summarization, the ground truth elements can be either frames of video snippets. Our video snippets can be either fixed length snippets or Shots, obtained from a shot detector. If the snippets are fixed length snippets (say, 2 or 3 seconds), we can use the cardinality constrained submodular maximization. If the snippets are shots from the video, the length of each shot can differ, and we have the more general knapsack constrained setting. While our system can handle each of these modes, we focus on the key-frame based method for our experiments, since we are interested in proving the utility of different summarization models. The insights will carry over to the other modes as well.

## 3.2. Submodular Functions as Summarization Models

This section describes the Submodular Functions used in our system. We divide these into Coverage Functions, Representation Functions and Diversity Functions.

### 3.2.1 Modeling Coverage

This class of functions model notions of coverage, i.e. try to find a subset of the ground set $X$ which covers a set of *concepts*. Below are instantiations of this.

**Set Cover Function:** Denote $V$ as the ground set and let $X \subseteq V$ be a subset (of snippets or frames). Further $\mathcal{U}$ denotes a set of concepts, which could represent, for example, scenes or objects. Each frame (or snippet) $i \in X$ contains a subset $U_i \in \mathcal{U}$ set of concepts (for example, an image covers a table, chair and person). The set cover function then is

$$f(X) = w(\cup_{i \in X} U_i), \tag{3}$$

where $w_u$ denotes the weight of concept $u$.

**Probabilistic Set Cover:** This is a generalization of the set cover function, to include probabilities $p_{iu_i}$ for each object $u_i$ in Image $i \in X$. For example, our convolutional neural network might output a confidence of object $u_i$ in Image $i$, and we can use that in our function. The probabilistic coverage function is defined as,

$$f(X) = \sum_{i \in \mathcal{U}} w_i [1 - \prod_{i \in X} (1 - p_{ij})]. \tag{4}$$

The set cover function is a special case of this if $p_{ij} = 1$ if Object $j$ belongs to Image $i$ (i.e. we use the hard labels instead of probabilities).

**Feature Based Functions:** Finally we investigate the class of Feature Based functions. Here, we denote an Image $i$ via a feature representation $q_i$. This could be, for example, the features extracted from the second last layer of a ConvNet. Denote $F$ as the set of features. The feature based function is defined as,

$$f(X) = \sum_{i \in F} \psi(q_i(X)) \tag{5}$$

where $q_i(X) = \sum_{j \in X} q_{ij}$, and $q_{ij}$ is the value of feature $i$ in Image $j$. $\psi$ is a concave function. Examples of $\psi$ are square-root, Log and Inverse Function etc.

### 3.2.2 Modeling Representation

Representation based functions attempt to directly model representation, in that they try to find a representative subset of items, akin to centroids and mediods in clustering.

**Facility Location Function:** The Facility Location function is closely related to k-mediod clustering. Denote $s_{ij}$ as the similarity between images $i$ and $j$. We can then define $f(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$. For each image $i$, we compute the representative from $X$ which is closest to $i$ and add the similarities for all images. Note that this function, requires computing a $O(n^2)$ similarity function. However, as shown in [36], we can approximate this with a nearest neighbor graph, which will require much smaller space requirement, and also can run much faster for large ground set sizes.

**Saturated Coverage Function:** The saturated coverage function [18] is defined as $f(X) = \min\{\sum_{i \in X} s_{ij}, \alpha \sum_{i \in V} s_{ij}\}$. This function is similar to Facility Location and attempts to model representation. This is also a Kernel based function and requires computing the similarity matrix.

**Graph Cut Functions:** We define the graph cut family of functions as $f(X) = \lambda \sum_{i \in V} \sum_{j \in X} s_{ij} - \sum_{i,j \in X} s_{ij}$. This function is similar to the Facility Location and Saturated Coverage in terms of its modeling behaviour.

### 3.2.3 Modeling Diversity

The third class of Functions are Diversity based ones, which attempt to obtain a diverse set of key points.

**Dispersion (Disparity) Functions:** Denote $d_{ij}$ as a distance measure between Images $i$ and $j$. Define a set function $f(X) = \min_{i,j \in X} d_{ij}$. This function is not submodular, but can be efficiently optimized via a greedy algorithm [3]. It is easy to see that maximizing this function involves obtaining a subset with maximal minimum pairwise distance, thereby ensuring a diverse subset of snippets or keyframes. Similar to the Minimum Disparity, we can define two more variants. One is Disparity Sum, which can be defined as $f(X) = \sum_{i,j \in X} d_{ij}$. This is a supermodular function. Another model is, what we call, Disparity Min-Sum which is a combination of the two forms of models. Define this as $f(X) = \sum_{i \in X} \min_{j \in X} d_{ij}$. This function is submodular [1].

**Determinantal Point Processes:** Another class of Functions are Determinantal Point Processes, defined as $p(X) = \text{Det}(S_X)$ where $S$ is a similarity kernel matrix, and $S_X$ denotes the rows and columns instantiated with elements in $X$. It turns out that $f(X) = \log p(X)$ is submodular, and hence can be efficiently optimized via the Greedy algorithm. Unlike the other choices of submodular functions investigated so far, this requires computing the determinant and is $O(n^3)$ where $n$ is the size of the ground set. This function is not computationally feasible and hence we do not use it in our system since we require near real time results in summarization.

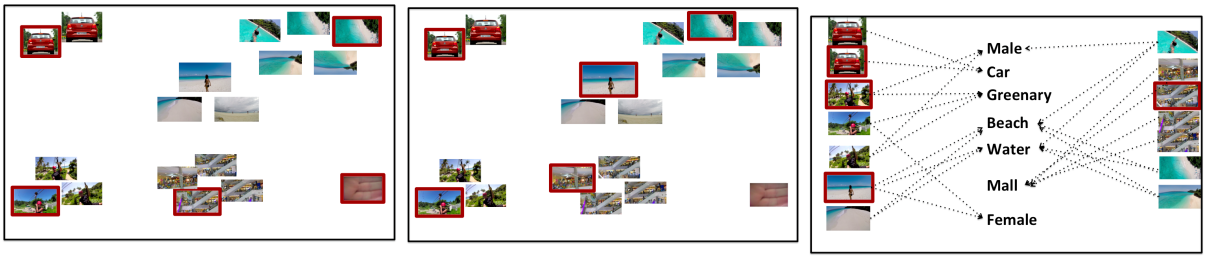


Figure 1. Illustration of the Difference between Diversity Functions, Coverage Functions and Representation Functions

| Name | $f(X)$ | $p_f(X)$ | $T_f^o$ | $T_f^p$ |
|---|---|---|---|---|
| Facility Location | $\sum_{i \in V} \max_{k \in X} s_{ik}$ | $[\max_{k \in X} s_{ik}, i \in V]$ | $O(n^2)$ | $O(n)$ |
| Saturated Coverage | $\sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \alpha_i\}$ | $[\sum_{j \in X} s_{ij}, i \in V]$ | $O(n^2)$ | $O(n)$ |
| Graph Cut | $\lambda \sum_{i \in V} \sum_{j \in X} s_{ij} - \sum_{i,j \in X} s_{ij}$ | $[\sum_{j \in X} s_{ij}, i \in V]$ | $O(n^2)$ | $O(n)$ |
| Feature Based | $\sum_{i \in \mathcal{F}} \psi(w_i(X))$ | $[w_i(X), i \in \mathcal{F}]$ | $O(n\lvert\mathcal{F}\rvert)$ | $O(\lvert\mathcal{F}\rvert)$ |
| Set Cover | $w(\cup_{i \in X} U_i)$ | $\cup_{i \in X} U_i$ | $O(n\lvert U\rvert)$ | $\lvert U\rvert$ |
| Prob. Set Cover | $\sum_{i \in \mathcal{U}} w_i[1 - \prod_{k \in X}(1 - p_{ik})]$ | $[\prod_{k \in X}(1 - p_{ik}), i \in \mathcal{U}]$ | $O(n\lvert\mathcal{U}\rvert)$ | $O(\lvert\mathcal{U}\rvert)$ |
| DPP | $\log\det(S_X))$ | SVD$(S_X)$ | $O(\lvert X\rvert^3)$ | $O(\lvert X\rvert^2)$ |
| Dispersion Min | $\min_{k,l \in X, k \neq l} d_{kl}$ | $\min_{k,l \in X, k \neq l} d_{kl}$ | $O(\lvert X\rvert^2)$ | $O(\lvert X\rvert)$ |
| Dispersion Sum | $\sum_{k,l \in X} d_{kl}$ | $[\sum_{k \in X} d_{kl}, l \in X]$ | $O(\lvert X\rvert^2)$ | $O(\lvert X\rvert)$ |
| Dispersion Min-Sum | $\sum_{k \in X} \min_{l \in X} d_{kl}$ | $[\min_{k \in X} d_{kl}, l \in X]$ | $O(\lvert X\rvert^2)$ | $O(\lvert X\rvert)$ |

Table 1. List of Submodular Functions used, with the precompute statistics $p_f(X)$, gain evaluated using the precomputed statistics $p_f(X)$ and finally $T_o^f$ as the cost of evaluation the function without memoization and $T_p^f$ as the cost with memoization. It is easy to see that memoization saves an order of magnitude in computation.

#### 3.2.4 Modeling Importance and Relevance

To model Importance or Relevance, we use Modular terms [25]. Given a specific task, we train a supervised model to predict the important frames in that video (for example, a *goal* might be considered important in a soccer video). Given this learnt model, we can predict the score of each frame, and rank the scores. This is exactly equivalent to optimize the modular function defined with these scores.

#### 3.2.5 Understanding Diversity, Representation and Coverage

Figure 1 demonstrates the intuition of using diversity, representation and coverage functions. Diversity based functions attempt to find the most different set of images. The leftmost figure in Fig. 1 demonstrates this. It is easy to see that the five most diverse images are picked up by the diversity function (Disparity Min), and moreover, the summary also contains the image with a hand covering the camera (the image on the right hand side bottom), which is an outlier. The middle figure demonstrates the summary obtained via a representation function (like Facility Location). The summary does not include outliers, but rather contains one representative image from each cluster. The diversity function on the other hand, does not try to achieve representation from every cluster. The third figure demonstrates coverage functions. The summary obtained via a coverage function (like Set Cover or Feature based function), covers all the concepts contained in the images (Male, Car, Greenery, Beach etc.).

### 3.3. Instantiations of the Submodular Functions

Having discussed the choices of the submodular functions and features, we go over the specific instantiations of submodular functions considered in our system. First consider Extractive and Query Based Summarization. For the Facility Location function and the disparity min function, we define the similarity kernel as:

$$s_{ij} = \langle F_s^i, F_s^j \rangle + \langle F_o^i, F_o^j \rangle + \text{corr}(H^i, H^j)$$

where $F_s$ represent normalized Deep Scene Features extracted using GoogleNet on Places205 [42], $F_o$ represents normalized Deep Object features using GoogleNet on ImageNet [34], $H$ represents the normalized color histogram features. Since the disparity min function uses a distance function, we use $d_{ij} = 1 - s_{ij}$. For Feature based functions, the feature-set $\mathcal{F}$ is a concatenation of the scene features $F_s$ and object features $F_o$. In order to define the Set Cover function, we define $U_i$ as the Scene and YOLO object labels corresponding to the Image. Recall that the labels for scenes and objects were chosen based on a pre-defined threshold (i.e. select scene and objects labels if the probability for the label is greater than a threshold). The Probabilistic Set Cover function is defined via a concatenation of

the probabilities from the scene and object models. Query based summarization for keyframes is identical to extractive summarization, except that we first get a groundset $V_q$ which is related to the query. The queries, are either objects, scenes, faces/humans with age and gender, text in the video, as well as meta data like subtitles etc.

Finally, for entity or concept based summarization, we extract the entities from the videos. Entities we consider are objects, faces etc. For faces, we use the VGG Face model from [24], pretrained on Celeb Face data for Face recognition. The objects are localized using YOLO [27]. We extract features from GoogLeNet [15, 34], along with color histogram [33]. The similarity kernel we use here is $s_{ij} = \langle F_o^i, F_o^j \rangle + \text{corr}(H^i, H^j)$.

Next we discuss the choice of the submodular functions. Facility Location, Disparity Min/Sum, Graph Cut, Saturated Coverage, and DPPs are instantiated using Similarity Kernels discussed above. Feature Based functions are defined directly via features, and we use the deep features as described above. In the case of the Set cover and probabilistic set cover functions, we use the labels and probabilities respectively from the deep models as the concepts.

### 3.4. Optimization Algorithms

The previous sections describe the models used in our system. We now investigate optimization algorithms which solve Problems 1 and 2. Variants of a greedy algorithm provide near optimal solutions with approximation guarantees for Problems 1-3 [39, 23, 32].

**Budget Constrained Submodular Maximization:** For the budget constrained version (Problem 1), the greedy algorithm is a slight variant, where at every iteration, we sequentially update $X^{t+1} = X^t \cup \text{argmax}_{j \in V \setminus X^t} \frac{f(j|X^t)}{c(j)}$. This algorithm has near optimal guarantees [32].

**Submodular Cover Problem:** For the Submodular Cover Problem (Problem 2), we again resort to a greedy procedure [39] which is near optimal. In this case, the update is similar to that of problem 1, i.e. choose $X^{t+1} = X^t \cup \text{argmax}_{j \in V \setminus X^t} f(j|X^t)$. We stop as soon as $f(X^t) = f(V)$, or in other words, we achieve a set which covers all the concepts.

**Lazy Greedy Implementations:** Each of the greedy algorithms above admit lazy versions which run much faster than the worst case complexity above [22]. The idea is that instead of recomputing $f(j|X^t), \forall j \notin^t$, we maintain a priority queue of sorted gains $\rho(j), \forall j \in V$. Initially $\rho(j)$ is set to $f(j), \forall j \in V$. The algorithm selects an element $j \notin X^t$, if $\rho(j) \geq f(j|X^t)$, we add $j$ to $X^t$ (thanks to submodularity). If $\rho(j) \leq f(j|X^t)$, we update $\rho(j)$ to $f(j|X^t)$ and re-sort the priority queue. The complexity of this algorithm is roughly $O(kn_R T_f)$, where $n_R$ is the average number of re-sorts in each iteration. Note that $n_R \leq n$, while in practice, it is a constant thus offering almost a factor $n$ speedup compared to the simple greedy algorithm.

| | Memoization | | | No Memoization | | |
|---|---|---|---|---|---|---|
| **Function** | **5%** | **15%** | **30%** | **5%** | **15%** | **30%** |
| Fac Loc | 0.34 | 0.4 | 0.71 | 48 | 168 | 270 |
| Sat Cov | 0.36 | 0.64 | 0.92 | 55 | 177 | 301 |
| Gr Cut | 0.39 | 0.52 | 0.82 | 41 | 161 | 355 |
| Feat B | 0.16 | 0.21 | 0.32 | 9 | 16 | 21 |
| Set Cov | 0.21 | 0.31 | 0.41 | 5 | 16 | 31 |
| PSC | 0.11 | 0.37 | 0.42 | 7 | 19 | 35 |
| DPP | 32 | 107 | 411 | 171 | 1003 | 4908 |
| DM | 0.11 | 0.61 | 0.82 | 21 | 125 | 221 |
| DS | 0.21 | 0.63 | 0.89 | 41 | 134 | 246 |

Table 2. Timing results in seconds for summarizing a two hour video for various submodular functions

## 4. Implementational Tricks

This section goes over implementation tricks via memoization. One of the parameters in the lazy greedy algorithms is $T_f$, which involves evaluating $f(X \cup j) - f(X)$. One option is to do a naïve implementation of computing $f(X \cup j)$ and then $f(X)$ and take the difference. However, due to the greedy nature of algorithms, we can use memoization and maintain a precompute statistics $p_f(X)$ at a set $X$, using which the gain can be evaluated much more efficiently. At every iteration, we evaluate $f(j|X)$ using $p_f(X)$, which we call $f(j|X, p_f)$. We then update $p_f(X \cup j)$ after adding element $j$ to $X$. Table 1 provides the precompute statistics, as well as the computational gain for each choice of a submodular function $f$. Denote $T_f^o$ as the time taken to naïvely compute $f(j|X) = f(X \cup j) - f(X)$. Denote $T_o^p$ as the time taken to evaluate this gain given the pre-compute statistics $p_X$. We see from Table 1, that evaluating the gains using memoization is often an order of magnitude faster. Moreover, notice that we also need to update the pre-compute statistics $p_X$ at every iteration. For the functions listed in Table 1, the cost of updating the pre-compute statistics is also $T_f^p$. Hence every iteration of the (lazy) greedy algorithm costs only $2T_f^p$ instead of $T_f^o$ which is an order of magnitude larger in every case. In our results section, we evaluate empirically the benefit of memoization in practice.

## 5. Results

Our system is implemented in C++. We use Caffe [12] and DarkNet [27] for deep CNNs and OpenCV for other computer vision tasks. A graphical representation of our system is depicted in Figure 4.

Figure 2 shows the results for extractive summarization as keyframes, extractive summarization on concepts or entities and query based summarization on keyframes. We compare the different summarization models under various scenarios and evaluation measures. Instead of comparing all the submodular functions described above, we consider representatives from each class of functions. We use Facility Location as a representative function, Disparity Min for Diversity and Set Cover as a choice for Coverage functions.

We next create a dataset of videos from different categories. We select 10 videos from the following cate-

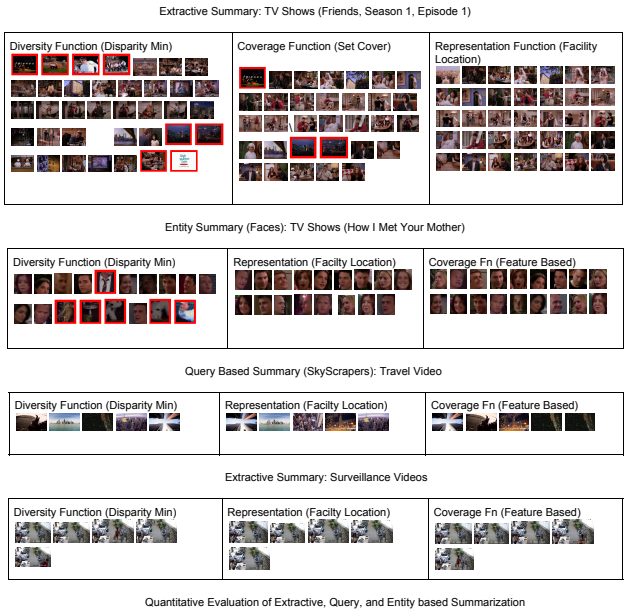


Figure 2. Illustration of the Results. The top figure shows the results from extractive summarization on TV shows, the second demonstrates entity summary on a TV show. The third figure shows the results of query based summarization on a query "SkyScraper" while the fourth one shows the results of extractive summarization on surveillance videos. In each case we compare Representation, Diversity and Coverage models. See the text for more details.

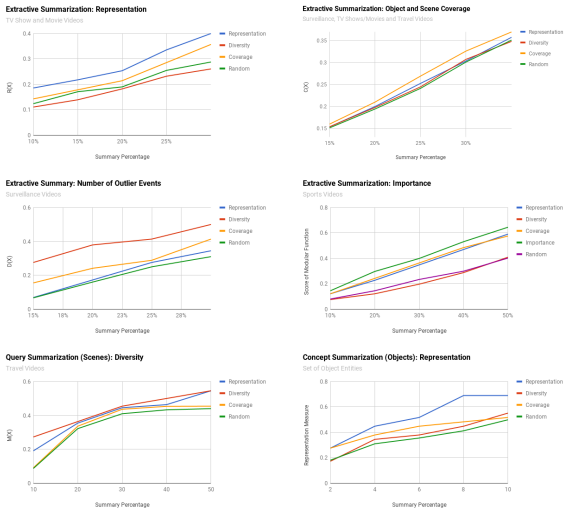


Figure 3. Comparison of Diversity, Coverage and Representation Models for various domains and scenarios. See the text for more details.

gories: Movies/TV shows, Surveillance camera footage, Travel videos and sports videos like Soccer. In the following sections, we annotate various events of interest (ground-truth) from these videos to define various evaluation criteria. The annotation mechanism and evaluation criteria is described in each of the sections below. The goal of this is to demonstrate the role of various summarization models.

**Extractive Summarization: Representation** The top Figure in Fig. 2 demonstrates the results of extractive summarization on Movies and TV shows. Diversity Models tend to pick up outlier events, which in this case, include transition scenes and other outliers. In contrast, the Representation function (Facility Location) tends to pick the representative scenes. The coverage function does something in between. In the case of a TV show, representative shots are probably more important compared to the transition scenes. To quantify this, define an evaluation measure as follows. We divide a movie (TV Show) into a set of scenes $S_1, \cdots, S_k$ where each scene $S_i$ is a continuous shot of events. We do not include the outliers (we define outliers as shots less than a certain length – for example transition scenes). Given a summary $X$, define $R(X) = \sum_{i=1}^{k} \min(|X \cap S_i|, 1)/k$. A summary with a large value of $R(X)$ will not include the outliers and will pick only single representatives from each scene. We evaluate this on 10 different TV show and movie videos. Figure 3 (top left) compares the representative, diversity, and coverage models and a random summary baseline. We see the representative model (Facility Location) tends to perform the best as expected, followed by the coverage model. The diversity model does poorly since it picks a lot of outliers.

**Extractive Summarization: Coverage** Next, we define an evaluation criteria capturing coverage. For each frame in the video (sampled at 1FPS), define a set of concepts covered $\mathcal{U}$. Denote $\mathcal{U}(X)$ as the set of concepts covered by a set $X$. For each frame of the video, we hand pick a set of concepts (scenes and objects contained in the video). Define the coverage objective as $C(X) = \mathcal{U}(X)/\mathcal{U}(V)$. Figure 3 demonstrates the coverage objective for the different models. We obtain this by creating a set of 10 labeled videos of different categories (surveillance, TV shows/movies, and travel videos). As expected, the coverage function (set cover) achieves superior results compared to the other models.

**Extractive Summarization: Outliers and Diversity** In the above paragraphs, we define two complementary evaluation criteria, one which captures representation and another which measures coverage. We argue how, for example, representation is important in Movies and TV shows. We now demonstrate how the diversity models tend to select outliers and anomalies. To demonstrate this, we select a set of surveillance videos. Most of our videos have repetitive events like no activity or people sitting/working. Given this, we mark all the different events (what we call outliers), including for example, people walking in the range of the camera or any other different activity. We create a dataset of 10 surveillance videos with different scenarios. Most of these videos have less activity. Given a set

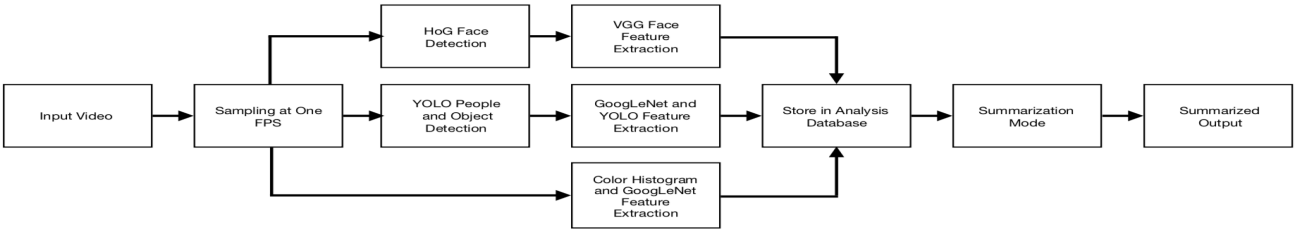


Figure 4. End-to-End Processing and Summarization of a Video

$S_1, S_2, \cdots S_k$ of these events marked in the video, define $D(X) = \sum_{i=1}^{k} \min(|X \cap S_i|, 1)$. Note this measure is similar to the representative evaluation criteria ($R(X)$) except that it is defined w.r.t the outlier events. Figure 3 (middle left) shows the comparison of the performance of different models on this dataset. As expected, the Diversity measures outperforms the other models consistently.

**Extractive Summarization: Importance** To demonstrate the benefit of having the right *importance* or *relevance* terms, we take a set of videos where intuitively the relevance term should matter a lot. Examples include sports videos like Soccer. To demonstrate this, we train a model to predict important events of the video (e.g. the goals, red card). We then define a simple Modular function where the score is the output of the classifier. We then test this out and compare the importance model to other summarization models. The results are shown in Figure 3 (middle right). As we expect, the model with the importance gets the highest scores.

**Query Summarization: Diversity** We next look at query based summarization. The goal of query based summarization is to obtain a summary set of frames which satisfy a given query criteria. Figure 2 (third row) qualitatively shows the results for the query "Sky Scrapers". The Diversity measure is able to obtain a diversity of the different scenes. Even if there is an over-representation of a certain scene in the set of images satisfying the query, the diversity measure tends to pick a diverse set of frames. The representation measure however, tends to focus on the representative frames and can pick more than one image in the summary from scenes which have an over-representation in the query set. We see this Figure 2. To quantify this, we define a measure $M(X)$ by dividing the video into a set of clusters of frames $S_1, \cdots, S_k$ where each cluster contains similar frames. These are often a set of continuous frames in the video. We evaluate this on a set of 10 travel videos, and compare the different models. We see that the diversity and representation models tend to perform the best (Figure 3, bottom left), with the diversity model slightly outperforming the representative models. We also observe that there are generally very few outliers in the case of query based summarization, which is another reason why the diversity model tends to perform well.

**Entity Summarization:** Lastly we look at Entity summarization. The goal here is to obtain a summary of the entities (faces, objects, humans) in the video. Figure 2 (second row) demonstrates the results for Entity summarization of Faces. We see the results for Diversity, Coverage and Representation Models. The diversity model tends to pick up outliers, many of which are false positives (i.e. not faces). The representation model skips all outliers and tends to pick representative faces. To quantitavely evaluate this, we define a representation measure as follows. We remove all the outliers, and cluster the set of entities (objects, faces) into a set of clusters $E_1, \cdots, E_k$ where $E_i$ is a cluster of similar entities. We evaluate this again on a set of 10 videos. Figure 3 (bottom right) shows the results for objects. The results for Faces is similar and in the interest of space, we do not include these. We see that the representation model tends to outperform the other models and skips all the outliers. The diversity model focuses on outliers and hence does not perform as well.

**Scalability** Finally, we demonstrate the computational scalability of our framework. Table 2 shows the results of the time taken for Summarization for a two hour video (in seconds) with and without memoization. The groundset size is $|V| = 7200$. We see huge gains from using memoization compared to just computing the gains using the Oracle models of the functions. All our experiments were performed on Intel(R) Xeon(R) CPU E5-2603 v3 @1.6 GHz (Dual CPU) with 32 GB RAM. We used a NVIDIA 1080 GTX 8GB GPU for the Deep Learning. For the two hour video, the preprocessing took around 20 minutes on a single GPU. It would be much faster on multiple GPUs and moreover, this is typically done only once.

## 6. Conclusion

This paper presents a unified picture of multi-faceted video summarization for extractive, query based and entity based summarization. In each case, we take a closer look at the different summarization models and argue the benefits of these models in different domains. We qualitatively and quantitatively argue this by comparing the results on several domains. Finally, we discuss various implementation tricks to build applications around video and image summarization in production systems.

## References


[1] S. Chakraborty, O. Tickoo, and R. Iyer. Adaptive keyframe selection for video summarization. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 702–709. IEEE, 2015.

[2] W.-S. Chu, Y. Song, and A. Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *Proceedings of IEEE CVPR*, pages 3584–3592, 2015.

[3] A. Dasgupta, R. Kumar, and S. Ravi. Summarization through submodularity and dispersion. In *ACL (1)*, pages 1014–1022, 2013.

[4] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 862–871. ACM, 2006.

[5] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in NIPS*, pages 2069–2077, 2014.

[6] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *In Proc. ECCV*, pages 505–520. Springer, 2014.

[7] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[8] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proc. CVPR*, pages 3090–3098, 2015.

[9] M. Gygli, Y. Song, and L. Cao. Video2gif: Automatic generation of animated gifs from video. In *In Proc. CVPR*, pages 1001–1009, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] R. K. Iyer and J. A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in NIPS*, pages 2436–2444, 2013.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[13] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):78, 2014.

[14] A. Krause. *Optimizing sensing: Theory and applications*. ProQuest, 2008.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in NIPS*, pages 1097–1105, 2012.

[16] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *In Proc. CVPR*, pages 1346–1353. IEEE, 2012.

[17] Y. Li and B. Merialdo. Multi-video summarization based on video-mmr. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

[18] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[19] H. Lin and J. Bilmes. Learning mixtures of submodular shells with application to document summarization. In *Uncertainty in Artificial Intelligence (UAI)*. AUAI, 2012.

[20] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Proc. CVPR*, pages 2714–2721, 2013.

[21] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan. From keyframes to key objects: Video summarization by representative object proposal selection. In *Proc. CVPR*, pages 1039–1048, 2016.

[22] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[24] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[25] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *In Proc. ECCV*, pages 540–555. Springer, 2014.

[26] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *In Proc. IEEE PAMI*, 30(11):1971–1984, 2008.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[28] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[29] P. Sinha and R. Jain. Extractive summarization of personal photos from life events. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[30] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, pages 5179–5187. IEEE Computer Society, 2015.

[31] M. Sun, A. Farhadi, B. Taskar, and S. Seitz. Salient montages from unconstrained videos. In *European Conference on Computer Vision*, pages 472–488. Springer, 2014.

[32] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[33] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[35] S. Tschiatschek, R. K. Iyer, H. Wei, and J. A. Bilmes. Learning mixtures of submodular functions for image collection summarization. In *Advances in NIPS*, pages 1413–1421, 2014.

[36] K. Wei, R. K. Iyer, and J. A. Bilmes. Fast multi-stage submodular maximization. In *ICML*, pages 1494–1502, 2014.

[37] K. Wei, R. K. Iyer, and J. A. Bilmes. Submodularity in data subset selection and active learning. In *ICML*, pages 1954–1963, 2015.

[38] W. Wolf. Key frame selection by motion analysis. In *In Proc. ICASSP*, volume 2, pages 1228–1231. IEEE, 1996.

[39] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[40] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1059–1067, 2016.

[41] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2513–2520, 2014.

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.