

# Deep Visual Semantic Embedding for Video Thumbnail Selection

## Master Thesis Report

12 August 2016

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Master in Computer Science

BY

Arun Balajee VASUDEVAN

Mentors :

Michael Gygli, PhD student, CVLab, ETH Zurich  
Anna Volokitin, PhD student, CVLab, ETH Zurich  
Dr. Achanta Radhakrishna, Scientist, IVRL, EPFL  
Prof. Luc Van Gool, CVLab, ETH Zurich  
Prof. Sabbine Susstrunk, IVRL, EPFL



Suisse  
2016



# Abstract

The goal of my thesis is to select a set of thumbnails from a video based on a given text query. The selected thumbnails are not only relevant to the user query but also diverse enough visually to represent the video content. In short, the thesis addresses the problem of query relevant video summarization in the form of keyframes. We use a deep Visual Semantic Embedding Model and submodular mixtures to address the problem. In this work, we leverage the additional textual side information of images along with their visual information to jointly train a deep Visual Semantic Embedding network. This network maps the text queries and video frames to the same Latent Semantic Embedding Space where they are comparable. Later, we pose the video summarization task as a subset selection problem. We learn a linear combination of submodular functions to create a video summary in the form of a set of thumbnails that are both query relevant and diverse. For evaluation of our model, seeing the non-availability of a video summarization dataset catering to summaries that are both text relevant and diverse, we introduce a new dataset that consists of 100 query-video pairs with the frames annotated for the query relevance and are grouped to clusters based on their visual similarity. We evaluate our model on three different datasets and compared with previous state of the art methods. Our method improve w.r.t mAP over baseline by 2.6% on MSR evaluation dataset, 2.71% on MediaEval dataset and 3.9% on our newly annotated dataset: Relevance and Diversity (RAD) Dataset. We also present some qualitative results for the selection of diversified query relevant frames from videos.



# Acknowledgements

I would like to thank both of my supervisors Michael Gygli and Anna Volokitin for their countless advice and friendly attitude during my thesis work. The discussion were never left without a pool of ideas. It is a great honour to work under the direct supervision of Michael Gygli.

I would like to express my sincere thanks to Prof. Luc Van Gool for providing support throughout the thesis. I would like to thank my advisors Dr. Achanta Radhakrishna and Prof. Sabine Susstrunk for their encouragement and comprehensive advice during the entire coursework and for their valuable feedback during my mid-presentation.

I wish to thank Michael for the ice-cream breaks that we had. Last not the least, I would wish to thank EPFL Swiss Mobility Exchange program for the financial support during my thesis.



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Thesis Organization . . . . .	6
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	Query Relevance . . . . .	7
2.2	Diversity . . . . .	8
<b>3</b>	<b>MSR Dataset</b>	<b>9</b>
<b>4</b>	<b>Query-Image Embedding Model</b>	<b>13</b>
4.1	Query Representation . . . . .	13
4.2	Image Representation . . . . .	14
4.3	Objective function . . . . .	14
4.4	Loss function . . . . .	15
4.4.1	L1, L2 rank loss . . . . .	15
4.4.2	Huber rank loss . . . . .	16
4.5	Training Data . . . . .	16
4.6	Validation . . . . .	16
<b>5</b>	<b>Query Agnostic Model</b>	<b>17</b>
<b>6</b>	<b>Submodular Maximization</b>	<b>19</b>
6.1	Submodularity . . . . .	19
6.2	Submodular Functions . . . . .	19
6.2.1	Relevance . . . . .	20
6.2.2	Diversity . . . . .	20
6.3	Thumbnail Subset Selection Objective . . . . .	20
<b>7</b>	<b>Experiments</b>	<b>23</b>
7.1	Dataset . . . . .	23
7.2	Data Preprocessing . . . . .	23
7.3	Implementation . . . . .	23
7.4	Network training . . . . .	24
7.5	Baseline . . . . .	24

## CONTENTS

---

7.5.1	Loss function Comparison	25
7.5.2	Query Agnostic model	26
7.6	LSTM Training	26
7.7	CNN-LSTM Training	27
7.8	Evaluation	27
<b>8</b>	<b>New Evaluation Dataset: Relevance and Diversity (RAD)</b>	<b>29</b>
8.1	Dataset Description	29
8.2	Dataset Annotation	30
8.2.1	Relevance Annotation Task	31
8.2.2	Diversity Annotation Task	31
8.3	Ground Truth Generation	32
8.3.1	Combine Annotations	32
8.4	Annotation Statistics	33
8.4.1	Correlation:Relevance Task	34
8.4.2	Normalized Mutual Information:Diversity Task	35
8.5	Quality Control Procedure	39
<b>9</b>	<b>Results and Discussion</b>	<b>41</b>
9.1	Relevance	42
9.2	Diversity	42
9.3	Performance Evaluation	43
9.3.1	MSR Dataset	43
9.3.2	MediaEval Dataset	44
9.3.3	New Evaluation Dataset: RAD	45
9.4	Submodular Shells	46
<b>10</b>	<b>Conclusion and Future Work</b>	<b>49</b>
<b>A</b>	<b>Supplementary Results</b>	<b>51</b>

# List of Figures

1.1	Selected applications of extracting query specific thumbnails from a video. . . . .	3
1.2	Visual Semantic Embedding Space. . . . .	4
1.3	Our Model. . . . .	5
3.1	t-SNE visualization of 300D vector representation of query words. Zoom in for better reading. . . . .	9
4.1	<b>Siamese Network.</b> Scoring layer computes cosine similarity between query and frame embedding output vector and the ranking loss enforces the objective of scoring positive pair higher than the negative one. . . . .	15
7.1	Training time for a mini-batch of 128 query-image pairs for the three cases mentioned above. . . . .	24
7.2	Training error and testing accuracy plots for the baseline. . . . .	25
7.3	Loss Functions . . . . .	25
7.4	Score Distribution with x-axis representing the Scores and y-axis representing frequency . . . . .	26
8.1	Snapshot of Instructions page of Annotation Tool . . . . .	30
8.2	Snapshot of the Relevance Annotation Tool . . . . .	31
8.3	Snapshot of the Diversity Annotation Tool . . . . .	32
8.4	Similarity matrix based on the diversity annotations on the left and hierarchical clustering to group the annotations . . . . .	33
8.5	Distribution of Relevance score over the dataset . . . . .	34
8.6	Correlation between half-split(2-3 split) combined annotations . . . . .	35
8.7	Distribution of number of clusters over the dataset . . . . .	36
8.8	<b>VI metric(distance) graph.</b> Nodes represents the clusterings and edges represents the distance between them. This graph is computed on a video from RAD dataset. . . . .	37
8.9	Mean and standard deviation of Normalized Mutual Information . . . . .	38
8.10	Snapshot of the Review Annotation page . . . . .	39
9.1	<b>Examples of selected thumbnails for different queries</b> (better viewed in colour). The groundtruth label is provided at the top right. Unnormalized score from our method is put at bottom right. . . . .	41

## LIST OF FIGURES

---

9.2	<b>Thumbnails with &amp; without Diversification</b> For query:Anaconda Snake, above video thumbnails are obtained. The first row contain the relevant frames from the Visual Semantic Embedding Model while the second row represents the diverse set of query relevant frames obtained with $w = [1, 1]$ .	42
9.3	t-SNE visualization of subset selection. Red labelled as the selected points	43
9.4	Precision-Recall curve of MSR Evaluation dataset for different methods	43
9.5	Precision-Recall curve of MediaEval dataset for different methods	44
9.6	Precision-Recall curve of New Evaluation dataset for different methods	45
9.7	Grid Search for finding the weights of sub-modular shells	46
9.8	Query relevant and diversified results using our model on newly annotated RAD dataset	47
A.1	Diversified Query Relevant Results. Unnormalized score from our method is put at bottom right corner (better viewed in colour)	52

# List of Tables

3.1	<b>Examples from Clickture dataset.</b> For the query London and Elephant, left most column represents the most clicked images while the right most column represents the least clicked images. The central image shows the click distribution over the images for the queries. . . . .	10
7.1	Comparison of the thumbnail selection methods . . . . .	28
9.1	Comparison of the different thumbnail selection methods on MSR Evaluation dataset . . . . .	44
9.2	Comparison of the thumbnail selection methods on MediaEval dataset . . . . .	45
9.3	RAD dataset: Comparison of the thumbnail selection methods using queries . . . . .	46

LIST OF TABLES

---

# Chapter 1

## Introduction

It has become a glaring fact that the number of online videos that gets uploaded is growing each day and they generate billions of views and millions of hours of watchtime. As a matter of fact, YouTube has over a billion users and every day people watch hundreds of millions of hours on YouTube and the number of people watching YouTube per day is up 40% year by year since March 2014 <sup>1</sup>. The rising video platforms face many challenges. It is practically being difficult to search out the right video from a whole lot of videos. This may be due to the increasing numbers or due to an ambiguous video thumbnail. Though we do not have control in rising video uploads, we shall try to express a video with an informative thumbnail for a better video search as in Figure 1.1(a). Furthermore, among these huge collections, suppose we shortlist a collection of relevant videos, we still tend to watch the whole set of videos in search of a particular content. This video level searching becomes more difficult if the videos are lengthy. However, it would make a good sense to present a condensed preview of a video depending on a query as in Figure 1.1(b).



Figure 1.1: Selected applications of extracting query specific thumbnails from a video.

With the increasing number of video contents and the delay in watching the elaborate video content introduce the need for video thumbnail extraction from the videos. A good video thumbnail represents a condensed preview of the complete video content. There are various options of condensed preview such as GIFs[9], montages, thumbnails [14, 25], etc. The recent trends even in popular social networking sites like Facebook, Google+ and others are to display the videos as GIFs. One of the closely explored problem is video summarization [40, 18, 8, 39, 33]. It becomes more challenging if the video summarization is based

<sup>1</sup><https://www.youtube.com/yt/press/statistics.html>

on a given text query. However, the selection of query dependent video thumbnails have many applications in the real world like video highlights detection, video search where query dependent results are generated, GIF generation from video [9], among others. In this project, we focus on the video thumbnails selection from the videos, which not only make a good video representation but also are relevant to the user query.

There are many conventional works of extracting frames based on the visual content. [14] uses web-images as a prior to facilitate the summarization of videos. They learn a classifier for different viewpoints of each class (eg: automobiles) and assign each video frame to a learned subclass (belongs to a viewpoint). Then for the output summary, they select  $k$  frames that are closest to the centroid of top  $k$  ranked subclass. Recently, the works have appeared where abundant semantic information associated with the videos such as the title of the video [39], their description, user query for the video search [25] and others, are considered to extract relevant frames. The use of the side information allowed frames of the video relevant to the query to be selected [24]. In [24], associated keywords of videos are queried to retrieve photos from *Flickr*. Later, the video frames' interestingness are measured by computing the feature distribution similarity between the frame and retrieved photo set. [39] uses titles of the video to retrieve images and further select shots both relevant and representative from learned canonical visual concepts focusing on shared region between images and given video.

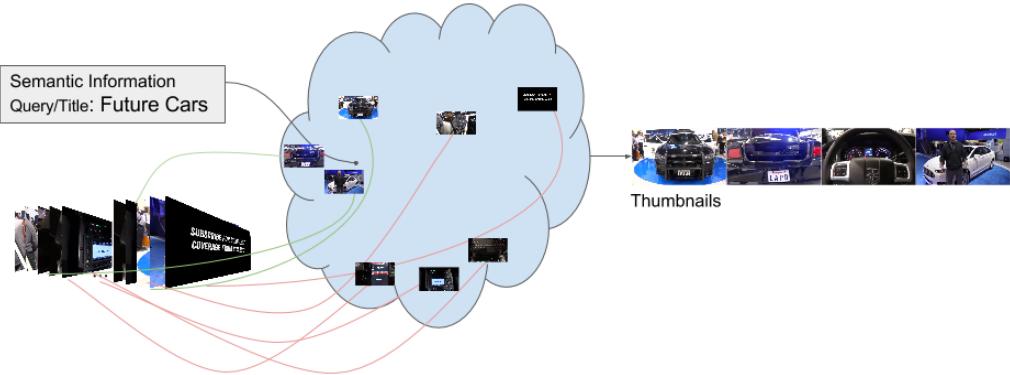


Figure 1.2: Visual Semantic Embedding Space.

In this problem of video thumbnail selection with additional semantic information, visual content and text information associated with the videos are mapped to the same latent semantic embedding space, where their projections show their clear relevance as given in Figure 1.2. We follow the work of [25] with some improvements. We use supervised learning techniques to train our model for the representation learning of text information and visual input. The representations of the textual input and the visual inputs are mapped to Visual Semantic Embedding vector space where semantic proximity between texts and images can be easily computed by the cosine similarity of their representations. For the supervised learning, we use a click through image dataset due to the absence of a publicly available click through video dataset. The click-through image dataset is a dataset comprising of text queries, clicked images and the number of clicks associated with them. Thus, text and images are embedded into the same space by joint training of queries and images of a click-through image dataset using the Visual Semantic Embedding Model. In this embedding space, query relevant frame embeddings would be ideally close to the query embedding compared to the proximity of the non-relevant frame embeddings. In Figure 1.2, we can see the mapping

of text information and video frames to the same embedding space where they are comparable. We output query specific thumbnails with properties: a) Query Relevance b) Diversity in selected thumbnails i.e. the output thumbnails are visually different. The Visual Semantic Embedding Model can leverage the semantic similarity of the queries and images which in turn helps with unseen textual and visual data. We leverage deep convolutional neural network (CNN) architecture of VGG19 [37, 17] for the input image, word2vec [29] and Long Short Term Memory (LSTM) network for the input text query. Word2vec is a neural language model which are well suited for extracting semantically meaningful dense vector representation of words.

Our textual content is a user query which is generally a combination of useful and non-informative words. In order to embed a query into the same space as the images, we input the word2vec vectors of query's words sequentially to LSTM network as in Figure 1.3 to learn a fixed length query representation. LSTM network can yield out a better vector representation for the query as it learns the importance of each word in the query. So, in our deep visual semantic model, there are two parallel networks, one for the textual query and the other for the image mapping as shown in Figure 1.3. The LSTM output from query part of the network and the projected fully connected layer from the input image part of the network are trained to map to the same latent semantic space. This trained embedding allows to leverage the relevance between the text queries and their semantically closer images. The relevance score between any unseen query-image pair can be computed from their projections in the latent semantic embedding space.

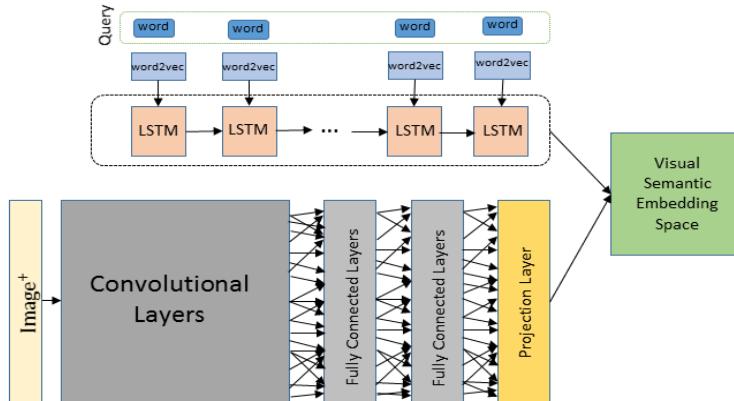


Figure 1.3: Our Model.

The query relevance score of all the sampled keyframes is exploited to rank the frames of the video. The ranked keyframes may not be diverse. Following [8], we use sub-modular mixtures to make a subset selection from all the keyframes of the video which are not relevant to the query but are also diverse in visual content. Finally, for a given query and a video, we get the output video summary as a diversified set of query relevant thumbnails. For the evaluation of our model, there is no publicly available video summarization dataset to our knowledge that concerns with query adaptive video summaries. Hence, we annotate a new dataset naming it Relevance And Diversity dataset (RAD) which we explain in detail in the Chapter 8. The main contributions of my thesis work are:

- Improvements on Deep visual semantic embedding model [25] using CNN-LSTM architecture and a better objective to map the textual and visual content to the same space.
- Annotation of a new dataset RAD of 100 query-video pairs with query relevance annotations for all the frames and cluster groupings of the frames based on visual similarity. We have also done a detailed analysis of annotations.
- Generating query dependent video summaries in the form of thumbnails using submodular mixtures and deep visual semantic embedding model.

## 1.1 Thesis Organization

The thesis is organized in the following manner. Chapter 3 explains about the MSR dataset that is used to learn the embedding model. The Query-Image Embedding model and Query agnostic model are explained in the following two chapters. Later, Chapter 6 describes about the concept of Submodular Maximization. Chapter 7 tries to explain the various experiments we conducted to choose our final network settings. Chapter 8 deals with how we annotate the new dataset and further describes the analysis on the same. Finally, Chapters 9 and 10 present the results and conclusions and mention some exploratory future works.

# Chapter 2

## Related Work

Conventionally, video thumbnail selection is performed by learning purely from visual content using various video attributes such as frame quality, attention measurements [12], camera motion [26], object, people detections [5, 34]. The above methods used the lone visual content of the video bereft of the side semantic information/metadata like title, tags or descriptions associated with videos. To get the query dependent video search, search engines initially used only the text information from the website where video was found. Later, search methods began to use the visual content information: Image retrieval [35, 19]. After receiving the input query, query relevant images were extracted. Then, the similarity between the searched images and thumbnail are leveraged to obtain the relevance between query and video thumbnail. For instance, in [24], photos from *Flickr* are extracted using the associated keywords of videos and then feature distribution similarity between the frame and retrieved photo set are computed for the frames interestingness. However, this is affected by the time consuming image search process [1]. Several learning methods [20, 42] are later employed to localize tags into videos and this helped in getting query dependent thumbnails according to the tags.

### 2.1 Query Relevance

Query dependent image search improved with the multi view embedding methods. [32] proposes a click-through-based cross view learning method which calculates the distance between an input query and an image in a latent common subspace giving out the query-image relevance. The deep Visual Semantic Embedding Model [4, 38] learns the semantic relationship between text labels and mapped images in rich semantic embedding space using deep learning. [39] uses titles of the video to retrieve images and further select shots both relevant and representative from learned canonical visual concepts focusing on shared region between images and given video. [25] uses a deep visual semantic embedding model generated query-image relevance score to rank the candidate thumbnails keyframes to yield the final video thumbnail. In order to fill the gap between the image search task and video thumbnail selection, they have also fine-tuned their trained model to the MSR click-through video dataset for a better thumbnail selection. They employ a multi-task deep visual semantic embedding model to exploit the query-image relevance from the click-through based image and video dataset. However, they generate a bias in the selection of candidate video thumbnail by using the hand crafted video attributes for the keyframe selection. Additionally, the objective function seems to have a mismatch as they train their model to score a positive query higher than

any random query for an image while in the inference stage, for a query, [25] scores all the frames based on its relevance to the query. Unlike [25], our objective function learns to score a positive image higher than any random images for a query which matches well with the inference stage.

We employ a deep visual semantic embedding model providing some improvements over [25] on the query embedding and the objective function. Firstly, we learn the query-image relevance by joint learning of two parallel networks- a CNN and a LSTM, to map the image and query respectively to the same latent semantic embedding space as in Figure 1.3. Unlike [25] which considers the embedding space vector from a single query word embedded using GloVe, we represent each word of the query using word embedding model word2vec/GloVe before it is passed sequentially to the LSTM network to yield a fixed length query embedding. We use the MSR Clickture click-through-based image dataset for training our model. Furthermore, we use the fixed length vector representation of query and uniformly sampled video frames to generate a rank score for each frame by computing the similarity measures like cosine similarity. This rank score can determine the selection of top-K frames for the final video thumbnail selection. These top-K frames may not be diverse visually in nature that results in redundancy in the selection of thumbnails because non-diverse frames (all frames look similar) would rather be less informative of the video.

## 2.2 Diversity

Here, the main goal is to select a diverse set of keyframes from a video that are visually distinct and they are well-representative of the video. There are many approaches in this area of video summarization [6, 8]. Maximal Marginal Relevance (MMR)[2] method which is widely used in NLP domain was adapted to video by [21], which selects the frames that optimizes for the relevance to the input video and reduces the redundancy within the summary. [8] solves the video summarization as a sub-modular maximization problem over a set of submodular functions defined separately for image interestingness and representativeness of videos. There are some recent works such as [46, 36] where they learn an end-to-end model that jointly optimize for the interestingness and diversity by using determinantal point process (DPPs). We follow [8] to frame our problem as a video summarization problem with submodular functions defined for query relevance and diversity.

# Chapter 3

# MSR Dataset

MSR Clickture<sup>1</sup> is a large scale real-world image dataset obtained from the massive amount of click data from the commercial search engine Bing. Click data provides strong connection between semantics of queries and images. This dataset is obtained as a triad  $\langle K, Q, C \rangle$  meaning that an image  $K$  was clicked  $C$  times in the search results of a query  $Q$ . In total, this dataset consists of 40 million unique images and 73.6 million unique queries.

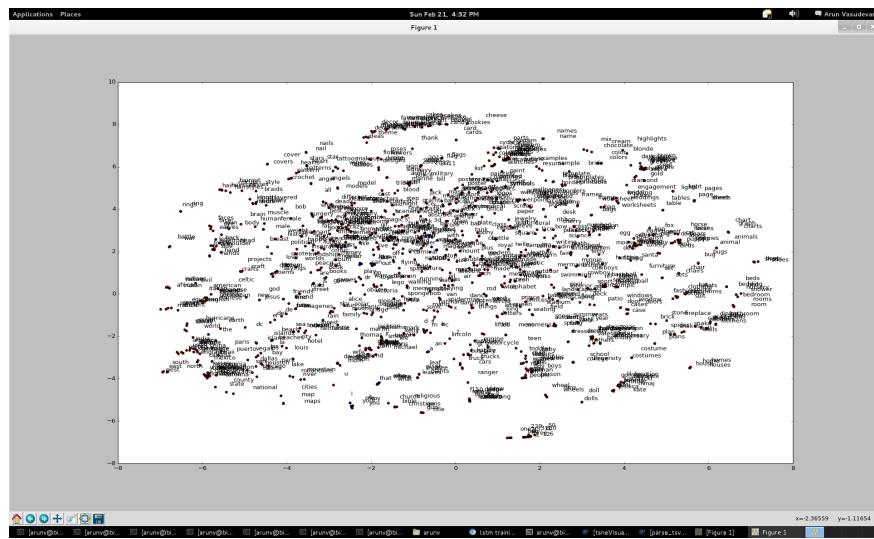


Figure 3.1: t-SNE visualization of 300D vector representation of query words. Zoom in for better reading.

We observe that top query words in the list ordered by frequency are stopwords or certain specific words such as image, pictures, etc. The incorporation of LSTM in query embedding model as in Figure 1.3 would have the benefit that it is able to learn the importance of individual words in the query to make a better representation of a query unlike the conventional method of taking the centroid of all the word vectors of the query for the query representation. To do a semantic analysis of the text queries, all the query words which

<sup>1</sup><http://research.microsoft.com/en-us/projects/clickture/>

comprise of 3.65 billion words from 73.6 million unique queries have been mapped into a 300-dimensional vector space using the word2vec word embedding model. The word2vec [29] takes the full query list as a text corpus and produces the word vectors as output. It first constructs a vocabulary from the training query list and then learns the vector representation of words. We use the word2vec model pre-trained on Google News Dataset and then finetuned for our text corpus <sup>2</sup>. We use 300 dimensions for word embedding as used in [25]. t-SNE dimensionality reduction [27] has been done to visualize them in 2D as shown in Figure 3.1. In Figure 3.1, we have plotted just 1000 words.

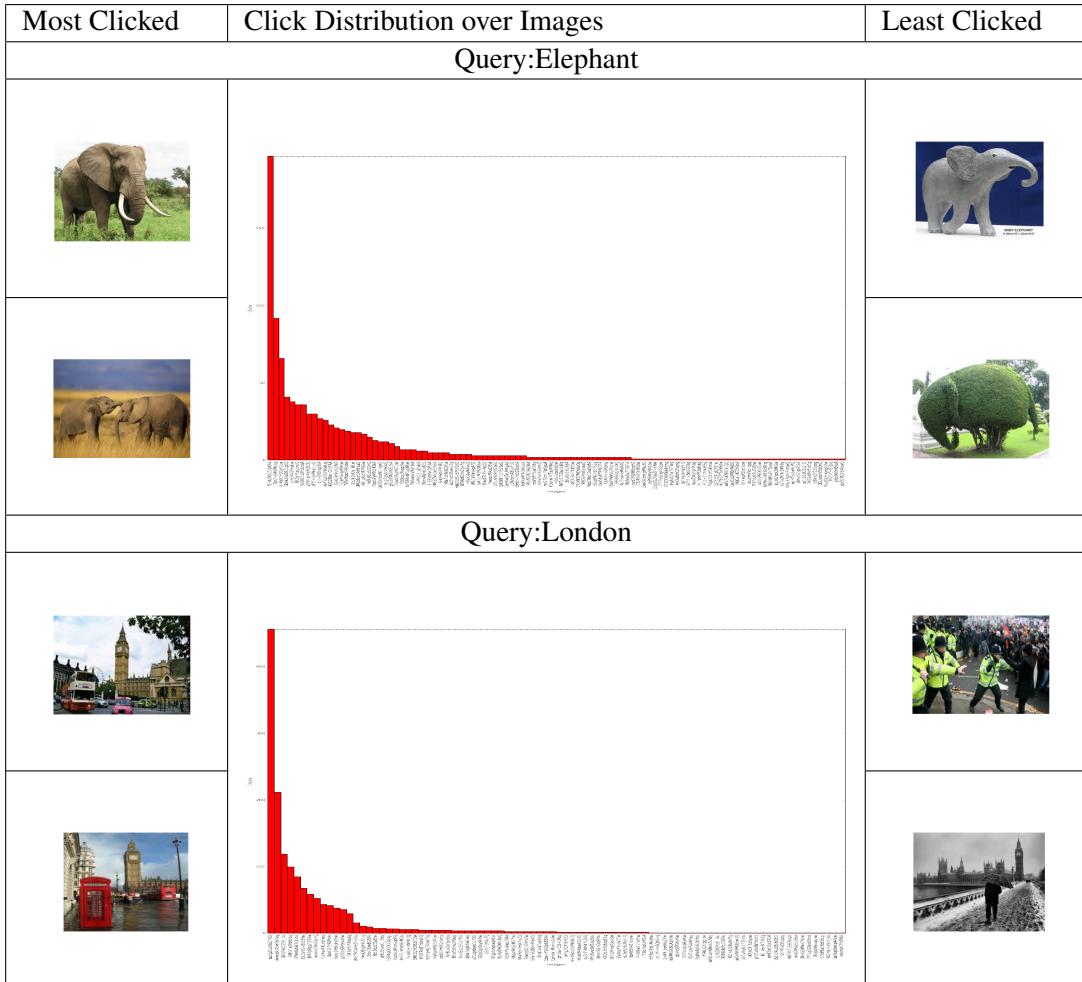


Table 3.1: **Examples from Clickture dataset.** For the query London and Elephant, left most column represents the most clicked images while the right most column represents the least clicked images. The central image shows the click distribution over the images for the queries.

In Figure 3.1, we observe that there are many groups of meaningful words formed into clusters of words, that is, semantically closer words are grouped together. This shows that vector similarity determines

<sup>2</sup><https://code.google.com/archive/p/word2vec/>

the semantic proximity in the latent semantic embedding space. The embedding of the images in the same embedding space can bring out the semantic proximity of visual content and text data which helps in finding the relevance between queries and images, even for queries that are unseen in training.

It is important to explore the distribution of queries and images of the dataset with regards to number of clicks. This helps to understand how the most and the least clicked images for a query differ in their characteristics, semantic content, etc. For each query, there are multiple instances of images with different number of clicks. In table 3.1, we plot the distribution of number of clicks over the images clicked for the query London. On the left side of the table, we have the images that have the maximum number of clicks while right side of the table has the minimally clicked images. Maximally clicked images are images that are aesthetically more favoured by a user than the minimally clicked bottom ones. In the middle, we have the histogram plot of number of clicks for each image-ID that are clicked for the search of query London. The distribution of clicks is influenced by the aesthetics of the image, image popularity and the order of search results. The images that comes in the tail-end are hard negatives which we did not use for the supervised learning of our model.



## Chapter 4

# Query-Image Embedding Model

The structure of our visual semantic embedding model is shown in Figure 1.3. The ultimate aim of our model is to map the images and queries into the Visual Semantic Embedding Space so that query and image embedding are semantically comparable in that space. Having the learned model, for a given text query and a video, we can embed the queries and the video frames into the same space. Then, we can find a set of query relevant frames depending on the proximity of the frame embeddings to the query embedding. In our model, we have two parallel networks for the images and queries separately which are jointly trained with a common loss function. The network is built in an end-to-end fashion for the training and the inference. During training, we feed the images as input to the CNN part of the network and their corresponding queries to the query embedding model as shown in Figure 1.3 and learn their projection to the same space.

### 4.1 Query Representation

For the input textual query, our model employs the word2vec word embedding model which is pre-trained on a corpus of  $73.6M$  queries, to represent the queries as 300 dimensional dense vectors. We choose the embedding space to be  $300D$  to make a good compromise between training speed, semantic quality and appreciative performance [25]. We projected every individual word of the queries into this embedding space. More than the individual embedding, it is important to get the query word importance and the context information in the query. Hence, we propose to use a LSTM network which takes a sequence of  $N$  words of the query. The average number of words in the query was around 5. However, we chose  $N$  to be 14 and clipped the words beyond in the query. The choice of  $N$  is a compromise between maximum length of query and the mean length. Though the maximum length of the query is 23, there are less than 0.1% of queries that has more than 14 words in them. We use LSTM networks to encode this variable length input query to yield a fixed dimensional vector which maps the query in the  $300D$  latent semantic embedding space. The LSTM network can result in a better vector representation for the query for it learns the importance of each word in the query and their word combinations. We perform the training of the LSTM networks from random initialization of weights. The LSTM model is as follows:

$$Q = \{f(q_1), f(q_2), \dots, f(q_n)\}, \text{ where } n \text{ is the \#words in the query,}$$

$$t = LSTM_{\theta_l}(Q),$$

where  $Q$  is the sequence of word2vec dense vector representation of vectors while  $q_i$  is an one-hot column vector at the  $i^{th}$  word in a word vocabulary. The function  $f$  indicates the word2vec word embedding model trained on complete set of queries of the MSR dataset. We use pre-trained word2vec model on Google News Dataset before we finetune it for the queries of MSR dataset. This word embedding model is kept unchanged during the training.  $LSTM(Q)$  with  $\theta_l$  represents the LSTM networks that takes in the variable length query as input and transforms it to yield a fixed length dense vector of  $300D$ . The final representation  $t$  for a query is a function of all the words in the query, aligning more strongly towards the words representing the query content. The LSTM consists of two hidden layers with 512 units each. All the weights are randomly initialized and we trained all sets of weights using Adaptive sub-gradient method (adagrad) [19] for updates with a fixed learning rate.

## 4.2 Image Representation

For representing images, we leverage the deep convolutional neural network (CNN) architecture of VGG-19 [2] by using their model with the pretrained weights. VGG-19 has 19 layers of which we use the first 18 layers and replaced the softmax prediction layer with a projection layer  $M$ . The layer  $M$  takes the input from the fully connected layer fc2 of VGG-19 and output a dense vector representation of image in the same latent semantic embedding space where queries occupy. Image embedding can be formulated as:

$$v = W_m(CNN_{\theta_c}(I)) + b_m,$$

where  $CNN(I)$  with parameters  $\theta_c$  transforms the image to yield 4096 dimensional activation units of the fully connected layer immediately before the softmax prediction layer. The matrix  $W_m$  and bias  $b_m$  are parameters of the projection layer  $M$  where  $W_m$  has dimensions  $h \times 4096$ , where  $h$  is the size of the Latent Semantic Embedding Space ( $h = 300$ ).  $v$  is the set of  $300D$  output vectors of the input images.

## 4.3 Objective function

We have the visual semantic embedding model as in Fig 4.1 that maps every image and the query into a set of vectors in a common  $300D$  latent semantic embedding space. We consider both positive and negative examples for the training. Positive pairs are set of queries and their corresponding clicked images as pairs while negative examples are set of queries and randomly selected images whose query has a low cosine similarity with the query of the positive example. Intuitively, positive examples would have a higher cosine similarity score  $h$  among the query-image pair than the negative example. This can be formulated as:

$$h(t, v^+) > h(t, v^-),$$

where  $t$  represents the query embedding vector while  $v^+$  and  $v^-$  are the embedding vectors of  $image^+$  and  $image^-$  respectively. This requires the positive query-image pair to score higher than negative pairs. Let us see the loss function for this ranking problem.

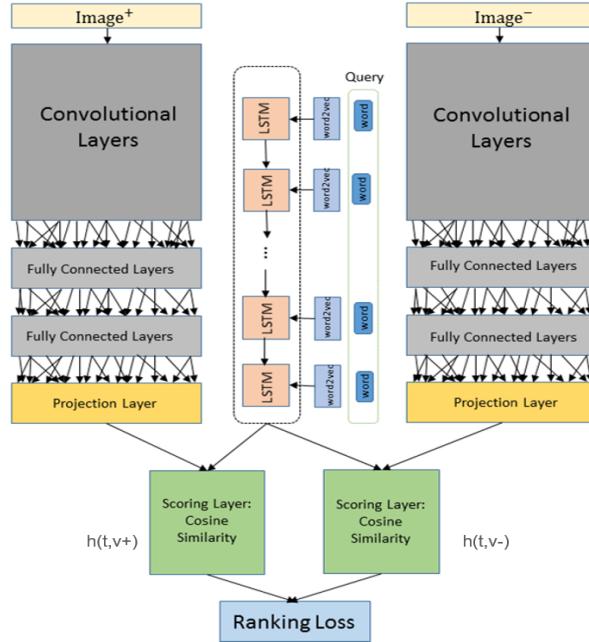


Figure 4.1: **Siamese Network.** Scoring layer computes cosine similarity between query and frame embedding output vector and the ranking loss enforces the objective of scoring positive pair higher than the negative one.

## 4.4 Loss function

The loss function should enforce the objective function by requiring the query embedding with respect to the positive image to have a higher cosine similarity compared to the embedding w.r.t. to the negative image. There are many choices for the loss function of ranking problem.

### 4.4.1 L1, L2 rank loss

Commonly used loss function for the ranking problem is an  $l_p$  loss, defined as:

$$l_p(t, v^+, v^-) = \max(0, \gamma - \hat{\mathbf{v}}^+ \hat{\mathbf{t}} + \hat{\mathbf{v}}^- \hat{\mathbf{t}})^p$$

where  $p=1,2$  are the common choices. The  $l_p$  loss imposes the ranking constraint to make a positive pair to score higher than the negative one by a margin of  $\gamma$ . In the case of margin violation,  $l_1$  loss applies a linear loss, while  $l_2$  incurs a quadratic loss. In the above equation,  $t$  is the unit length normalized vector representation of the query in Latent Semantic Embedding Space and  $\mathbf{v}^+$  is unit length normalized output from CNN part of the network for the input image from the given query-image pairs while  $\mathbf{v}^-$  is the unit length normalized CNN network output for a random image whose corresponding query has a low cosine similarity (inner product of unit length vectors) ( $< 0.5$ ) with the query  $Q$ .

#### 4.4.2 Huber rank loss

MSR Clickture dataset is a compiled online data of triplets of user queries, the clicked images and number of clicks correspondingly. The noise in this type of data is inevitable. The main drawback of  $l_1$  loss is that it over-penalizes over small margin fluctuations while  $l_2$  loss penalizes the outliers heavily due to its quadratic function [9]. To overcome this, we use the Huber loss formulation [10] to the ranking setting. The Huber loss is:

$$l_{Huber}(t, v^+, v^-) = \begin{cases} \frac{1}{2}l_2(t, v^+, v^-), & \text{if } u \leq \delta \\ \delta l_1(t, v^+, v^-) - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}$$

where  $u = \gamma - \mathbf{v}^+ \mathbf{t} + \mathbf{v}^- \mathbf{t}$ ,  $\delta$  decides the point where Huber loss becomes linear. From the equation, we observe that the Huber loss penalizes quadratically for small margin violations and linear for strong violations.

We perform the experiments on all three losses and have shown the comparison plots as in Chapter 7. We infer that the Huber loss performs better than  $l_1$  and  $l_2$  losses. We set the parameter of margin  $\gamma$  as 1 and  $\delta$  as 1.5. We explain the choice of these parameters in Chapter 7. Loss function is computed as the summation over a mini-batch of 128 query-image pairs and it encourages to align the positive query-image pair in the embedding space and distance apart the negative query-image pair. We use the unit length normalized output vector representation for the query and image in the loss function because we observe that the mapping of the queries and images become highly divergent in the case of unnormalized inner product in the loss function.

### 4.5 Training Data

MSR Clickture dataset has a huge collection of queries, clicked image URLs and the number of clicks. For the model training, we extract triplets of  $\{query, image^+, image^-\}$  where  $query$  is a text searched by the user,  $image^+$  is the corresponding image clicked for the query which is clicked by the user,  $image^-$  is any random image from the dataset collection whose corresponding query has a cosine similarity less than 0.5 with that of query that the user searched for.

### 4.6 Validation

For validating the network, we calculate the accuracy of the network as the percentage of number of triplets  $query, image^+, image^-$  which has a higher cosine similarity score between  $query$   $image^+$  pair than the  $query$   $image^-$  pair. The expected random performance of any model for this score function is 50%.

## Chapter 5

# Query Agnostic Model

In the previous model, the Query Image Embedding model learns the semantic relation between queries and video frames by embedding them into a common latent semantic embedding space. Query Agnostic (QA) model tries to rank those frames high which are aesthetically closer to a photograph than a video frame, which is more likely to be blurry and poorly composed. The assumption is that the images/photographs are generally well composed compared to video frames. This may be because the videos generally comprise of moving objects/scenes or they are taken with moving cameras which in turn make the frames blurred and less composed. This difference between an image and video frame can be leveraged to select a thumbnail which can be assumed to be more close to an image than a video frame. The QA model will be worthy of learning generic properties of an image indicating whether it is well-composed, independent of the content.

We add an extra node to the projection layer that learns to provide the query agnostic score for each video frame. In this formulation, we use the same ranking problem constraints as seen in Chapter 3. However, we differ in negative examples used for training. In triplets  $T$  of  $\{query, image^+, image^-\}$  as the training data,  $query$  and  $image^+$  are query-image pairs from MSR Clickture dataset while 10% of  $image^-$  are video frames taken randomly from the collections of videos of the video2GIF video dataset [9]. The ranking formulation is:

$$s(image^+) > s(image^-), \forall (image^+, image^-) \in T,$$

where  $s$  denotes the query agnostic score. The above ranking formulation is combined with the ranking formulation of query Image embedding model as in Chapter 3. Instead of embedding images to a  $d$  dimensional vector, the network is learned to project an image to  $d + 1$  dimensional vector output of which  $d$  dimensions mark the image embedding in the  $d$ -dimensional visual semantic vector space and the  $(d + 1)^{st}$  dimension represents the query agnostic score. The overall loss function is:

$$\begin{aligned} l_p(t, v^+, v^-) &= \max(0, \gamma - \mathbf{v}^+ \mathbf{t} + \mathbf{v}^- \mathbf{t} - v_{d+1}^+ + v_{d+1}^-)^p, \mathbf{v}_{qa}^+ \in R^{d+1}, \\ \mathbf{t}_{qa} &= [\mathbf{t}, 1], \mathbf{t} \in R^d, \\ \mathbf{v}_{qa}^+ &= [\mathbf{v}^+, v_{d+1}^+], \mathbf{v}^+ \in R^d, v_{d+1}^+ \in R, \\ \mathbf{v}_{qa}^- &= [\mathbf{v}^-, v_{d+1}^-], \mathbf{v}^- \in R^d, v_{d+1}^- \in R, \\ l_p(\mathbf{t}_{qa}, \mathbf{v}_{qa}^+, \mathbf{v}_{qa}^-) &= \max(0, \gamma - \mathbf{v}_{qa}^+ \mathbf{t}_{qa} + \mathbf{v}_{qa}^- \mathbf{t}_{qa})^p, \mathbf{v}_{qa}^+ \in R^{d+1}, \end{aligned}$$

where  $\mathbf{t}_{qa}$  is the query embedding output vector  $t$  appended by a constant which we set as 1.  $v_{d+1}^+$  and  $v_{d+1}^-$  represents the query agnostic score of the positive and negative sample. We set  $d$  as 300. From the equation, we see that QA model is not influenced by the queries. Hence, it tries to learn the intrinsic properties of the images such as aesthetics, interestingness [24, 7], etc. to rank the positive image sample higher than the negative image sample. The  $l_p$  loss is a general loss function used in the literature for imposing the ranking constraint to score the positive sample higher than the negative sample [9]. We also experiment replacing the  $l_p$  loss with Huber loss [10]. The details of the experiment can be seen in Chapter 7.

# Chapter 6

## Submodular Maximization

In the previous chapters, we have seen the extraction of relevant frames to a given text query from a video. However, it is highly probable that the extracted frames would be visually similar as the consecutive frames in an uniformly sampled set of video frames are likely to be close enough in visual space. Among the visually similar (non-diverse) frames, a frame does not give any additional information from other frames. Hence, a subset selection of frames from a video for a given query should be both query relevant and have a diverse set of frames which are visually quite distinct. The problem of subset selection of relevant keyframes avoiding the redundant ones in the context of video summarization, has been solved by different approaches such as Video-MMR[21], submodular maximization[8], and recently by deep architectures[46].

### 6.1 Submodularity

In a mathematical perspective [16], submodularity is a property of set functions that fulfill the property of diminishing returns. Let  $V$  is a finite set and  $S \subset V$  and submodular function  $f : 2^V \rightarrow R$  which assigns a value  $f(S)$  for each subset  $S$ . In our case, ground set  $V$  can be uniformly sampled set of frames of a video and  $S$  be the subset of frames. The key properties of submodular functions are:

- *Diminishing Returns:* A function  $f$  is submodular if for every  $A \subset S \subset V$  and  $e \in V \setminus S$ , it satisfies:

$$\Delta(e|A) \geq \Delta(e|S)$$

where  $\Delta(e|S)$  is the marginal gain in the value of  $f$  by the addition of  $e$ , i.e.  $\Delta(e|S) := f(S \cup e) - f(S)$ .

- *Closure:* Non-negative linear combinations of submodular functions are also submodular. In other words, if  $g_1, \dots, g_n : 2^V \rightarrow R$  are submodular, and  $\alpha_1, \dots, \alpha_n \geq 0$ , then  $f(S) = \sum_{i=1}^n \alpha_i g_i(S)$  is submodular as well [16].

### 6.2 Submodular Functions

Submodular functions have been used for summarization of documents [23, 22], image collection [15] and video summarization[8]. Our problem can be treated as a video summarization problems as in [8]. In our

problem of subset selection of keyframes from a video that are query-relevant and diverse, we have to define submodular functions separately for relevance and diversity. Later, the weighted linear combinations of submodular function jointly optimize for the selection of keyframes that are relevant and diverse. Let us define a video as a set of uniformly sampled frames:  $V = f_1, \dots, f_n$  and  $Y_V$  be power set  $P(V)$  from which we select a optimal subset  $y^* \in Y_V$ .

### 6.2.1 Relevance

We predict the relevance score of each frame locally without taking into account any of the neighbourhood frames. This prediction is done by using the visual semantic embedding model as discussed in Chapter 4. For a given text query, we compute the query relevance score for each frame of the video by inputting the query and frames to the visual semantic embedding model. We use:

$$f^{rel}(y) = \sum_{k \in y} I(k)$$

where  $k$  is a keyframe in the solution  $y$  and  $I$  is the relevance score output of the visual semantic embedding model by the computing cosine similarity between the normalized text and frame embedding. The above function  $f^{rel}$  is a modular function due to its non-dependence on the other selected frames.

### 6.2.2 Diversity

Here, the function scores how well the selected subset is diverse in nature. We use an objective that favours a diverse sets of solutions. Ideally, this diverse set should have keyframes that are at the maximum distance in the visual space from every other keyframe from the same set. We use fc2 features of the fully connected layer of our CNN model 1.3 to map each frame to the Visual Semantic Embedding Space. We fix the cardinality of the output set to be  $k$ . Further, the objective is to select a set of  $k$  frames, such that the sum of distances between the datapoints in the visual semantic space is maximized, i.e.

$$f^{div}(X, y) = \begin{cases} \sum_{i \in y} \min_{j < i} \|x_i - x_j\|^2, & \text{if } i > 1, \\ 1, & \text{otherwise,} \end{cases}$$

where  $X$  is the set of fc2 4096-dimensional feature representation of frames in subset  $y \subset V$ . The function  $f^{div}$  is submodular objective function as it is a monotone function following the diminishing returns property.

## 6.3 Thumbnail Subset Selection Objective

Using the above submodular objectives, we can now estimate the overall thumbnail subset selection objective, similar to summarization objective in [8], which jointly optimize to select keyframes that are relevant and diverse. The objective is a weighted linear combination of the submodular functions defined for relevance and diversity. We are given a video  $V$  and a budget  $B$ , the cardinality constraint for the set. Let  $Y_V$  denotes the set of all possible solutions  $y \subset V$ . The overall objective is:

$$y^* = \arg \max_{y \in Y_V} o(x, y),$$

where  $x$  are the fc2 features extracted from the video  $V$ . We can define the objective as a linear combination of relevance and diversity submodular objectives:

$$o(x, y) = w_1 f^{rel}(x, y) + w_2 f^{div}(x, y),$$

We keep  $w_1$  and  $w_2$  to be non-negative and  $f^{rel}$  and  $f^{div}$  are monotone submodular functions [31]. Though we have the possible number of solutions growing exponentially with the length of the video, we can find a near optimal solution for the above equation in an efficient way using the submodular maximization [31]. Maximization of submodular functions under cardinality constraints can be solved by greedy algorithms with lazy evaluations which yields a good approximation of the optimal solution to the optimization of the above equation[30, 8]. We also learn the weight vector  $w = [w_1, w_2]$  to know the importance of relevance and diversity in the selection of subset of keyframes. We have training data comprising of query-video pair and their corresponding subset of keyframes which cater to query relevance and diversity. We use grid search to learn the weights.



# Chapter 7

## Experiments

### 7.1 Dataset

We use the MSR Clickture dataset in our experiments to learn our model. The complete dataset consists of  $73.6M$  queries and  $40M$  images. Having a huge dataset, we made a subset consisting of 0.5 million  $\{query, image^+, image^-\}$  triplets, where  $query$  is a textual phrase,  $image^+$  is the image clicked for the query,  $image^-$  is any random image from the dataset whose corresponding query has a cosine similarity of less than 0.5 with  $query$ , phrase query. The training and validation data are chosen in the ratio of 10 : 1.

### 7.2 Data Preprocessing

The images are processed to remove its mean value while word2vec operation are done on the words of queries to generate  $300D$  dense vector representation for the query words. These are the preprocessing steps for the preparation of training data.

### 7.3 Implementation

We experimented with different ways of feeding the query-image inputs to the network. Firstly, we prepared the training data on-the-fly during the training and fed to the network. Reading the queries and images from the disk, preprocessing the images to remove its mean value, word2vec operation on queries, etc are done along with the training process. This consumed a lot of time because of the time-expensive preprocessing procedure. Secondly, we tried creating hdf5 files for the training data consisting of the preprocessed images and the dense vector representation of the queries. This significantly improved the training time for each mini-batch. HDF5 has the advantage that it can store very large datasets and provide fast access. However, in our scenario we have a model trained on a relatively large dataset that does not fit in memory. Performing data processing and training in a single process seems a bit less efficient because GPU is idle when hdf5 data is read off the disk and nothing else is done while the GPU is at work. Hence, thirdly, we tried data processing and training in parallel using python package Fuel. Using Fuel, we ran data processing server in a separate process different from the process where we ran the training script. We observe that the Input/Output operations are executed while GPU is busy and thus wasting less time for the data to

be available. Figure 7.1 depicts the training time for training a mini-batch of size 128 query-image pairs, observed in all the above three cases in the respective order. We use NVIDIA GeForce GTX Titan X GPU for our training.



Figure 7.1: Training time for a mini-batch of 128 query-image pairs for the three cases mentioned above.

## 7.4 Network training

We initialize the CNN with the parameters trained from VGG-19 [37]. We use adagrad weight updates [3], with mini-batches of 128 query-image pairs, to optimize the model. We use dropout regularization in the added projection layer while VGG has already dropout in their fully connected layers set to 0.5 [37]. However we keep no dropouts on the LSTM layers [45]. We also use clip gradients at 5 as in [13]. The learning rate starts from 0.01 and it is decreased after every 30-50 epochs. In the CNN part, we maintain 1000 times lower learning rate for pretrained VGG layers than the final projection layer. We propose to train the model in three steps. In order to evaluate our model, we compare the following methods on the dataset:

## 7.5 Baseline

In the baseline framework, we used deep CNN architecture with pre-trained weights from VGG- 19 [37] with an additional projection layer as shown in Figure 1.3 for the image input. For the input query, we use a simple word embedding model- word2vec for each word in the query while mean of all word2vec vector representation of all words in the query is used for the query vector representation. We keep the word2vec model, which is pretrained on the complete dataset, fixed. In the training process, we train only the parameters of the projection layer of the CNN part of the network while holding the query representations fixed. Here, we lock all the top 18 layers of VGG-19 and train only the projection layer on the dataset. The following figure depicts the experimental results of the baseline:

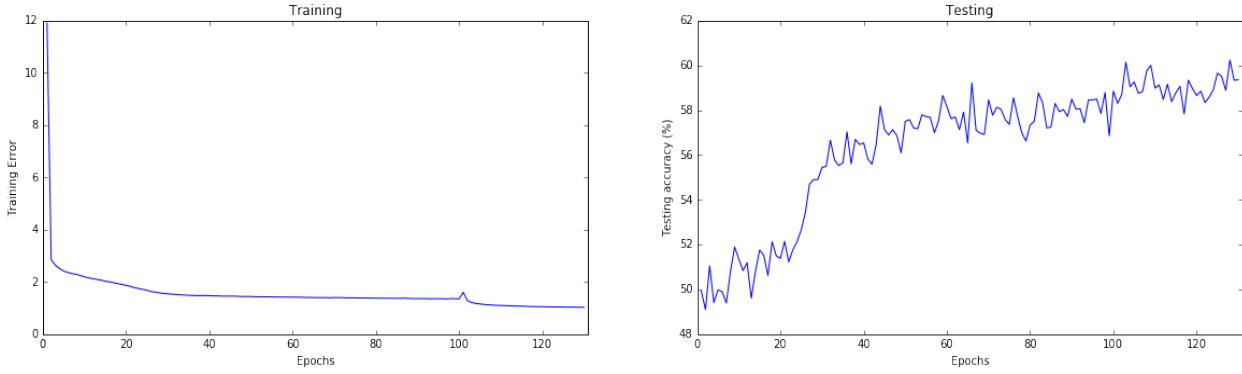


Figure 7.2: Training error and testing accuracy plots for the baseline.

We started the baseline training with the learning rate of 0.01 and then decreased to 0.001 after 100 epochs which is clearly seen in the left side image of Figure 7.2. The right side of the image depicts the testing accuracy of the baseline across the epochs. We decrease the learning rate to 0.001 after 100 epochs after observing the testing accuracy which becomes stagnant between 60-100 epochs. We note that it resulted in the increase of the testing accuracy to 60%. This accuracy is not too far from the random performance of 50%.

### 7.5.1 Loss function Comparison

We compared our baseline with different loss function as mentioned in Chapter 4. Figure 7.3 shows the comparison plot. We observe that Huber loss achieves better testing accuracy compared to  $l_1$  and  $l_2$  loss. The euclidean loss is added to the Huber loss to make a better visualization of 300D query and image embedding in 2D space using t-SNE visualization and it is rather not helpful for better performances.

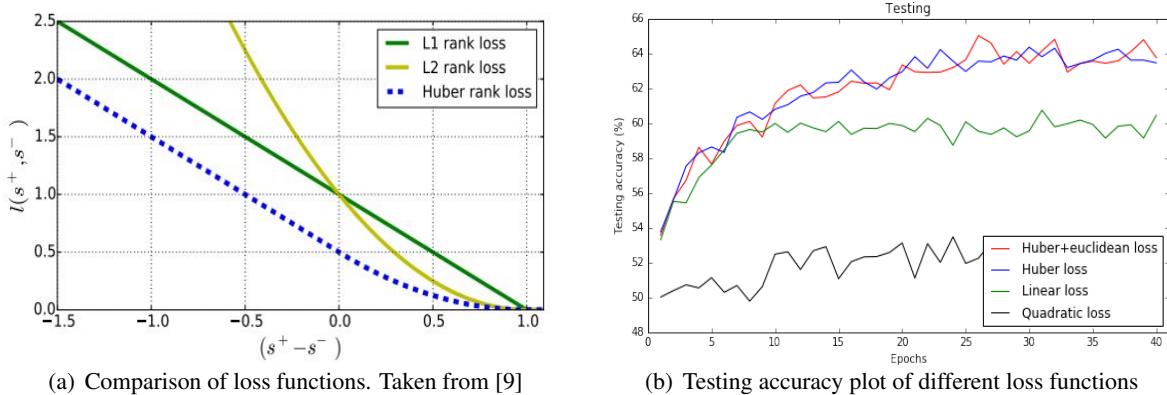


Figure 7.3: Loss Functions

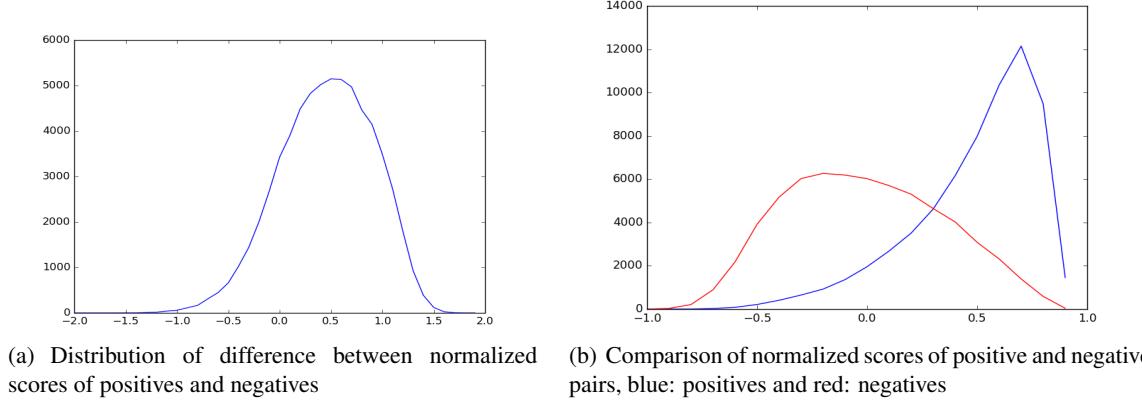


Figure 7.4: Score Distribution with x-axis representing the Scores and y-axis representing frequency

In Figure 7.4, we have plotted the distribution of query relevance scores for the training samples consisting of positives and negatives. Figure 7.4(b) shows clearly that the positive samples have been learned to score higher than the negative samples with the mean of positives being greater than 0.5 and the mean of negatives is less than zero. To be more clear, we also plot the difference between the normalized scores of positives and negatives as in Figure 7.4(a). We take the same parameters for Huber loss function as in [9]. To analyze the chosen parameter, as in Figure 7.4(a), we plot the distribution of difference between scores of positives and negatives where X-axis shows the difference in scores and Y-axis: the number of examples. Since we keep margin as 1 and  $\delta = 1.5$ , we ensure a Huber loss by the following: linear loss for difference in score (X-axis)  $< -0.5$ , a squared loss for  $-0.5 < X\text{-axis} < 1.0$ , and no loss for  $X\text{-axis} > 1.0$ .

### 7.5.2 Query Agnostic model

Here, we train the baseline with the query agnostic (QA) model. Since the aim of the model is to learn the aesthetic property of images independent of queries, we used video frames from arbitrary videos as the negative samples in training. From the evaluation, we note that the network trained with negative video frames alone learn to identify a snap point [44] from the collection of video frames, however it fails to learn the semantic relation between queries and images. We give a mixture of images from both MSR dataset and arbitrary video frames as negative sample to the training. We still observe that QA model did not improve our results for the selection of query relevant thumbnails as in table 7.1.

## 7.6 LSTM Training

Here, we incorporate the LSTM network for the query input in addition to the baseline framework. We feed the variable length sequence of word2vec vector representation of each word of the query to LSTM to yield a fixed length  $300D$  dense vector in the Latent Semantic Embedding Space. On the other hand, we use the deep CNN architecture inclusive of the projection layer exactly from the baseline to map the image in the same embedding space. We keep the CNN part fixed during the LSTM training with the pre-trained weights

from the baseline while we train the LSTM network using random initializations. As a whole, here we train the projection layer and LSTM network jointly.

## 7.7 CNN-LSTM Training

Here, we train the complete model initializing the CNN part of the network from the baseline and the LSTM part from random initializations. However, the derivatives of the loss function are back propagated through all fully connected layers of CNN and LSTM for finetuning.

## 7.8 Evaluation

We test all our methods with all 749 title-video pairs provided by the MSR dataset [25]. We use titles for the evaluation as queries are not provided with the MSR Evaluation dataset. The usage of titles for evaluation has some demerits. Titles are generally free-formed, unconstrained and often ambiguous with many numerous concepts embed into one [39]. Our evaluation is not directly comparable with the evaluations provided in [25] as they evaluated the models using queries while we evaluate using the titles. Hence, we re-implemented their method to make a direct comparison. In order to evaluate different methods for query-dependent thumbnails selection, we compare the following methods:

- *Random* Here, the method randomly selects one image from the candidate thumbnails as final thumbnail, as required by the HIT@1 metric.
- *Liu et al*[25] Query-dependent thumbnail selection method where the deep visual-semantic embedding model is trained on 0.5M subset from click through based image dataset.
- *Ours:  $l_1$*  Our model trained with  $l_1$  loss on 0.5M subset from click through based image dataset. It differs from [25] in the objective function by using  $\{query, image^+, image^-\}$  triplets and learning to score a positive image higher than any random image w.r.t. a query.
- *Ours:Huber* Our model trained with Huber loss on 0.5M subset from click through based image dataset.
- *Ours:Huber+QA* Our model with the query agnostic model, trained with Huber loss on 0.5M subset from click through based image dataset.
- *Ours:Huber+LSTM* The deep Visual Semantic Embedding Model is same as the *Ours:Huber* except for the use of LSTM query model instead of the average word2vec embedding model for the query. The model is trained with Huber loss on 0.5M subset from click through based image dataset.
- *Ours:Huber+LSTM+QA* The above model is trained as query agnostic model with Huber loss on 0.5M subset from click through based image dataset.
- *Ours:CNN-LSTM* Here, we finetune the CNN on *Ours:Huber+LSTM*, and trained with Huber loss on 0.5M subset from click through based image dataset.

We evaluate our methods by the criteria HIT@1 as in [25]. The HIT@1 is computed by the hit ratio for the highest ranked or first selected thumbnail. They also compute the Mean Average Precision (mAP) over all the candidate thumbnails by:

$$mAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}),$$

where  $Q$  is the query set,  $m_j$  is the number of positive thumbnails in each query-video pairs.  $Precision$  is the average precision at the position of returned  $k^{th}$  positive thumbnails. The HIT@1 results are shown in the table. The Labels in the table VG represent Very Good and G represents Good. The thumbnails in the evaluation set is labelled for a given query by five different scores: VG:Very Good, G: Good, F: fine, B:Bad, VB:Very Bad.

Method	HIT@1: VG	HIT@1:VGorG	Spearman Correlation	mAP
Random	$28.2 \pm 1.5$	$57.17 \pm 1.5$	-	-
Liu et al(Baseline)[25]	33.00	59.81	0.112	0.603
Baseline[25]+LSTM	32.03	60.48	0.138	0.607
Ours:L1	32.42	62.61	0.139	0.611
Ours:Huber	32.61	62.21	0.132	0.608
Ours: Huber+QA	31.83	61.81	0.149	0.603
Ours: Huber+LSTM	32.42	63.15	0.178	0.621
Ours: Huber+LSTM+QA	35.93	63.28	<b>0.183</b>	0.619
Ours: CNN-LSTM	<b>37.11</b>	<b>66.22</b>	0.179	<b>0.626</b>

Table 7.1: Comparison of the thumbnail selection methods

For each model in the table 7.1, the numbers of mAP are obtained by averaging the predictions obtained from the models at each epoch. From the table 7.1, we can infer that the Ours:L1 and Ours:Huber performs on a par with Liu et al [25]. Liu et al model has been re-implemented to compare with our approach as they evaluated the models with queries while we evaluate using titles (queries not available). We observe that query agnostic model does not seem to have learned a better model for thumbnail selection. However, the addition of LSTM to the baseline model and a better objective with triplets  $\{query, image^+, image^-\}$  clearly seem to have a positive effect. The performance on the evaluation dataset has improved by around 4% in HIT@1:VG and 5% in HIT@1:VG or G, compared to the baseline.

- We can infer that the improved objective with triplets  $\{query, image^+, image^-\}$  provides a improvement of 1.4% on mAP, observing *Baseline + LSTM* and *Ours : Huber + LSTM*.
- We also observe that LSTM boosts the performance in the two following cases: a) *Ours : Huber + LSTM* improves by 1.3% over *Ours : Huber* b) *Ours : Huber + LSTM + QA* does 1.6% improvement over *Ours : Huber + QA*.

We tried different combinations of methods before we finalize our methods for the final evaluation on different datasets. We have tabulated our results on the final model on 3 different dataset in Chapter 9.

## Chapter 8

# New Evaluation Dataset: Relevance and Diversity (RAD)

We propose a new dataset, designed to support the evaluation of relevance and diversity in the selection of query dependent video thumbnails from videos. The dataset comes with the associated relevance and diversity assessments performed by human annotators. We crowdsource the annotation task using Amazon Mechanical Turk (AMT).

## 8.1 Dataset Description

We create a dataset consisting of 100 text query-video pairs. Given the significant role of selection of text queries, we take the top searched queries between 2008 and 2016 in YouTube search engine, which is made publicly available by Google <sup>1</sup>. In order to choose a diverse set of videos, we extract the top searched queries from 22 different video categories: Arts and Entertainment, Autos and Vehicles, Beauty and Fitness, Books and Literature, Business and Industrial, Computer and Electronics, Finance, Food and Drink, Games, Health, Hobbies and Garden, Internet and Telecom, Jobs and Education, Law and Government, News, People and Society, Pets and Animals, Real Estate, Science, Shopping, Sports and Travel. We note that the extracted queries from each category are certain general terms that are associated with that category. For instance a) Art and Entertainment - music, remix, full movie and others b) Food and Drink - cake, chocolate, etc. We observe that the extracted queries are not descriptive in nature. To make the query descriptive and natural, we use the above queries to search in YouTube again to avail the auto-search suggestions. Subsequently, we extract the top ten search suggestions for each of the above non-descriptive query for all the categories. We use search suggestions as the final queries for our dataset which not only become more descriptive in nature but also emerge to be trending queries of YouTube search engine. Thus, we compiled 100 queries for our dataset.

For the video selection, we extract the highest ranked video along with its metadata from the retrieved video results in YouTube search engine. The videos span 2-3 minutes long for each query. We also ensure that the retrieved videos are under Creative Commons licenses that allow redistribution. For each video, the retrieved metadata consist of video's id, title, URL of video from YouTube. Thus, we have 100 query-video

---

<sup>1</sup><https://www.google.com/trends/explore>

pairs for our dataset.

## 8.2 Dataset Annotation

Ground truth annotation of the dataset is dependent on the use case scenario of the annotated dataset. The proposed dataset is annotated in the view of evaluation of relevance and diversity associated with query dependent video thumbnails selection from videos. The annotation task is performed by crowd workers at AMT. We employ a web application<sup>2</sup> to facilitate the process of annotation. The definitions of relevance and diversity are adopted as:

**1. Relevance:** A video frame is considered to be relevant for the query if the frame depicts the meaning of the text query. Bad quality photos such as the one that is severely blurred, out of focus, etc are not considered relevant.

**2. Diversity:** A set of selected video frames are diverse if it depicts different visual characteristics of the query i.e. most of the perceived visual information is different from one frame to another.

In our Human Intelligence Task(HIT) annotation task, there are two subtasks- one for relevance and the other for diversity. Each HIT task is shown only one query-video pair. Here, the annotator is asked to perform the relevance task followed by the diversity task. Both the tasks are not time restricted. Figure 8.1 depicts the instruction page provided to workers.

Go To Task

Image Annotation Task: Relevance and Diversity

**Instructions**

This assignment consists of two related tasks. First, you will complete the relevance task, where you will assess the relevance of all the images to a test query (a phrase). In the second task, the diversity task, in which you will group frames of a video by visual similarity. This task is called the diversity task because the number of groups created will reflect the diversity of the visual content within the video.

- First, The Relevance task
  - You are given frames from a video in sequential order
  - Your task is to assess how relevant the given frames are to a test query
    - If you don't understand the query, please skip this task or search the web to get more information
  - Annotate the given frames as Very Good, Good, Not Good or Trash.
    - Please label the frames as Trash only when they are "bad", e.g. blurry, don't have any clear objects, are dark or that show cuts between video sequences
    - After annotating the frames, press the "Submit Relevance" button
- Second, The Diversity task
  - You are shown the same frames as before
  - Your task is to assign frames of the video to several groups, according to their visual similarity.
    - Place the visually similar frames into the same group
    - Groups need not consist of consecutive frames. Jumping around to select frames is fine too
    - You can create as many groups as you like
    - After finishing this task, press the "Submit HIT" button
  - After completion the HIT will be completed

**Task 1:Relevance Task**

- Annotate Relevance for each frame one by one.
- Use Keyboard Arrow Keys/Mouse Clicks to navigate and annotate the relevance as: Very Good, Good or Not Good

**Task 2:Diversity Task**

- Group Visually Similar frames. For eg: images within the red box
- Press **Ctrl** + Click the first and last frame to select multiple frames continuously.
- Select the frames and then drag&drop to one of the box to group the visually similar images.

Figure 8.1: Snapshot of Instructions page of Annotation Tool

<sup>2</sup>[https://people.ee.ethz.ch/~arunv/div\\_rel\\_annotator](https://people.ee.ethz.ch/~arunv/div_rel_annotator)

### 8.2.1 Relevance Annotation Task

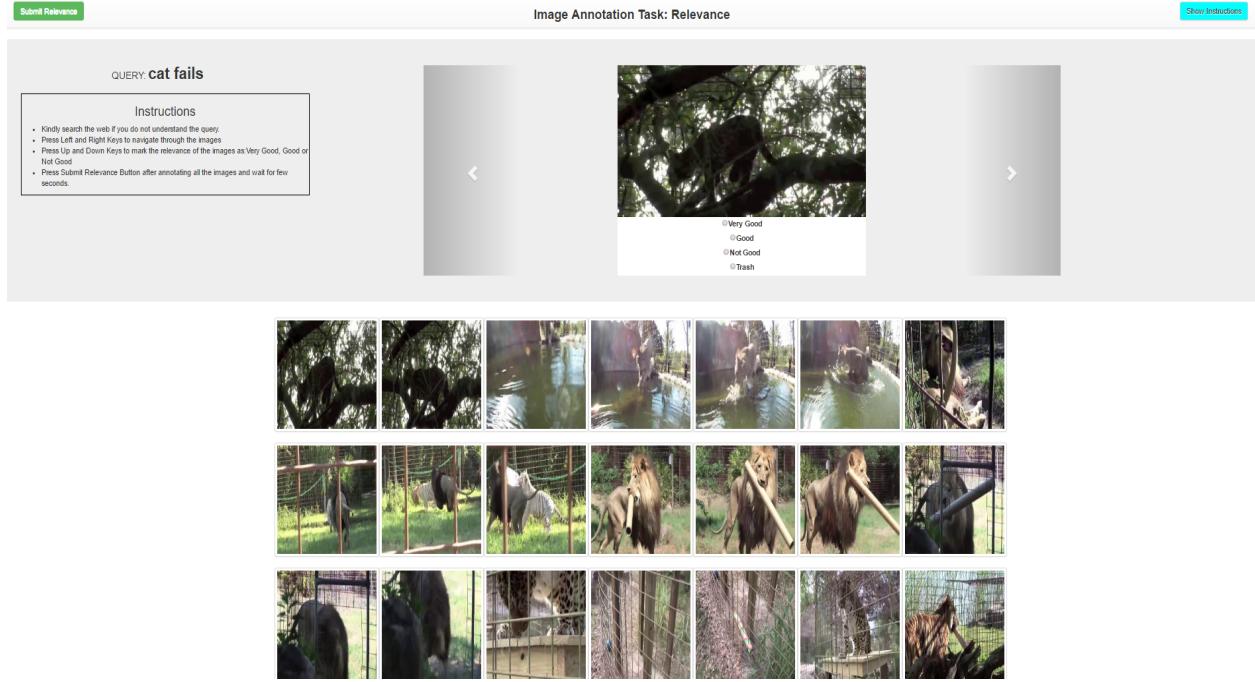


Figure 8.2: Snapshot of the Relevance Annotation Tool

In this task, annotators are allowed to annotate the relevance for all the frames one at a time. Relevance is annotated for all the frames one by one so that the annotator gets a good idea about the video content. This helps in the following diversity task to group the frames easily. The annotators are shown the frames one by one and are asked to annotate their relevance with respect to the query as "Very Good", "Good", "Not Good" or "Trash", which are the label ratings for query relevance. The annotator is also shown the query throughout the process. The snapshot of relevance task in our annotation tool is shown in Figure 8.2.

### 8.2.2 Diversity Annotation Task

Figure 8.3 is the snapshot of diversity annotation task in our web application. Diversity is annotated for all video frames of a video. The video frames are sampled uniformly from videos at a rate of 1 frame per sec(fps). Annotators are provided with a thumbnail list of these video frames. The annotators are also provided with boxes and they are required to group the frames in these boxes/clusters based on the visual similarity. One box should ideally contain only visually similar frames. All the frames that are annotated as "Trash" in the previous task are put in the Group0 automatically. The web application facilitates the option of selecting the group of video frame thumbnails and drag&drop (select, drag and drop using mouse) option to cluster it inside the given box. Unlike [11], the annotators are allowed to create arbitrary number of boxes/clusters. Full size versions of the video frame thumbnail could be seen by clicking on them. The web application also provides other utilities like resetting the diversity task, undo the last step of drag&drop

and others.

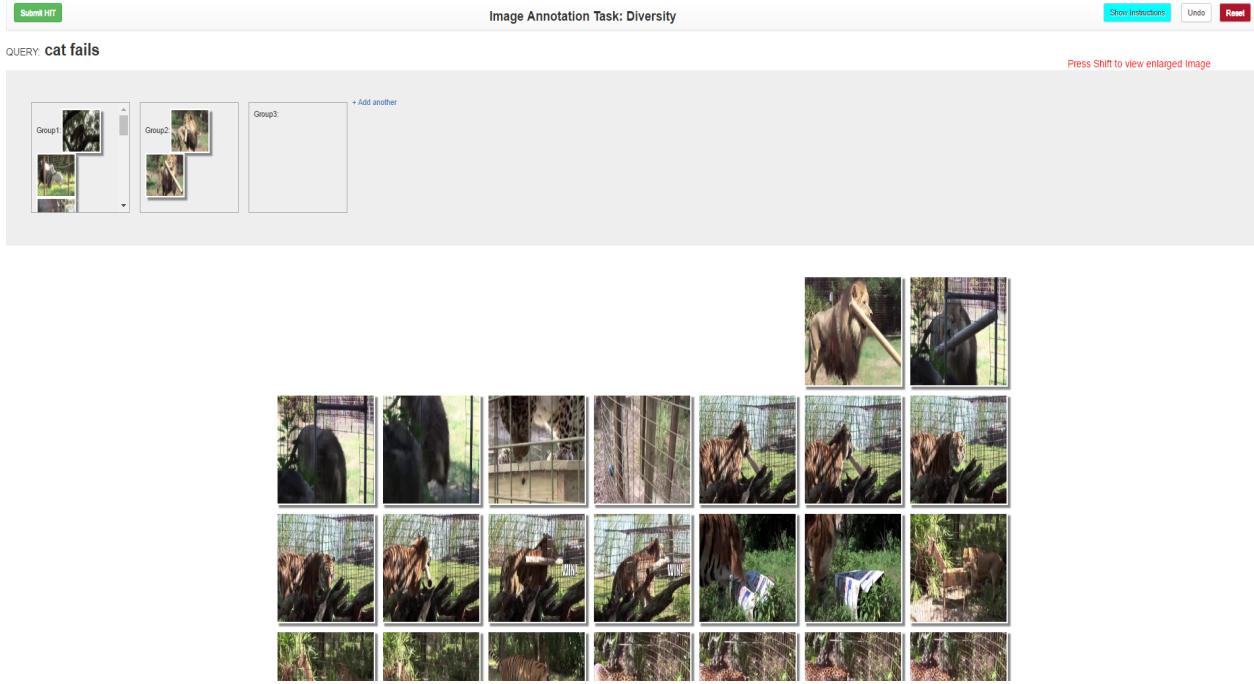


Figure 8.3: Snapshot of the Diversity Annotation Tool

## 8.3 Ground Truth Generation

The relevance and diversity ground truth is collected from five different annotators i.e. each HIT task is performed by five different workers. For every query-video pair in the dataset, we have ground truth annotations of relevance and diversity task from five different annotators. For certain tasks, we need to have a single ground truth annotation of relevance score and diversity (cluster number) information. Since annotations are performed by five different workers and tasks are being subjective, it is likely that annotations are bound to have some score variations. We combine the annotations of relevance and diversity tasks separately from five different annotators.

### 8.3.1 Combine Annotations

In the relevance task, for every query, annotators annotate the relevance scores for all the frames of a video. The 4 different labels of relevance scores are mapped as follows: a) 3-Very Good b) 2-Good c) 1-Not Good and 0-Trash. For each frame, we perform majority voting from five annotations to decide one relevance score. We also tried averaging the relevance scores where we observe the combined relevance score to have been affected by the outlier annotation of one of the workers. Majority voting seems to be more reliable than averaging the scores for combining the relevance annotations.

In the diversity task, it is not intuitive to combine the cluster information (clustering) of five annotations. An important criteria for comparing clusterings is based on counting the number of pairs of points on which two clusterings agree/disagree. We plot a similarity matrix  $K \times K'$ , whose  $kk'$ 'th element is the number of clusters in which frames  $f_k$  and  $f'_{k'}$  occurs together.

$$n_{kk'} = \|\ C_{ij} \| \ f_k \in C_{ij} \text{ and } f'_{k'} \in C_{ij},$$

where  $i$  represents the clustering number and  $j$  represents the cluster number within the clustering.

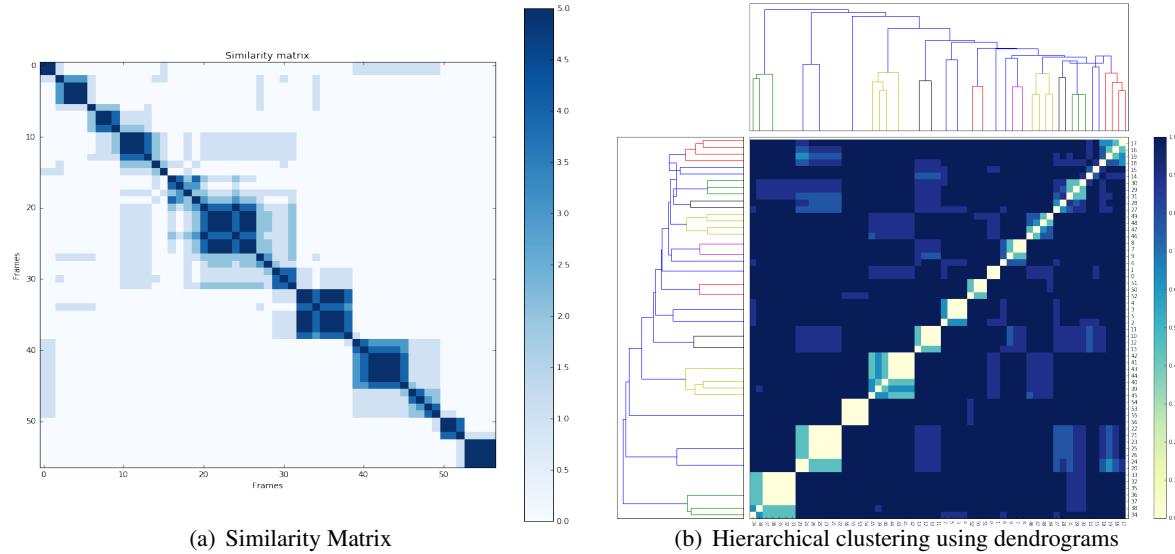


Figure 8.4: Similarity matrix based on the diversity annotations on the left and hierarchical clustering to group the annotations

Figure 8.4 shows the similarity matrix on frames obtained from a particular query-video annotated by five annotators. The video has 58 frames which forms the X and Y-dimensions for the similarity matrix.  $S_{ij}$  element of the similarity matrix represents the number of clusters in which frames  $f_i$  and  $f_j$  occur together. The matrix is a symmetric matrix and square blue boxes groupings in the matrix represents the clusters. Matrix should have contained 5 or 0, if the annotations from five annotators are alike. But, we have variations in the annotations due to its subjective nature. Each annotation has different number of clusters and hence we take the median of these numbers as the number of clusters in the resultant clustering. We tried hierarchical clustering, spectral clustering and k-means clustering algorithms (python scikit-learn) to obtain the resultant clustering with the median number of clusters. Figure 8.4 (right) depicts the hierarchical clustering performed on similarity matrix data. [41]

## 8.4 Annotation Statistics

The relevance and diversity ground truth is collected from five different annotators i.e. each HIT task is performed by five different workers. It is important to measure how much annotators agree themselves

in their annotations. The agreement between pairs of annotators is to be calculated using statistics that measure the level of agreement discarding the agreement by chance. Let us analyze the Relevance and diversity annotations separately.

### 8.4.1 Correlation:Relevance Task

Relevance Task allows an annotator to label each of video frame as: Very Good, Good, Not good or Trash. In Figure 8.5, we have plotted the distribution of relevance label annotations over the dataset of 100 videos in the decreasing order of the length of the video. In the relevance task, we have relevance label annotations from five annotators. Before plotting the distribution plot, we combine the annotations of 5 annotators to a single ground truth relevance annotation. We use majority voting to combine from five labels for each frame of the videos. The distribution of labels over the dataset is VG: 16.73%, G: 61.61%, NG: 13.58% and Trash:8.08%.

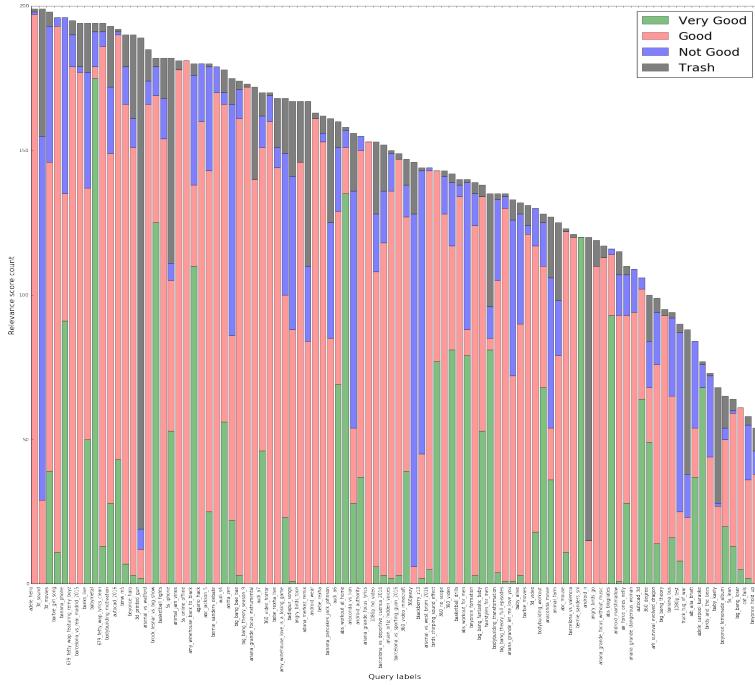


Figure 8.5: Distribution of Relevance score over the dataset

The relevance task labels are mapped to relevance scores as mentioned above. Hence, we have a set of relevance score as a vector from each annotation. This makes the correlation analysis of the annotations easier. Spearman's Rank Correlation is one of the common measure of correlation in statistics. Higher the value of correlation, the lesser the variations between the annotations which inturn shows higher agreement between the annotators.

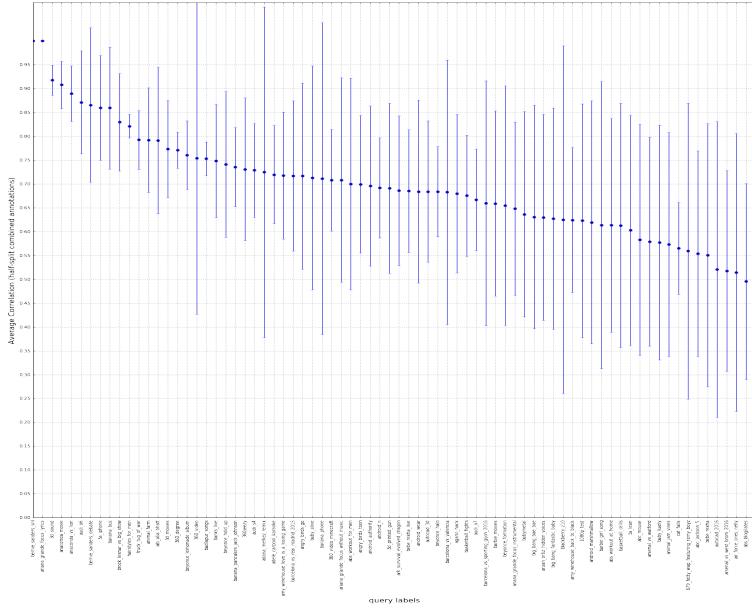


Figure 8.6: Correlation between half-split(2-3 split) combined annotations

To analyze the relevance score annotations, we plot the distribution of Spearman’s Rank Correlation stats for each video in the descending order of the mean values. To make a comparison, we analyze the correlation between the half-split combined annotations, as done by [43]. Each video has five annotations of which we extract all combinations of half-split of 5 to 2 and 3. For each of these half-split of a particular video, we combine 2 and 3 annotations correspondingly into a single annotation each by averaging their labels before we compute the Spearman’s rank correlation. Furthermore, we compute mean and standard deviation of these correlation values over all possible half-splits of the video annotations. We observe that the average of correlation score over 100 videos comes out to be  $\rho = 0.69$  compared to  $\rho = 0.41$  observed in [43] which computes the consistency for labels for a similar task. This  $\rho = 0.69$  shows that our annotations are of high quality. However, we observe low scores of correlation for the following videos:

- Videos created from a few set of images such as lyrics of music videos.
- Videos with minute visual changes and captured from a stationary camera.

The above cases are ambiguous for the workers to annotate. Hence, the correlation scores are low. We plan to remove such videos such as lyrics videos in the later annotation of the dataset.

### 8.4.2 Normalized Mutual Information:Diversity Task

In figure 8.7, we see the number of clusters in each video as annotated by AMT annotators. The green, blue and red plots correspond to the maximum, mean and minimum number of clusters respectively in each video annotation. To analyze and combine these five clusterings of each video, we have various metrics used in the literature such as Variation of Information, Normalized Mutual Information and among others. [41]

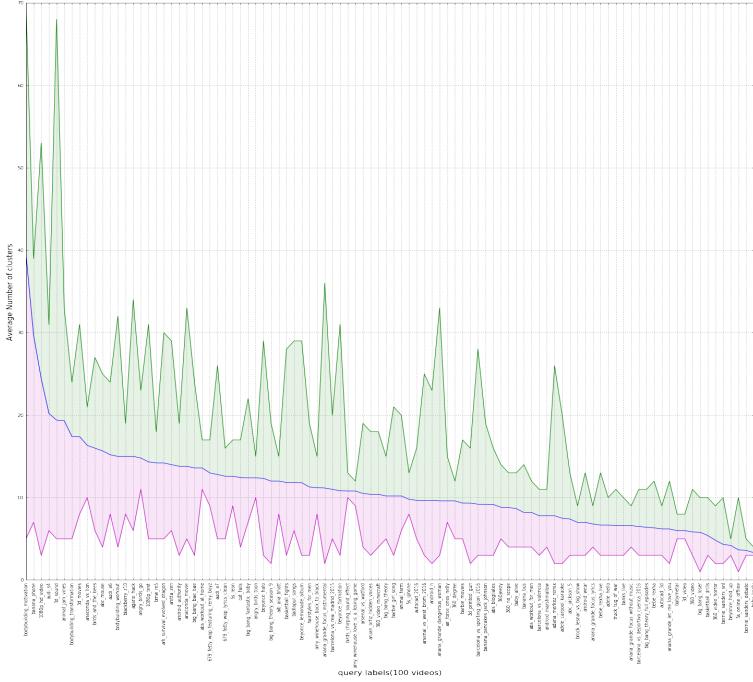


Figure 8.7: Distribution of number of clusters over the dataset

Variation of Information(VI) is a metric which measures the amount of information lost and gained in changing from clustering  $C$  to clustering  $C'$ [28]. This metric can be used to find the VI metric between all the pairs of annotated clusterings. The VI metric is:

$$\begin{aligned}
 VI(C, C') &= H(C) + H(C') - 2 * I(C, C'), \\
 H(C) &= -\sum_{k=1}^K P(k) \log(P(k)), \\
 I(C, C') &= \sum_{k=1}^K \sum_{k'=1}^{K'} P(k, k') \log\left(\frac{P(k, k')}{P(k)P(k')}\right), \\
 P(k, k') &= \frac{\|C_k \cap C'_{k'}\|}{n}, \\
 P(k) &= \frac{n_k}{n},
 \end{aligned}$$

where  $P(k)$ ,  $k = 1, \dots, K$  and  $P(k')$ ,  $k' = 1, \dots, K'$  are the random variables associated with the clusterings  $C$ ,  $C'$ .  $P(k)$  is the probability of a frame to be picked from cluster  $C_k$ .  $n(k)$  is the number of frames in cluster  $C_k$  while  $n$  is the total number of frames. Let  $P(k, k')$  represent the probability that a frame belongs to  $C_k$  in clustering  $C$  and  $C_{k'}$  in  $C'$  which is the joint distribution of the random variable associated with the two clusterings.

VI is calculated using  $H(C)$  and  $I(C, C')$ .  $H(C)$  is entropy associated with cluster  $C$  which is the uncertainty of a frame to be in which cluster is it going to be in.  $I(C, C')$ , mutual information between two clusterings  $C, C'$  is the information that one clustering has about the other. Intuitively, suppose we have a random frame from video  $V$ . The uncertainty about its cluster in  $C'$  is measured by  $H(C')$ . Suppose

it is known about the cluster the frame belong to in clustering  $C$ . Then, how much does this knowledge reduce the certainty about  $C'$ ? This reduction in uncertainty averaged over all the frames is  $I(C, C')$ [28]. Having the entropies of  $C$  and  $C'$  and the mutual information between the clusterings, we get the  $VI$  metric comparison the two clusterings. If the two clusterings are identical, we have  $VI = 0$ . As the value of  $VI$  grows, the information shared between the clusterings decreases. The main properties of  $VI$  are: a) Symmetric b) Follows Triangle inequality. As a consequence of these properties, with this metric, we can move from comparing two clusterings to analyzing large sets of clusterings[28].

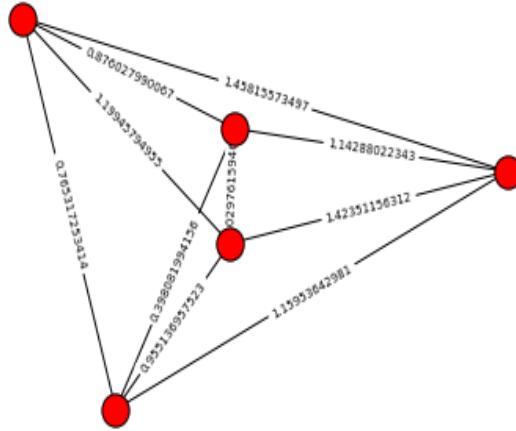


Figure 8.8: **VI metric(distance) graph.** Nodes represents the clusterings and edges represents the distance between them. This graph is computed on a video from RAD dataset.

Figure 8.8 is plotted by computing the  $VI$  metric between all pairs of clusterings obtained from the diversity annotation task of five annotators. The closer the nodes in the graph, the more is the agreement between the annotators.

Variation of Information has good properties to be a metric for comparing two clusterings and can compare the consistency of labellings for a video with the unnormalized values of  $VI$  to be identical ( $VI = 0$ ) or random ( $VI = \text{large number}$ ). However, the normalization of  $VI$  is important to get the distances comparable across videos. In the literature,  $VI$  has an upper bound  $VI(C, C') \leq 2 * \log(K)$ ; where  $K$  is the maximal number of clusters of any dataset in the experiment. In our dataset with videos of different lengths, the true  $K$  is ambiguous as we do not have a clear maximum number for the number of clusters over all videos. To make the range of  $VI$  to be  $[0,1]$  for every video, we have to use  $K$  as the total number of video frames in that video but this makes  $VI$  metric to be depended on the length of video.

For the above limitations of  $VI$ , we use Normalized Mutual Information(NMI) metric, a comparable metric to  $VI$  using the measures of mutual information and entropy. It satisfies the properties: 1. Range of NMI is  $[0, 1]$  where pair of clusterings reaches 1 if identical while the value 0 if the pair of clusterings is

independent; 2. The metric is independent of number of clusters in the clusterings and number of elements in the clusters; 3. Same properties of VI such as Symmetric, triangle inequality. NMI between two clusterings is defined as:

$$\frac{VI(C, C')}{H(C) + H(C')} = 1 - \frac{2 * I(C, C')}{H(C) + H(C')},$$

$$NMI(C, C') = \frac{2 * I(C, C')}{H(C) + H(C')},$$

$$0 \leq NMI(C, C') \leq 1.$$

In the above equation, we shall clearly see the relation between  $VI$  and  $NMI$ .  $NMI(C, C') = 1$  for  $C = C'$  and  $NMI(C, C') = 0$  if  $P(i, j) = 0$  or  $P(i, j) = P(i).P(j)$  (from the definition of mutual information). The denominator of  $NMI$  becomes zero if the both clusterings are trivial. In the annotation of our dataset, trivial clusterings are clusterings where annotators have annotated all frames of any particular video to a single cluster. We have removed that particular annotation of a worker if the clustering is trivial and a new HIT was reuploaded for that video. In the case where all the annotated clusterings are trivial, we remove the video and its corresponding annotations from the final dataset.

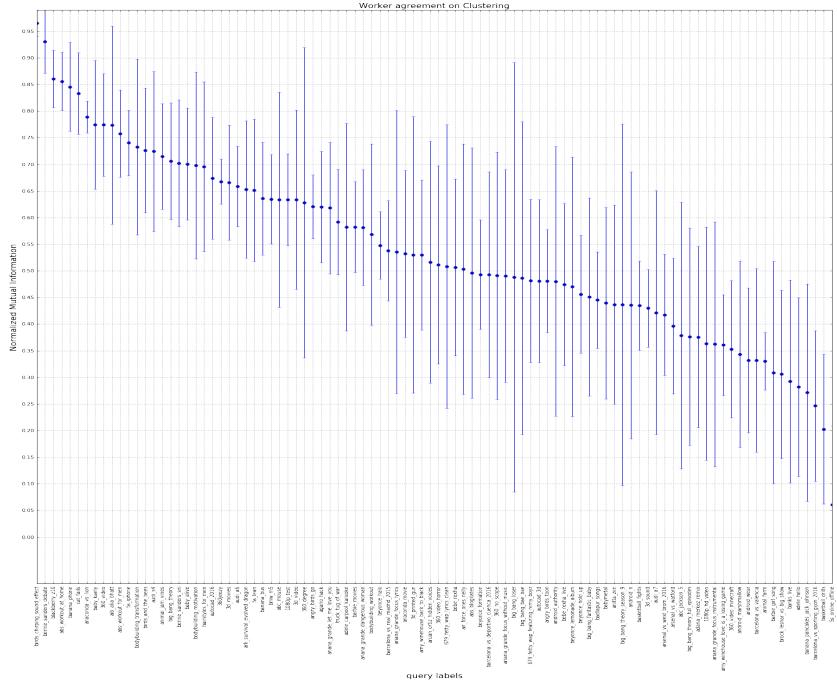


Figure 8.9: Mean and standard deviation of Normalized Mutual Information

Figure 8.9 depicts the mean and the standard deviation of  $NMI$  between the annotators for each video, plotted in the decreasing order of the mean. The above plot has been plotted for 100 videos of our dataset. We note that the average of all the  $NMI$  values over all the videos is 0.541.

## 8.5 Quality Control Procedure

We have conducted more than 600 HITs in AMT comprising of 120 different videos with each having five assignments. Initially, we reviewed annotations manually for the first 200 HITs and we gave qualification for the workers if their annotations were unambiguous. In total, 48 unique workers got qualified from these 200 HITs among 55 workers who participated. So far, we have collected the annotations for 120 videos of which we have removed 12 videos due to its bad quality or being a static video. For annotation of rest of the dataset, we plan to get annotated from these qualified workers to make it more trustworthy.

Figure 8.10 is the snapshot of the visualization page we use to review the annotations. In figure 8.10, left side shows the groups that contain the visually similar frames that the annotator annotated. The first row is the trash which contains the trash frames while the following rows have each row representing a group. We have a slider on top to check the relevance annotation label of each frame individually.

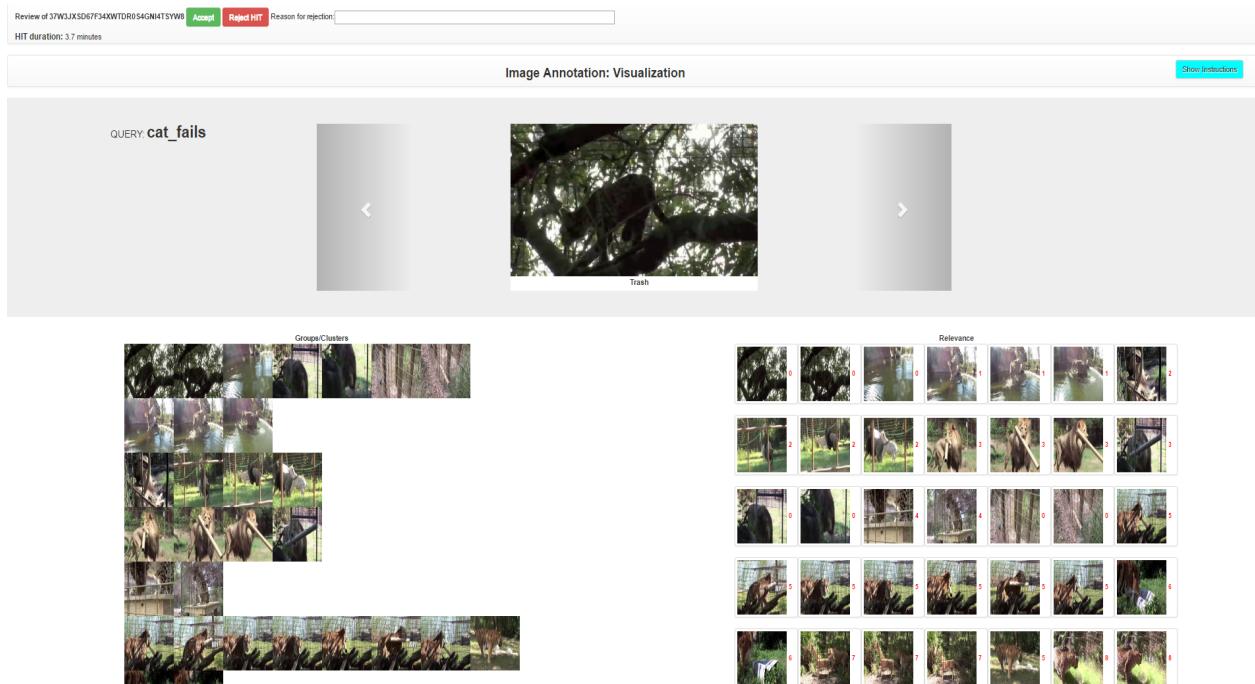


Figure 8.10: Snapshot of the Review Annotation page

We use the above review annotation page to review the first 200 HITs from which we have qualified 48 workers for further annotations. Since we allow only the qualified workers to annotate the dataset, we observe that the annotations become more reliable. For later annotations, we automate the acceptance of the HITs with few random manual checks.

The analysis of relevance and diversity annotations help to remove some particular poor annotations or a complete video with all the annotations in certain cases. The correlation analysis of relevance annotations helps to compare the annotators agreement for the videos. We observe low correlation scores(poor annotator agreement) for certain type of videos such as lyrics music videos, video captured from stationary cameras,

etc which we subsequently remove from the dataset. The analysis of clusterings using Normalized mutual information also helps in identifying trivial clusterings among the annotations. Trivial clustering can be identified when  $NMI$  become undefined for any particular clustering. We remove such annotations (trivial clusterings) from the dataset and re-upload the HIT till we get five annotations for a video. We see that  $NMI$  and correlation analysis played a major role in the removal of certain annotations or a complete video from the final dataset.

Thus, we use the correlation and clustering analysis to check and maintain the quality of our dataset annotations.

# Chapter 9

## Results and Discussion

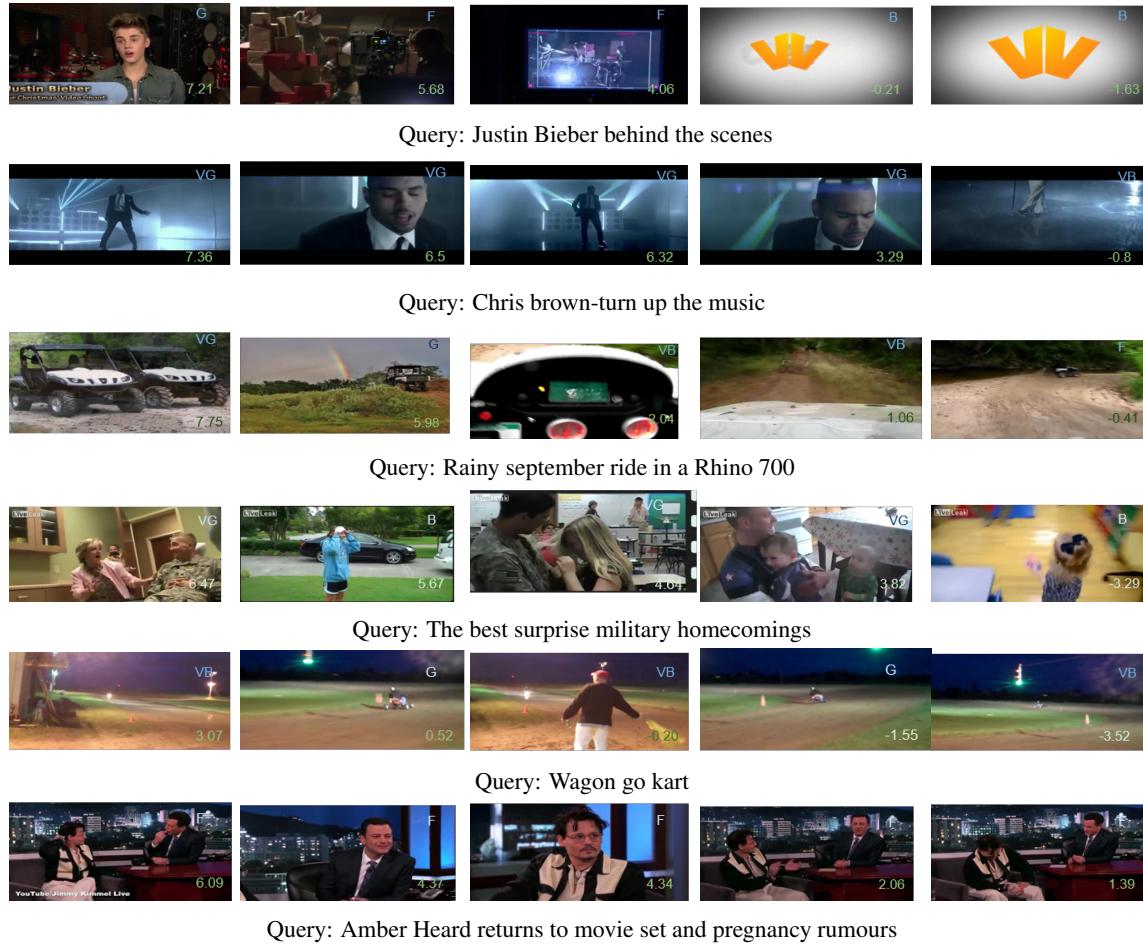


Figure 9.1: **Examples of selected thumbnails for different queries** (better viewed in colour). The groundtruth label is provided at the top right. Unnormalized score from our method is put at bottom right.

## 9.1 Relevance

Figure 9.1 shows some selected thumbnails for different queries-video pairs to visualize the results. The figure has ground truth labels at its top right corner and unnormalized score of our method at its bottom right corner. The scores are not normalized in the figure to allow us to visualize the query-thumb nail relevance over the queries. We have ranked thumbnails in the decreasing order of their scores. In the first four rows of examples, we clearly see that the thumbnails produced are semantically relevant to the corresponding query. We see that the thumbnail highly ranked for "Justin Bieber" is his image itself, for "Rhinos" - images of a tractor in fields and for "military" - soldiers with their family. Furthermore, we provide two failure examples in the last rows of Figure 9.1.

## 9.2 Diversity

In Figure 9.1, we also note that the thumbnails selected for some queries appear similar visually. In the second example of the figure, we see that the highly ranked thumbnails are quite similar. Hence, we use the submodular formulation as explained in Chapter 5 to obtain the diverse set of query relevant thumbnails. We have kept the weight vector as  $w = [1, 1]$  i.e. giving equal importance to relevance and diversity for the subset selection. Later, we show the results after learning the weights which maximizes the summarization objective.



Figure 9.2:  **Thumbnails with & without Diversification** For query:Anaconda Snake, above video thumbnails are obtained. The first row contain the relevant frames from the Visual Semantic Embedding Model while the second row represents the diverse set of query relevant frames obtained with  $w = [1, 1]$ .

In Figure 9.2, we observe that query relevant frames from the Visual Semantic Embedding Model appear relevant to the query "Anaconda Snake" but three out of five selected thumbnails look visually similar. However, in the second row, we have more diverse set of relevant frames selected. In this diversification, we see that some less relevant frames have also been ranked high, for eg, the fifth thumbnail in the second row. We predict that this may be due to the weight vector  $w$  which gives equal importance to relevance and diversity. Later, we learn the weight vector that decide the importance to be given to both relevance and diversity. This is learned using ground truth annotations obtained from the newly annotated Evaluation RAD dataset. In the subsequent sections, we present the results of relevance scores on three different dataset. We also present some visualization results of relevance and diversity on the new dataset: RAD.

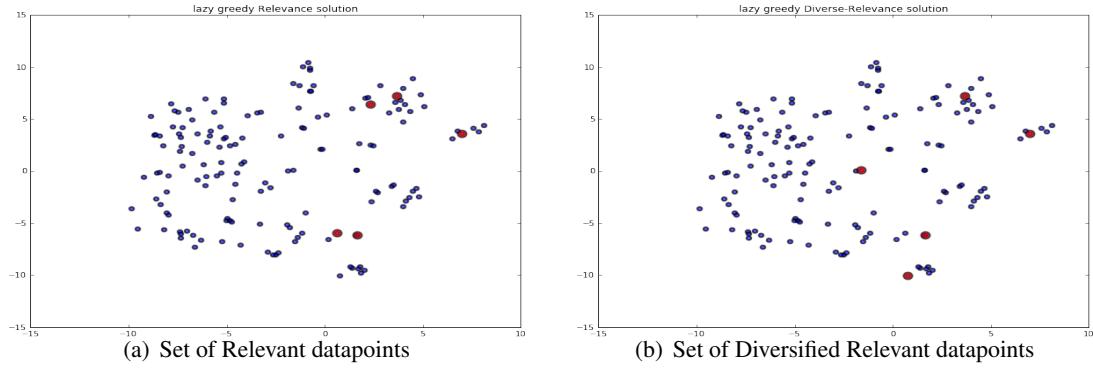


Figure 9.3: t-SNE visualization of subset selection. Red labelled as the selected points

Figure 9.3 shows t-SNE representation of fc2 features extracted from all frames of a video. The selected thumbnails are marked in red. We see a clear distinction in the diversification of the selection of thumbnails from Figure 9.3(a) and 9.3(b).

## 9.3 Performance Evaluation

### 9.3.1 MSR Dataset

Figure 9.4 depicts the Precision-Recall curve for different methods evaluated on MSR Evaluation dataset. We have already seen a lot about the MSR dataset and the methods in the previous sections. The curves in the figure clearly shows the improvement over the baseline [25].

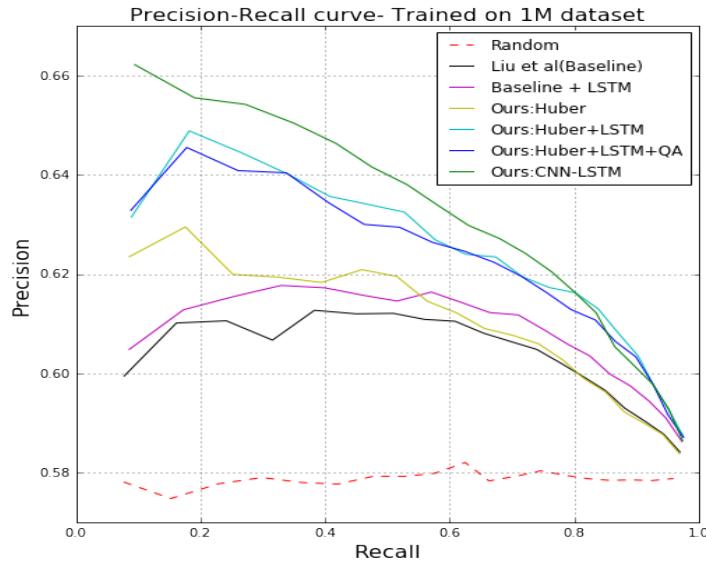


Figure 9.4: Precision-Recall curve of MSR Evaluation dataset for different methods

The following table 9.1 shows the comparison of over different methods over different metrics. We see that our method made an improvement of 2.6% over the baseline w.r.t mAP. We see that LSTM boosts the performance in the two cases: *Ours* : *Huber + LSTM* improves by 1.3% over *Ours* : *Huber* while *Ours* : *Huber + LSTM + QA* does 1.6% improvement over *Ours* : *Huber + QA*. However, observing *Baseline + LSTM* and *Ours : Huber + LSTM*, we can infer that the improved objective with triplets  $\{query, image^+, image^-\}$  provides a improvement of 1.4% on mAP. Thus, our better aligned objective function and LSTM network for query modelling played a significant role in improving the performance. We observe that query agnostic(QA) score does not improve mAP over the model without QA.

Method	HIT@1: VG	HIT@1:VGorG	Spearman Correlation	mAP
Random	$28.2 \pm 1.5$	$57.17 \pm 1.5$	-	-
Liu et al(Baseline)[25]	33.00	59.81	0.112	0.603
Baseline[25]+LSTM	32.03	60.48	0.138	0.607
Ours:L1	32.42	62.61	0.139	0.611
Ours:Huber	32.61	62.21	0.132	0.608
Ours: Huber+QA	31.83	61.81	0.149	0.603
Ours: Huber+LSTM	32.42	63.15	0.178	0.621
Ours: Huber+LSTM+QA	35.93	63.28	<b>0.183</b>	0.619
Ours: CNN-LSTM	<b>37.11</b>	<b>66.22</b>	0.179	<b>0.626</b>

Table 9.1: Comparison of the different thumbnail selection methods on MSR Evaluation dataset

### 9.3.2 MediaEval Dataset

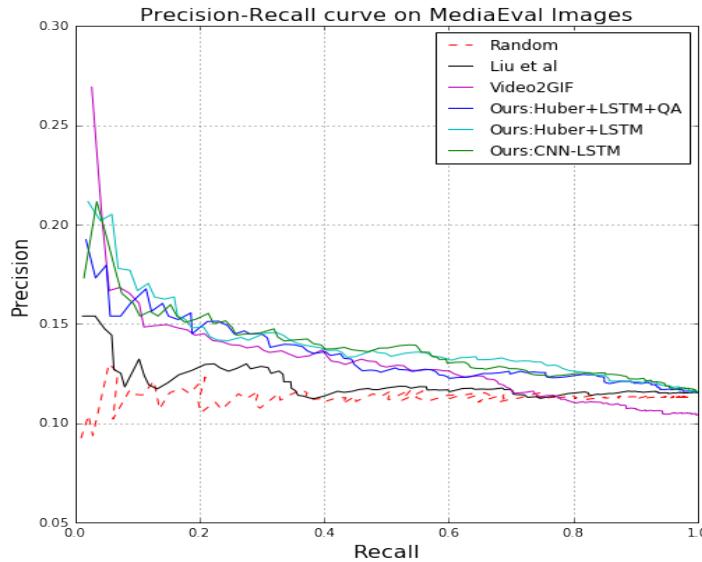


Figure 9.5: Precision-Recall curve of MediaEval dataset for different methods

This dataset is provided in the MediaEval 2016 challenge on Predicting Media Interestingness Task. The dataset consists of 52 movie trailers of Hollywood-like movies and their movie titles. The data consists of collections of keyframes extracted from the video shots which are obtained after the manual segmentation of the trailers. The extracted keyframes correspond to the frame in the middle of each video shot. This helped in comparison of results in image and video interestingness [9] as in figure 9.5.

Figure 9.5 depicts the Precision-Recall curve of different methods evaluated on MediaEval dataset. As in 9.2, our method made an improvement of 2.73% over the baseline. We also observe that [9] leads in HIT@1 metric. This may be due to the advantage of additional information for [9] as the input being a video shot than a video frame. However, *Ours : Huber + LSTM* made an improvement of 1.25% over [9] in mAP and an improvement of 0.05 over baseline in Spearman’s Rank Correlation. We use titles of the videos which are the movie titles. They are generally less related to video content. Titles provide limited help for our model to make a query relevant thumbnail selection. However, our model performs as compared to [9] in mAP and shows a improvement of 2.73% over [25].

Method	HIT@1: VG	Spearman Correlation	mAP
Random	$10.48 \pm 4.4$	-	-
Liu et al(Baseline)[25]	15.38	0.0217	0.1623
Video2GIF[9]	<b>25.0</b>	0.0672	0.1893
Ours: Huber+LSTM	21.15	0.0671	<b>0.1896</b>
Ours: Huber+LSTM+QA	19.23	0.0602	0.1811
Ours: CNN-LSTM	17.03	<b>0.0715</b>	0.1863

Table 9.2: Comparison of the thumbnail selection methods on MediaEval dataset

### 9.3.3 New Evaluation Dataset: RAD

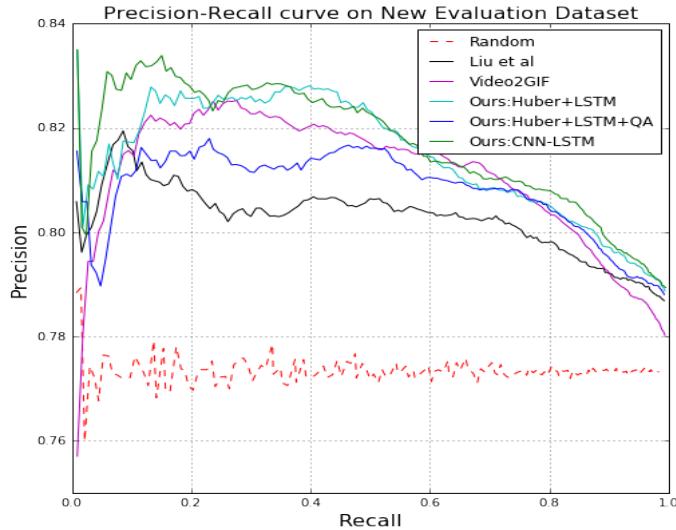


Figure 9.6: Precision-Recall curve of New Evaluation dataset for different methods

Figure 9.6 depicts the Precision-Recall curve for different methods on the newly annotated RAD dataset. Our method made an improvement of 3.9% over baseline w.r.t mAP and 0.084 w.r.t Spearman’s Rank Correlation. Here, we evaluated for [9] by extracting a video segment around the annotated frames of RAD dataset. We note that [9] is competitive in the overall with a high Spearman’s Rank Correlation but performs low in HIT@1 results.

Method	HIT@1: VG	HIT@1: VG or G	Spearman Correlation	mAP
Random	$28.06 \pm 4.5$	$77.05 \pm 3.5$	-	0.773
Liu et al(Baseline)[25]	28.12	79.61	0.112	0.80
Video2GIF[9]	28.98	74.76	<b>0.197</b>	0.806
Ours: Huber+LSTM	29.68	82.52	0.190	0.810
Ours: Huber+LSTM+QA	31.25	80.58	0.189	0.804
Ours: CNN-LSTM	<b>35.93</b>	<b>82.52</b>	0.196	<b>0.812</b>

Table 9.3: RAD dataset: Comparison of the thumbnail selection methods using queries

## 9.4 Submodular Shells

As explained in Chapter of Submodular Maximization, we use the summarization objective as a weighted linear combinations of submodular functions defined for relevance and diversity. We introduce these two submodular functions to produce diversified query relevant thumbnail selection results. To find the weights for each of the above submodular functions, we execute the grid search over the weights. Keeping the weight of relevance shell as a constant 1, we search for the weight of the diversity shell that maximizes the summarization objective.

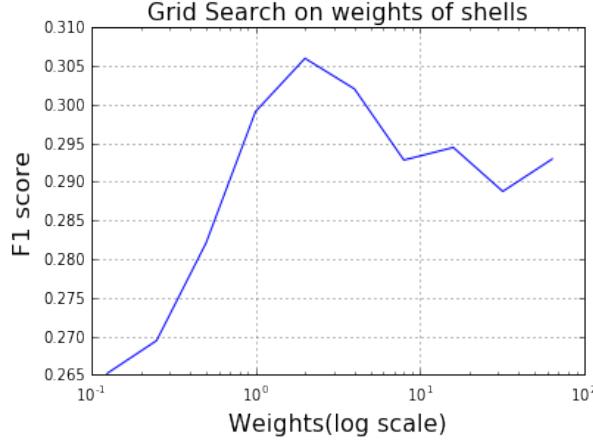


Figure 9.7: Grid Search for finding the weights of sub-modular shells

Figure 9.7 shows that the summarization objective gets maximized at the weight of 2. Hence, we use the the weight vector of  $\{f^{rel}, f^{div}\} = [1, 2]$ . Figure 9.8 shows the qualitative results for the query relevant keyframes extracted with and without diversification.



Figure 9.8: Query relevant and diversified results using our model on newly annotated RAD dataset

In Figure 9.8, for every query, the first rows represent diversified query relevant frames obtained using the weight vector  $[1, 2]$  while the second rows represent query relevant frames without diversification using the weight vector  $[1, 0]$  and the third rows represent uniformly sampled elements from the video. These keyframes are arranged in the decreasing order of the summarization objective. In figure 9.8, the frames correspond to the videos of the newly annotated dataset. For the queries 1 and 3, we can clearly observe that the first row set of keyframes not only extract relevant frames but they are diverse in nature too while the second row set of keyframes seems to be visually similar. For example, we see that all 5 images in the second row of the query 1 and last 2 images of second row in query 3 are visually similar. For the query 4, we observe that the results without diversification appear better as the video may be diverse in itself.

Some additional qualitative results on diversified query relevant thumbnail selection results are in Appendix A.

# Chapter 10

## Conclusion and Future Work

In this thesis work, given a query and a video, we propose a model based on deep networks and submodular mixtures to make a subset selection of diversified query relevant thumbnails from the video. The model comprises of a deep Visual Semantic Embedding Model that projects the query and all video frames into the Latent Semantic Embedding Space. Based on query and frame embeddings, we rank all video frames depending on their embeddings' proximity to the query embedding. Further, the model uses submodular objectives defined for query relevance and thumbnail diversity which is maximized for the overall thumbnail subset selection objective. This is jointly optimized to select frames that are both query relevant and diverse and thus yielding a query relevant video summary in the form of keyframes. Based on our results of our model, let us see some of the conclusions of my thesis:

**Deep visual semantic embedding model.** We use a supervised approach to learn LSTM networks to embed the text query and CNN networks to embed frames into the same Latent Semantic Embedding Space by joint training. The improvements on Deep visual semantic embedding model [25] using CNN-LSTM architecture and a better objective with training triplets  $\{query, image^+, image^-\}$  significantly improved our results on the extraction of query relevant thumbnails. We also observe that a query agnostic model which learns to score well-composed frames (photograph like) high, did not give a significant boost to our model performance. However, comparing our method against previous state of the art methods, we see that our model outperforms them significantly on relevance prediction.

**RAD Dataset.** For the evaluation of query relevant video summarization, there is a shortage on the availability of a dataset. MSR Evaluation dataset [25] is a close match to the problem but it lacks the queries and more importantly, ground truth thumbnails provided in the dataset are not diversified (not representative of video). Hence, we introduce a new dataset (RAD) comprising of 100 query-video pairs with query relevance annotations for all the frames and cluster groupings of the frames based on visual similarity. This dataset caters to the evaluation of selection of diversified set of query relevant thumbnails for videos. The detailed analysis of relevance and diversity annotations help to remove some particular poor annotations or a complete video with all the annotations in certain cases. The correlation analysis of relevance annotations helps to compare the annotators agreement for the videos. The analysis of clusterings using Normalized mutual information helps in identifying trivial clusterings among the annotations. Identification of trivial clusterings and correlation agreement among annotators being poor are some of the criteria that we use to remove the video annotations from the dataset. In the final dataset, the relevance and diversity annotations of our dataset are of high quality and consistent as we observe a high level of agreement between the annotators

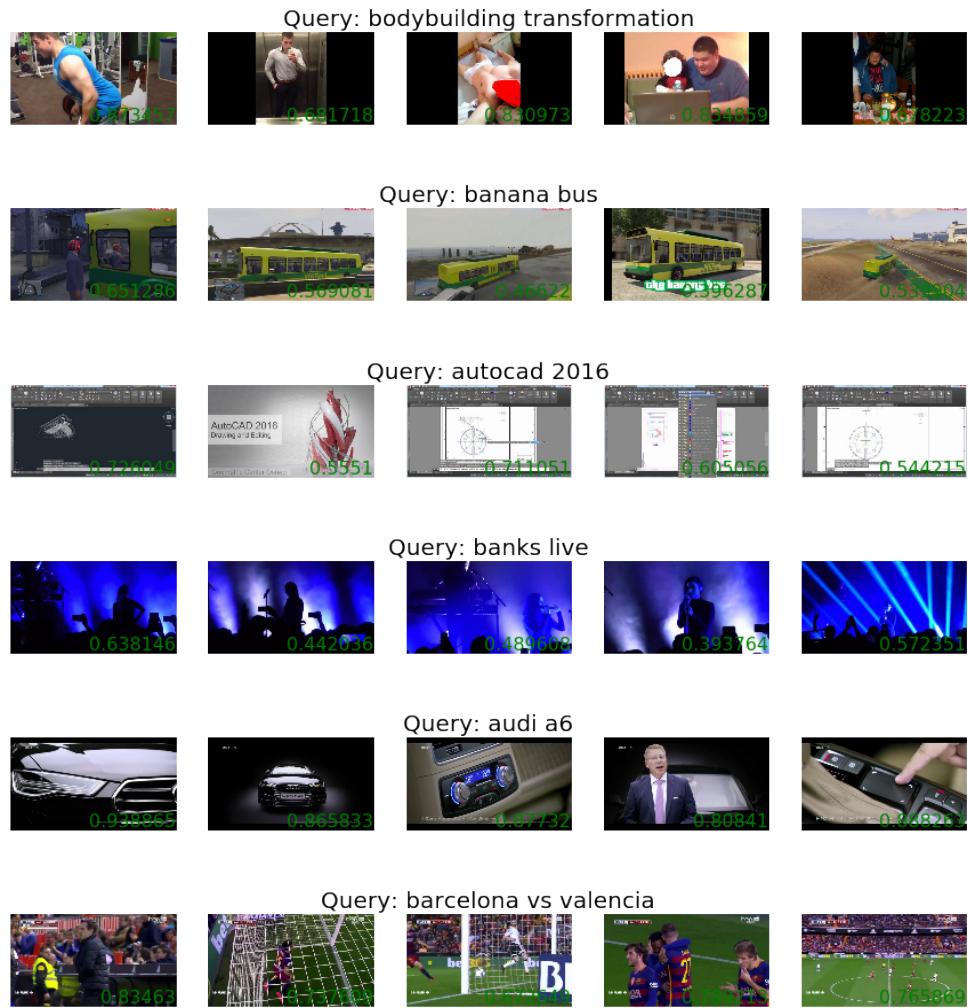
with an average correlation score of 0.69. We introduce this dataset to evaluate on the query relevant video summaries in the form of keyframes.

**Query Relevant Video Summarization.** Our problem of query relevant thumbnail extraction can be seen as query relevant video summarization. We adopt the technique of summarization using submodular mixtures from [8]. We define two separate submodular shells: a) Relevance shell which uses the deep Visual Semantic Embedding Model to score each frames depending on its text query relevance, b) Diversity shell tries to maximize the distance between the image embedding in Visual Semantic Space for which we use fc2 representation from our CNN architecture. Having learned the weights for these shells using grid search, we observe that weights of {Relevance, Diversity}=[1, 2] maximize the overall summarization objective. Thus, we generate query dependent video summaries in the form of thumbnails using the submodular mixtures and deep visual semantic embedding model. In the qualitative results, we can see clearly that diversified summaries bring out a diverse set of thumbnails that are also query relevant.

**Future Work.** Text query relevant video summarization is still an emerging problem [39]. There are a good number of less explored problems in this direction such as visual question answering from videos, query adaptive GIF creation and among others. Another interesting improvisation can be in the model architecture. Till date, there are no available end-to-end deep inference networks to our knowledge that solve this problem of query relevant video summarization. Though there are deep networks that tackle relevance, simultaneous relevance scoring and diversification of the solution set still seems to be a challenging problem for a deep network in end-to-end fashion.

## Appendix A

# Supplementary Results



## APPENDIX A. SUPPLEMENTARY RESULTS

---

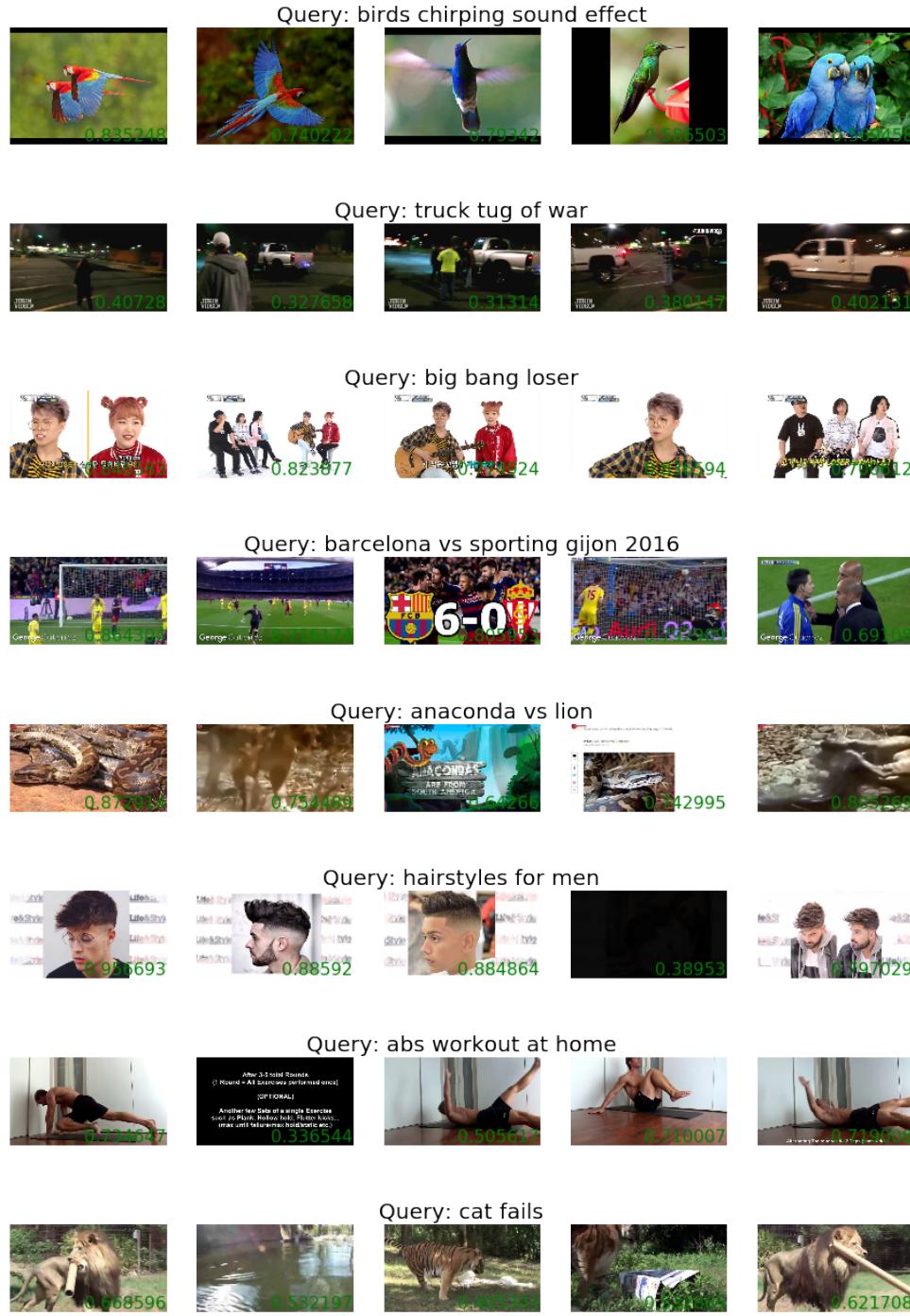


Figure A.1: Diversified Query Relevant Results. Unnormalized score from our method is put at bottom right corner (better viewed in colour)

# Bibliography

- [1] Lamberto Ballan, Marco Bertini, Alberto Del Bimbo, and Giuseppe Serra. Enriching and localizing semantic tags in internet videos. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1541–1544. ACM, 2011.
- [2] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.
- [3] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [4] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.
- [5] Joydeep Ghosh, Yong Jae Lee, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, 2012.
- [6] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pages 2069–2077, 2014.
- [7] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc Van Gool. The interestingness of images. *ICCV*, 2013.
- [8] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3090–3098, 2015.
- [9] Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. *CVPR*, 2016.
- [10] Peter J Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

## BIBLIOGRAPHY

---

- [11] Bogdan Ionescu, Anca-Livia Radu, María Menéndez, Henning Müller, Adrian Popescu, and Babak Loni. Div400: a social image retrieval result diversification dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pages 29–34. ACM, 2014.
- [12] Hong-Wen Kang and Xian-Sheng Hua. To learn representativeness of video frames. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 423–426. ACM, 2005.
- [13] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [14] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2698–2705, 2013.
- [15] Gunhee Kim, Leonid Sigal, and Eric P Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4225–4232, 2014.
- [16] Andreas Krause and Daniel Golovin. Submodular function maximization.
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [18] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization.
- [19] Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1):1–19, 2006.
- [20] Haojie Li, Lei Yi, Bin Liu, and Yi Wang. Localizing relevant frames in web videos using topic model and relevance filtering. *Machine Vision and Applications*, 25(7):1661–1670, 2014.
- [21] Yingbo Li and Bernard Merialdo. Multi-video summarization based on video-mmr. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010.
- [22] Hui Lin and Jeff 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.
- [23] Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.
- [24] Feng Liu, Yuzhen Niu, and Michael Gleicher. Using web photos for measuring video frame interestingness.

- [25] Wu Liu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3715, 2015.
- [26] Jiebo Luo, Christophe Papin, and Kathleen Costello. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(2):289–301, 2009.
- [27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [28] Marina Meilă. Comparing clusteringsan information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.
- [29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [30] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.
- [31] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.
- [32] Yingwei Pan, Ting Yao, Tao Mei, Houqiang Li, Chong-Wah Ngo, and Yong Rui. Click-through-based cross-view learning for image search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 717–726. ACM, 2014.
- [33] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.
- [34] Alex Rav-Acha, Yael Pritch, and Shmuel Peleg. Making a long video short: Dynamic video synopsis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 435–441. IEEE, 2006.
- [35] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999.
- [36] A. Sharghi, B. Gong, and M. Shah. Query-Focused Extractive Video Summarization. *ArXiv e-prints*, July 2016.
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [38] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

## BIBLIOGRAPHY

---

- [39] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TvsuM: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.
- [40] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *European conference on computer vision*, pages 787–802. Springer, 2014.
- [41] Silke Wagner and Dorothea Wagner. *Comparing clusterings: an overview*. 2007.
- [42] Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. Event driven web video summarization by tag localization and key-shot identification. *Multimedia, IEEE Transactions on*, 14(4):975–985, 2012.
- [43] Yufei Wang, Zhe Lin, Xiaohui Shen, Radomir Mech, Gavin Miller, and Garrison W Cottrell. Event-specific image importance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4810–4819, 2016.
- [44] Bo Xiong and Kristen Grauman. Detecting snap points in egocentric video with a web photo prior. In *European Conference on Computer Vision*, pages 282–298. Springer, 2014.
- [45] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [46] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. *arXiv preprint arXiv:1605.08110*, 2016.