Video Caption Generation Using Deep Learning and NLP
Final Project E6895: Advanced Big Data and AI

Sunjana Ramana Chintala[1]
UNI : sc4921
sc4921@columbia.edu
Columbia University, New York

Abstract

With the increasing volume of Multimedia operations in our everyday life, advancements in the digital space have become essential. Captioning is a natural language processing task that can revolutionize this segment. For instance, Web browsing greatly relies on finding the tags/titles of media. Image and Video Captioning would allow automatic annotation of the occurrences. This would not only facilitate the development of efficient search algorithms that search by contents of the media but would also help in the design of better recommendation systems for users. The goal of this project is to build Image Captioning and Video Captioning models. In the first part of the project, an Image Captioning Model is designed using Deep Learning and Encoder-Decoder architecture. In the second part, the scope of the model is expanded to Video Captioning. Flickr8k and TRECVID-VTT Data are used for the tasks. BLEU scores are calculated for the evaluation of the models. Greedy and Beam Search Algorithms are used for Real-Time Testing.

1. Introduction

Interactive Media has been rising to prominence in the current era. Multimedia of all forms namely picture, video, text and audio have become an essential part of everyday life. With such prevalence, the enhancement of these modes of communication is of utmost importance. Exploring various methods of improving the digital media space will ensure a better quality of life for humans.

There have been several advancements in the Image and Video Space. Work has been done on faster processing of videos, improving pixel distribution and so much more. One such avenue where much research is being conducted is Contextual Description of Images and Videos. For human beings, language is the mode of communication that allows the exchange of thoughts and ideas. When a particular incident is to be explained, language plays a vital role in the construction of our thoughts. This is the primary reason for why AI interfaces must be made aware of this skill.

When humans look at an image, the brain processes the following:

- Position of people/objects
- Actions of people
- What are the most important events in the image
- The colours
- The number etc.

After that, it derives the most important conclusion from the image. Assigning this task to an AI would require consideration of all these components. Therefore, Image Captioning is Challenging. In the case of Video Captioning, an additional challenge of considering the Spatio-temporal aspects exists. This makes Video Captioning far more challenging.

For this purpose, great research is being conducted in this space. Though demanding, Image and Video Captioning have plenty of advantages. Better Image Captioning Models would allow faster and more specific web search results. Not only this, these models can be used for developing equipment that would serve as an aid for visually challenged people. Face detection paired with Captioning would enable in the design of advanced security systems. The performance of Virtual Assistants can also be improved. Similarly, Video Captioning can help in advanced web search applications. Instead of searching for content just with tags or titles, content can be searched by Videos and their context. This would help in building better Recommendation systems that would use Video-to-text tags as their inputs.

The primary goal of this project is to build an Automatic Image Caption Generator. Once that is done, the concept is expanded to that of Deep Video Understanding and a Video Captioning Generator is designed.

2. Related Work

Image Captioning was stated as a fundamental problem in Artificial Intelligence by the benchmarking papers of Show and Tell. Image captioning connects Computer Vision and NLP. The foremost studies in the field of Image Captioning were based upon the construction of the generative model with a deep recurrent architecture. These models were quite accurate initially(3). By introducing an Attention Model, the accuracy of the models was improved. By using Attention Models, the model was automatically able to learn and fix its gaze on salient objects(2). These studies have improved in recent years, by the introduction of deep-learning-based techniques. Using CNN and now, more advanced pre-trained models, the performance and strengths have significantly improved(4).

Video Action recognition is a representative task for Deep Video Understanding. Understanding the actions that are being performed in a video is the preliminary step for video description generation. Several challenges pertain to modelling long-range videos, minimizing computational costs and evaluating valuation metrics. While videos have complex dynamics, open-domain descriptions often have temporal structure sensitivity. To address these problems GRU, LSTMs have been used in deep-video-understanding tasks. Another challenge is extracting the features from the videos(8; 7; 6). Transfer Learning models like VGG16, ResNet152, and Inception_V3 have been used for this task. These combined with LSTMs have proven to have good accuracy(1).

3. Overview

In the first part of the project, the goal is to design an Automatic Image Captioning Generator. The image captioning generator model would take an image as the input and carefully construct a generic textual description in a few words as the output.

Once this is implemented, the goal of the second part of the project is to extend the Concept of Image Captioning to Deep Video Understanding. In the next part, a Video Captioning Generator is designed. The model takes a video as the input and constructs a short textual description as the output.

4. Data

There are two kinds of datasets that have been used in this project, one being images and the other being videos. The following are the names of the two data sources:

1.) Flickr8k Dataset The Flickr8k Dataset consists of 8092 images with five captions explaining each image. In total, there are about forty thousand labelled captions. This dataset is obtained from kaggle.

Figure 1. A glimpse of the Flickr8k Data

2.) TRECVID 2022 - VTT

TRECVID 2022 provides a dataset for their Video Summarization Task. The dataset has the following characteristics:

- Short Videos (3 to 10 seconds)
- TRECVID-VTT task from 2016 to 2021
- Total of 10000 videos
- There are 6478 URLs from Twitter Vine
- The remaining from Flicker and Vimeo
- Each video has between 2 and 5 captions

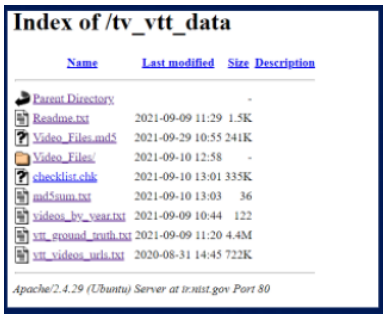

Figure 2. A glimpse of the TRECVID-VTT Data

The 3Vs of Big Data are satisfied by this datasets because of the following reasons:-

- **Volume**: The Flick8k Dataset has over 8000 videos. TRECVID dataset has more than 10000 videos. Hence there is enough volume of data.
- **Velocity**: The velocity at which images are videos are uploaded to the internet is surely high. Additionally TRECVID dataset has videos with varying lengths. Hence there is Velocity.
- **Variety**: The dataset has a collection of videos from a variety of sources. Hence there is a variety in collection.

5. Methodology

To achieve Video Caption Generation, the preliminary task of Image Captioning is important to implement. Therefore, this section is divided into two parts:
Part1 - Automatic Image Caption Generation
Part2 - Expanding the Scope of Image Captioning to Video Captioning

5.1. Automatic Image Caption Generation

The task of Image Captioning has a variety of sub-tasks within it. The overview of the system is seen in the following:

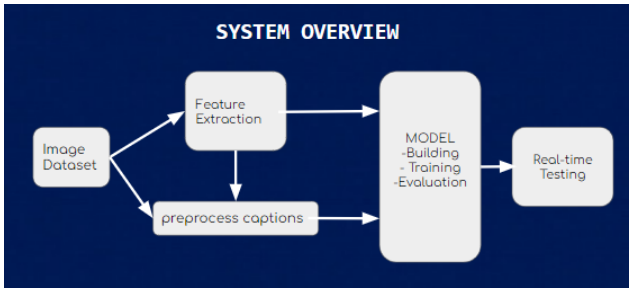

Figure 3. Overview of Methodology for Image Caption Generation

Data Preparation: As mentioned earlier, Flickr8k dataset is obtained from Kaggle. Once downloaded, the data is loaded into a dataframe where it can be seen that the images each have a Unique ID. Associated with this ID is a set of five captions. To ensure smooth processing, each Image ID is grouped with its captions and stored in the form of an array.

5.1.1. Feature Extraction:

To ensure that the system understands an image, it would have to be converted to a machine-readable format. For this purpose, feature extraction is crucial. Feature extraction, as its name suggests, converts an image into a stack of numbers. These numbers become are essentially the pixel distribution within the image.

The process of feature extraction for images has had several developments in the past. Transfer Learning is one method that has great significance in this space. Keras has a set of state-of-the-art deep learning models with pre-trained weights on ImageNet. VGG16, a powerful machine learning model, is used for this particular task. The advantage of using VGG16 is that it has a simple yet powerful architecture that is suitable for benchmarking.

5.1.2. Captions Pre-processing:

There are five captions for each Image ID. These captions are grouped with the IDs in the Data Preparation step. In the pre-processing section the following operations are performed on the data.

- Tokenizing the text
- Converting to lowercase
- Removing punctuation from each token
- Removing tokens with numbers in them
- Storing the data as a string
- Convert into Vocabulary

5.1.3. Model

The first part of this section is to build the model. To achieve this, the model is added with three layers namely, feature extractor, sequence model and decoder model. The final model looks like the following:

Model: "model_13"

Layer (type)	Output Shape	Param #	Connected to
input_17 (InputLayer)	[(None, 33)]	0	[]
input_16 (InputLayer)	[(None, 4096)]	0	[]
embedding_3 (Embedding)	(None, 33, 256)	1921792	['input_17[0][0]']
dropout_6 (Dropout)	(None, 4096)	0	['input_16[0][0]']
dropout_7 (Dropout)	(None, 33, 256)	0	['embedding_3[0][0]']
dense_9 (Dense)	(None, 256)	1048832	['dropout_6[0][0]']
lstm_3 (LSTM)	(None, 256)	525312	['dropout_7[0][0]']
add_3 (Add)	(None, 256)	0	['dense_9[0][0]', 'lstm_3[0][0]']
dense_10 (Dense)	(None, 256)	65792	['add_3[0][0]']
dense_11 (Dense)	(None, 7507)	1929299	['dense_10[0][0]']

Total params: 5,491,027
Trainable params: 5,491,027

Figure 4. The Final Model for Image Captioning

Once the model has been set-up, it is time to train the model with the Flickr8k Data. Since all the data cannot be loaded at once, a data generator is used for the task. The model is fitted. The parameter for the model with the most minimal loss is selected to be the final model.

After training, the model is tested for a thousand images. To evaluate the model, the BLEU score is calculated. The results can be seen below. The scores are quite good for the constructed model.

```
Dataset: 6000
Descriptions: train=6000
Vocabulary Size: 7507
Description Length: 33
Dataset: 1000
Descriptions: test=1000
Photos: test=1000
```

Figure 5. Image Captioning Model Evaluation

To improve the output of the model, an attention block is added to the system. The advantage of adding an attention block allows the model to focus on only the most important parts in a system and ignore the rest. Therefore, the generated output of the Visual Attention Model only takes into account, the relevant parts of an image.

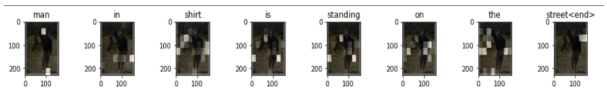

Figure 6. Image Captioning Model with Attention

5.1.4. Real-Time Testing

As the model is now ready, real-time testing can be done. For this, an image from the internet is selected and it is given as the input to the real-time model. The output of the final model is given as follows:

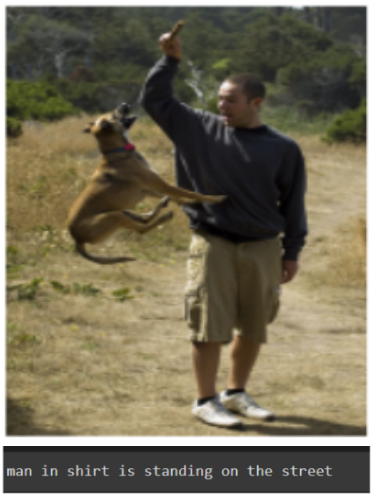

Figure 7. Real-Time Image

5.2. Video Captioning

The second part of the study is to expand the scope of Image Captioning to Video Captioning.A Video is a sequence of images. Therefore, it is only obvious that using the video frames as image inputs to the Image Captioning Model should serve the purpose.

The following figure gives the System Overview for a Video Captioning Model.

TRECVID 2022 Data is used for this section. The "vtt_videos_urls.txt" data is used for Training and "Video.Files/xxxx.webmd" is used for testing. Once the

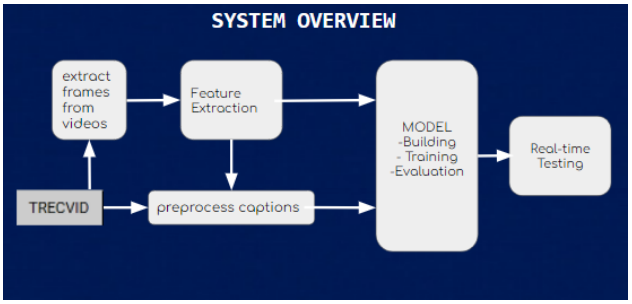

Figure 8. Overview of Methodology for Video Caption Generation

data is prepared, a series of four steps are followed. Though latter steps are very similar to those followed in Image Captioning, it is the primary step that draws the distinction. Due to this change, there are subtle modifications that appear and this effects the latter steps. Therefore, all of the steps will be explained in detail, in order to ensure comprehensive detailing.

5.2.1. Video Frame Extraction

Training data consists of over six thousand urls. The length of each video in the url is different. Therefore, a constant number of frames is necessary to facilitate analysis. Generally a video has about 24 frames in a second. Since the videos are 3 to 10 seconds long, it would be wise to choose a small yet accommodating number. A constant number of 80 frames is chose from each video. A glimpse at the frames of the first video is given below:

Figure 9. Video Frames after Extraction

5.2.2. Feature Extraction

Similar to Image Captioning, pre-trained model are used for feature extraction. In addition to VGG16, InceptionResnetV2 is also used for feature extraction. The features were stored as numpy arrays of shape (80,4096) and (80,1536) respectively.The final softmax layer is removed for both of the models.

```
model_final.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
flatten (Flatten)            (None, 25088)             0
fc1 (Dense)                  (None, 4096)              102764544
fc2 (Dense)                  (None, 4096)              16781312
=================================================================
Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0
_________________________________________________________________
```

Figure 10. VGG16 Model for Video Feature Extraction

```
model = InceptionResNetV2(weights="imagenet", include_top=True, input_shape=(299, 299, 3))
out = model.layers[-2].output
model_final = Model(inputs=model.input, outputs=out)
model_final.summary()
```

```
block8_10 (Lambda)              (None, 8, 8, 2080)   0           ['block8_9_ac[0][0]',
                                                                  'block8_10_conv[0][0]']
conv_7b (Conv2D)                (None, 8, 8, 1536)   3194880     ['block8_10[0][0]']
conv_7b_bn (BatchNormalization  (None, 8, 8, 1536)   4608        ['conv_7b[0][0]']
)
conv_7b_ac (Activation)         (None, 8, 8, 1536)   0           ['conv_7b_bn[0][0]']
avg_pool (GlobalAveragePooling  (None, 1536)         0           ['conv_7b_ac[0][0]']
2D)
==================================================================================================
Total params: 54,336,736
Trainable params: 54,276,192
Non-trainable params: 60,544
__________________________________________________________________________________________________
```

Figure 11. InceptionResnetV2 Model for Video Feature Extraction

5.2.3. Captions Cleaning and Pre-Processing

The Maximum length of the captions is about 37 words and the minimum length is about 3. For this reason, when the model is trained the way it is, the problem of excessive padding arises. To elaborate, the model predicts blank spaces and this would lead to uneven textual predictions. To avoid this problem, a fixed range of 6-10 words is chosen as the input to the model. By limiting the lengths of the input, consistency can be achieved. '<start>' and '<end>' tags are added at the start and end of the captions.In addition to this, the captions are paired with the captions.

5.2.4. Model

An Encoder-Decoder Model is used for Training. Since a video is a sequence of images and the temporal nature of the frames is to be considered, using a RNN architecture would be the best choice. Therefore, LSTMs are used in the Encoder-Decoder architecture.

The following figure shows the final model used for VGG16 analysis:

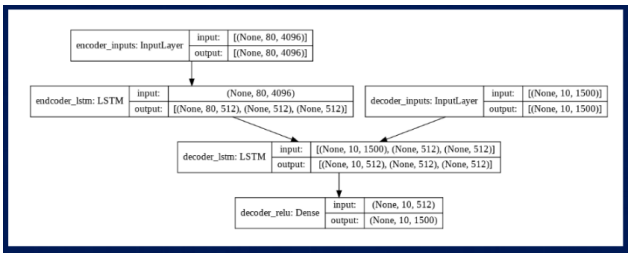

Figure 12. Encoder-Decoder Model Architecture

The model is trained for over 150 epochs. With early stopping, the model trains for about 50 epochs. The loss and accuracy plots are generated.Post that, testing data is given as input to the model. For this part "Greedy Search" and "Beam Search" Decoding methods are used to get the results, The difference between the two is that Greedy Decoding calculates the best option based on the next word token whereas beam search queries for multiple tokens by taking the future into consideration. The BLEU score for testing data for both VGG16 and InceptionResnetV2 are calculated. It can be seen that the BLEU score for VGG16 is better than the more advanced InceptionResnetV2 Model.BLEU followed by the numbers 1, 2, 3 and 4 signify the BLEU values for 1gram, 2-gram, 3-gram and 4-gram values.

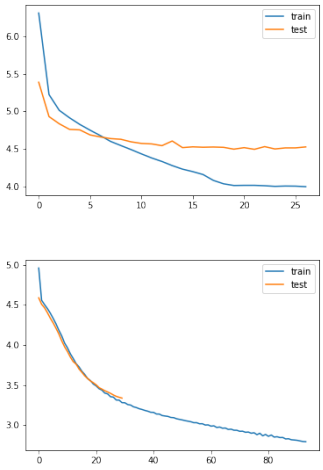

Figure 13. Loss Plot for VGG16 and InceptionResnetV2

```
BLEU-1: 0.137363
BLEU-2: 0.370625
BLEU-3: 0.551265
BLEU-4: 0.608790
```

Figure 14. BLEU score for Video Captioning Model VGG16

```
BLEU-1: 0.071918
BLEU-2: 0.268175
BLEU-3: 0.453995
BLEU-4: 0.517856
```

Figure 15. BLEU score for Video Captioning Model Inception-ResnetV2

5.2.5. Real-Time Testing

A random pick of about 24 video urls is chosen from the data. Real-time testing results are generated for the VGG16 features as this is the better performing model. It can be seen that VGG16 paired with LSTM Encoder-Decoder gives fairly accurate results.

6. Results

Through careful re-construction, a Video Captioning Model was built on top of the Image Captioning framework with the VGG16 architecture. The successful reproduction of a few of the test images is given as follows:

From the results, it can be seen that the outputs closely align with the meaning of the actual captions. Even though the length of the sentences are smaller than the actual captions, the captions tend to capture the essence of the videos.

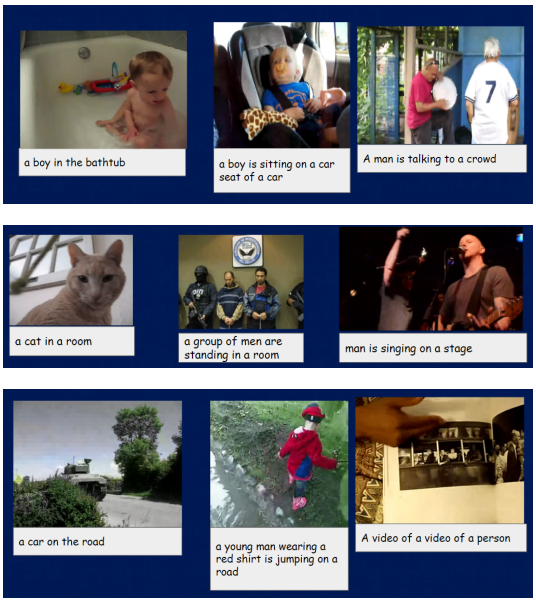

Figure 16. Sample Results for Video Captioning

ID	Actual	Predicted
16	Tan cat bats and bites at cord held by fingers	a cat in a room
38	Several men stand by a wall	a group of men are standing in a room
295	Toddler in bathtub plays with toys	a man in a bathtub
297	three dolls moving side to side while music is playing in an indoor room	a group of women dancing on a stage
302	Two men dressed in black talk in front of a yelling audience	a man is dancing on a stage
307	Two orange cats wrestle on a light tan couch	a dog is sleeping on a couch
309	A group of three guys are sitting at a dining table talking in a room during the daytime	two men are talking in a room
384	a man is walking down a street	a man is walking down a street
397	A bus is stalled blocking 2 lanes of traffic on a snowy winter night	a young man is on a tree on a street
408	Baby rides in car seat in car with a pacifier in its mouth	a boy sitting on a car seat of a car
411	Indoors, a woman then a man tap dance on small raised area on wooden floor	two men are dancing on a stage
462	A person in a yellow life preserver tries to swim over to a small boat.	a person in a water in the daytime
539	On a stage, two men with guitars and one in skeleton mask and pirate hat holding sparklers	a man is singing on a stage
572	a biker is cycling on the road, at day time	a man is riding in a car
573	A small child in pink boots and a red jacket plays in a puddle then carries a stick	a young man wearing a red shirt is jumping on a road
589	A man guiding a plow pulled by an ox walks through a muddy hillside field	a man is running on a field
599	Person watching items suspended by wire being raised and lowered	a young man in a kitchen
606	Someone showing photos in a picture album	a video of a video of a person
676	Toddler walking in a room is holding onto a chair and falls down	a toddler is dancing in a bed
694	An explosion shakes a building during the night	a crowd of people in a night
692	a group of people are dancing in a street	a group of people are dancing in a street
745	An US Army tank riding down the street with trees on either side on a sunny day	a car on the road
756	Two female Park Rangers talk into the camera on a cloudy day	a man is talking to a crowd
799	A young woman sings along with a song, then dances to the same music in a room	a woman is talking to the camera in a room

Figure 17. Actual v/s Predicted Captions for a Sample of Outputs

7. Conclusion

Multimedia formats such as Images and Videos have become an integral part of everyday life. Image and Video Captioning would allow the construction of better search algorithms and can prove to be extremely beneficial for a variety of reasons. In this project, both Image and Video Captioning models are built. VGG16 and InceptionResnetV2 are used for feature extraction. Encoder-Decoder Architectures with LSTM layers are used for sequential training of the models. The results show that the model with features extracted from VGG16 tends to have better BLEU scores when compared to InceptionResnetV2 for the TRECVID-VTT data. The textual description for about

a thousand videos is obtained in the testing phase. Real-time testing is done for both Image and Video Captioning Models.

8. Future Scope

The study can be extended to perform captioning with other Pre-Trained image captioning models. The scope can be extended to multi-modalities like Audio etc.

Accessibility

https://github.com/Sapphirine/video_caption_generation.

Software and Data

All code was written in python.

References

[1] A. Jeyanthi Suresh, J. Visumathi, Inception ResNet deep transfer learning model for human action recognition using LSTM, Materials Today: Proceedings, 2020, , ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2020.09.609.

[2] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

[3] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.. (2014). Show and Tell: A Neural Image Caption Generator.

[4] Lu, Y., Zhu, S.C., Wu, Y.. (2015). Learning FRAME Models Using CNN Filters.

[5] Hossain, M., Sohel, F., Shiratuddin, M., Laga, H.. (2018). A Comprehensive Survey of Deep Learning for Image Captioning.

[6] Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., Zhang, Z., Tighe, J., Manmatha, R., Li, M.. (2020). A Comprehensive Study of Deep Video Action Recognition.

[7] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.. (2015). Sequence to Sequence – Video to Text.

[8] Tanti, M., Gatt, A., Camilleri, K.. (2017). What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?.