Video Summarization

Aseem Kannal 71700718K
Atharva Bhagwat 71700740F
Sameer Kolhar 71701180B
Purva Sheth 71701133L

Guide: Prof. A. A. Chandorkar

Need & Motivation

- Large amount of videos are uploaded daily, making it difficult to learn important information without spending much time.
- This can be solved by a system which shortens videos keeping the vital information for the user.

Problem Statement

For a given video file, the goal is to generate a video file that contains the summarised information of the original video file. The resultant video file is to be produced based on the factors such as context, length, activity and relevance to the subject of the video enclosed as a model. Also, use the aforementioned model in an application environment and produce a final application.

Hardware Requirements

- NVidia GTX 1080 or higher GPU
- Amazon Simple Storage Service (S3) instance
- Amazon Elastic Compute Cloud (EC2) instance

Software Requirements

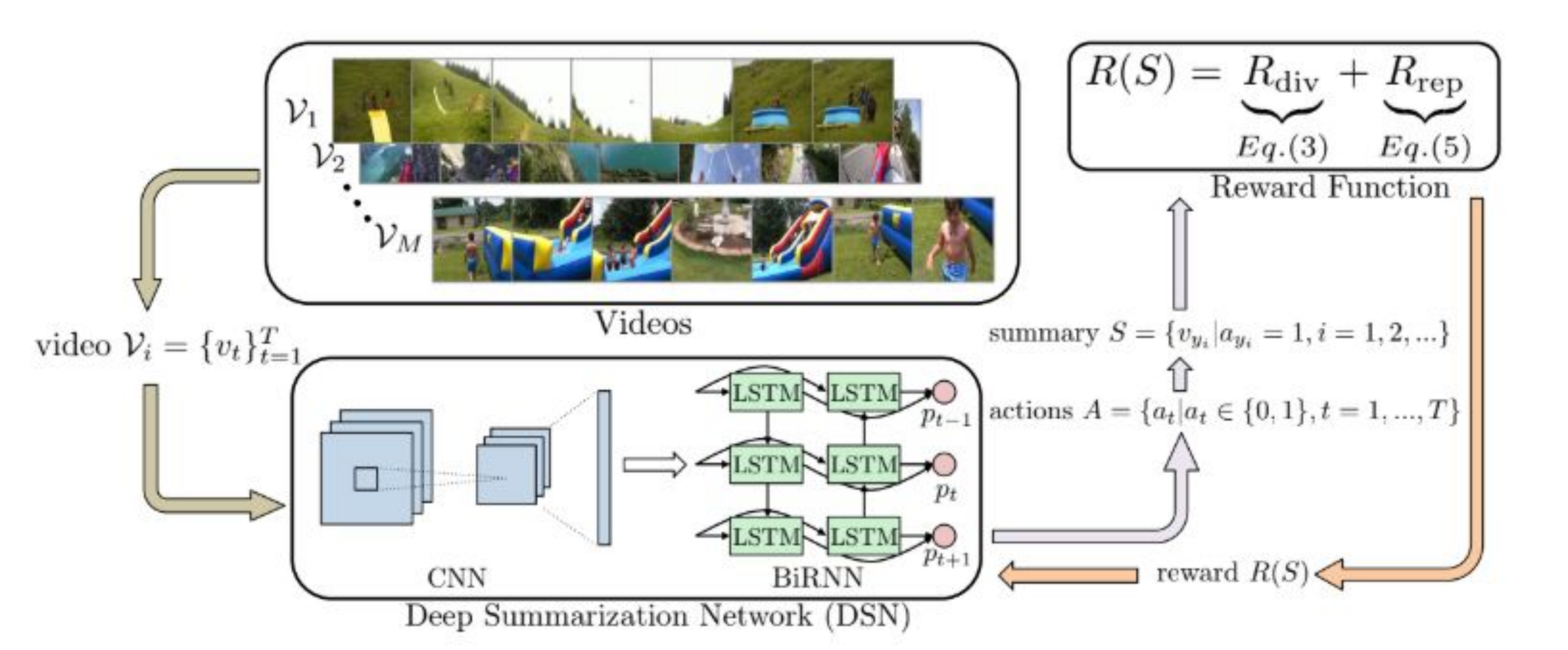
- PyTorch
- OAuth 2.0
- React Web Framework
- Redux JavaScript Library
- npm - package manager
- Compute Unified Device Architecture (CUDA) Toolkit 9.0 or higher

Literature Survey

Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward

Published at the 32nd AAAI Conference on Artificial Intelligence, 2017. This paper uses a sequential decision making process to develop a Deep Summarization Network (DSN).

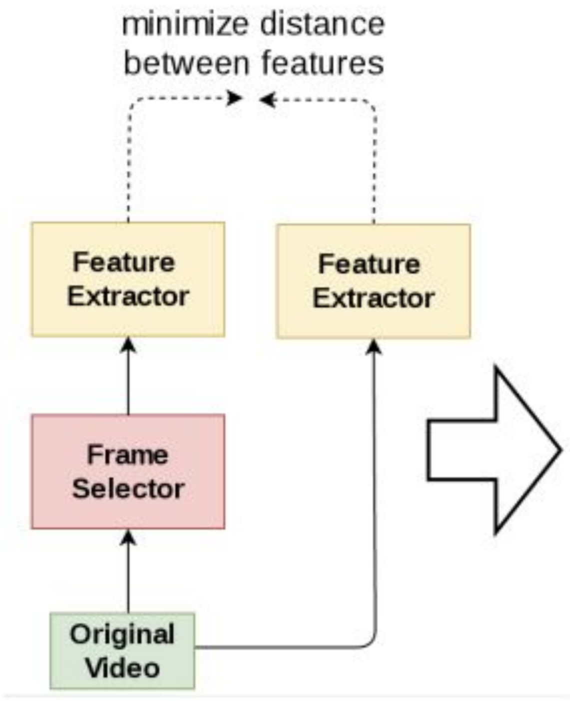
For each video frame DSN predicts a probability which indicates how likely a frame is selected, and then takes actions based on the probability distributions to select frames, forming video summaries.


$R(S) = \underbrace{R_{\text{div}}}_{Eq.(3)} + \underbrace{R_{\text{rep}}}_{Eq.(5)}$
Reward Function
$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_M$
Videos
video $\mathcal{V}_i = \{v_t\}_{t=1}^T$
summary $S = \{v_{y_i} | a_{y_i} = 1, i = 1, 2, ...\}$
actions $A = \{a_t | a_t \in \{0, 1\}, t = 1, ..., T\}$
LSTM
p_{t-1}
p_t
p_{t+1}
CNN
BiRNN
reward $R(S)$
Deep Summarization Network (DSN)

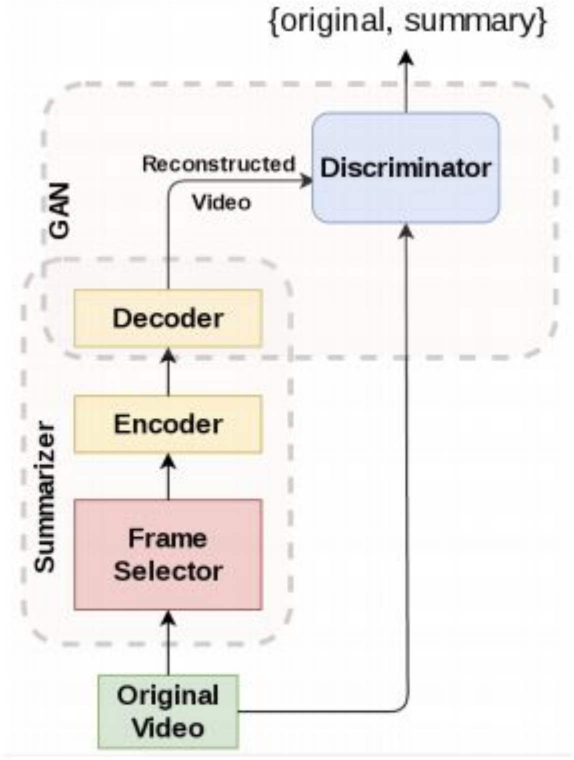
Unsupervised Video Summarization with Adversarial LSTM Networks

This paper addresses the problem of selecting sparse subset of video frames that optimally represent the input video. It proposes a unsupervised approach using generative adversarial framework consisting of the summarizer and discriminator for learning.

The summarizer is a LSTM network aimed at selection of video frames. The discriminator is another LSTM network aimed at distinguishing between the original video and its reconstruction from the summarizer.


minimize distance
between features
Feature Extractor
Feature Extractor
Frame Selector
Original Video

(a)

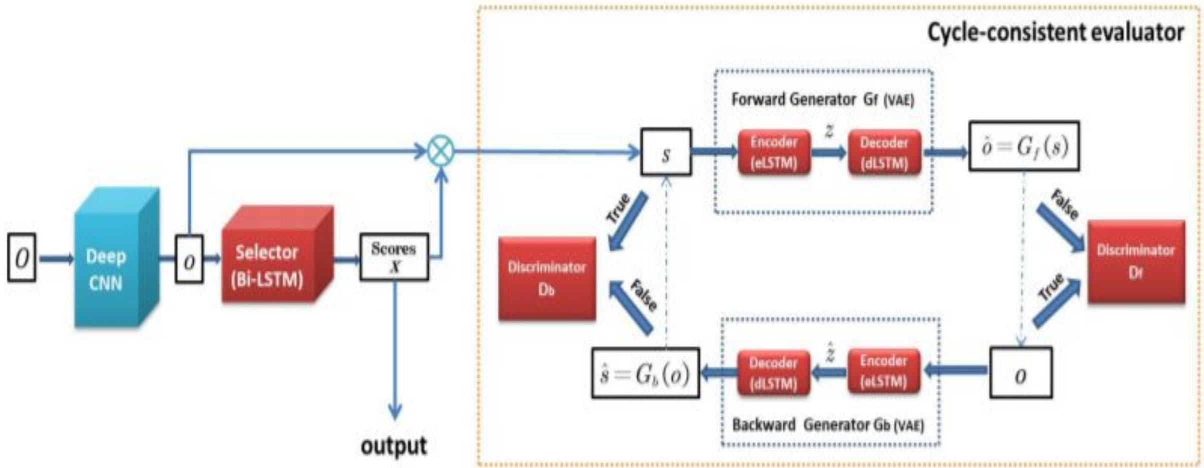

{original, summary}
GAN
Reconstructed Video
Discriminator
Decoder
Encoder
Summarizer
Frame Selector
Original Video

(b)

Cycle-SUM: Cycle-consistent Adversarial LSTM Networks for Unsupervised Video Summarization

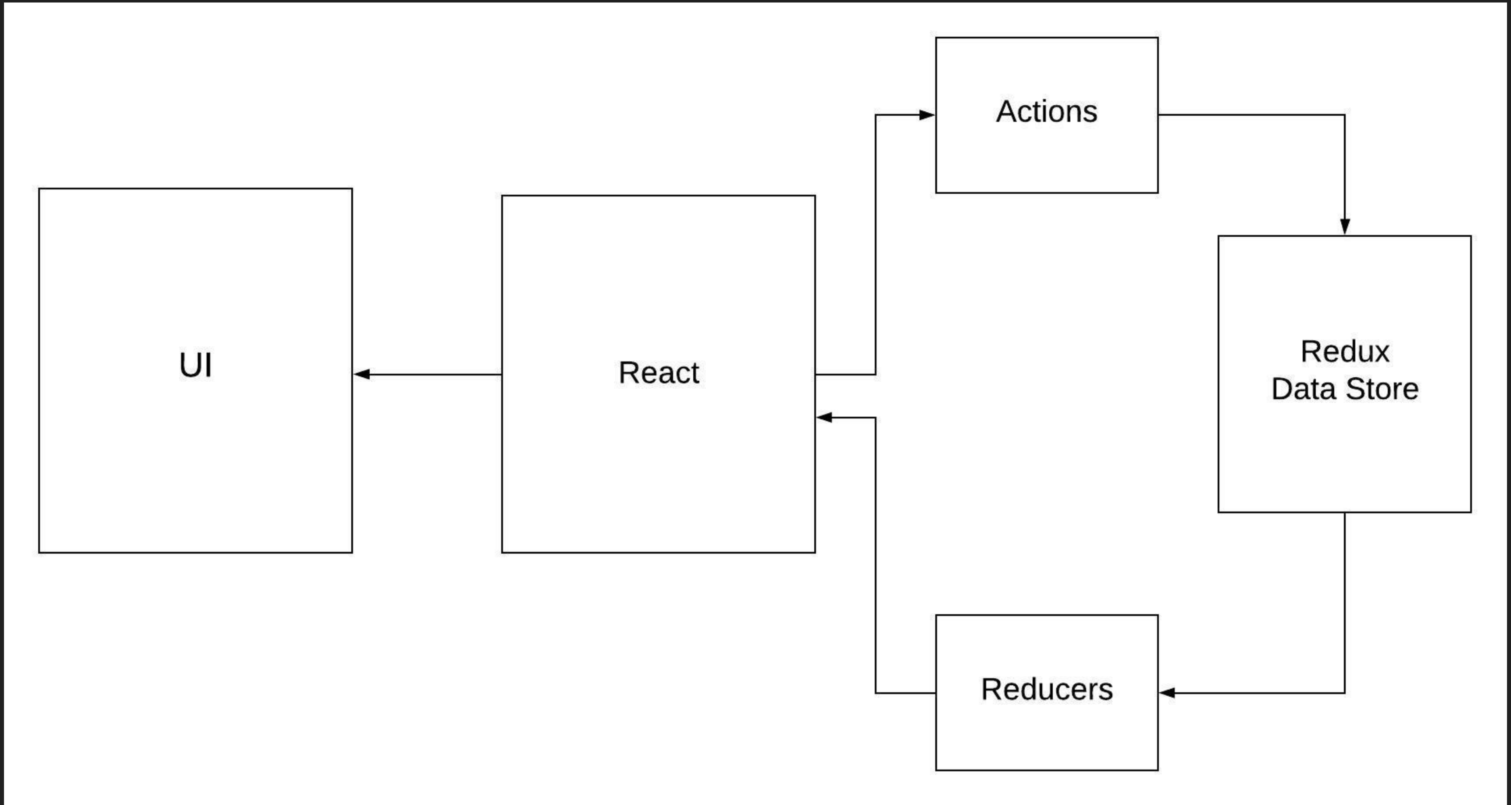
This paper proposes a model termed Cycle-SUM, which consists of a frame selector and a cycle consistent learning-based evaluator. The selector is a bidirectional LSTM network that learns the video representations that embed the long-range relationships among the video frames.

The evaluator consists of two Generative Adversarial Networks (GANs), in which the forward Generative Adversarial Network (GAN) is learned to reconstruct original video from summary video, while the backward GAN learns to invert the processing.

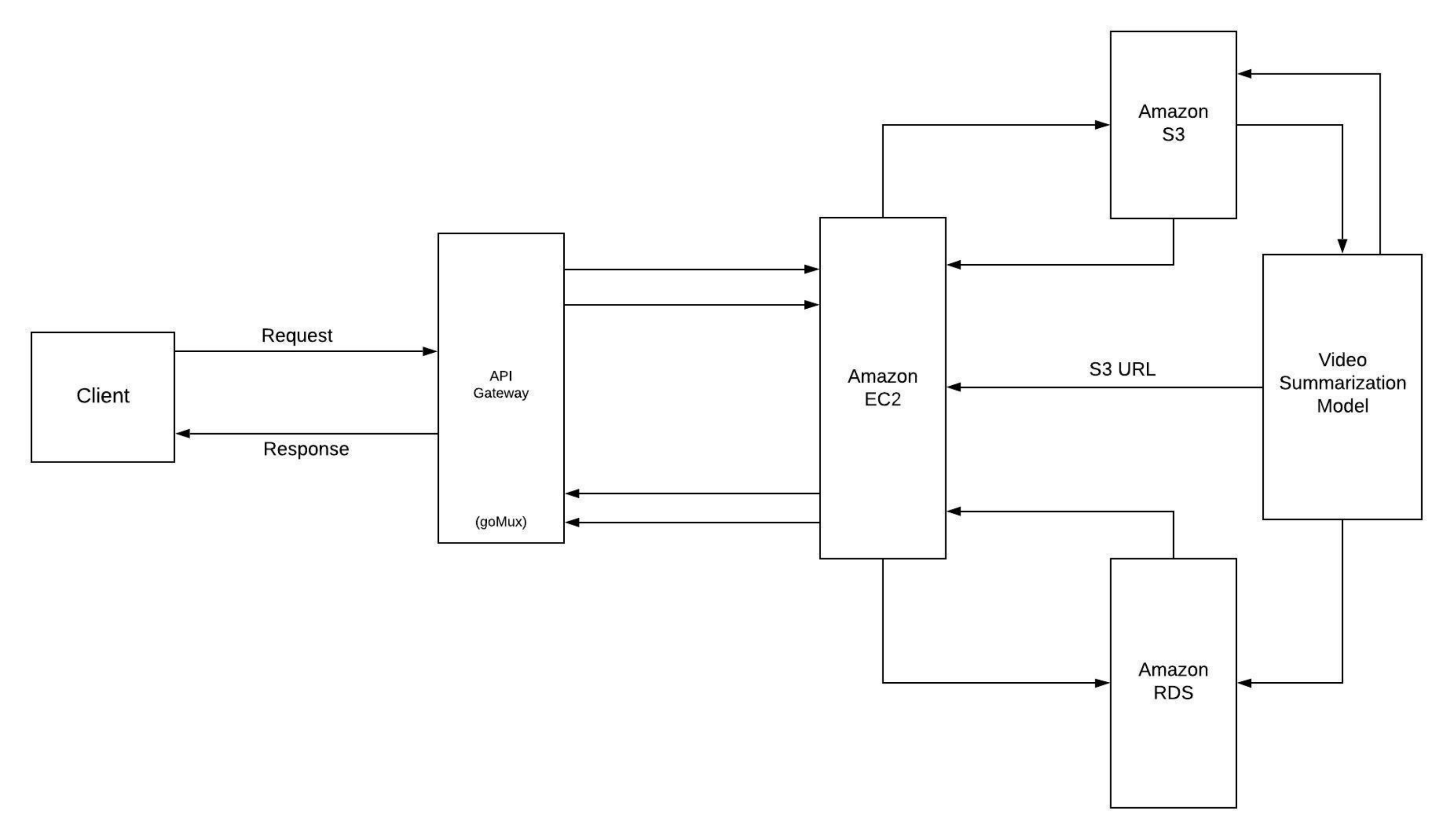


Cycle-consistent evaluator
Forward Generator Gf (VAE)
Encoder (eLSTM)
Decoder (dLSTM)
z
s
$\tilde{o} = G_f(s)$
True
False
Discriminator Db
Discriminator Df
O
Deep CNN
o
Selector (Bi-LSTM)
Scores X
output
$\hat{s} = G_b(o)$
Decoder (dLSTM)
Encoder (eLSTM)
$\hat{z}$
o
Backward Generator Gb (VAE)

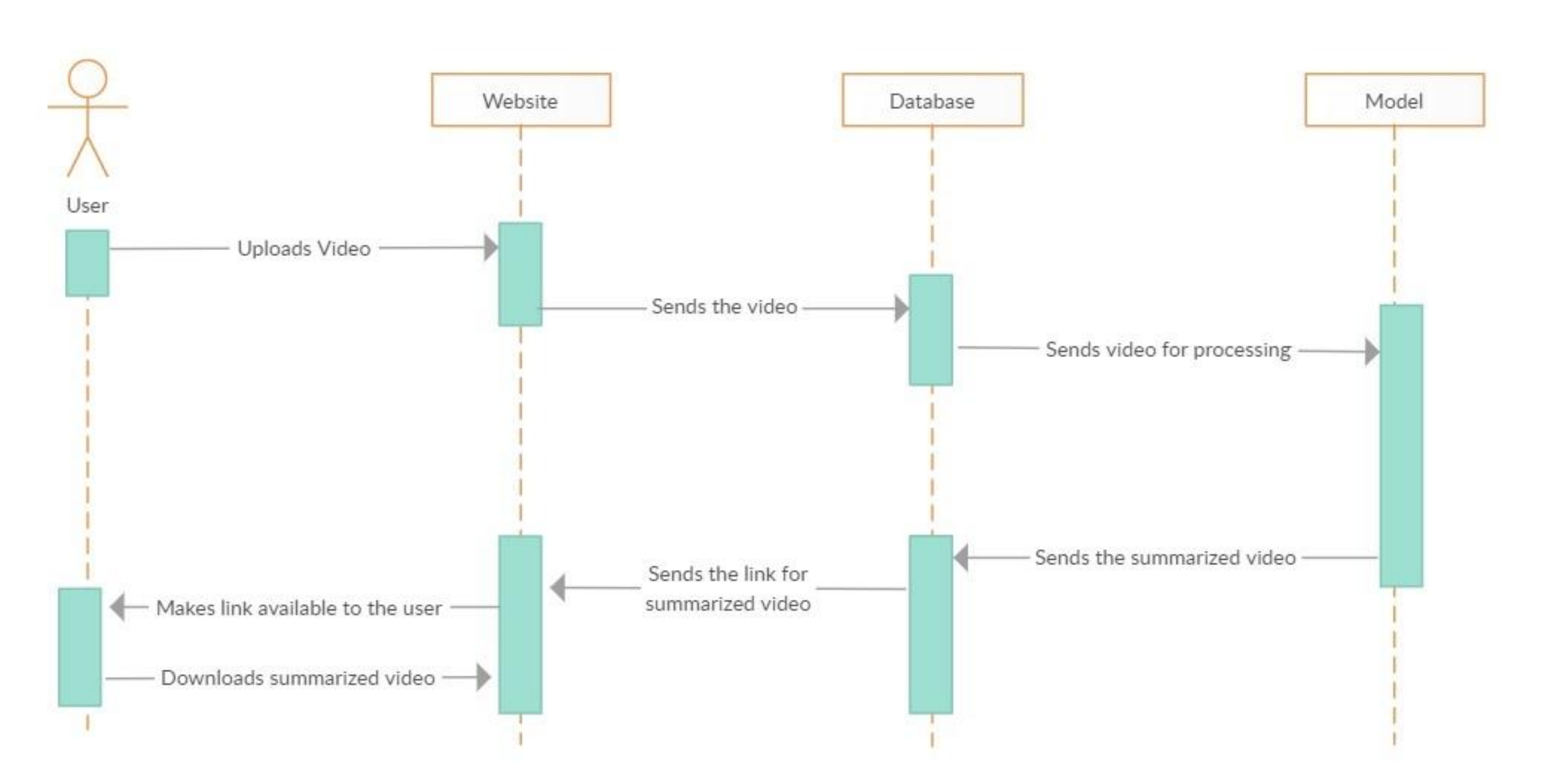
Front End Architecture Diagram



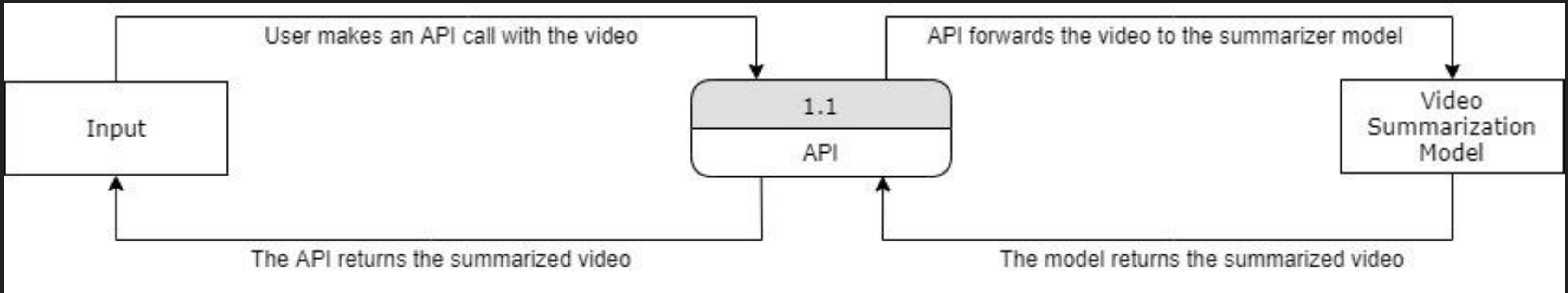
Back End Architecture Diagram



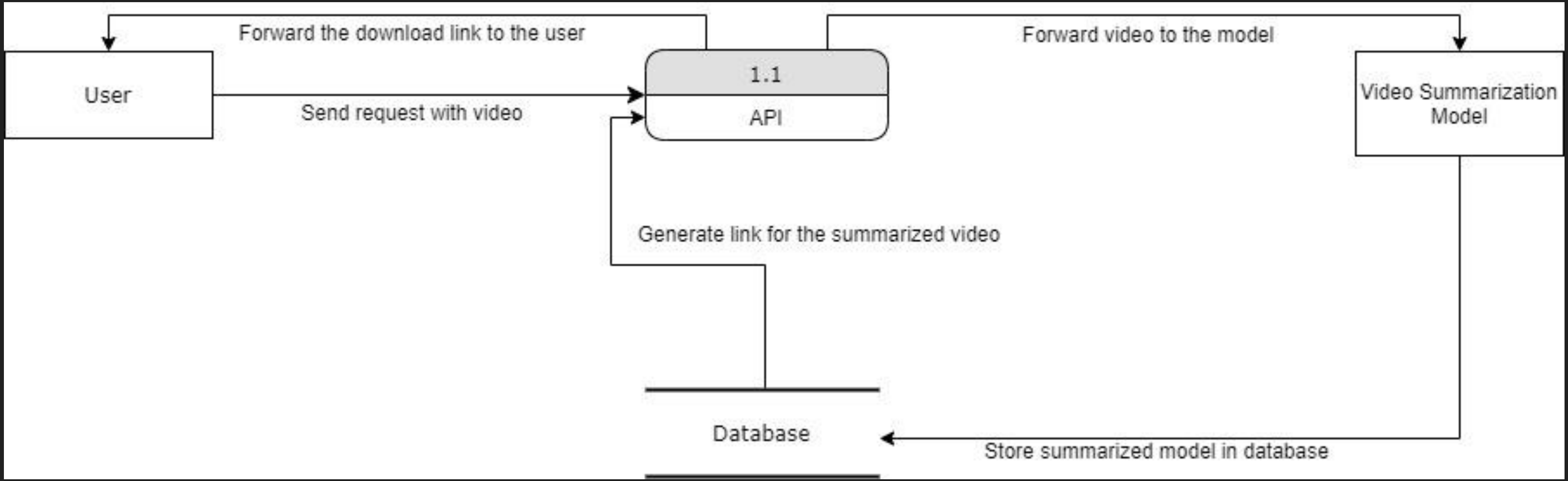
Sequence Diagram



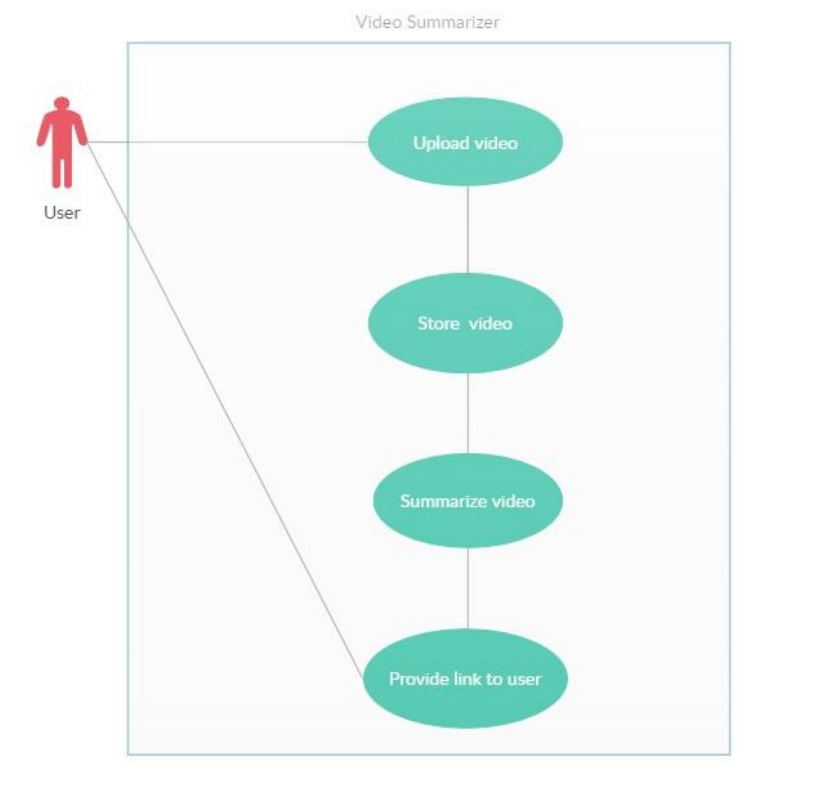
Data Flow Diagram: Level 0



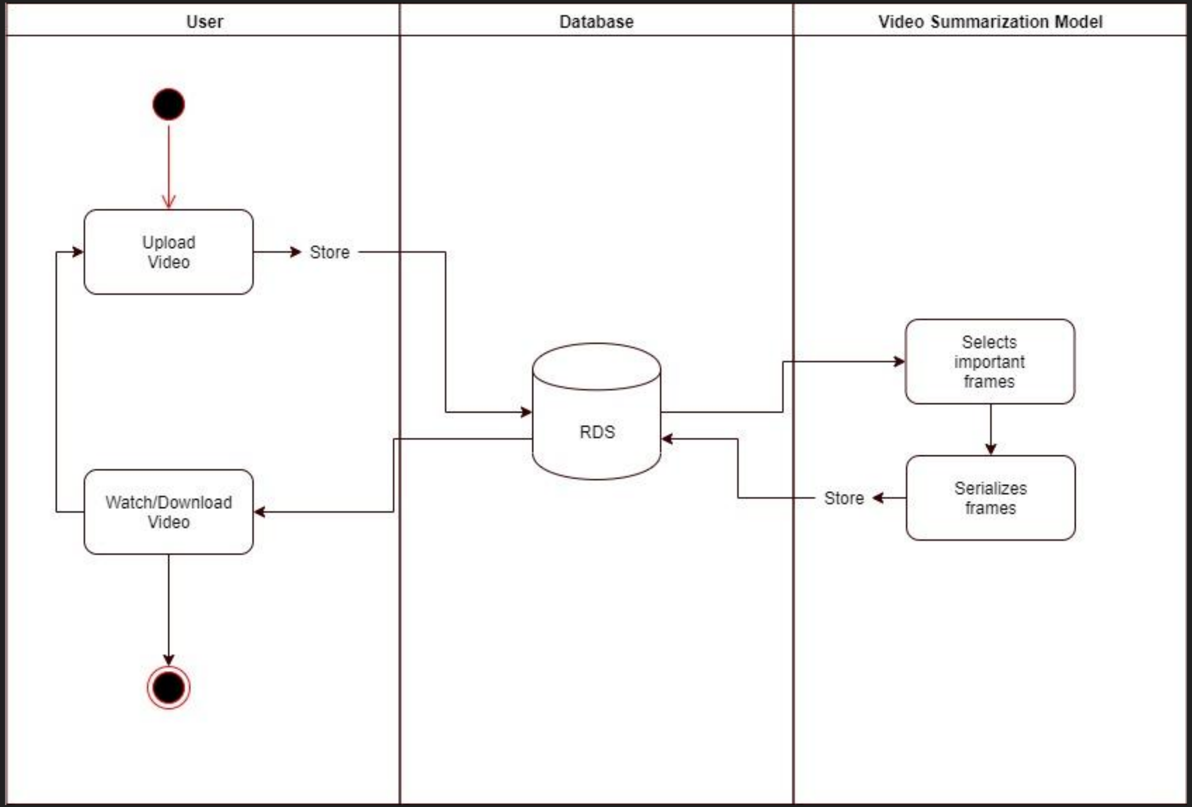
Data Flow Diagram: Level 1



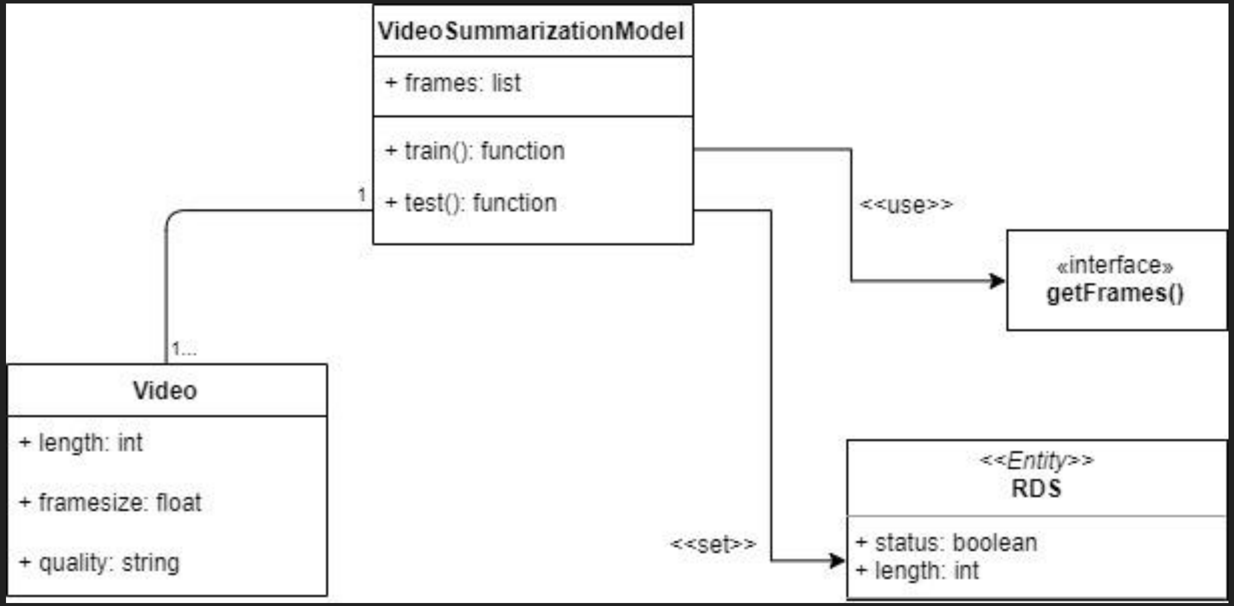
Use Case Diagram



Activity Diagram

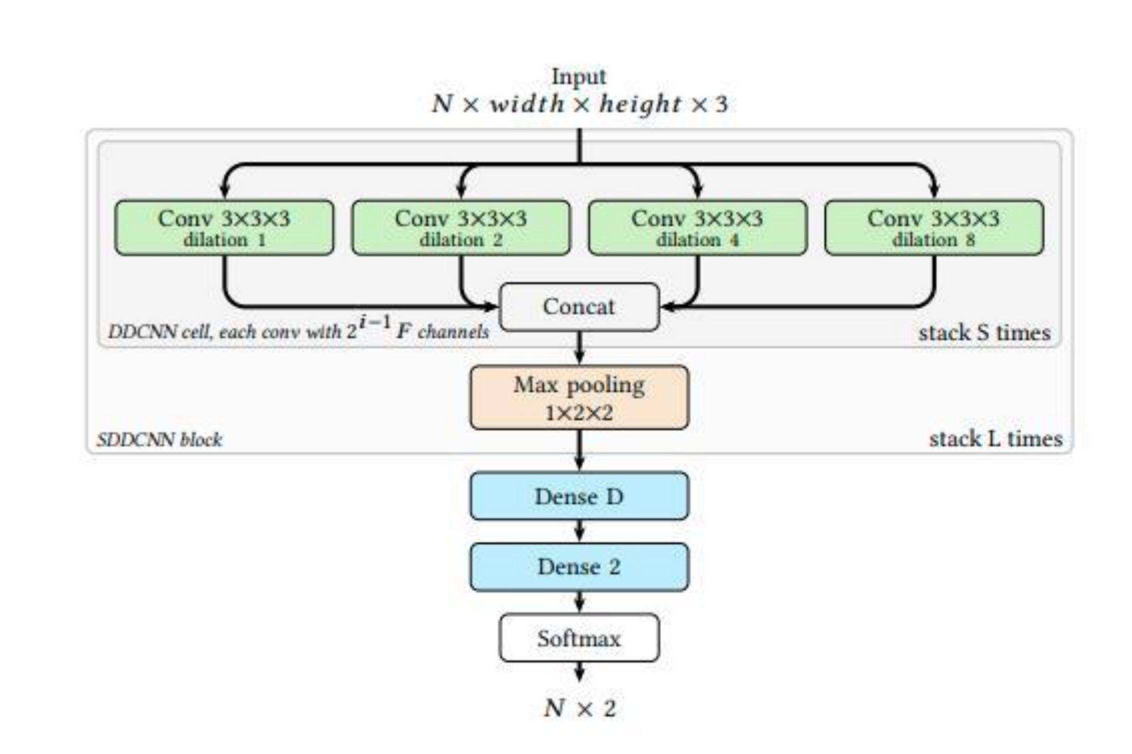


Class Diagram



Algorithms Used

- Shot Boundary Detection:
    - This module is executed for pre-processing the uploaded video. This module will detect shot boundaries and create a metadata file for the video. This metadata file is used during the summarization.
    - A deep learning approach is used to determine the shot boundaries. A Neural Network of TransNet architecture is used.



TransNet architecture for S = 1 and L = 1. N is length of video sequence.

S -> number of Dilated DCNN cells stacked

L -> number of stacked layers

- 0/1 Knapsack:
  - We apply solution of the 0/1 Knapsack problem using dynamic programming for choosing frames in the summarized video. The length of summarized video is 15% of the original video. Hence, based on the scores of the frames we choose all the frames which add up to 0.15 $*$lengthOfOriginalVideo.

- Cycle-GAN:

    - A model termed Cycle-SUM, which consists of a frame selector and a cycle consistent learning-based evaluator is used to generate summaries. The selector is a bidirectional LSTM network that learns the video representations that embed the long-range relationships among the video frames. The evaluator consists of two GANs, in which the forward GAN is learned to reconstruct original video from summary video, while the backward GAN learns to invert the processing.

Testing

- Snapshot Testing
    - Developing UI's that are pixel perfect can achieved with the help snapshot tests. To make sure the UI is consistent throughout the app's usage, we require snapshot testing. We used Jest framework to test the UI on our frontend platform.

- Alpha Testing

    - The objective of this form of testing is to identify all possible issues or defects before releasing it into a production environment. Alpha testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha Testing is conducted at the developer’s site. Inhouse virtual user environment was created for this type of testing.

- Unit Testing
  - It is the most popular from tests that can be used to assure the developer that a particular method would suffice the requirements of the product being delivered. Many utility methods and abstractions can be tested with the use of unit testing. Unit tests are also, often, simpler tests to write and to run.

Results

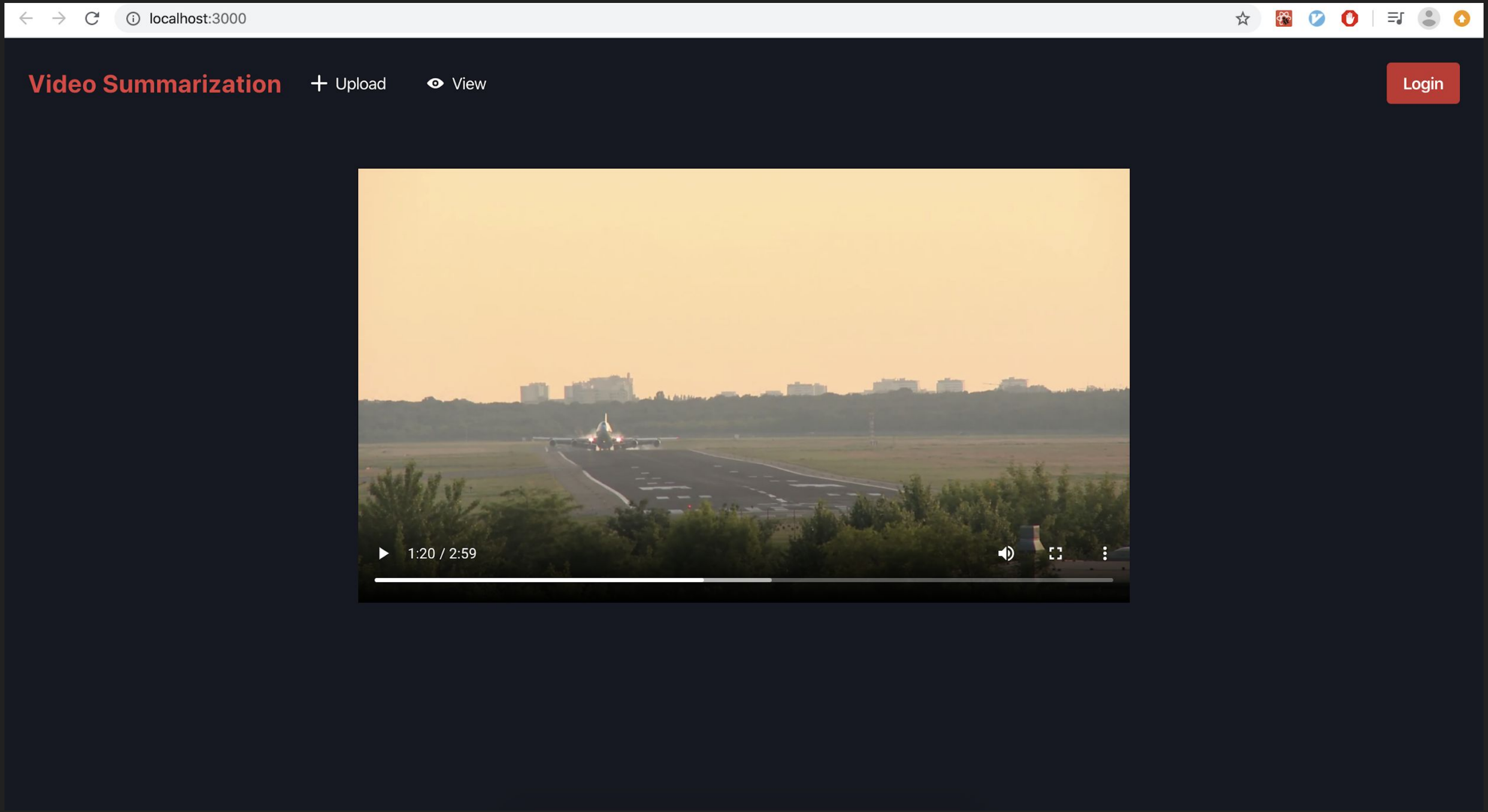
- SUM-GAN-AEE

```
FScores for Summe Dataset for 100 epochs
[57.41455099361301, 55.43136560714281, 55.03014249801121, 55.350918265389055, 55.01415387540639, 54.668103629426696, 55.40813855737173, 55.597316916426145, 55.59906826238121,
55.153411953309934, 55.14146847662916, 54.01425318441102, 54.869766226628826, 55.123860997092535, 55.37579133541061, 55.51066976095458, 55.46305669719161, 55.650452273206064,
55.22806739918875, 54.79934966482118, 54.087593914266236, 54.64244786836058, 54.61410115158402, 53.78204402246043, 54.84559161779496, 54.49853466205688, 54.66023206870053,
54.84546145228O745, 55.320186696792405, 55.122659732836794, 55.231967753732555, 55.19274781462774, 54.44913850098078, 55.387830732747354, 55.992181758599074, 55.5496932259285 2,
56.164841677562755, 54.998045235751974, 56.00973565673284, 56.63363947853101, 56.824670734498525, 55.546578047153616, 55.4008800236757, 55.537266037809204, 55.537266037809204,
55.477799423667946, 55.537266037809204, 55.477799423667946, 55.298834033330515, 55.71623142814663, 55.71623142814663, 55.89742167169088, 55.6589896672122, 55.477799423667946,
55.6589896672122, 55.6589896672122, 55.432045501438076, 55.6589896672122, 55.59634733392522, 55.834779338403905, 55.70790568803318, 55.834779338403905, 55.834779338403905,
55.47310152106834, 55.106976446544365, 55.468654263879934, 55.468654263879934, 55.468654263879934, 55.468654263879934, 55.468654263879934, 55.106976446544365, 55.34540845102306,
55.34540845102306, 55.38611420713103, 55.34540845102306, 55.38611420713103, 55.21232856857512, 55.34540845102306, 55.38611420713103, 55.793110963467214, 55.55467895898853,
55.106976446544365, 55.34540845102306, 55.112779889235, 56.08317374017649, 56.01186763186888, 55.106976446544365, 55.3424765742385, 55.85230152711532, 55.551107519567154,
56.2888728087206, 56.2888728087206, 56.2888728087206, 56.2888728087206, 56.2888728087206, 56.2888728087206, 56.2888728087206, 56.2888728087206, 55.551107519567154,
55.78373608135522, 56.2888728087206]
```

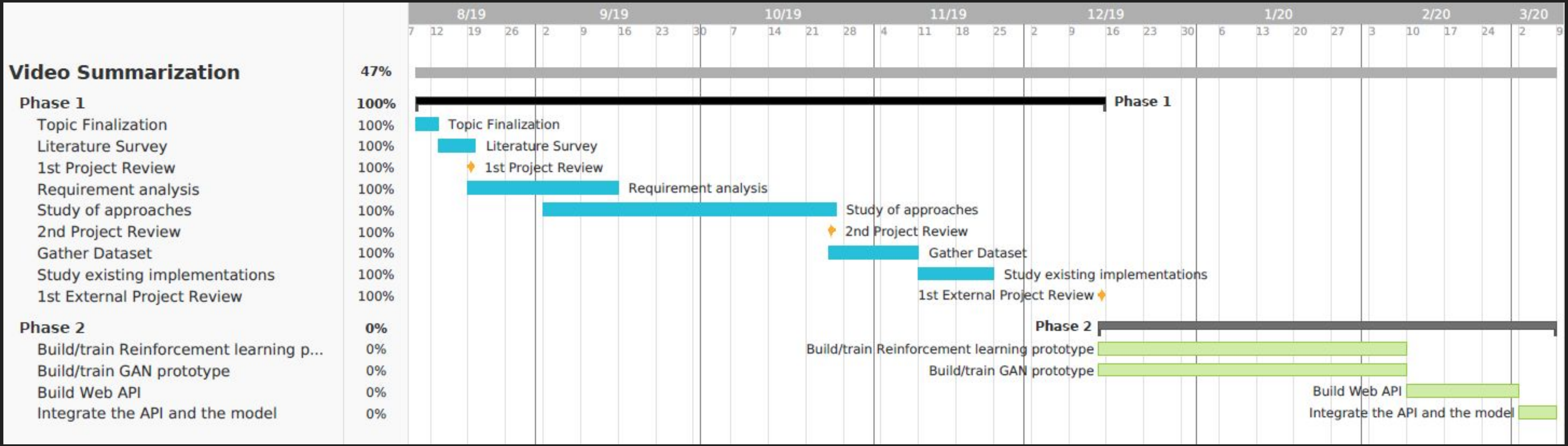
- SUM-GAN-SL

```
FScores for 100 epochs (Summe Dataset)
[38.70381762092773, 45.24900474313529, 36.33895250291619, 35.40260157835253, 34.74454843261277, 34.46107278063688, 34.46107278063688, 31.29892454849587, 34.06309726706819,
34.06309726706819, 40.71169482077581, 34.06309726706819, 35.08274210613588, 35.81125982910085, 35.81125982910085, 42.45068174245701, 42.45068174245701, 42.14979047685985,
42.14979047685985, 44.397743193515836, 44.397743193515836, 43.96062273900332, 41.71267002234733, 43.95763207177789, 43.95763207177789, 43.969610378155295, 43.969610378155295,
43.963616010205435, 43.963616010205435, 43.963616010205435, 43.963616010205435, 43.963616010205435, 43.963616010205435, 43.963616010205435, 44.3711113901592, 44.3711113901592,
44.3711113901592, 44.06543732318066, 44.06843320176187, 43.30686972865887, 45.402706798329795, 43.00703907531367, 40.400593973182275, 41.33993658109173, 44.91650094623715,
43.8078022507233l, 43.596317344525865, 43.596317344525865, 43.596317344525865, 43.596317344525865, 43.596317344525865, 43.596317344525865, 42.51560203953594, 43.62430073504978,
43.62430073504978, 43.62430073504978, 44.7050160400397, 42.51560203953594, 42.51560203953594, 43.62430073504978, 43.62430073504978, 43.62430073504978, 42.59869944368578,
42.59869944368578, 42.59869944368578, 43.62430073504978, 42.59869944368578, 42.59869944368578, 42.59869944368578, 42.59869944368578, 45.16180468578261, 45.16180468578261,
45.16180468578261, 45.14074187060215, 45.16180468578261, 45.152762004326135, 45.152762004326135, 45.05764446645232, 46.16634316196616, 46.17836329569014, 45.152762004326135,
45.152762004326135, 45.069664600176296, 45.13474750265229, 45.13474750265229, 46.17836329569014, 45.152762004326135, 45.1026476806401, 46.98146421476177, 46.04145997288553,
47.8657182784285, 47.16466557096332, 47.18268007263717, 49.43063278929315, 49.43063278929315, 49.41261828761931, 49.41261828761931, 49.41261828761931, 49.43063278929315,
49.41261828761931, 49.41261828761931]
```

- Frontend of the Platform



Project Timeline



Applications

- Make media for social media platforms
- Sports Highlights
- Previews for Movies & TV Shows

Conclusion

We studied various algorithms which support the problem statement. These algorithms helped us to work on a wide range of methods and libraries to improve efficiency of the final algorithm. With this project we successfully built an application which will summarize uploaded videos and allow the user to download the summarised videos.

Future Scope

- Stream processed videos from within the platform.
- Use audio as an aspect to summarize the video providing the user a consistent experience with respect to the original content.
- Summarize the video according to the time constraint provided by the user.

References

- K. Zhou, Y. Qiao, and T. Xiang, “Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,” p. 8, December 2017.
- L. Yuan, F. E.H. Tay, P. Li, L. Zhou, and J. Feng, “Cycle-sum: Cycle-consistent adversarial LSTM networks for unsupervised video summarization,” CoRR, vol. abs/1904.08265, 2019.
- B.Mahasseni, M.Lam, and S.Todorovic, “Unsupervised video summarization with adversarial lstm networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, pp. 1–10, 2017.

- J. Lokoˇc, G. Kovalˇc´ık, T. Souˇcek, J. Moravec, and P.ˇCech, “A framework for effective known-item search in video,” in In Proceedings of the 27th ACM International Conference on Multimedia (MM’19), October 21-25, 2019, Nice, France, pp. 1–9, 2019.