Content Based Video Search Engine

Abhay
School of Computer Science and Engineering
Galgotias University Greater Noida, Uttar Pradesh
abhayverma6300@gmail.com

Himangshu Das
School of Computer Science and Engineering
Galgotias University Greater Noida, Uttar Pradesh
himangshu4das@gmail.com

Abstract— **Search in a typical search engine is based on hand coded tags. The person uploading the video assigns a few tags to it like #xyz. So, when a typical user searches for a video, he gets search results based on these tags. Therefore, these searches are not dynamic and quite limited. We propose a search engine where the search will be based on the content of a video and not the tags associated with it. The way to do this is that we'll have CNNs (Convolutional Neural Networks) at backend of the webpage, when a user searches for an object, person, or an animal the search engine will return the videos which really contains the required object. The search engine will offer thousands of classes to search for. Users will be able to upload a video, it will be processed automatically and after that when a user searches for an object in it, it will return the exact timestamps where that object is in that video. Further users will be able to search more than one objects, and the search engine will return all the timestamps of the video which contains those objects separately and in the same frame if there is any.**

Keywords: Content based, video search, Convolutional neural network, timestamp.

I. INTRODUCTION

We have proposed a different type of search engine which searches on the content of a video. Video sharing platforms rank videos on the search engine results page (SERP) by following signals:

1. Keyword relevance based on metadata (Title, description, and keywords)
2. Engagement metrics (Watch time, likes, comments, etc.)

Video sharing platforms look at how well your titles, descriptions, and content match a user's search query. When it recency are two of the biggest factors in SERP ranking for video. Video sharing platforms also factor in users' watch history; how many videos have users watched from your channel, and when did they last watch videos on the same topic? Additional rankingsignals include:

- How often users click on your video from theSERP
- View velocity (how quickly a video's views grow)
- How frequently a channel uploads new videos. New videos rank higher right away, which encourages creators to consistently create new content.
- Session time (how much time viewers spend on video sharing platforms after watching a video).

We propose a search engine where the search will be based on the content of a video and not the tags, description, name associated with it. The way to do this is that we'll have CNNs (Convolutional Neural Networks) at the backend of the webpage, when a user searches for an object, person, or an animal the search engine will return the videos which really contain the required object**.**

All calculations and data cited in this paper are with respect to ImageNet dataset.

II. LITERATURE SURVEY

A. Content Based Video Search: Is there a Need, and Is ItPossible? [1]

The paper notes that CBVR faces similar challenges to content-based image retrieval (CBIR), such as the semantic gap, which refers to the difference between low-level features used in content-based search techniques and high-level human understanding and interpretation of the visual and audio content. The lack of meaningful and comprehensive text annotation for videos makes an approach based on content similarity promising. The paper also notes that the differences between a high-level search intention and the low-level features used in CBVR may further exacerbate the semantic gap problem.To address these challenges, the paper suggests that CBVR can benefit from advances in CBIR research, particularly in the use of deep learning techniques for feature extraction and similarity matching. The paper also highlights some current work by the authors' team in developing a system for CBVR using a combination of deep neural networks and natural language processing techniques.

Overall, the paper provides a valuable analysis of the challenges and potential solutions for CBVR, highlighting the importance of addressing the semantic gap problem and leveraging advances in CBIR research.

B. Content – Based Video Retrieval[2]

The paper discusses how the increasing volume of digital video content, including professional video content and user-created content, presents an opportunity for the development of content-based video retrieval systems. While academic research initially led the development of such systems, digital video search has become a commonplace activity on the World Wide Web due to the millions of digital content items uploaded and downloaded daily.

C. *Content based video retrieval systems* [3]

Search engines use web crawlers to automatically compile their listings, which are essentially an index or catalog of web pages. These crawlers, also known as spiders, search through the web and make a copy of each page they find, which is then added to the index. As the crawlers revisit sites regularly, any updates or changes made to a website can impact its ranking on the search engine. However, it may take some time for a newly added page to be indexed and made available to users searching through the search engine.

D. *A distributed Content-Based Video Retrieval system for large datasets* [4]

Various content-based video retrieval (CBVR) approaches have been developed that rely on low-level features, such as color histograms, motion, texture, and shape-based retrieval of video objects. Motion-based indexing, in combination with other low-level features like color, has shown to improve the performance of CBVR systems significantly. Color is a robust feature and can be represented through various techniques such as color histogram, color correlogram, color coherence vectors, and color distribution entropy (CDE) method. These methods have been proposed by different researchers and are used for extracting color features.

III. IMPLEMENTATION

A. *Problem Formulation*

The problem with these search methods are that the search is majorly based on metadata of a video like its name, description, tags, number of views etc. We don't actually know what the video contains. It is more dependent on the person's ability to give the video a proper metadata.

B. *Tools and Technology Used*

Python based Deep Learning models such as ResNet [5], MobileNet [6] will be exploited for the development and experimentation of the projects. And dataset will be created through ImageNet [4] for training.

Tools:

- Python
- Google Colab
- Resnet and MobileNetV2
- Pycharm
- Open CV
- Flask

C. *Working*

In this study, we utilized a Python code that employs the MobileNetV2 pre-trained model to detect objects in a given video file. The code uses OpenCV to capture and process each frame of the video, reducing the frame rate to 5fps to optimize processing time. The MobileNet model predicts the top 5 objects in each frame and their confidence score. The results are stored in a Pandas dataframe and written to a parquet file for further analysis. The parquet file can be read as a PyArrow table and converted back to a pandas dataframe. This approach provides an efficient method for object detection in videos, allowing for the collection and analysis of large amounts of data. This method has potential applications in a range of fields, including computer vision, surveillance, and security.

IV. DESIGN OF THE PROJECT

The search engine has the following sections:

1. When user logs in, the UI prompts the user with two images and an option to select from 4 different options for the first iamge. The first image has one correct label to select for authentication. Once the user selects the right label, the prompts moves to ask for a label in a text box for the second image. This image is also asked in a similar manner from 'n' number of other users post authentication. Once we get sufficient number of labels for a particular image we hold the most common label as the label for the unlabeled image. This labeled data is further used to train the models.
2. The user can upload videos and/or use previously stored videos for querying.
3. After querying the user interface returns the videos with the match along with the frames and time of match.

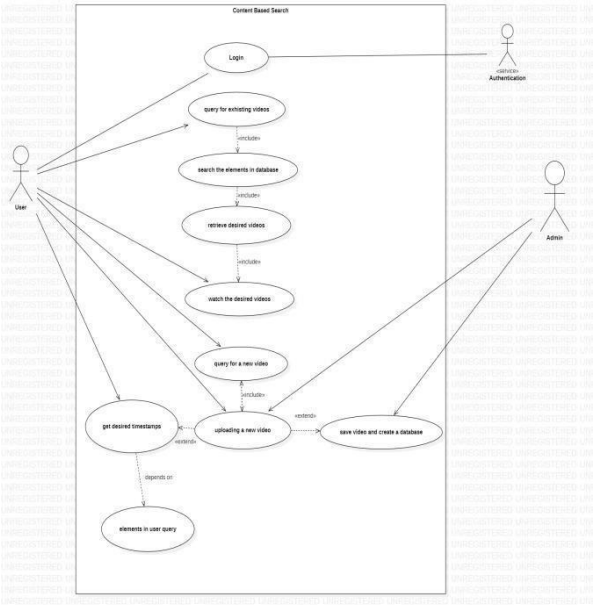

Fig 1: Proposed system use case diagram.

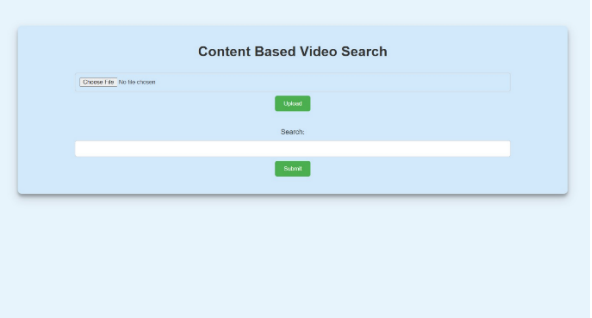

Fig 2: Webpage sample.

V. MODEL SELECTION

A. *ResNet50 vs ResNet152*

ResNet152 provides lower top1 and top5 errors as compared to ResNet50. In fact ResNet152 provides one of the lowest errors as compared to its counterpart ResNets.

The top1 error is the proportion of the time the classifier does not provide the highest score to the correct class. The top5 error rate is the percentage of times the classifier failed to include the proper class among its top five guesses.

The following data is calculated on ImageNet dataset.

Method	Top-1 Error	Top-5 Error
ResNet 34 B	21.84	5.71
ResNet 34 C	21.53	5.60
ResNet 50	20.74	5.25
ResNet 101	19.87	4.60
ResNet 152	19.38	4.49

Table 1: Error rates

However considering the Float Operations per Second (FLOPS) we realize that ResNet50 significantly lightweight, making it optimal for mobile application and deployment.

Method	FLOPS (in billions)
ResNet 152	11.3
ResNet 101	7.6
ResNet 50	3.8
ResNet 34	3.6
ResNet 18	1.8

Table 2: FLOPS of ResNets

B. *MobileNet*

MobileNet is a neural network architecture dedicated to mobile application due to its very low FLOPS. MobileNet provides top1 error accuracy of 71.88% and top5 error accuracy of 90.29% with just 314 million FLOPS (Tested on ImageNet[7]). Therefore, it is optimal for light weight mobile application.

The following parameters of the architecture are what makes it efficient as well as accurate[8]:

- 1x1 Convolution
- Batch Normalization
- Convolution
- Depthwise Separable Convolution
- Dropout
- Inverted Residual Block
- Residual Connection
- ReLU6
- Max Pooling
- Softmax

There are two ways in which our search engine will work –

1. The user is prompted with a picture with labels to select from. If the user selects the right option for the image, The system gets some confidence on the labelling of the user. Next the user is prompted with another image to label. A similar but random process is carried out with multiple users. Once a sufficient number of feedbacks are received from the users, we find the most common labels given by the users for the specific image. This is set as the label for the new image and will be used to train the models through transfer-learning further.
2. A user will visit the page and search for some content query. Based on that query the search algorithm will sort existing videos available on the site with timestamps for every object(content) asked by the user.
3. A user can also upload a video. In this case, based on the query the user gives specific models will be picked (to get fast results) and then inferenced on the video. After creating the dataset, it would return the labeled timestamps. The user will be able to jump to the timestamp that contains the object queried.

VI. RESULTS

The following table shows a sample of processing by the proposed design. This sample can further be used by applying keyword queries to match detected objects. The classes can further be increased using more datasets as required. The data is stored in parquet files for easy integration and querying. The data stored contains the path of the file, frame number, object detected, the confidence the model has on the classification and time in minutes:seconds.

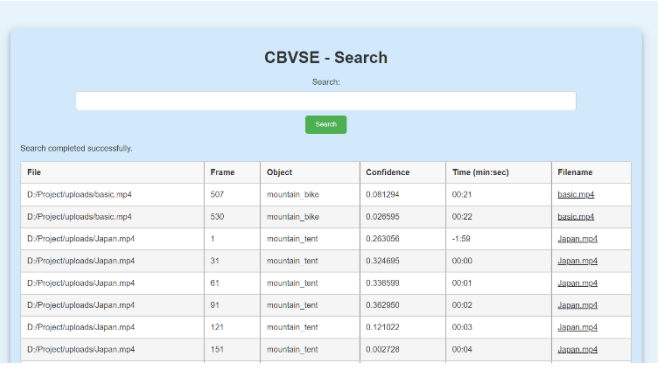

Fig 3: Sample website post querying.

File	Frame	Object	Confidence	Time (min:sec)
D:\Project\uploads\pex_vid.mp4	201	stethoscope	0.738525	00:03
D:\Project\uploads\pex_vid.mp4	251	computer_keyboard	0.535635	00:04
D:\Project\uploads\pex_vid.mp4	301	computer_keyboard	0.508877	00:05
D:\Project\uploads\pex_vid.mp4	551	computer_keyboard	0.468968	00:10
D:\Project\uploads\pex_vid.mp4	601	computer_keyboard	0.452141	00:11
D:\Project\uploads\pex_vid.mp4	501	computer_keyboard	0.449430	00:09
D:\Project\uploads\pex_vid.mp4	101	cleaver	0.423708	00:01
D:\Project\uploads\pex_vid.mp4	401	computer_keyboard	0.396637	00:07
D:\Project\uploads\pex_vid.mp4	351	computer_keyboard	0.283172	00:06
D:\Project\uploads\pex_vid.mp4	501	mouse	0.279969	00:09

Fig 4: Sample dataset post implementation

VII. CONCLUSION

To conclude the report, the project created not only resolves the issues that exists in the old model of video searching and parsing but also provide convenient solution to the user to access a more specific and accurate search query over the ones provided by metadata and tags. Factors like time complexities are also accounted for due to which models like ResNet 152 are not chosen for the application. Finally the paper concludes the effective implementation of content based video search with MobileNetV2 and ResNet50. Further we can implement instance segmentation and object detection using same methods to get even better and convienient results.

VIII. REFERENCES

[1] Huang, Zi, et al. "Content-Based Video Search: isthere a need, and is it possible?." *2008 International Workshop on Information-Explosion and Next Generation Search*. IEEE, 2008.

[2] Patel, B. V., and B. B. Meshram. "Content based video retrieval." *arXiv preprint arXiv:1211.4683* (2012).

[3] Patel, B. V., and B. B. Meshram. "Content based video retrieval." *arXiv preprint arXiv:1211.4683* (2012).

[4] Saoudi, El Mehdi, and Said Jai-Andaloussi. "A distributed content-based video retrieval system for large datasets." *Journal of Big Data* 8.1 (2021): 1-26.

[5] ImageNet - https://www.image-net.org/

[6] Deep Residual Learning for Image Recognition Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.

[7] “A lightweight deep neural network with higher accuracy” Liquan Zhao ,Leilei Wang,Yanfei Jia,Ying Cui

[8] “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” - Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam