International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

Parking Surveillance Footage Summarization

Shesha Sai Balaji K P[1], Sourav N R[2], Dr. Shashikala[3]

[1,2] *VIII Semester, B.E., Dept. of ISE, BNMIT, Bangalore, Karnataka, India*
[3]*Professor and Head, Dept. of ISE, BNMIT, Bangalore, Karnataka, India*

Abstract – *Security has become a priority in everyone's life. As a result, the number of security cameras has increased exponentially, necessitating massive storage capabilities to archive the footage. Due to many inactive parts, reviewing the entire footage is time-consuming and monotonous. This work aims to provide users with a textual summary of the vehicles in the footage making reviewing easier. Firstly, unnecessary frames are removed to obtain a shorter video. The resultant video is then processed frame-by-frame using CNN for object classification and number plate recognition to obtain the textual summary.*

Key Words: CNN, OpenCV, Security, Summary, TensorFlow

1. INTRODUCTION

Surveillance is the monitoring of behavior, a range of activities, or data with the goal to acquire information, influence, manage, or direct. Security cameras, usually referred to as surveillance cameras, are employed to keep an area under observation. These are used in a variety of circumstances by a range of businesses, institutions, or organizations. Typically attached to a recording device or an IP network, they can be watched by a security guard or law enforcement officer to see if any criminal conduct is occurring. One such kind of surveillance is "Parking Surveillance". It deals with monitoring and recording the activities occurring in the parking lot. It is possible to detect if an improper activity has occurred by viewing the selected footage at the appropriate times.

Deep learning is an important domain in AI, and a lot of research is being done on applying deep learning algorithms to image datasets to obtain valuable insights and forecast possible future outcomes. Convolution Neural Network (CNN) excels in several tasks connected to computer vision, such as detection, classification, and recognition of both images and videos.

2. RELATED WORK

Advancement in technology and processing capabilities has equipped us with the ability to develop, deploy, and test complicated neural networks.

Yubo An et al, [1] put forth a method for Video Summarization that uses Temporal Interest Detection and Key Frame Prediction for supervised video summarization. A flexible universal network frame is constructed to predict frame-level importance scores and temporal interest segments simultaneously for the purpose of video summarization, which is presented as a problem of integrating sequence labelling and temporal interest detection. The following are the primary contributions and innovations: (i) The video summarization is first described as a problem combining sequence labelling and temporal interest detection. (ii) Based on the problem specification, a universal network frame is presented that forecasts frame-level relevance scores and temporal interest segments at the same time. The two components are then merged with different weights to produce a more thorough video summary. (iii) The universal framework successfully employs the convolutional sequence network, long short-term memory (LSTM), and self-attention mechanism. A general framework can also be easily adapted to other effective models.

V. Choudhary et al, [2] remove spatial redundancy by displaying two activities that took place in distinct frames at different spatial locations in a single frame, spatial redundancy is eliminated. Temporal redundancy is eliminated by identifying frames with low activity and discarding them. Then a stroboscopic video is generated that traces the path of the retrieved object. The following steps are used to create a video summary from the input video: (i) Objects are extracted from the input video. (ii) A collection of non-overlapping segments is chosen from the original objects. (iii) Each segment is given a temporal shift, resulting in a shorter video summary with no occlusions between objects.

K. Muhammad et al, [3] presented a productive CNN-based summarising technique for resource-constrained devices' surveillance videos. Shot segmentation, which is the foundation of video summarising systems, affects the overall quality of the final summary. Within each shot, the frame with the highest memorability and entropy score is considered a keyframe. When compared to state-of-the-art video summarising approaches, the suggested method performs encouragingly on two benchmark video datasets. This framework aims to propose an energy-efficient CNN-based VS technique for surveillance videos captured by resource-constrained devices. It is motivated by the power of CNNs for diverse applications. Shot segmentation based on deep features, computing picture memorability and entropy for each frame of the shots, and keyframe selection from each shot for summary generation are all part of the framework.

Finally, a color histogram difference based post-processing step is used to remove the duplicate frames.

M. Rochan et al, [4] addresses the problem of video summarization. The objective is to choose a portion of frames from the input video to generate a summary video that accurately captures the input video's key points. Video summary is a useful tool for video search, retrieval, browsing, and other uses because there are so many videos available online. The difficulty of summarising videos might be viewed as a problem with sequence labelling. This work offers completely convolutional sequence models to address video summarization, in contrast to prior approaches that utilize recurrent models. Prior to adapting a well-known semantic segmentation network for video summary, the authors first create a novel relationship between semantic segmentation and video summarization. The usefulness of these models has been demonstrated by in-depth tests and analysis on two benchmark datasets. The challenge of video summary is referred to as a keyframe selection problem. Fully convolutional networks (FCNs) have been suggested for video summarization. FCNs are frequently employed in semantic segmentation.

M. Prakash et al, [5] compares and contrasts the 2 categories of Video-to-video summarization: (i) Static Video Summarization and (ii) Dynamic Video Summarization. In the former, relevant key-frames are extracted while in the latter, relevant video shots are extracted. Video Summarization techniques may be based on the classification of the clustering-based approaches. To extract relevant parts of the video several features of the frames like Colour Distribution, Contrast, sharpness, edges can be considered. RNNs (Recurrent Neural Networks) like LSTM (Long Short-Term Memory) have been used to annotate the images by giving the description of the image in the textual format. The same concept can be extended to annotate the extracted relevant parts of the summarized video.

S. Sah et al, [6] proposes techniques that leverage methods that make use of current developments in text summarising, video annotation and summarization to summarize hour-long videos to text. The important contributions of this work: (i) The capability of dividing a video into superframe parts and ranking each segment according to the visual quality, cinematography standards, and consumer preference. (ii) integrating new deep learning findings in image categorization, recurrent neural networks, and transfer learning to advance the field of video annotation. In order to create textual summaries of videos that are legible by humans, (iii) textual summarising techniques are used, and (iv) knobs are provided so that both the length of the video and the written summary can be adjusted. The proposed approach has 4 main components: (i) Finding intriguing sections in the entire video. (ii) selection of the key frames from these intriguing portions. (iii) A deep video-captioning network is used to create annotations for these keyframes. (iv) To provide a paragraph summary of the events in the video, the annotations are condensed.

G. Dhiraj Yeshwant et al, [7] suggest a technique for Automatic Number Plate Recognition (ANPR). Today, ANPR systems are employed in a variety of settings, including automated toll collection, parking systems, border crossings, traffic management, law enforcement, etc. A histogram-based strategy was utilised in this work, which has the advantage of being straightforward and quicker than any other approach. To obtain the necessary information, this method goes through four basic processes. Image capture, plate localization, character segmentation, and character recognition are these four phases. The execution consists of: (i) Change a colour image to a grayscale image (ii) Dilation: a method that enhances a given image by filling in image gaps, sharpening object edges, reuniting broken lines, and boosting brightness. (iii) Horizontal and Vertical Edge Processing (iv) Passing histograms through a low-pass digital filter to prevent loss of important information in further steps (v) Filtering out unwanted regions in an image (vi) Segmentation: In this stage, the regions are detected where the license plate is probable to be present. (vii) Character recognition: The entire alphanumeric database is compared to each individual character using Optical Character Recognition (OCR).

U. Shahid et al, [8] propose an algorithm that is specifically modelled to identify automobile licence plates. The system is first taught using the dataset of collected number plates, and this process is repeated until the machine learns. Successful machine learning will result in more processing. An image of a car taken by a camera at a distance of two to three feet serves as the input. This image is processed using Number Plate Extractor (NPE), which separates the characters of the image and outputs it for segmentation. The data for each character is then stored in a row matrix. Finally, the recognition component uses a trained neural network to identify the characters and generate the licence number.

G. Keerthi Devipriya et al, [9] deals with Classification and Recognition of images. They use a classifier algorithm and an API that includes a collection of pictures to compare the uploaded image with the set of images present in the data set under consideration, exposing the performance of the training models. The image is added to the appropriate class after being determined for it. Images are classified using a machine learning model that compares them and places them in the corresponding classes. The implementation includes: (i) Machine Learning Process that implements the classification of images by utilizing the strategy of supervised learning. (ii) Image classification using Bag of features: The concept of 'Bag of Features' has been inspired by "Bag of words". Every feature of the image has been considered as a word in this model, analogous to the Bag of Words, and comparison has been done similar to the word comparison in documents.

M. Azlan Abu et al, [10] propose an image classification architecture using Deep Neural Network (DNN), also known as Deep Learning by using the TensorFlow framework. The input data largely focuses on the flower category, which has five (5) distinct flower variations. Due to its high accuracy rate, a deep neural network (DNN) has been selected as the best alternative for the training procedure. In the findings, the categorization accuracy of the images is shown as a percentage. The average outcome for roses is 90.585 percent, and the typical outcome for other types of flowers is up to 90 percent or higher. This procedure is divided into four stages. The procedure continues with the collection of some of the images (inputs), followed by the application of DNN, and finally, the classification of all images into their groups.

The proposed approach summarizes the input footage to containing only those frames with vehicles in them. Objects are detected from these frames using the OpenCV and TensorFlow modules. CNN algorithm is used for the object classification into two/four-wheelers. The classified objects are further processed to obtain the license number.

3. METHODOLOGY

3.1 Preprocessing and Segmentation

Image enhancement necessitates image processing. A pre-processing phase is performed before the images are fed into the proposed structure. Images are extracted from videos via OpenCV. The first step is to lower the original image's dimensionality by downscaling it from 512 x 512 x 1 pixels to 128 x 128 x 1 pixels. This will let the network operate more quickly and with less complicated calculations. These photos are processed to detect vehicles, resulting in a collection of images that exclusively contain vehicles. These photos are merged to create a new video that solely contains vehicles. To enable the algorithm to train on unsorted data and prevent it from concentrating on a certain area of the entire dataset, the data is separated and then shuffled. Training, validation, and test sets of data are each assigned a unique set of target labels (68% for training and 32% for system test and validation). Finally, we improve the model's robustness and reduce overfitting by enhancing the photos to the point where the system recognizes them as brand-new. In addition to the geometric augmentation, the photographs are subjected to a grayscale distortion (salt noise).

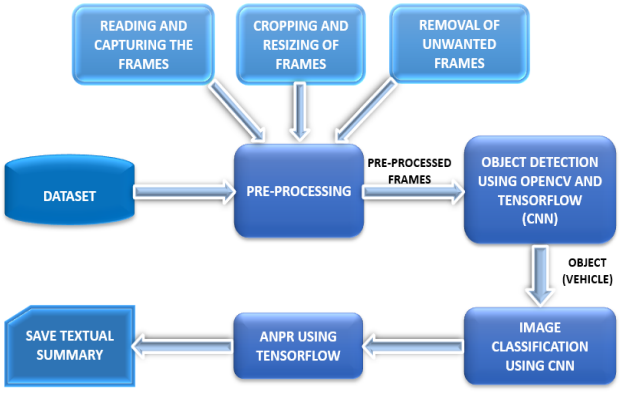

Fig-1: Architecture of the proposed work

Algorithm: PROCESS-VIDEO (video.mp4):
Description: pre-processes the actual footage to generate the summarized footage and
further processes the frame to generate a textual summary
Input: video.mp4
Output: Summarised video

1 If task = "pre_process", then
1.1 Capture the input video
vidcap = cv2.VideoCapture(video)
1.2 Extract the frames from the captured video
vidcap.read()
1.3 If a frame is extracted, then
1.3.1 Write the frame to an intermediate directory
1.3.2 Crop the frame to remove the unwanted space using img.crop()
1.3.3 Resize the frame to a specific size using img.resize() so that all frames are uniform
1.3.4 If a vehicle is present in that frame, then
I.3.4.1 Copy the frame to the result directory
1.4 Detect the vehicles from the extracted and resized images
1.4.1 if car is detected, then
1.4.1.1 Save the images with the cars in car_images folder
2 Otherwise, if task = "extract", then
2.1 For each frame with a vehicle in it, do:
2.1.1 Detect and recognize the number plate characters
2.1.2 If length(number_plate) == 10, then
2.1.2.1 Detect the vehicle class (two-wheeler/four-wheeler)
2.1.2.2 Write the result into a text file

3 Generate a video by combining the extracted frames with vehicles in the result directory

3.2 Feature Extraction and Object Detection

The Feature Extraction stage is required because certain features must be extracted in order for them to be unique. After determining if a vehicle is present, the last frame is examined. The representation set of huge data is reduced by using a feature vector. It contains information about the image feature that is extracted. Feature extraction is nothing but transforming such input data into a set of features. The main elements needed for image classification are extracted at this stage. The segmented vehicle image is used, and texture characteristics are extracted from it to illustrate the image's texture property. We use a pre-trained vehicle number plate model in conjunction with the images containing vehicles.

Algorithm: VEHICLE-DETECT (image.png):
Description: detects if there is any vehicle in the considered frame,
image.png
Input: image.png
Output: image.png gets saved to the output folder "/Result" if a
vehicle is detected

1 Read the image using cv2.imread()
2 Feed the image as input to OpenCV DNN
3 Repeat for all the objects detected
3.1 obtain the confidence_score of detection
3.2 if confidence_score > 40%, then
3.2.1 Obtain the id of the detection
3.2.2 Draw a box around the detection
3.2.3 Obtain the label of the detection
3.2.4 if label = 'car', then
3.2.4.1 Write the image to the output folder
4 Return the result

3.3 Classification and Number Plate Detection

Vehicle classification and number plate detection are accomplished using priorly generated features. Deep learning techniques are used to classify vehicles as either four-wheelers or two-wheelers. The main goal of deep learning algorithms is to learn and make informed choices automatically. In the proposed method, the CNN algorithm is used for classification.

Algorithm: VEHICLE-CLASS-DETECTION (image.png):
Description: classifies the detected vehicle into a 2-wheeler or a 4-wheeler
Input: image.png.
Output: Classification as a two-wheeler or four-wheeler

1 Read the image using cv2.imread()
2 Convert the image into a pixel array:
[height x width x channels]
image.img_to_array(img)
3 Add a dimension to the pixel array because the model expects the shape:
(batch_size, height, width, channels)
4 Import the pre-trained Keras CNN model
5 Compute the prediction score using the predict method in keras
6 Classify the object in the image based on the prediction scores for the labels
7 Return the classification label

Algorithm: NUMBER-PLATE-DETECT (image)
Description: detects the number plate and recognizes the characters on the number plate of the vehicle in image.png
Input: path to image.png
Output: detected number plate of the vehicle in image.png

1 Method: PROCESS-IMAGE (image)
1.1 Get the location of the number plate using the method 'GET-PLATE (image)'
1.2 If there exists at least 1 number plate image, then
1.2.1 Convert it to a grayscale image
1.2.2 Apply inverse threshold binary to obtain clear number plate characters using OTSU threshold approach
1.3 Obtain all the boundary points (x,y) of the characters in the number plate
1.4 for each boundary point (x,y), do
1.4.1 obtain x, y, width, height of character
1.4.2 crop the characters based on the ratio (height/width)
1.5 Load the CNN model trained on mobilenet dataset: LOAD-MODEL(path)
1.6 Load the class labels for character recognition
1.7 for each character detected, do
1.7.1 char = PREDICT-FROM-MODEL (character, model, labels)
1.8 Print the detected number plate final_string

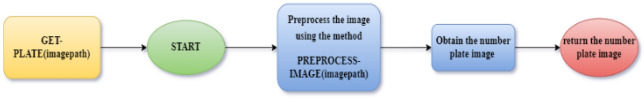

Fig-2: GET_PLATE method

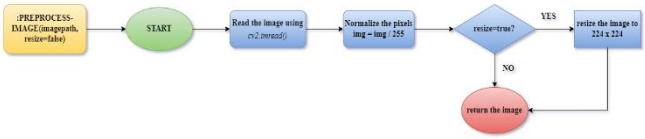

Fig-3: PREPROCESS_IMAGE method

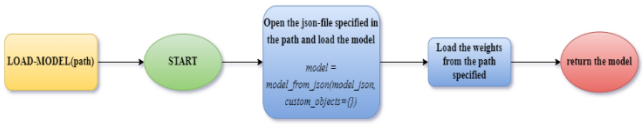

Fig-4: LOAD_MODEL method

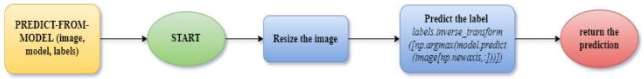

Fig-5: PREDICT_FROM_MODEL method

4. RESULTS AND DISCUSSIONS

The analysis of the number plate detection results obtained is tabulated in Table1. The results depend on the following:

- Image/frame quality - Better the image quality, better the results. Lower quality images make it difficult to detect the number plate region and also the cropping of characters is inaccurate.
- Obstructions to the number plate region in the image/frame - If some object is obstructing the number plate region, then the obstructed part of the plate will not be detected accurately.
- The angle of the vehicle in the image (straight/inclined angle) - If the image is at an inclined angle in the frame, then the detection is not accurate. A straight view of the image number plate region yields better results.
- The distance of the vehicle from the camera - The closer the detected number plate region to the camera, the better the accuracy of character detection.

Table -1: Tabular analysis of the ANPR results

Videos	No. of frames						
	No characters	1-5 characters		6-9 characters		10 characters	
		Total	Right	Total	Right	Total	Right
10 secs	31	10	4	2	0	4	2
15 secs	33	3	0	9	2	9	3
30 secs	66	13	2	9	0	10	3
10 mins	92	15	1	19	4	20	3
Accuracy		17.07 %		15.39 %		25.58 %	

Chart-1 provides a graphical analysis of ANPR module:

- The accuracy score when 1-5 characters are detected is better compared to when 6-9 characters are detected i.e., the accuracy has reduced. This is dependent on the image quality and angle of vehicle (number plate) in the image. In these frames, the number plate of the vehicle is not clearly visible due to its distance from the camera.
- When all 10 characters are detected, the accuracy score rises. All 10 characters are detected when entire number plate is visible i.e., when the vehicle (number plate) is straight and clearly visible in the image

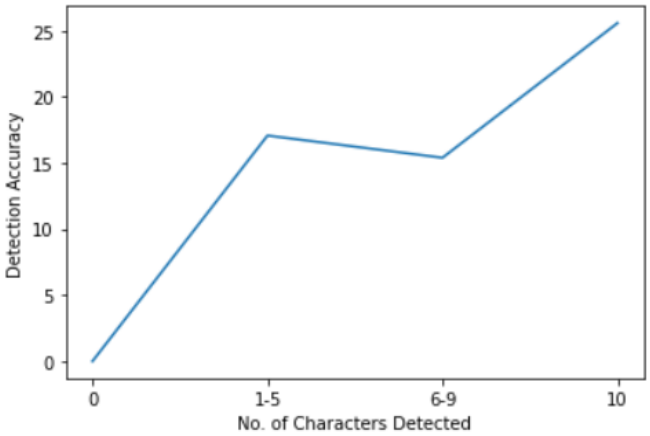

Chart-1: Line Graph analysis of ANPR

5. CONCLUSION AND FUTURE ENHANCEMENT

Textual description of visual content like images, particularly videos, is a difficult task, but a necessary step forward into machine intelligence and many applications in day-to-day activities such as generating video subtitles, downsizing the content of a longer video into small texts, etc. Video labeling is more challenging than image labeling due to a large number of objects, activities, backgrounds, varying picture qualities, and other unwanted information. One of the most significant roadblocks to textual labeling of video is the user perspectives.

The proposed work aims to provide users with a textual summary of the vehicles in the surveillance footage making its review easier. The summary consists of registration number, classification as a two-wheeler/four-wheeler of the vehicles detected in the footage.

This work can be further extended to multiple views of the area using footages from different CCTV cameras and integrating it human activity recognition thereby detecting any abnormal events happening.

ACKNOWLEDGEMENT

We would like to thank BNM Institute of Technology, Bangalore, Karnataka, India, for the constant support and encouragement for the successful completion of this project. We extend our gratitude to the **Dept. of ISE, BNMIT**, for providing the computing facilities in the "**GPGPU Computing for Computationally Complex Problems Laboratory**", which is supported by **VGST** under KFIST-L1..

REFERENCES

[1] Yubo An, Shenghui Zhao. "A Video Summarization Method Using Temporal Interest Detection and Key Frame Prediction", September 2021, eprint arXiv:2109.12581 (arXiv.org)

[2] Choudhary, Vikas, and Anil K. Tiwari. "Surveillance Video Synopsis." 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 2008. doi:10.1109/ICVGIP.2008.84

[3] T. Hussain, K. Muhammad, A. Ullah, Z. Cao, S. W. Baik and V. H. C. de Albuquerque, "Cloud-Assisted Multiview Video Summarization Using CNN and Bidirectional LSTM," in IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 77-86, Jan. 2020, doi: 10.1109/TII.2019.2929228.

[4] Rochan, M., Ye, L., Wang, Y. (2018). Video Summarization Using Fully Convolutional Sequence Networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11216. Springer, Cham. https://doi.org/10.1007/978-3-030-01258-8_22

[5] Madhura Prakash., Krishnamurthy G. N., "Deep Learning Approach for Video to Text Summarization," in International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), vol. 10, issue 1, Jan. 2021. doi: 10.15680/IJIRSET.2021.1001032

[6] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud'Hommeaux and R. Ptucha, "Semantic Text Summarization of Long Videos," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 989-997, doi: 10.1109/WACV.2017.115

[7] Gaikwad Dhiraj Yeshwant, Samhita Maiti, P. B. Borole, 2014, Automatic Number Plate Recognition System (ANPR System), International Journal Of Engineering Research & Technology (IJERT) Volume 03, Issue 07 (July 2014)

[8] Shahid, Umar. (2017). Number Plate Recognition System. 10.13140/RG.2.2.23058.71361

[9] G.K. Devipriya, E. Chandana, B. Prathyusha, T. S. Chakravarthy., "Image Classification using CNN and Machine Learning," in International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), vol. 5, issue. 2, 2019. doi: https://doi.org/10.32628/CSEIT195298

[10] Abu, Mohd Azlan & Indra, Nurul Hazirah & Abd Rahman, Abdul & Sapiee, Nor & Ahmad, Izanoordina. (2019). A study on Image Classification based on Deep Learning and Tensorflow. 12. 563-569.