



Effective Video Scene Analysis for a Nanosatellite Based on an Onboard Deep Learning Method

Natnael Alemayehu Tamire ¹ and Hae-Dong Kim ^{2,*}

¹ Aerospace System Engineering, University of Science and Technology (UST), Daejeon 34113, Republic of Korea; natnael@ust.ac.kr

² Department of Aerospace and Software Engineering, Gyeongsang National University, Jinju 52828, Republic of Korea

* Correspondence: haedkim@gnu.ac.kr

Abstract: The latest advancements in satellite technology have allowed us to obtain video imagery from satellites. Nanosatellites are becoming widely used for earth-observing missions as they require a low budget and short development time. Thus, there is a real interest in using nanosatellites with a video payload camera, especially for disaster monitoring and fleet tracking. However, as video data requires much storage and high communication costs, it is challenging to use nanosatellites for such missions. This paper proposes an effective onboard deep-learning-based video scene analysis method to reduce the high communication cost. The proposed method will train a CNN+LSTM-based model to identify mission-related sceneries such as flood-disaster-related scenery from satellite videos on the ground and then load the model onboard the nanosatellite to perform the scene analysis before sending the video data to the ground. We experimented with the proposed method using Nvidia Jetson TX2 as OBC and achieved an 89% test accuracy. Additionally, by implementing our approach, we can minimize the nanosatellite video data download cost by 30% which allows us to send the important mission video payload data to the ground using S-band communication. Therefore, we believe that our new approach can be effectively applied to obtain large video data from a nanosatellite.

Keywords: nanosatellites; satellite video; onboard video scene analysis; disaster monitoring; deep learning



Citation: Tamire, N.A.; Kim, H.-D. Effective Video Scene Analysis for a Nanosatellite Based on an Onboard Deep Learning Method. *Remote Sens.* **2023**, *15*, 2143. <https://doi.org/10.3390/rs15082143>

Academic Editors: Peter M. Atkinson and Ce Zhang

Received: 31 January 2023

Revised: 6 April 2023

Accepted: 13 April 2023

Published: 19 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to their low cost and quick development cycle, nanosatellites have become more popular in recent years for remote sensing and disaster monitoring [1,2]. Nanosatellites have constrained onboard computer (OBC) processing power and communication bandwidth in comparison to larger spacecraft.

Can we use a satellite to capture real-time videos of the earth's surface to track moving objects and record special events and disasters that occur on our planet? With the advancement of space-borne satellites outfitted with payload cameras capable of recording high-resolution videos, this dream could become a reality. High-resolution space-borne videos can potentially change how we observe the earth. This is because satellite videos can be used for various purposes, such as city-scale traffic surveillance, the 3D reconstruction of urban buildings, and earthquake relief efforts [3,4].

There will be several challenges when comparing video payload data to image payload data. Because video data is larger than image data, having high-resolution video data will necessitate more storage and communication costs (downloading bandwidth and time) to send all of the video data to the ground. Once the video data has been downloaded to the ground, it must be processed in a timely and efficient manner. Furthermore, the majority of the frames from the payload video data may be ineffective or irrelevant to

the mission under consideration [5]. So, sending all the video data to the ground station without first analyzing and processing them will be costly. Figure 1 depicts the challenge of having a nanosatellite with a video payload.

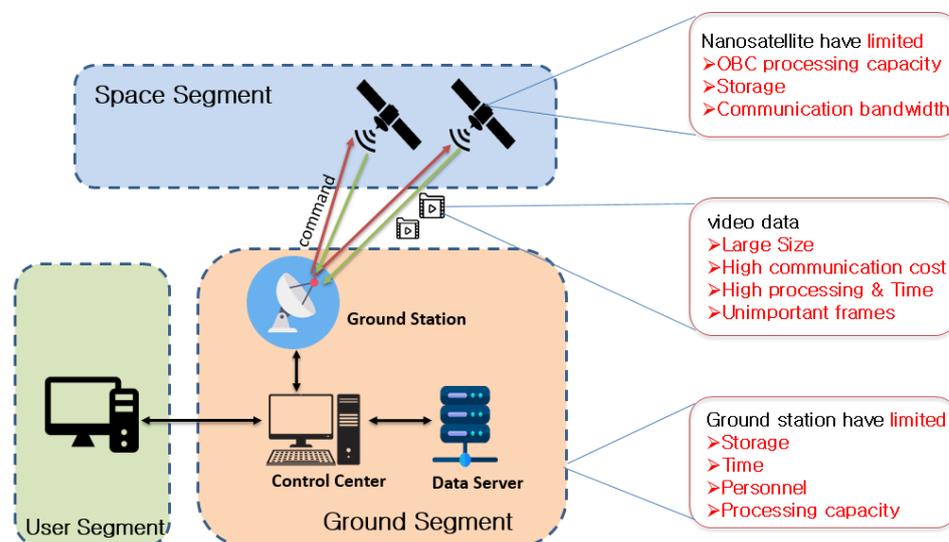


Figure 1. Nanosatellite video payload challenges. (Red line indicates the command sent to the nanosatellites, the green line indicates the download of the video payload data from the satellite to the ground, and the black line the transfer of the payload data in the ground system).

Many kinds of research works are being carried out to overcome these limitations and increase the onboard processing capability of nanosatellites by using GPUs and FPGAs [5–7]. One of the advantages of having the capability of onboard processing in nanosatellites is reducing the data to be downloaded [8–10].

Satellite video data processing has recently gained attention in aerospace engineering and remote sensing. Most of the research focuses on two areas: the super-resolution and tracking of objects using satellite data.

The first group of researchers focused on improving the quality of video taken from the satellite by increasing the low resolution to high resolution. Study [11] proposed a super-resolution method using a CNN for “Jilin-1” satellite video data without any pre-processing or postprocessing. In [12], the authors proposed a novel end-to-end deep neural network that generates dynamic up-sampling filters and generates much sharper HR videos; study [13] used feature extraction, a residual network containing ResNet, CNN, and up-sampling, for generating an HR video.

The other group of researchers tried to find a better solution for detecting objects such as cars, trains, or ships from satellite video imagery and tracking their movement. In the study [14], a predicting-attention-inspired Siamese network (PASiam) was proposed to track moving targets in space-borne satellite videos. In [3], a vehicle detection method using background subtraction with 94.7 % accuracy was proposed. In [15], vehicle detection in satellite video using FoveaNet was proposed, which achieved a 0.84 F1 score. In studies [5,16], a framework was proposed that fuses both the spatial feature and temporal (motion) representation of a satellite video for scene classification and identifying events of interest by using deep learning methods.

The researchers share two things in common, however: they all concentrated on processing video data after they have been received by the ground station, and the video data they used in the research came from the SkySat or Jilin-1 satellites, which are microsatellites and use X-band communication for downlink.

Meanwhile, In September 2020, the European Space Agency (ESA) launched a 6U CubeSat named PhiSat-1 into low Earth orbit. The mission was to demonstrate the use of onboard artificial intelligence (AI) to perform cloud detection from the images taken by

the payload hyperspectral camera to remove cloudy images before sending them to the ground. The AI model they proposed achieved 92% accuracy [17].

Lofquist and Cano [18] examined the adequacy of the NVIDIA Jetson Nano and NVIDIA Jetson AGX Xavier, which have low capacity and high performance, to be onboard a nanosatellite OBC and perform object detection in images by applying compression. The object detection network models they examined were the single shot multibox detector and region-based fully convolutional network and their performance was measured in terms of execution time, memory consumption, and accuracy. In study [19], the authors examined the usability of the Jetson board onboard a nanosatellite by performing a radiation test.

However, to the best of the authors' knowledge, onboard video payload data processing to minimize the downlink communication cost for nanosatellites with a video payload has not been deeply investigated. In this paper, we propose a new effective approach to minimize the high downlink cost incurred on a nanosatellite with video payload data. For this purpose, the deep learning method was used to identify important sceneries that were related to the mission from the videos and send only those to the ground.

Therefore, the contribution of this paper is developing an effective mechanism for minimizing the downloading communication cost for a nanosatellite with a video payload by applying deep learning video scene analysis onboard the satellite.

2. Problem Scenario

To the best of the authors' knowledge, to date, there have been no launched nanosatellite missions with video payloads, but there are ongoing projects such as High-Resolution Image and Video CubeSat (HiREV) by the Korea Aerospace Research Institute (KARI) [20].

HiREV is a 6U CubeSat capable of obtaining a 5 m color (3 m monochromatic) image and high-definition (HD) video. HiREV is designed to follow a sun-synchronous orbit with 97.8 inclination at 400 km altitude [20]. The optical payload camera of HiREV takes full high definition (HD) color video and uses the S-band for downloading the payload data, as depicted in Figure 2. The images and video taken by the payload camera are transferred to the onboard computer (OBC) through gigabit ethernet and are stored in the OBC memory until they are sent to the ground. HD video poses a storage and download bottleneck for the HiREV system because it has limited storage and uses an S-band for transmission. HD video demands more storage and transmission bandwidth than standard photos and videos.

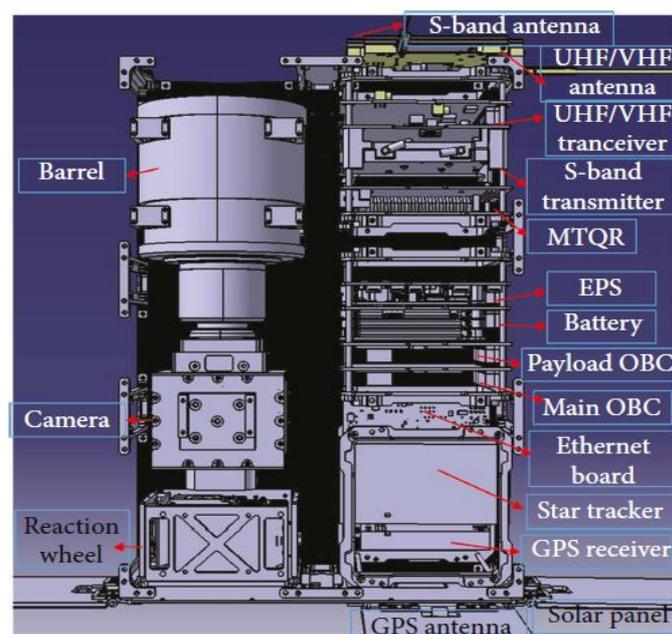


Figure 2. HiREV 6U Nanosatellite Subsystem [20].

To analyze the problem of the downlink bottleneck encountered by nanosatellites with video payloads, we took the HiREV CubeSat as our scenario satellite and assumed its mission was to monitor flood disasters in Ethiopia and the East Africa region, as flood disasters are becoming frequent in this area. The video data from HiREV were downloaded to the ground station located in Adama, Ethiopia.

The full-HD color video taken by HiREV has a duration of 120 s (2 min), a frame rate of 7 fps, and a frame size of 1920×1080 pixels, with 8-bit color depth as shown in Table 1.

$$\text{Raw Video Size} = (\text{Frame Rate} \times \text{Duration} \times \text{Frame Size})/1024 \quad (1)$$

Table 1. HiREV payload specification.

Specification	Value
Ground resolution	5 m from 400 km orbit
Pixel resolution	1920×1080 pixel
Frame rate	7 FPS
Focal length	500 mm
Aperture	89 mm
Data interface	Gigabit ethernet

So, using (1), the size of a raw video file data from the HiREV optical payload will be:

$$\text{Raw Video Size} = (7 \times 1920 \times 1080 \times 120)/1024 = 1701 \text{ Mb} = 1.7 \text{ Gb.}$$

After the compression of the raw file data with the MP4 compression method, the size will be reduced to 250 Mb. The duty cycle for the payload is 10% of one orbit. That is 30 videos per day, so the size of video data stored in the satellite to be downloaded would become $30 \times 250 = 7500 \text{ Mb} = 7.5 \text{ Gb}$.

Using the STK[®] Simulation tool, we measured the mean duration for our satellite to access and communicate with the ground station in a day to be 236 s.

$$\text{Data size to be downloaded} = \text{Bandwidth} \times \text{Mean Access Duration} \quad (2)$$

Based on this analysis and using (2), the video data to be downloaded per day using S-band communication will be:

$$\text{Data size to be downloaded} = 2 \text{ Mbps} \times 236 \text{ s} = 472 \text{ Mb}$$

From the scenario analysis, we examine that by having a nanosatellite with a video payload and S-band downlink communication, the amount of video data to be downloaded to the ground per day is 427 Mb, while the payload camera of our satellite record and store 7.5 Gb per day. Additionally, this means we will not be able to download all the videos in a day and eventually run out of storage in the satellite. As our mission is flood disaster monitoring, which requires timely data, it will be difficult to achieve the intended mission.

After analyzing the problem scenario, we observe that using S-band communication will not be efficient enough to send all the video payload data to the ground and this will cause a bottleneck on the onboard memory and the general operation of the mission. To solve the problem, we propose an approach to have an onboard video analysis that can help to use the memory efficiently and minimize downlink costs.

3. Video Scene Analysis

Video analysis is the capability of automatically analyzing video to detect and determine temporal and spatial events. Video scene analysis is an automatic process to recognize and classify objects from live-video sequences. It is a recent research topic due to its vital

importance in many applications, such as real-time vehicle activity tracking, pedestrian detection, surveillance, and robotics [21]. Despite its popularity, video scene analysis is still an open, challenging task and requires more accurate algorithms. However, advances in deep learning algorithms for video scene analysis have emerged in the last few years for solving the problem of real-time processing.

Deep Learning Algorithms for Video Scene Classification

The use of deep learning techniques has exploded during the last few years, and one of the areas is scene recognition from visual data. The most widely used deep learning algorithms for video scene classification are CNN and RNN [16].

For image classification, among all the variants of deep learning algorithms, CNN is the most widely used supervised learning model. A video consists of an ordered sequence of frames. Each frame contains spatial information, and the sequence of those frames contains temporal information. To model both aspects, we used a hybrid architecture that consists of convolutions (for spatial processing) as well as recurrent layers (for temporal processing). Specifically, we used a convolutional neural network (CNN) and a recurrent neural network (RNN) consisting of GRU layers. This kind of hybrid architecture is popularly known as a CNN-RNN [17].

One of the recurrent neural networks (RNN) is long short-term memory (LSTM). Unlike CNN, LSTM has internal mechanisms called gates that regulate the flow of information. During training, these gates learn which data in a sequence are important to keep and which they can forget. This allows the network to pass relevant information down the long chain of sequences to make a prediction.

4. Proposed Methodology

To tackle the scenario problem of minimizing the downlink cost for transmitting full-HD color video payload data from a nanosatellite with limited processing and communication capability, onboard processing with a lightweight and fast method is required.

We propose an onboard deep learning video scene analysis method that takes a raw video from the payload camera and recreates the video with only the important frames that contain the scenery related to the mission, disaster monitoring, which is smaller in size when sent to the ground. Using deep learning techniques, we train and create a lightweight model for video scene classification using a hybrid convolutional neural network (CNN) and long short-term memory (LSTM) model that can be easily loaded onboard the satellite and requires low processing capacity.

In our approach, the first step is for the required scene classes to be trained with the related satellite video data on the ground and then for the model to be loaded to the OBC. To classify the important (mission-related) and nonimportant frames from the video, we used the proposed hybrid CNN+LSTM model shown in Figure 3. The proposed CNN+LSTM model contains three convolutional layers with max pooling, an LSTM layer, one fully connected layer, and the output layer with two classes. The CNN layer was used for visual feature extraction and the LSTM layer was for sequence learning. The reason we chose these models is that as the number of CNN layers increases, the model becomes heavy and takes a longer time to perform the classification, while a smaller number of CNN layers will be faster and lighter at the expense of accuracy. As nanosatellites have limited processing capacity, the models that are processed onboard should be fast and lightweight.

Onboard the satellite, after the payload camera takes the video, the video data will be sent to the OBC. In the OBC, blur detection and removal will be performed to identify video frames with blurs due to the motion of the satellite. The blur detection is carried out using Gaussian blur detection. This step is advantageous as blurry frames are not important and their removal will decrease the frame size to be processed by the next step which is the scene analysis part.

The scene analysis is the major part to solve the downlink and memory bottleneck problem. In this part, we first detect the mission-related objects identified in the trained

model from the video using the pretrained model. After that, each frame will be classified based on the trained classes in the model and the scene time for each class will be calculated. Then, the frames that contain the class that is related to the mission will be identified. The other frames that are classified to contain nonimportant frames (not related to the mission objects) will be discarded. After identifying the required class of objects in the frames, we reconstruct the video from the frames that contain these objects. Then, the reconstructed video will be compressed using a compression algorithm. Finally, the compressed video, which is smaller in size and contains only the important frames, in our case related to the flood disaster, will be sent to the ground. The proposed method architecture is depicted in Figure 4.

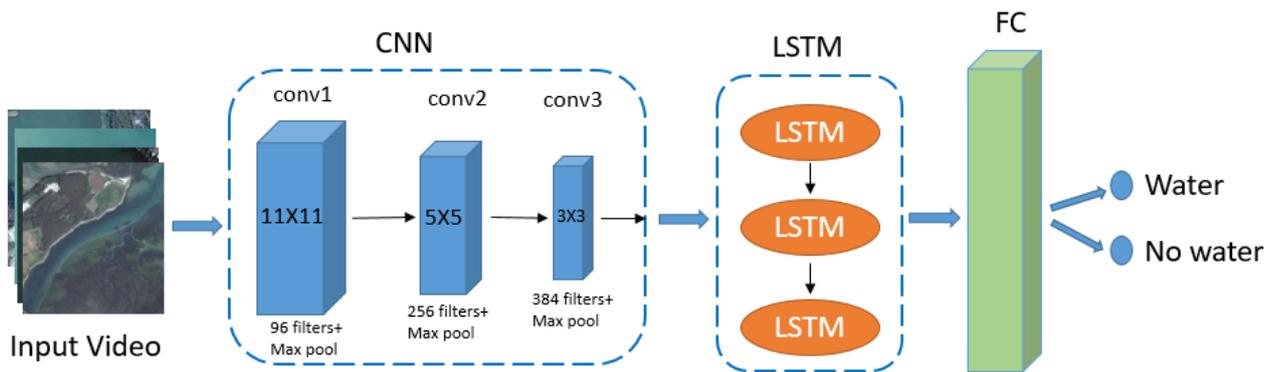


Figure 3. Proposed CNN+LSTM model for video scene analysis.

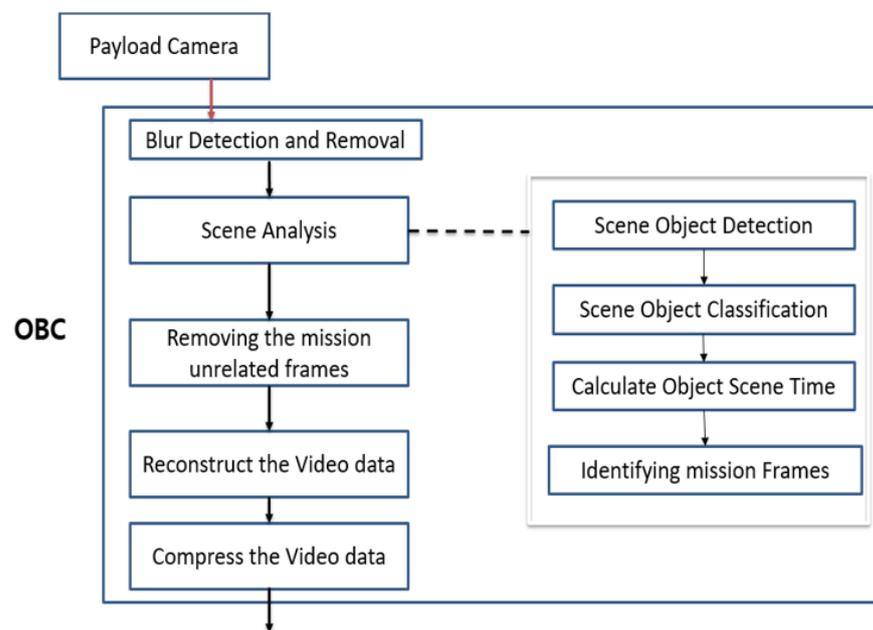


Figure 4. Architecture for the proposed onboard deep-learning-based video analysis.

5. Experiment and Results

5.1. Experiment Setup

For training the deep learning CNN+LSTM-based scene analysis in the ground system, we used a laptop with Core I7, 8th Gen, and a GPU Processor with 16 GB RAM. For testing the trained model onboard, we used the Nvidia Jetson TX2 development board.

For the experiment, we implemented the proposed CNN+LSTM video scene classification model using the NVIDIA Deep Learning GPU Training System (DIGITS) that has TensorFlow and Keras.

Because we could not obtain a satellite video dataset for training the model, we used a satellite image dataset and changed it into a video dataset. We used the publicly available RESISC45 dataset, which is the benchmark for remote sensing image scene classification (RESISC) created by Northwestern Polytechnical University (NWP). This dataset contains 31,500 images RGB images of 256×256 size and varying spatial resolution ranging from 20 cm to more than 30 m, covering 45 scene classes with 700 images in each class. However, as we were interested in sceneries that were related to water bodies such as rivers, sea, and wetlands, we rearranged the images into two classes: one class containing images with water bodies in them and the other containing images without any water bodies or scenery in them.

The video dataset we created contained two scene classes: one class for frames that contained flood-disaster-related scenes (water class) and the other class that did not contain flood-disaster-related scenes (no water class). The satellite video dataset used contained 10,096 image frames (30 videos with 300 frames each) for training and 1264 image frames (3 videos) for validation in the ground classification training, and 380 image frames (1 video) for testing the scene classification network onboard the Nvidia Jetson TX2, as shown in Figure 5.

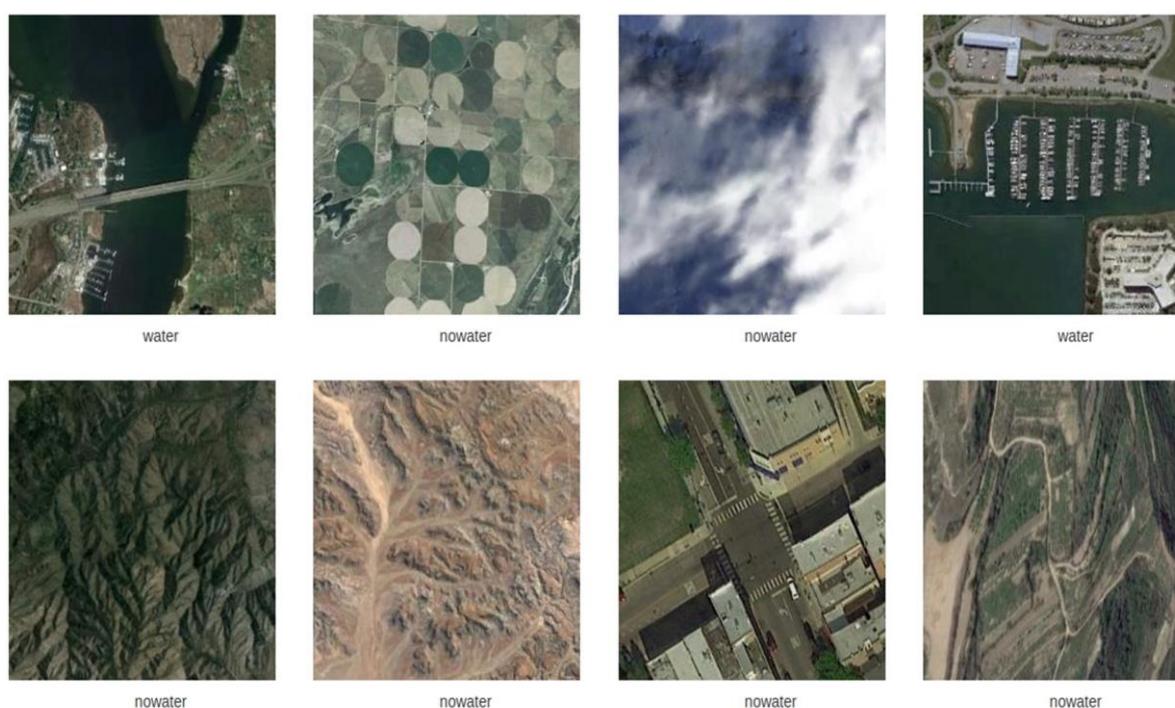


Figure 5. Video scene classification output classes from the test Video.

For comparing the performance of our proposed model, we implemented three video classification models: Alexnet, MobileNet, and GoogleNet. The reason we chose these models for comparison is that they are widely used for classification on mobile and embedded systems. We evaluated the comparison based on the classification accuracy, processing speed, and memory saving of the recreated video.

5.2. Experiment Results

We used 1 video with 115 frames, of which 50 frames contained no-water sceneries and 65 which contained water sceneries, to test the accuracy of the training. After training the scene classification with the models with 200 epochs, we obtained the confusion matrix of test accuracy results for the models as shown in Tables 2–5.

The test accuracy of the proposed CNN+LSTM model was 90% for the No Water class and 86% for the Water class.

Table 2. Confusion matrix for the proposed CNN+LSTM model.

	No Water	Water	Per-Class Accuracy
No Water	45	5	90.0%
Water	9	56	86.15%

Table 3. Confusion matrix for the AlexNet model.

	No Water	Water	Per-Class Accuracy
No Water	44	6	88.0%
Water	11	54	83.08%

Table 4. Confusion matrix for the MobileNet model.

	No Water	Water	Per-Class Accuracy
No Water	44	6	88.0%
Water	8	55	87.6%

Table 5. Confusion matrix for the GoogleNet model.

	No Water	Water	Per-Class Accuracy
No Water	43	7	86.0%
Water	12	53	81.5%

As the ground training result shows, the proposed model achieved a good scene classification accuracy compared to the other models. For the onboard experiment, we loaded the trained models on the Nvidia Jetson TX2 to test the performance of our scene classification models. Compared to the other models, the proposed model had a small number of parameters for the classification, so the storage size was smaller, and the classification speed was higher. To test the accuracy of the models onboard the Nvidia Jetson TX2, we created 1 video that had 380 frames, of which 193 were water scenes and 187 were no-water scenes; Tables 6–9 show the confusion matrixes for the test.

Table 6. Confusion matrix for the proposed CNN+LSTM model onboard Nvidia Jetson TX2.

	No Water	Water	Per-Class Accuracy
No Water	167	20	89.3%
Water	20	173	89.6%

Table 7. Confusion matrix for the AlexNet model onboard Nvidia Jetson TX2.

	No Water	Water	Per-Class Accuracy
No Water	163	24	88.6%
Water	32	161	83.4%

Table 8. Confusion matrix for the MobileNet model onboard Nvidia Jetson TX2.

	No Water	Water	Per-Class Accuracy
No Water	169	18	90.3%
Water	23	170	88.6%

Table 9. Confusion matrix for the GoogleNet model onboard Nvidia Jetson TX2.

	No Water	Water	Per-Class Accuracy
No Water	160	27	85.5%
Water	31	160	83.0%

The original video file used in the onboard test had a duration of 4.10 min and 75 Mb. Additionally, after applying the proposed approach onboard the Nvidia Jetson TX2 for both models, the reconstructed video file that was to be downloaded became reduced in duration and size. For the Alexnet model, the reconstructed file size had a duration of 1.73 min and a file size of 25 Mb, but only contained 83% of the important flood-disaster-related frame; for the MobileNet model, the reconstructed file size had a duration of 1.89 min and a file size of 30 Mb and only contained 8% of the important frames, while for the proposed CNN+LSTM model, the reconstructed file size had a duration of 1.87 min and a file size of 28 Mb and contained 89% of the important frames needed by our satellite mission. From the result, we can see that by implementing our approach, we can minimize the nanosatellite video data download cost by 30%.

Consequently, by using the proposed approach, the HD video data file taken by the satellite payload camera which is originally large-sized can be downloaded with minimized storage and downlink cost using S-band communication. The proposed approach can solve the downlink bottleneck described in the case of the HiREV nanosatellite mission scenario.

6. Conclusions

We proposed onboard deep-learning-based video scene analysis approach for a nanosatellite with video payload data and a flood disaster monitoring mission. Satellite video data can play a great role in disaster-monitoring missions as they can provide semantic and rich information compared to image data. As nanosatellites have limited storage, processing, and communication capability, performing onboard video scene processing before storing and sending the large raw data can significantly reduce the downlink cost.

The proposed method trains the model on the ground to minimize the burden of training the model onboard, which has a low processing capability, therefore ensuring an easy and lightweight model to be loaded onboard. Using the lightweight video scene model onboard the nanosatellite OBC, the proposed approach was able to effectively use the storage and decrease the downlink cost drastically.

The proposed onboard video scene approach can help reduce the manpower and effort in the ground station that would be wasted going through the video files to find a video scene that contains mission-related information, such as a disaster, because the proposed approach only sends the mission-related video frames.

Using the proposed methodology, nanosatellites will soon be able to accommodate a payload camera that can take HD videos and send the video data using low communication channels such as S-band and the ground mission operation can be easy for nanosatellite constellations with video payload data. The study can be further extended by considering various disasters and increasing the complexity by having many scene classification classes.

Author Contributions: Conceptualization, N.A.T. and H.-D.K.; methodology, N.A.T. and H.-D.K.; software, N.A.T.; validation, N.A.T. and H.-D.K.; formal analysis, N.A.T.; investigation, N.A.T. and H.-D.K.; resources, N.A.T.; data curation, N.A.T.; writing—original draft preparation, N.A.T.; writing—review and editing, N.A.T. and H.-D.K.; visualization, N.A.T.; supervision, H.-D.K.; project administration, N.A.T.; funding acquisition, H.-D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea funded by the Ministry of Science and ICT under Grant 2022M1A3C2074536, Future Space Education Center.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Crusan, J.; Galica, C. NASA's CubeSat Launch Initiative: Enabling broad access to space. *Acta Astronaut.* **2019**, *157*, 51–60. [[CrossRef](#)]
2. Villela, T.; Costa, C.A.; Brandão, A.M.; Bueno, F.T.; Leonardi, R. Towards the thousandth CubeSat: A statistical overview. *Int. J. Aerosp. Eng.* **2019**, *2019*, 5063145. [[CrossRef](#)]
3. Ahmadi, S.A.; Ghorbanian, A.; Mohammadzadeh, A. Moving vehicle detection, tracking and traffic parameter estimation from a satellite video: A perspective on a smarter city. *Int. J. Remote Sens.* **2019**, *40*, 8379–8394. [[CrossRef](#)]
4. Yang, T.; Wang, X.; Yao, B.; Li, J.; Zhang, Y.; He, Z.; Duan, W. Small moving vehicle detection in a satellite video of an urban area. *Sensors* **2016**, *16*, 1528. [[CrossRef](#)]
5. Gu, Y.; Wang, T.; Jin, X.; Gao, G. Detection of event of interest for satellite video understanding. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7860–7871. [[CrossRef](#)]
6. Lüdemann, J.; Barnard, A.; Malan, D.F. Sub-pixel image registration on an embedded Nanosatellite Platform. *Acta Astronaut.* **2019**, *161*, 293–303. [[CrossRef](#)]
7. Adams, C.; Spain, A.; Parker, J.; Hevert, M.; Roach, J.; Cotten, D. Towards an integrated GPU accelerated SoC as a flight computer for small satellites. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–7.
8. Lim, S.; Kim, J.-H.; Kim, H.-D. Strategy for on-orbit space object classification using deep learning. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2021**, *235*, 2326–2341. [[CrossRef](#)]
9. de Melo, A.C.C.P.; Café, D.C.; Borges, R.A. Assessing Power Efficiency and Performance in Nanosatellite Onboard Computer for Control Applications. *IEEE J. Miniat. Air Sp. Syst.* **2020**, *1*, 110–116. [[CrossRef](#)]
10. Archiga, A.P.; Michaels, A.J.; Black, J.T. Onboard image processing for small satellites. In Proceedings of the NAECON 2018-IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 23–26 July 2018; pp. 234–240.
11. Xiao, A.; Wang, Z.; Wang, L.; Ren, Y. Super-resolution for “Jilin-1” satellite video imagery via a convolutional network. *Sensors* **2018**, *18*, 1194. [[CrossRef](#)]
12. Jo, Y.; Oh, S.W.; Kang, J.; Kim, S.J. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3224–3232.
13. Wu, J.; He, Z.; Zhuo, L. Video satellite imagery super-resolution via a deep residual network. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 2762–2765.
14. Shao, J.; Du, B.; Wu, C.; Pingkun, Y. Pasiam: Predicting attention inspired siamese network, for space-borne satellite video tracking. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1504–1509.
15. Pflugfelder, R.; Weissenfeld, A.; Wagner, J. On learning vehicle detection in satellite video. *arXiv* **2020**. arXiv 2001.10900.
16. Gu, Y.; Liu, H.; Wang, T.; Li, S.; Gao, G. Deep feature extraction and motion representation for satellite video scene classification. *Sci. China Inf. Sci.* **2020**, *63*, 140307. [[CrossRef](#)]
17. Giuffrida, G.; Diana, L.; de Gioia, F.; Benelli, G.; Meoni, G.; Donati, M.; Fanucci, L. Cloudscout: A deep neural network for on-board cloud detection on hyperspectral images. *Remote Sens.* **2020**, *12*, 2205. [[CrossRef](#)]
18. Lofqvist, M.; Cano, J. Optimizing Data Processing in Space for Object Detection in Satellite Imagery. *arXiv* **2021**. arXiv 2107.03774.
19. Slater, W.S.; Tiwari, N.P.; Lovelly, T.M.; Mee, J.K. Total ionizing dose radiation testing of NVIDIA Jetson nano GPUs. In Proceedings of the 2020 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 22–24 September 2020; pp. 1–3.
20. Cho, D.-H.; Choi, W.-S.; Kim, M.-K.; Kim, J.-H.; Sim, E.; Kim, H.-D. High-resolution image and video CubeSat (HiREV): Development of space technology test platform using a low-cost CubeSat platform. *Int. J. Aerosp. Eng.* **2019**, *2019*, 8916416. [[CrossRef](#)]
21. Kraus, M.; Pollok, T.; Miller, M.; Kilian, T.; Moritz, T.; Schweitzer, D.; Beyerer, J.; Keim, D.; Qu, C.; Jentner, W. Toward mass video data analysis: Interactive and immersive 4D scene reconstruction. *Sensors* **2020**, *20*, 5426. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.