ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
httpa://doi.org/10.31803/tg-20210204161210

Original scientific paper

Tracking Keypoints from Consecutive Video Frames Using CNN Features for Space Applications

Janhavi Borse*, Dipti Patil, Vinod Kumar

Abstract: Hard time constraints in space missions bring in the problem of fast video processing for numerous autonomous tasks. Video processing involves the separation of distinct image frames, fetching image descriptors, applying different machine learning algorithms for object detection, obstacle avoidance, and many more tasks involved in the automatic maneuvering of a spacecraft. These tasks require the most informative descriptions of an image within the time constraints. Tracking these informative points from consecutive image frames is needed in flow estimation applications. Classical algorithms like SIFT and SURF are the milestones in the feature description development. But computational complexity and high time requirements force the critical missions to avoid these techniques to get adopted in real-time processing. Hence a time conservative and less complex pre-trained Convolutional Neural Network (CNN) model is chosen in this paper as a feature descriptor. 7-layer CNN model is designed and implemented with pre-trained VGG model parameters and then these CNN features are used to match the points of interests from consecutive image frames of a lunar descent video. The performance of the system is evaluated based on visual and empirical keypoints matching. The scores of matches between two consecutive images from the video using CNN features are then compared with state-of-the-art algorithms like SIFT and SURF. The results show that CNN features are more reliable and robust in case of time-critical video processing tasks for keypoint tracking applications of space missions.

Keywords: artificial intelligence; convolutional neural network; feature descriptor; machine learning; space missions

1 INTRODUCTION

Although many space missions have been successfully conquered by national and international government bodies, automation in the space related tasks is still developing. Many research challenges related to space exploration are still in their early stage of development. One of the reasons is that space applications are time critical and decision making in constrained time span is really very important. While revolving around a target planet a spacecraft always keeps on taking videos of a scene ahead using on board cameras for study purpose. These real time videos are needed to be processed within time constraints for different purposes. Extracting features useful for further space exploration and navigation tasks is at primary stage. The motion of a spacecraft result in spatially transformed images of the same scene majority of times. Detecting the most informative keypoints from videos in real-time is a challenging task. Further tracking keypoints between two consecutive video frames is a next important task for many flow estimation algorithms. In this paper, tracking keypoints between consecutive video frames is achieved using CNN features. We propose a methodology for keypoints tracking which will be suitable for time critical space applications.

The existing state-of-the-art feature detectors & descriptors [1-8] are efficient enough but are computational expensive for real-time applications. These algorithms are used for detecting and describing the most informative points of interest from an image. Important keypoints are extracted from an image and these keypoints are described in a way to suit an application of interest. Certainly, applications define that how the features must be described and represented. Statistical terms like mean, standard deviation of image intensity and yet any higher moments may serve as an efficient means of feature description. Few functions like energy, entropy or any other complex frequency transforms may also serve as feature descriptors. A keypoint extraction algorithm, Harris Corner detector [9] used the combination of corner points and edge points to describe the features but it fails to describe surfaces or an object as a whole. Scale Invariant Feature Transform (SIFT) [10] algorithm was introduced in 2004, which showed a huge paradigm shift in feature extraction and description. SIFT addressed the challenge of invariance to affine transformations along with being the most efficient descriptor. It used difference of gaussian function for detecting potential keypoints and used im-age gradient magnitude, direction from local neighbourhood for keypoint description. It is prevalently used for object detection and image matching tasks. But it is computation-ally intensive and hence not suitable for time critical appli-cations. Speeded procedure for keypoint extraction is brought up to by Speeded up robust features (SURF) [11, 12], Features from accelerated segment test (FAST) [13, 14], Binary robust independent elementary features (BRIEF) [15] and Oriented FAST and rotated BRIEF (ORB) [16] eventually. Amongst all of these algorithms SURF is found to deliver good quality and also computationally efficient features. ORB algorithm is a combination of FAST & BRIEF and works faster than SURF but its features are not found suitable for image matching tasks. Moreover, theses algorithms are standalone versions and cannot be trained for real time functioning.

With advancements computational resources and data sources few deep learning techniques [17, 2] are also developed recently. These are meant for object detection, recognition and other computer vision tasks. To name a few, deep neural network models like Inception [18], VGG [19], XCeption [20], ResNet [21] are already being implemented and tested on variety of datasets. Few new techniques have been proposed, which use transfer learning by finetuning few parameters of these already built models for completing their tasks.

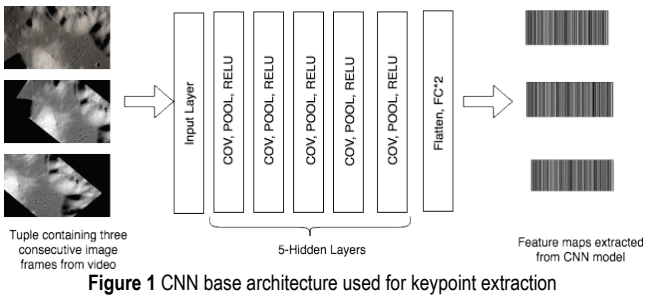


Figure 1 CNN base architecture used for keypoint extraction

Glancing through the literature we arrive at a decision of devising a model that will be suitable for time critical space applications and does not involve any complexity of detection, description of keypoints and further it must be useful for keypoints tracking operation. The solution to this problem can be using a convolutional neural network (CNN) model for feature extraction, which it does seamlessly and autonomously without any overhead of complex computations and seems reliable in real time functioning.

To address the aforementioned problem, we propose a 7-layer CNN model for keypoint extraction from three consecutive video frames. The paper discusses the complete meth-od of feature extraction from the proposed model, the model parameters, and its application to image tracking. Further to test the robustness and reliability of this method for space applications, it is compared with existing SIFT and SURF algorithms.

The organization of this paper is as follows: Section 2 discusses proposed CNN architecture, methodology. Subsection 2.1 explains experimental setup and dataset used. The performance metrics are stated in subsection 2.2. Section 3 discusses results and comparisons with state-of-the-art algorithms and finally Section 4 discusses conclusion.

2 PROPOSED METHODOLOGY

The proposed CNN architecture consists of five convolutional layers, one flattened layer and two fully connected (FC) layers as shown in Fig. 1. First layer being the input layer accepts image frames from input video sequence. The model outputs F number of 2-dimensional feature maps which are further used for feature matching and fed to Nearest Neighbor algorithm. The parameters of the model are set empirically. The flatten layer is used to convert 3-dimensional input to 1-dimensional tensor for faster processing of images. Two fully connected layers are introduced at the end to extract features. The number of computational units in last layer are equal to the feature vector dimensions. There is no need of nonlinear function at the output as the job is to just extract features not the recognition or classification task.

Each input image $I^{i}_{(M,N)}$ is padded to preserve the size of original image to get a padded image, $I^{p}_{(M+p,N+p)}$. An image is convolved (*) with f_i number of filters, each of which size is ($m_i \times n_i$) and then passed through a non-linear functional unit. Then the padded image is passed into the convolution layer to get an output image as,

$$I^{o}_{(M,N)} = MaxPool\left[Re\,LU\left(I^{p}_{(M+p,N+p)} * f_{i(m,n)} \right) \right] \quad (1)$$

Rectifier Linear Unit (ReLU) is used as an activation function for all hidden layers. Max pooling is applied on each successive output after padding. In short, each hidden unit in the model is a combination of convolution layer, a ReLU unit and a pooling layer as in any general case. Detailed procedure for keypoints matching through CNN features is detailed in Fig. (2).

Using proposed methodology shown in figure (2), CNN features can be used to track the keypoints in two consecutive images, which can be further useful in flow detection algorithms. A lunar descent video (credit: https://svs.gsfc.nasa.gov/) is used for experimentation purpose. From the video with known frame rate, images are extracted. To train the model for all kinds of transformations, each image from the dataset is undergone 100 different transformation operations. Such transformed dataset contains both reference image and its transformed versions and is known as augmented dataset D_{aug}. Subsection 2.1 will discuss this in more details. The features extracted from reference image I^{ref} works as reference values for the features extracted from its transformed versions, $I^{\text{trans}} = (I_R^{\text{ref}} + I_T^{\text{ref}} + I_S^{\text{ref}})$. The reference descriptors will be used to perform the matching between the consecutive descriptors of its transformed versions. These features represent the key points of interest from each image.

$$f_{\text{loss}} = \frac{\sum_{\text{ref, aug} \in D_{\text{aug}}} \underset{\boldsymbol{W}}{min} \left\| D_p^{\text{ref}} - D_p^{\text{aug}} \right\|}{D_{\text{aug}}} \quad (2)$$

In (2) $\boldsymbol{W}$ is the weight vector of the model. $\boldsymbol{W}$ is adjusted during each epoch to minimize loss function, f_{loss}. The algorithm starts with initial weights fetching from pretrained model VGG net and further finetuning the base model with these starting weights.

Steps for keypoint matching are described below:

For each image tuple (I^{ref}, I^{1}, I^{2}) in D_{aug} repeat the following steps:

- Supply this tuple to CNN Model to extract keypoints $(K_p^{\text{ref}}, K_p^2, K_p^3)$
- Apply image matching technique to find matched keypoints between two consecutive images,

$$\boldsymbol{K}_p^{\text{matched}} := NearestNeighbour(K_p^{\text{ref}}, K_p^1)$$

- Pass vector $\boldsymbol{K}_p^{\text{matched1}}$ to compute matching score between two images.
- Repeat step 2 and 3 for next augmented image i.e. the pair $(K_p^{\text{ref}}, K_p^2)$ and get $\boldsymbol{K}_p^{\text{matched2}}$

The procedure shown in Fig. 2 is adopted for computing matching score for all test samples through studied

algorithms for unbiased evaluation. Initially features were extracted from reference image and then from its transformed version. An efficient matching algorithm called Fast Library for Approximate Nearest Neighbours (FLANN) is used to match keypoints from both images. Those keypoints are matched whose nearest neighbors from both images have equal contribution in representing that keypoint.

All the key points in the common region of reference image and its transformed image are called correspondences between the two images. After computing the maximum correspondences, the algorithm tries to find the correct matches using some threshold.

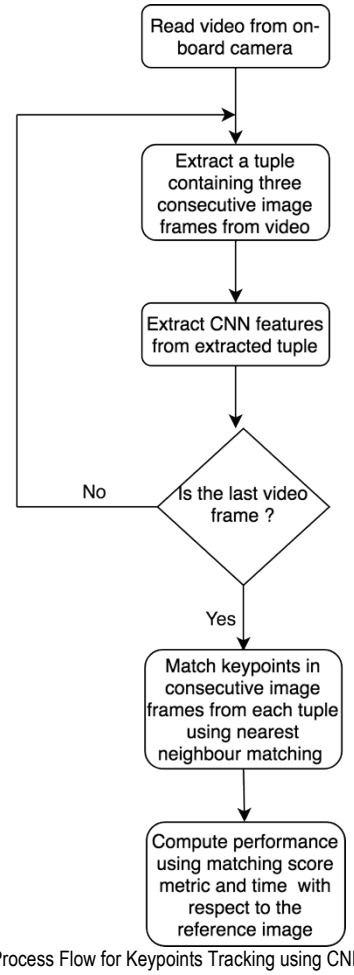


Figure 2 Process Flow for Keypoints Tracking using CNN features

2.1 Experimental Setup

The CNN model was implemented and trained on Intel core i7 processor with 16 GB RAM with NVDIA GTX 1650 graphics card for performance boost while training. The implementation is tested on 2.4 GHz Intel Core i5 processor with 10 GB DDR3 RAM. Programming language Python 3.7 in tensor flow environment is used for implementation of this work.

A video freely available on the website https://svs.gsfc.nasa.gov/ is used for generation of augmented dataset. A python script was written for extracting image frames from a spacecraft landing video. This video is an animated view of landing site of Apollo 17 mission. This video was created by the sources from Lunar Reconnaissance Orbiter (LRO) photographs and elevation maps. Total 915 images were extracted from this video to create a raw image dataset. From raw image dataset few images were selected at random, and 100 known affine transformations were applied to generate an augmented image dataset which contains total 3489 images. Size of each image is 640 × 360 pixels. Sample images from the dataset are shown in Fig. 3. Training with such augmented data will increase the robustness in the decision for feature matching algorithms. The known transformation matrices would help in finding the exact matches. All these efforts are based on the pre assumption that while descending, a satellite captures the same scene with different orientations, scales and very little spatially translated versions of it. So ultimately the maximum number of descriptors must match with its reference image.

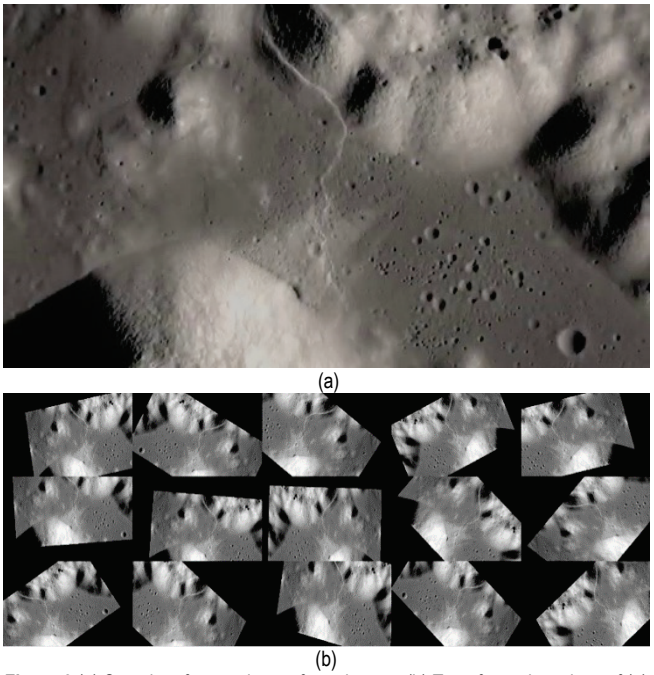


Figure 3 (a) Sample reference image from dataset, (b) Transformed versions of (a) from augmented dataset

For training CNN, padding is used to avoid loss of input dimensions. Stride of 2 is employed to use all parts of image evenly while performing convolution.

The performance of proposed model is evaluated using matching score [3, 6, 22-24], a metric which is widely being used for feature tracking. To check the reliability of our model, it is compared with the state-of-the art algorithms like SIFT [10] and SURF.

2.2 Performance Metrics

Performances of the system is evaluated using matching score [3] and time taken for feature extraction.

Matching score is computed through image correspondences which are matched key points in common regions of two images for which homography is known. It is

the measure of accuracy as it matches descriptors of logically same key points from two different images. Matching score is calculated by (3).

$$Matching\ Score = \frac{C^{+} \bigcap C^{*}}{F_{\text{ref}}} \tag{3}$$

Where, C^{+} is maximum image correspondences, C^{*} is number of correct matches and F_{ref} being the number of descriptors of reference image.

3 RESULTS & DISCUSSION

3.1 Keypoints Tracking using CNN features

Tab- 1 shows matched keypoints between reference and transformed image frames. The first column contains few transformed sample images from the augmented database. Descriptors of all these test samples are compared with reference image shown in Fig. 3(a). The second column contains the images showing visualizations of this matching. Only initial 10 matching points are shown inside the image pairs. Last column shows the corresponding matching score computed through CNN descriptors. It can be observed that, visualizations show the matched keypoints between the two images.

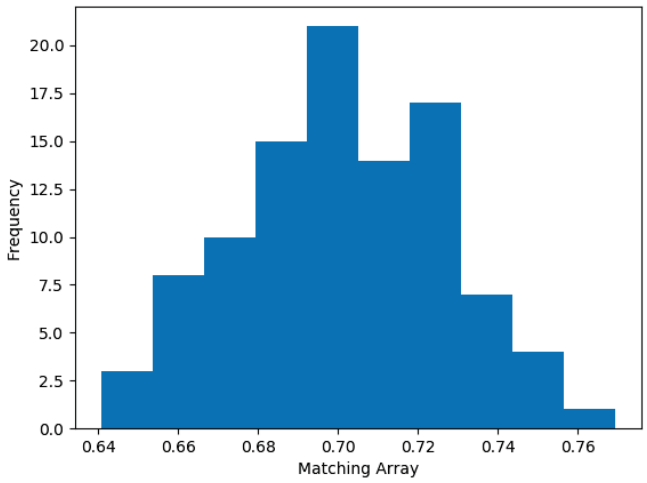
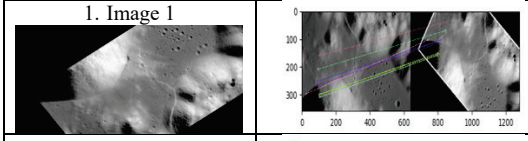
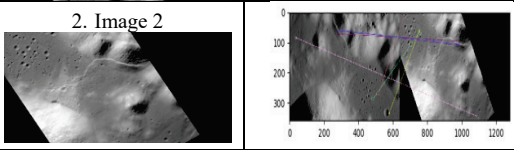
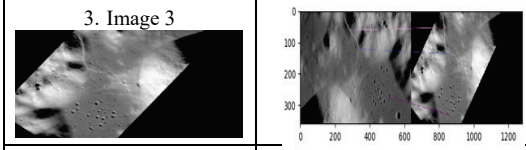
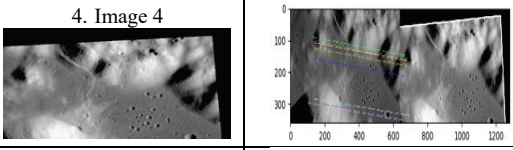
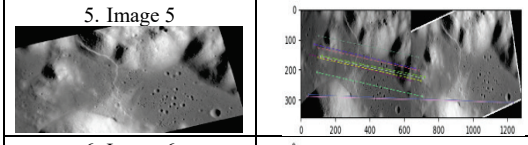
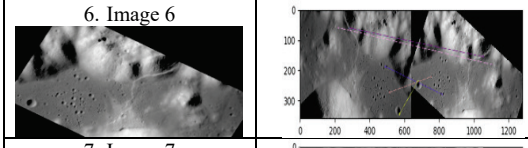
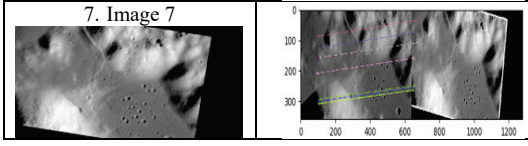


Figure 4 Frequency Histogram of Matching Score

Matching score coveys the correct matches out of total matched keypoints. It is the measure of accuracy in the image tracking algorithms. As all the test samples are spatially transformed versions of the same reference image, ideally matching score should be on the higher side i.e. nearer to 1. Image 5 in the Tab. 1 shows the highest matching score, while Image 3 has the lowest matching score. In image 5, the keypoints below $y \leq 300$ are incorrectly matched, but rest shown points are accurately matched. It can be observed that the test image is slightly ~30° rotated version of reference image and hence maximum scene of reference image is present inside the test image. Estimated matching score of 0.75 guarantees that. On the other hand, image 3 is almost ~180° vertical flip along with quite a large translated version of reference image. So, the chances of finding match between the two are lesser than the previous case. This is explained by its estimated matching score of 0.6582. Rest all images and their visual matching with reference image are self - explanatory.

Fig. 4 shows the frequency distribution of matching score obtained by applying matching algorithm on the 100 test samples. Maximum samples are found in the range of 0.69 to 0.71.

Table 1 Keypoints Tracking using CNN Features
(Note: For better visualization please zoom in the images)

Input Test Image	Visual Matching between reference image and test image	Matching Score
1. Image 1		0.7207
2. Image 2		0.6953
3. Image 3		0.6582
4. Image 4		0.7246
5. Image 5		0.7500
6. Image 6		0.6992
7. Image 7		0.6895

3.2 Comparison with State-of-the-Art Methods

Matching score of the CNN model is compared with SIFT and SURF descriptors. Tab. 2 shows the range and variation of matching score obtained from SIFT, SURF and CNN descriptors. The average value of matching score is highest equal to 0.7008 for CNN descriptors and lowest i.e. 0.3079 for SURF descriptors. SIFT value is 0.4744. As the test images are transformed versions of reference image, the visual matching between two images should be more than 0.50. It can be seen that only CNN descriptors are able to reach that visual threshold. CNN descriptors are showing

visual perfection in computing matching score as compared with the other two descriptors. The variations in the readings are least and almost negligible in case of CNN descriptors. On the contrary, there are much variation is shown in the performance of SIFT and SURF descriptors. This shows robust performance of CNN model for test samples. Tab. 2 also shows the maximum and minimum values of matching score for all the three algorithms. The minimum values shown by SIFT and SURF do not match with visual matching context. The minimum value computed by CNN model is also more that our visual threshold. Hence CNN model seems to be more reliable than other two models for our problem. These statistics are visually represented in Fig. 5. These visual representations would further clarify the discussions in the previous section.

Table 2 Keypoints Matching Score Statistics from Lunar Descent Video Frames for Different Algorithms

Algorithms	SIFT	SURF	CNN
Mean Value	0.4744	0.3079	0.7008
Max Value	0.6357	0.5663	0.7695
Min Value	0.0145	0.0156	0.6406
Variance	0.0079	0.0119	0.0006

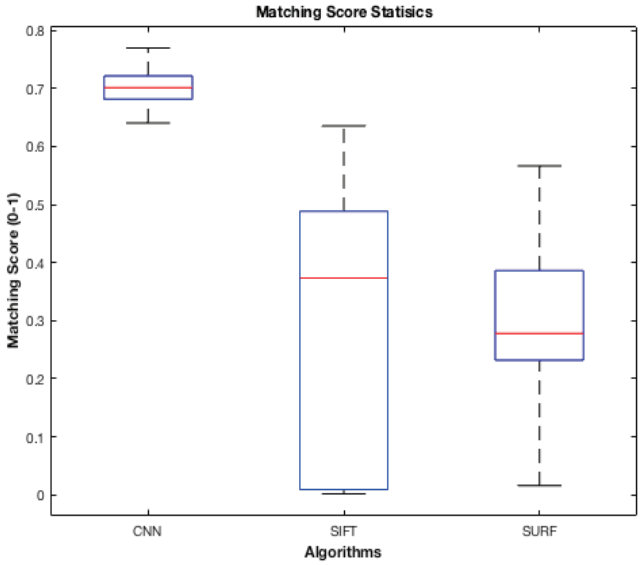


Figure 5 Statistical analysis of matching score obtained from conventional algorithms and CNN model

The visual comparison of matching score computed by three algorithms is shown in Tab. 3. As per human perception of visual context, image 1 is obtained by horizontal flipping and 30° rotation of reference image (Fig. 3(a)). So maximum context of the original image is saved in the test image. The matching score should reflect that visual matching. CNN descriptors show maximum matching as shown in table. The same interpretation is suitable for image 4 and image 5, where CNN model outperforms. Image 2, 6, and 7 are spatially rotated and translated versions of reference image and hence the less area is matched.

This is contributed by decrease in matching score of all the descriptors. But here also as per visual perception, it should be more that threshold 0.50 which is shown by CNN model. For image 7 performance of SIFT and SURF is also good almost near threshold but for image 2 it is quite low. For image 6, SIFT and SURF failed to find correct matching between reference and test image. But CNN shows a robust performance. The same interpretation is for image 3.

Table 3 Visual Comparison of Matching Score Based on SIFT, SURF and CNN Descriptors
(Note: For better visualization please zoom in the images)

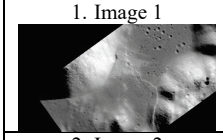
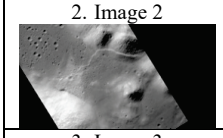
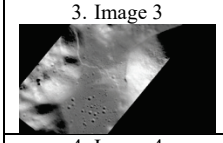
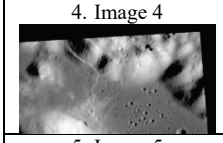
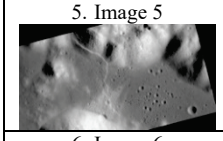
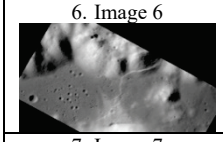
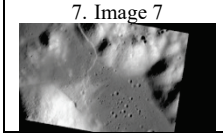
Input Test Image	Matching Score by		
	SIFT descriptors	SURF descritpors	CNN descritpors
1. Image 1 	0.5282	0.2838	0.7207
2. Image 2 	0.2185	0.1823	0.6953
3. Image 3 	0.1473	0.0982	0.6582
4. Image 4 	0.5318	0.4564	0.7246
5. Image 5 	0.5573	0.4322	0.7500
6. Image 6 	0.0982	0.0543	0.6992
7. Image 7 	0.5774	0.4721	0.6895

Table 4 Processing Time of SIFT, SURF and CNN MODELS

Evaluation Criterion	SIFT	SURF	CNN
Average Time (seconds)	8.1573	5.3321	**0.0348**

Tab. 4 shows the average processing time enquired for all the algorithms. Processing time is computed by adding descriptor extraction time and feature matching time. As CNN is trained model, it requires very less time for processing of a single test image which is only 34.8 milliseconds. The highest processing time is needed for SIFT descriptors as it requires lots of computations. As our problem requires real time processing of space videos, CNN seems more convenient and reliable model as far as time constraints of space applications are considered.

Overall performance of CNN model is more reliable and robust as compared with state-of-the-art methods and hence it seems more suitable for real time video processing of space missions.

4 CONCLUSION

Real time space mission tasks are time critical and hence for such tasks processing time plays an important parameter of evaluation. Keypoints, which are special points of interest inside an image must be tracked between consecutive image frames of a real time video captured by on board spacecraft cameras. It is useful for many flow detection algorithms and other space applications. In this paper a methodology using CNN descriptors is proposed for such time critical applications. A new 7-layer CNN model is developed and features thus extracted are used for keypoints matching between consecutive image frames of a lunar descent video. The CNN is trained using pre-trained VGG model parameters and fine-tuned for the new data. Total 100 test image samples for a single reference image were used for evaluating the performance of the CNN model. It is observed that, CNN descriptors are time efficient, robust and hence reliable for image tracking applications. Statistical analysis of matching score shows less variations in the CNN descriptors and is ideal for real time performance.

Notice

This paper was presented at IC2ST-2021 – International Conference on Convergence of Smart Technologies. This conference was organized in Pune, India by Aspire Research Foundation, January 9-10, 2021. The paper will not be published anywhere else.

5 REFERENCES

[1] Awad, A. I. & Hassaballah, M. (2016). *Image Feature Detectors and Descriptors: Foundations and Applications*. Studies in Computational Intelligence 630, Springer International Publishing. https://doi.org/10.1007/978-3-319-28854-3

[2] Fischer, P., Dosovitskiy, A., & Brox, T. (2014). Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT. *ArXiv*, abs/1405.5769.

[3] Boyraz, P. & Bayraktar, E. (2017). Analysis of Feature Detector and Descriptor Combinations with a Localization Experiment for Various Performance Metrics. *Turkish Journal of Electrical Engineering and Computer Sciences, 25*(3), 2444-2454. https://doi.org/10.3906/elk-1602-225

[4] Lenc, K. & Vedaldi, A. (2018). Large scale evaluation of local image feature detectors on homography datasets. https://arxiv.org/pdf/1807.07939.pdf

[5] Li, S. (2017). A review of feature detection and match algorithms for localization and mapping. *IOP Conference Series: Materials Science and Engineering, 231*, 012003. https://doi.org/10.1088/1757-899X/231/1/012003

[6] Mikolajczyk, K., Tuytelaars, T., Schmid, C. et al. (2005). A Comparison of Affine Region Detectors. *Int J Comput Vision, 65*(1-2), 43-72. https://doi.org/10.1007/s11263-005-3848-x

[7] Moura, G. M. & Silva, R. L. S. (2017). Analysis and Evaluation of Feature Detection and Tracking Techniques using OpenCV with Focus on Markerless Augmented Reality Applications. *Journal of Mobile Multimedia, 12*(3-4), 291-302. https://doi.org/10.26421/JMM12.3-4

[8] Salahat, E. & Qasaimeh, M. (2017). Recent advances in features extraction and description algorithms: A comprehensive survey. *2017 IEEE International Conference on Industrial Technology (ICIT)*, Toronto, ON, 1059-1063, https://doi.org/10.1109/ICIT.2017.7915508

[9] Harris, C. & Stephens, M. (1988). A Combined Corner and Edge Detector. In C. J. Taylor, editors, *Proceedings of the Alvey Vision Conference*, 23.1-23.6. Alvey Vision Club. https://doi.org/10.5244/C.2.23

[10] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision, 60*, 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[11] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. *Proceedings, Part I, 9th European Conference on Computer Vision - ECCV 2006*, Graz, Austria, 404-417. https://people.ee.ethz.ch/~surf/eccv06.pdf

[12] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding, 110*(3), 346-359. https://doi.org/10.1016/j.cviu.2007.09.014

[13] Rosten, E. & Drummond, T. (2006) Machine Learning for High-Speed Corner Detection. In: Leonardis A., Bischof H., Pinz A. (eds) *Computer Vision – ECCV 2006. Lecture Notes in Computer Science, vol 3951*, Springer, Berlin, Heidelberg, 430-443. https://doi.org/10.1007/11744023_34

[14] Rosten, E., Porter, R., & Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell., 32*(1), 105-119. https://doi.org/10.1109/TPAMI.2008.275

[15] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *Proc. IEEE Int. Conf. Comput. Vis.*, 2548-2555. https://doi.org/10.1109/ICCV.2011.6126542

[16] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proc. IEEE Int. Conf. Comput. Vis.*, 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[17] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234-241.

[18] Szegedy, C. et al. (2015). Going deeper with convolutions. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[19] Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 1-14.

[20] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, 1800-1807. https://doi.org/10.1109/CVPR.2017.195

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[22] Ehsan, S., Kanwal, N., Clark, A. F., & McDonald-Maier, K. D. (2010). Improved repeatability measures for evaluating performance of feature detectors. *Electron. Lett., 46*(14), 998-1000. https://doi.org/10.1049/el.2010.1442

[23] Mouats, T., Aouf, N., Nam, D., & Vidas, S. (2018). Performance Evaluation of Feature Detectors and Descriptors beyond the Visible. *Journal of Intelligent & Robotic Systems, 92*, 33-63. https://doi.org/10.1007/s10846-017-0762-8

[24] Schönberger, J., Hardmeier, H., Sattler, T., & Pollefeys, M. (2017). Comparative Evaluation of Hand-Crafted and Learned

Local Features. *Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, HI, 6959-6968. https://doi.org/10.1109/CVPR.2017.736

Authors' contacts:

Janhavi H. Borse, Assistant Professor
(Corresponding author)
Sandip Institute of Technology & Research Centre,
Affiliated to Savitribai Phule Pune University,
Ganeshkhind Road, Pune, Maharashtra 411007, India
Tel.: 9225367554, E-mail: borse.janhavi@gmail.com

Dipti D. Patil, Associate Professor
MKSSS's Cummins College of Engineering for Women,
Karve Nagar, Pune, Maharashtra 411052, India
E-mail: diptivt@gmail.com

Vinod Kumar, Division Head - Control Dynamics Design Group
U R Rao Satellite Centre,
Old Airport Road, Vimanapura, Bangalore, Karnataka 560017, India
E-mail: vinodkkaushik@gmail.com