ON-BOARD GCPS MATCHING WITH IMPROVED TRIPLET LOSS FUNCTION

Guangqi Xie [1], Zhiqi Zhang [1, 2], Ying Zhu [1], Shao Xiang [1], Mi Wang [1, *]

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China. - (xiegqrs, zzq540, yzhu1003, xshao15, wangmi)@whu.edu.cn
[2] School of Computer Science, Hubei University of Technology, Wuhan 430079, China.

Commission II, WG II/1

KEY WORDS: On-board, GCPs Matching, Intelligent Remote Sensing Satellite, Improved Triplet Loss Function

ABSTRACT:

Intelligent remote sensing satellite system is an important direction to solve the problem of intelligent processing on-board. It can realize the real-time on-board intelligent processing of important targets. The accuracy of geometric positioning information is the basis for subsequent intelligent processing. Therefore, this paper corrects the positioning information by GCPs (Ground Control Points) matching on-board. Considering the limited storage and computing performance of satellites, this paper designs a lightweight GCPs deep feature extraction convolutional neural network based on MobileNetV2 as feature extraction model, and trains this network with an improved triplet loss function. The Songshan calibration field images constructed by Wuhan University was used as the GCPs image, and 30,399 image patches were extracted and embedded as GCPs feature library. The size of the GCPs library is a size of 15.3M, and size of the lightweight depth feature extraction model is 9.83M, which can be pre-stored on the satellite for positioning with GCPs matching on-board. In addition, this paper tested feature extraction performance on an embedded device Nvidia Jeston Xavier which simulates the performance of the device on the satellite. At Xavier 30W max power consumption model, a single frame takes 0.005 seconds, and under Xavier 15W power consumption model, a single frame takes 0.009 seconds. At 10W power consumption model, a single frame takes 0.018 seconds, which can meet the performance requirements on the satellite. In addition, the experiments in this paper show that the positioning accuracy is within 30 meters. The work done in this paper will be experimented on the Luojia-3-01 intelligent remote sensing satellite.

1. INTRODUCTION

Dramatic increase in satellite data not only provides a rich source for subsequent processing and services, but also puts pressure on satellite-ground data transmission links and ground processing and storage systems. Especially in the field of high-efficiency applications, the images taken on-board cannot be provided to users in real-time. Intelligent satellites can extract and distribute effective information on-board in real-time. Therefore, there is an urgent need to research the processing technology on-board.

Satellite	Country	Time	Processing On-board	Processor
EO-1	United States	2000	Detection	Mongoose V
BIRD	Germany	2001	Pre-processing	DSP/FPGA
NEMO	United States	2003	Compression	DSP
X-SAT	Singapore	2006	Reject invalid data	FPGA/ StrongARM
Pleiades-1/2	France	2011/2012	Pre-processing	FPGA

Table 1. Application of processing on-board of remote sensing satellites

Since the 1990s, intelligent remote sensing satellite on-board processing technology has been researched by researchers. Table 1 shows the processing on-board in recent years(Hayden et al., 2004; Straight et al., 2010). DSP (Digital Signal Processing) and FPGA (Field Programmable Gate Array) were the main processor in these satellites. However, with the rapid development of software and hardware in recent years, the ARM + GPU structure has gradually been tried for processing on-board. The upcoming launch of Luojia-3-01 satellite (Wang Mi, 2019), jointly designed by DFH Satellite Co. and Wuhan University, will support this mode. Compared with FPGA and DPS, ARM + GPU mode has better portability and developability. It can easily transplant ground processing algorithms to satellite processing. However, limitations of storage space and performance are always bottleneck of on-board processing. Therefore, remote sensing image processing algorithms need to be developed for this new architecture. The control point matching, as the basis of the subsequent high-processing, needs more in-depth research.

2. RELATED WORKS

Since AlexNet (Krizhevsky et al., 2012) has achieved great success in the field of image processing using deep convolutional networks, deep learning methods have been widely used in the field of image processing. It has also gained better application in remote sensing image processing (Ma et al., 2019), such as the classification(Cai et al., 2018; Gong et al., 2017; Hamida et al., 2018), object detection(Dong et al., 2019; Vetrivel et al., 2018; Yu et al., 2016) and segmentation(Kemker et al., 2018; Zhang et al., 2018). Siamese structure was used for two sets of patches matching (Zagoruyko and Komodakis, 2015) by train the distance between matched and no-matched pairs respectively, which provided a new idea for image matching using deep learning. The triplet loss function (Schroff et al., 2015) trains the matched and no-matched pairs at once for face recognition. The Triplet loss function was improved by studies (Chen et al., 2017; Cheng et al., 2016) for better performance, which shows that the triplet structure has good scalability. Compare with triplet structure, Siamese structure works to a secondary objective: drive the distance between matched pair as close to 0 as possible (Vo and Hays, 2016). For remote sensing images, images from different locations may cover the same area. Such images are likely to be considered to come from the same location. The second objective will be useful for GCPs matching. Therefore, this paper proposes a new improved triplet loss function for remote sensing image matching on-board combining the advantages of Siamese structure and triplet structure.

* Corresponding author

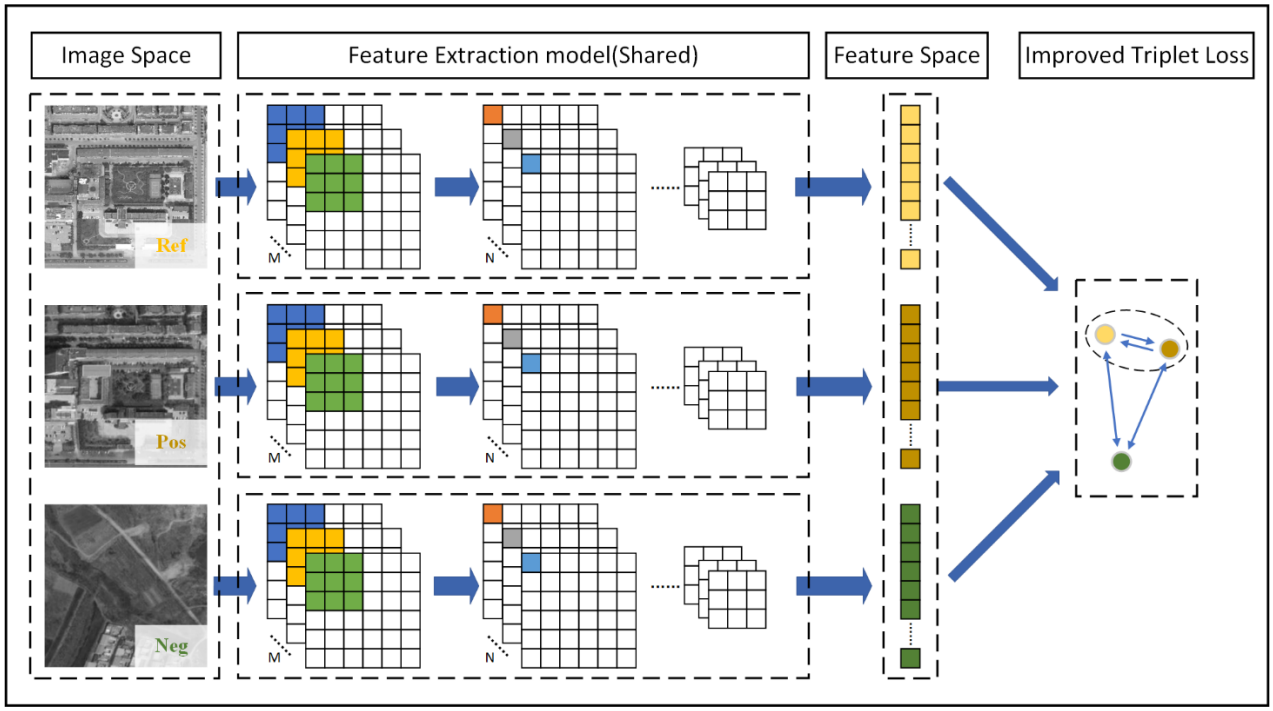

Figure 1. Improved triplet training framework.

In addition, with the deepening of the layers of the deep neural network, the space and performance requirements of the model have gradually increased, which caused great difficulties for embedded devices and mobile devices. For example, devices on intelligent satellite cannot meet the needs of most deep networks. Therefore, many lightweight networks have been proposed for embedded devices with limited performance (Howard et al., 2019; Howard et al., 2017; Ma et al., 2018; Sandler et al., 2018; Zhang et al., 2017). MobileNet is one of the better performers in lightweight deep convolutional networks.

In summary, it is difficult to store the traditional GCPs library and match with complex deep convolutional networks on-board due to the limited performance and storage space of equipment on-board. Therefore, this paper used a lightweight feature extraction model with improved triplet loss function to embed GCPs image patches into D-dimensional space as GCPs library to store on-board, and extracted feature of image taken on-board by this model to match with GCPs library for positioning. The work will be experimented on the Luojia-3-01 intelligent remote sensing satellite.

3. METHODS

In this section, the method of this article is introduced. The first part is the overall framework, the second part is the lightweight feature extraction network, the third part is the improved triplet loss function, and the fourth part is the on-board positioning for satellite based on GCPs matching.

3.1 The Overall Framework

As shown in the Figure 1, in general, the image patches were transformed into the feature space after being extracted by the shared feature extraction model. The feature extraction model optimized by the improved triplet loss function. Namely, this paper strives for an embedding $f(x)$, from an image x into a feature space $\mathbb{R}^d$, such that the squared distance between all GCPs picture and target picture, independent of imaging conditions, of same position is small, whereas the squared distance between a pair of GCPs images from different position is large. The images were embedded into feature space as the GCPs library.

Ref (reference, GCPs images) and Pos (positive, images from different sources at the same location as the GCPs) in image space are images of different sources at corresponding positions, and Neg (negative, images in different locations) is an image of different positions. Ref and Pos form positive pairs. Ref and Neg form negative pairs. This paper also defines second negative pairs which formed by Pos and Neg.

The optimization of the traditional triplet loss function is to make the feature distance of positive pairs smaller than the feature distance of negative pairs. The purpose of the improved triplet loss function optimization is to make the feature distance of positive pairs approach 0, and make the feature distance of negative pairs, second negative pairs larger than the feature distance of positive pairs. Because positive pairs are images from different sources at the same location, their two features should be as similar as possible, and negative pairs belong to different locations, their features should be different.

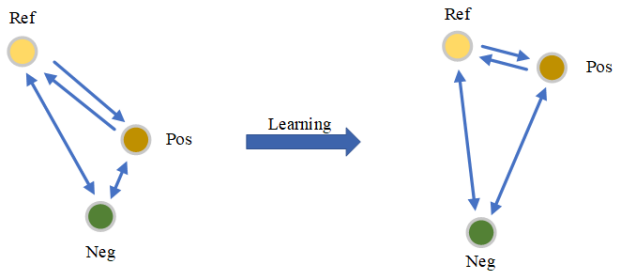

Figure 2. The improved triplet loss resembles the image features of same location, making the image features of different location tend to be different

For feature extraction networks, different deep convolutional neural networks can be used. Considering on-board performance limitations, this paper used a lightweight deep convolutional network MobileNetV2 (Sandler et al., 2018) with Inverted Residuals and Depthwise Separable Convolutions structures.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-105-2020 | © Authors 2020. CC BY 4.0 License.

In conclusion, as shown in Figure 2, after learning with the improved triplet loss function, the image features extracted by the shared network at the same location will tend to be similar, and the image features at different locations will tend to be different.

3.2 Lightweight Feature Extraction Network

Traditional FPGA and DSP processing cores are difficult to meet the needs of on-board intelligent processing. In recent years, embedded devices based on ARM + GPU structures have been tried for on-board intelligent processing, such as the upcoming launch of Luojia-3-01, which has on-board processing capabilities with ARM + GPU structures. Deep convolutional networks have proven to be powerful in the field of image processing, but they consume too much computing resources. There are also rich requirements in the embedded mobile terminal, so many lightweight frameworks (Howard et al., 2019; Howard et al., 2017; Ma et al., 2018; Sandler et al., 2018; Zhang et al., 2017) have been proposed, and MobileNets(Howard et al., 2019; Howard et al., 2017; Sandler et al., 2018) is one of the best networks among them.

In this paper, MobileNetV2(Sandler et al., 2018) was selected as the basic feature extraction network. The key to MobileNetV2 lightweighting is the depth separable convolution. It is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1 * 1 convolution called a pointwise convolution (Howard et al., 2017) (feature extraction model in Figure 1). In addition, it also adds the idea of Resnet residuals(He et al., 2016; Xie et al., 2016), which is called inverted residuals (Sandler et al., 2018), to improve accuracy. Therefore, it can also ensure that the accuracy can meet the requirements in the process of lightweighting.

In order to prevent overfitting, weaken the unimportant feature variables, and extract important feature variables, this paper adds the ReLU6 and L2 regularization layers before the output of MobileNetV2.

3.3 Improved Triplet Loss Function and Training on Ground

Triplet loss function was proposed in FaceNet (Schroff et al., 2015) and achieved good results in face recognition. It can also be used in image matching (Vo and Hays, 2016). The embedding is represented by $f(x) \in \mathbb{R}^d$. It embeds an image x into a d-dimensional hypersphere Euclidean space. Its expression is as follows:

$$\mathrm{L} = \sum_i^N \left[\max(\left\|f(x_i^R) - f(x_i^P)\right\|_2^2 - \left\|f(x_i^R) - f(x_i^N)\right\|_2^2 + \alpha, 0)\right] \quad (1)$$

Where x_i^R (Ref) is the image of GCPs, x_i^P (Pos) is the image from different sources at the same location as the GCPs, x_i^N (Neg) is image in different locations, α is a margin that is enforced between positive and negative pairs.

This will make,

$$\left\|f(x_i^R) - f(x_i^P)\right\|_2^2 + \alpha < \left\|f(x_i^R) - f(x_i^N)\right\|_2^2 \quad (2)$$

Here the image x_i^R is closer to the image x_i^P from same location than any image x_i^N from different location.

However, for remote sensing images, images from different locations may cover the same area. Such images are likely to be considered to come from the same location. Therefore, it is necessary to judge the most matching among these images. This paper added an item to the Triplet loss function to minimize the distance between x_i^R and x_i^P, and make it approach 0. Its expression will be as follows:

$$\mathrm{L} = \sum_i^N \left[\max(\left\|f(x_i^R) - f(x_i^P)\right\|_2^2 - \left\|f(x_i^R) - f(x_i^N)\right\|_2^2 + \alpha, 0) + \beta\left\|f(x_i^R) - f(x_i^P)\right\|_2^2\right] \quad (3)$$

Where β is a margin that to adjust the minimum similarity rate. At the same time, in order to make the x_i^N father away from the x_i^R and x_i^P, this paper also added an item in the formula to make the distance between the x_i^P and x_i^N lager than the distance between the x_i^R and x_i^N. This is visualized in Figure 2. Finally, the improved triplet loss function will look like the follows:

$$\mathrm{L} = \sum_i^N \left[\max\left(\left\|f(x_i^R) - f(x_i^P)\right\|_2^2 - \left\|f(x_i^R) - f(x_i^N)\right\|_2^2 + \alpha, 0\right) + \max\left(\left\|f(x_i^R) - f(x_i^P)\right\|_2^2 - \left\|f(x_i^P) - f(x_i^N)\right\|_2^2 + \alpha, 0\right) + \beta\left\|f(x_i^R) - f(x_i^P)\right\|_2^2\right] \quad (4)$$

In this paper, $\alpha = 0.5$, and $\beta = 0.4$.

3.4 On-board Positioning Based on GCPs Matching

The GCPs image patches can be embedded into a d-dimensional hypersphere Euclidean space as GCPs library, which can save a lot of hard disk space on the satellite. The d-dimensional depth GCPs library and lightweight feature extraction model will be stored on the satellite. For the image to be matched, the depth features of the region need to be extracted and compared with the GCPs to match. The specific calculation flowchart is shown in Figure 3. Where $CP(x_\tau, y_\tau)$ are all GCPs in the image area and S_n is the step for different epochs.

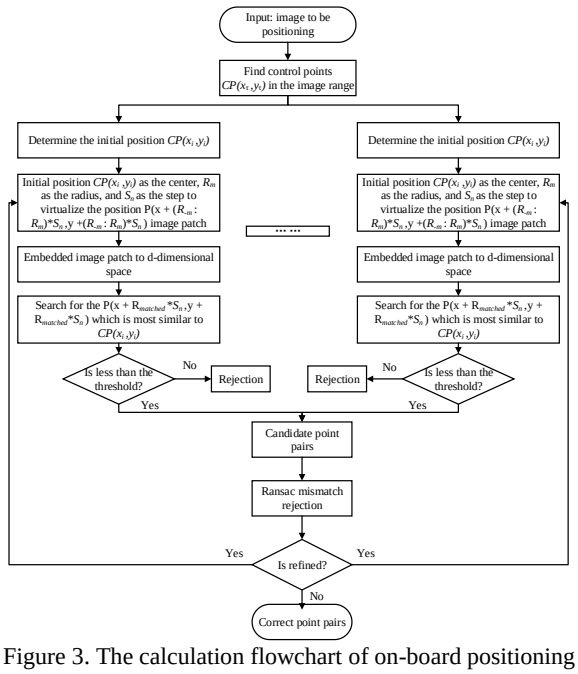

Figure 3. The calculation flowchart of on-board positioning based on GCPs matching.

We also describe the specific algorithm with the pseudo code in Table 2. Where $\mathbb{R}_{\text{search}}$ is the group of searched range for different epochs and $P(x, y)$ is the image patch with in $\mathbb{R}_{\text{search}}$. The forth step in the Table 2 is to normalize the direction and scale to the GCPs image patch. In order to balance search range and search accuracy, multiple epochs can be performed at

different steps and search ranges. And, $S_n = \max(\mathbb{R}_{n+1}) * S_{n+1}$. For example, a total of three epochs are calculated in this paper. First epoch, $\mathbb{R}_1 \in \{-4{:}4\}$, $S_1 = 50m$; second epoch, $\mathbb{R}_2 \in \{-5{:}5\}$, $S_2 = 10m$; third epoch, $\mathbb{R}_3 \in \{-10{:}10\}$, $S_3 = 1m$.

Algorithm1	On-board positioning based on GCPs matching
Input:	Image to be positioned
Output:	Image was positioned
1.	Find GCPs $CP(x_\tau, y_\tau)$ in the image area
2.	Initialize $\mathbb{R}_{search}$ as the search range and S_n as the step
3.	**For** $CP(x_i, y_i)$ **in** $CP(x_\tau, y_\tau)$:
4.	Virtualize the $P(x + (R_j\ in\ \mathbb{R}_{search}) * S_n, y + (R_j\ in\ \mathbb{R}_{search}) * S_n)$ image patch;
5.	Embedded image patch to d-dimensional space;
6.	Search for minimum Dist($P(x_{matched}, y_{matched}), CP(x_i, y_i)$);
7.	**If** Dist($P(x_{matched}, y_{matched}), CP(x_i, y_i)) > Threshold$:
8.	Reject $P(x_{matched}, y_{matched})$;
9.	**Else**:
10.	CandidatePointPairs.append($P(x_{matched}, y_{matched}), CP(x_i, y_i)$)
11.	**End for**
12.	CandidatePointPairs.Ransac()
13.	**If** IsRefined = = True:
14.	Initialize $CP(x_\tau, y_\tau)$ as CandidatePointPairs. $CP(x_i, y_i)$
15.	Goto 2.
16.	CorrectPointPairs = CandidatePointPairs
17.	Image positioned by CorrectPointPairs

Table 2 On-board positioning based on GCPs matching

4. EXPERIMENTS

4.1 Experiments Data

The positioning accuracy of GCPs image has an important influence on the results of GCPs matching. This paper selects China (Songshan) satellite remote sensing calibration field data as the GCPs image. This test field was jointly constructed by China Resources Satellite Application Center, Wuhan University and The PLA Information Engineering University. The area of test field is 9000 square kilometres, located between Zhengzhou and Luoyang, with an east-west length 105 KM and a north-south 80 KM. Its high-precision DOM (Digital Orthophoto Map) and DEM (Digital Elevation Model) were produced using aerial photogrammetry through more than 400 uniformly distributed high-precision GCPs. The ground resolution of the DOM is 0.4 meters, and the ratio of the DEM is 1:5000. The DOM data of the Songshan test site is shown in the Figure 4.

In this paper, Google (https://www.google.com/) and Arcgis (https://www.arcgis.com/) images in the same area are used as training, testing, and validation data, respectively. In order to facilitate on-satellite matching, all images in this paper are panchromatic images.

4.2 Feature Extraction Experiment with Improved Triplet Loss Function

Depth feature extraction model is the basis for subsequent localization through matching, and its accuracy will directly affect the positioning accuracy. The depth feature extraction will embed GCPs image patches into the d-dimensional space as GCPs library, and the library will be stored on-board to meet the storage limit.

As shown in the Figure 4, this paper extracts evenly distributed SIFT feature points on the DOM as GCPs, and takes each GCPs as the center, and crops a 255 * 255 image as the GCPs image patch. Google images were selected as training images, and Arcgis images were used as test and verification images. After removing the image patch with obvious changes, 30,399 image patches can finally be extracted. The Google image and Arcgis image were also extracted as 30,399 image patches in the same position, and the orientation and sale of these image patches were normalized to GCPs image based on projection information. Each GCPs image patch and Google or Arcgis image patch at same position was set as a positive pair which their depth feature needs to be similar. In contrast, a negative pair consists of a GCPs image patch and the least similarity image patch in other position, and a second negative pair formed by the image patches from positive and negative pair except the GCPs image patch.

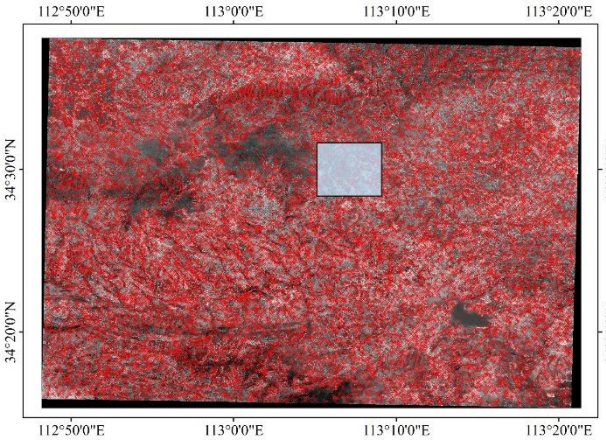

Figure 4. China (Songshan) satellite remote sensing calibration field. The red cross in the picture is the center of the selected GCPs image. The light blue box is the matching experiment area, which is located in the verification area.

Among them, the pairs consist of GCPs and Google image patch were constituted training data set, GCPs and Arcgis image patches are divided into test and verification dataset by 113.08E. West 113.08E is the test data set, and east 113.08E is the verification data set. The light blue box in the Figure 4 is the matching experiment area, which is located in the verification area. In this paper, a GCPs image patch was set as 255 * 255, and d-dimensional was set as 128-dimensional because of its best accuracy and efficiency (Schroff et al., 2015). Therefore, it can compress almost 500 times for each GCPs image.

Dataset was trained on Nvidia RTX 2080TI, and the size of feature extraction model is 9.83M. This model was used to embedded 30,399 GCPs image patches into 128-dimensional feature space as GCPs library with a size of 15.3M. The model and GCPs feature space were used in the subsequent experiments of on-board positioning by GCPs matching.

4.3 On-board Positioning Experiment Based on GCPs Matching

The pre-trained feature extraction model and the GCPs feature library extracted by the it can be pre-loaded on the satellite before launching or transmitted via the satellite-to-ground transmission link. The image captured on the satellite will embed into d-dimensional space through the feature extraction model. The d-dimensional feature will match in the GCPs feature library. The matched point will be used for further positioning.

The light blue box in Figure 4 is the matching experiment area, which is located in the verification area. The Arcgis image in this area has neither participated in training nor testing, and can not only be used to evaluate the positioning accuracy, but also to evaluate the generalization ability of the model.

The GCPs, Google and Arcgis image have projection information because they are both orthoimages. They have accurate positions without positioning. Therefore, this paper shifts the Google and Arcgis image with a (dx, dy) offset respectively, and then uses Algorithm 1 to experimentally match and positioning.

The experiments on the on-board simulation devices Nvidia Xavier, Nvidia RTX 2060 and Nvidia RTX 2080TI. The offsets

were set as (163m, 152m), (175m,152m) and (175m,175m). The images to be tested were Google and Arcgis experiment area images which was shown as the light blue box in Figure 4.

5. RESULTS AND DISCUSSIONS

5.1 Accuracy of Feature Extraction with Improved Triplet Loss Function

This section discusses and analyzes the accuracy of training with improved triplet loss function. It can be seen from the data in Table 3 that the results of test and validation are consistent with training results, which shows that the trained model has good generalization ability. The image patches for each GCPs are trained, although the newly captured image patches are not trained, such as the test and validation dataset which have the same accuracy. This just like face recognition. If there is no face data in the training dataset, then it can never be recognized.

Dataset	Metrics	Accuracy
Train	$\mathrm{Dist(Ref, Pos)} < \mathrm{Dist(Ref, Neg)}$	0.973
	$\mathrm{Dist(Ref, Pos)} < 0.7$	0.938
	$\mathrm{Dist(Ref, Neg)} > 0.7$	0.900
	$\mathrm{Dist(Ref, Pos)} < 0.7$ and $\mathrm{Dist(Ref, Neg)} > 0.7$	0.844
	Mean(Dist(Ref, Pos))	0.373
	Mean(Dist(Ref, Neg))	1.262
Test	$\mathrm{Dist(Ref, Pos)} < \mathrm{Dist(Ref, Neg)}$	0.971
	$\mathrm{Dist(Ref, Pos)} < 0.7$	0.929
	$\mathrm{Dist(Ref, Neg)} > 0.7$	0.903
	$\mathrm{Dist(Ref, Pos)} < 0.7$ and $\mathrm{Dist(Ref, Neg)} > 0.7$	0.839
	Mean(Dist(Ref, Pos))	0.380
	Mean(Dist(Ref, Neg))	1.279
Val	$\mathrm{Dist(Ref, Pos)} < \mathrm{Dist(Ref, Neg)}$	0.975
	$\mathrm{Dist(Ref, Pos)} < 0.7$	0.947
	$\mathrm{Dist(Ref, Neg)} > 0.7$	0.897
	$\mathrm{Dist(Ref, Pos)} < 0.7$ and $\mathrm{Dist(Ref, Neg)} > 0.7$	0.850
	Mean(Dist(Ref, Pos))	0.367
	Mean(Dist(Ref, Neg))	1.246

Table 3. Accuracy of feature extraction

The threshold between positive pairs and negative pairs is 0.7, which can be determined through the density distribution map. The correct rate of positive relative less than negative image pair is 0.973, the probability of positive relative less than 0.7 is 0.938, the probability of negative image pair is greater than 0.7 is 0.900, and the probability of satisfying positive relative less than threshold 0.7 and negative image pair greater than 0.7 is 0.844. These can show that the model has a good ability to distinguish positive and negative image pairs. In addition, the average value of positive relative is 0.373. It can be obtained from the improved triplet loss function that the positive value will be closer to 0, indicating that the depth feature values of different image patches in the same area tend to be consistent.

5.2 Accuracy of Positioning

This section discusses and analyzes the results of on-board positioning experiment. It is can be seen in the Figure 5 and Figure 6, this paper shifts (163m, 152m) for the Google and Arcgis images of the verification area which is shown in Figure 4, and uses algorithm 1 (Table 2) for GCPs matching, respectively. Because the Arcgis image here is not in the train and test dataset, it can reflect the generalization ability and robustness of the model. Figure 5 and Figure 6 show the results of the first epoch of automatic matching when $\mathbb{R}_1 \in \{-4{:}\,4\}$, $S_1 = 50m$. The algorithm matches the correct points from the GCPs library on the images to be matched as basis for subsequent positioning.

It can be seen from the Figure 5 and Figure 6 that the points matched can be evenly distributed on the image, and the matching accuracy is good visually. For Arcgis images, it does not participate in training and testing, but can also accurately match GCPs. Therefore, in the feature, new images taken in the satellite can also matched on-board with GCPs library for positioning.

What can be seen in Table 5, this paper calculates and analysis the positioning accuracy of the three epochs in Algorithm 1. This paper also adds experiments with different offsets and adds a comparison with the traditional SIFT features. The SIFT 128-dimensional features extracted from the GCP image were pre-loaded on-board. The SIFT features extracted from the target image achieved on-board were matched with the pre-loaded features in the same area, and ransac algorithm were used to eliminate the mismatched points. The best results were shown in bold in the Table 5. The results show that the positioning accuracy of this algorithm is within 30 meters. The accuracy of matching in first epoch has a greater impact on the accuracy of subsequent matching. However, its ability to match small offsets is insufficient. Because it is difficult to distinguish the deep features of the micro-offset image from the GCPs, the matching ability of the micro-offset image need to be further improved with more training dataset. The matching accuracy of conventional SIFT algorithm was generally lower than the algorithm in this paper. In addition, the SIFT algorithm is difficult to match the images with large difference in the control point images, such as infrared images. However, the algorithm in this paper provides good robustness by increasing the range of the training set.

5.3 Efficiency Analysis

This paper evaluates the efficiency of the Nvidia Xavier that simulates on-board devices. This paper also evaluates the efficiency of other devices, such as Nvidia RTX2060 and RTX2080ti. Table 2 shows the efficiency of different devices. Among them, there are 64 frames in a control point when $\mathbb{R}_1 \in \{-4{:}\,4\}$, 100 frames in a control point when $\mathbb{R}_2 \in \{-5{:}\,5\}$, 400 frames in a control point when $\mathbb{R}_3 \in \{-10{:}\,10\}$.

Devices	$\mathbb{R}_1 \in \{-4{:}\,4\}$ /s	$\mathbb{R}_2 \in \{-5{:}\,5\}$ /s	$\mathbb{R}_3 \in \{-10{:}\,10\}$ /s	single frame/s
Jeston Xavier - 10W	1.254	1.873	7.319	0.0189
Jeston Xavier - 15W	0.629	0.942	3.517	0.0094
Jeston Xavier - 30W	0.359	0.534	2.024	0.0053
RTX2060	0.119	0.180	0.675	0.0018
RTX2080TI	0.077	0.099	0.364	0.0010

Table 4 efficiency analysis

As can be seen in the Table 4, the number of frames to be processed was different for different search ranges, but the processing time of each frame was the same. And for Xavier, the different power modes were also different. However, it can meet the efficiency needs of on-board processing within a reasonable search range.

Offset / m	Image	$\mathbb{R}_1 \in \{-4{:}\,4\}, S_1 = 50$		$\mathbb{R}_2 \in \{-5{:}\,5\}, S_1 = 10$		$\mathbb{R}_3 \in \{-10{:}\,10\}, S_1 = 1$		SIFT	
		x / m	y / m	x / m	y / m	x / m	y / m	x / m	y / m
(163, 152)	Google	16.90	2.15	16.90	7.90	**10.92**	**10.12**	23.23	31.10
	Arcgis	**16.90**	**2.15**	16.90	11.35	17.11	10.95	16.41	27.56
(175, 152)	Google	**32.49**	**2.15**	29.24	11.84	34.11	12.65	141.52	67.55
	Arcgis	32.49	2.15	22.74	12.38	**21.44**	**9.15**	6.25	27.03
(175, 175)	Google	32.49	26.43	**27.44**	**19.54**	28.60	20.03	57.88	36.50
	Arcgis	32.48	26.92	26.45	15.90	25.50	13.49	**5.68**	**4.93**

Table 5. Positioning accuracy

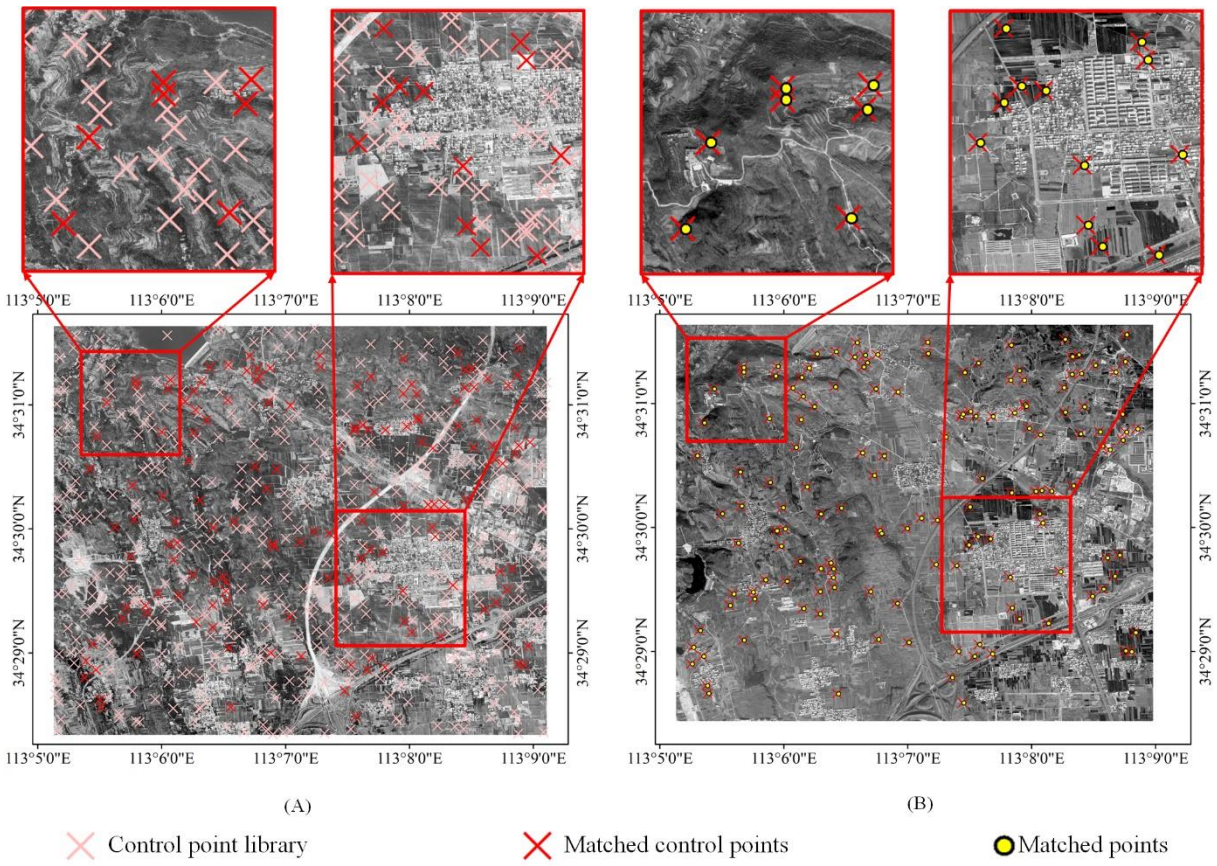

Figure 5. The results of Google image matching. (A) is GCPs image, and (B) is the Google image with (163m, 152m) offset.

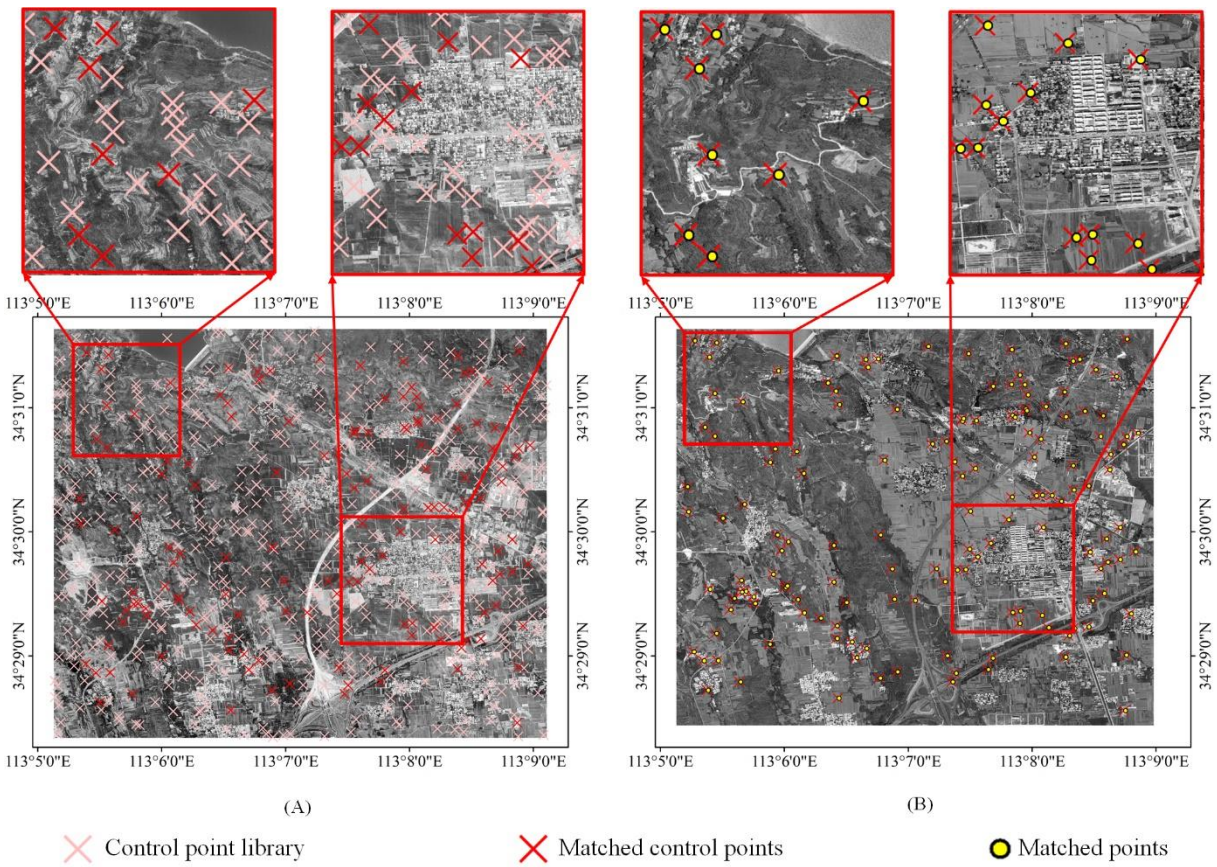

Figure 6. The results of Arcgis image matching. (A) is the GCPs image, and (B) is the Arcgis image with (163m, 152m) offset.

6. CONCLUSION

In this paper, image positioning on-board is achieved through GCPs matching on-board. This paper proposes an improve triplet loss function and trains a lightweight feature extraction model with it. The model embeds GCPs image patches into 128-dimensional feature space as GCPs library. Size of the extraction model and GCPs library are 9.83M and 15.3M respectively, then the model and library can be stored on-board for satellite. This paper also designed an algorithm for positioning on-board through GCPs matching. The images taken on-board will be positioned by GCPs matching real-time and the positioning accuracy is within 30 meters.
In subsequent studies, the positioning accuracy will be further improved under this framework. And the work done in this paper will be experimented on the upcoming Luojia-3-01 intelligent remote sensing satellite.

ACKNOWLEDGEMENTS

Acknowledgements for remote sensing images of Google (https://www.google.com/) and Arcgis (https://www.arcgis.com/). This work was supported by the National Natural Science Foundation of China under project 61825103, 91838303, 41801382, 61901307, and 91738302.

REFERENCES

Cai, Y., Guan, K., Peng, J., Wang, S., Seifert, C., Wardlow, B., Li, Z., 2018. A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach. Remote Sensing of Environment. 210, 35–47. doi.org/10.1016/j.rse.2018.02.045

Chen, W., Chen, X., Zhang, J., Huang, K., 2017. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-Identification. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 403–412.

Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N., 2016. Person Re-Identification by Multi-Channel Parts-Based CNN With Improved Triplet Loss Function. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1335–1344.

Dong, Z., Wang, M., Wang, Y., Zhu, Y., Zhang, Z., 2020. Object Detection in High Resolution Remote Sensing Imagery Based on Convolutional Neural Networks With Suitable Object Scale Features. IEEE Transactions on Geoscience and Remote Sensing. 58, 2104–2114. doi.org/10.1109/tgrs.2019.2953119

Gong, M., Yang, H., Zhang, P., 2017. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. ISPRS Journal of Photogrammetry and Remote Sensing. 129, 212–225. doi.org/10.1016/j.isprsjprs.2017.05.001

Hamida, A.B., Benoit, A., Lambert, P., Amar, C.B., 2018. 3-D deep learning approach for remote sensing image classification. IEEE Transactions on geoscience and remote sensing. 56, 4420–4434. doi.org/10.1109/TGRS.2018.2818945

Hayden, S.C., Sweet, A.J., Christa, S.E., Tran, D., Shulman, S., 2004. Advanced diagnostic system on earth observing one.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H., 2019. Searching for MobileNetV3. Presented at the Proceedings of the IEEE International Conference on Computer Vision, pp. 1314–1324.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs].

Kemker, R., Salvaggio, C., Kanan, C., 2018. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. ISPRS Journal of Photogrammetry and Remote Sensing, Deep Learning RS Data. 145, 60–77. doi.org/10.1016/j.isprsjprs.2018.04.014

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing. 152, 166–177. doi.org/10.1016/j.isprsjprs.2019.04.015

Ma, N., Zhang, X., Zheng, H.-T., Sun, J., 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. Presented at the Proceedings of the European Conference on Computer Vision (ECCV), pp. 116–131.

Mi, W., Fang, Y., 2019. Intelligent remote sensing satellite and remote sensing image real-time service. Acta Geodaetica et Cartographica Sinica. 48, 1586. doi.org/10.11947/j.AGCS.2019.20190454

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823.

Straight, S., Doolittle, C., Cooley, T., Gardner, J., Armstrong, P., Nadile, R., Davis, T., 2010. Tactical Satellite-3 Mission Overview and Initial Lessons Learned. Small Satellite Conference.

Vetrivel, A., Gerke, M., Kerle, N., Nex, F., Vosselman, G., 2018. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. ISPRS Journal of Photogrammetry and Remote Sensing, Geospatial Computer Vision. 140, 45–59. doi.org/10.1016/j.isprsjprs.2017.03.001

Vo, N.N., Hays, J., 2016. Localizing and Orienting Street Views Using Overhead Imagery, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 494–509. doi.org/10.1007/978-3-319-46448-0_30

Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., 2017. Aggregated Residual Transformations for Deep Neural Networks. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500.

Yu, Y., Guan, H., Zai, D., Ji, Z., 2016. Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-Hough-forests. ISPRS Journal of Photogrammetry and Remote Sensing. 112, 50–64. doi.org/10.1016/j.isprsjprs.2015.04.014

Zagoruyko, S., Komodakis, N., 2015. Learning to Compare Image Patches via Convolutional Neural Networks. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4353–4361.

Zhang, R., Li, G., Li, M., Wang, L., 2018. Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. ISPRS Journal of Photogrammetry and Remote Sensing, ISPRS Journal of Photogrammetry and Remote Sensing Theme Issue "Point Cloud Processing". 143, 85–96. doi.org/10.1016/j.isprsjprs.2018.04.022

Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856.