The 20th INTERNATIONAL SCIENTIFIC CONFERENCE
INFORMATION TECHNOLOGIES AND MANAGEMENT 2022
April 21-22, 2022, ISMA University of Applied Science, Riga, Latvia

Pretrained Deep Neural Network Models for Image Change Detection

Symagulov[1], Y.Kuchin[1], Jan Rabcan[2] , Ye.Kulakova[1], I. Assanov[1], R. Mukhamediev[1]

[1]*Satbayev University (KazNRTU), Kazakhstan, Almaty*

[2]*University of Zilina, Slovakia, Zilina*

Abstract

This paper describes a method for estimating the difference between images acquired from a UAV while flying along the same routes. The method is based on the use of embeddable images acquired by pre-trained neural networks. The main advantage of the method is that there is no need to train a deep neural network.

Keywords: embedding, deep learning, cosine distance, computer vision, UAVs

1 Introduction

Image change detection [1,2] is one of the current tasks in the field of computer vision, which is based on a set of photos taken at different moments in time. Change detection methods are applied to many tasks, such as anomaly detection with video surveillance and satellite cameras, road quality verification, and automation in the field of precision agriculture [3]. However, existing change detection methods are often trained to recognize specific objects such as machinery, people or plants, or to recognize specific changes such as changes in plant conditions [4].

The purpose of this paper is to evaluate the feasibility of using pre-trained deep neural networks to solve the problem of detecting significant changes in frames without additional training of the neural network.

2 Data and method

Image processing involves tasks of identification (cv1), verification (cv2), recognition (cv3) and determination (cv4) of visible object characteristics (speed, size, distance, etc.). The cv2 problem is often solved using so-called Siamese networks [5], where two images are processed by two identical, pre-trained networks. The method based on the application of pre-trained deep neural networks is efficient in terms of the effort involved in image partitioning and model training. One type of cv2 class task is to identify frames of two video sequences containing significant changes. For example, such video sequences can be obtained by flying a UAV along the same route. A well-proven method of face recognition is based on obtaining vector representations (so-called embeddings) of frames after the images have passed through the network layers. The results obtained (image vectors) are compared using a triplet loss function, which can be implemented as triplet distance embeddings [6] or a triplet of probabilistic embeddings [7]. However, this approach still requires some pre-training of the network, which may be difficult to implement in some operational monitoring tasks. Therefore, in this paper we use a direct comparison of image embeddings, which can be done by calculating the cosine distance, or cosine similarity between the embedding vectors. Since the scalar product of the vectors and the cosine of the angle between them are related by the relation (1):

$$e^{(1)} \cdot e^{(2)} = \|e^{(1)}\| \cdot \|e^{(2)}\| \cdot \cos(\theta) \tag{1}$$

therefore the cosine similarity can be calculated as follows (2):

$$\cos(\theta) = \frac{\sum_{i=1}^{N} e_i^{(1)} * e_i^{(2)}}{\sum_{i=1}^{N}[e_i^{(1)}]^2 * \sum_{i=1}^{N}[e_i^{(2)}]^2}. \tag{2}$$

The cosine distance can be calculated with the computational libraries as follows:

```
import scipy.spatial.distance as ds
dist2 = ds.cosine(embedding1,embedding2)
```

The value of dist2 can be between -1 and 1. A value of 0 means that the vectors are the same or, more precisely, point in the same direction in multidimensional space (the angle between them is 0). Significantly different vectors will have a distance estimate at a level close to 1.

The cosine distance value is used to match parallel frames from the two flybys based on the cosine distance between their vector representations. Then, based on experiments, a threshold of cosine distance has to be selected, above which the frames will be marked as having significant differences.

During the computational experiments, synthetic data obtained in the Unreal Engine 4 3D environment was used. The data represents videos of virtual UAV overflights over fields, with varying illumination, wind speed (which slightly changes the UAV's route), and season (vegetation colour) for different overflights. Random objects were added to the field, the frames with which the model should identify. The figure 1 shows a fragment of the video file in which there are foreign objects on the 4th frame.

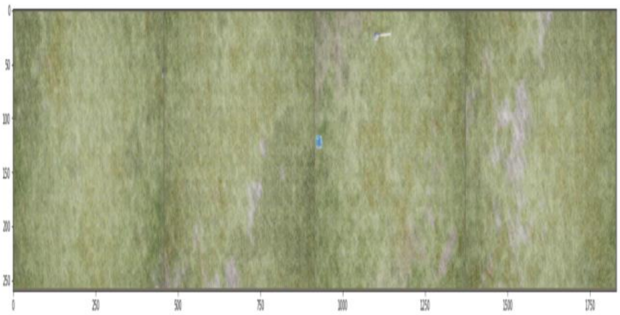

Figure 1 - Four frames of the video file

Alexnet [8] and ResNet 50 [9] were used as pre-trained models.

The AlexNet network architecture can be summarised as follows: Input (227 x 227 x 3) -> Conv (11 x 11 x 96, s = 4) - > Max pool (3 x 3, s = 2) -> Conv (5 x 5 x 256, same) -> Max pool (3 x 3, s = 2) -> Conv (3 x 3 x 384, same) -> Conv (3 x 3 x 384, same) -> Conv (3 x 3 x 256, same) -> Max pool (3 x 3, s = 2) -> -> FC (9216) > FC (4096) > FC (4096) > Softmax (1000) > Output. Conv - convolution layer, FC - full connected layer. The word same in convolution layer designation means that stride and padding are chosen so that the size of output tensor coincides with the size of input tensor. The original 227 x 227 three layer image is transformed into a 6 x 6 x 256 tensor which is fed to the full-link neural network to perform classification. The Resnet 50 architecture is even more extensive: Input (224x224x3) -> Conv (7x7x64, s=2) -> Max pool ((3x3x64, s=2) -> 3x Conv(3x3x64, s=1)->3xConv(1x1x256, s=1) ->4x Conv(1x1x128, s=1) ->4x Conv(3x3x128, s=1) ->4x Conv(1x1x512, s=1) ->6x Conv(1x1x256, s=1) ->6x Conv(3x3x256, s=1) ->6x Conv(1x1x1024, s=1) ->3x Conv(1x1x512, s=1) ->3x Conv(3x3x512, s=1) ->3x Conv(1x1x2048, s=1) -> Avg pool -> FC(1000) -> Softmax -> Output.

4 Results and discussion

The table 1 contains the results of computational experiments which show that, in general, pre-trained models can be used to determine the difference between frames under certain conditions. For each data set, 2,000 different thresholds were enumerated and the one with the maximum F1 Score was selected. This maximum F1 Score is also presented in the table, along with the area under the ROC curve.

TABLE 1 Model testing results

Model&Dataset	ROC AUC	Top F1 Score
resnet50_land1.csv	0,875059723	0,859813084
alexnet_land1.csv	0,819079471	0,859813084
alexnet_land2.csv	0,895814379	0,736378205
resnet50_land2.csv	0,905669552	0,793485342
alexnet_land4_orig.csv	0,710267544	0,570752714
resnet50_land4_orig.csv	0,740790885	0,606711409
resnet50_land4_sdvig.csv	0,583277779	0,443386955
alexnet_land4_sdvig.csv	0,622562454	0,453596288
alexnet_land5.csv	0,718294092	0,832342449
resnet50_land5.csv	0,749291395	0,832818074
resnet50_land6_spring.csv	0,67740601	0,473432056
alexnet_land6_spring.csv	0,682125913	0,476976209
alexnet_land7_fall.csv	0,73668296	0,55743326
resnet50_land7_fall.csv	0,753649462	0,621779859
alexnet_land7_spring.csv	0,777882066	0,58454387
resnet50_land7_spring.csv	0,797022712	0,635791881

It can be seen that resnet50 performs better (by 3-5%) than alexnet. Both networks performed poorly for the land6_spring image set, which reflects abrupt changes in weather conditions. The networks also failed to cope with the shift of land4_sdvig images, illustrating inaccurate UAV overflights along the route.

5 Conclusion

The use of pre-trained deep neural networks for frame difference detection has potential for practical applications. Experiments have shown that the quality of such classification increases significantly when monitoring within the same season under the same weather and lighting conditions. The method does not require extensive frame marking work and is suitable for classifying any changes. The resnet50 model showed the best results on the dataset used. However, a significant change of the UAV route or a change in weather conditions leads to a drastic decrease in the quality of frame classification. Further research should evaluate the possibility of partial marking to improve the quality of change identification.

6 Acknowledgements

This research was funded by the: Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan, grant number AP08856412 « Development of Intelligent Data Processing and Flight Planning Models for Precision Farming Tasks Using UAVs»; grant number BR10965172 «Space monitoring and GIS for quantitative assessment of soil salinity and degradation of agricultural lands in South Kazakhstan». Publication was partially realized with support of Operational Program Integrated Infrastructure 2014 - 2020 of the project: Intelligent operating and processing systems for UAVs, code ITMS 313011V422, co-financed by the European Regional Development Fund and ERASMUS+ ACeSYRI: Advanced Centre for PhD Students and Young Researchers in Informatics reg.no. 610166-EPP-1-2019-1-SK-EPPKA2-CBHE-JP.

EUROPEAN UNION
European Regional Development Fund
OP Integrated Infrastructure 2014 – 2020

MINISTRY
OF TRANSPORT
AND CONSTRUCTION
OF THE SLOVAK REPUBLIC

References

[1] Minu S, Shetty A. 2015 A comparative study of image change detection algorithms in matlab. – Aquatic procedia. **Vol. 4. 1366-1373**.

[2] Radke R. J. et al. 2005 Image change detection algorithms: a systematic surve. – IEEE transactions on image processing **14 (3) 294-307**.

[3] Mukhamediev R I, Symagulov, Kuchin Y, Zaitseva E, Bekbotayeva A, Yakunin K, Tabynbaeva L. 2021 Review of Some Applications of Unmanned Aerial Vehicles Technology in the Resource-Rich Country. Applied Sciences, 11(21), 10171.

[4] Yang C. 2020 Remote sensing and precision agriculture technologies for crop disease detection and management with a practical application example //Engineering **6 (5) 528-532**

[5] Taigman Y et al. 2014 Deep Face: Closing the gap to human-level performance in face verification. - Proceedings of the IEEE conference on computer vision and pattern recognition **1701–1708**.

[6] Schroff F, Kalenichenko D, Philbin J 2015 FaceNet: A unified embedding for face recognition and clustering Proceedings of the IEEE conference on computer vision and pattern recognition **815–823**

[7] Sankaranarayanan S. et al. 2016 Triplet probabilistic embedding for face verification and clustering. - IEEE 8th international conference on biometrics theory, applications and systems (BTAS). – IEEE **1–8**.