The 18th International Confernce on Automation Technology (Automation 2021),
November 12-14, 2021, Kimmen, Taiwan

Military Camouflage Detection using Transfer Learning on BlendMask

Bo-Xin Liu, Chung-Hsien Tsai*, and Pei-Yu Lin

Chung Cheng Institute of Technology, National Defense University, Taoyuan City, Taiwan
*Corresponding author: chunghsien.tsai@ccit.ndu.edu.tw

Abstract: This study applies transfer learning to a BlendMask model to effectively improve the military camouflage detection efficiency. Military camouflage is designed to blend in the target with the surroundings as much as possible by modifying its shape, contour and color, making it difficult for sensor detection. Presently, the lack of training data for camouflaged targets has made deep learning-based object detection a significant and challenging research area in military technology development. This research utilizes the high descriptive capability of the BlendMask model for image pixels and adopts transfer learning to reduce the required datasets size to propose a Transfer Learning on BlendMask framework. This framework aims to address the training needs of camouflaged object detection using a small size of training sets. A series of experimental results demonstrate that the proposed framework can improve the detection efficiency by at least 10% compared to the original approach.

Keywords: Military Camouflage Detection; Transfer Learning; BlendMask

1. Introduction

Computer vision applications have become an integral part of human life. The application can be found in the fields of intelligent transportation, data warehousing, and security surveillance. In the military field, which has also benefited from rapid advances in computer vision, modern warfare is adapted to a new, rapidly changing operation model. Computer vision and autonomous weapon systems will revolutionize the traditional military forces and lead the new era of national defense and military innovation. Military camouflage are imitations of animal camouflage in nature. In particular, in the ground battlefields, the effectiveness of military camouflage has become one of the critical issues to combat survival, making military camouflage detection a popular research topic in recent years. To avoid their predators, animals in nature conceal themselves via the resemblance between their bodies and habitats. Camouflage is primarily achieved through their body colors, although a small number of them can further conceal themselves via their body shapes [1]. Camouflage in the battlefields translates biomimicry into military use, which is a crucial tactic for soldiers. A good camouflage blends the outline of a soldier into the surroundings to evade detection and recognition from the enemy, and consequently allows effective battles. Therefore, camouflage combat tactics are of great importance to many military techniques and operational strategies.

To attack targets effectively, the accuracy of enemy target detection needs to be improved after the emergence of new AI-automated weapons. On the other hand, from the defense perspective, only comparatively effective military camouflages can survive in battles. Therefore, a good military camouflage needs to be difficult for human visual recognition and simultaneously challenging for sensor detection. The effective distinction of the camouflaged target from the surroundings has, therefore, become a hot research topic for both offense and defense. Among the deep learning models for image classification, the target detection methods can be roughly divided into two categories: object detection and object segmentation [2]. Object detection aims to locate the target object in the image using the deep learning model but difficult to separate out the outline of the object. However, object segmentation focuses on identifying the foreground or background objects via pixel analysis. If a pixel is determined to be in the foreground, the next step is to determine the foreground object that the pixel belongs to. Owing to the different occlusion levels, the shapes of the objects cannot be easily seen, which complicates object detection according to the object shapes. However, object segmentation is possible based on the pixel colors, which means that segmentation technique has to predict the object category for each pixel in the image. In cases where the shape of the camouflaged object is not discernible, segmentation technique is a practical solution. More specifically, instance segmentation is more suitable for military camouflage applications because it can detect objects and delineate obscure object contours through pixel quality.

In this way, instance segmentation can be the starting point of research on detecting camouflaged military objects with disrupted shapes. Nevertheless, military camouflage detection technology is still in its infancy owing to the lack of training data. Consequently, it is difficult to construct a deep learning-based algorithm for camouflage target detection. However, few-shot learning is common in other research fields, such as tumor detection training using a small size of X-ray images training set [3] in the medical field. Even with sufficient samples, sample labeling can introduce further complications to the problem owing to the lack of effective labeling tools. Consequently, few-shot learning algorithms have been increasingly discussed in recent years. Meta-learning in transfer learning, among others, utilizes pre-trained models to transfer the trained weights or parameters for subsequent use. It then fine-tunes the features based on the datasets of the field of interest. Because features in the old and new datasets share certain similarities, this technique not only reduces the training cost using the optimized parameter weights of the original trained model, but also prevents overfitting caused by the small sample size. The problem associated with insufficient training data can thus be solved, enabling possible implementation of deep learning-based military camouflage detection.

2. Related work

Deep learning-based object segmentation algorithms are mainly classified into semantic segmentation and instance segmentation, as depicted in Fig. 1 and Fig. 2. Both segmentation approaches are performed pixel-wise. The only difference is that the semantic segmentation can only segment the foreground from the background and no additional segmentation is made if there exists more than one object of the same type. Conversely, instance segmentation can distinguish objects of the same type, in addition to the aforementioned function of semantic segmentation. Object detection only gives bounding box (Bbox) and category outputs, whereas instance segmentation gives mask and category outputs. Hence, instance segmentation possesses the characteristics of both semantic segmentation and object detection.

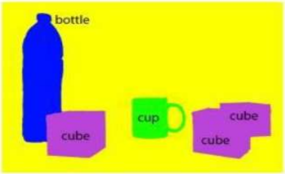

Fig. 1. Semantic segmenation

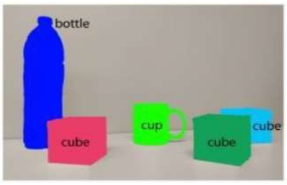

Fig. 2. Instance segmenation

The technique has gradually evolved into two branches based on bottom-up segmentation and top-down detection [4]. This is also the classification commonly used in many instance-segmentation studies. The idea of top-down detection is to first find the area where the instance is located (bounding box) by object detection, and then performing segmentation within the detection box, with each segmentation result being a different instance output. The bottom-up approach, on the contrary, first segments an image pixel-wise semantically and then combines the resulting segments into different instances through clustering, distance metric learning, or other algorithms. Regardless of whether a bottom-up or top-down approach is used, the key lies in the representation or parameterization of instance masks, or alternatively, whether the resulting mask is a local or global mask and how it can be represented or parametrized. Fig. 3 shows the differences between local and global masks. A local mask is generated by segmentation after box acquisition, such that the required mask is cropped out of the box. A global mask is obtained directly from the image. A mask can be represented intuitively using binary notation, that is, as a matrix of 0’s and 1’s. Nonzero entries indicate object existence, whereas zero entries represent the background. A binary mask can also be encoded into a fixed dimensions vector.

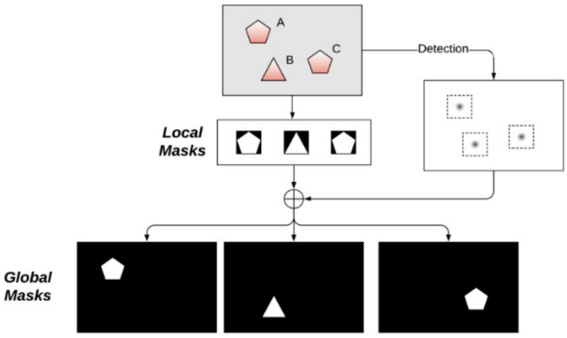

Fig. 3. Local Masks and Global Masks

According to the recent overall development trends, instance segmentation can be defined as a method to achieve both object detection and semantic segmentation. Examples include CenterMask [7], Yolact++ [5], BlendMask [6], SOLOv2 [8] and other deep learning models. Most instance segmentation techniques employ global masks, but model design differences result in performance differences. For example, Table 1 compares the performances of the three global mask approaches: Yolact++, SOLOv2, and BlendMask. The preliminary evaluation was conducted using the COCO val2017 dataset and default training weight parameters. To ensure fairness, the same backbone network is selected whenever possible: the backbone network for Yolact++ is ResNet50-FPN, and those for BlendMask and SOLOv2 are ResNet50. The effectiveness of the detection frame and mask is given on the left and right sides of the slash, respectively. From the table, BlendMask excels in accuracy. Its modularized and pipelined execution processes, combined with the top-down and bottom-up methods, not only improves the high-dimensional location information, but also takes the pixel-level prediction results into consideration, enabling more effective use of the full image details. This study will use transfer learning on BlendMask to conduct a few-shot learning study for military camouflage detection.

Table 1. Method comparsion

	mAP	AP50	AP75
Yolact++	35.2/33.7	55.8/52.7	38.1/35.5
SOLOv2	-/37.56	-/57.99	-/39.83
BlendMask	42.73/37.77	61.43/58.55	46.33/40.61

3. Rationale and Methodology

3.1 BlendMask

BlendMask is a one-stage dense instance segmentation method that combines the advantages of the top-down and bottom-up methods. It adds a bottom module to the anchor-free FCOS [9] detection model to extract low-level detail features and give a predicted instance-level attention map. While both the FCIS [10] and YOLACT [11] models combine high-level Bbox information with low-level per-pixel position-sensitive instance features. BlendMask utilizes a blender module to more effectively fuse these two features. In particular, it uses the top-down method to generate high-dimensional instance-level information (for example, Bbox location) and relies on the bottom-up method to obtain per-pixel prediction for fusion. The proposed blender module claims to be more efficient than the cropping approach in FCIS and the weighted additive approach in YOLACT. It excels in fusing instance-level full-domain information and low-level features providing details and location information. The higher-level features generated by the top-down method corresponds to a wide perceptive field and are more representative of the overall information (such as posture) of instances. The lower-level features produced by the bottom-up method retains location information and provides more detailed information of the instances. BlendMask generalizes the proposal-based mask generation. A more detailed and location-sensitive mask is created by the enhancement of instance-level information. The BlendMask network consists of three components: a Bottom module, a top layer, and a Blender module. The Bottom module is responsible for predicting the score maps. The features generated by Backbone or FPN are input into the module to calculate the required score maps, known as the bases. The top layer predicts the instance attention. For each Bbox predicted by FPN, a detection tower is concatenated with a convolution layer to obtain a top layer and generate the top-level attention map of the base for the corresponding Bbox. Finally, for each detected instance, the Blender module linearly combines the bases (score maps) and attention maps to generate the final instance masks.

The current neural network used by BlendMask is a deep neural network, which requires a large amount of data when it has to be trained from scratch. Consequently, most of its applications are in more general fields, and few applications are found in the military field, where datasets are insufficient. Nevertheless, recognizing everyday objects is not entirely different from detecting military objects: in both cases, humans or vehicles have to be discerned from the surroundings in the given images. The deficiencies of the existing BlendMask model can be complemented by transfer learning using this similarity. The weights can be transferred accordingly to reduce the reliance on large datasets, making it possible to complete the training using fewer samples.

3.2 Transfer Learning on BlendMask

The gradient descent method of the feedforward training method in the deep learning model is trained in accordance with the formula 1, where Θ represents the extraction of feature maps in the backbone network, θ represents the classifier, and α is the set learning rate, ∇L_D represents the loss function used by the model, using the feature map and classifier learned the previous time minus the learning rate multiplied by the value of the loss function to extract the next feature map and to update classifier.

$$[\Theta, \theta] \longleftarrow [\Theta, \theta] - \alpha \nabla L_D([\Theta, \theta]) \qquad (1)$$

Therefore, training a deep neural network model from scratch requires a multitude of data and a high-performance computing platform. It can be incredibly time-consuming. Using weights and parameters that are already well-trained by others built dataset can significantly reduce the training time. Transfer learning applied the characteristics which the features identified in the new task do not deviate much from those in the old task to reduce collection cost of sufficient datasets and to avoid the overfitting problems caused by insufficient data. Our method utilized fine-tuning technique of transfer learning to design the different number of frozen layers to retain the low-dimensional feature weights learned in the old task, and to train the high-dimensional features weights. In this way, if the military camouflage training data set is small, the overfitting problems of the deep learning model can be alleviated. On the one hand, it can avoid catastrophic forgetting caused by the old task learning model.

This study uses Transfer Learning on BlendMask to propose a loss function suitable for military camouflage detection, as shown in Formula 2. In Formula 2, we do not update the feature map extraction, which refer to the classifier to learn, where the i of Θ_i belongs to 1 to 5. When i is equal to 1, it means the stem layer in Fig. 4. When i is equal to 2, it means stage1, and the rest can be deduced by analogy. This corresponds to 1 to 5 in different freeze level, and then r means the LR we set. ∇L_T is the loss function. For the loss function here, we directly use the loss function used by blendmask, as in formula 3. L_{cls} means focal loss, and L_{reg} is the calculated value of IoU loss function. N_{pos} refers to the quantity of positive sample, after the calculation is completed, $\hat{\theta}$ is the overall loss function.

$$\hat{\theta} \longleftarrow \theta - r\nabla L_T([\Theta_i, \theta]), i \in \{1,2,3,4,5\} \tag{2}$$

$$L(\{p_{x,y}\},\{t_{x,y}\}) = \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) + \frac{1}{N_{pos}} \sum_{x,y} \mathbb{1}\{C^*_{x,y} > 0\} L_{reg}(t_{x,y}, t^*_{x,y}) \tag{3}$$

As shown in Figure 4, five freeze layer in the backbone network ResNet-50 were selected for the purpose of this study. Weights for feature extraction were extracted from the backbone network and partially transferred. These pre-trained weights were retained by freezing. Freeze1, freeze2, and freeze3… in Figure 4 refer to the methods used for each freezing: Freeze1 only freezes the stem block of ResNet-50, whereas freeze2 freezes the stem block and stage1, and so on. After extracting features from the backbone network, the results were sent to each module of BlendMask by FPN. New feature maps were used to fine-tune each module, including the score maps for the bottom module and the predicted instance attention for the top layer. The blender module, which is the combination of the first two, was also fine-tuned. This approach is based on the resemblance of the feature maps between the new and old tasks. It reduces the backbone network training cost and prevents over-fitting caused by small datasets, allowing more attention on the training of the subsequent modules.

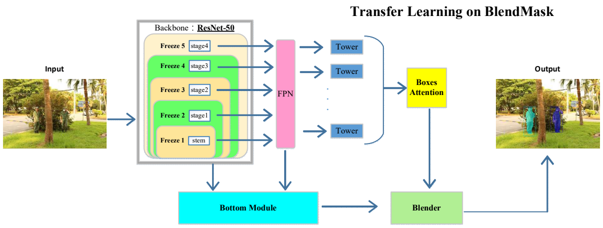

Fig.4. Transfer Learning on BlendMask

4. Experimental Results and Analysis

4.1 Environment Setup

The proposed "Transfer Learning on BlendMask" model is built on a Supermicro 7048GR workstation with a Dual Intel Xeon E5-2640V4s processor, a main memory of 128 GB, and a single Nvidia Tesla P100-16GB GPU. The operating system Ubuntu 18.04 is used, and CUDA 10.1 is adopted to accelerate the training speed and improve the performance of the deep learning model. PyTorch 1.6 is used as the deep model framework, and Python 3.7 is employed. The environment is managed via docker for various experiments, and an open-source toolkit is used to construct instance segmentation models.

In this study, fine-tuning Transfer Learning on BlendMask was used to evaluate the feasibility of camouflage detection with a small amount of training data. The public dataset provided by Kaggle [12] was used. It mainly contains images of armed military officers or riot police officers in various scenarios. After data sorting, approximately 2000 images were found to be usable. Among them, 100 images were selected and labeled for training, and another 100 were chosen for performance evaluation. When the dataset was ready, weights provided by Github were used as the pre-trained weights, and the model performance was evaluated as the benchmark for subsequent experiments. The concept of the ablation experiment was then adopted to adjust the parameters to obtain various variables with evident fine-tuning effects on BlendMask in transfer learning. The results confirm the feasibility of the small size of training set.

4.2 Performance Evaluation

To evaluate the performance of the proposed method, three performance factors, namely AP, AP50, and AP75, were computed. AP stands for average precision, which is commonly used in target detection evaluation. In object detection, the IoU value is used to determine the accuracy. AP50 is equivalent to the average precision when IoU equals 0.5, and AP75 corresponds to that when IoU is 0.75. In this study, BlendMask is used to determine whether camouflage detection can be realized with a small sample training. The backbone network used is ResNet-50. Table 2 lists the performance of the original pre-trained weights before training, and the values are taken as the performance benchmark.

Table 3 compares the performance of the model when different numbers of layers are frozen so that the weights of the frozen layers remain unchanged the training. The number 0 indicates that none of the layers are frozen, whereas 1 indicates that the stem block in ResNet-50 is frozen. Similarly, 2 denotes that the stem block and stage1 are frozen, and 3 indicates that the stem block, stage1, and stage2 are frozen, and so on. ResNet-50 is composed of one stem block and four residual stages, thus, five layers can be frozen at most.

Table 2. Original Performance of BelndMask model

Method	Backbone	AP	AP50	AP75
BlendMask	ResNet-50	59.13/ 47.43	87.07/ 85.70	66.54/ 49.10

Table. 3. The benchmark of the proposed method

Freeze	Max_iter	AP	AP50	AP75
0	1000	69.50/57.49	95.04/93.63	76.25/65.18
1		69.64/57.13	95.15/93.97	77.17/63.38
2		69.47/57.66	95.02/93.60	78.38/65.63
3		68.74/57.11	94.87/93.52	75.29/63.51
4		66.82/55.61	94.41/92.43	72.31/61.70
5		64.22/52.43	92.64/90.09	72.91/56.92
0	2000	70.49/60.66	94.89/95.31	78.63/69.08
1		69.96/59.78	94.65/95.98	73.97/66.41
2		70.38/60.96	94.80/95.24	79.68/69.28
3		70.44/60.28	94.83/95.32	78.28/64.55
4		67.84/57.82	94.18/93.64	75.14/63.63
5		66.19/54.40	93.55/91.62	73.31/59.32

As shown in table 3, after training with 100 images, the model performance improves significantly, compared to that in Table 1. The performance after different numbers of iterations will be discussed as follows. With 1000 iterations, both freeze0 and freeze1 give the best performance, but, in general, the performance of freeze0, freeze1, and freeze2 is comparable. With 2000 iterations, the best performance is between that of freeze0, freeze1, and freeze2. Unlike the performance when 1000 iterations are carried out, the performance of freeze3 is worse than that of freeze2 when 2000 iterations are done. Moreover, in both cases, the performance of freeze4 and freeze5 is reduced. This is likely due to the discrepancy between the feature maps of the old and new tasks. The subsequent modules cannot satisfactorily detect the features using the feature maps from the backbone network when too many layers are frozen. This means that the feature maps in freeze0, freeze1, and freeze2 are better than those in freeze3, freeze4 and freeze5. Hence, subsequent evaluation will place emphasis on freeze0, freeze1, and freeze2 and evaluate their performances in different epochs. Table 3 depicts the model performance under different epochs. Because satisfactory performance is mostly observed in freeze0, freeze1, and freeze2 (Table 3), only the model performance of these three cases under different epochs are compared in Table 4. There is a significant improvement in performance, compared to that presented in Table 2. Furthermore, optimal performance mostly occurs after 80 epochs. Therefore, it can be deduced that a higher number of epochs can lead to better model performance. When the number of epochs reaches 80, the accuracy is improved by more than 10%, which is quite significant.

5. Conclusions

The use of fine-tuning adopted from transfer learning to train BlendMask has presented promising results in camouflaged object detection. In addition to the modularized and pipelined execution processes of BlendMask, the combination of top-down and bottom-up methods not only improves the high-dimensional location information, but also integrates the pixel-level prediction results. Hence, the details of the entire image are more effectively used. Fine-tuning, adopted from transfer learning, uses a small amount of data to adjust the model weights. Owing to the presence of pre-trained weights from previous tasks, no overfitting occurs despite the small dataset used. The proposed model successfully combines the learning experience from the old task with the new task and applies it to generalized tasks. The findings of this study not only provide a solution for camouflaged object detection, but also proves that fine-tuning in transfer learning is capable of training models with limited data. For future deep learning network models, the problem of having insufficient data can be resolved this way. If there are other tasks in the future, the proposed model can be adopted.

5. Acknowledgement

The authors acknowledge the supports provided by the Ministry of Science and Technology of Taiwan, R.O.C., under grant MOST 109-2221-E-606-007.

6. References

[1] Date, A. R., & Shah, S. K. (2017, August). Camouflage moving object detection: A review. I*n 2017 International Conference on Computing, Communication, Control and Automation* (*ICCUBEA*) (pp. 1-6). IEEE.

[2] Wu, H., Liu, Q., & Liu, X. (2018). A review on deep learning approaches to image classification and object segmentation. *TSP*, 1(1), 1-5.

[3] Jiang, H., Diao, Z., & Yao, Y. D. (2021). Deep learning techniques for tumor segmentation: a review. *The Journal of Supercomputing*, 1-45.

[4] Wang, W., Shen, J., Cheng, M. M., & Shao, L. (2019). An iterative and cooperative top-down and bottom-up inference network for salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5968-5977).

[5] Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J.,(2019). YOLACT++: Better real-time instance segmentation, arXiv preprint arXiv:1912.06218.

[6] Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Yan, Y. (2020). BlendMask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8573-8581).

[7] Lee, Y., & Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13906-13915).

[8] Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C. (2020). SOLOv2: Dynamic and fast instance segmentation. arXiv preprint arXiv:2003.10152.

[9] Tian, Z., Shen, C., Chen, H., & He, T. (2020). Fcos: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[10] Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017). Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2359-2367).

[11] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9157-9166).