Exploring Color Modification in Multi-view Feature Fusion Networks

Nathan Zhao
Stanford University
nathanzh@stanford.edu

Patrick Li
Stanford University
prli@stanford.edu

Kaden Nguyen
Stanford University
kadenn@stanford.edu

1. Abstract

The existing camouflaged object detection (COD) literature roughly falls into 2 camps: novel model architecture trained on open source datasets, and existing models trained on new, domain-specific datasets. Due to the increasing relevance of military camouflage data, we set out to combine both approaches, modifying the architecture of an existing MFFN model to increase performance on a specific military camouflage dataset, ACD1k. Camouflaged object detection on military image data is a challenging task for a variety of reasons, such as relative sparsity and lack of dedicated model architectures. The MFFN model draws inspiration from human behavior when detecting camouflaged images, using multiple "views" that simulate a human looking at an object from several different angles. Our approach involved improvements on the attention mechanism, a tailored data preprocessing pipeline, and a data combination effect method that allowed the model to generalize from open source dataset. For attention, we implemented a separate "color attention" group to help the model incorporate features from various color transformed "views" of each image. For data preprocessing, we identified a flaw in the model's default image normalization, substituting our custom computed statistics for normalization. For the combination effect, we used the open-source CAMO dataset to fortify the insufficient military training data. While our modifications to the attention mechanism did not result in any improvements to performance, our data preprocessing and combination yielded significantly increased performance over all metrics. Our project suggests that more work is needed on modifying the MFFN model, potentially involving a heuristic to understand which views would make a good addition. Nevertheless, our improved performance across other approaches indicates how important data preprocessing and combination are for practical applications of existing models.

2. Introduction

Camouflaged object detection has been a pressing concern of computer vision researchers in the last few years. The majority of research in the field has been inspired by camouflage found in nature, the results of millennia of evolution. This trend is born out in the most widely available open source datasets, COD10K [3] and CAMO [6].

The papers introducing both datasets provide a breakdown of where the camouflaged images were sorced as a visualization:

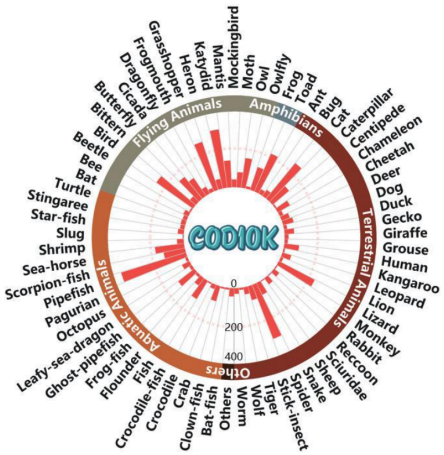

Figure 1. COD10K Labels [3]

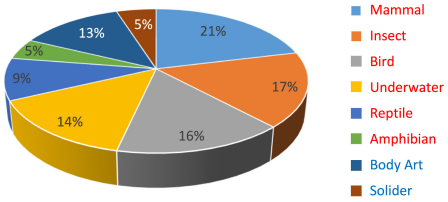

Figure 2. CAMO Labels [6]

In both cases, images other than camouflaged animals found in nature compose a small minority of the image data. While these datasets have been invaluable for proposing novel model architectures, there is still an open question on how well these models generalize to practical domains and applications. We seek to answer this question by applying the cutting edge MFFN [9] model on the task of camouflaged military images, while proposing further refinements

to the architecture. While camouflage in nature evolves over millions of years, military camouflage has undergone massive changes in the brief century since its creation. Because of the increased pace of development of camouflage, mechanisms used to defeat camouflage must also improve faster than evolution, with machine learning for computer vision as a possible solution.

The task of military camouflage has several unique challenges. Firstly camouflage created by humans has access to several tools found to lesser extents in the natural world. Human camouflage makes use of drastically changing the silhouette of a soldier, such as with ghillie suits. These suits make up the majority of pictures in the ACD1k dataset and often directly incorporate natural materials such as shrubbery in the design, making camouflage object detection harder. Outside of issues inherent to military camouflage, there is a lack of datasets publicly available that make training good models harder, let alone finding an existing open-source model.

The problem of accurately detecting camouflaged soldiers has major ethical implications for the future of the military. To see the increasing role Silicon Valley is playing in the defense industry, one only need to look at the recent defense contracts that startups such as Anduril Industries are starting to win. This problem can only be solved by developing better models and procedures for training on military-specific data. Without accurate detection, computer vision systems run the risk of causing civilian collateral damage or failing to identify lethal enemy targets. We as a group also chose this project due to a separate curiosity to explore the similarities between natural and military camouflaged data.

Our results are the product of 3 groups of experiments. For our experiments modifying the attention mechanism of the model, our proposed "color attention" group never surpassed the performance of the base model. However, our custom data preprocessing led to small performance increases across all categories, and our data combination effect with images from the CAMO dataset drastically improved performance. Detailed results of our runs can be found in the experiments section.

3. Data

Two open-source datasets, COD10K [3] and CAMO [6], are commonly used in the COD literature, including most of the papers we surveyed. While COD10K consists of 10000 images as the name suggests, only 5066 of them are of camouflaged objects. 3040 of these images are in the train set, and 2026 of them are in the test set. CAMO consists of 1250 camouflaged images, split into a train set of 1000 images and a test set of 250 images. There are an equivalent number of non-camouflage images, but they are not relevant for training our model. The reason why we focus on only images with a camouflaged object and background is that our base MFFN model is designed to be trained on all-camouflage datasets, matching the intuition that multiple views are mainly advantageous when the object detection task is complicated with camouflage.

To supplement these datasets with domain-specific military camouflage images, we used another open-source dataset from Kaggle, the ACD1k dataset [4]. This dataset consists of 748 training images and 330 testing images, all of soldiers in varied environments with varied camouflage technology. All the data we trained on uses ground truth objectness masks to identify where the camouflaged object is. To clean, separate, and combine these 3 datasets, we used a combination of Python scripts and shell commands.

While the original MFFN model uses default ImageNet data preprocessing and normalization, we experimented with custom preprocessing as mentioned in the experiments section.

4. Related Work

Our model is inspired by the intuition and general structure of the Multi-Feature Fusion Network model (MFFN) [9], which utilizes the multi-view strategy for camouflage detection (which we talk about in the Methods section). With the previous MFFN model, results were found with this model being trained on the COD10K [5] and CAMO [6] datasets, consisting of camouflaged and non-camouflaged images of mostly animals. Our model utilizes dataset normalization, extra attention layers for multiple color and brightness views, and dataset expansion. We elaborate on these ideas in the Experiments section. The results of this original MFFN model on our military camouflage object detection is listed in the first row of Table 3.

There is also related work on military camouflaged object detection (MCAM). In this paper, a SINet-V2 model is used on an MCAM dataset. We want to test the multi-view strategy on the MCAM dataset [5] and similar datasets to beat this existing research. We also aim to see if this novel idea of analyzing images from multiple perspectives to gather more information will generalize well beyond camouflaged animal detection. They also bolstered training of their SINet-V2 model with camouflaged animal pictures, and we also use this idea to strengthen our military camouflaged detection model.

In Camouflaged Instance Segmentation research [6], the model is split into an object detection step and then a binary segmentation step to separate the object from the background. This idea has traditionally used Region Based Convolutional Neural Networks (RCNNs) which runs relatively slow. This particular paper combined instance segmentation methods from Mask RCNN, Cascade Mask RCNN, MS RCNN, Retina Mask, and CenterMask. This paper also used a combination of camouflage and noncamouflage data, with a roughly 50-50 split. We adapted this idea by testing

dataset expansion with our model.

We also referred to other state-of-the-art research on object detection, including saliency estimation research [8]. This paper breaks salient object detection into steps. First, Simple Linear Iterative Clustering is used to decompose the image into 5 distinct colors. Then, a saliency map is made to distinguish the object from the background. Finally, it uses high dimensional Gaussian filters for smoothing on blurriness in the image. State-of-the-art papers on object detection provided novel ideas to address edge detection and handling blurry images. The camouflaged instance segmentation and simple linear iterative clustering papers contained relevant, recent ideas on camouflaged object detection, but we did not directly use these ideas for our model.

We also used a variety of existing metrics for calculating the accuracy of created foreground maps. The statistical F-measure treats the problem like a binary classification one. It returns a score by comparing the predicted and actual foreground maps by pixel. First, the weighted F-measure [7] is used to generalize the definition of the F-measure by considering the problem as multiple binary classification problems. It does this by also considering the location and neighborhood of each pixel-wise comparison. The structure measure [1] leaves the scope of traditional pixel-wise comparisons and instead considers the general structure of the predicted foreground map and compares that to the structure of the ground truth map. Structure measure separately evaluates region-aware and object-aware structural similarity. Enhanced-alignment measure (E-measure) [2] uses a combination of pixel-wise comparisons and overall image-based comparisons. This captures both global and local statistical information.

Using various measures for our tests ensures that our model performs better than existing work based on pixel-wise comparisons, structure-based comparisons, and more. For practical uses of military camouflage object detection, we want to ensure that the structure of an object is correctly identified, and that the outline of the object is as accurate as possible. For military application, this gives us better understanding of what we are aiming at, the structural features of the object being discerned, and which actions they may be taking (e.g. a tank aiming itself towards towards a base).

5. Methods

We will be basing our model off the MFFN model, which takes images from an existing dataset, then considers the image under affine, mirror, and resizing transformations to generate multiple perspectives of the image, simulating different object orientations and distances from the camera. A ResNet which we initialize with pretrained ImageNet weights is responsible of extracting features from each of the views. The model then passes this output through the CAMV module, which extracts information from this combination of features to generate a combined encoding of image features. Finally, the model restores each feature map through upsampling to the original size to get the output. We chose the MFFN model due to its multi-view architecture, better allowing it to handle the blurred images common in military applications. The larger diagram is visible in Figure 3, featuring its view combinations, attention mechanisms, and important upsampling layers.

The core idea of the MFFN model, considering various transformations of an image, has had unprecedented success in camouflage detection research in identifying semantic cues of camouflaged objects. These multiple viewpoints of the image enables the encoder to learn more detailed information about the boundary and contents of the image with less data. The CAMV module then decodes the complementary relationships between the various viewpoints and expresses these relationships as output values.

Utilizing multiple views captures greater nuance of the images. Stacking multiple views in front of the CAMV module is similar to considering information of an image from multiple perspectives, extracting the most important features from each persepctive to make the most informed decision.

Using the MFFN model, we train with an SGD optimizer with momentum = 0.9 and weight decay of 5e-4. We train for 50 epochs with batch size 8 and learning rate 0.05.

Our approach to building off of this base model comprises 3 separate directions, all informed by various intuitions. Our first direction involves improvements to the base model architecture, with focus on modifying the attention mechanism to increase the number of views for our proposed ”color attention.” Our second direction involves image preprocessing. A critical part of training a good model, our preprocessing focuses on edge detection as well as normalizing our domain-specific military dataset to have better performance than the base model. Our third direction involves combining the military training dataset with additional camouflaged images from the CAMO dataset [6]. This approach, inspired by the MCAM paper [5], uses non-military camouflage image data to combine with the relatively sparse ACD1k dataset.

6. Experiments

6.1. Color Attention

In order to build on the model architecture from the MFFN paper [9] that served as our starting point, we first verified that the model had enough expressivity to support more views. To do so, halved the size of the hidden downsampling layers from 32 to 16. The results of running our low-dimensional model are shown in Table 1.

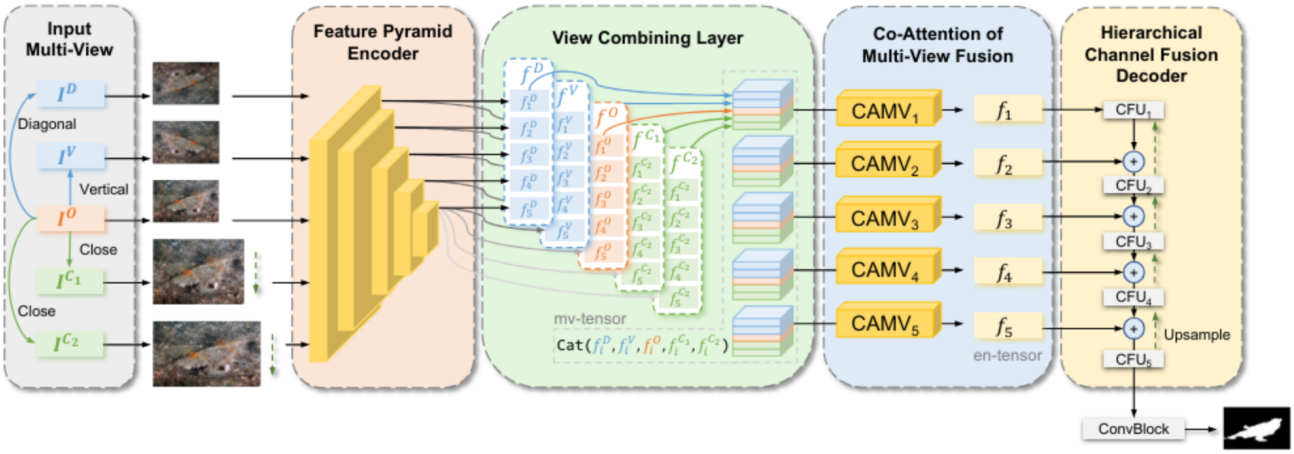

Figure 3. The original MFFN architecture from the paper [9], showing the 5 views and CAMV module used to combine them with attention

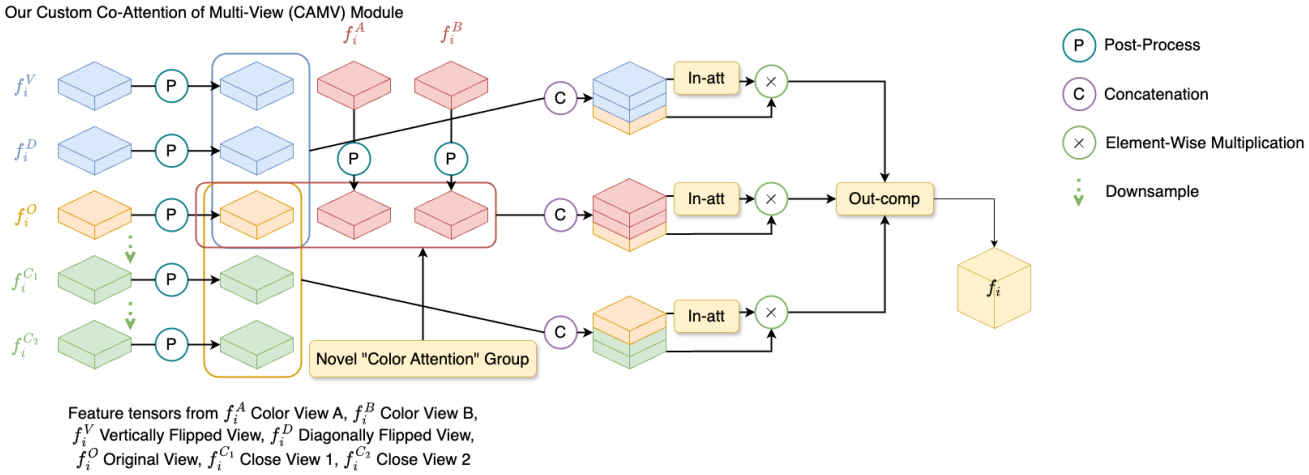

Figure 4. Our modified architecture of the CAMV model from the MFFN paper. We have added a third view type, color, that is processed by an intra-class attention (*In-att*) mechanism along with the other two existing types. These three attention groups are fused by the complementation of external classes (*out-comp*). Like in the original model, our CAMV is essentially a two-stage attention mechanism.

Data	smeasure	mae	meanfm	meanem
ACD1k	0.852	0.049	0.828	0.902
ACD1k low-dim	0.851	0.049	0.827	0.902

Table 1. Results on ACD1k dataset of base MFFN model and low-dim MFFN model with halved hidden dimensions

Halving the hidden dimensions barely reduces the performance of the model, meaning the base model has sufficient parameters to train on more features without compromising performance. This informed our decision to introduce more views into the MFFN model as outlined below.

The MFFN intuition is that several views increase performance for COD tasks, as inspired by nature. This is done through the co-attention of multi-view (CAMV) module, which performs 2 stages of attention, one over each attention group and one at the end to fuse all groups into one feature vector. The views from the original MFFN model include the original view f_i^O, the vertically flipped f_i^V, the diagonally flipped f_i^D, first close view $f_i^{C_1}$ and the second close view $f_i^{C_2}$. The original MFFN model divided the views into 2 attention groups: a "viewing angle" group including f_i^O, f_i^V, f_i^D and a "viewing distance" group including $f_i^O, f_i^{C_1}, f_i^{C_2}$.

By using the Co-Attention of Multi-View (CAMV) module to essentially attend over views of an input image, the same feature extraction resnet can be applied to all different views of images. This matches the general MFFN intuition, as humans and animals use the same visual cortex to process views of a camouflaged object from different angles

and distances.

Our contribution to the model was creating a 3rd "color attention" group consisting of f_i^O, f_i^A, f_i^B. Here, f_i^A and f_i^B are stand-ins for the several pairs of color transformations we tried when training our updated MFFN model. Our modified architecture is shown in Figure 4, and the results of our various color-attention runs can be seen in Table 2.

We first tried brightness and contrast color views with relatively low transformation strength. This was mainly a proof of concept to test that our implementation of color attention actually worked, as the new color views were not modified a large amount. Due to the similar if slightly degraded performances to the base model, we were confident our implementation was correct.

Our next run involved jitter and channel shuffle color views in an attempt to apply more significant changes to input images. Intuitively, we thought this would give the model more features to work with. Additionally, just as the various distance/angle views lead to distance/angle invariance in the base model, we hoped that color shifts would lead to a form of "color invariance". Having our model be invariant to color was especially important as our dataset consisted of both woodland, snow, and desert camouflage, all with wildly different color palettes. Unfortunately, the performance of this model was quite poor.

Our next run involved two RGB shuffles as color views that were an attempt to address the poor performance of the previous model. We thought our poor performance might be due to random jitter and channel shuffle being too nondeterministic, making it impossible to train a consistent model to combine features through linear combinations such as an attention mechanism. Thus, we provided two hard-coded shuffles of the channels, mapping RGB channels to BRG and GBR respectively. Our work did not pay off, as this model has similarly poor performance. Finally, we tried hue and saturation as a less extreme transformation similar to RGB shuffling, but this also had no effect.

f_i^A	f_i^B	smeasure	meanfm	meanem
None	None	**0.852**	**0.828**	**0.902**
Brightness	Contrast	0.851	0.823	0.900
Jitter	Chan. Shuf.	0.844	0.813	0.894
BRG Shuf.	GBR Shuf.	0.845	0.815	0.890
Hue	Saturation	0.849	0.824	0.898

Table 2. Results on ACD1k dataset of various models with "color attention". Rows with "None" are the base model. f_i^A and f_i^B are the feature types.

The results of our experiments for color attention are quite underwhelming. Despite rounds of debugging and tweaking the pairs of views, our best model performs slightly worse than the original model with only 2 attention groups.

We theorize that the "color attention" system does not work because color is likely not interchangable within military CAMO datasets, where in general images are expected relatively consistent with earthy tones. Additionally, beyond this, color channels may not necessarily be variable within images: when one channel is interchanged with another or there are saturation/hue shifts, the detected object is likely unnatural and indescribable through the initial ResNet's feature extractions, and most of the time, the new views therefore are ineffective.

To implement color attention, we utilized the Albumentations library to introduce transformations of brightness, contrast, jitter, etc. For channel shuffle, modification was applied algorithmically with our own implementation.

6.2. Normalization

Proper normalization is standard practice for training any good machine learning model, as demonstrated in class. Normalization gives all the data a normal distribution at the beginning of training, leading to a model that is not biased towards one portion of the dataset or another. The base MFFN model normalizes with the common practice used by both PyTorch and the Albumentations library. In this method, the mean and standard deviation of the ImageNet dataset are used to normalize the data, under the assumption that input images probably follow a similar distribution. We were unconvinced that the same would apply to ACD1k with its military specific images, so we wrote a script to compute the mean and standard deviation across channels for our training dataset. The results are shown below.

Dataset	Mean	STD
Default	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
ACD1k	[0.340, 0.438, 0.453]	[0.263, 0.251, 0.259]
Comb.	[0.382, 0.456, 0.463]	[0.282, 0.258, 0.259]

Table 4. Calculated mean and standard deviation of each dataset. "Default" indicates ImageNet statistics and "Comb." indicates the combined ACD1k + CAMO dataset from our later dataset combination experiment.

As shown, our manually computed statistics vary significantly from the default values, and our intuition was validated by the performance of our various models. As seen in Table 3, normalization improved the performance of both our model on ACD1k and on ACD1k + CAMO across all metrics. Our results indicate that proper data normalization is highly important for specific applications of COD models. A small addition to preprocessing can provide performance improvement with no drawbacks.

Data	smeasure	wfmeasure	mae	adpfm	meanfm	maxfm	adpem	meanem	maxem
ACD1k	0.852	0.791	0.049	0.830	0.828	0.837	0.908	0.902	0.915
ACD1k Custom Norm	0.854	0.794	0.049	0.830	0.830	0.839	0.909	0.903	0.917
ACD1k+CAMO	0.860	0.802	0.045	0.832	0.832	0.841	0.914	0.904	0.921
ACD1k+CAMO Custom Norm	**0.866**	**0.809**	**0.042**	**0.836**	**0.838**	**0.850**	**0.917**	**0.913**	**0.927**

Table 3. Our 3 best results compared to training the base model on just the ACD1k dataset. The biggest jump in performance comes after the combination effect with CAMO. Normalizing the training data with our custom computed summary statistics instead of ImageNet defaults consistently yields a smaller boost to performance, with larger effect on the richer, combined data

6.3. Dataset Combination

While normalizing based on our specific data yielded a small improvement and makes sense in the context of all the data processing we did in CS 231n, we were not fully satisfied with our model. To see where to go from here, we returned to the paper that inspired us to focus on military camouflage initially, the MCAM paper [5]. While this paper uses a different private military dataset as well as a very different SINet-V2 model architecture from us, their main contribution was based on combining the military training data with various non-military open-source COD datasets.

Based on these results, we trained our model on a combination of the ACD1k training dataset as well as all 1250 camouflage images from the CAMO dataset. This is because in the MCAM paper, the best results were obtained after the combination effect with CAMO, possibly due to its similar size to the military dataset and greater proportion of non-animal data. Our model was evaluated during testing on just the test set of ACD1k, allowing us to directly compare performance with our existing models. The results of 4 models across all metrics are shown in Table 3. Model 1 was trained on ACD1k with default normalization as a control, model 2 was trained on ACD1k with our custom mean and std, model 3 was trained on the combined dataset with default normalization, and model 4 was trained on the combined dataset with custom normalization. We immediately noticed our largest performance increase to date due to the combination effect compared to our base model, even with default ImageNet normalization.

We tried a second run on the combined data, this time with normalization based on our computed mean and standard deviation. Interestingly enough, performance increases across all categories between the default and custom normalized combined dataset are over twice the performance increases between the default and custom results on the base ACD1k dataset. The increased benefit of normalization when provided with a richer training dataset leads us to believe that the scarcity of military-specific training data is the main bottleneck for our and probably other military COD models. The existing MFFN model seems to extract most of the possible performance from the limited ACD1k training data, making it hard to substantially boost performance merely through changes to architecture.

7. Conclusion

Our research discovered new insights into improvements for the existing MFFN model, particularly within the field of camouflaged military object detection. Our most exhaustive idea for an architecture improvement on the MFFN model, adding a color attention layer, was unfortunately unable to produce a significant increase in results alone. However, as seen in Table 3, our other experiments including manually normalizing training data and the combination effect with non-military COD data yielded much better performance.

With improved accuracy on existing work, our model and experiments provide insights that may be relevant to current applications of machine learning for military camouflaged object detection. While our model needs significant improvement before being put into practice, our ideas can also be used to improve progress in the field of camouflaged object detection. By building onto our idea of increased attention groups as well as basic practices such as normalization and combination, we can construct models better able to ethically identify enemy combatants.

Some future ideas could be generalizing our multi-view model to similar fields like occluded object detection or object detection in smoke. Intuitively, the concept of considering many transformations of an image to gather information from multiple perspectives could improve the success of object detection in these fields as well. For example, we may explore blurring standard camouflaged images as another set of views. Such research could be critical help for firefighters as well as search and rescue teams, potentially saving many lives.

References

[1] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji. Structure-measure: A new way to evaluate foreground maps, 2017. 3

[2] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji. Enhanced-alignment measure for binary foreground map evaluation, 2018. 3

[3] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2

[4] A. Haider. Adaptive camouflaged dataset (acd1k), 2023. 2

[5] K.-S. Hwang and J. Ma. Military camouflaged object detection with deep learning using dataset development and combination. *The Journal of Defense Modeling and Simulation*, 0(0):15485129241233299, 0. 2, 3, 6

[6] T.-N. Le, Y. Cao, T.-C. Nguyen, M.-Q. Le, K.-D. Nguyen, T.-T. Do, M.-T. Tran, and T. V. Nguyen. Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite. *IEEE Transactions on Image Processing*, 31:287–300, 2022. 1, 2, 3

[7] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014. 3

[8] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012. 3

[9] D. Zheng, X. Zheng, L. T. Yang, Y. Gao, C. Zhu, and Y. Ruan. Mffn: Multi-view feature fusion network for camouflaged object detection, 2022. 1, 2, 3, 4