The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

CTENet: A Weakly Supervised Approach to Camouflage Detection Using Enhanced Texture, Contrast, and Edges

Fateme Ekhtiari * Ali Jafari† Mohammad Erfan Mesbah‡

Abstract

Identifying camouflaged objects in images is a major challenge in computer vision, requiring precise differentiation from complex backgrounds. Traditional methods often struggle due to the unpredictability of camouflage patterns, necessitating extensive labeling efforts. Deep learning can help, but a lack of labeled data hinders progress. To address this, we employed weak supervision using scribble annotation to reduce labeling efforts while maintaining accuracy. We introduced CTENet (Contrast Texture Enhanced Network), trained on the S-COD dataset. CTENet features four key modules:The LCC module simulates the visual suppression process of the visual system to improve image contrast and clarity. The TEM module utilizes the receptive field present in the visual system to enhance texture. The boundary ambiguity between foreground and background has been a fundamental challenge; thus, the EDB module was proposed to assist the network by reinforcing these boundaries. The CBAM module is used to highlight important features in the image. Experimental results demonstrate that our model outperforms previous approaches in camouflage detection.

Keywords: Camouflaged objects, Deep learning, scribble annotation, Weak supervision

1 Introduction

Finding camouflaged objects in images is a big deal and has many applications. Being able to find and locate these hidden objects is crucial for military surveillance, wildlife monitoring, and security systems. For example, animals use camouflage to survive and protect themselves from predators (see Figure 1). Detecting camouflaged objects is a tough challenge that requires many techniques including:

Image Processing: Image processing algorithms can be used to analyze images and find objects that blend in with the background. For example, edge detection can be used to find the edges of objects and separate them from the background.

Figure 1: Camouflage in Wildlife [8]

Machine Learning: Machine learning models can be trained on sample images to detect camouflaged objects. For example, a deep neural network can be trained with camouflaged images and non-camouflaged objects to detect camouflaged objects in new images.

Computer Vision: Computer vision can be used to extract features of camouflaged objects and separate them from the background. Research on finding camouflaged objects is moving fast, new and innovative methods are being developed. AI algorithms can recognize objects based on patterns and features, reducing human error.

The goal of this research is to detect camouflaged objects in images using deep learning methods. Detection of camouflaged objects has many applications in art[4] , medical diagnostics[10], industrial defect detection[12] [22], agriculture[5], surveillance cameras[2], defense, security, and military[15]. Although COD methods have performed very well, they are heavily dependent on pixel-by-pixel annotation in large datasets. The first challenge is that pixel-by-pixel annotation is time-consuming. Annotating an image takes almost several minutes, making it difficult to build large datasets. In contrast, based on our experience, annotating using lines takes only a few seconds depending on the type of camouflaged object, which is several times faster than pixel-by-pixel annotation.

Another challenge in camouflage detection is the unclear boundaries between objects and their backgrounds. This ambiguity makes it harder to separate the object from the background, making this task more difficult than other object recognition tasks. In many

*Faculty of Electrical and Computer, Malek Ashtar University of Technology, fa3eme.ekhtiyari@gmail.com
†Faculty of Electrical and Computer, Malek Ashtar University of Technology, iustuser@mut.ac.ir
‡Faculty of Electrical and Computer, Malek Ashtar University of Technology, mailto:messbah.m.e@gamil.com

cases camouflaged objects blend in with the background naturally, making it hard to identify using traditional algorithms. So we need to develop more advanced algorithms and new methods to improve the accuracy under such conditions. Given these challenges, further research on camouflage detection is needed to provide better solutions for object detection in complex environments.

We used weakly supervised methods that first used the S-COD[1] dataset labeled with lines and then proposed CTENet to solve the ambiguous boundary problem. Initially, some studies used traditional feature-based methods (such as texture, brightness, color, etc.) for foreground-background differentiation, which had high computational costs. In 2019, the first deep-learning network was introduced[14]. This network heavily relies on training data. If the training data does not include sufficient and diverse samples of camouflaged objects, the network's performance may decline. Additionally, this method may not perform well under varying lighting conditions or complex backgrounds. In 2020, the SINet[9] architecture was introduced as the first robust deep learning network that also depended on training data along with their labels; pixel-level labeling for training data is highly labor-intensive. In recent years, CNN-based methods employing complex strategies have achieved significant advancements in COD tasks. For example, SINetV2[8] and BASNet[25] use multi-stage approaches for initial segmentation that succeed through enhancement and refinement techniques. However, most of these deep network-based approaches that have achieved superior performance still require high training samples, imposing a considerable annotation burden.

In Section Two we looked at the previous work. In Section Three we will explain our approach and architecture for improving camouflage system performance in Section Three. Finally, we will detail our proposed methods and architecture and present the results obtained from experiments and evaluations conducted on datasets in the last section while highlighting the importance of these findings for advancing future research in this field.

2 Literature Review

2.1 Camouflaged object detection

In recent years, deep learning approaches have successfully replaced classical features with learned features to a large extent. Network-based methods, particularly deep neural networks, are recognized as significant innovations in this field. These methods leverage the power of deep learning to identify complex patterns and nonlinear features in large datasets, which is why they are increasingly used for detecting camouflaged objects. Initially, deep learning-based methods in the domain of camouflage were based on neural networks, but subsequently, transformers demonstrated remarkable advancements by capturing long-range features. Recently, CNN-based approaches have made significant advancements in camouflaged object detection (COD) with the release of large-scale datasets. Some works attempt to extract the subtle features of camouflaged objects from the background through carefully designed feature exploration modules, such as background feature learning[20], texture learning[27], and frequency domain learning[26]. Additionally, multi-task learning frameworks are commonly used for COD. These methods generally introduce tasks such as classification, edge/boundary detection, and object ranking. Furthermore, some methods identify camouflaged objects by mimicking the behavioral patterns or visual mechanisms of predators, such as search and identification processes and zooming in on camouflaged objects. Although CNN-based models have achieved promising performance, these methods do not examine long-range dependencies due to limited receptive fields, which are critical for COD in images containing diverse objects. Given the superiority of transformers in modeling long-range dependencies, recent studies have sought to leverage their potential in various visual applications such as image classification, semantic segmentation, and object detection, which have seen significant advancements. By utilizing attention mechanisms, transformers perform better than CNN-based models in capturing long-range dependencies[3]. They can learn image data in a sequential approach. Unlike convolutional layers, the multi-head self-attention layer in transformers has dynamic weights and a global receptive field, making it more effective and powerful in capturing non-local knowledge. This property has been utilized in COD tasks in recent years; however, transformers suffer from high computational and memory costs. FSPNet[13] is designed to improve local modeling and feature aggregation, while OSFormer[24] is introduced as the first one-stage system for segmenting camouflaged instances. These models effectively combine local capabilities and long-range dependencies through innovative mechanisms, contributing to greater accuracy in object identification.

2.2 Weakly Supervised Learning

In weakly supervised learning, training data is annotated with incomplete or inaccurate labels. The level of supervision is "weak" because the provided annotations are not as precise as those in fully supervised learning. The goal of weakly supervised learning methods is to learn from this limited information to predict or classify

[1] scribble camouflaged object detection

new samples. There are various forms of weak supervision; some common types include partial labels and inaccurate labels[19]. Weakly supervised learning allows models to achieve similar or better performance compared to fully supervised methods using weaker and less costly labels. This approach is particularly useful when accurate labeled data is not available or is expensive to obtain. Weakly Supervised Segmentation (WSS) models are designed to exploit weak labels instead of relying on precise pixel annotations[16]. For example, these labels can include image-level labels, bounding boxes, line segments, and points.

2.3 Weakly Supervised Camouflaged Objects Detection

recent advancements in semantic segmentation images have largely been driven by deep learning techniques, most of which relate to deep learning techniques. This research also examines deep learning-based semantic segmentation methods, particularly weakly supervised approaches, as research in this area is limited and there is a need for more efficient methods. Despite advancements, there are still shortcomings in the existing methods, and studying weakly supervised learning could enhance the efficiency of fully supervised learning and move toward unsupervised learning.

a. Segmentation of Histopathology Images: The author examines an approach for segmenting histopathology images using point annotation that is weakly supervised. This method improves segmentation accuracy by using a contrast-based variational model that identifies important image features through contrast difference analysis. This approach is especially applicable in histopathology images that require high precision[23].

b. Line-Based Dataset (S-COD): This dataset is the first collection for Weakly Supervised COD (WSCOD) and presents a line-based architecture that expands lines into camouflaged areas by increasing contrast. However, the sparsity of line labels and the lack of explicit guidance create challenges in accurately determining the boundaries of camouflaged objects[11].

c. (MiNet): This paper proposes a new network called MiNet for WSCOD that addresses challenges arising from insufficient lines. MiNet includes an RGM module that utilizes extracted regional features to produce distinct edge maps and has also designed a region boundary refinement network that iteratively and multi-level refines object boundaries[17].

3 Methodology

3.1 Overview of Structure

In this section, we will examine and explain the proposed architecture of CTENet, which is utilized in the present research. Additionally, the various modules employed in this architecture will be introduced in detail to clarify the role of each in the process of identifying and segmenting camouflaged objects. Given the importance of these stages, we will strive to thoroughly and scientifically investigate each relevant aspect, providing a suitable foundation for future analyses. The introduced framework, named CTENet, is designed for image segmentation. This network leverages the main structure of ResNet-50 to extract input features at different scales. CTENet consists of four modules:

- **Local Context Contrast Module (LCC):** This module enhances contrast and image clarity by mimicking the visual suppression process and improving boundary lines in hidden areas.
- **Texture Enhancement Module (TEM):** Utilizing the receptive field present in the visual system, this module enhances texture to aid in better object detection.
- **Edge Detection Boundary Module (EDB):** This module emphasizes the need for guidance to direct segmentation correctly, alongside the importance of enhancing texture and contrast.
- **Convolutional Block Attention Module (CBAM):** This module is designed to highlight key features in the image.

CTENet combines the power of deep learning algorithms with the simplicity of scribble annotation through weak supervision, leveraging the strengths of both techniques to facilitate accurate identification and highlighting of camouflaged objects in complex scenes. Overall, the LCC module[11] is used for low-level features, while TEM[8] is used for high-level features to obtain contrast-rich information and detailed texture information. Initially, input is fed into the ResNet-50 network to extract multi-scale features (see Figure 2). These features are then transferred to the LCC, TEM, EDB[17], and CBAM modules. The extracted features are obtained at five different levels (Xi), which include lower-level features (X1 and X2) as well as higher-level features (X3 and X4). In addition, CTENet utilizes an auxiliary extractor (CBAM)[21] to obtain key image features. The extracted features are integrated using a combination of multiplication and cross-aggregation strategies. CTENet ultimately produces multi-level segmentation maps as output and also extracts an intermediate feature map for calculating the loss function.

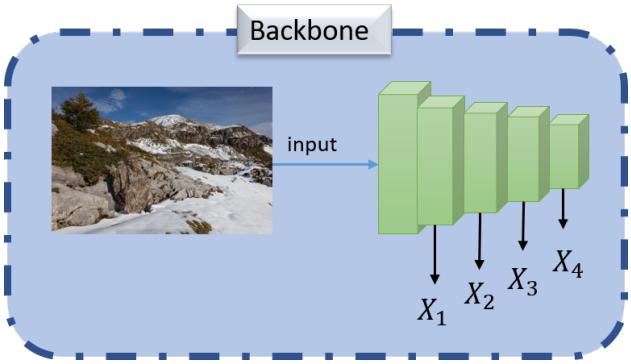

Figure 2: extracting features from ResNet50

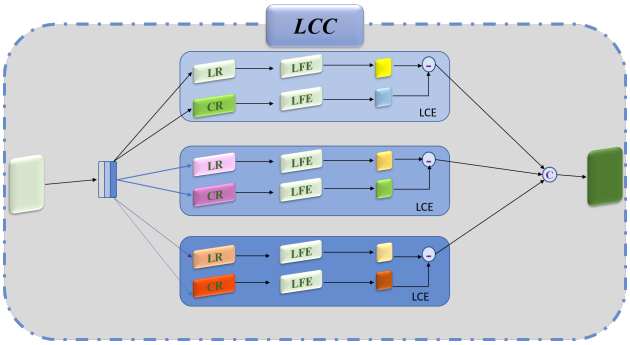

Figure 3: LCC module [11]

During the training process, two loss functions are used to guide segmentation and ensure the stability of predictions, each of which includes two additional loss functions. The two loss functions are:

- **Guided Feature Loss Function**

Contextual Affinity (CA) and Semantic Significance (SS) functions.

- **Consistency Loss Function**

Cross-View (CV) and Intra-View (IV) functions.

3.2 Input Data

The training dataset is defined such that Xn is the input, Yn is the annotation map, and Nimg is the total number of training images. Specifically in this context, Yn is represented as scribble lines where one represents the foreground, two represent the background, and zero represents unknown pixels.

3.3 Local Context Contrast Module (LCC)

Since camouflaged objects typically share different low-level features (such as texture, color, and intensity) with the background, detecting subtle differences is not easy. The visual suppression process in the retinas of mammals enhances clarity and contrast in visual responses by suppressing the activity of neighboring cells. LCC utilizes the capability that occurs in the human visual system known as visual suppression to capture and enhance low-level differences. The LCC module takes two low-level features (f1,f2) as input, which include texture, color, and intensity information processed through two branches of LCE with different receptive fields. Initially, the input feature Fin is reduced to 64 dimensions using a 1×1 convolution layer with batch normalization and ReLU activation.

The resulting feature($F_{\text{low}} \in 64 \times H \times W$) is then fed into three extractors (LCEs) aimed at focusing on different sizes of receptive fields. Each LCE consists of a Local Receptor (LR), a Context Receptor (CR), and two Local Feature Extractors (LFEs)(see Figure 3).

3.3.1 Local Receptor (LR):

The reduced-dimension feature enters a 3×3 convolution layer with a dilation rate of one, providing the extracted F_{local} to the LFE. This process covers adjacent input values.

3.3.2 Context Receptor (CR):

Similarly, the reduced-dimension feature enters a 3×3 convolution layer with a dilation rate of d_{context} providing the extracted contextual feature to the LFE. This process captures global and contextual information from the image due to its larger dilation rate. In the two LCEs, the dilation rates d_{context} in the context receptor are set to two, four, and six respectively, while in the local receptor, the dilation rate remains fixed at one to ensure that only local features are captured.

3.4 Enhanced texture module (TEM)

Enhancing texture in computer vision networks is of particular importance, as this capability helps the network effectively separate the background from camouflaged objects. Textures act as complex and rich patterns in images that can provide key information about local features and the structure of the image. By enhancing this aspect, we expect to achieve better results in identifying camouflaged objects, allowing the network to extract the edges of camouflaged items with greater accuracy. This precision in edge extraction not only aids in more accurate object identification but also leads to better differentiation from the background. As a result, enhancing texture recognition not only increases the accuracy and efficiency of the model but also enables deeper and more precise analysis of complex scenes (see Figure 4).

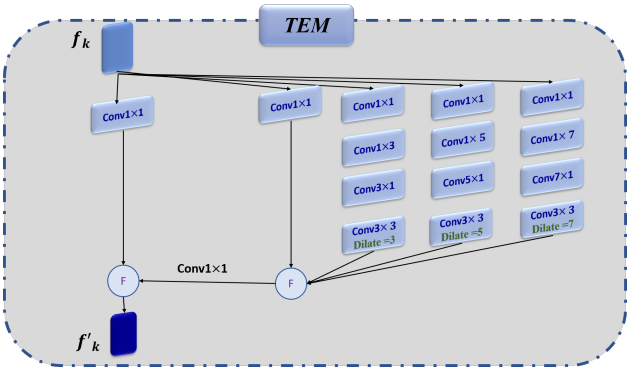

Figure 4: TEM module [8]

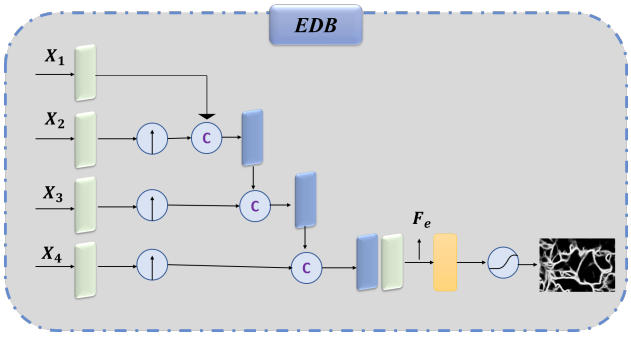

Figure 5: EDB module [17]

- **Parallel Branches (f_{bi}):** The TEM module consists of four parallel branches, each with different dilation rates (d = 1,3,5,7) (see the image above).
- **Convolution Layers**: Each branch includes various convolution layers, including a 1×1 convolution layer for reducing channel size and two convolution layers of (2i-1) × (2i-1) with different dilation rates.
- **Branch Combination**: The four branches are combined, and then the channel size is reduced with a 3×3 convolution layer.
- **Shortcut Branch**: A shortcut branch is used to preserve the original information in the model.
- **ReLU Function**: At the end of the module, a ReLU function is used to introduce non-linearity to the output.
- **Asymmetric Convolutions**: Instead of standard convolutions of (2i-1) × (2i-1), two asymmetric convolutions of (2i - 1) × 1 and 1 × (2i - 1) are utilized to improve the model's efficiency.

3.5 Edge Detection Boundary Module (EDB)

In the context of detecting camouflaged objects, enhancing contrast and texture is recognized as two key factors in improving identification accuracy. However, we have decided to add an additional mechanism (EDB) for edge detection to this process. This decision was made due to the challenges present in delineating the boundary between the target and the background. In many previous works, the unclear boundary between the target and background has been cited as a significant barrier to the accurate identification of camouflaged objects.

Edge detection allows us to extract more precise information about the structure and shape of objects. Edges typically represent abrupt changes in intensity or color and can serve as indicators of object boundaries. By utilizing edge detection mechanisms, we can identify important features that may be overlooked in the process of enhancing contrast and texture. The inputs to this module are the layers Xi = {1, 2, 3, 4} from the ResNet50 network. This module extracts coarse edge features and then estimates the coarse edge map. As shown in the upper left corner of (see Figure 5), each of the main features is processed by a 3×3 convolution block along with normalization and a ReLU activation function. The features are then upsampled to the same size. Finally, these features, which include rich details of edges and high-level semantic information, are gradually aggregated through concatenation operations and 1×1 convolution blocks, processed with normalization and a ReLU activation function. Ultimately, a 3×3 convolution block is applied to extract the coarse edge feature Fe. Additionally, a 3×3 convolution layer and a Sigmoid function are applied to Fe to produce the coarse edge map e. This map is used as a guide in the camouflage detection process within our architecture This approach aims to guide segmentation using these maps to improve target predictions.

3.6 Attention Module (CBAM)

By analyzing features in both channel and spatial dimensions, the CBAM can focus on important features while reducing unnecessary ones. This leads to the extraction of more precise information from images. The CBAM attention module is recognized as an effective tool for improving the performance of convolutional neural networks. This module simultaneously identifies and enhances important features using two types of attention: channel attention and spatial attention. The CBAM convolution block attention module provides a simple yet effective attention mechanism for feed-forward convolutional neural networks (see Figure 6).

Using an intermediate feature map, it sequentially infers attention maps in two separate dimensions: channel

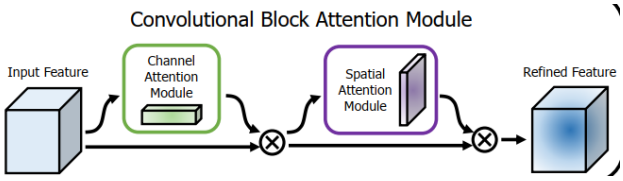

Figure 6: CBAM module [21]

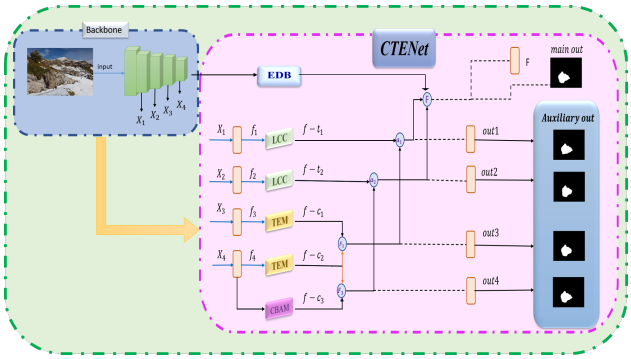

Figure 7: proposed CTENet

and spatial. These attention maps are then multiplied with the input features to improve adaptive features. Since CBAM is a lightweight and general module, it can be integrated into any CNN architecture with minimal costs and is trainable alongside base CNNs. These characteristics make CBAM an efficient tool for enhancing the performance of neural networks.

3.7 proposed Architecture of CTENet

Based on the ideas presented in previous sections, various experiments were conducted, and the results will be presented in the next chapter. Based on the results of these experiments, several concepts were added to the base architecture, resulting in a proposed architecture that we named CTENet, as shown in Figure 7. The overall structure of the architecture consists of four modules: the LCC module, the TEM module, the CBAM module, and the EDB module.

The Contrast Enhancement Module (LCC) is implemented with three sub-modules (LCE) with different dilation rates of 2, 4, and 6. Initial experiments were conducted with various dilation rates, and by combining these dilation rates of 2, 4, and 6, we achieved better results (see Figure 7).

4 Loss Functions

Scribble annotations complicate the learning process for accurately defining the boundaries of camouflaged objects due to limitations in the information they provide. To improve the accuracy and stability of predictions in the segmentation process, a new loss function called feature-guided loss has been proposed. This loss function is based on the semantic features extracted from the model and reduces computational overhead by weighting features according to their importance relative to the final prediction. To ensure consistency in predictions under various conditions and reduce inconsistencies in interpolation, a consistency loss function has been employed. This function not only examines the consistency of predictions across different scenes but also evaluates the consistency of predictions at each pixel in the feature map. With this approach, the ultimate goal is to enhance the accuracy and efficiency of deep learning models in identifying and segmenting objects.

4.1 Feature Guided Loss

Due to the use of scribble line-based methods and limited labeled data, many images remain unlabeled, resulting in uneven boundaries. To leverage the available information, the CRF[2] function has been proposed, which utilizes pixel features such as color and position; however, it has lower effectiveness in detecting camouflaged objects. For this reason, Feature Guided Loss has been designed to predict clearer boundaries in camouflaged object detection by using both simple and complex features. These two features are:

- Simple pixel features (contextual affinity)
- Complex features learned by a neural network (semantic significance)

4.1.1 CA Loss

The main idea of CA Loss is based on the premise that nearby pixels with similar features are usually grouped into similar categories; therefore, this loss function focuses on an n × n area for a specific pixel. Additionally, it employs the kernel method proposed in CRF Loss to measure the similarity of visual features such as colors and positions.

$$K_{\text{vis}}(i,j) = \exp\left(-\frac{\|S(i)-S(j)\|^2}{2\sigma_S^2} - \frac{\|C(i)-C(j)\|^2}{2\sigma_C^2}\right) \quad (1)$$

In the formula above, S(i) and C(i) represent the position and color of pixel i. σ_S and σ_C are hyperparameters. The concept is that similar pixels should have similar predictions, so the function Lca can be expressed as follows:

$$D(i,j) = 1 - P_i P_j - (1-P_i)(1-P_j) \quad (2)$$

$$L_{\text{ca}} = \frac{1}{M}\left(\sum_i \frac{1}{K_d(i)} \sum_{j \in K_d(i)} K_{\text{vis}}(i,j) D(i,j)\right) \quad (3)$$

D (i, j) calculates the probability that pixel i, j belongs to different classes. P(i, j) is the probability of positive labels for pixel i, j. $K_d(i)$ represents the n × n

[2]Conditional random field

neighbors centered around pixel i. M is the total number of pixels. Through Lca, the model can quickly learn from the entire image and produce relatively good predictions (P).

4.2 SS Loss

In this section, instead of using visual information such as color and position, a feature map (F) and position are utilized for the kernel exploitation method. By mimicking how humans detect objects and using higher semantic information, it can help improve the detection of camouflaged objects.

4.2.1 Covariance

The importance of each feature channel is determined by its covariance with the model's prediction and is calculated only on classified pixels.

$Sig_i = \mathrm{cov}(F_i, P), \quad i \in \{1, \ldots, C\}$ (4)

Then, the top N channels are selected based on this covariance to create a feature map with semantic information. To focus on boundary areas, pixels are classified, and if the prediction is above 0.8, they are considered valid classes.

$K_{\mathrm{sem}} = \exp\left(-\frac{\|S(i)-S(j)\|^2}{2\sigma_S^2} - \frac{\|\hat{F}(i)-\hat{F}(j)\|^2}{2\sigma_C^2}\right)$ (5)

$L_{\mathrm{ss}} = w_{\mathrm{ss}} \cdot \frac{1}{M}\sum_k \frac{1}{|R_k|}\sum_{(i,j)\in R_k} K_{\mathrm{sem}}(i,j)D(i,j)$ (6)

F_i is the feature mapping of channel i. R_k represents valid boundary areas, and wss is a hyperparameter that increases with the number of epochs since the model has not yet learned the displayed features well at the beginning.As a result, Feature Guided Loss can be expressed as the sum of both cost functions: $L_{\mathrm{ft}} = L_{\mathrm{ca}} + L_{\mathrm{ss}}$

4.3 Consistency Loss

Weakly supervised learning methods face challenges in detecting camouflaged objects due to the high visual similarity between the foreground and background, leading to inconsistent predictions. These methods do not perform satisfactorily in complex conditions, and self-supervised learning attempts to reduce this inconsistency by calculating the difference between the input network representation and treating it as constraint loss. Recently, weakly supervised methods have also utilized a similar cost function to improve prediction accuracy. However, there are still limitations to applying this cost function in line-based weak supervision; one challenge is that consistency and stability in the unit map are not considered. In this context, two proposed consistency functions are cross-view consistency and internal representation consistency.

4.3.1 Cross-View (CV) Loss Function

A model that performs well in object detection should be able to recognize the same objects in other images, even after transformations have been applied. To ensure this capability, cross-view loss is defined. For a neural network function $f_\theta(\cdot)$ with parameters , and some transformations $T(\cdot)$, with input x, the ideal state is defined as follows:

$f_\theta(T(x)) = T(f_\theta(x))$ (7)

The SSIM index is used to compare two images, and the final cross-view loss function is defined as follows:

$L_{\mathrm{cv}} = \frac{1}{M}\sum_{(i,j)}\left((1-\alpha)\frac{1-\mathrm{SSIM}(P_{(i,j)},\hat{P}_{(i,j)})}{2} + \alpha|P_{(i,j)} - \hat{P}_{(i,j)}|\right)$ (8)

$\alpha = 0.85$, P and $\hat{P}$ are the input prediction map and its transformation, respectively. M is the total number of pixels, and i, j are the indices of a pixel in each map.

4.3.2 Internal View (IV) Loss Function

The IV refers to the consistency of predictions within a prediction map, where the model's predictions should be stable and reliable within the specific range and features of an object. Near the boundaries, forcing the model to predict with confidence can be misleading. To guide the model toward confident predictions within the object, entropy is used, and a soft indicator is employed to filter out noisy predictions. The internal view consistency loss function is defined as follows:

$P_{\mathrm{entropy}} = \sum_{(i,j)}(-P\log P - (1-P)\log(1-P))$ (9)

$L_{\mathrm{iv}} = w_{\mathrm{iv}} \cdot \frac{1}{|I-B|} \cdot P_{\mathrm{entropy}}$ (10)

B is the set of all pixels close to the boundary, $(i,j) \in I - B$ wiv is the weight of the cost function, and it is typically set to 0.05 in practice. The entropy threshold for pixels near the boundary is 0.5. Note that this cost function is added in the final training stage when the predictions are relatively accurate. Finally, we define consistency loss as the sum of the two functions CV and IV:

$L_{\mathrm{cst}} = L_{\mathrm{cv}} + L_{\mathrm{iv}}$

4.4 Final Objective Function

The final objective function includes supervision for multiple outputs. For the main output, all introduced cost functions, along with the partial cross-entropy function, are combined to apply stronger supervision to the model.

$L_{\mathrm{pce}} = \frac{1}{N}\sum_{i\in\hat{P}}(-y_i\log\hat{y}_i - (1-y_i)\log(1-\hat{y}_i))$ (11)

It was observed that using the SS cost function did not significantly improve the performance of the auxiliary outputs (Out1 to Out4). Additionally, the CV function did not provide substantial improvements for these outputs either. Therefore, to achieve a balance

between efficiency and accuracy, it was decided to utilize the auxiliary loss only through IV consistency and Lca.

$L_{\text{iaux}} = L_{\text{ipce}} + L_{\text{ica}} + L_{\text{iv}}(i = 1, 2, 3, 4)$ (12)

where Lix is the cost function applied for auxiliary output i. Note that each output is sampled with two-dimensional interpolation to match the input size. Finally, the overall objective function for the output is defined as follows:

$L = L_{\text{cst}} + L_{\text{ft}} + L_{\text{pce}} + \sum_{i=1}^{n} B_i L_{\text{aux}}^{i}$ (13)

5 Experiments

5.1 Dataset

The first dataset based on annotations for camouflaged object detection (COD) is called S-COD, which we also utilized. This dataset includes 3,040 images from the COD10K training set and 1,000 images from the CAMO training set, with the remainder reserved for testing.

5.2 Evaluation Metrics

We use four evaluation metrics: Mean Absolute Error (MAE)[18], Structure Measurement (Sm)[6], Enhanced Measurement (Em)[7], and Weighted F Measure ($Fw\beta$)[1].

5.3 Implementation Details

In this method, the PyTorch deep learning library was used for implementation, and experiments were conducted on a powerful GeForce RTX 4090 GPU to achieve greater speed and efficiency. During the training phase, input images were prepared by applying transformations such as horizontal flips, random cropping, and resizing to 320×320 pixels to increase data diversity and prevent overfitting. For model optimization, we used the Stochastic Gradient Descent (SGD) algorithm with a momentum of 0.9, weight decay of 5e-4, and a triangular learning rate schedule with a maximum learning rate of 1e-3. The batch size was set to 16, and the number of training epochs was 150, with model training taking approximately 12 hours.

5.4 Result

We initially implemented the base architecture based on the article[11] and trained it. Finally, We decided to implement the changes that had improved the model and to run another round of tests. According to the various results obtained from multiple experiments, we concluded that we should re-examine the changes that improved the network more closely. In these changes, the CBAM and TEM modules showed significant impact; therefore, by combining these modules with the base architecture and increasing the expansion rates and

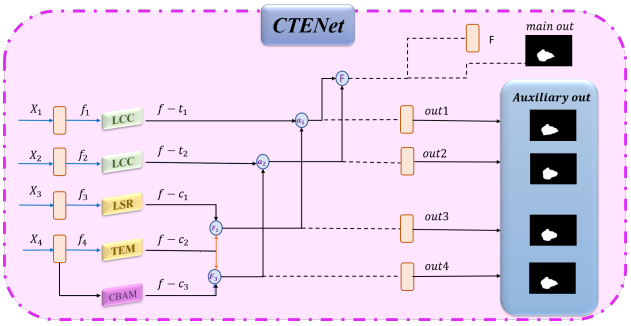

Figure 8: First Experiment

contrast enhancement within the range of (2, 4, 6), we arrived at a new architecture CTENet.

We conducted our experiments in two categories: the first involved combining the new modules with the base modules, and the second involved removing some of the base modules and replacing them with new modules(different experiments, see figure 9).

- **First Experiment:**

Contrast Enhancement: Three sub-modules of LCE were placed in LCC with expansion rates of (2, 4, 6).
LSR: The third-level feature was given to this module.
TEM: The fourth-level feature was given to this module.
CBAM: The fourth-level feature was given to this module.
Since all results indicate an improvement in scribble label performance, we attempted to conduct subsequent experiments using scribble labels (see figure 8).

Based on the previous experiments, numerous tests were conducted, leading to the final experiment that resulted in the proposed architecture CTENet7.

- **Final Experiment**

Contrast Enhancement: Three sub-modules of LCE were placed in LCC with expansion rates of (2, 4, 6).
LSR[3]: This module was removed, and TEM was added.
TEM: The fourth-level feature was given to this module.
EDB: Four features extracted from the network are provided to this module.
CBAM: The fourth-level feature was given to this module.

According to the last expriment, the results in Table 1, 2 , 3 show growth in some evaluation metrics compared to the base architecture. Enhancing texture is of higher importance in camouflage detection. Therefore, we proposed an architecture named CTENet,

[3]logical semantic relationship

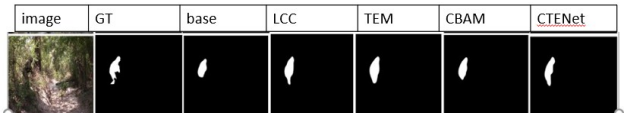

Figure 9: A sample of result with different expriment

Table 1: comparison based on the dataset CAMO

Method	CAMO			
	MAE	S	E	Fw
SS	0.092	0.735	0.815	0.641
SCWS	0.104	0.718	0.812	0.614
CRNet	0.092	0.735	0.815	0.641
ours	0.095	0.735	0.824	0.646

Table 2: comparison based on the dataset CHAMELEON

Method	CHAMELEON			
	MAE	S	E	Fw
SS	0.065	0.772	0.858	0.662
SCWS	0.055	0.785	0.890	0.683
CRNet	0.046	0.818	0.897	0.744
ours	0.046	0.814	0.896	0.736

Table 3: comparison based on the dataset COD10K

Method	COD10K			
	MAE	S	E	Fw
SS	0.065	0.678	0.764	0.469
SCWS	0.057	0.716	0.821	0.546
CRNet	0.049	0.733	0.832	0.469
ours	0.05	0.731	0.834	0.575

which considers the significance of structure and contrast alongside texture to potentially outperform current models. The findings indicate that texture enhancement plays a crucial role in camouflage detection, as it provides richer information about the local features of objects, aiding in better differentiation between the target and background. Additionally, the presence of the boundary enhancement module can help identify potential camouflage areas. Results presented in the latest article on weakly supervised camouflage detection demonstrate significant advancements in this field, not only regarding detection accuracy but also in the models' ability to handle existing challenges.

6 Conclusion

In response to the challenges of determining the ambiguous boundaries of camouflaged objects and backgrounds, a new architecture called CTENet has been proposed for camouflage detection in images. This architecture consists of four key modules: the Contrast Enhancement Module (LCC), the Texture Enhancement Module (TEM), the Object Boundary Enhancement Module (EDB), and the Channel Attention Module (CBAM). The proposed network first learns low-level features to extend lines into broader areas and then determines the actual foreground and background by analyzing texture using logically related semantic information. This network was trained on generated data (S-COD), resulting in improved performance compared to the baseline architecture.These results demonstrate the success of the CTENet architecture in enhancing camouflage detection accuracy in images.

7 Discussion

Future work will focus on producing a larger dataset for model training, We will also explore advanced attention mechanisms to improve feature prioritization in camouflage detection. Further examination of edge detection algorithms is essential to refine our ability to identify subtle boundaries in camouflaged objects. Additionally, we aim to investigate the application of metric learning techniques to enhance classification accuracy. The integration of quantum methods with detection algorithms presents an innovative approach that could improve processing efficiency. Lastly, developing techniques for camouflage detection in video will be crucial for real-time applications in military and security contexts.

Comments reviews: Most of the computational resources in this architecture are used in the backbone, and other modules (except for the backbone) do not require significant resources. Therefore, if MobileNet or EfficientNet is used in the backbone, this architec-

ture can be easily utilized in edge devices.To extend CTENet to video and real-time applications, object tracking ideas can be incorporated into this architecture so that a camouflaged object can be tracked in subsequent frames after being detected.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1597–1604. IEEE, 2009.

[2] H. Bi, C. Zhang, K. Wang, J. Tong, and F. Zheng. Rethinking camouflaged object detection: Models and datasets. *IEEE transactions on circuits and systems for video technology*, 32(9):5708–5724, 2021.

[3] Z. Chen, K. Sun, and X. Lin. Camodiffusion: Camouflaged object detection via conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1272–1280, 2024.

[4] H.-K. Chu, W.-H. Hsu, N. J. Mitra, D. Cohen-Or, T.-T. Wong, and T.-Y. Lee. Camouflage images. *ACM Trans. Graph.*, 29(4):51–1, 2010.

[5] P. Chudzik, A. Mitchell, M. Alkaseem, Y. Wu, S. Fang, T. Hudaib, S. Pearson, and B. Al-Diri. Mobile real-time grasshopper detection and data aggregation framework. *Scientific reports*, 10(1):1150, 2020.

[6] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017.

[7] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018.

[8] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao. Concealed object detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):6024–6042, 2021.

[9] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2777–2787, 2020.

[10] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 263–273. Springer, 2020.

[11] R. He, Q. Dong, J. Lin, and R. W. Lau. Weakly-supervised camouflaged object detection with scribble annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 781–789, 2023.

[12] T. He, Y. Liu, C. Xu, X. Zhou, Z. Hu, and J. Fan. A fully convolutional neural network for wood defect location and identification. *IEEE Access*, 7:123453–123462, 2019.

[13] Z. Huang, H. Dai, T.-Z. Xiang, S. Wang, H.-X. Chen, J. Qin, and H. Xiong. Feature shrinkage pyramid for camouflaged object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5557–5566, 2023.

[14] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto. Anabranch network for camouflaged object segmentation. *Computer vision and image understanding*, 184:45–56, 2019.

[15] M. Liu and X. Di. Extraordinary mhnet: Military high-level camouflage object detection network and dataset. *Neurocomputing*, 549:126466, 2023.

[16] Y. Mao, J. Zhang, Z. Wan, X. Tian, A. Li, Y. Lv, and Y. Dai. Generative transformer for accurate and reliable salient object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[17] Y. Niu, L. Yang, R. Xu, Y. Li, and Y. Chen. Minet: Weakly-supervised camouflaged object detection through mutual interaction between region and edge cues. In *ACM Multimedia 2024*, 2024.

[18] 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. IEEE, 2012.

[19] W. Shen, Z. Peng, X. Wang, H. Wang, J. Cen, D. Jiang, L. Xie, X. Yang, and Q. Tian. A survey on label-efficient deep image segmentation: Bridging the gap between weak supervision and dense prediction. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9284–9305, 2023.

[20] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu. Context-aware cross-level fusion network for camouflaged object detection. *arXiv preprint arXiv:2105.12555*, 2021.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[22] Y.-J. Xiong, Y.-B. Gao, H. Wu, and Y. Yao. Attention u-net with feature fusion module for robust defect detection. *Journal of Circuits, Systems and Computers*, 30(15):2150272, 2021.

[23] H. Zhang, L. Burrows, Y. Meng, D. Sculthorpe, A. Mukherjee, S. E. Coupland, K. Chen, and Y. Zheng. Weakly supervised segmentation with point annotations for histopathology images via contrast-based variational model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15630–15640, 2023.

[24] Q. Zhang, Y. Ge, C. Zhang, and H. Bi. Tprnet: camouflaged object detection via transformer-induced progressive refinement network. *The Visual Computer*, 39(10):4593–4607, 2023.

[25] P. Zheng, D. Gao, D.-P. Fan, L. Liu, J. Laaksonen, W. Ouyang, and N. Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *arXiv preprint arXiv:2401.03407*, 2024.

[26] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4504–4513, 2022.

[27] J. Zhu, X. Zhang, S. Zhang, and J. Liu. Inferring camouflaged objects by texture-aware interactive guidance network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3599–3607, 2021.