BRAUDE
College of Engineering, karmiel

Capstone Project Phase A

Camouflaged Object Detection with Diffusion Model

24-2-R-5

Students:

Hagar Tibi – 209063411 – Hagar.Tibi@e.braude.ac.il

Roi Darom – 313264822 – Roi.Darom@e.braude.ac.il

Supervisors: Dr. Renata Avros & Prof. Zeev Volkovich

Table of Contents

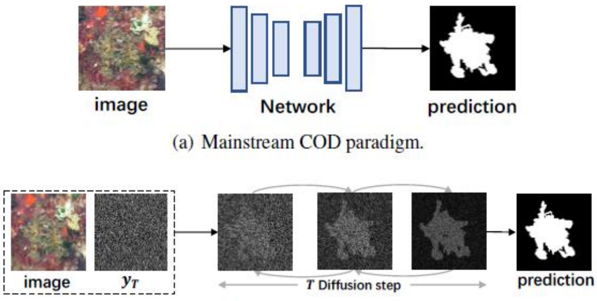
Abstract

This paper presents a novel approach to image restoration task using the diffCOD (Diffusion Camouflaged Object Detection) model. The framework utilizes diffusion models to address the challenge of restoring degraded images by treating it as a process of reversing noise effects. Through training, the model learns to manage types of noise and degradation levels by reversing the noise added to ground-truth images. During inference, the model refines noisy inputs into restored images. The framework includes an Injection Attention Module (IAM) to support the denoising process by incorporating conditional semantic features, and a Feature Fusion (FF) module to combine multi-scale features, helping to maintain and improve image details. This approach aims to enhance noise removal and image quality in the restoration process. This approach is particularly relevant in fields such as medical imaging, remote sensing, and photography, where high-quality image restoration is critical for accurate analysis and visualization.

1. Introduction

Camouflaged object detection (COD) identifies objects blending seamlessly with their surroundings, posing a challenge due to their high similarity with backgrounds. This task is vital in fields like agricultural pest detection, medical image segmentation, and industrial defect detection. Traditional COD methods, often inspired by human vision, utilize convolutional neural networks and auxiliary cues to improve accuracy. However, these methods struggle with the complex nature of camouflaged objects. Recently, diffusion models have excelled in tasks like image synthesis by learning the reverse diffusion process, though their use in COD is not well-explored. diffCOD approaches COD as a denoising diffusion process, adding Gaussian noise to ground-truth masks during training, which the model learns to reverse. During inference, it refines noisy masks into accurate segmentation through forward-and-reverse diffusion steps. diffCOD enhances denoising by incorporating encoded input image priors and integrating semantic features using a cross-attention-based Injection Attention Module (IAM).

In addition to camouflaged object detection, this project aims to modify the diffCOD model to restore and enhance image quality of damaged images. By applying the denoising diffusion process, the model can effectively reduce various types of noise and interferences, resulting in clearer and higher-quality images. This enhancement is crucial for applications where image quality is compromised due to aging, environmental factors, or other sources of degradation.



(a) Mainstream COD paradigm.

(b) Diffusion-based COD paradigm.

***Figure 1**. mainstream COD approach vs Diffusion based approach [1]*

2. Related Work

2.1. Camouflaged Object Detection Models

Camouflaged object detection (COD) involves identifying objects that blend into their surroundings. Traditional COD methods, which are non-generative, segment objects directly from the background using various strategies. These strategies often focus on enhancing feature representation and improving segmentation accuracy through:

2.1.1. Edge Semantic Information

Methods such as BGNet for edge semantic information highlight object structures and boundaries by introducing additional cues. Edge semantic information involves identifying and emphasizing the edges or boundaries of objects within an image. BGNet (Boundary Guided Network) leverages this information to enhance object detection and segmentation by providing clearer delineation of objects. This is especially useful in complex scenes where precise edge detection is crucial for accurate object recognition.

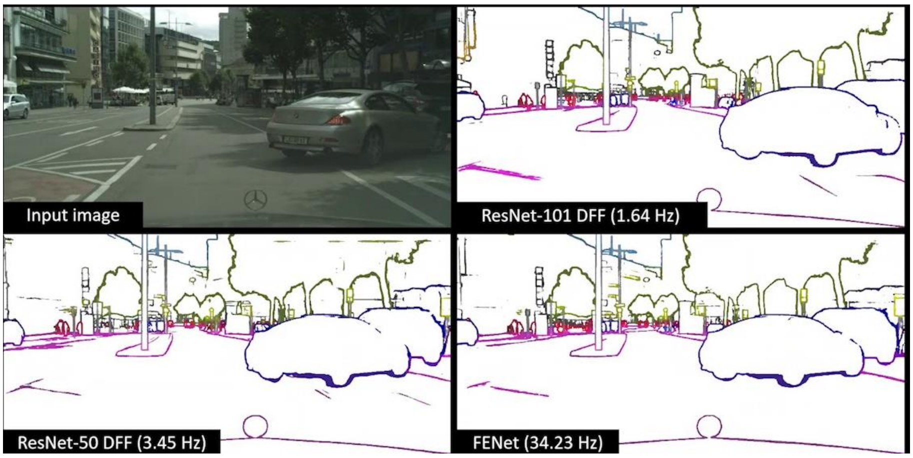


***Figure 2**. Semantic Edge Detection Network Example [2]*

2.2. Frequency Domain Features

Methods that utilize frequency domain features enhance detection accuracy by focusing on specialized information, like diffCOD's use of encoded input image priors to improve mask refinement. Frequency domain features, as implemented in methods like FDCOD, enhance detection accuracy by analyzing the frequency components of an image. These methods transform image data into the frequency domain, allowing for the examination of periodic patterns and structures that may not be easily visible in the spatial domain. By leveraging these features, they can identify and emphasize important characteristics, leading to more accurate object detection.

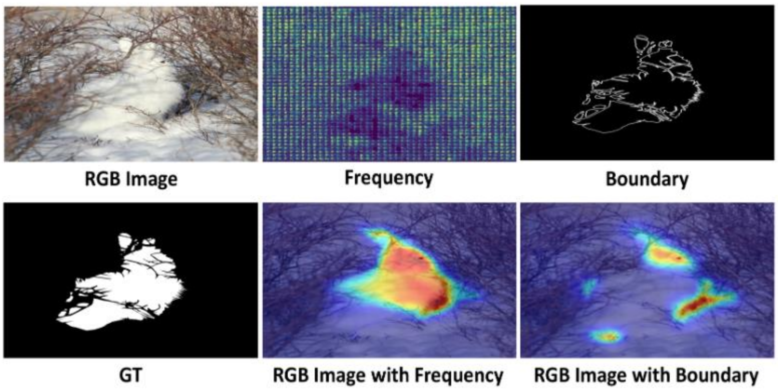


Figure 3*. FDCOD* with *different additional cues.* [3]

2.2.1. Multi-task Learning

Methods that utilize multi-task learning enhance detection accuracy by performing multiple related tasks simultaneously, which can lead to improved performance on each individual task. By sharing information across tasks, these methods can learn more robust and generalized features. For example, SegMaR leverages multi-task learning to enhance object detection by combining segmentation and recognition tasks.

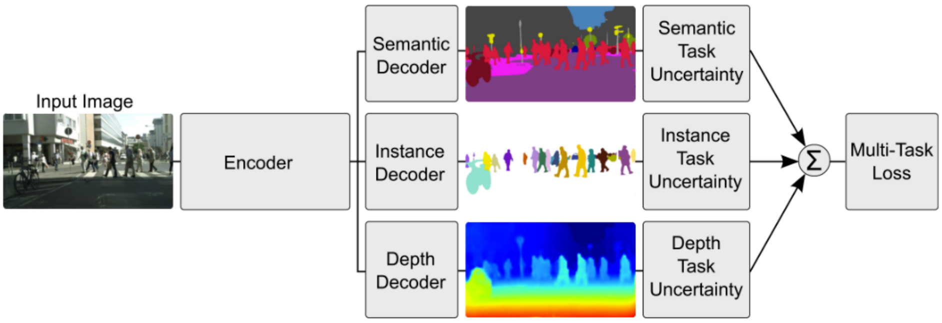


Figure 4*. Uncertainty-based loss function weighting for multi-task learning . [4]*

2.2.2. Transformer-based Methods

Transformer is A type of deep learning model that has revolutionized the fields of natural language processing (NLP) and, more recently, computer vision. It is designed to handle sequential data and can capture long-range dependencies through a mechanism known as attention.

FSPNet, for example, employ transformers to process the input data in a way that accounts for both local and global information. This enables them to effectively identify and differentiate between objects, even in complex and cluttered environments.

Transformer-based methods are also highly effective for image restoration. By capturing long-range dependencies and relationships within the data, these models can maintain consistency and coherence across the entire image. This is crucial for tasks such as noise reduction and detail enhancement, ensuring that the restored images retain both local details and global context, resulting in higher visual fidelity and overall quality.

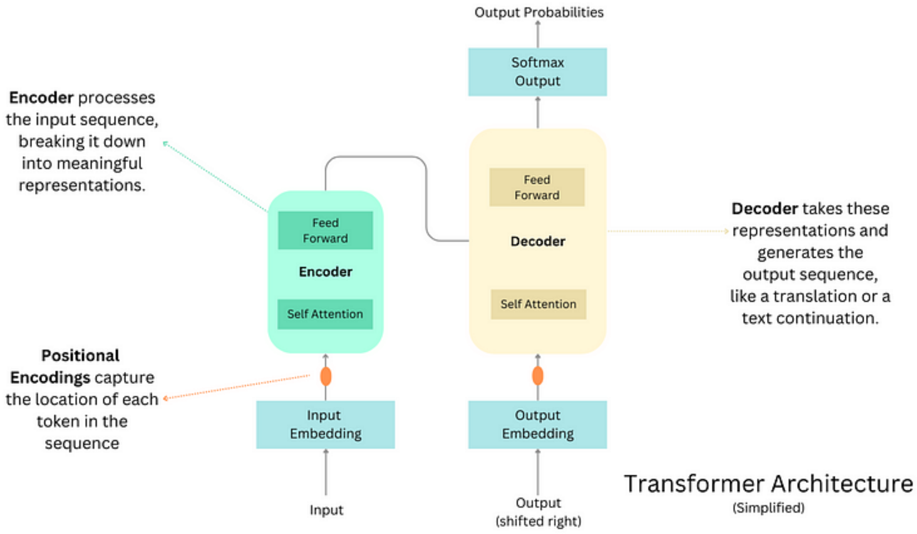

Output Probabilities
Softmax Output
Encoder processes the input sequence, breaking it down into meaningful representations.
Feed Forward
Encoder
Self Attention
Feed Forward
Decoder
Self Attention
Decoder takes these representations and generates the output sequence, like a translation or a text continuation.
Positional Encodings capture the location of each token in the sequence
Input Embedding
Output Embedding
Input
Output (shifted right)
Transformer Architecture
(Simplified)

Figure 5*. transformer architeture simplified. [5]*

2.3. Current Diffusion methods

Existing COD methods often struggle with accurate segmentation in complex scenarios. To address these challenges, generative models, specifically denoising diffusion models, are introduced into the COD task. These models progressively refine object masks from noisy images, achieving exceptional performance, particularly for objects with fine textures. Recent advances in diffusion models have shown promising results in various segmentation tasks

2.3.1. MedSegDiff

MedSegDiff is a diffusion-based method designed for medical image segmentation. By leveraging diffusion models, it improves segmentation accuracy by capturing complex patterns and structures within medical images, which are often challenging to delineate using traditional methods. This approach allows for more precise and detailed segmentations, enhancing the overall diagnostic capabilities in medical imaging.

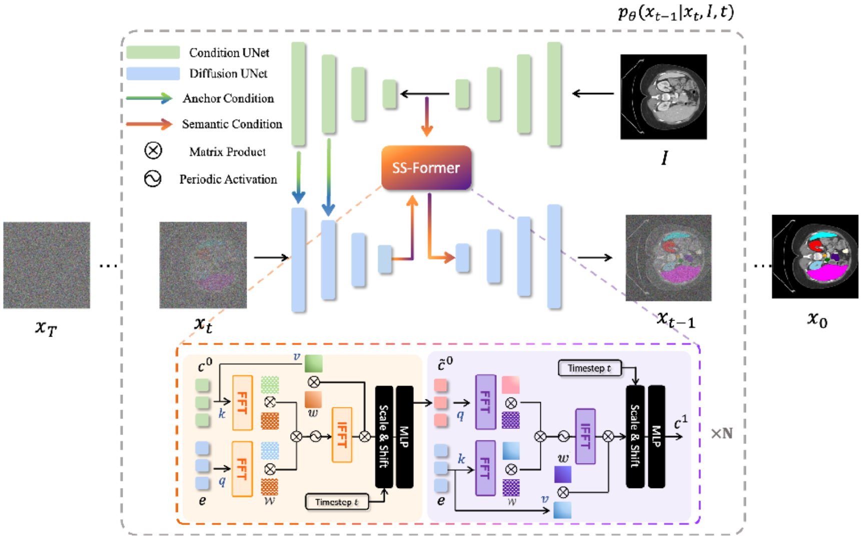


Figure 6. *Diffusion Based medical image segmentation. [6]*

2.3.2. ODISE

ODISE combines a trained text-image diffusion model with a discriminative model to achieve open-vocabulary panoptic segmentation, allowing for broader segmentation capabilities. This method enables the segmentation of a wide range of object categories by leveraging the strengths of both generative and discriminative approaches, thus enhancing the versatility and accuracy of the segmentation process in diverse scenarios.

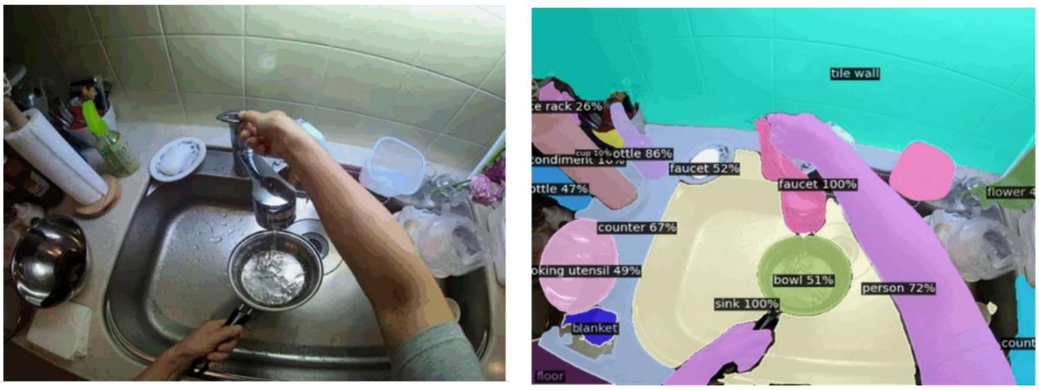
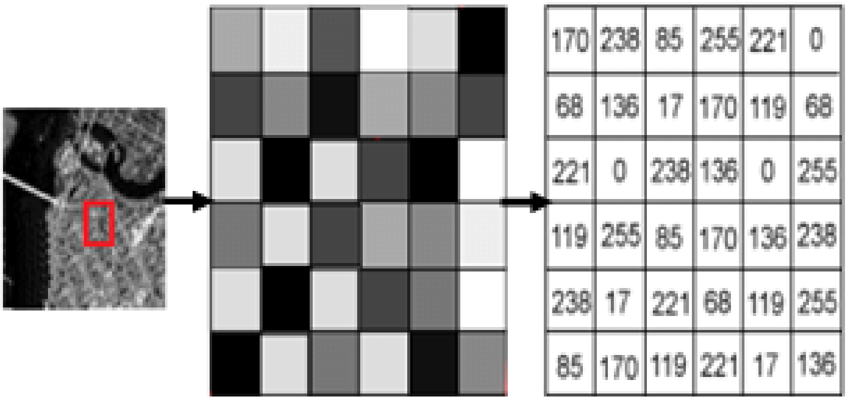


Figure 7. *visual result of Open-vocabulary DIffusion-based panoptic Segmentation [7]*

3. Background

3.1. Image representation

Image representation involves converting a picture into a format that computers can process and understand. Pictures consist of tiny dots called pixels, each with a specific color or brightness value. These pixels are organized in a grid format that defines the image's dimensions and color information, such as the RGB format for color images. In computational terms, an image can be represented as a two-dimensional array arranged in rows and columns, with x and y coordinates representing the pixel values, typically ranging from 0 (black) to 255 (white). The goal of image representation is to extract significant details from the picture, enabling the computer to perform tasks like object recognition, image classification, or segmentation.

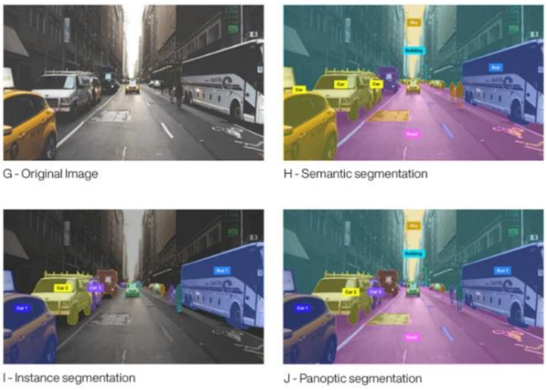


170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Figure 8*. image representation by pixels [8]*

3.2. Image segmentation

Image segmentation is a crucial task in computer vision that involves partitioning an image into multiple segments or regions, each representing different objects or areas within the scene. This process transforms a pixel-level representation of an image into a higher-level abstraction, facilitating tasks such as object detection, recognition, and analysis. Techniques for image segmentation can be broadly categorized into classical methods and modern deep learning approaches. Classical methods include thresholding, edge detection, and region-based techniques, which rely on pixel intensity and local features. In contrast, deep learning-based methods, particularly convolutional neural networks (CNNs) and transformer models, leverage hierarchical feature extraction and end-to-end learning to achieve superior accuracy and robustness. Advanced techniques such as fully convolutional networks (FCNs), U-Net, Mask R-CNN, and semantic segmentation models incorporate multi-scale feature (multi-scale features capture information at various levels of detail within an image, from fine textures to broader structures, enabling more comprehensive analysis and understanding) learning, skip connections, and contextual information to effectively segment complex scenes with high precision.


G - Original Image

H - Semantic segmentation

I - Instance segmentation

J - Panoptic segmentation

Figure 9*. Differenet types of image segmentation. [9]*

3.3. Image restoration

Image restoration is a fundamental task in computer vision that involves reconstructing or recovering an image that has been degraded by noise, blur, or other distortions. The objective is to restore the image to its original or near-original state, thereby enhancing its quality and making it suitable for further analysis and processing. Image restoration techniques are crucial for various applications, including medical imaging, satellite imagery, historical document preservation, and general photography. Image restoration techniques aim to reverse image degradations using various algorithms and models, categorized into classical methods and modern deep learning approaches. Classical methods include filtering techniques like mean, median, and Wiener filtering to reduce noise and enhance quality; deconvolution to reverse blurring; and interpolation methods such as bilinear and bicubic to restore lower resolution images. Modern deep learning approaches encompass convolutional neural networks (CNNs) that learn complex patterns to remove noise and artifacts, autoencoders that compress and reconstruct images, and generative adversarial networks (GANs) that generate high-quality restored images by learning the distribution of clean images.

3.4. Diffusion Mathematical Background

The diffusion probability model has reaped plenty of attention due to its simple training process and excellent performance. It is mainly divided into forward process and reverse process.

3.4.1. Markov Chain

A Markov chain is a mathematical system that undergoes transitions from one state to another within a finite or countable number of possible states. It is characterized by the Markov property, which states that the future state of the process depends only on the present state and not on the sequence of events that preceded it. Formally, this can be expressed as:

$$P(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_{n+1} = x \mid X_n = x_n)$$

where X_n represents the state at step n.

A key feature of Markov chains is their use in both forward and reverse processes.

In the context of diffusion models, Markov chains are utilized to iteratively refine noisy images through a series of probabilistic transitions. The forward process involves adding Gaussian noise to the data, and the reverse process, modeled as a Markov chain, progressively removes this noise. This denoising process can be described by a sequence of states $\{X_t\}$ where X_0 is the original data, X_t is the noise, and intermediate states represent the noisy versions of the data. The goal is to learn the transition probabilities that effectively reverse the noise addition, leading to the recovery of the original data from the noisy observations.

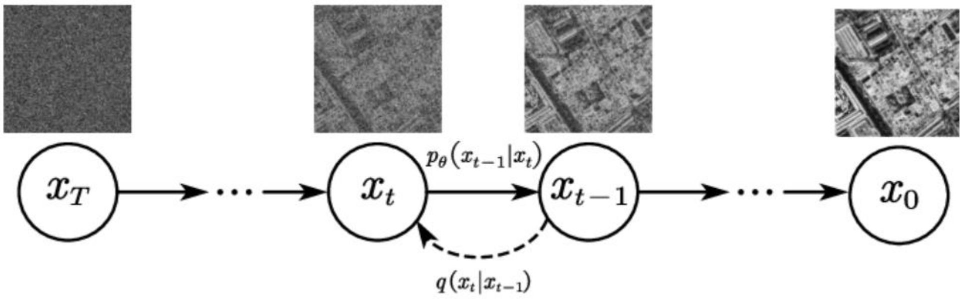


Figure 10. *Denoising diffusion model bases on Markov chain, where q and t represent adding and removing noise respectively. [10]*

3.4.2. Noise

Noise is essentially disturbances that obscure or interfere with the intended signal for certain data. Noise is created by environmental sources or electronic interference. In the field of image processing, noise manifests as unwanted pixel values, leading to loss of brightness and details in the image.

- Gaussian Noise

Gaussian (normal) distributed random noise is applied to the gray values of original images. Gaussian noise is commonly used in modelling scenarios due to its mathematical properties and prevalence in real-world noise sources. The mathematical properties of this distribution allow averaging over many pixels to help detect and cancel out the noise.

- Speckle Noise

Speckle noise (Multiplicative Noise) is commonly found in radar and medical imaging, particularly in ultrasound images. It is a granular noise that occurs due to the interference of multiple wave reflections. Unlike Gaussian noise, which is additive, speckle noise is multiplicative, meaning it affects the intensity of the pixels proportionally.

- Salt-and-Pepper Noise

This type of noise manifests as random occurrences of black and white pixels within an image, resembling the look of salt and pepper. It is caused by sudden and sharp disturbances in the image signal, such as faulty memory locations or malfunctioning pixels in camera sensors.

3.4.3. Mask

A binary or multi-class image that indicates specific regions of interest within the original image. Each pixel in the mask corresponds to a pixel in the original image and holds a value that signifies whether that pixel belongs to the objects of interest (foreground) or the background.

3.4.4. Forward Process Formulation

The diffusion model operates through two main processes: the forward process and the reverse process. In the forward process, Gaussian noise with variance $\beta_t \in (0,1)$ is gradually added to the original image x_0 transforming it step by step until it becomes a completely noisy image that resembles an isotropic Gaussian distribution (symmetric around its mean). This process is described by the equation:

$$\mathrm{q}(x_t \mid x_{t-1}) = N\left(x_t, \sqrt{1-\beta_t} x_{t-1}, \beta_t I\right)$$

breakdown of the parameters:

- x_t: The noisy image at time step t.
- x_{t-1}: The noisy image at the previous time step t−1.
- $\mathrm{q}(x_t \mid x_{t-1})$: the probability of moving to state x_t given the previous state x_{t-1}.
- N: Denotes a Gaussian distribution.
- $\sqrt{1-\beta_t} x_{t-1}$: The mean of the Gaussian distribution, which is a scaled version of x_{t-1}.
- $\beta_t I$: The covariance matrix of the Gaussian distribution, with β_t controlling the amount of noise added at each step and I being the identity matrix indicating isotropic (equal in all directions) noise.

The latent variable x_t can be directly obtained from the original image x_0 using:

$$\mathrm{q}(x_t \mid x_0) = N\left(x_t, \sqrt{1-\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I\right)$$

breakdown of the parameters:

- x_t: The noisy image at time step t.
- x_0: The original image.
- $\mathrm{q}(x_t \mid x_0)$: The probability distribution of x_t given the original image x_0.
- $\sqrt{1-\bar{\alpha}_t} x_0$: The mean of the Gaussian distribution, which is a scaled version of the original image x_0.

- $(1-\bar{\alpha}_t)I$: The covariance matrix of the Gaussian distribution, indicating the amount of noise added to the original image.

here $\bar{\alpha}_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$. that means x_t can be seen as a noisy version of x_0, where the noise level increases with time t.

This project transitions from camouflaged object detection (COD) to image restoration by leveraging a diffusion model to understand and reverse noise effects. The goal is to restore images to their near-original state. The forward process, which adds Gaussian noise to simulate degradation, is crucial for training the model to recognize noise patterns and their impact on image features. Parameters like β_t and $\bar{\alpha}_t$ play a key role in preserving important features while adding noise. During training, the model learns to denoise and restore images through the reverse diffusion process, effectively handling various noise types and degradation. This controlled noise addition is essential for developing a robust framework that achieves clean, artifact-free image restorations, ensuring the model can handle diverse degradation scenarios and produce high-quality results.

3.4.5. Reverse Process Formulation

The reverse process involves converting the noisy image back into the original image through a series of steps. This is formulated as:

$$p_\theta(x_{t-1}|x_t) = N\left(x_{t-1}; \mu_\theta(x_t, \mathrm{t}), \sum\nolimits_\theta (x_t, t)\right)$$

Here $p_\theta(x_{t-1}|x_t)$ is the probability of transitioning back to state x_{t-1} from x_t with μ_θ and Σ_θ being learned parameters that adjust the mean and variance of the distribution.

breakdown of the parameters:

- x_t: The noisy image at time step t.
- x_{t-1}: The noisy image at the previous time step t−1.
- $\mu_\theta(x_t, \mathrm{t})$: The mean function parameterized by θ, which predicts the mean of the Gaussian distribution for the reverse step.
- $\sum_\theta(x_t, t)$: The covariance (variance) function parameterized by θ, which predicts the variance of the Gaussian distribution for the reverse step.

The combination of the forward process q and the reverse process p forms a variational auto-encoder. The objective is to minimize the variational lower bound (VLB), which measures the difference between the true data distribution and the model distribution. The VLB is defined as:

$$L_{vlb} = L_0 + L_1 + \cdots + L_{T-1} + L_T$$

Each term in the VLB represents a different aspect of the learning process:

- $L_0 \coloneqq -log p_\theta(x_0|x_1)$
  measures the likelihood of the original image x_0 given the first step in the reverse process, x_1. It ensures that the reverse process starts off correctly reconstructing x_0 from the initial noisy image.

- $L_{t-1} := D_{KL}\big(q(x_{t-1}|x_t, x_0) \;||\; p_\theta(x_{t-1}|x_t)\big)$

  This is the KL divergence between the true posterior distribution $q(x_{t-1}|x_t, x_0)$ and the learned reverse process distribution $p_\theta(x_{t-1}|x_t)$. It measures how well the reverse process matches the forward process at each step t.

- $L_T := D_{KL}\big(q(x_T|x_0) \;||\; p(x_T)\big)$

  measures how well the final noisy image x_t matches the assumed Gaussian distribution. It ensures that the final noisy state is consistent with the Gaussian noise model.

3.5. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a specialized type of artificial neural network designed primarily for analyzing visual data. CNNs are widely used in image and video recognition, as well as in other areas such as natural language processing. This network mimics the structure of neurons in the human brain and is highly effective for tasks like image segmentation and classification.

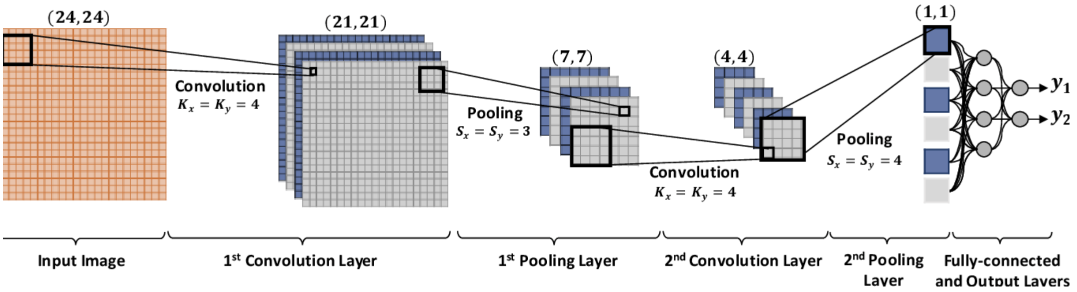


Figure 11. *Convolution Neural network [11]*

The architecture of a CNN comprises five key components:

Input Layer:

This layer represents the image based on its pixel values, typically divided into RGB (Red, Green, and Blue) channels.

Convolution Layer:

This primary layer extracts feature maps from the image. During each iteration, a filter of size H×W×D (height, width, depth) is initialized to represent the weights. The filter convolves over the image, computing dot products and connecting pixels into a single neuron. An activation function is then applied to each neuron to facilitate visual representation and user comprehension.

Hyperparameters used in the Convolution Layer:

Stride: Determines the step size between filter applications over the image, controlling the displacement of the filter across the image's length and width.

Padding: Adds extra pixels, usually zeros, at the image borders to preserve spatial dimensions and maintain edge features.

Batch Size: Specifies the number of samples passing through the network in each forward/backward iteration, affecting training stability and speed.

Activation Function: Introduces non-linearity to each neuron, enabling the network to capture and represent complex relationships among data pixels, thus enhancing feature maps.

Pooling Layer: This layer completes the feature extraction process by reducing the dimensions of the feature maps, making them more manageable.

Fully Connected Layer: Combines and normalizes the features extracted from previous layers. It contains neurons that connect to the entire input volume.

Output Layer: Composed of neurons arranged in a grid, with the number of neurons determined by the task requirements. Each neuron represents the probability of the input pixel belonging to a specific category.

3.6. U-net Architecture

U-Net is a convolutional neural network architecture specifically designed for biomedical image segmentation. The network is structured in a symmetric U-shape, consisting of a contracting path (encoder) and an expansive path (decoder).

- Encoder Path: The downsampling part of U-Net that reduces the image size while extracting features using convolutional and max pooling layers.
- Decoder Path: The upsampling part of U-Net that reconstructs the image to its original size, refining features with upsampling and convolutional layers.
- Bottleneck: The central part of U-Net where the feature maps are most compressed, serving as the transition between the encoder and decoder.
- Skip Connections: Links between corresponding layers in the encoder and decoder, allowing high-resolution features from the encoder to be combined with the upsampled features in the decoder, preserving spatial information.

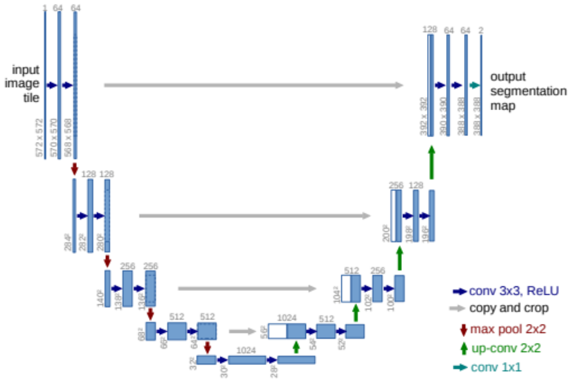


Figure 12. *Architecture of U-net model [12]*

Now let us look at the flow of the U-net network:

- Encoder Path:

  Structure: Composed of multiple convolutional layer blocks with max-pooling layers in between.

  Function: Captures high-level features and reduces the spatial dimensions of the input image through down-sampling operations.

- Decoder Path:

  Structure: Composed of multiple convolutional layer blocks with up-sampling layers in between.

  Function: Increases the spatial dimensions of the feature maps, expanding them back to the original image size.

- Skip Connections:

  Function: Connects corresponding layers in the encoder and decoder paths, allowing the network to recover spatial details lost during down-sampling by directly concatenating feature maps from the encoder to the decoder.

- Final Layer:

  Structure: Typically, a 1x1 convolutional layer.

  Function: Reduces the number of channels to produce a final segmentation map with the same dimensions as the input image. This is followed by an activation function for semantic segmentation tasks.

- Training:

  Loss Function: The loss function (also known as a cost function or objective function) is a mathematical function that takes in the model's predictions and the actual values and outputs a single number representing the difference (or error). Lower values of the loss function indicate better model performance.

  The Parameters that used to Improve the Loss:

  Model Architecture: Choosing and fine-tuning the structure of the model (e.g., number of layers in a neural network, types of layers).

  Data Augmentation: Enhancing the training dataset with transformations like rotations, flips, and scaling to improve generalization.

  Cross-Validation: Using different subsets of data to validate the model performance and ensure it generalizes well to unseen data.

  Hyperparameter Tuning: Systematically searching for the best hyperparameters (e.g., using grid search, random search, or Bayesian optimization).

3.6.1. U-net block

The U-Net block utilizes convolutions like a standard CNN block. However, instead of applying a single filter, the U-Net block performs multiple convolutions (typically 2-3) followed by activation functions to learn more complex features. The result of these convolutions is a feature map that captures specific aspects of the image based on the applied filters.

Each U-Net block comprises two identical components:

1. Convolutional: A 3x3xC_{out} filter is applied to the image. In the encoding layer, the number of feature maps increases, while in the decoding layer, they decrease. The filter weights are adjusted during model training.

2. Activation Function (ReLU): The Rectified Linear Unit (ReLU) activation function returns 0 for negative inputs and keeps positive inputs unchanged. This introduces non-linearity with minimal computational overhead.

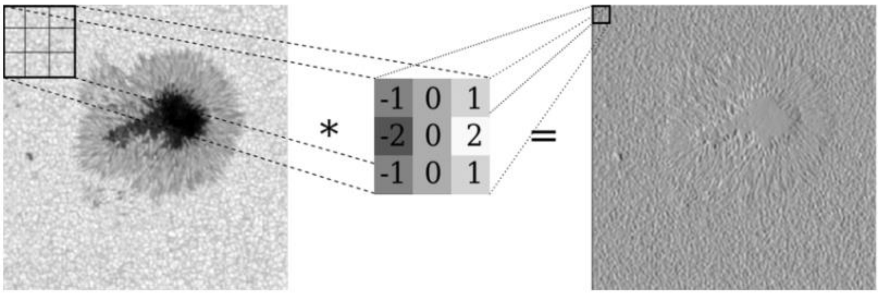


***Figure 13**. convolution with a filter [13].*

3.6.2. Up-sampling and Down-sampling

Up-sampling and Down-sampling are techniques used to alter the size of an image. Up-sampling, also known as pooling, reduces the image size by half, while down-sampling increases the image size.

1. Down-Sampling: Typically performed using max-pooling. This method involves applying a filter of a specific size to the image and selecting the maximum value within each filter region.
2. Up-Sampling: Several methods can be employed to increase the image size:
    a. Nearest Neighbors: The pixel values are duplicated to adjacent pixels based on the filter size.

b. Bed of Nails: Pixels are placed at specific intervals determined by the filter size, consistently across the entire image.
c. Max Up-Pooling: The maximum value chosen during down-sampling is placed at the same index during the up-sampling step.
d. Pixel Shuffle: Pixels within each feature map are rearranged. Each set of
e. $r \times r$ elements in the channel (where r is the scaling factor) is reshaped into a single element in a new, enlarged height and width dimension.

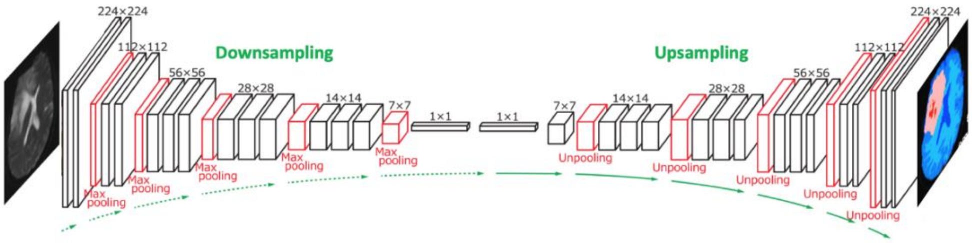


Figure 14. *Up and Down sampling [14]*

3.7. Framework Components

The diffCOD framework utilizes a diffusion model to tackle the camouflaged object detection (COD) task. The core denoising network of diffCOD is built on a U-Net architecture. To obtain robust conditional semantic features, the model integrates multi-scale features extracted from a Vision Transformer (ViT) backbone, combined through a feature fusion (FF) process. This approach ensures that the resulting features are rich in multi-scale details. Furthermore, to enable the texture patterns and localization information in the conditional semantic features to effectively guide the denoising process, an injection attention module (IAM) based on cross-attention is introduced. This IAM reduces the discrepancy between diffusion features and image features, leveraging the strengths of both to enhance performance.

3.7.1. Vision Transformer (ViT)

The Vision Transformer (ViT) is an architecture that adapts the transformer model, originally designed for natural language processing, to image processing tasks. It divides an image into a sequence of smaller patches. Each patch is flattened and embedded into a fixed-size vector, serving as an input token for the transformer model.

The ViT process begins with embedding these image patches and adding positional encodings to retain spatial information. These embeddings are then passed through multiple transformer layers, which include multi-head self-attention mechanisms and feed-forward neural networks. The self-attention mechanism allows the model to capture long-range dependencies and contextual relationships across different parts of the image, effectively modelling global information.

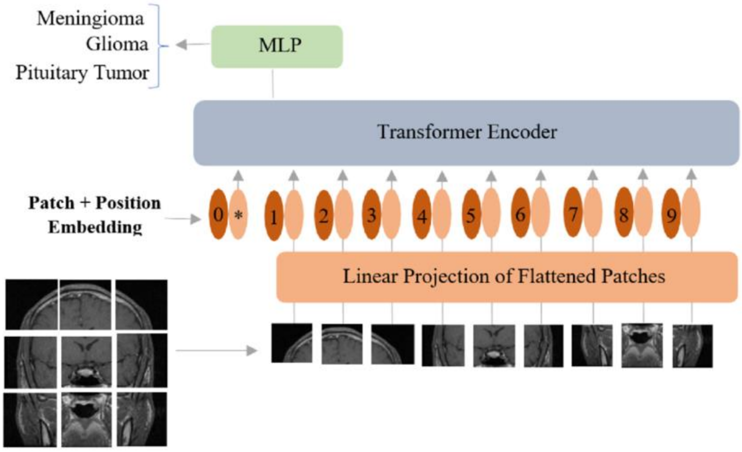

Meningioma
Glioma
Pituitary Tumor
MLP
Transformer Encoder
Patch + Position Embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 15. *Vision Transformer mode adopted for classification of brain tumors from MRI. MLP (multilayer perceptron): is the extra learnable patch embedding to be used by the final classification head. [15]*

The primary purpose of ViT is to leverage the power of transformers in capturing complex patterns and relationships across an entire image. This approach leads to improved performance in various vision tasks, such as image classification, segmentation, and object detection, by providing a global understanding of the image content.

3.7.2. Feature Fusion (FF)

Feature Fusion (FF) is a technique used in machine learning and computer vision to combine features from multiple sources or scales to enhance the performance of a model. In image processing, features at different scales capture varying types of information. Low-level features capture edges and textures, while high-level features capture shapes and objects. By integrating these multi-scale features, feature fusion combines fine details with broader contextual information, enriching the overall representation. This integration is especially useful in tasks such as image restoration, segmentation, and object detection, where both local and global contexts are crucial.

Feature fusion employs techniques like concatenation, addition, weighted sum, and attention mechanisms to combine features. Concatenation joins features from different sources or scales along a specified dimension, while addition combines them elementwise, maintaining the same dimensionality. The weighted sum approach involves learned weights during training, allowing dynamic adjustment of each feature set's importance. Attention mechanisms weigh features based on relevance, focusing the model on the most critical aspects.

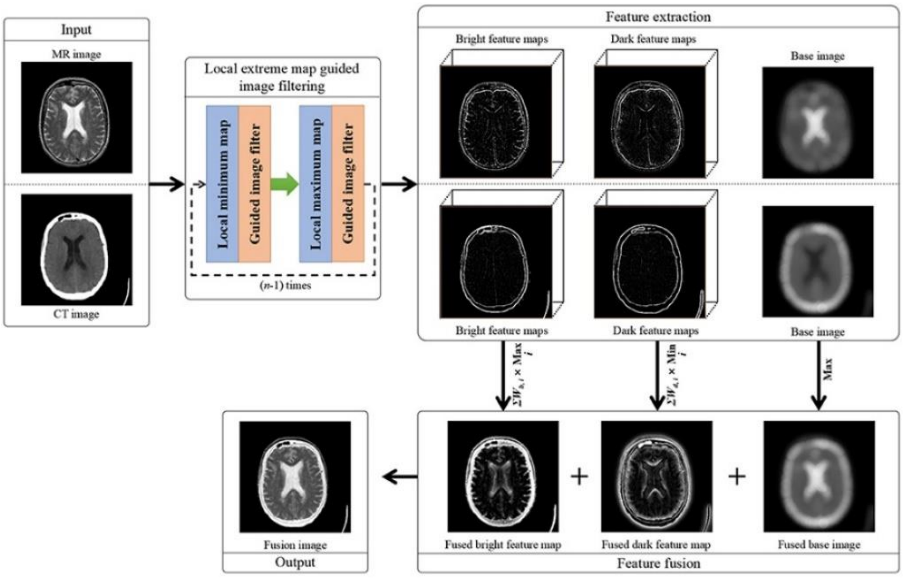


Figure 16. *fusing MR and CT images using local extreme map guided filtering, feature extraction, and feature fusion (FF) to enhance image quality by combining bright, dark, and base feature maps. [16]*

In the diffCOD model, FF processes the multi-scale features extracted by the Vision Transformer (ViT) backbone. The FF module has three branches, each handling one of the feature scales (Xp1, Xp2, Xp3). Each branch applies two convolution operations to enhance the features. The features from the three branches are then aggregated using a single convolution operation, resulting in a unified feature map. By leveraging FF, the diffCOD model can effectively combine multi-scale and complementary features to create richer and more robust representations. This enhanced representation significantly improves the model's performance in detecting and segmenting camouflaged objects.

Modifying the FF module to better support image restoration and enhancement involves optimizing the combination and utilization of multi-scale features. Fine-tuning convolution operations and adjusting aggregation strategies in the FF module improve integration of multi-scale features, aiding in the identification and correction of noise and artifacts. This results in cleaner, higher-quality restorations. Dynamic weight adjustment techniques, like learned weights or attention mechanisms, prioritize relevant features and preserve crucial details. Enhancing convolutional operations and incorporating adaptive feature weighting mechanisms further improve sharpness, clarity, and visual fidelity. This approach ensures high-quality, accurate, and efficient processing across applications, including medical imaging and general image enhancement tasks.

3.7.3. Injection Attention Module (IAM)

The Injection Attention Module (IAM) enhances the denoising process in the diffusion model by integrating texture and location information from the original features. The IAM, positioned in the middle of the U-Net-based denoising network and uses cross-attention mechanisms to merge multi-scale features effectively.

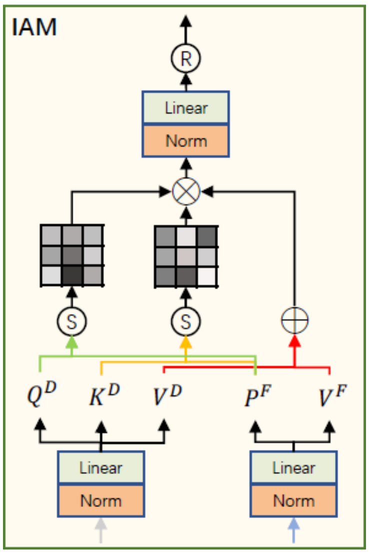


***Figure 17**: IAM architecture. [17]*

The IAM takes two inputs: the multi-scale fusion feature F from the Feature Fusion (FF) module and the deepest feature D from the diffusion model.

These inputs undergo the following process:

1. **Feature Transformation**

The deepest feature D is linearly projected to produce the query Q_{D}, key K_{D}, and value V_{D}. The fusion feature F is linearly projected to generate P_{F}, and V_{F}. Unlike typical attention mechanisms, F does not generate queries and keys for direct similarity comparison, instead, it uses P_{F}, for this purpose. Those are formulated by:

$Q_{\mathrm{D}} = D \cdot W_D^Q$ - The query matrix derived from the deepest feature D.
$K_{\mathrm{D}} = D \cdot W_D^K$ - The key matrix derived from the deepest feature D.

$V_{\mathrm{D}} = D \cdot W_D^V$ - The value matrix derived from the deepest feature D.

$P_{\mathrm{F}} = F \cdot W_F^P$ - The intermediary projection of F, used for similarity comparison.

$V_{\mathrm{F}} = F \cdot W_F^V$ - The value matrix derived from the fusion feature F.

Where $W_D^Q, W_D^K, W_D^V, W_F^P, W_F^V$ are the learned projection matrices, and d is the dimensionality of these projections.

2. **Similarity Computation**

The similarity between the query Q_{D} and the intermediary projection P_{F} is as computed as followed:

$$M_1^{att} = Softmax\left(\frac{Q_{\mathrm{D}} \cdot P_F^T}{\sqrt{d}}\right)$$

Similarly, the similarity between the key K_{D} and the intermediary projection P_{F} is computed:

$$M_2^{att} = Softmax\left(\frac{K_{\mathrm{D}} \cdot P_F^T}{\sqrt{d}}\right)$$

These similarity scores are normalized using the SoftMax function, resulting in attention maps M_1^{att} and M_2^{att} (Attention maps are a core component of the attention mechanism in machine learning models, especially in tasks like image processing and natural language processing. They help the model understand which parts of the input data are most important for making accurate predictions or decisions).

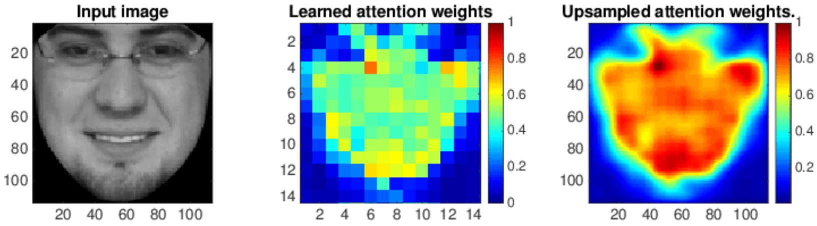


***Figure 18**: Heat map visualization of learned attention. [18]*

3. **Attention Weight Application**

The weighted values are combined to form the final output:

$$O_{\mathrm{I}} = M_1^{att} \cdot M_2^{att} \cdot (V_{\mathrm{D}} + V_{\mathrm{F}})$$

The output feature map O_{I} serves as a refined representation that combines the strengths of both the multi-scale fusion features and the deep diffusion features. By leveraging cross-attention, O_{I} effectively integrates texture and localization information, making it highly effective for tasks that require detailed and context-aware feature representations.

4. Expected Achievements

The project purpose is to adapt the stated image reconstruction to another attention consisting of image rebuilding by their vague noisy and degraded versions. Using a diffusion-based model, the process involves adding controlled noise to ground-truth images during training, allowing the model to learn and reverse these effects, ultimately restoring images to their original state. This approach effectively addresses various types of noise and degradation, significantly improving image quality in applications such as medical imaging by enhancing the visibility and clarity of anatomical structures for more accurate diagnoses. The Feature Fusion (FF) module will be fine-tuned to better integrate multi-scale features, helping the model identify and correct noise and artifacts at various levels, resulting in sharper, clearer images. The objective is to determine the model's effectiveness in image restoration tasks, ensuring high performance in accuracy, efficiency, and visual quality.

Evaluation metrics

1. Structure-measure (Sα): Evaluates the structural similarity between the predicted mask and the ground truth, focusing on object and region perception.

2. Weighted F-measure (Fωβ): A weighted version of the mean F-measure that

combines accuracy and recall.
3. Mean F-measure (Fm): Measures the harmonic mean of precision and recall.
4. Mean E-measure (Em): Evaluates both pixel-level matching and image-level statistics to measure overall and local accuracy.
5. Mean Absolute Error (MAE): Measures the average pixel-level error between the predicted mask and the ground truth.
6. Peak Signal-to-Noise Ratio (PSNR): Measures the ratio between the maximum possible power of a signal and the power of corrupting noise.
7. Structural Similarity Index (SSIM): Assesses the similarity between the restored image and the original image based on luminance, contrast, and structure.
8. Visual Information Fidelity (VIF): Evaluates the visual quality of the restored image based on human visual perception.

Success criteria

The success of the project will be measured based on the accuracy and efficiency of the algorithm in both detecting and segmenting camouflaged objects, as well as restoring and enhancing image quality. The success criteria include:

1. Accurate Detection and Segmentation
2. Structural Similarity
3. Balanced Precision and Recall
4. Pixel-level and Image-level Accuracy
5. Pixel-wise Error Minimization
6. Efficient Processing
7. Image Restoration Quality
8. Visual Similarity
9. Visual Information Fidelity

5. Research Process

5.1.Process

In part A of the project, the research process began by exploring new areas within the extensive field of camouflaged object detection and image restoration. An in-depth study was conduct on articles, algorithms, models, and architectures related to diffusion models. The work was divided into two main parts. First, different types of noise were examined, such as Gaussian, salt-and-pepper, and speckle noise, affect an image. We analyzed the characteristics and challenges of each noise type and explored methods that restore images to their cleanest versions.

Second, the focus shifted to studying the diffCOD architecture and its implementation process. This involved analyzing how the architecture works, what inputs it takes, and what outputs it produces. We also tested techniques to enhance the effectiveness of image restoration using diffCOD, aiming to obtain clearer and higher-quality results. While the primary goal was focused on image restoration, we also explored methods to enhance image quality as a secondary goal.

In part B of the project, the presented model will be developed, incorporating the insights from part A. The goal is to provide a reliable implementation capable to reach the stated purposes. The project success will be estimated by the average accuracy and overall performance of the model in restoring image, with a secondary focus on enhancing image features.

5.2.Product

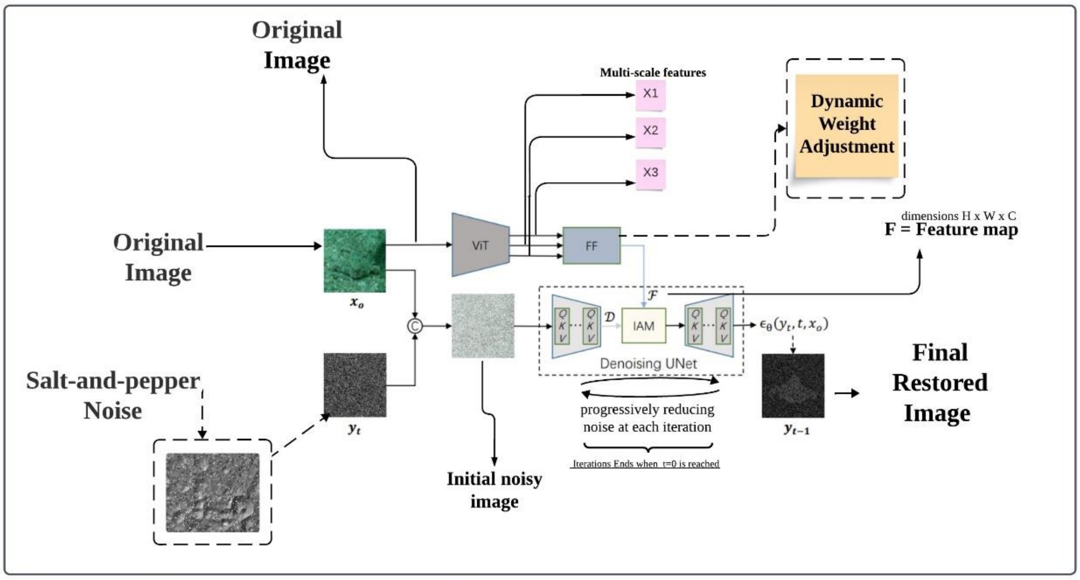


Figure 19. Workflow of the model. The input feeds a given image concatenating with "salt-and-pepper" noise into a denoising diffusion model with a U-Net architecture as the core component for denoising. An injection attention module (IAM) is designed to implicitly guide the diffusion process with the conditional semantic features that have gone through the Vision Transformer (ViT) and the Feed Forward (FF) module, emphasizing the importance of the FF in processing and refining features. This allows the model to take full advantage of the correspondence between image features and diffusion information, ultimately achieving effective image restoration.

1. Input Image Preparation

Input:

X_0: Original image before any restoration process.

Y_t: Salt-and-Pepper noise that added iteratively to the ground truth.

Process:

Adding "Salt and Pepper" Noise to X_0 to simulate degradation scenarios. This part of the process called forward process and he is disrupting pixels to either black or white values. This process alters the pixel value distribution, introducing sharp peaks at the minimum and maximum intensity values.

Output:

X_t: Noisy image generated for training purposes

2. Vision Transformer (ViT) Backbone

Input:

X_0: Original image from the image preparation stage.

Process:

Divide the image into smaller patches. These patches are processed to extract multi-scale features at different levels of detail.

Output:

(X_1^p, X_2^p, X_3^p): Multi-scale features representing different levels of detail from the image. These features capture fine details and broader patterns at different resolutions.

3. Enhanced Feature Fusion (FF) Module

Input:

(X_1^p, X_2^p, X_3^p): Multi-scale features extracted by the ViT backbone.

Process:

Enhanced Multi-Scale Integration:

By using fine-tuned convolution operations to capture and integrate multi-scale features, the convolution operations adjust kernel size, stride, and padding to ensure dimensions are preserved. The kernel size, which defines the height and width of the filter used in convolution, plays a critical role in determining the scale of features captured. Smaller kernel sizes (e.g., 3x3) are used to capture fine-grained details, while larger kernel sizes (e.g., 5x5) capture broader patterns. Adjusting these parameters allows the model to effectively process and integrate multi-scale features for improved performance.

Dynamic Weight Adjustment:

Weights are dynamically learned during training to prioritize the most relevant features.

Attention mechanisms (self-attention and cross-attention) help in focusing on important features and suppressing irrelevant ones.

Output:

F_u: A unified feature map that integrates multi-scale information and prioritizes relevant features, assisting the restoration process in the denoising block.

4. Denoising U-Net

U-Net Encoder Input:

F_u: The unified feature map from the FF module.

X_t: The initial noisy image from the preparation stage.

Process:

The noisy image X_t undergoes successive convolutional layers and max-pooling operations. This process captures increasingly abstract features while reducing spatial dimensions, leading to the bottleneck. The intermediate output of the above process is the deepest feature map (also called D)

U-Net Encoder Output:

D: The deepest feature from the U-Net module.

4.1 Injection Attention Module (IAM)

IAM Input:

F: The unified fusion feature map from the FF module.

D: The deepest feature from the U-Net module.

Process:

The deepest feature D is linearly projected to produce the query Q_{D}, key K_{D}, and value V_{D}. The fusion feature F is linearly projected to generate P_{F}, and V_{F}. The IAM computes similarity scores between these projections to form attention maps, which are then used to weight and combine the features.

IAM Output:

O_I: A feature map (O_I) that effectively merges texture and localization information for refined representations.

U-Net Decoder:

Input:

O_I: The further refined denoised image from the IAM.

Process:

Applies the denoising steps iteratively.

Each iteration work in the following structure:

O_I, which contains refined, context-aware features, is combined with the up-sampled features from the bottleneck layer. This combination leverages the detailed texture and localization information in O_I, enhancing the decoder's ability to produce accurate and detailed segmentation maps as it progresses through the up-sampling layer and integrates skip connections.

Output:

Y_{t-1} restored: The final denoised and restored image for iteration t, which closely resembles the original image before degradation. This part of the process called reverse process and he is mapping from the noisy ground-truth image Y_t to the restored image Y_{t-1} step by step until the segmented image is acquired.

5.3.Related Diagrams

Use Case Diagram:

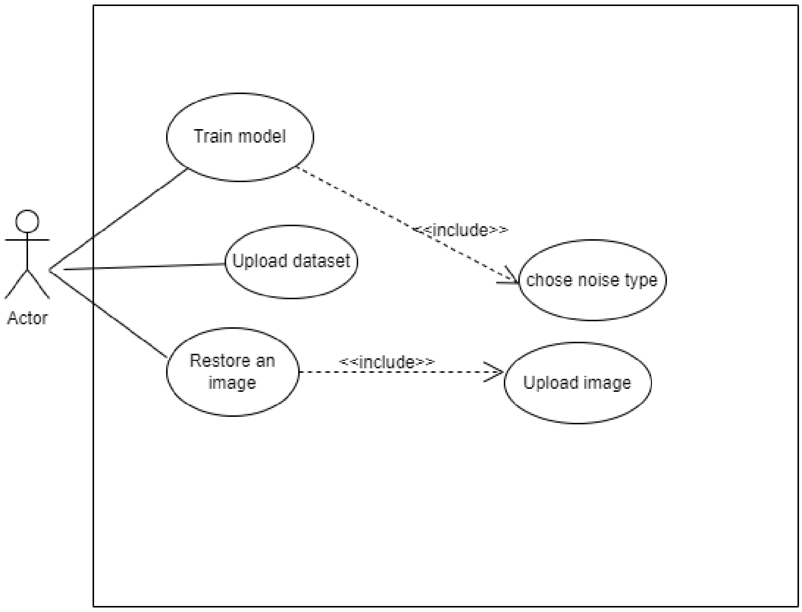


Figure 19: *Use Case Diagram*

Sequence Diagram:

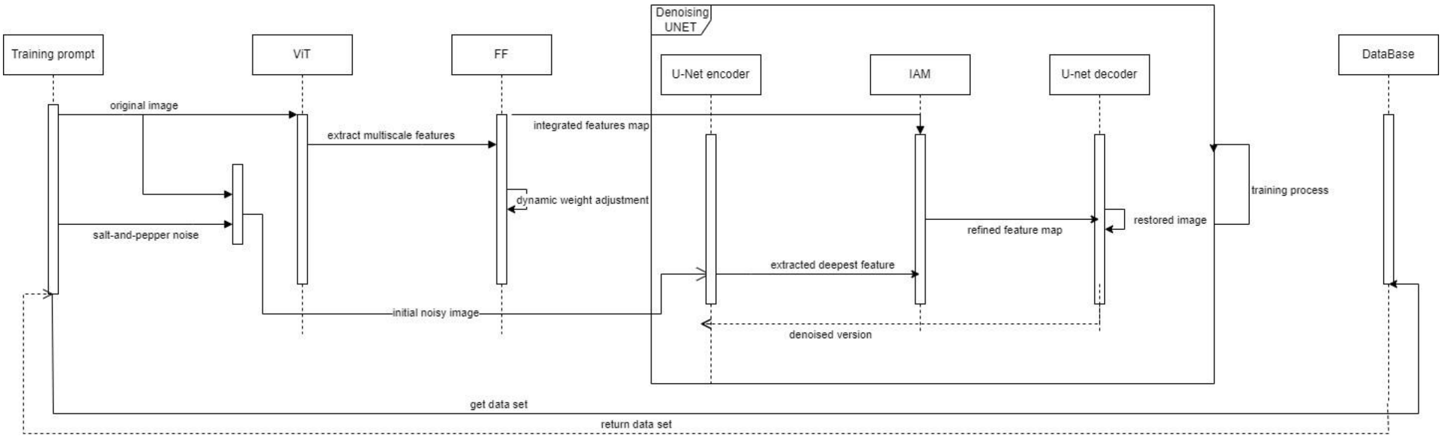


Figure 20: *Sequence Diagram*

6. Evaluation

The evaluation plan for the image reconstruction model involves a comprehensive assessment of its performance across diverse scenarios and conditions. Initially, a variety of datasets, including natural and medical images, are selected to test the model's robustness and generalization ability. These datasets undergo preprocessing steps, such as resolution reduction and noise addition, to simulate real-world degradation. The model is trained and validated using metrics like PSNR and SSIM to monitor its learning progress. Once trained, the model is tested on unseen data, evaluating its ability to restore images accurately and maintain visual fidelity. The evaluation includes quantitative measures, such as PSNR and SSIM, and qualitative assessments through visual inspections and expert reviews to ensure usability and clarity. Additionally, the evaluation plan incorporates the following tests:

- **Test robustness to varying noise types** (e.g., Gaussian, salt-and-pepper) to ensure consistent performance across different noise conditions.
- **Validate extraction consistency across different image resolutions** to confirm that the model performs effectively regardless of input resolution.
- **Test integration and enhancement of multi-scale features for noise correction** to ensure the model effectively combines and utilizes features for better restoration.
- **Test adaptability to dynamically changing input feature sets** to verify that the model maintains performance despite variations in input data.
- **Verify the performance with weighted sum adjustments to prioritize features**, ensuring that the model emphasizes critical features for improved restoration.
- **Test efficiency of the denoising process under different computational loads** to ensure the model maintains speed and effectiveness even under varying processing demands.

Comparative analysis against state-of-the-art models and stress testing further validates the model's effectiveness, scalability, and robustness in real-world applications.

6.1.Testing plan

Module	Test Description	Expected result
Input Preprocessing	Ensure raw images are resized, noise-added, and formatted correctly.	Images are resized to the target dimensions (e.g., 256x256 pixels), with noise added to match specified parameters (e.g., 10% Gaussian noise). Format conversion to a unified type (e.g., PNG) is verified.
	Test handling of different image formats (e.g., JPEG, PNG, TIFF).	All image formats are successfully processed and converted, with no data loss or errors, achieving 100% format compatibility.

	Verify that extreme noise levels do not crash or stall preprocessing.	The preprocessing module handles noise levels up to 50% Gaussian noise without crashing, maintaining processing time within 10% of baseline speed.
	Test the ability to handle batch processing of images.	The system processes batches of 100 images within a 5-minute timeframe, maintaining consistent quality across all outputs.
Feature Extraction	Verify extraction of relevant features from noisy images across multiple scales.	Extracted feature maps consistently capture essential details, with feature detection accuracy exceeding 90% across all scales.
Denoising and Reverse Process	Validate ability to remove noise and restore images to near-original state.	Denoised images achieve PSNR values of at least 30 dB, indicating effective noise removal.
Injection Attention Module (IAM)	Ensure integration of semantic features to refine and enhance restoration.	Improved image quality is evidenced by a 10% increase in SSIM, confirming the IAM's enhancement capabilities.
	Validate the influence of attention mechanisms on restoration quality.	Attention mechanisms contribute to a 15% improvement in edge detail accuracy, as measured by edge detection metrics.
Output Module	Confirm final output images meet quality standards in sharpness and clarity.	Restored images achieve SSIM scores above 0.85 and PSNR values above 30 dB, indicating high-quality restoration.
	Test output consistency across multiple test cases and conditions.	Output consistency is verified with less than 5% variation in SSIM and PSNR across different test cases.
	Evaluate performance with edge cases, such as very high or low-resolution images.	High-resolution images maintain SSIM above 0.8, while low-resolution images achieve at least a 0.75 SSIM.
	Check endless loop in training	

7. AI tools

We used ChatGPT for improving spelling and grammar of sentences and sometimes also referring to articles about complex subjects.

8. References

[1]. Chen, Z., Gao, R., Xiang, T.-Z., & Lin, F. Diffusion Model for Camouflaged Object Detection. School of Informatics, Xiamen University, Xiamen, China; Department of Computer Science and Engineering, HKUST, Hong Kong, China; G42, Abu Dhabi, UAE.

[2]. Zhou, Y., Ge, R., McGrath, G., & Loianno, G. FENet: Fast Real-time Semantic Edge Detection Network. Presented at New York University, Tandon School of Engineering, in collaboration with Qualcomm Technologies Inc.

[3]. Chen, Z., Zhang, X., Xiang, T.-Z., & Tai, Y. Adaptive Guidance Learning for Camouflaged Object Detection. School of Intelligence Science and Technology, Nanjing University, Suzhou, China; College of Computer Science, Nankai University, Tianjin, China; G42, Abu Dhabi, UAE. Correspondence to: T.-Z. Xiang and Y. Tai

[4]. Kendall, A., Gal, Y., & Cipolla, R. (2017). Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics

[5]. https://medium.com/@tech-gumptions/transformer-architecture-simplified-3fb501d461c8

[6]. https://www.semanticscholar.org/paper/MedSegDiff-V2%3A-Diffusion-based-Medical-Image-with-Wu-Fu/3f77a62ae888c3b816eabd354a6dd0fc6b9528ea

[7]. https://arxiv.org/abs/2303.04803

[8]. https://www.researchgate.net/figure/Digital-image-representation-by-pixels-vii_fig2_311806469

[9]. https://www.telusinternational.com/insights/ai-data/article/guide-to-image-segmentation

[10]. https://www.researchgate.net/figure/Denoising-diffusion-model-forward-and-backward-process-qxtx-t-1-p-th-x-t-1-xt_fig1_369946317

[11] 1D convolutional neural networks and applications: A survey Serkan Kiranyaz a,⇑ , Onur Avci b , Osama Abdeljaber c , Turker Ince d , Moncef Gabbouj e , Daniel J. Inman f

[12] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. "U-net: Convolutional networks for biomedical image segmentation". MICCAI, 9351:234–241, 2015.

[13]. Enhancing SDO/HMI images using deep learning - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/An-example-of-a-convolution-with-a-filter-In-this-example-a-vertical-border-locating_fig1_317543623 [accessed 20 Jul 2024]

[14] Up and Down sampling - https://mriquestions.com/upsampling.html.

[15]. https://www.researchgate.net/figure/Vision-transformer-model-adopted-for-classification-of-brain-tumors-from-MRI-MLP_fig2_364269781

[16].https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.1055451/full

[17] Chen, Z., Gao, R., Xiang, T.-Z., & Lin, F. Diffusion Model for Camouflaged Object Detection. School of Informatics, Xiamen University, Xiamen, China; Department of Computer Science and Engineering, HKUST, Hong Kong, China; G42, Abu Dhabi, UAE

[18]. https://www.researchgate.net/figure/The-heat-map-visualization-of-the-learned-attention-weights-by-our-spatial-attention_fig4_321306916