**Custom Dataset and Multi-Channel Representation**

To facilitate supervised training for image tampering localization, we implemented a custom PyTorch dataset tailored to the CASIA dataset. Each data sample is represented as a triplet: **(original image, tampered image, binary tampering mask)**. The dataset focuses on the tampered images as inputs and their corresponding masks as ground truth labels for segmentation.

For each tampered image, we apply a standard preprocessing pipeline consisting of resizing and normalization. In addition to the RGB channels, we extract two complementary representations: the **noise residual** and the **frequency domain** transform, as described in Section X. These three components—RGB image, noise map, and frequency map—are concatenated along the channel dimension, resulting in a **9-channel input tensor** of shape [9×224×224][9 \times 224 \times 224][9×224×224].

This multi-modal representation provides the model not only with standard visual information but also with **low-level signals** that are often indicative of digital manipulation, such as compression artifacts, frequency inconsistencies, and unnatural noise patterns. By integrating these weak cues directly into the learning process, the segmentation network is better equipped to **identify subtle traces of tampering** that may be imperceptible in the RGB domain alone.

**Frequency and Noise-Based Feature Extraction**

To enhance the discriminative power of the input representation, we incorporate two additional modalities derived from the RGB image: a **frequency domain representation** and a **noise residual map**. These representations are computed using custom functions described as follows.

**Frequency Extraction.**  
Given an RGB image tensor, we compute its frequency representation by applying a 2D Fast Fourier Transform (FFT) independently to each color channel. The resulting complex-valued spectrum is centered using a frequency shift, and the magnitude is then logarithmically compressed via a log1p transform to reduce dynamic range. Finally, the values are normalized to the range [0, 1]. This process emphasizes both low- and high-frequency patterns, which are relevant for detecting visual inconsistencies and tampering artifacts.

**Noise Residual Extraction.**  
To isolate high-frequency noise, we perform a convolution with a 3×3 averaging filter (box filter) over each RGB channel, producing a blurred version of the image. The residual is obtained by subtracting the blurred image from the original. This residual map captures local anomalies and fine-grained artifacts often introduced by splicing or compression.

The extracted **frequency** and **noise** maps are stacked along the channel dimension with the RGB image, resulting in a 9-channel tensor [9×H×W][9 \times H \times W][9×H×W], which serves as enriched input for the downstream convolutional neural network.

**Methodology**

In this work, we propose a deep learning approach for the segmentation of manipulated regions in digital images using a modified DeepLabv3 architecture with a ResNet-50 backbone.

**Dataset and Preprocessing**  
We utilize the CASIA 2.0 dataset, which contains authentic images, tampered images, and corresponding binary ground truth masks indicating manipulated regions. Triplets comprising a tampered image, its manipulation mask, and the corresponding authentic image were generated. To enrich the input representation, each sample is transformed into a 9-channel tensor by concatenating the RGB image with additional noise and frequency domain features, extracted to provide supplementary cues for manipulation detection.

**Model Architecture**  
The DeepLabv3 model with a ResNet-50 backbone pretrained on ImageNet was adapted by replacing the initial convolutional layer to accept 9 input channels instead of the standard 3. The output layer was configured to predict a single binary segmentation mask.

**Training Procedure**  
The dataset was randomly split into training (70%), validation (15%), and test (15%) sets. Training was performed using the Binary Cross Entropy with Logits loss function and the Adam optimizer with a learning rate of 1e-4. An early stopping mechanism with a patience of 3 epochs was employed based on validation loss to prevent overfitting. The best model checkpoint was saved during training.

**Evaluation Metrics**  
Model performance was evaluated on the test set using Intersection over Union (IoU) and F1-score metrics, comparing predicted masks against ground truth annotations. Predicted masks were saved as binary PNG images for qualitative assessment.

**Inference and Visualization**  
An inference pipeline was implemented to process batches of images and generate predicted segmentation masks, which are subsequently binarized using a threshold of 0.5. Additionally, visualization functions allow inspection of input images alongside their noise and frequency components to aid qualitative analysis.