UDC 004.056.55

Influence of SRM Filters Preprocessing on Stego Data Localization in Digital Images

Pavlo Yatsura[1], Dmytro Progonov[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Abstract

Early detection and counteraction to unauthorized transmission of sensitive information via publicly available networks are topical tasks today. Of special interest are steganalysis methods aimed for effective destruction of hidden messages embedded into innocuous media files, like digital images. However, practical usage of such methods introduces significant changes into statistical and spectral parameters of processed images, thus revealing the intrusion into stego channels. There are proposed novel methods for localization positions of embedded stego bits into cover images and pointwise processing only these positions. The article quantifies the impact of cover images preprocessing on accuracy of stego bits localization. The case of Spatial Rich Model (SRM) filters usage is considered, while stego bits position detection is performed using novel deep neural networks, such as Unet, LinkNet, PSPNet and FPN models. The results of comparative analysis of localization accuracy proved effectiveness of SRM filters usage, namely to increase of localization accuracy up to five times (from 2.01% to 10.9% of Intersection-over-Union metric values) even for modern adaptive embedding (like MG and MiPOD) and low cover image payload values (about of 3%-5%). Obtained results create preconditions for development of high-accuracy methods for localization positions of stego bits embedded into cover images according to novel embedding methods.

Keywords: steganography, steganalysis, artificial neural networks, digital images, SRM, cybersecurity

Introduction

The overwhelming amount of processed information and digital media in modern networks makes it possible to create hidden channels for covert communication and unauthorized information sharing between malefactors [1, 2]. Distinctive feature of such channels is data embedding into innocuous files that are processed and transmitted in communication systems, such as multimedia files, text data etc. This allows for overcoming popular intrusion detection systems and effectively counteracting usage of widespread signature-based detection methods [2, 3].

One of the most widely used types of digital media for message (stegodata) hiding is multi-media, namely digital images (DI) [4, 8]. A wide range of proposed methods for DI processing, for example lossy compression, perceptual quality enhancement, denoising, as well as appearance of noisy-like textures (ex.: sands, grass leaves), makes such files an attractive candidate for stegodata embedding. Thus, the development of effective steganalysis methods for DI aimed at detecting, removing or destruction of hidden data is a critically important task [6].

The majority of modern steganalysis methods for DI is based on exploiting statistics of residuals obtained after context suppression in processed images [3, 7]. As an example, we may mention the popular Spatial Rich Model (SRM) [7] that remains one of key tools for the design of novel stegodetectors. The SRM suppresses semantic content and amplifies high-frequency changes where adaptive embedding methods usually inject stegobits [3, 9, 8, 10, 11, 12]. Then, the advanced feature extraction methods are applied to obtained residuals, for example using novel deep convolutional neural networks (CNN) [4]. Finally, extracted features are used to adjust parameters of classifier module in stegodetector (SD) to label processed image as either cover, or stego one [2].

Despite the appearance of advanced SD for digital images, their detection accuracy highly depends on availability of prior information about features of used embedding methods. On the other hand, applying destruction methods (for example, lossy compression) allows for introducing irreversible distortions into stego

image. However, this is achieved by appearance of significant and specific changes of statistical and spectral features of processed DI. This discloses to malefactors the fact of security team intrusion into the communication channel. Thus, of special interest is novel methods for stego data extraction or even replacement [4]. Such methods allow for effective counteraction to message transmission by preserving minimal impact on cover image (CI) statistical features.

In the paper [4, 13] it is proposed to apply novel DI segmentation networks for localization pixels that have been used for embedding individual bits of stego data. However, their effectiveness may be insufficient for reliable localization since stego bits are scattered over the whole image instead of forming the sole groups. In this study, we put forward a hypothesis that integrating image preprocessing stage, namely applying of a separate filter from SRM bank [7], into a steganalysis pipeline can increase the localization rate of hidden data (stegobits). The aim of this paper is to quantitatively evaluate the impact of preprocessing of stego images formed according to novel adaptive embedding methods, on accuracy of stego bits localization.

Our contribution can be summarized as follows:

- We obtained quantitative estimations of stego bits localization accuracy changes by introducing images preprocessing with SRM filters. Usage of these filters with popular types of segmentation networks, based on U-Net, FPN, LinkNet, and PSPNet architectures, allows for increasing the localization accuracy, namely Intersection-over-Union (IoU), up to five times (from 2.01% to 10.9%) on the ALASKA dataset.
- We analyzed influence of a single SRM filter on DI segmentation models performance based on EfficientNet-B3, ResNet-34, and Mobile-NetV2 backbones. We established that usage of EfficientNet-B3 model allows achieving the tradeoff between localization accuracy and computation time. Applying of ResNet-34, and MobileNetV2 leads to decreasing a bit of IoU by preserving of computation overhead.
- It is revealed that image preprocessing with SRM filters has the most impact on stego images formed by HUGO embedding methods (changes of IoU is up to 8%) even for low payload ($\Delta_\alpha = 3\%$). In contrast, stego bits localization accuracy changes for MG and MiPOD methods are much smaller (about of 5% and 1% respectively). Therefore, additional research is required to provide reliable detection of embedded stego bits localization for these methods.

The structure of the paper is as follows: used terms and abbreviation are presented in the Section 1. The Section 2 is devoted to a literature review for considered steganographic methods and the novel CNN based stegodetectors for digital images. The Section 3 follows with description of proposed solution, while results of performance evaluation by its practical usage are presented in Section 4. Next, Section 5 contains discussions based on finding of the previous section. Finally, Section 6 presents the conclusions of the paper.

1. Notations and definitions

In **boldface** we indicate high-dimensional arrays, matrices, and vectors. Their individual elements will be denoted by the corresponding lower-case letters in *italic*. Calligraphic font is reserved for sets.

The $\|\cdot\|_2$ is L_2 (Euclidean) norm of a vector. The "$*$" denotes the convolutional operation. The $p_{i,j}$ is the probability of changing the pixel of cover image with coordinate (i, j).

We assume that cover (**X**) and stego (**Y**) images are represented in grayscale and have resolution of $H \times W$ (pixels). The color depth is assumed to be 8 bits.

The embedding rate (Δ_α) quantifies how much payload is inserted per cover image's pixel, measured in bits per pixel (BPP). It represents the average number of hidden bits carries at a specified distortion level of CI.

2. Literature Review

2.1. Adaptive embedding methods

Modern steganography for multimedia files focuses on increasing the robustness of hidden data against novel statistical steganalysis methods, especially based on usage of the neural networks [4]. One popular approach to achieve this goal is the usage of adaptive embedding strategies that is aimed at minimization of CI parameters alterations during message hiding [14-18]. This is achieved by solving the following optimization problem [14]:

$$D(\mathbf{X},\mathbf{Y}) = \sum_{i,j} p_{i,j}|\mathbf{Y}_i - \mathbf{X}_i| \to min, \quad (1)$$

under restriction of

$$\sum_{i,j} \mathrm{H}\left(\{p_{i,j}^0, p_{i,j}^+, p_{i,j}^-\}\right) = m, \quad (2)$$

where p_i is the per-pixel cost, $\mathbf{X}_i$ and $\mathbf{Y}_i$ are cover image and stego image pixels; m is the bitlength of payload; $\mathrm{H}(\pi)$ is entropy function for distribution π; $\{p_i^0, p_i^+, p_i^-\}$ are the probabilities of changing of cover image's pixel brightness on "0", "+1", or "-1" values respectively.

Modern adaptive embedding methods (AEM) strive to minimize a content-aware distortion (1). Changes made to DI pixels are steered into textured regions and in smooth, predictable areas, reducing statistical detectability [14, 19]. A typical pipeline estimates a per-pixel cost (for example, from local residuals/gradients), then uses trellis syndrome coding (or similar constrained coding) to embed a target value while respecting the cost map. This allows for significantly reducing (up to 5-20%) the detectability (probability of correct classification by stegodetector) of formed stego images, especially at low embedding rates (less than 10%) [14, 16].

The distinctive features of widely used AEM are presented in next subsections.

2.1.1. Steganographic method HUGO

The HUGO (Highly Undetectable steGO) [14] is adaptive method for message hiding into the spatial domain of DI. The core idea of this method is to detect and modify only pixels whose change minimally perturbs the image's statistical structure (for example, spatial regularities, gradients of pixel brightness, contrast, texture).

In contrast to Least Significant Bit (LSB) based embedding methods where embedding is performed for randomly selected pixels of CI, HUGO method is aimed at building a high-dimensional local-feature description [14]. Then, these descriptions are used to gauge how noticeable a (±1) intensity tweak would be to a steganalyst. This makes possible creation a per-pixel cost map by measuring the distance between feature vectors before and after a such change (3):

$$p_{i,j} = \left\|\boldsymbol{\phi}\left(\mathbf{X}_{i,j}\right) - \boldsymbol{\phi}\left(\mathbf{X}_{i,j\pm1}\right)\right\|_2, \quad (3)$$

where $\boldsymbol{\phi}\left(\mathbf{X}_{i,j}\right)$ is the vector of local features (for example, gradients, contrast, textural characteristics) by altering the brightness of cover image's pixel with coordinate (i,j); $\boldsymbol{\phi}\left(\mathbf{X}_{i,j\pm1}\right)$ is the related vector for the case of perturbed brightness by 1 of pixel with coordinates $(i, j+1)$. The vector $\boldsymbol{\phi}\left(\mathbf{X}_{i,j}\right)$ is used to describe the degree of correlation of intensity values of adjacent pixels in the original (unmodified) image **X**.

Given the cost map (3), HUGO method derives a probability distribution of modification outcomes per pixel and embeds stegobits into cover image using a set of allowed changes (often ±1) with the smallest values of target function $D(\mathbf{X},\mathbf{Y})$ (1). The embedding level is managed by parameter λ according to the following formula [14]:

$$p_{i,j} = \frac{2e^{-\lambda p_{i,j}}}{1+2e^{-\lambda p_{i,j}}} \quad (3)$$

where $\lambda > 0$ is a parameter, whose value is adjusted depending on the specified amount of embedded data. For example, λ=0.1–0.3 for small payloads (approx. 1–3%) and λ = 0.5–0.8 for larger payloads (more than 5%) [14].

The HUGO method is one of the first examples of practical usage of adaptive embedding strategies. The novel AEM, like MiPOD (Minimizing the Probability of Detection) method [15], use similar approaches for CI distortions estimation while applying more sophisticated functions to estimate these perturbations.

2.1.2. Steganographic method MiPOD

Unlike HUGO's distortion-minimizing formulation, the MiPOD method [15] derives per-pixel modification probabilities by minimizing the miss-detection probability under a local statistical model of the cover noise.

The MiPOD method follows the minimization of the average detectability of per pixel changes under a fixed embedded payload restriction (4):

$$D(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{\left(1-2\cdot p_{i,j}\right)^2}{\sigma_{i,j}^2}, \quad (4)$$

where $\sigma_{i,j}^2$ is the local variance on the prediction-error residual around (i,j) position, reflecting

the degree of variability of noise-like fluctuations in that fragment of the DI.

The MiPOD method uses a probabilistic embedding rule: each pixel (i, j) of CI is assigned a modification probability $p_{i,j}$ computed from the local variance $\sigma_{i,j}^2$, i.e., the variability of intensities in the pixel's neighborhood. Then, higher variance in (4) implies lower visibility of a change and thus a higher admissible modification probability.

To avoid exceeding the specified payload, the MiPOD method constrains the sum of the information entropy of all probabilities to the target number of bits (2) by usage of the binary entropy (6):

$$H(p) = -p \cdot \log_2(p) - \\ -(1-p) \cdot \log_2(1-p) \quad (5)$$

This allocates changes primarily to complex or textured regions of DI, where they are harder to detect using statistical descriptors by preserving the stego data bitlength.

Modification locations from the resulting probability distribution map are sampled according to local DI characteristics. Thus, changing only those pixels of CI, where it is least noticeable and quantitatively justified by the information budget [15].

Let us note that the idea of variance-driven probabilities estimation used in the MiPOD method leads to creation of separate group of embedding methods. These methods are focused on as accurately as possible estimation of local variability of pixel brightness and taking into account its correlation for adjacent groups of CI pixels. One example from this group is the MG embedding method.

2.1.3. Steganographic method MG

The MG (Multivariate Generalized Gaussian Cover Model) [16] method replaces variance-driven probabilities with a gradient/texture-based surrogate: modification costs are derived from local edge/texture strength, so changes are preferentially allocated to high-entropy regions. This relies on a DI statistical model that captures the multivariate distribution of local CI blocks via a generalized Gaussian distribution.

Embedding of a message into CI according to the MG method is performed in several steps. At first, adjacent pixels of CI are grouped into blocks of size $k \times k$ (e.g., 3×3 or 5×5 pixels). Then, each block is vectorized in a row-wise manner, treated as a realization of a random vector from Gaussian probability distribution.

On the second stage, the distribution density of the vectorized block $\mathbf{x} \in \mathbb{R}^{1\times n}$ is approximated using the multivariate generalized Gaussian density (6):

$$f(\mathbf{x}) = \frac{\beta\Gamma\left(\frac{n}{\beta}\right)}{\pi^{n/2}2^{n/2\beta}\Gamma\left(\frac{n}{2\beta}\right)|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right)\right)^{\beta} \quad (6)$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ is the covariance matrix encoding dependencies within the block; n is the block-vector dimension (e.g. $n = k^2$ for a $k \times k$ block); $\beta > 0$ controls "peakedness" of the block's density distribution and $\Gamma(\cdot)$ is the gamma function. Parameters $\boldsymbol{\Sigma}$ and β are estimated per CI so the model reflects its local statistics [16].

Before modifying a pixel, the MG method compares the distribution density of the vectorized block pre- and post-change of single-pixel intensity tweak. For an original block vector x and its modified counterpart x', the detectability cost calculates as follows [16]:

$$p = -\log_2\left(\frac{f(\mathbf{x}')}{f(\mathbf{x})}\right) \quad (7)$$

where larger p indicates a change less consistent with the CI model and thus more detectable.

Finally, the set of CI pixels is selected whose alteration yields the smallest p in eq. (7), i.e., modifications that minimally disturb the model-consistent statistics of local blocks, preserving the image's statistical plausibility while adapting to texture and noise variability.

Complementing adaptive embedding methods, traditional steganalysis proceeds from the detector's side [20]. Recent CNN models replace handcrafted features with end-to-end trained representations of ones, preceded by SRM-based residual front-ends. This allows for achieving state-of-the-art detection accuracy (more than 95%) by preserving fixed computation overhead level. The next subsections summarize SRM-based, then outline CNN-based pipelines and their integration with residual priors.

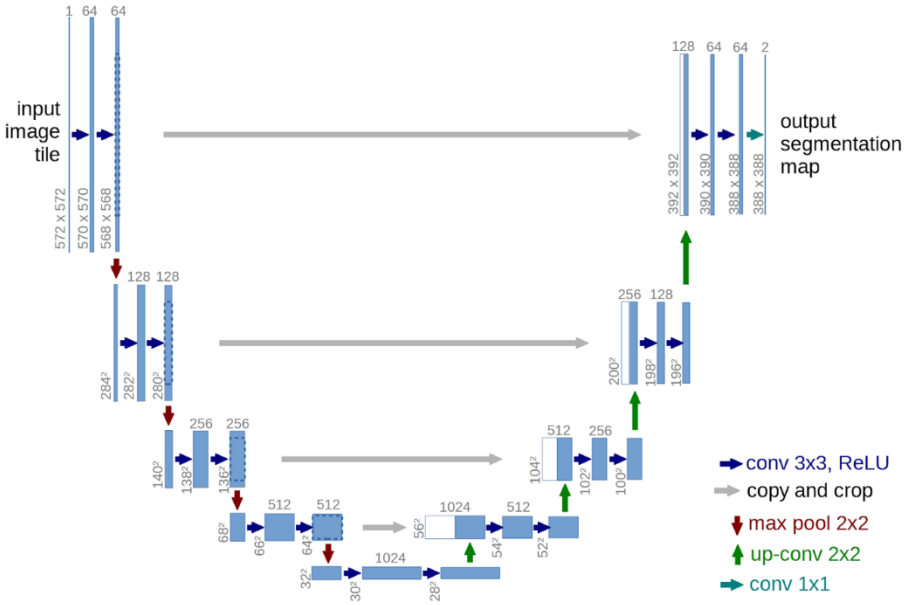

Figure 1. Architecture diagram of Unet model [7]

2.2. Statistical Steganalysis Methods and Spatial Rich Model

The evolution of steganalysis methods has progressed significantly, transitioning from traditional statistical approaches to complex models based on deep learning [8]. This shift has been driven by the increasing sophistication of AEM, which traditional detection methods often struggle to reveal the data reliably [4].

Many classical steganalysis methods focused on the statistics of CI residuals transformation coefficients, generated by applying various filters to the image [9]. One of the most prominent and effective approaches in this category is the Spatial Rich Model [7]. The model is based on using high-dimensional feature vectors constructed from a large set of linear and non-linear high-pass filters. SRM filters suppress scene content of DI and convert images into residual maps in which embedding artifacts have higher signal-to-content ratio (9):

$$\mathbf{r}_k = \mathbf{h}_k * \mathbf{X}, \tag{8}$$

where $\mathbf{r}_k$ is the resulting map of residual for the k-th SRM filter; $\mathbf{h}_k$ is the k-th SRM filter. The subsequent quantization of residuals (8) and estimation co-occurrence statistics for these residuals allows for tracing negligible changes of DI caused by hidden data injection [3, 9, 12].

While being highly effective for detecting content-independent and early adaptive methods, SRM-based detection methods face challenges such as high feature dimensionality, redundancy, and long duration of feature extraction procedure [3]. Furthermore, existing rich models for color image steganalysis may not fully leverage the fact that content-adaptive steganography often changes pixels in complex textured regions with higher probability [21]. Thus, the next step is to study such representations extraction directly from data, which motivates Section 2.3 on deep learning in steganalysis.

2.3. Deep Learning models in Digital Images Steganalysis

Given the rapid development of deep CNN, the approach to DI steganalysis has undergone significant changes. The data-driven feature learning capability of CNNs is a significant advantage over handcrafted features for rich-model (e.g., SRM model), as handcrafted features are often difficult to compose for accurate detection of stegobits [22]. Also, deep learning models allow for effective mitigation with limited prior information about used embedding methods in comparison with widespread statistical SD [23].

Modern CNN based steganalysis methods often employ architectures originally developed for other computer vision tasks, such as image classification or semantic segmentation [24, 25]. Of special interest is usage of novel segmentation models to detect positions of CI pixels used for separate stego bits embedding.

This makes it possible to predict a "stego probability map" or a binary mask indicating modified pixels [20]. However, this task is complicated by negligible differences between cover and formed stego images. Thus, thorough adjustment of segmentation models is required for solving this task.

Let us consider the popular CNN based segmentations models that are perspective for solving tasks for stego probability map creation.

2.3.1. Unet segmentation model

The U-Net is a symmetric encoder–decoder CNN model with skip connections [7]. The model is based on fusion of fine encoder detail with decoder context for dense prediction of a binary stego mask. The architecture of the Unet model is presented in Fig.1.

U-Net offers a strong accuracy–efficiency balance and reliably outperforms pyramid-only decoders on localization [7, 11, 25-28]. Skip connections at each scale restore high-resolution cues lost by downsampling, while concatenating encoder and decoder features before refinement allows for restoring edges and small structures (Fig. 1). These features are beneficial for localizing weak, sparse changes caused by stegobits injection into CI.

In the next subsection we consider LinkNet that is based on further evolution of U-Net model for more accurate semantic segmentation by preserving low memory usage.

2.3.2. LinkNet segmentation model

The LinkNet is an encoder–decoder CNN model designed for accurate and real-time semantic segmentation with low parameter count and memory use [29]. The distinctive feature of LinkNet in comparison with considered U-Net model is replacing concatenative skips in the U-Net model with additive skip connections and residual upsampling [29]. This makes possible reduction of memory usage and preserving fine spatial details. The architecture of the LinkNet model is presented in Fig.2.

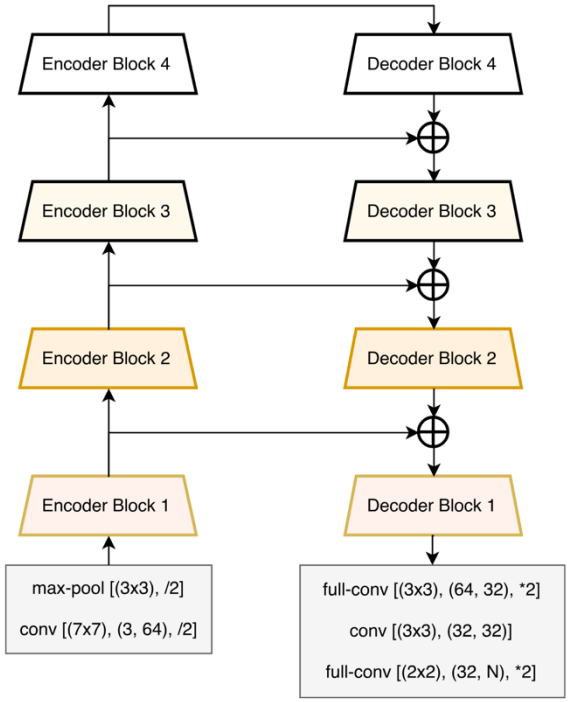

Figure 2: Architecture diagram of LinkNet model [29]

At each stage, LinkNet restores fine structure of CI using direct links from the corresponding encoder feature maps, which keeps computation low compared with heavy decoder designs [29] (Fig. 2). The encoder reuses a standard classification backbone to extract multiscale features of DI, while the decoder is lightweight and reconstructs spatial detail through staged upsampling (Fig. 2).

Residual learning inside blocks follows the standard formulation [29]:

$$\mathbf{z} = \mathbf{s} + \mathrm{F}(\mathbf{s}; \mathbf{W}), \qquad (9)$$

where $\mathbf{s}$ is the block input, $\mathrm{F}(\cdot; \mathbf{W})$ is the residual mapping realized by stacked convolutions with parameters $\mathbf{W}$ (tensors). This construction prevents gradients from explosion or vanishing during model training, and enables usage of deeper encoders.

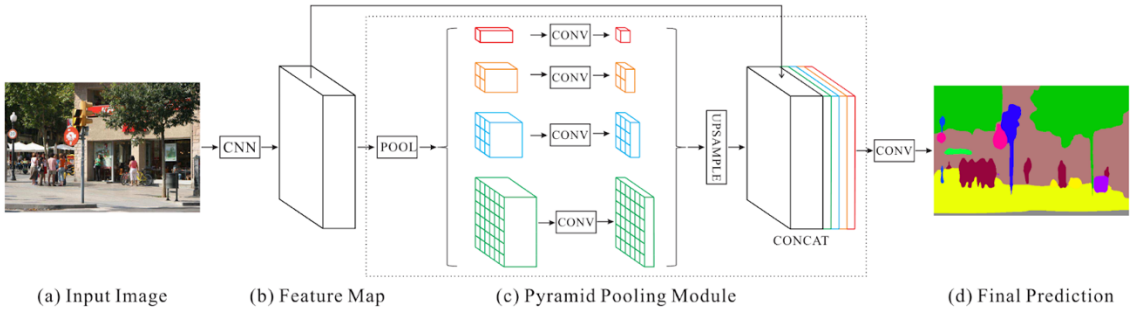

Figure 3. Architecture diagram of PSPNet model [30]

Encoder–decoder linkage for LinkNet model uses additive fusion after channel alignment (10):

$$\mathbf{u}_{l-1} = \mathrm{Up}(\mathbf{u}_l) - \mathrm{g}(\mathbf{e}_l) \qquad (10)$$

where $\mathbf{u}_l$ are decoder features at scale l; Up(·) is spatial upsampling procedure; $\mathbf{e}_l$ are encoder features at the matching scale l; g(·) is a 1×1 convolution to match channels before element-wise addition.

Having established a lightweight baseline with LinkNet's additive skips and residual upsampling, we next consider the advanced Feature Pyramid Networks (FPN), which emphasize multi-scale fusion via a top-down pathway with lateral connections.

2.3.3. FPN segmentation model

The FPN is a top-down architecture with lateral connections that builds semantically strong, multi-scale feature maps from a single backbone. The backbone (e.g., a standard CNN) produces a pyramid of encoder features at different spatial resolutions. The architecture of the FPN model is presented in Fig.3.

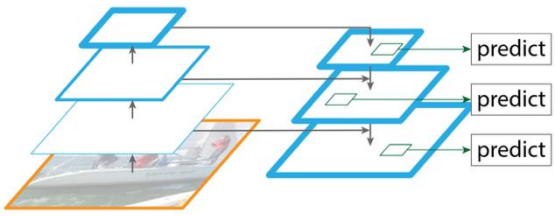

Figure 4: Architecture diagram of FPN model [31]

FPN constructs a second, top-down pyramid by progressively upsampling high-level features (fig. 4). Then, the features are being merged with same-resolution encoder maps via lateral 1×1 projections, yielding feature maps that are both high-level and high-resolution [31].

Pyramid outputs support dense prediction by attaching a lightweight head at each level and either combining results or selecting the appropriate scale at inference. The design preserves strong semantics across resolutions without heavy decoders. This allows for improving model robustness to object/structure size variation while maintaining computational efficiency [31].

While FPN strengthens multi-scale features via a top-down pathway with lateral connections, the novel Pyramid Scene Parsing Network (PSPNet) model complements this by aggregating global context through pyramid pooling. In the next subsection we examine features of the PSPNet model and how they may affect stego bits localization accuracy for stego images.

2.3.4. PSPNet segmentation model

The PSPNet augments a fully convolutional backbone with a pyramid pooling module that aggregates global context at multiple spatial scales and fuses it back into the feature map for dense prediction. The architecture of the PSPNet model is presented in Fig.4.

A shared encoder produces a high-level feature map (Fig .4). Then, the pyramid module performs spatial pooling at several grid sizes (e.g., 1×1, 2×2, 3×3, 6×6). The pooled maps are processed with a 1×1 convolution, and are upsampled to preserve the original feature resolution. Finally, the results are concatenated with encoder features before the final prediction head (Fig. 3) [30].

Pyramid pooling and fusion is done as follows [30]:

$$\mathbf{F}^{(k)} = \mathrm{Up}\left(\phi^{(k)}\left(\mathrm{Pool}_k(\mathbf{F})\right)\right),$$
$$\mathbf{F}_{pp} = \mathrm{concat}\left(\mathbf{F}, \mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(K)}\right) \qquad (10)$$

where **F** is the encoder feature map, $\text{Pool}_k(\cdot)$ is spatial pooling at the k-th bin size, $\phi^{(k)}(\cdot)$ is a 1×1 convolution reducing channels, Up(·) is the upsampling to the spatial size of **F**, $\mathbf{F}^{(k)}$ are the upsampled pooled features, $\mathbf{F}_{pp}$ is the concatenated feature forwarded to the prediction head.

The discussed segmentation models are often combined with backbone networks that are responsible for extracting robust and discriminative features from the input images. We are going to review them in the following subsections.

2.3.5. State-of-the-art backbone models for image segmentation networks

In this section we analyzed advantages and limitations of popular backbones models for DI segmentation models under the task of stego probability map creation. We considered the ResNet [32], EfficientNet [27] and MobileNetV2 [28] models.

The ResNet is a deep convolutional backbone model that introduces identity shortcuts to enable residual learning. Instead of forcing each stack of convolutions to learn a full mapping, the network allows for estimating the residual relative to the input, and adding it back through a skip connection [32]. This mechanism, mitigates with vanishing/exploding gradients, and makes neural network training process stable at substantial depth, removing the degradation observed in plain CNN models [32].

While ResNet is a strong baseline, EfficientNet employs compound depth-width-resolution scaling to improve the accuracy-efficiency trade-off [27]. The key feature of EfficientNet-like models is compound scaling of depth, width, and input resolution of CI [27]. This avoids overfitting (from width-only) and optimization failures (vanishing/exploding gradients and degraded training stability) (from depth-only) while delivering better accuracy-efficiency trade-offs than ad-hoc scaling. The architecture diagram is presented in Fig.5.

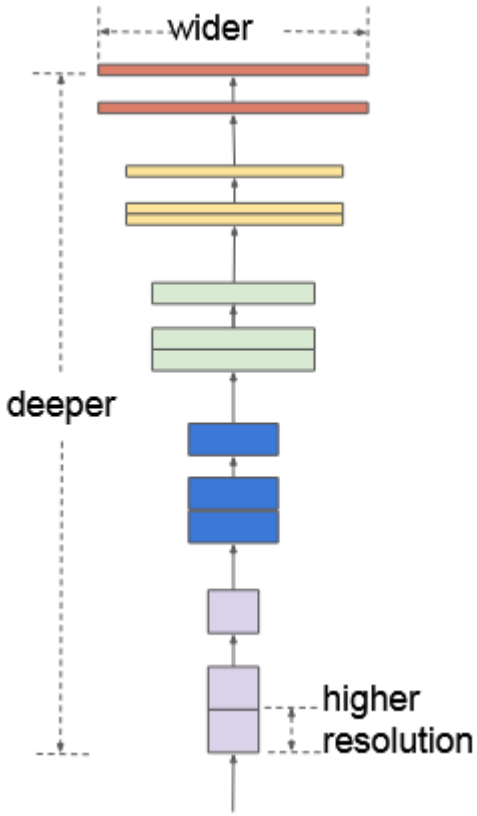

Figure 5: Architecture diagram of EfficientNet model [27]

The core block of the EfficientNet model is MBConv with depthwise separable convolutions and squeeze-and-excitation attention (Fig. 5). A depthwise convolution captures spatial patterns at low computation cost. Then, detected patterns are stacked with stride-based downsampling to form a standard feature pyramid for detection or segmentation heads.

In practice, EfficientNet backbones provide high representational power, making them attractive candidate for encoders for dense prediction tasks [27].

The MobileNetV2 is a lightweight convolutional backbone model built from inverted residual blocks with linear bottlenecks [28]. The architecture of the model is presented on Fig. 6.

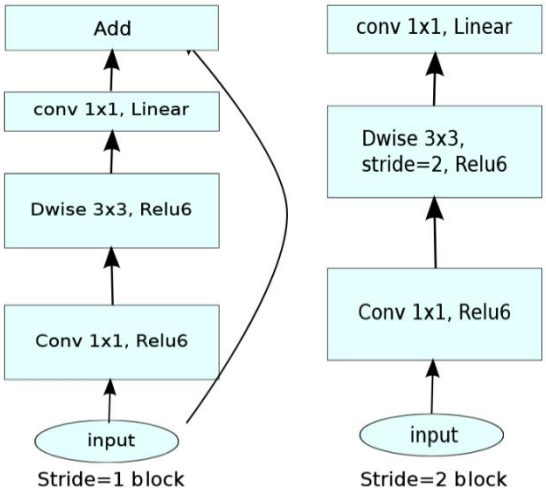

Figure 6: Architecture diagram of MobileNetV2 model [28]

The key design choice for MobileNetV2 model is the linear projection at the block output (no activation), which preserves information that would otherwise be lost by applying nonlinearities in low-dimensional spaces [28]. Each block expands channels with a series of 1×1 pointwise and a depthwise 3×3 convolutions, then projects back to a narrow bottleneck with another 1×1 pointwise layer (Fig. 6). When input and output tensor shapes match, a shortcut connection is used. Otherwise, the block performs downsampling by stride in the depthwise stage.

Considered state-of-the-art segmentation models allows for accurate detection of fine objects on DI that makes them perspective for creation of stego probability map. The reviewed backbone models allow for increasing the localization accuracy by introducing the multistage processing of extracted features. Nevertheless, performance of these models for image steganalysis tasks highly depends on image context detection and suppression. The proposed solution for improving stego bits localization accuracy by applying modern segmentation networks is presented in the next section.

3. Proposed solution

Modern SRM-based pipelines excel at suppressing content and modeling residual dependencies, while CNNs learn task-specific representations that localize weak, sparse changes. To couple these strengths, we propose to combine these advantages into a unified segmentation pipeline presented on Fig. 7.

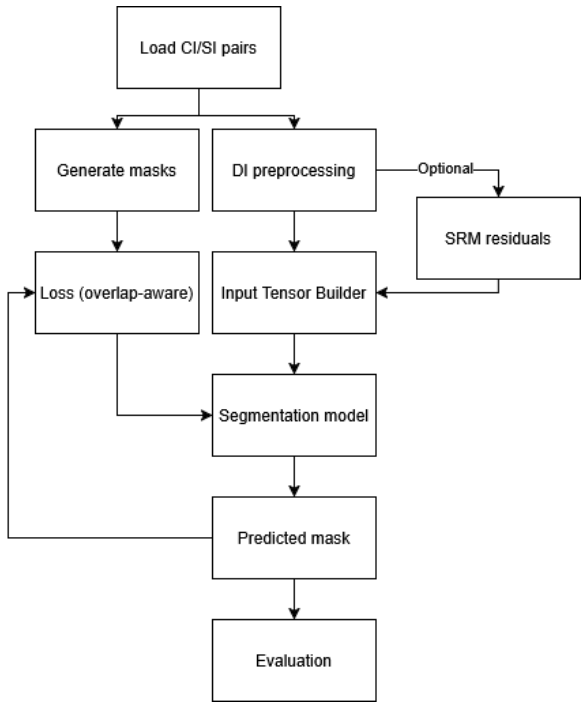

Figure 7: Flow diagram of proposed segmentation pipeline

At first, we convert an input DI $\mathbf{I}$ into residual maps by convolving with SRM kernels $\mathbf{h}_k$ according to eq. (8) [3] (Fig. 7). This is an optional step that is only relevant for training sets that use SRM filters as a preprocessing step. As an example of SRM filter we used a 5×5 Laplacian high-pass filter (L5).

Residual stacks emphasize prediction errors where adaptive embedding tends to leave traces, i.e., high-frequency, noise-like perturbations that disrupt local pixel dependencies and co-occurrence patterns while suppressing smooth scene content [3, 9, 10]. Then, obtained residuals are input to the segmentation model to estimate the stego embedding map. Finally, the network produces a per-pixel probability of marked a pixel as containing of stego bits via a sigmoid head:

$$\hat{\mathbf{z}} = \sigma(\mathrm{f}_\theta(\mathbf{z})) \qquad (12)$$

where $\mathrm{f}_\theta(\cdot)$ is a CNN model with parameters θ, $\mathbf{z}$ is either the normalized image or the concatenated residual stack, $\sigma(\cdot)$ is the logistic function.

Training of the segmentation model is aimed at optimization the following composite loss:

$$L = \lambda_f \cdot L_{focal} + \lambda_j \cdot \left(1 - \mathrm{IoU}(\mathbf{y}, \hat{\mathbf{y}})\right) + \lambda_d \cdot (1 - \mathrm{Dice}(\mathbf{y}, \hat{\mathbf{y}})) \quad (13)$$

$$L_{focal} = -\alpha_t (1 - p_t) \log_2(p_t), \quad IoU = \frac{\sum_i p_i g_i}{\sum_i p_i + \sum_i g_i - \sum_i p_i g_i + \varepsilon} \quad (14)$$

$$Dice = \frac{2 \sum_i p_i g_i}{\sum_i p_i + \sum_i g_i + \varepsilon} \quad (15)$$

where L_{focal} is the binary focal loss; IoU and Dice are the Jaccard (known as Intersection-over-Union index) and Dice (also known as Dice-Sørensen coefficient) coefficients; $\lambda_f, \lambda_j, \lambda_d > 0$ values weigh the terms; $p_i \in [0; 1]$ is the predicted probability, $g_i \in \{0; 1\}$ is the ground truth, and $\varepsilon > 0$ is a constant used for prevention division by zero. Usage of composite loss in eq. (13) allows effectively mitigating with severe class imbalance [11, 12, 26].

4. Evaluation results

Evaluation of the proposed solution was performed on the ALASKA dataset [33] (subset of 10000 DI). Test images are standardized to grayscale with resolution set to 512×512 (pixels) and paired with binary change maps derived from pixelwise CI–SI differences. For each CI, the respective stego image is generated with HUGO [14], MiPOD [34], and MG [21, 35, 36] methods. Payload values Δ_α span low to moderate ranges, namely $\Delta_\alpha \in \{3\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$. Fixed train/validation/ test splits are reused across runs: 70% is designated for training, 10% is used for validation, and 20% is for testing.

The segmentation models based on considered architectures (U-Net, LinkNet, FPN, PSPNet) and backbones (ResNet, EfficientNet, MobileNetV2) are trained under identical data splits. Training procedure was aimed to minimize a composite loss (15) with Nadam optimizer and early stopping on validation loss. The best-validation-loss checkpoint is used for test evaluation.

The distribution of IoU values for models trained with and without an SRM filter (L5) are presented at Fig. 8-9. Fig. 8 data is averaged for all embedding methods and payloads, and backbone EfficientNet is chosen as the one, that showed the best results. Fig. 9 data is averaged for all segmentation models and payloads with EfficientNet model backbone.

Figures 8-9 show a clear rightward shift in IoU SRM residuals are appended to the input. The distributions are well separated: SRM boxes sit above for IoU (Fig. 8-9) their non-SRM counterparts with limited quartile overlap. Fig.8 shows that on average Unet, FPN, and Linknet segmentation models display comparable results (IoU approx. 0.19 for SRM, and 0.07 for non-SRM). At the same time PSPNet shows the worst

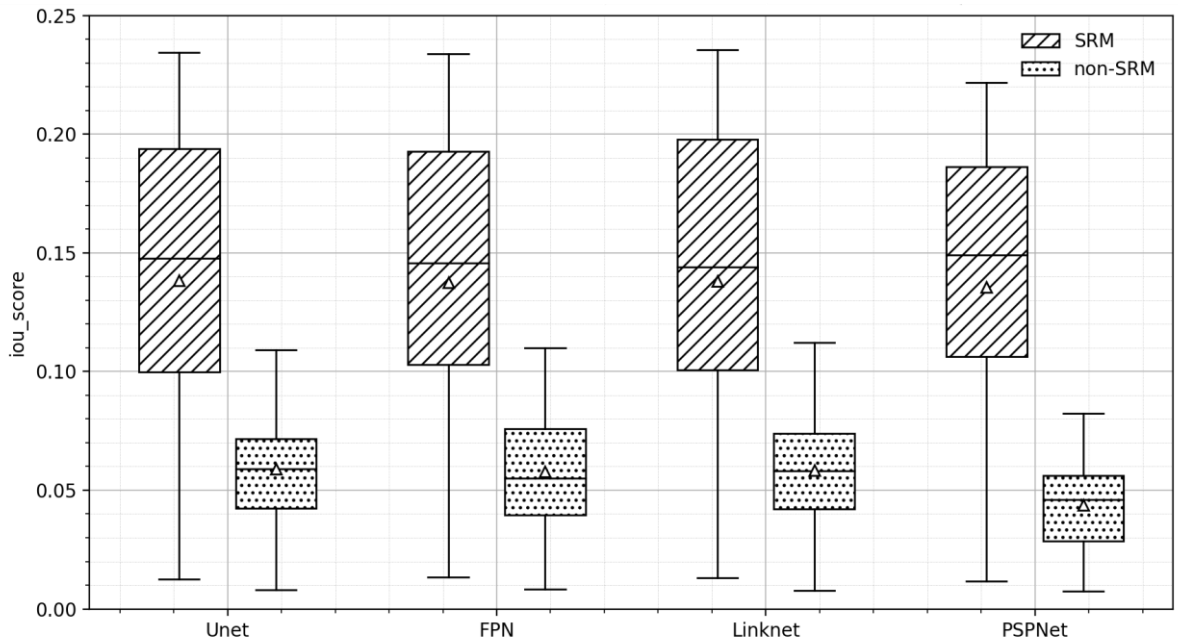

Figure 8: Distribution of IoU metric values with and without the SRM applied in the processing pipeline for all embedding methods, every segmentation model, and EfficientNet backbone model. The lines within boxes corresponds to mean value of distribution, while triangular point relates to estimated median value.

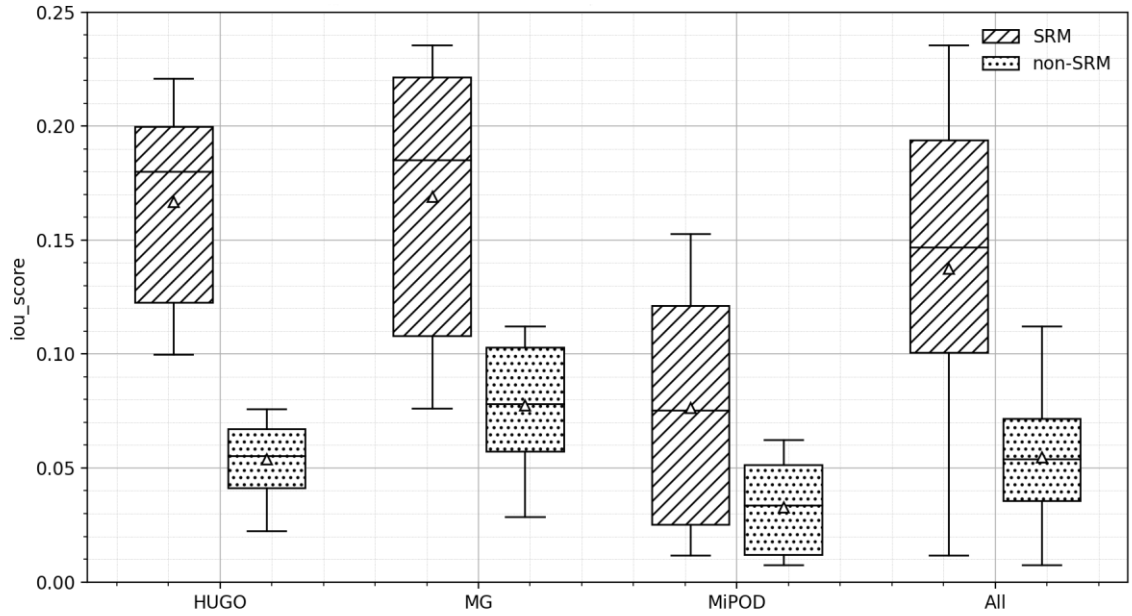

Figure 9: Distribution of IoU metric values with and without the SRM applied in the processing pipeline 0, and EfficientNet backbone model. The lines within boxes corresponds to mean value of distribution, while triangular point relates to estimated median value.

results in the group for both SRM and non-SRM cases. Such behavior can be explained by the workings of the model, i.e. use of pyramid pooling and deep downsampling of residual patterns of DI. This also aligns with the data shown on Fig. 8, difference in non-SRM case

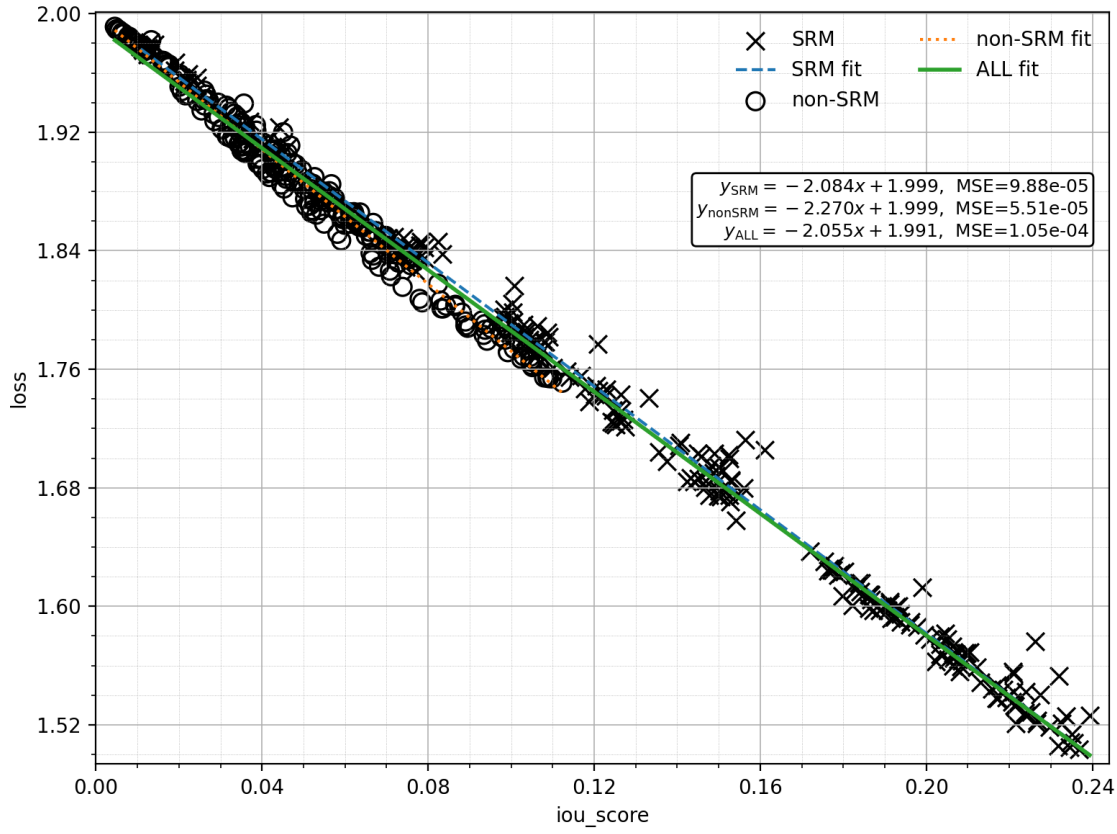

Figure 10. Spread of Loss metric values by IoU values with and without the SRM applied in the processing pipeline

with the rest of models is approx. 27% (max value), and in SRM case its only about 6%.

wise). Out of all the AEM's only MG has value spread intersection for SRM and non-SRM cases. This can be explained the that the embedding

Table 1. Changes of Intersection-over-Union index across HUGO, MG, and MiPOD embedding methods for considered segmentation models with EfficientNet network used as a backbone

Embedding method	Segmentation model	IoU (non-SRM), %	IoU (SRM), %	ΔIoU, %
Cover images payload $\Delta_\alpha = 3\%$				
HUGO	Unet	3.64%	9.99%	6.35%
	FPN	3.52%	10.30%	6.78%
	Linknet	3.59%	10.08%	6.49%
	PSPNet	2.23%	10.63%	8.40%
MG	Unet	4.73%	7.91%	3.18%
	FPN	4.46%	8.34%	3.88%
	Linknet	4.53%	7.61%	3.08%
	PSPNet	2.85%	7.82%	4.97%
MiPOD	Unet	0.81%	1.26%	0.45%
	FPN	0.83%	1.33%	0.50%
	Linknet	0.78%	1.31%	0.54%
	PSPNet	0.76%	1.18%	0.43%
Cover images payload $\Delta_\alpha = 20\%$				
HUGO	Unet	6.13%	18.41%	12.28%
	FPN	5.99%	17.72%	11.72%
	Linknet	5.95%	18.28%	12.33%
	PSPNet	4.61%	17.70%	13.09%
MG	Unet	8.95%	19.38%	10.43%
	FPN	8.90%	18.62%	9.72%
	Linknet	9.00%	18.41%	9.41%
	PSPNet	6.12%	17.98%	11.86%
MiPOD	Unet	3.54%	7.62%	4.08%
	FPN	3.40%	7.47%	4.06%
	Linknet	3.33%	7.79%	4.46%
	PSPNet	3.25%	7.48%	4.23%
Cover images payload $\Delta_\alpha = 50\%$				
HUGO	Unet	7.06%	21.68%	14.62%
	FPN	7.58%	21.70%	14.12%
	Linknet	7.55%	21.31%	13.76%
	PSPNet	6.22%	22.09%	15.87%
MG	Unet	10.87%	23.45%	12.58%
	FPN	10.99%	23.37%	12.38%
	Linknet	11.23%	23.56%	12.33%
	PSPNet	8.24%	22.16%	13.92%
MiPOD	Unet	6.21%	14.78%	8.56%
	FPN	5.98%	15.28%	9.31%
	Linknet	6.23%	14.41%	8.18%
	PSPNet	5.58%	15.23%	9.65%

Taking a look at Fig. 9 shows us, confirms previously obtained results, that using SRM preprocessing increases IoU score up to 5 times compared to non-SRM case. Also, Fig. 9 shows that MiPOD performs the best out of all AEM's, while MG shows the worst results (localization-

done by MiPOD usually goes to the smoother areas of a DI, where the SRM residuals are small. The same applies for the worst performing MG method, which favors high-frequency areas of DI. Worth mentioning the effect of SRM filter for HUGO AME: IoU score changes by 72%

(SRM vs non-SRM). Which can be explained by that, that SRM almost "matches" how HUGO works (3).

The spread of Loss metric values by IoU values with and without the SRM applied in the processing pipeline is represented on Fig. 10. We can see the tight, monotone trade-off: as IoU increases, loss decreases linearly, confirming that the composite loss is well aligned with the localization metric. Mean Square Error (MSE) linear approximation is $9.88 \cdot 10^{-5}$ for the SRM case, and $5.51 \cdot 10^{-5}$. Which supports that the spread follows linear distribution. The cloud forms several "tiers" along the same curve (visible around IoU 0.12, 0.16, and 0.20), consistent with configuration factors such as payload or architecture/backbone.

Markers separate clearly along the operating curve. Non-SRM values concentrate in the left–upper region (IoU less than 0.12; loss more than 1.76), with only a few points approaching the mid-tier. SRM values occupy the mid and right–lower regions and sets the envelope of best results, extending to IoU approx. 0.24 at loss approx. 1.52. In the overlap band (IoU 0.10–0.13, loss 1.74–1.80) both stacks appear, but beyond 0.15 IoU the points are predominantly SRM.

5. Discussion

Table 1 illustrates the delta of IoU metric with and without usage of SRM filters applied. Values provided for all used embedding methods, all segmentation models paired with EfficientNet backbone. Payload values vary across $\Delta_\alpha \in \{3\%, 20\%, 50\%\}$. The EfficientNet summary (Table 1) confirms this trend across embedding methods and architectures. At Δ_α= 3%, gains are modest for MiPOD (4%–5%) and moderate for HUGO/MG (3%–8%). At Δ_α=20%, gains widen to 9%–13% for HUGO/MG and 4%–4.5% for MiPOD. At Δ_α = 50%, gains are the largest: HUGO: 13.7%–15.9%, MG: 12.3%–14%, MiPOD: 8.2%–9.7%. Results often yielding more than 3 times relative IoU lift over non-SRM baselines. PSPNet consistently shows the strongest ΔIoU within each embedding method (e.g., HUGO: 8.4%/13.1%/15.9% at $\Delta_\alpha \in \{3\%, 20\%, 50\%\}$), indicating that global-context decoders benefit most from residual pre-emphasis.

Results indicate a consistent aggregate benefit from SRM preprocessing. Boxplots (Figs. 8–9) show clear separation: the median IoU rises from 0.05 (non-SRM) to 0.14–0.15 (SRM), and the median loss drops from 1.88 to 1.70. Quartile overlap is limited, and the IoU–loss scatter (Fig. 10) places SRM on the high-IoU/low-loss frontier, with non-SRM concentrated in the left–upper region. Thus, for the pooled experiments, SRM shifts the attainable operating regime toward better localization with lower loss values.

Across the experiment, encoder–decoder architectures with skip connections and multi-scale fusion (U-Net and FPN, with LinkNet close behind) consistently achieve the highest IoU score values. They preserve high-resolution spatial detail, which is critical for localizing weak, high-frequency stego residuals. PSPNet's values improve when SRM is used but generally trails off because pyramid pooling aggregates features coarsely and provides weaker boundary fidelity than skip-heavy decoders.

Backbone changes have a secondary, but noticeable, effect. EfficientNet typically yields the best IoU score (better low-level representations and compound scaling), ResNet-34 is a robust runner-up, and MobileNetV2 trades a small IoU drop for lower computational costs. Usage of SRM increases all IoU scores but does not change the ordering: architecture dominates, backbone fine-tunes the localization point.

Conclusions

Experimental evaluation results proved that applying of image preprocessing step, namely a single filter from SRM bank, allows for considerably (up to 5 times) improving localization accuracy of stego bits embedding into cover image by AEM. We revealed that skip-connected, multiscale decoders (namely, used in U-Net, FPN, LinkNet segmentation models) achieve the highest IoU, while PSPNet trails in absolute IoU. With EfficientNet backbone we observe changes of IoU index value (SRM vs. non-SRM) up to 16% for HUGO and MG embedding methods and typically approx. 9%–14% at moderate/high payloads ($\Delta_\alpha \geq 20\%$). Gains at $\Delta_\alpha = 3\%$ are smaller but yet still present. Among the used AEM's, stego bits embedded according to MG method is the easiest to localize, while HUGO is intermediate, and MiPOD is the hardest one. Backbone choice sets the accuracy–efficiency envelope: EfficientNet offers the best balance (localization vs. computing costs), ResNet is a robust

generalist, and MobileNetV2 reduces compute costs with modest IoU score values.

These results align with theory and results obtained for stegodetectors in stego image revealing tasks. The SRM acts as a residual prior that attenuates scene content and amplifies weak, high-frequency perturbations left by steganographic embedding. Architectures that fuse fine-scale detail through encoder–decoder skip connections exploit this residual emphasis most effectively (for example, U-Net, FPN and LinkNet models). Whereas PSPNet's large pooling bins trade boundary precision for global context, limiting its benefit. Usage of MG and HUGO embedding methods leads to producing sharper residuals, so SRM improves their separability more than MiPOD does, which explicitly smooths changes to avoid stegodetection.

The revealed best and worst cases underscore the effect size relative to non-SRM. The best combination is EfficientNet with Unet/FPN for accurately localization positions of stego bits embedded according to MG and HUGO methods. The worst one is applying of PSPNet segmentation model to process the stego images formed by MiPOD embedding method. These contrasts establish the practical value of SRM as a generic, plug-in preprocessor that raises the attainable localization frontier.

Obtained results helps us to construct a practical recipe for building a robust SD: use a skip-connected multiscale segmentation model (Unet or FPN) with an EfficientNet backbone, use SRM preprocessing unit, keep DI augmentations conservative to preserve embedding statistics.

References

[1] W. M. Eid, S. S. Alotaibi, H. M. Alqahtani, and S. Q. Saleh, "Digital image steganalysis: Current methodologies and future challenges," IEEE Access, vol. 10, pp. 92321–92336, 2022, doi: 10.1109/ACCESS.2022.3202905.

[2] J. Fridrich, "Steganography in Digital Media: Principles, Algorithms, and Applications," Cambridge Univ. Press, 2009, add ISSN, ch. 8 (overview of probabilistic embedding and STC; MiPOD context).

[3] J. Fridrich, "Breaking HUGO – the process discovery," Binghamton Univ., 2011, tech. report/slides

[4] P. Yatsura and D. Progonov, "A Review of Modern Methods for Steganalysis and Localization of Embedded Data in Digital Images," Theoretical and Applied Cybersecurity, vol. 5, no. 1, pp. 91–103, Nov. 2023. [Online]. Available: https://tacs.ipt.kpi.ua/article/view/328265/325734. doi: 10.32782/tacs-2023.1.9.

[5] P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System: The ins and outs of organizing BOSS," in Proc. SPIE Media Watermarking, Security, and Forensics, vol. 7880, 2011, pp. 78800K-1–78800K-13.

[6] Z. Jin, Z. Yang, Y. Chen, and J. Zhang, "IAS-CNN: Image adaptive steganalysis via convolutional neural network combined with selection channel," International Journal of Distributed Sensor Networks, vol. 16, no. 3, Mar. 2020.

[7] Fridrich, J., and Kodovský, J., "Rich Models for Steganalysis of Digital Images," IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868–882, 2012.

[8] I. C. Camacho and K. Wang, "Convolutional neural network initialization approaches for image manipulation detection," Digital Signal Processing, vol. 129, 103676, 2022. GIPSA-Lab

[9] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 2014, pp. 4206–4210. doi:10.1109/ICIP.2014.7025854.

[10] Z. Wang, O. Byrnes, H. Wang, et al., "Data Hiding With Deep Learning: A Survey Unifying Digital Watermarking and Steganography," IEEE Transactions on Multimedia, vol. 25, pp. 6423-6441, 2023.

[11] X. Shi, B. Tondi, B. Li, et al., "CNN-based steganalysis and parametric adversarial embedding:A game-theoretic framework," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3125-3139, 2020.

[12] I. Hussain, J. Zeng, and S. Tan, "A survey on deep convolutional neural networks for image steganography and steganalysis," KSII Trans. Internet Inf. Syst., vol. 14,

[13] М.М. Маманчук, Д.О. Прогонов, "Локалізація позицій стегобітів, вбудованих до зображень-контейнерів з використанням адаптивних стеганографічних методів HUGO та WOW," in Proc. Theoretical and Applied Cybersecurity (TACS-2023), Igor Sikorsky

Kyiv Polytechnic Institute (IFTI), Kyiv, Ukraine, 2023.

[14] S. Saha, P. Sarkar, M. K. Ghosh, et al., "WOW based Steganography for Digital Images," 2018 International Conference on Communication and Signal Processing, pp. 0003-0007, 2018.

[15] V. Himthani, V. S. Dhaka, M. Kaur, et al., "Comparative performance assessment of deep learning based image steganography techniques," Multimedia Tools and Applications, vol. 81, no. 2, pp. 1827-1851, 2022.

[16] T. Denemark, J. Fridrich, and V. Holub, "Further study on the security of S-UNIWARD," 2014 IEEE International Conference on Image Processing, pp. 3133-3137, 2014.

[17] V. Holub and J. Fridrich, "Digital image steganography by minimizing the probability of detection," in Proc. IEEE ICASSP, 2012, pp. 1785–1788.

[18] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in Information Hiding (IH 2010), LNCS 6387.

[19] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in Proc. IEEE WIFS, 2012, pp. 234–239.

[20] H. Kheddar, M. Hemis, Y. Himeur, et al., "Deep Learning for Diverse Data Types Steganalysis: A Review," Expert Systems with Applications, vol. 227, p. 119859, 2023.

[21] W. Tang, B. Li, M. Barni, et al., "Improving Cost Learning for JPEG Steganography by Exploiting JPEG Domain Knowledge," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 5313-5327, 2021.

[22] J. Kodovskỳ and J. Fridrich, "Quantitative steganalysis using rich models," IEEE Transactions on Information Forensics and Security, vol. 8, no. 8, pp. 1327-1338, 2013.

[23] Denemark, Holub, Fridrich, "Selection-Channel-Aware Rich Model for Steganalysis of Digital Images," WIFS 2014 (aka SCA-RM).

[24] M. Chaumont, "Deep learning in steganography and steganalysis," Multimedia Tools and Applications, vol. 79, no. 19-20, pp. 13813-13837, 2020.

[25] S. Zhang, H. Zhang, X. Zhao, et al., "A Deep Residual Multi-scale Convolutional Network for Spatial Steganalysis," Neural Processing Letters, vol. 50, no. 3, pp. 2505-2519, 2019.

[26] J. Zhu, X. Zhao, and Q. Guan, "Detecting and Distinguishing Adaptive and Non-Adaptive Steganography by Image Segmentation," Multimedia Tools and Applications, vol. 77, no. 23, pp. 31057-31073, 2018.

[27] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," CVPR, 2016.

[28] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning, pp. 6105-6114, 2019.

[29] M. Yedroudj, F. Comby, M. Chaumont, "Yedroudj-Net: An Efficient CNN for Spatial Steganalysis," IEEE TIFS, vol. 13, no. 11, pp. 2784–2797, 2018.

[30] S. Arivazhagan, E. Amrutha, W. S. L. Jebarani, et al., "Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning based techniques," Applied Soft Computing, vol. 93, p. 106380, 2020.

[31] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," IEEE VCIP, 2017.

[32] H. Zhao, J. Shi, X. Qi, et al., "Pyramid Scene Parsing Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 6230-6239, 2017.

[33] R. Cogranne, Q. Giboulot, and P. Bas, "The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis 'into the wild'," in *Proc. ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, Paris, France, Jul. 2019, pp. 125–137, doi: 10.1145/3335203.3335726

[34] S. Kang, H. Park, and J.-I. Park, "Identification of Multiple Image Steganographic Methods Using Hierarchical ResNets," Sensors, vol. 21, no. 17, p. 5865, 2021.

[35] J. Fridrich and J. Kodovský, "Multivariate Gaussian model for designing additive distortion for steganography," in Proc. IEEE ICASSP, Vancouver, Canada, 2013, pp. 2949–2953. dde.binghamton.edu+1.

[36] DDE Lab, "Steganographic Algorithms: MG," Binghamton Univ., software and notes, 2012.