پنهان‌نگاری معکوس‌پذیر در تصاویر رمزشده با استفاده از همبستگی پیکسل‌های مجاور و کدبندی حسابی

عمار محمدی[۱]، دانشجوی دکتری؛ منصور نخکش[۲]، دانشیار

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد - یزد - ایران - mohammadi_a@stu.yazd.ac.ir
۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد - یزد - ایران - nakhkash@yazd.ac.ir

چکیده: این مقاله روشی در پنهان نگاری معکوس پذیر در تصویر رمز شده را معرفی می کند که از همبستگی پیکسل های مجاور در تصویر بهره می برد. در روش پیشنهادی، تصویر اصلی می تواند با الگوریتم رمز دلخواه رمز شود. بیت هایی با ارزش بیشتر در پیکسل های تصویر به منظور ایجاد فضای خالی برای پنهان نگاری بیت های داده به کار گرفته می شوند. در این رویکرد تصویر به قالب[۱]های مجزا تقسیم و مرکزی ترین پیکسل در هر قالب به عنوان پیکسل مبنا در نظر گرفته می شود. خطای پیش بینی بین شدت پیکسل مبنا و شدت پیکسل های دیگر در قالب محاسبه و پیش بینی محلی نامیده می شود. این خطا به منظور بدست آوردن یک ویژگی از ظرفیت ذخیره هر قالب تحلیل می گردد. ویژگی های محاسبه شده برای تمام قالب ها توسط کدبندی حسابی، فشرده و ویژگی های فشرده شده به همراه بیت های داده در تصویر تعبیه می شوند. در گیرنده در ابتدا ویژگی های فشرده شده استخراج، سپس نافشرده[۲] شده و به منظور بازیابی بدون اتلاف تصویر اصلی و استخراج بیت های داده به کار گرفته می شوند. نتایج آزمایش تایید می کند که روش پیشنهادی روش های جدید در این زمینه را بهبود می دهد.

واژه‌های کلیدی: تصویر رمز شده، همبستگی، خطای پیش بینی، پنهان نگاری معکوس پذیر، کدبندی حسابی.

Reversible Data Hiding in Encrypted Images using Correlation of Neighboring Pixels and Arithmetic Coding

A. Mohammadi[1], PhD Student; M. Nakhkash[2], Associate Professor

1- Faculty of Electrical and Computer Engineering, Yazd University, Yazd, Iran, Email: mohammadi_a@stu.yazd.ac.ir
2- Faculty of Electrical and Computer Engineering, Yazd University, Yazd, Iran, Email: nakhkash@yazd.ac.ir

Abstract: This paper presents a reversible data hiding method in encrypted image that employs correlation of neighboring pixels in the image. In the proposed method, original image may be encrypted by desire encryption algorithm. More significant bits of the pixels in the image are exploited to vacate room for embedding data bits. In the approach, image is divided into separated blocks and most central pixel of each block is considered as reference one. The prediction error between the intensity of other pixels and reference one is calculated and denoted local prediction. This error is analyzed determining a feature of block embedding capacity. Calculated features for all blocks are compressed employing arithmetic coding and embedded in the image along with data bits. At the recipient, at first, compressed features are extracted, then they are uncompressed and used to lossless reconstruction of the original image and extraction of the data bits. Experimental results confirm that the proposed algorithm outperforms state of the art ones.

Keywords: Encrypted image, Correlation, Prediction error, Reversible data hiding, Arithmetic coding.

تاریخ ارسال مقاله: ۱۳۹۸/۰۸/۱۱
تاریخ اصلاح مقاله: ۱۳۹۸/۰۹/۲۲
تاریخ پذیرش مقاله: ۱۳۹۸/۰۹/۲۳
نام نویسنده مسئول: منصور نخکش
نشانی نویسنده مسئول: ایران - یزد - دانشگاه یزد - دانشکده مهندسی برق و کامپیوتر.

۱- مقدمه

پنهان نگاری معکوس پذیر[۳] در تصویر رشد چشمگیری در سال های اخیر داشته است [۱]. در پنهان نگاری معکوس پذیر داده در تصویر به صورتی پنهان می شود که تصویر اصلی به شکل بدون اتلاف[۴] قابل بازیابی باشد و داده بدون خطا استخراج شود. روش های مطرح در پنهان نگاری معکوس پذیر به سه دسته کلی تقسیم می شوند که عبارتند از: گسترش تفاوت[۵] [۲] ، تغییر هیستوگرام[۶] [۳] و فشرده سازی بدون اتلاف [۴]. روش ها ی پیشرفته تر همچنین از روش مکمل پیش بینی خطا[۷] به منظور افزایش ظرفیت ذخیره در سطح مشخصی از اعوجاج تصویر بهره می برند [۵-۹]. هرچقدر پیش بینی بهتری انجام شود هیستوگرام تیزتری[۸] از خطا فراهم می شود.

پنهان نگاری معکوس پذیر در تصویر رمز شده، نیازی است که به واسطه محاسبات ابری[۹] مطرح شده و توجه محققان بسیاری را در سال های اخیر بر انگیخته است. در این نوع از پنهان نگاری سه طرف عامل داریم که عبارتند از: صاحب تصویر، پنهان کننده داده و گیرنده. صاحب تصویر که اعتمادی به متولی کانال ارتباطی ندارد به منظور حفظ حریم شخصی تصویر را قبل از ارسال به کانال رمز می کند. باید خاطرنشان کرد صاحب تصویر علاقه ای به فشرده سازی تصویر قبل از رمز نگاری ندارد. در سمت دیگر، پنهان کننده پیام که می تواند متولی کانال باشد در حالی که مجوزی برای دسترسی به اطلاعات تصویر اصلی ندارد اجازه دارد داده های مفیدی را در تصویر رمز شده جاسازی کند. روشی که پنهان کننده داده به کار می برد باید به شکلی باشد که بازیابی بدون اتلاف تصویر اصلی و استخراج بدون خطای داده را تضمین کند. این چالشی است که موجب گسترش روش های پنهان نگاری معکوس پذیر در تصویر رمز شده شده است.

روش های مطرح در پنهان نگاری معکوس پذیر در تصویر رمز شده به سه دسته اصلی تقسیم می شوند. که عبارتند از: الگوریتم هایی مبتنی بر ایجاد فضا قبل از رمز نگاری (**افقر**) [۱۰-۱۷]، ایجاد فضا توسط رمز نگاری (**افتر**) [۱۸-۲۴] و ایجاد فضا بعد از رمز نگاری (**افبر**) [۲۵-۳۰].

در **افقر** پیش از رمز نگاری پیش پردازشی به منظور ایجاد فضای خالی محقق می شود.

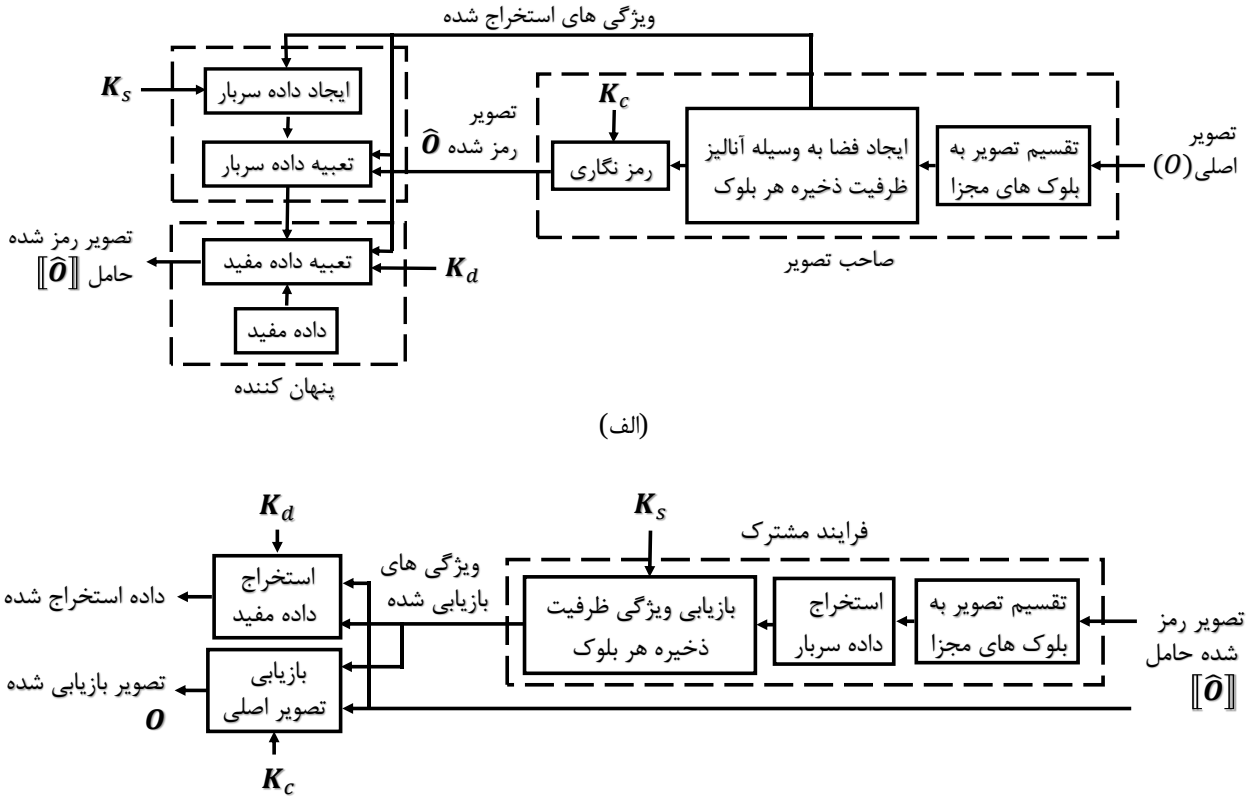
بیشتر روش ها در **افتر** بر مبنای استفاده از بایت های رمز یکسان به منظور رمز کردن چند پیکسل مجاور در تصویر شکل گرفته اند. در این روش ها همبستگی پیکسل ها مجاور در فرایند رمز نگاری حفظ می شود. اگرچه به واسطه حفظ این همبستگی پنهان کننده می تواند داده را در تصویر رمز شده پنهان کند وجود این همبستگی می تواند به معنای نشت اطلاعات تصویر اصلی نیز باشد.

روش های مطرح در افقر و افتر روش های جدایی پذیر هستند به این معنا که استخراج داده در گیرنده به داشتن تصویر رمز گشایی شده گره نخورده است. به بیان دیگر در روش های جدایی ناپذیر استخراج پیام تنها با داشتن تصویر رمز گشایی شده در گیرنده امکان پذیر است.

در طرح های **افبر** پنهان کننده هیچ اطلاعاتی از تصویر اصلی ندارد. تصویر اصلی رمز می شود و پنهان کننده ای که کاملا نا آگاه به اطلاعات تصویر اصلی است در تصویر رمز شده فضایی به منظور پنهان نگاری داده فراهم می کند. بعضی از روش های **افبر** مطرح شده جدایی پذیر [۲۵، ۲۷] و برخی دیگر جدایی ناپذیر [۲۶، ۲۸، ۲۹] هستند. در طرح [۳۰] دو روش مجزا یکی جدایی پذیر و دیگری جدایی ناپذیر مطرح شده است.

روش مطرح شده در این مقاله از نوع **افقر** است. به بررسی روش های مطرح شده در این حوزه می پردازیم. با استفاده از روش های سنتی پنهان نگاری معکوس پذیر، Ma و همکاران [۱۱] با پنهان نگاری کم ارزش ترین بیت های[۱۰] برخی از پیکسل ها در پیکسل های دیگر فضایی به منظور پنهان نگاری بیت های داده فراهم می کنند. لازم به ذکر است که این فضا قبل از رمز نگاری تصویر فراهم می شود. بعد از رمز نگاری فضای فراهم شده به منظور تعبیه بیت های پیام به کار گرفته می شود. طرح پیشنهادی در [۱۵] روش [۱۱] را با معرفی سطح بهم پیوسته تنک[۱۱] بهبود دادند. همچنین Puteaux و Puech [۱۷] دو روش متفاوت بر مبنای **افقر** معرفی کرده اند. این روش ها ظرفیت بالایی را به منظور پنهان نگاری بیت های داده فراهم می کند. در یکی از این روش ها بازیابی بدون اتلاف تصویر اصلی در گیرنده امکان پذیر است. این روش همبستگی پیکسل های مجاور را به خدمت می گیرد به طوری که یک پیکسل به وسیله پیکسل های کناری خود پیش بینی می شود. بر این اساس خطای پیش بینی برای هر پیکسل محاسبه و بر اساس این خطا ها برچسب هایی در نظر گرفته می شود. این برچسب ها شامل یک بیت صفر و یا یک است. بعد از رمز نگاری بر اساس این بر چسب ها بیت های داده در با ارزش ترین بیت[۱۲] های پیکسل های رمز شده تعبیه می شوند. از آنجایی که این برچسب ها بر اساس پیش بینی خطا شکل گرفته اند می توان با انتخاب پیش بینی کننده های مختلف روش های متفاوتی را به منظور فراهم آوردن این برچسب ها پیشنهاد کرد. همچنین با افزایش تعداد بیت های اختصاص یافته برای هر برچسب بیت های بیشتری از پیکسل های رمز شده را به منظور پنهان نگاری بیت های داده به کار گرفت. برای مثال [۱۲] پیش بینی کننده MED[۱۳] را به منظور بدست آوردن خطای پیش بینی به خدمت گرفت و در پی آن بر چ سب هایی متنا سب با خطای محا سبه شده در نظر گرفته می شوند.

الگوریتم مطرح شده در این مقاله با الهام از طرح [۱۷] مطرح می شود. در طرح پیشنهادی از پیش بینی کننده محلی به منظور پیش بینی خطا استفاده کرده ایم. این پیش بینی کننده برای اولین بار در [۹] مطرح شده است. بر این اساس مرکزی ترین پیکسل در هر قالب به عنوان پیکسل مبنا در نظر گرفته شده و تفاوت پیکسل های دیگر از پیکسل مبنا در هر قالب خطای پیش بینی را محقق می سازد. با بررسی خطا می توان ویژگی از ظرفیت ذخیره هر قالب را در نظر گرفت.



شکل ۱: نمای کلی از طرح پیشنهادی. (الف) پنهان نگاری داده و ایجاد تصویر رمز شده حامل. (ب) استخراج داده و بازیابی تصویر اصلی.

این ویژگی مشخص می کند که هر قالب چه ظرفیتی را می تواند به منظور پنهان نگاری داده بعد از رمز نگاری فراهم سازد. این ویژگی ها برای تمام قالب ها محاسبه شده و توسط کدبندی حسابی به صورت بدون اتلاف فشرده می شوند و ویژگی های فشرده شده به همراه بیت های داده بعد از رمز نگاری در تصویر رمز شده به شکل معکوس پذیر تعبیه و تصویر رمز شده حامل را می سازند. در گیرنده در ابتدا ویژگی ها استخراج و بازیابی می شوند. با بهره گیری از آنها، بیت های داده استخراج و تصویر اصلی، بدون اتلاف بازیابی می گردد. طرح پیشنهادی از نوع جدایی پذیر است. روش پیشنهادی بهبود محسوسی را نسبت به روش هایی که ظرفیت بالایی را فراهم کرده اند نشان می دهد.

۲- روش پیشنهادی

در این بخش روش پیشنهادی که شامل تحلیل ظرفیت فراهم شده، ایجاد فضا قبل از رمز نگاری، رمز نگاری تصویر، تعبیه بیت های داده و استخراج ویژگی، استخراج داده مفید و بازیابی تصویر اصلی است به تفصیل بیان می شود. در شکل ۱ شمای کلی از طرح پیشنهادی مطرح شده است. شکل ۱-الف روش پنهان نگاری در تصویر رمز شده را تشریح می کند که منجرب به ایجاد تصویر رمز شده حامل می گردد و شکل ۲-ب فرایند استخراج بیت های داده و بازیابی تصویر اصلی را شرح می دهد. با توجه به شکل ۱-الف به منظور پنهان نگاری داده، تصویر در ابتدا به قالب های مجزا تقسیم و با پیش پردازش، ویژگی مربوط به هر قالب محاسبه می گردد. این ویژگی ها به منظور تعبیه بیت های داده بعد از رمز نگاری تصویر به کار گرفته می شوند. پیکسل های تصویر می تواند به هر روش و یا با هر الگوریتم استانداردی رمز شوند و حتی قالب ها نیز می توانند حین رمز نگاری جا به جا شوند به شرطی که ویژگی متناظر با هر قالب در دسترس و مکان بیت ها در هر قالب بدون تغییر باشد. رمز نگاری می تواند به واسطه یک کلید امن ورودی (K_c) به الگوریتم رمز محقق می شود. به علاوه این ویژگی ها فشرده و به وسیله کلید K_s توسط پنهان کننده داده رمز و به عنوان داده سربار در نظر گرفته می شود. K_s کلیدی است که به صورت مشترک در دسترس مالک تصویر و پنهان کننده داده قرار دارد. داده سربار به شکل سلسله مراتبی به همراه داده مفید در تصویر رمز شده تعبیه و در نتیجه تصویر رمز شده حامل ساخته می شود. داده مفید قبل از تعبیه با کلید K_d توسط پنهان کننده داده رمز می شود. در گیرنده همان طور که در شکل ۱-ب نمایش داده شده است استخراج بیت های داده و بازیابی بدون اتلاف تصویر اصلی محقق می شود. با توجه به شکل تصویر در ابتدا به قالب های مجزا تقسیم می شود. سپس ویژگی قالب ها استخراج، توسط کلید K_s رمز گشایی و نافشرده شده و به منظور بازیابی تصویر اصلی و استخراج بیت های داده به کار گرفته می شوند. اگر چه انجام این پردازش می تواند به طور مستقل به وسیله مالک تصویر و یا پنهان کننده داده انجام شود انجام آن توسط پنهان کننده داده عملی تر است. باید توجه شود که استخراج این ویژگی ها

خود به در اختیار داشتن این ویژگی ها وابسته است این تناقض به وا سطه یک شیوه سل سله مراتبی که بعدا مطرح خواهد شد حل می شود. در نهایت تصویر ا صلی به صورت بدون اتلاف با ا ستفاده از کلید K_c و ویژگی های استخراج شده بازیابی و داده مفید توسط پنهان کننده داده تنها با داشتن K_d به شکل مجزا استخراج می گردد.

۲-۱- تحلیل ظرفیت ذخیره فراهم شده

در [۱۷] که یک روش پنهان نگاری ظرفیت بالا در تصویر رمز است خطای پیش بینی (𝔢) بر طبق رابطه زیر محاسبه می شود.

$$𝔢 = 𝔦 - 𝔦' \qquad (۱)$$

این رابطه تفاوت بین شدت یک پیکسل (𝔦) و مقدار پیش بینی آن ($𝔦'$) را به عنوان خطای پیش بینی مشخص می کند. [۱۷] نشان می دهد که اگر

$$|𝔢| < 64 \qquad (۲)$$

برقرار با شد یک بیت داده می تواند در با ارزش ترین بیت 𝔦 تعبیه شود به طوری که در گیرنده توسط $𝔦'$ قابل بازیابی باشد. الگوریتم مطرح شده در [۱۷] می تواند حداکثر یک بیت در هر پیکسل پنهان کند. با الهام از رابطه (۲)، می توان ظرفیت ذخیره بیشتری برای هر پیکسل با در نظر گرفتن رابطه (۳) فراهم کرد.

$$|𝔢| < 2^n \quad 0 \le n \le 7 \qquad (۳)$$

بر این اساس ظرفیت ذخیره در هر 𝔦 که با n' نمایش داده می شود توسط رابطه (۴) بدست می آید.

$$\begin{cases} n' = 8 - n - 1 & n \neq 0 \\ n' = 8 & n = 0 \end{cases} \qquad (۴)$$

بر طبق (۴)، $n' = 7$ هرگز محقق نمی شود به این دلیل که یک بیت به منظور علامت خطای پیش بینی شده در نظر گرفته و از مقدار ظرفیت فراهم شده کم می شود. برای $n = 0$، رابطه (۳) مقدار $𝔢 = 0$ را فراهم می کند و بنابراین ۸ بیت از داده می تواند در 𝔦 تعبیه شود. به بیان دیگر زمانی که شدت پیکسل دقیقا برابر با مقدار پیش بینی شده می با شد می تواند به صورت کامل با بیت های داده جایگزین شود. در $n=7$، خطای پیش بینی می تواند هر مقداری در $-128 < 𝔢 < 128$ داشته باشد. بدین معنی است که هیچ ظرفیتی برای پنهان نگاری بیت های داده وجود ندارد. در $n = 6$، فقط یک بیت ظرفیت $(n' = 1)$ همان طور که در [۱۷] مطرح شده است، فراهم می شود.

رو ند پنهان نگاری داده در شدت یک پیکسل مثلا $𝔦 = 𝔦_7𝔦_6𝔦_5𝔦_4𝔦_3𝔦_2𝔦_1𝔦_0$ که شامل هشت بیت از کم ارزش ترین بیت $𝔦_0$ تا

$\mathbb{I}_{f_k}(1)$	$\mathbb{I}_{f_k}(2)$		$\mathbb{I}_{f_k}(3)$
$\mathbb{I}_{f_k}(4)$	$𝔦_{l_k}$		$\mathbb{I}_{f_k}(5)$
$\mathbb{I}_{f_k}(6)$	$\mathbb{I}_{f_k}(7)$		$\mathbb{I}_{f_k}(8)$

شکل ۲: پیکسل مبنا و پیکسل های مجاور در یک بلوک با اندازه 3×3

$\mathbb{E}_k(1)$	$\mathbb{E}_k(2)$		$\mathbb{E}_k(3)$
$\mathbb{E}_k(4)$	$𝔦_{l_k}$		$\mathbb{E}_k(5)$
$\mathbb{E}_k(6)$	$\mathbb{E}_k(7)$		$\mathbb{E}_k(8)$

شکل ۳: خطای پیش بینی برای یک بلوک با اندازه 3×3

پرارزش ترین آن $𝔦_7$ می باشد توسط رابطه (۵) محقق می شود. در این رابطه $D = \{d_0, d_1, \dots, d_{n'-1}\}$، n' بیت از داده بر روی 𝔦 پنهان و پیکسل حامل $[\![𝔦]\!]$ را می سازد.

$$[\![𝔦]\!] = \sum_{k=1}^{n'} (2^{8-k} \times d_{k-1}) + \sum_{k=n'+1}^{8} (𝔦_{8-k} \times 2^{8-k}) \qquad (۵)$$

در این رابطه $\sum_{i=n}^{m}$ در صورتی که $n > m$ باشد صفر در نظر گرفته می شود. در یک توصیف مختصری از روند پنهان نگاری داده در ابتدا n توسط رابطه (۳) محاسبه می شود سپس n' به وسیله رابطه (۴) بدست می آید و در نهایت n' بیت از داده در 𝔦 توسط رابطه (۵) تعبیه می گردد.

به عنوان مثال روند پنهان نگاری را با فرض $|𝔢| < 32$ تشریح می کنیم. در ابتدا با استفاده از (۳)، $n = 5$ محاسبه می شود. سپس $n' = 2$ با در نظر گرفتن n و رابطه (۴) بدست می آید. در نهایت با استفاده از رابطه (۵) d_0 و d_1 در 𝔦 تعبیه و پیکسل حامل $[\![𝔦]\!]=d_0d_1𝔦_5𝔦_4𝔦_3𝔦_2𝔦_1𝔦_0$ را می سازد.

در سمت گیرنده، واضح است که n' بیت داده در بیت هایی با ارزش بالاتر $[\![𝔦]\!]$ تعبیه شده اند. بنابراین با دانستن n'، می توان داده را استخراج و پیکسل اصلی 𝔦 را بازیابی کرد. خاطرنشان می کنیم با داشتن n' مقدار n به راحتی محاسبه می شود و برعکس. در $n = 0$، $𝔦 = 𝔦'$ و در نتیجه 𝔦 به راحتی بازیابی می شود. این در حالی ا ست که برای $0 < n < 7$ تنها یک مقدار واحد برای 𝔦 وجود دارد که خطای پیش بینی آن در رابطه (۳) صدق می کند. برای اثبات می توانیم خطای پیش بینی را به صورت زیر دوباره نویسی کنیم.

$$-2^n < 𝔢 = (𝔦 - 𝔦') < 2^n \qquad (۶)$$

با توجه به (۶) هر تغییری در n' بیت با ارزش بالاتر از 𝔦 می تواند دستکم $\pm 2^{8-n'}$ تغییر در مقدار 𝔦 ایجاد کند. به بیان دیگر این امر افزایش و یا کاهشی در مقدار خطا دستکم به اندازه $2^{8-n'} = 2^{n+1}$ را موجب می شود. باید توجه داشت که مقدار جدید خطا $𝔢_n$ در رابطه $𝔢_n = 𝔢 + 2^{n+1}$ و یا $𝔢_n = 𝔢 - 2^{n+1}$ صدق می کند و در نتیجه $𝔢_n$ چون خود خطای جدید است می باید در رابطه ۶ صدق کند. بنابراین داریم $𝔢_n > 2^n$ و یا $𝔢_n < -2^n$ که این خود در تناقض با رابطه (۶) است. نتیجه اینکه تنها یک مقدار واحد برای مقدار بازیابی شده 𝔦 وجود دارد که بتواند محددیت خطا در رابطه (۶) را برآورده سازد.

۲-۲- ایجاد فضا قبل از رمز نگاری

در این بخش روش ما به منظور ایجاد فضا قبل از رمز نگاری تشریح می گردد. در طرح [۹] روش پیش بینی محلی به منظور پنهان نگاری پیام در تصاویر پزشکی مطرح شده است. در پیش بینی محلی تفاوت پیکسل های مجاور از یک پیکسل مبنا در یک قالب از تصویر بدست می آید که خطای پیش بینی را در قالب شکل می دهد. در روش پیشنهادی، ما از ایده پیش بینی محلی به منظور ایجاد فضا قبل از رمز نگاری بهره می بریم.

فرض می کنیم تصویری داریم که شامل $M \times L$ پیکسل است. این تصویر به تعداد N_b قالب با اندازه $m \times l$ تقسیم می شود. این قالب ها را به صورت $B = \{b_1, b_2, \dots, b_k, \dots, b_{N_b}\}$ نمایش می دهیم. این تقسیم می تواند با انتخاب اندازه های ۲ × ۲ ، ۳ × ۳ و یا ۴ × ۴ برای قالب ها محقق شود. در این قالب ها یک پیکسل مبنا وجود دارد که به منظور پیش بینی بقیه پیکسل ها به کار گرفته می شود. به طور کلی تمام پیکسل ها در تصویر می توانند به سه دسته $\mathbb{I}_f$، $\mathbb{I}_l$ و $\mathbb{I}_b$ تقسیم شوند. $\mathbb{I}_l$ پیکسل های مبنا در کل تصویر و $\mathbb{I}_f$ پیکسل های مجاور این پیکسل ها در قالب ها هستند. $\mathbb{I}_b$ پیکسل های مرزی در سطر و یا ستون های آخر تصویر هستند که در یک قالب قرار نمی گیرند. در (۷)، (۸) و (۹) این دسته پیکسل ها نمایش داده شده اند.

$$\mathbb{I}_l = \left\{\mathbb{i}_{l_1}, \mathbb{i}_{l_2}, \dots, \mathbb{i}_{l_k}, \dots, \mathbb{i}_{l_{N_b}}\right\} \quad (۷)$$

$$\mathbb{I}_f = \left\{\mathbb{I}_{f_1}, \mathbb{I}_{f_2}, \dots, \mathbb{I}_{f_k}, \dots, \mathbb{I}_{f_{N_b}}\right\} \quad (۸)$$

$$\mathbb{I}_b = \{\mathbb{i}_1, \mathbb{i}_2, \mathbb{i}_3, \dots., \mathbb{i}_{k'}, \dots, \mathbb{i}_{K'}\} \quad (۹)$$

در جایی که K' تعداد پیکسل های مرزی و $\mathbb{i}_{l_k}$ نمایش شدت یک پیکسل مبنا و $\mathbb{i}_{k'}$ نمایشی از شدت یک پیکسل مرزی است. همچنین $\mathbb{I}_{f_k}$ یک مجموعه از پیکسل های مجاور در قالب kام است که $\mathbb{i}_{l_k}$ پیکسل مبنا در آن قالب است.

شکل ۲ نمایشی از kامین قالب با فرض اندازه قالب ۳ × ۳ می باشد. در این شکل $\mathbb{I}_{f_k}(i)$، $1 \le i \le 8$، ۸ پیکسل مجاور را در قالب kام تصویر به نمایش می گذارد.

همان طور که گفته شد بین $\mathbb{i}_{l_k}$ و $\mathbb{I}_{f_k}$ (پیکسل های مجاور) در قالب kام همبستگی وجود دارد. تفاوت محلی بین $\mathbb{I}_{f_k}$ و $\mathbb{i}_{l_k}$ به عنوان معیاری از این همبستگی در نظر گرفته می شود. این تفاوت، خطای پیش بینی نامیده شده و توسط رابطه زیر بدست می آید.

$$\mathbb{E}_k(i) = \mathbb{I}_{f_k}(i) - \mathbb{i}_{l_k} \quad 1 \le i \le 8 \quad (۱۰)$$

خطای پیش بینی، به دلیل افزونگی کمتر نسبت به پیکسل های اصلی می توانند ظرفیت بیشتری به منظور پنهان نگاری داده فراهم کنند. با در نظر گرفتن اندازه قالب ۳ × ۳، ماتریس خطای پیش بینی با محاسبه $\mathbb{E}_k(i)$ برای $1 \le i \le 8$ محاسبه می شود (شکل ۳).

به منظور بهره گیری از روش پنهان نگاری که در بخش ۲-۱ مطرح شد برای قالب k ماکزیمم مقدار از اندازه المان های ماتریس $\mathbb{E}_k$ بدست می آید و m_k نامیده می شود. m_k به جای $|m|$ در (۳) به منظور محاسبه n جایگزین و n محاسبه شده n_k نامید می شود. در ادامه، n'_k با استفاده از رابطه (۴) با جایگزین کردن n_k به جای n محاسبه می شود. به دلیل اینکه خطایی در قالب یافت نمی شود که اندازه آن بیشتر از m_k باشد، n'_k بیت کمترین ظرفیتی است که می تواند توسط تمام پیکسل های مجاور در قالب kام فراهم شود. n'_k به عنوان ویژگی از ظرفیت فراهم شده قالب kام در نظر گرفته می شود. تعداد پیکسل ها در یک قالب با اندازه $m \times l$ برابر $(m \times l - 1)$ است بنابراین کل ظرفیت ذخیره داده برای قالب kام از رابطه زیر بدست می آید.

$$\mathbb{C}_{b_k} = (n'_k) \times (m \times l - 1) \quad (۱۱)$$

با توجه به تعداد کل قالب ها در تصویر که N_b است ظرفیت ذخیره داده برای کل تصویر از رابطه زیر بدست می آید.

$$\mathbb{C} = \sum_{k=1}^{k=N_b} \mathbb{C}_{b_k} \quad (۱۲)$$

مجموعه $\mathbb{N}' = \{n'_1, n'_2 \dots, n'_k, \dots, n'_{N_b}\}$ ویژگی ظرفیت ذخیره داده برای قالب های 1 تا N_b را شامل می شود. این مجموعه در گیرنده به منظور استخراج داده و بازیابی تصویر اصلی به کار گرفته می شود. اعضای این مجموعه به صورت سلسله مراتبی توسط کدبندی حسابی فشرده و سپس به منظور ایجاد داده سربار رمز می شوند. بعد از رمز نگاری تصویر اصلی، داده سربار به همراه داده مفید توسط $\mathbb{N}'$ پنهان نگاری می گردند.

۲-۳- رمز کردن تصویر اصلی

در طرح پیشنهادی از هر الگوریتم و یا روش رمز نگاری می توان بهره برد. بدین ترتیب رمز نگاری می تواند توسط کلید K_c محقق شود. تنها محدودیت در طرح پیشنهادی این است که در رمز نگاری در هر قالب چیدمان پیکسل ها و یا بیت های هر پیکسل نباید تغییر کنند و یا حداقل اینکه این تغییرات در اختیار پنهان کننده قرار بگیرد. اگر تصویر اصلی را با O نمایش دهیم $\hat{O}$ نمایشی از تصویر رمز شده است. بر این اساس می توان پیکسل های تشکیل دهنده $\hat{O}$ را با توجه به (۷)، (۸) و (۹) به سه دسته به صورتی که در ادامه می بینیم نمایش داد.

$$\hat{\mathbb{I}}_l = \left\{\hat{\mathbb{i}}_{l_1}, \hat{\mathbb{i}}_{l_2}, \dots, \hat{\mathbb{i}}_{l_k}, \dots, \hat{\mathbb{i}}_{l_{N_b}}\right\} \quad (۱۳)$$

$$\hat{\mathbb{I}}_f = \left\{\hat{\mathbb{I}}_{f_1}, \hat{\mathbb{I}}_{f_2}, \dots, \hat{\mathbb{I}}_{f_k}, \dots, \hat{\mathbb{I}}_{f_{N_b}}\right\} \quad (۱۴)$$

$$\hat{\mathbb{I}}_b = \{\hat{\mathbb{i}}_1, \hat{\mathbb{i}}_2, \hat{\mathbb{i}}_3, \dots., \hat{\mathbb{i}}_{k'}, \dots, \hat{\mathbb{i}}_{K'}\} \quad (۱۵)$$

۲-۴- پنهان نگاری داده سربار و داده مفید

با توجه به اینکه استخراج داده و بازیابی تصویر اصلی در گیرنده با دانستن مقادیر $\mathbb{N}'$ محقق می شود بنابراین $\mathbb{N}'$ باید در تصویر رمز شده تعبیه شود. بنابراین $\mathbb{N}'$ به صورت سلسله مراتبی فشرده، رمز و به همراه داده مفید در تصویر رمزشده تعبیه می شود. بر این اساس $\mathbb{N}'$ تبدیل به I زیر مجموعه به صورت $\mathbb{N}' = \{\mathbb{N}'_1, \mathbb{N}'_2, \dots, \mathbb{N}'_i, \dots, \mathbb{N}'_I\}$ می شود. هر $\mathbb{N}'_i$ شامل N_b/I ویژگی از ظرفیت ذخیره قالب ها می باشد. برای مثال $\mathbb{N}'_1 = \left\{n'_1, n'_2, .., n'_{\left(\frac{N_b}{I}\right)}\right\}$ برابر N_b/I ویژگی اول است. از آنجایی که این ویژگی ها متعلق به قالب های مجاور هستند می توان با فشرده کردن تفاوت این ویژگی ها توسط کدبندی حسابی به کارایی بهتری در فشرده سازی دست یافت. با رمز کردن داده های فشرده شده داده های سربار برای هر دسته با نمایش $\widehat{\mathcal{N}}' = \{\widehat{\mathcal{N}}'_1, \widehat{\mathcal{N}}'_2, \dots, \widehat{\mathcal{N}}'_i, \dots, \widehat{\mathcal{N}}'_I\}$ ساخته می شود. در ادامه نحوه پنهان نگاری داده شامل سربار و مفید را در تصویر رمز شده تشریح می کنیم. قالب های متناظر با هر $\widehat{\mathcal{N}}'_i$ را به صورت B_i که خود از N_b/I زیر قالب تشکیل شده است در نظر می گیریم برای مثال $B_1 = \{b_1, b_2, \dots, b_k, \dots, b_{(N_b/I)}\}$. بنابراین مجموعه قالب ها را می توان به صورت $B = \{B_1, B_2, \dots, B_i, \dots, B_I\}$ بازنویسی کرد. بعد از رمز نگاری تصویر می توانند به صورت $\hat{B} = \{\hat{B}_1, \hat{B}_2, \dots, \hat{B}_i, \dots, \hat{B}_I\}$ در نظر گرفته شود. در این صورت هر $\hat{B}_i$ شامل N_b/I قالب رمز شده کوچک تر است برای مثال $\hat{B}_1 = \{\hat{b}_1, \hat{b}_2, \dots, \hat{b}_k, \dots, \hat{b}_{(N_b/I)}\}$. شکل ۴ نمایی از این دسته بندی ها را برای اندازه قالب ۳ × ۳ نشان می دهد. برای شروع پنهان نگاری $\widehat{\mathcal{N}}'$ به عنوان داده سربار باید در $\hat{B}$ تعبیه شود. این امر به صورت سلسله مراتبی و با تعبیه $\widehat{\mathcal{N}}'_2$ در $\hat{B}_1$، $\widehat{\mathcal{N}}'_3$ در $\hat{B}_2$ و به طور کلی $\widehat{\mathcal{N}}'_i$ در $\hat{B}_{i-1}$ و $\widehat{\mathcal{N}}'_I$ در $\hat{B}_{I-1}$ محقق می شود. نتیجه این پنهان نگاری قالب های حامل $\{[\![\hat{B}_1]\!], [\![\hat{B}_2]\!], \dots, [\![\hat{B}_{i-1}]\!], \dots, [\![\hat{B}_{I-1}]\!]\}$ است. بعد از پنهان نگاری هر $\widehat{\mathcal{N}}'_i$ در $\hat{B}_{i-1}$، باقی مانده ظرفیت به منظور تعبیه داده مفید به کار گرفته و ظرفیت قالب آخر $(\hat{B}_I)$ تماما به منظور پنهان نگاری داده مفید استفاده می شود. داده مفید قبل از تعبیه توسط K_d رمز می گردد. الگوریتم پنهان نگاری داده در هر قالب در بخش ۲-۱ تشریح شد به طوری که (۵) به منظور تعبیه داده در هر $1 \le i \le (m \times l - \hat{\mathbb{I}}_{f_k}(i)$، (1 به کار گرفته می شود. در جایی که n' و $\mathbb{i}$ به ترتیب با n'_k و $\hat{\mathbb{I}}_{f_k}(i)$ جایگزین می شوند.

$\widehat{\mathcal{N}}'_1$ تنها دسته ای از داده سربار است که پنهان نگاری نشده است. بدین دلیل به سراغ پیکسل های مرزی می رویم که همان طور که در شکل ۴ ترسیم شده با $\hat{\mathbb{I}}_b$ نمایش داده شده اند. در بخش بعدی شیوه ای متفاوت به منظور تعبیه $\widehat{\mathcal{N}}'_1$ در $\hat{\mathbb{I}}_b$ تشریح می گردد.

۲-۵- تعبیه داده در پیکسل های مرزی

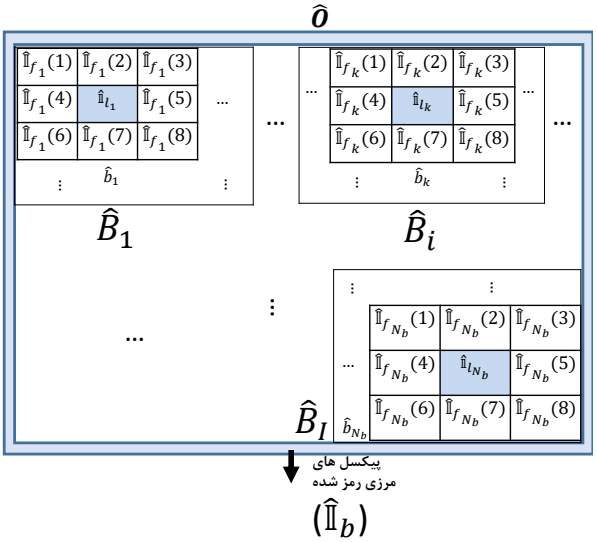
فرایند تعبیه $\widehat{\mathcal{N}}'_1$ در پیکسل های مرزی رمز شده $(\hat{\mathbb{I}}_b)$ در این بخش تشریح می شود. به منظور تعبیه $\widehat{\mathcal{N}}'_1$، فضایی می باید در $\hat{\mathbb{I}}_b$ قبل از رمز نگاری فراهم شود. همان طور که در بخش ۲-۱ مطرح شده ظرفیت ذخیره برای هر پیکسل با تحلیل خطای پیش بینی مشخص می شود. پیش بینی هر $\mathbb{i}_{k'}$ توسط پیکسل قبلی آن $\mathbb{i}_{k'-1}$ خطای پیش بینی را به صورت ذیل می سازد.

$$\mathbb{e}_{k'(k'-1)} = \mathbb{i}_{k'} - \mathbb{i}_{k'-1} \quad (1 < k' \le K') \qquad (۱۶)$$

یک بیت ظرفیت در با بیشترین ارزش ترین بیت $\mathbb{i}_{k'}$ فراهم می شود زمانی که رابطه زیر برای خطای پیش بینی برقرار باشد.

$$\left|\mathbb{e}_{k'(k'-1)}\right| < 64 \qquad (۱۷)$$

بر این اساس، خطای پیش بینی برای تمام پیکسل های مرزی با استفاده از (۱۶) محاسبه و مجموعه $\{\mathbb{e}_{21}, \mathbb{e}_{32}, \dots, \mathbb{e}_{k'(k'-1)}, \dots, \mathbb{e}_{K'(K'-1)}\}$ برای $(K'-1)$ خطا حاصل می شود. با توجه به این واقعیت که پیکسل های مرزی همسایه هستند و تغییرات شدید در پیکسل های مجاور به ندرت اتفاق می افتد، انتظار می رود اکثر خطاهای محاسبه شده طبق رابطه (۱۷) کمتر از ۶۴ باشند هر چند که این رابطه در لبه ها برقرار نیست. بنابراین $(K'-1)$ برچسب به صورت $\{l_2, l_3, \dots, l_{k'}, \dots, l_{K'}\}$، $l_{k'} \in \{0,1,2\}$، به $(K'-1)$ خطا نسبت داده می شود تا مشخص کند که این خطا رابطه (۱۷) را برقرار می کند یا نه. محاسبه هر $l_{k'}$ مطابق با $\mathbb{e}_{k'(k'-1)}$ طبق الگوریتم ۱ تعیین می گردد. در این الگوریتم $\lfloor . \rfloor$ و $|.|$ به ترتیب گرد به سمت عدد صحیح کوچکتر و قدر مطلق را محاسبه می کنند. مجموعه برچسب ها فشرده و سپس به وسیله K_s رمز شده و با $\widehat{\mathcal{L}}$



شکل ۴: دسته بندی بلوک های رمز شده شامل پیکسل های مرزی، $m = l = 3$

الگوریتم ۱: فرایند برچسب زدن برای هر $\mathbb{e}'_{k'(k'-1)}$، $(1 < k' \le K')$

for $k' = 2$ **to** K' **do**
$d_{MSB} = ||\lfloor \mathbb{i}_{k'}/128 \rfloor - \lfloor \mathbb{i}_{k'-1}/128 \rfloor||$
if $((|\mathbb{e}'_{k'(k'-1)}| < 64)$ **then**
$l_{k'} = 0$
else if $((|\mathbb{e}'_{k'(k'-1)}| \ge 64)$ **and** $(d_{MSB} == 0)$ **then**
$l_{k'} = 1$
else if $((|\mathbb{e}'_{k'(k'-1)}| \ge 64)$ **and** $(d_{MSB} == 1)$ **then**
$l_{k'} = 2$
end if
end for

الگوریتم ۲: بازیابی $\mathbb{I}'_{f_k}(i)$ **با توجه به مقادیر** $e'(i)$ **و** $\boldsymbol{n'_k}$ **برای** $(1 \le i < m \times l)$

for $i = 1$ **to** $(m \times l - 1)$ **do**
$e'(i) = \mathbb{I}'_{f_k}(i) - \mathbb{i}_{l_k}$
if $(n'_k == 8)$ **or** $(n'_k == 0)$ **then**
$\mathbb{I}_{f_k}(i) = \mathbb{I}'_{f_k}(i)$
else if $|e'(i)| < 2^{(8-n'_k-1)}$ **then**
$\mathbb{I}_{f_k}(i) = \mathbb{I}'_{f_k}(i)$
else if $e'(i) < 0$
$\mathbb{I}_{f_k}(i) = \mathbb{I}'_{f_k}(i) + 2^{8-n'_k}$
else if $e'(i) > 0$
$\mathbb{I}_{f_k}(i) = \mathbb{I}'_{f_k}(i) - 2^{8-n'_k}$
end if
end for

الگوریتم ۳: بازیابی $\mathbb{i}_{k'}$ **با استفاده از** $[\![\mathbb{i}_{k'}]\!]$، $\mathbb{i}_{k'-1}$، $l_{k'}$ **و** $e'_{k'(k'-1)}$ $(1 < k' \le K')$ **برای**

for $k' = 2$ **to** K' **do**
$e'_{k'(k'-1)} = [\![\mathbb{i}_{k'}]\!] - \mathbb{i}_{k'-1}$
$\mathbb{i}_{k'} = [\![\mathbb{i}_{k'}]\!]$
if $((|e'_{k'(k'-1)}| \ge 64)$ **and** $(l_{k'} == 0))$ **then**
$\mathbb{i}_{k'} = [\![\mathbb{i}_{k'}]\!]\ xor\ 128$
else if $(l_{k'} == 1)$ **then**
$V_1 = [\![\mathbb{i}_{k'}]\!]\ and\ 127$
$V_2 = \mathbb{i}_{k'-1}\ and\ 128$
$\mathbb{i}_{k'} = V_1 + V_2$
else if $(l_{k'} == 2)$ **then**
$V_1 = [\![\mathbb{i}_{k'}]\!]\ and\ 127$
$V_2 = \mathbb{i}_{k'-1}\ and\ 128$
$V_2 = V_2\ xor\ 128$
$\mathbb{i}_{k'} = V_1 + V_2$
end if
end for

نمایش داده می شوند. $\widehat{\mathcal{L}}$ توسط رابطه (۱۸) به $\widehat{\mathcal{N}}'_1$ الصاق شده و $\widehat{\mathcal{LN}}$ ایجاد می شود.

$$\widehat{\mathcal{LN}} = \widehat{\mathcal{L}} \cup \widehat{\mathcal{N}}'_1 \qquad (۱۸)$$

مجموعه $\widehat{\mathcal{LN}}$ در $\hat{\mathbb{I}}_b$ تعبیه می شود و پیکسل های مرزی رمز شده حامل $([\![\hat{\mathbb{I}}_b]\!])$ را می سازد. در این پنهان نگاری $(k'-1)$امین بیت از $\widehat{\mathcal{LN}}$ یعنی $\widehat{\mathcal{LN}}_{k'-1}$ در $\hat{\mathbb{i}}_{k'}$ طبق رابطه (۱۹) به منظور ایجاد پیکسل حامل $[\![\hat{\mathbb{i}}_{k'}]\!]$ پنهان نگاری می گردد.

$$[\![\hat{\mathbb{i}}_{k'}]\!] = 128 \times \widehat{\mathcal{LN}}_{k'-1} + (\hat{\mathbb{i}}_{k'}\ mod\ 128),\ 2 < k' \le K' \qquad (۱۹)$$

در این صورت مجموعه $[\![\hat{\mathbb{I}}_b]\!] = \{\hat{\mathbb{i}}_1, [\![\hat{\mathbb{i}}_2]\!], [\![\hat{\mathbb{i}}_3]\!], \dots, [\![\hat{\mathbb{i}}_{k'}]\!], \dots, [\![\hat{\mathbb{i}}_{K'}]\!]\}$ نمایشی از پیکسل های مرزی رمز شده حامل است. پیکسل اول $(\hat{\mathbb{i}}_1)$ بدون تغییر باقی می ماند و به منظور بازیابی پیکسل های دیگر در گیرنده به کار گرفته می شود.

۲-۶- استخراج داده و بازیابی تصویر اصلی

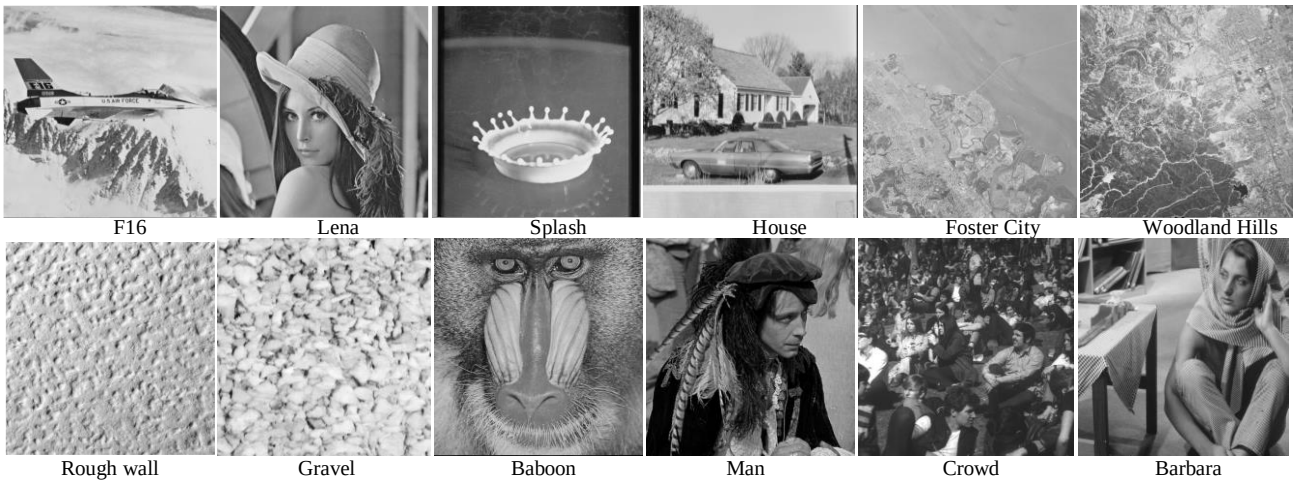
استخراج داده و بازیابی تصویر اصلی در این بخش تشریح می گردد. در گیرنده در ابتدا n'_k ها بازیابی می شوند. نحوه بازیابی آنها در زیربخش بعد توضیح داده می شود. در اینجا با فرض در اختیار داشتن n'_k ها به روند استخراج داده و بازیابی تصویر اصلی می پردازیم.

با داشتن $\boldsymbol{n'_k}$ ها، بیت های داده از هر پیکسل مجاور $[\![\hat{\mathbb{I}}_{f_k}(i)]\!]$، $1 \le i < m \times l$ از $[\![\hat{b}_k]\!]$ با توجه به رابطه زیر استخراج می شود.

$$D_i^k = \frac{\sum_{i'=1}^{n'_k} (2^{8-i'} \times [\![\mathbb{I}_{f_k}(i)]\!]_{8-i'})}{2^{8-n'_k}},\ 1 \le i < m \times l,\ 1 \le k \le N_b \qquad (۲۰)$$

در جایی که $[\![\hat{P}_{f_k}(i)]\!]_a$، a امین بیت از پیکسل $[\![\hat{P}_{f_k}(i)]\!]$ برای $0 \le a < 8$ است. اگر $\boldsymbol{n'_k} \ne 0$ باشد، D_i^k داده استخراج شده در قالب k ام و از پیکسل مجاور i ام است. با کنار هم قرار دادن D_i^k ها، D استخراج می شود. D شامل داده مفید رمز شده و داده سربار است. پنهان کننده توسط K_d می تواند داده مفید را رمز گشایی کند. در اینجا مقادیر بازیابی شده به صورت بولد[۱۴] نمایش داده می شوند. برای مثال $\boldsymbol{n'_k}$ بازیابی شده n'_k است.

در فرایند بازیابی تصویر اصلی، هر $[\![\hat{b}_k]\!]$ باید به مقدار اصلی خود b_k تبدیل گردد. بنابراین $[\![\hat{b}_k]\!]$ توسط K_c رمزگشایی شده و نتیجه این رمزگشایی قالب $[\![\boldsymbol{b_k}]\!]$ است که شامل $[\![\mathbb{I}_{\boldsymbol{f_k}}]\!]$ و $\mathbb{i}_{\boldsymbol{l_k}}$ می باشد. بازیابی i امین عضو از مجموعه $[\![\mathbb{I}_{\boldsymbol{f_k}}]\!]$ یعنی $([\![\mathbb{I}_{\boldsymbol{f_k}}(i)]\!])$ در دو مرحله انجام می شود. در مرحله اول $\mathbb{I}'_{f_k}(i)$ توسط رابطه (۲۱) بدست می آید.



شکل ۵: تصاویر تست

جدول ۱: ظرفیت ذخیره فراهم شده توسط الگوریتم پیشنهادی برای تصاویر تست در اندازه بلوک های مختلف

اندازه بلوک		تصاویر										
		Barbara	Crowd	Man	Baboon	Gravel	Rough wall	Woodland Hills	Foster City	House	Splash	Lena	F16
2×2	کل ظرفیت ذخیره (بیت)	۵۸۵۶۱۸	۷۶۷۶۵۲	۶۱۶۰۶۲	۴۱۳۴۸۷	۶۱۶۶۶۵	۴۸۵۱۴۵	۴۸۰۶۴۸	۶۶۷۶۳۵	۷۱۲۳۲۶	۸۳۱۹۹۹	۷۲۱۸۷۵	۷۹۳۱۲۸
	داده مفید (بیت)	۳۹۰۵۴۰	۵۷۲۵۸۰	۴۲۰۹۹۰	۲۱۸۴۱۰	۴۲۱۵۹۰	۲۹۰۰۷۰	۲۸۵۵۷۰	۴۷۲۵۶۰	۵۱۷۲۵۰	۶۳۶۹۲۰	۵۲۶۸۰۰	۵۹۸۰۵۰
	داده سربار (بیت)	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰	۱۹۵۰۸۰
	داده مفید (bpp)	۱/۴۹	۲/۱۸	۱/۶۱	۰/۸۳	۱/۶۱	۱/۱۱	۱/۰۹	۱/۸	۱/۹۷	۲/۴۳	۲/۰۱	۲/۲۸
	مجموع $\mathcal{L}$ (بیت)	۲۱	۸۸	۶۷	۱۳۸۰	۸۱	۱۴۷	۸۹	۲۱	۵۳۱	۱۲۱	۲۱	۷۹
3×3	کل ظرفیت ذخیره (بیت)	۵۷۳۹۸۰	۷۴۲۹۶۰	۵۹۶۲۸۰	۳۵۶۰۳۰	۵۹۸۲۹۰	۴۴۱۹۴۰	۴۲۱۰۱۰	۶۷۱۴۱۰	۷۰۴۶۸۰	۸۵۸۱۳۰	۷۳۳۰۰۰	۸۰۷۴۲۰
	داده مفید (بیت)	۴۸۷۲۸۰	۶۵۶۲۶۰	۵۰۹۵۸۰	۲۶۹۳۳۰	۵۱۱۵۹۰	۳۵۵۲۴۰	۳۳۴۳۱۰	۵۸۴۷۱۰	۶۱۷۹۸۰	۷۷۱۴۳۰	۶۴۶۳۰۰	۷۲۰۷۲۰
	داده سربار (بیت)	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳	۸۶۷۰۳
	داده مفید (bpp)	۱/۹۷	۲/۵۶	۲/۰۳	۱/۱۶	۲/۰۷	۱/۴۸	۱/۳۹	۲/۳۶	۲/۴۸	۳/۰۴	۲/۵۷	۲/۸۶
	مجموع $\mathcal{L}$ (بیت)	۴۹	۱۸۱	۴۸	۱۰۲۸	۴۹	۱۱۱	۱۳۶	۹۱	۵۱۴	۱۱۱	۲۱	۷۰
4×4	کل ظرفیت ذخیره (بیت)	۵۱۸۳۹۰	۶۵۲۴۰۰	۵۱۴۱۹۰	۲۹۸۳۵۰	۵۲۶۹۵۰	۳۶۴۱۹۰	۳۴۶۱۹۰	۶۳۷۹۲۰	۶۴۰۰۱۰	۸۰۴۸۱۰	۶۷۸۲۹۰	۷۴۱۴۷۰
	داده مفید (بیت)	۴۶۹۶۱۰	۶۰۳۶۲۰	۴۶۵۴۱۰	۲۴۹۵۸۰	۴۷۸۱۸۰	۳۱۵۴۱۰	۲۹۷۴۱۰	۵۸۹۱۵۰	۵۹۱۲۳۰	۷۵۶۰۴۰	۶۲۹۵۱۰	۶۹۲۶۹۰
	داده سربار (بیت)	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱	۴۸۷۷۱
	داده مفید (bpp)	۱/۷۹	۲/۳	۱/۷۸	۰/۹۵	۱/۸۲	۱/۲	۱/۱۳	۲/۲۵	۲/۲۶	۲/۸۸	۲/۴	۲/۶۴
	مجموع $\mathcal{L}$ (بیت)	۲۱	۸۸	۶۷	۱۳۸۰	۸۱	۱۴۷	۸۹	۲۱	۵۳۱	۱۲۱	۲۱	۷۹

$$\mathbb{I}'_{f_k}(i)=\sum_{i'=1}^{8-\mathbf{n}'_k}(2^{i'-1}\times[\![\mathbb{I}_{\boldsymbol{f_k}}(\boldsymbol{i})]\!]_{i'-1})+\sum_{i'=9-\mathbf{n}'_k}^{8}(2^{i'-1}\times(\hat{\mathbb{i}}_{\boldsymbol{l_k}})_{i'-1}).\quad 1\le i<m\times l,1\le k\le N_b \qquad (۲۱)$$

و در مرحله دوم، $\mathbb{I}_{f_k}(i)$ توسط الگوریتم ۲ بازیابی می شود. در (۲۱)، $\mathbf{n}'_k$ بیت از $\boldsymbol{p}_{\boldsymbol{l_k}}$ که ارزش بیشتری دارند در بیت های متناظر از $[\![\mathbb{I}_{\boldsymbol{f_k}}(\boldsymbol{i})]\!]$ به منظور ایجاد $\mathbb{I}'_{f_k}(i)$ جایگزین می شوند. فرایند بازیابی برای همه قالب های رمز شده حامل بر طبق روش اشاره شده به منظور بازیابی تصویر اصلی انجام می گردد.

۲-۷- بازیابی ویژگی ظرفیت ذخیره هر قالب

در بخش قبل استخراج داده از قالب های رمز شده حامل و بازیابی تصویر اصلی با استفاده از ویژگی ظرفیت ذخیره هر قالب تشریح گردید. در اینجا روش بازیابی این ویژگی ها را توضیح می دهیم. بر اساس در مرحله اول به منظور بازیابی $\mathbb{N}'$، باید $\widehat{\mathcal{LN}}$ از $[\![\hat{\mathbb{I}}_b]\!]$ بازیابی شود. با توجه به این واقعیت که $\widehat{\mathcal{LN}}$ در بیت هایی با ارزش بالاتر از $[\![\hat{\mathbb{I}}_b]\!]$ تعبیه شده است، $\widehat{\mathcal{LN}}$ برای هر $[\![\hat{\mathbb{i}}_{k'}]\!]$ توسط رابطه زیر بازیابی می شود.

$$\widehat{\boldsymbol{\mathcal{LN}}}_{\boldsymbol{k'-1}}=\lfloor[\![\hat{\mathbb{i}}_{k'}]\!]/128\rfloor \quad 1<k'\le K' \qquad (۲۲)$$

در جایی که $(k'-1)$ امین بیت از $\widehat{\mathcal{LN}}$ یعنی $\widehat{\boldsymbol{\mathcal{LN}}}_{\boldsymbol{k'-1}}$ از $[\![\hat{\mathbb{i}}_{k'}]\!]$ استخراج می شود. تعداد بیت های استخراج شده $K'-1$ است. $\widehat{\boldsymbol{\mathcal{LN}}}$ حاصل اتصال $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{1}}$ و $\widehat{\boldsymbol{\mathcal{L}}}$ است. با توجه به تعداد بیت های $\widehat{\boldsymbol{\mathcal{L}}}$ که در ابتدای آن ذخیره شده است می توان $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{1}}$ و $\widehat{\boldsymbol{\mathcal{L}}}$ را تفکیک کرد. در نتیجه $\widehat{\boldsymbol{\mathcal{L}}}$ تفکیک و به منظور بازیابی برچسب ها رمز گشایی و نافشرده می شود. با استفاده از این برچسب ها پیکسل های مرزی احیا می گردند. همچنین $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{1}}$ رمز گشایی و به منظوز ایجاد $\mathbb{N}'_1$ نافشرده می شود. $\mathbb{N}'_1$ به منظور استخراج داده از $[\![\hat{B}_1]\!]$ شامل داده مفید و $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{2}}$ به کار گرفته می شود. به طور مشابه $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{2}}$ به منظور استخراج داده از $[\![\hat{B}_2]\!]$ که شامل داده مفید و $\widehat{\boldsymbol{\mathcal{N}}}'_{\boldsymbol{3}}$ است به کار گرفته می شود و به همین ترتیب تمام داده ها از تصویر رمز شده استخراج می گردد.

آنچه باقی می ماند تنها بازسازی پیکسل های مرزی است. فرایند بازسازی این پیکسل ها در واقع بازیابی ارزشمندترین بیت هر پیکسل است.

اگر مجموعه $[\![\mathbb{I}_b]\!]=\{\hat{\mathbb{i}}_1,[\![\hat{\mathbb{i}}_2]\!],[\![\hat{\mathbb{i}}_3]\!],\dots,[\![\hat{\mathbb{i}}_{k'}]\!],\dots,[\![\hat{\mathbb{i}}_{K'}]\!]\}$ را به عنوان پیکسل های مرزی بعد از رمز گشایی در نظر بگیریم به دلیل اینکه پیکسل اول از $\mathbb{I}_b$ یعنی $\hat{\mathbb{i}}_1$ بدون تغییر باقی مانده است به وسیله آن می توان پیکسل $[\![\hat{\mathbb{i}}_2]\!]$ را بازیابی کرد و به طور کلی $[\![\hat{\mathbb{i}}_{k'}]\!]$ با استفاده از $\hat{\mathbb{i}}_{k'-1}$ و الگوریتم ۳ بازیابی می شود. در این الگوریتم عمگرهای "xor" و "and" بیتی اعمال می شوند و $\{\boldsymbol{l}_2,\boldsymbol{l}_3,\dots,\boldsymbol{l}_{k'},\dots,\boldsymbol{l}_{K'}\}\in(\mathbf{0},\mathbf{1},\mathbf{2})$ با به کار گیری رابطه (۲۲) از قبل در دسترس است.

۳- نتایج آزمایش

چندین آزمایش به منظور تشریح کارایی الگوریتم پیشنهادی به خصوص از نظر افزایش ظرفیت ذخیره داده در نظر گرفته می شود. در این آزمایش ۹ تصویر در مقیاس خاکستری F16، Lena، Splash، House، Foster City، Woodland Hills، Rough wall، Gravel، Baboon از پایگاه داده SIPI و سه تصویر Man، Crowd، Barbara از پایگاه داده Miscelaneous (شکل ۵) و ۱۰۰۰۰ تصویر از پایگاه داده BOWS2 در نظر گرفته می شود. تمام تصاویر دارای اندازه ۵۱۲×۵۱۲ هستند.

ظرفیت کلی فراهم شده، تعداد بیت های داده مفید و سربار و داده مربوط به مجموعه $\mathcal{L}$ در جدول ۱ برای سه اندازه قالب 2×2، 3×3 و 4×4 لیست شده است. ظرفیت کلی فراهم شده مجموع داده مفید و سربار است. همچنین تعداد بیت های داده مفید به تعداد کل پیکسل های تصویر محاسبه و در جدول با عنوان (bpp) مشخص شده است.

همان طور که دیده می شود با افزایش اندازه قالب ها کاهش قابل توجه در اندازه داده سربار محقق می شود. اگرچه تغییرات در ظرفیت کلی فراهم شده برای بیشتر تصاویر تست ناچیز است. به بیان دیگر، استفاده از اندازه قالب های 3×3 و 4×4 داده سربار کمتری نسبت به اندازه قالب 2×2 فراهم می کند در حالی که برای بیشتر تصاویر تست تغییرات قابل توجه ای در تعداد بیت های داده مفید اتفاق نیفتاده است. بنابراین استفاده از اندازه قالب 3×3 نسبت به 2×2 برای همه تصاویر تست بهبود بیشتری فراهم آورده است. همچنین استفاده از اندازه قالب 4×4 نسبت به 2×2 برای بیشتر تصاویر تست به خصوص تصاویر نرم تر ظرفیت مفید بیشتری را ایجاد کرده است. برای تمام تصاویر، استفاده از اندازه قالب 3×3 در مقایسه با 4×4 ظرفیت نهان نگاری را افزایش داده است به این دلیل که تقارن بیشتری بین پیکسل مبنا و پیکسل های مجاور در یک قالب 3×3 نسبت به 4×4 وجود دارد. این تقارن پیش بینی بهتری را از پیکسل های مجاور توسط پیکسل مبنا موجب می شود. تصاویر نرم تر نظیر F16 و Splash به دلیل همین پیش بینی بهتر به مراتب ظرفیت ذخیره بیشتری را نسبت به Baboon، Woodland Hills و Rough Wall برای تمام اندازه قالب ها فراهم می آورند.

برای اندازه قالب های 2×2 و یا 4×4، سطر اول تصویر به عنوان پیکسل های مرزی در نظر گرفته می شوند. این پیکسل ها حداکثر ۲۰۴۸ بیت ظرفیت فراهم می کنند. از طرف دیگر برای اندازه قالب 3×3، دو سطر و یا ستون اول تصویر به عنوان پیکسل های مرزی در نظر گرفته می شود که معادل ۲۰۴۴ بیت ظرفیت فراهم شده است. این پیکسل ها توسط الگوریتم ۱ برچسب زده می شوند. این برچسب ها توسط کدبندی حسابی فشرده و همان طور که گفته شد مجموعه $\mathcal{L}$ را ایجاد می کنند. تعداد بیت های تشکیل دهنده $\mathcal{L}$ در

جدول ۱ نمایش داده شده است. پیکسل های مرزی در Baboon و House ناهموارتر[۱۵] از بقیه تصاویر هستند بنابراین برای این تصاویر تعداد بیت های بیشتری به $\mathcal{L}$ اختصاص می یابد.

در جدول ۲، کارایی الگوریتم پیشنهادی برای همه اندازه قالب ها برای ۱۰۰۰۰ تصویر تست از پایگاه داده Bows2 بررسی شده است. برای اندازه قالب های 2×2، 3×3 و 4×4 به ترتیب ۷۶۸، ۶۸۰ و ۵۱۲ قالب به عنوان مجموعه ای از قالب هایی که $\hat{B}_i$ را شکل می دهند (شکل ۴) به کار گرفته شده است. اگرچه برای تصاویر خیلی ناهموار تعداد این قالب ها می تواند نصف نیز شود که به دلیل افزایش تعداد بیت ها در مجموعه $\mathcal{L}$ و در پی آن کاهش ظرفیت ذخیره در پیکسل های مرزی است.

برای اندازه قالب های 3×3 و 4×4، تصاویر دارای شماره ۱۴۷۸ و ۹۴۴۸ از پایگاه داده Bows2 به ترتیب بهترین و بدترین ظرفیت داده مفید را فراهم می کنند. برای 2×2، بهترین و بدترین به ترتیب شماره های ۱۴۷۸ و ۵۵۴۷ هستند. این سه تصویر در شکل ۶ نمایش داده شده اند. همان طور که در جدول ۲ تشریح شده است، استفاده از اندازه قالب 4×4 نسبت به اندازه های دیگر برای بهترین تصویر (شماره ۱۴۷۸) ظرفیت بهتری را فراهم می آورد. اگر چه برای بدترین تصاویر استفاده از اندازه 2×2 نتایج بهتری را نسبت به اندازه قالب های دیگر به دست می دهد. به طور میانگین استفاده از اندازه قالب 3×3 بهبودی را به ترتیب بیشتر از ۰/۴ bpp و ۰/۱ bpp نسبت به اندازه قالب های 2×2 و 4×4 موجب می شود. از منظر پیچیدگی محاسباتی، استفاده از اندازه قالب های کوچکتر تعداد قالب ها را افزایش و در پی آن تعداد ویژگی ها، داده سربار و پیچیدگی محاسباتی را بیشتر می کند.

در جدول ۳ طرح پیشنهادی با دو طرح ظرفیت بالا [۱۷] و [۱۸] برای دو اندازه قالب متفاوت مقایسه شده است. شبیه سازی طرح [۱۸] برای بهترین α و β و اندازه قالب 3×3 انجام شده است. در این طرح α و β دو پارامتر مهم تاثیر گذار در ظرفیت فراهم شده است که دارای حدود $B \le 7$ و $\alpha \le 1$ می باشد. همچنین تاثیر مدولاسیون پیکسل ها که تاثیر اندکی در کاهش ظرفیت ذخیره فراهم شده دارد در پیاده سازی طرح [۱۸] در نظر گرفته شده است. این مدولاسیون با توجه به ماهیت تصادفی آن می تواند تاثیر متفاوتی بر ظرفیت فراهم شده در هر بار اجرای الگوریتم داشته باشد.

ظرفیت ذخیره برای یکی از دو روش پنهان نگاری معکوس پذیر در تصویر رمز مطرح شده در [۱۷] که بازیابی بدون اتلاف تصویر اصلی را تضمین می کند در جدول ۳ تشریح گردیده است. در این روش به غیر از Barbara و Baboon بقیه تصاویر ظرفیتی نزدیک به bpp ۱ نشان می دهند در حالی که در روش پیشنهادی برای تمام تصاویر تست ظرفیتی بیشتر از bpp ۱ برای هر دو اندازه قالب فراهم می شود.

طرح پیشنهادی و [۱۸] ظرفیت ذخیره بهتری را برای تصاویر نرم تر نظیر F16 و Splash و ظرفیت کمتری را برای تصاویر ناهموارتر نظیر Baboon، Rough wall و Woodland Hills فراهم می کنند. برای تمام تصاویر تست و هر دو اندازه قالب روش پیشنهادی ظرفیت ذخیره بهتری از روش مطرح در [۱۸] ایجاد می کند. این بهبود برای تصویر House حتی بیشتر از bpp ۰/۷۷ است. به علاوه در طرح پیشنهادی هیچ پیکسل و یا ویژگی از تصویر اصلی باقی نمانده است که رمز نشده باشد در صورتی که در [۱۸] تفاوت بین برخی پیکسل ها رمز نشده باقی می ماند.

۴- نتیجه

در این مقاله، روش پنهان نگاری معکوس پذیر در تصویر رمز شده معرفی شده است که می تواند ظرفیت بالایی را به منظور ذخیره داده فراهم کند. این امر با بهره گیری از همبستگی پیکسل های مجاور در یک قالب از تصویر و پیش بینی محلی محقق می شود. در این پیش

جدول ۲: تحلیل کارایی الگوریتم پیشنهادی با به کارگیری پایگاه داده BOWS2 که شامل ۱۰۰۰۰ تصویر تست است

اندازه قالب	2×2			3×3			4×4		
	بهترین تصویر	بدترین تصویر	میانگین	بهترین تصویر	بدترین تصویر	میانگین	بهترین تصویر	بدترین تصویر	میانگین
کل ظرفیت ذخیره (بیت)	۱۴۳۵۲۶۶	۲۶۷۷۹۵	۸۲۲۸۰۸	۱۶۷۰۰۴۸	۱۷۵۸۸۰	۸۳۶۹۸۷	۱۷۳۳۴۷۵	۹۳۹۹۰	۷۸۲۲۱۰
داده مفید (بیت)	۱۳۹۲۷۷۰	۱۱۱۹۷۹	۶۷۴۵۹۶	۱۶۵۲۳۶۰	۱۰۸۲۲۴	۷۸۰۶۹۱	۱۷۲۲۷۷۱	۵۹۲۸۶	۷۵۰۹۵۲
داده سربار (بیت)	۴۲۴۹۶	۱۵۵۸۱۶	۱۴۸۲۱۲	۱۷۶۸۸	۶۷۶۵۶	۵۶۲۹۶	۱۰۷۰۴	۳۴۷۰۴	۳۱۲۵۸
داده مفید (bpp)	۵/۳۱	۰/۴۳	۲/۵۷	۶/۳۰	۰/۴۱	۲/۹۸	۶/۵۷۲	۰/۲۲۶	۲/۸۷
مجموع $\mathcal{L}$ (بیت)	۲۱	۷۱۸	۸۵	۲۱	۱۴۵۴	۱۲۳	۲۱	۱۶۳۹	۸۵
شماره تصویر	۱۴۷۸	۵۵۴۷	-	۱۴۷۸	۹۴۴۸	-	۱۴۷۸	۹۴۴۸	-

جدول ۳: مقایسه ظرفیت ذخیره فراهم شده در الگوریتم پیشنهادی برای تصاویر تست با دو طرح جدید مطرح در پنهان نگاری معکوس پذیر میزبان رمز

تصاویر		F16	Lena	Splash	House	Foster City	Woodland Hills	Rough wall	Gravel	Baboon	Man	Crowd	Barbara	
[۱۷]		۰/۹۸	۰/۹۸	۰/۹۹	۰/۹۴	۰/۹۹	۰/۹۷	۰/۹۹	۰/۹۹	۰/۷۵	۰/۹۲	۰/۹۸	۰/۷۶	bpp
[۱۸]		۲/۲۱	۲/۰۲	۲/۶۶	۱/۵۵	۱/۶	۰/۹۹	۱/۲۶	۱/۶۵	۰/۷۵	۱/۴۶	۱/۷۵	۱/۲۹	bpp
طرح	3×3	۲/۴۶	۲/۲۴	۲/۷۲	۲/۰۸	۲/۰۹	۰/۹۶	۱/۰۳	۱/۶۵	۰/۷۶	۱/۶	۲/۱۳	۱/۶۲	bpp

بینی تفاوت بین پیکسل های مجاور و پیکسل مبنا به منظور دستیابی به خطای پیش بینی محاسبه می شود. با ارزیابی این خطاها ویژگی ظرفیت ذخیره هر قالب به دست آمده و از آنها به منظور ایجاد فضا برای پنهان نگاری قبل از رمز نگاری تصویر بهره می بریم. این ویژگی ها فشرده و بعد رمز و به عنوان داده سربار به همراه داده مفید در تصویر رمز شده تعبیه می شوند. این تعبیه خود با بهره گیری از ویژگی ظرفیت ذخیره هر قالب و به شیوه سلسله مراتبی محقق می شود. در گیرنده در ابتدا داده سربار استخراج و در ادامه با رمزگشایی و نافشرده سازی ویژگی ظرفیت ذخیره هر قالب بازیابی می شود. سپس با استفاده از این ویژگی ها تصویر اصلی به طور کامل بازیابی و داده مفید بدون خطا استخراج می شود.

طرح پیشنهادی بهبود محسوسی در ظرفیت ذخیره داده نسبت به روش های موجود فراهم می کند. همچنین در روش ما به شرط عدم تغییر چیدمان پیکسل ها و بیت ها در هر قالب می توان از هر روش رمز نگاری و یا هر الگوریتم رمز نگاری بهره برد.

مراجع

[1] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210-3237, 2016.

[2] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

[3] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, Mar. 2006.

[4] T. Kalker, and F. M. Willems, "Capacity bounds and constructions for reversible data-hiding," in *Proc. International Conference on Digital Signal Processing* , Santorini, Greece, Greece, 2002, pp. 71-76.

[5] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999, Jul. 2009.

[6] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1091-1100, Jul. 2013.

[7] C.-H. Yang, and M.-H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Processing*, vol. 4, no. 4, pp. 223-234, Aug. 2010.

[8] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524-3533, Dec. 2011.

[9] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129-1143, Jun. 2009.

[10] S. Xiang, and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3099-3110, Nov. 2018.

[11] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553-562, Mar. 2013.

[12] Z. Yin, Y. Xiang, and X. Zhang, "Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding," *IEEE Transactions on Multimedia*, Aug. 2019.

[13] Y.-C. Chen, T.-H. Hung, S.-H. Hsieh, and C.-W. Shiu, "A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3332-3343, Dec. 2019.

[14] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public-key cryptography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1622-1631, Sep. 2016.

[15] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132-1143, May, May 2016.

[16] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226-233, Nov. 2015.

[17] P. Puteaux, and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1670-1681, Jul. 2018.

[18] S. Yi, and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 51-64, Jan. 2019.

[19] D. Xu, and R. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Processing*, vol. 123, pp. 9-21, Jun. 2016.

[20] Z. Yin, B. Luo, and W. Hong, "Separable and error-free reversible data hiding in encrypted image with high payload," *The Scientific World Journal*, vol. 2014, Apr. 2014.

[21] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118-127, Jan. 2014.

[22] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777-2789, Dec. 2016.

[23] H. Ge, Y. Chen, Z. Qian, and J. Wang, "A high capacity multi-level approach for reversible data hiding in encrypted images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2285 - 2295, Aug. 2019.

[24] X. Zhang, "Commutative reversible data hiding and encryption," *Security and Communication Networks*, vol. 6, no. 11, pp. 1396-1403, 2013.

[25] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 826-832, Apr. 2012.

[26] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441-452, Mar. 2016.

[27] Z. Qian, and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636-646, Apr. 2016.

[28] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255-258, Apr. 2011.
[29] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199-202, Apr. 2012.
[30] X. Wu, and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," *Signal Processing*, vol. 104, pp. 387-400, Nov. 2014.

زیرنویس‌ها

[1] Block
[2] Uncompressed
[3] Reversible data hiding
[4] Lossless
[5] Difference expansion
[6] Histogram modification
[7] Prediction error
[8] Sharper
[9] Cloud computing
[10] LSB
[11] Patch-level sparse
[12] MSB
[13] Median edge detector
[14] Bold
[15] Rougher