MINUT 2021 (1), s. 98–105
ISSN 2719-3063

Ewelina JAMROZIK[1]

[1]Katedra Informatyki i Automatyki, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Aleja Powstańców Warszawy 12, 35-959 Rzeszów

Przykład zastosowania steganografii dla dwuwymiarowych obrazów cyfrowych

Streszczenie. W artykule przedstawiono wykorzystanie klasycznego algorytmu steganograficznego opierającego się na metodzie najmniej znaczących bitów (ang. *Least Significant Bit*) w szyfrowaniu obrazów, która polega na ukrywaniu informacji w najmniej znaczących bitach dla każdego z trzech kolorów modelu RGB opisujących dany piksel. Wyjaśniony został sposób reprezentacji obrazu oraz jego przetwarzania. W ocenie wydajności i jakości procesu posłużono się histogramami, wskaźnikiem jakości obrazu PSNR (ang. *peak signal-to-noise ratio*) i miarą podobieństwa strukturalnego SSIM (ang. *structural similarity index measure*) pomiędzy obrazami na różnych etapach. Wykazane zostały zmiany, które zaszły w obrazach na drodze procesu steganograficznego, a mogą być informacją dla osób wykrywających takie działania. Oceniony został stopień zachowalności i utraty danych. Na skutek przeprowadzonego procesu jakość obrazów uległa nieznacznemu pogorszeniu, co wykazały obliczone parametry.
Słowa kluczowe: szyfrowanie, SSIM, PSNR, przetwarzanie obrazu, Python.

1. Wstęp

Bezpieczeństwo staje się ważną częścią współczesnego systemu komunikacji. Pojawiła się szczególna potrzeba zachowania poufności i integralności danych oraz zabezpieczenia ich przed nieautoryzowanym dostępem. Dziedzinami, które umożliwiają utajnienie przekazu są kryptografia oraz steganografia. Pierwsza z nich zajmuje się ochroną informacji poprzez zaszyfrowanie jej treści, natomiast druga przez zamaskowanie faktu jej istnienia ukrywając ją w innej [2].

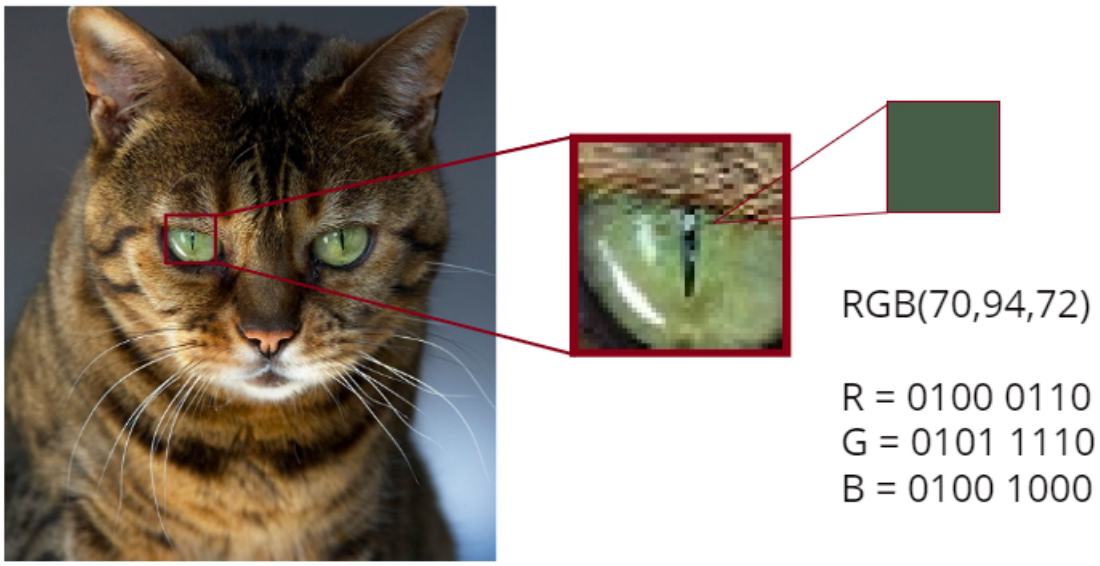
Celem steganografii jest przechowanie ważnych informacji w taki sposób, aby osoba postronna nie była w stanie wykryć ich istnienia. Nośnikami danych są najczęściej pliki graficzne (BMP, JPEG, GIF itp.), dźwiękowe (MP3, WAVE), video (AVI), pliki HTML, a nawet nagłówki pakietów protokołów komunikacyjnych, czy też ich zależności czasowe (uzyskanie określonych opóźnień w reakcji na ustalone zdarzenie, celowo opóźnione pakiety). W obiektach steganograficznych jako dodatkowe zabezpieczenie ukrywanych informacji zaczęto stosować kryptografię, co istotnie wpłynęło na oferowany przez nie poziom zabezpieczenia [2].

Autor korespondencyjny: E. Jamrozik (jamrozik.ewelina@gmail.com).
Data wpłynięcia: 14.08.2020.

Można zauważyć, że bardzo często wykorzystywanymi nośnikami są pliki graficzne. Wiąże się to z dużą pojemnością danych, które można w nich umieścić bez wzbudzania podejrzeń. Sama obecność plików graficznych na stronach internetowych czy też w przesyłanych wiadomościach nie jest zaskakująca [2]. Jest to ponadto interesujący obiekt eksperymentów pod kątem przetwarzania obrazów. Opiera się na traktowaniu ich jak wszystkie inne sygnały i analizowaniu w analogiczny sposób.

2. Materiały, metody i wyniki

U podstaw steganografii obrazu leży zrozumienie koncepcji obrazu cyfrowego jako dwuwymiarowej macierzy pikseli. Poszczególne piksele opisują zawartość obrazu poprzez przechowywanie wartości reprezentujących jasność danego koloru w określonym punkcie. Dla modelu przestrzeni barw RGB (przyjętego w tym artykule) są to 3 kolory – czerwony, zielony, niebieski. Ich wartości zmieniają się w zakresie [0,255], więc można je zakodować używając 1 bajtu, czyli 8 bitów. Reprezentacja w systemie binarnym pozwala na wyodrębnienie bitów bardziej i mniej znaczących (skrajny lewy bit jest najbardziej znaczącym, natomiast prawy najmniej). Informacja ta może być wykorzystana przy dokonywaniu zmian w obrazach, ponieważ będą one proporcjonalnie widoczne do tego, jak znaczący bit wartości dla kanału wybranego piksela zostanie zmieniony. Na rysunku 1 przedstawiono reprezentację przykładowego piksela dla obrazu pochodzącego ze zbioru *Open Images Dataset* [4].



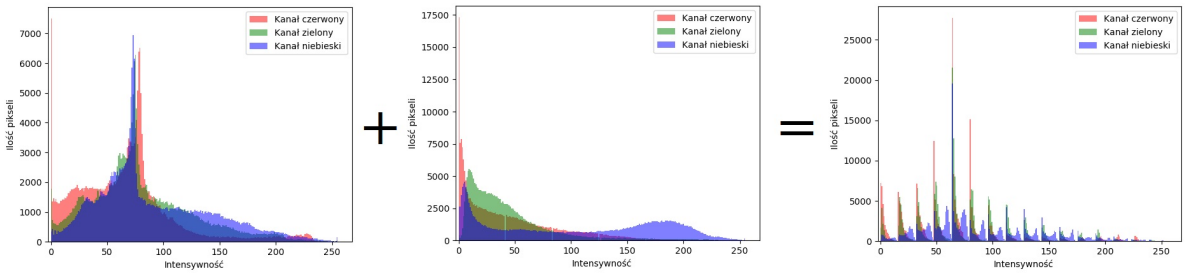
Rysunek 1. Reprezentacja obrazu

Jedna z najpopularniejszych metod steganograficznych stosowanych również dla obrazów – LSB (*Least Significant Bit*), opiera się na jego reprezentacji binarnej i wykorzystaniu najbardziej i najmniej znaczących bitów. Proces ukrywania obrazu w innym rozpoczyna się od pobrania dwóch obrazów – bazowego oraz tego, który ma zostać ukryty. Następnym krokiem jest dokonanie na nich konwersji do tablicy wartości pikseli. Można założyć, że $[i][j][k]$ będzie wartością piksela w miejscu (i, j) dla kanału k (barwy), gdzie i zmienia się od 1 do szerokości obrazu, j od 1 do wysokości obrazu, natomiast k od 0 do 2. W ten sposób dla obrazu bazowego tworzy się trójwymiarową tablicę *img1*, a dla obrazu ukrywanego analogicznie

img2, przy czym wartości zapisane są w systemie binarnym. Reprezentacje te są łączone w tablicę *img3*, reprezentującą jeden wyjściowy obraz steganograficzny. Realizowane jest to poprzez połączenie ze sobą 4 bitów najbardziej znaczących obrazu bazowego oraz ukrywanego. Obraz wyjściowy w miejsce 4 bitów najbardziej znaczących otrzymuje bity od obrazu bazowego, natomiast w miejsce 4 bitów najmniej znaczących te od obrazu ukrywanego. Pozwala to na zachowanie podobieństwa pomiędzy obrazem bazowym a obrazem steganograficznym, a także późniejsze uzyskanie zadowalającej jakości obrazu otrzymanego na drodze ekstrakcji ukrytego. Operacja ta jest wykonywana dla 3 kanałów każdego piksela, co zostało przedstawione na przykładzie na rysunku 2. Z uwagi na to ważne jest aby obraz, który zostanie ukryty miał rozmiar co najwyżej taki sam jak obraz bazowy [1]. W przedstawionym w artykule eksperymencie użyto obrazu *Melly* (klasa Cat) oraz *14.3.11* (klasa Flower) ze zbioru *Open Images Dataset*, przeskalowanych do tego samego rozmiaru – 500x500 pikseli [4].

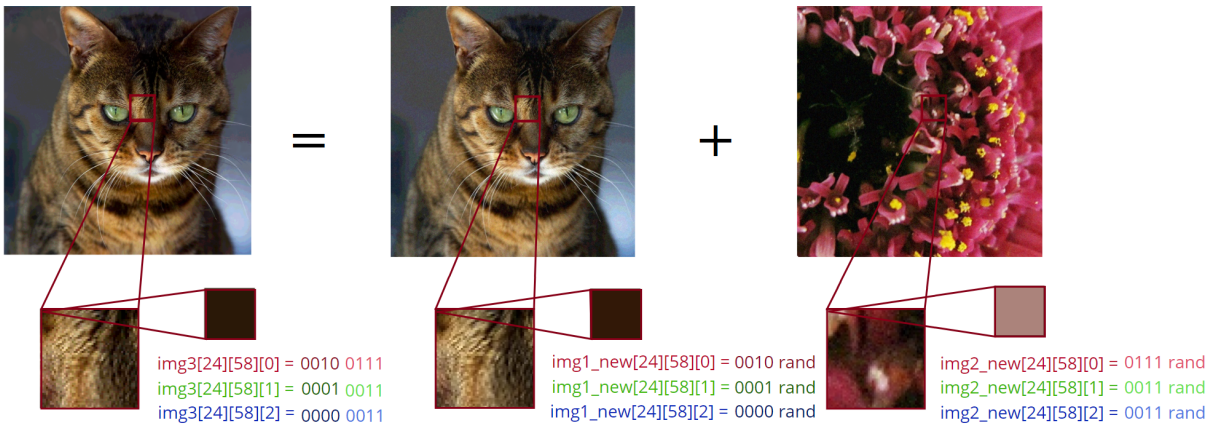
Rysunek 2. Proces ukrywania obrazu w innym

Na drodze konwersji w obraz steganograficzny (wyjściowy) dochodzi do utraty jakości. Wynika to z operacji wykonanych na obrazie – szumu utworzonego przez obraz ukryty. Wizualnie obraz drugi wydaje się jednak dobrze zamaskowany, choć na wyjściowym widoczny jest wpływ barw dominujących w obrazie ukrytym. Szczególnie wrażliwe na to okazały się obszary o małym zróżnicowaniu kolorystycznym, jak tło na wybranym obrazie. Zmiany te wykryć można poprzez porównanie histogramów poszczególnych obrazów (rysunek 3), czyli wykresy przedstawiające ilość pikseli o danym poziomie jasności.



Rysunek 3. Histogramy obrazów wejściowych oraz wyjściowego

Proces odwrotny, czyli ekstrakcja obrazu ukrytego polega na wczytaniu obrazu steganograficznego (złożonego), dokonaniu na nim konwersji do tablicy wartości pikseli *img3*, a następnie wydobyciu z niej dwóch – reprezentujących obraz bazowy *img1_new* i ukryty *img2_new*, by finalnie zwrócić je w postaci obrazów. Wydzielenie reprezentacji opiera się na rozdzieleniu 4 bitów najbardziej i najmniej znaczących dla każdego kanału kolejnych pikseli. Obrazy wyjściowe (wydzielone) otrzymują je jako bity najbardziej znaczące, a reszta uzupełniana jest losowo, co pozwala uzyskać naturalny efekt. Proces ten został przedstawiony na rysunku 4.

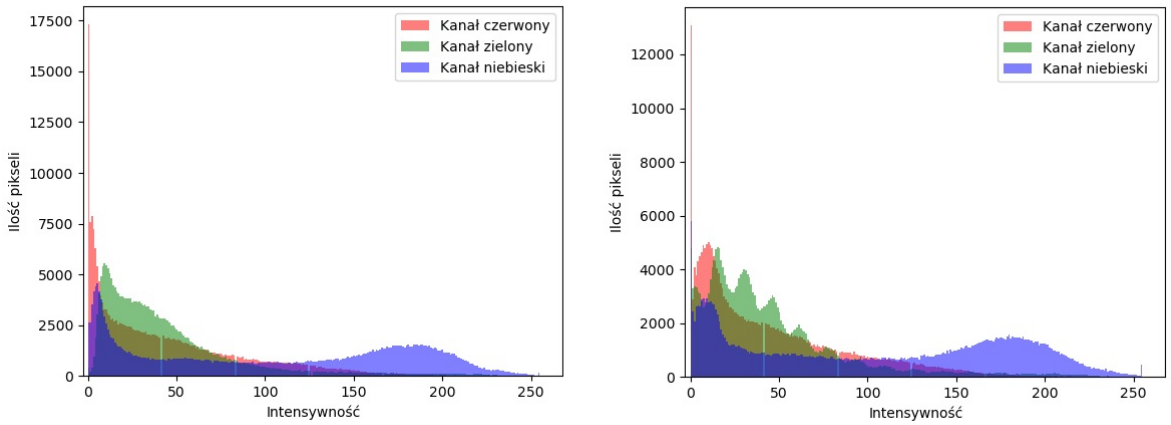


Rysunek 4. Proces ekstrakcji obrazów z obrazu steganograficznego

Rysunek 5. Obraz bazowy oryginalny (po lewej) oraz po wyodrębnieniu z obrazu steganograficznego (po prawej)

Rysunek 6. Obraz ukryty oryginalny (po lewej) oraz po wyodrębnieniu z obrazu steganograficznego (po prawej)

Oba obrazy po wyodrębnieniu różnią się od pierwotnych – wykazują gorszą jakość oraz zmianę barw w wybranych obszarach, co zostało przedstawione na rysunku 5 i 6. Jest to szczególnie ważne dla obrazu ukrytego, ponieważ oznacza ingerencję procesu w zapisaną informację. Jej stopień na poziomie barw sprawdzić można poprzez porównanie histogramów obrazu oryginalnego i po wyodrębnieniu (rysunek 7).



Rysunek 7. Histogram obrazu ukrytego oryginalnego (po prawej) oraz po wyodrębnieniu z obrazu steganograficznego (po lewej)

Sam obraz steganograficzny oraz proces ukrywania i ekstrakcji nie powinny być oceniane jedynie wizualnie, czy za pomocą rozkładu pikseli. Należy wyznaczyć parametry, dzięki którym możliwe będzie bardziej jednoznaczne wyznaczenie różnic pomiędzy obrazami na poszczególnych etapach, a następnie na ich podstawie wyciągnąć wnioski dotyczące jakości zastosowanego algorytmu. W tym celu wykorzystane zostaną wskaźniki charakteryzujące obrazy.

Jednym z powszechnie stosowanych, używanym do pomiaru jakości obrazu jest szczytowy stosunek sygnału do szumu (PSNR). Ta miara służy do określenia poziomu zniekształceń obrazu. Biorąc pod uwagę dwa obrazy x i y (są to oznaczenia uniwersalne stosowane we wszystkich współczynnikach, ponieważ wskaźniki obliczane będą dla różnych par obrazów) o wielkości (w, h) zapisane jako tablice o indeksach $[i][j][k]$, PSNR można zdefiniować jako funkcję błędu średniokwadratowego (MSE):

$$MSE = \frac{1}{3wh} \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=0}^{2} (x_{i,j,k} - y_{i,j,k})^2, \tag{1}$$

$$PSNR = 10 \log_{10} \frac{sc^2}{MSE}, \tag{2}$$

gdzie sc to współczynnik reprezentujący maksymalną możliwą wartość w numerycznej reprezentacji obrazu [3, 6].

W zaprezentowanym eksperymencie sc przyjmuje wartość 255, ponieważ w MSE przyjęto jako próbki wartości kanałów poszczególnych pikseli, a one zapisane są na 8-bitach z maksymalną wartością 255. W przypadku obrazów identycznych MSE równy jest 0, a szum nie występuje, wtedy też nie jest możliwe wyznaczenie PSNR. PSNR równe 0 dB oznacza maksymalną różnicę pomiędzy obrazami, a wraz ze wzrostem jego wartości, rośnie podobieństwo pomiędzy obrazami, co oznacza lepszą jakość badanego obrazu [3, 6].

Inną miarą jest wskaźnik podobieństwa strukturalnego (SSIM) pomiędzy dwoma obrazami. Jest on metryką o wiele dokładniejszą, ponieważ nie porównuje wartości pikseli, ale elementy obrazu postrzegane przez człowieka. SSIM najczęściej stosowane jest dla obrazów w skali szarości, ale istnieje kilka różnych podejść do wyznaczania jego wartości dla głębi kolorów RGB. Jednym z nich jest obliczenie podobieństwa strukturalnego dla każdego z kanałów osobno, a następnie wyznaczenie na ich podstawie jednego dla całego obrazu, dla obrazów x i y według wzoru:

$$SSIM(x,y) = \sum_{k=0}^{2} \frac{1}{3} SSIM(x_k, y_k), \tag{3}$$

gdzie x_k i y_k oznaczają wartości pikseli odpowiednio obrazu x i y dla danego kanału k [3, 5].

SSIM opiera się na obliczeniu trzech składników – luminancji l, kontrastu c i struktury s, a jego wynik jest multiplikatywną kombinacją tych trzech warunków:

$$SSIM(x_k, y_k) = [l(x_k, y_k)]^{\alpha} \cdot [c(x_k, y_k)]^{\beta} \cdot [s(x_k, y_k)]^{\gamma}, \tag{4}$$

gdzie α, β, γ to parametry umożliwiające regulowanie znaczenia poszczególnych składników. Składowe można zapisać jako:

$$l(x_k, y_k) = \frac{2\mu_{x_k}\mu_{y_k} + c_1}{\mu_{x_k}^2 + \mu_{y_k}^2 + c_1}, \tag{5}$$

$$c(x_k, y_k) = \frac{2\sigma_{x_k}\sigma_{y_k} + c_2}{\sigma_{x_k}^2 + \sigma_{y_k}^2 + c_2}, \tag{6}$$

$$s(x_k, y_k) = \frac{\sigma_{x_k y_k} + c_3}{\sigma_{x_k}\sigma_{y_k} + c_3}, \tag{7}$$

gdzie μ_{x_k} i μ_{y_k} to wartości średnie jasności, σ_{x_k} i σ_{y_k} odchylenia standardowe, $\sigma_{x_k y_k}$ kowariancja krzyżowa oraz stałe c_1, c_2:

$$c_1 = (p_1 R)^2, \tag{8}$$

$$c_2 = (p_2 R)^2, \tag{9}$$

gdzie R to zakres dynamiczny, a p_1 i p_2 stałe regularyzacji. Przyjmując równe znaczenie trzech warunków, czyli $\alpha = \beta = \gamma = 1$ oraz stałą c_3 jako:

$$c_3 = \frac{c_2}{2}, \tag{10}$$

otrzymuje się [5]:

$$SSIM(x_k, y_k) = \frac{(2\mu_{x_k}\mu_{y_k} + c_1)(2\sigma_{x_k y_k} + c_2)}{(\mu_{x_k}^2 + \mu_{y_k}^2 + c_1)(\sigma_{x_k}^2 + \sigma_{y_k}^2 + c_2)}. \tag{11}$$

SSIM wykorzystuje stałe regularyzacji dla luminancji, kontrastu i składników strukturalnych, aby uniknąć niestabilności w obszarach obrazu, w których średnia jasność lub odchylenie standardowe jest bliskie zeru. Dlatego dla tych stałych należy używać małych wartości niezerowych. Domyślnie $p_1 = 0,01$ i $p_2 = 0,03$, natomiast zakres dynamiczny $R = 255$, ponieważ wartości SSIM są obliczane osobno dla każdego kanału. Wskaźnik podobieństwa strukturalnego przyjmuje wartości z zakresu [-1, 1], gdzie 1 wskazuje, że obrazy są identyczne [3, 5].

Tabela 1. Wyznaczone wartości wskaźników

Wskaźnik	*img1* oraz *img3*	*img2* oraz *img2_new*
MSE	$51,90$	$45,92$
$PSNR$ $[dB]$	$30,98$	$31,51$
$SSIM$	$0,92$	$0,89$

Wyznaczone wskaźniki dla obrazu steganograficznego (tabela 1) obliczone zostały poprzez porównanie obrazu bazowego (*img1*) oraz stanowiącego połączenie dwóch (*img3*). Mają one znaczenie przy ocenie tajności procesu, ponieważ wskazują różnice w obrazie oryginalnym, a steganograficznym i mogłyby zostać wykorzystane przez osobę chcącą wykryć podobne działania. Wartość wskaźnika PSNR na poziomie 30,98 dB można uznać za zadowalający, ze względu na fakt ukrycia znacznej ilości informacji w obrazie (drugi obraz). SSIM wynoszące 0,92 natomiast wskazuje duże podobieństwo pomiędzy obrazami.

Możliwe jest również wyznaczenie mapy lokalnych wartości SSIM, czyli ich graficznej reprezentacji. Zawiera ona wartości SSIM dla poszczególnych pikseli, gdzie małe wartości SSIM są wyświetlane jako ciemne piksele, natomiast duże jako jasne. Regiony o małej lokalnej wartości SSIM odpowiadają obszarom, w których obraz steganograficzny wyraźnie różni się od obrazu bazowego [3]. Wygenerowana mapa (rysunek 8) zawiera głównie tło, co potwierdza że jednolite pod względem kolorystycznym obszary są szczególnie wrażliwe na zmiany.

Rysunek 8. Mapa lokalnych wartości SSIM dla obrazu steganograficznego

Wskaźniki dla obrazu wyodrębnionego obliczone zostały na podstawie obrazu ukrytego oryginalnego oraz wyodrębnionego z obrazu steganograficznego. Mają one znaczenie przy ocenie procesu pod kątem stopnia zachowalności i utraty danych, co przekłada się na jego jakość. Błąd średniokwadratowy MSE oraz obliczony na jego podstawie wskaźnik PSNR wskazują, że pojawił się szum, co oznacza że dane nie zostały zachowane w niezmienionej formie. Jednak parametr SSIM wykazuje podobieństwo obrazów na poziomie 0,89.

3. Wnioski

Algorytmy tworzenia plików steganograficznych można projektować z dużą dowolnością i dostosowywać w zależności od wymagań. Jednak z uwagi na ich mnogość i na wpływ zbyt wielu czynników, są one trudne w bezpośrednim porównywaniu. Nawet istniejące miary jakości obrazu, które służą do oceny procesu na różnych etapach, także obliczane są na wiele sposobów oraz zależą od przetwarzanych obrazów. Większość z nich zaprojektowana została z myślą o tych w skali szarości, przez co stosowanie ich dla kolorowych np. o modelu przestrzeni barw RGB wymaga użycia pewnych przekształceń lub zamiany w obraz monochromatyczny, co jednak wiąże się ze znaczną utratą danych [1]. Zakres wartości wskaźników nie jest jednoznaczny i zależy od złożoności analizowanych obrazów, więc ocena uzyskanych wyników także może być trudna do porównania z otrzymanymi przez innych autorów. Zastosowany w artykule algorytm pomimo swej prostoty okazał się skuteczny i wydajny, a otrzymane z jego użyciem obrazy wykazują się dobrymi parametrami. Może być on modyfikowany i rozwijany w przyszłości. Jego optymalizacja to ciekawy obiekt badań, szczególnie w połączeniu z technikami kryptograficznymi.

Obrazy są doskonałymi nośnikami danych, charakteryzują się dużą pojemnością, wystarczającą nawet do przechowywania w nich innych obrazów, czyli danych o takich samych rozmiarach. Jednak wraz ze wzrostem ilości ukrywanych informacji, jakość obrazów steganograficznych będzie spadała, a sam proces będzie łatwiejszy do wykrycia. W przypadku gdy ukrywanymi danymi są także obrazy, znaczny wpływ ma zbieżność występujących barw w określonych punktach z obrazem bazowym. Na skutek tego wyniki mogą się nieznacznie różnić dla innych par obrazów. Przy wykonywaniu każdej z operacji obrazy nie wracają zupełnie do swojej pierwotnej formy (wykazują gorszą jakość już wizualnie). Jest to skutek potwierdzonych przez histogramy i wskaźniki zmian w ich budowie, które jednak są podstawą zastosowanego algorytmu.

Podziękowania

Autorka pragnie podziękować recenzentom za trud włożony w recenzje.

Literatura

1. A. Cheddad, J. Condell, K. Curran, P. M. Kevitt, *Digital image steganography: survey and analysis of current methods*, Signal Processing, vol. 90, no. 3, pp. 727–752, 2010.
2. W. Garbarczuk, P. Kopniak, *Steganologia: współczesne metody ochrony informacji (przegląd)*, Pomiary Automatyka Kontrola (2005), nr 3, s. 21-25.
3. *MATLAB 2018b*, The MathWorks Inc., Natick, Massachusetts, 2018.
4. *Open Images Dataset*, https://storage.googleapis.com/openimages/web/index.html.
5. Z. Wang, A. Bovik, H.R. Sheikh, E.P. Simoncelli, *Image Qualifty Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing 13(4) (2004), pp. 600–612.
6. K.A. Zhang, A. Cuesta-Infane, K. Veeramachaneni, *SteganoGAN: High Capacity Image Steganography with GANs*, arXiv preprint arXiv:1901.03892, 2019.