Tekstur og dekkval i biletsteganografi

Hans Georg Schaathun
⟨georg@schaathun.net⟩
Høgskolen i Ålesund, Institutt for Ingeniør- og Realfag
Postboks 1517, 6025 Ålesund

Samandrag

Steganografi er teknikkar for hemmeleg kommunikasjon, der sjølve eksistensen av den hemmelege meldinga må haldast løynd. Ulike formar for steganografi er kjende sidan oldtida. I vår tid kan informasjon skjulast i bilete og multimedia ved å bruka enkle moduleringsteknikkar. Metodar for å oppdaga skjult informasjon kallar me for steganalyse.

I dette arbeidet ser me på korleis eit skjønnsomt val av dekkbilete kan gjera steganografi vanskelegare å påvisa. Me ser på fordelinga av koeffisientverdiar i diagonalkomponenten frå ein *wavelet*-dekomposisjon. Dersom me modellerer fordelinga som ei generalisert normalfordeling, kan me bruka formparameteren β som eit mål for tekstur i biletet. Dekkval basert på β er imponerande effektivt for å oppnå upåviseleg steganografi.

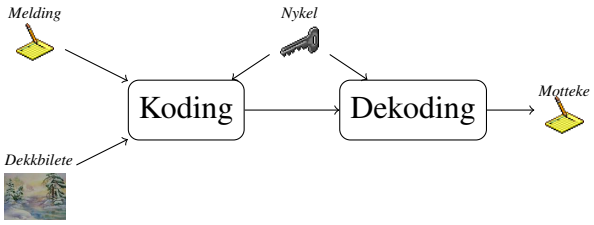
1 Innleiing

Moderne steganografi vert som regel tilskrive Simmons [18], som formulerte fanganes problem. Alice og Bob er fengsla. Dei har lov til å kommunisera, men fangevaktaren Wendy overvaker alt dei sender. Somme tema, som t.d. fluktplanar, er openbert ikkje lovlege. Har Alice og Bob slikt å diskutera, må dei gjera det i løynd. Kryptering har liten verdi for Alice og Bob. Krypterte meldingar ser ut som tilfeldige, meiningslause data, og Wendy vil umiddelbart fatta mistanke og blokkera kommunikasjonen. For å diskutera ulovlege tema, må Alice og Bob skapa eit inntrykk av at dei har ei normal, uskuldig samtale samstundes som dei utvekslar løynde meldingar.

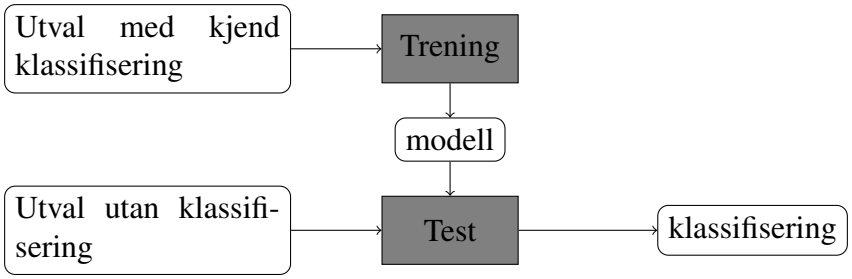
Eit velkjent døme på steganografi er usynleg blekk. Alice kan senda eit brev med to meldingar; ei uskuldig melding skrive med vanleg blekk og ei løynmelding skrive med usynleg blekk. For Wendy ser det då ut som eit heilt uskuldig og lovleg brev, medan Bob, som veit kva han skal sjå etter, kan behandla brevet slik at løynmeldinga vert synleg.

Moderne steganografi konsentrerer seg om digitale medium. I prinsippet kan ein gøyma meldingar i alle typar filer, som musikk, tekst, video, programvare, osv. Likevel er bilete det dominerande mediet i litteraturen, med ei lang rekkje lovande løysingar som er relativt enkle å forstå og implementera. I dette arbeidet vil me fokusera på steganografi i pixmap-bilete (t.d. PBM). Av omsyn til lesarar med generell interesse prioriterer me ein grundig innføring i problemområdet, og håper å koma tilbake med ein meir detaljert analyse for spesielt interessert i eit anna forum.

Denne artikkelen vart presentert på konferansen NIK-2012; sjå http://www.nik.no/.



Figur 1: Eit typisk system for steganografi ved modifikasjon.



Figur 2: Maskinlæringssystem. (Diagram som publisert ved NIK2011.)

2 Problemdefinisjon

Steganografi

Eit typisk system for steganografi, eller stego-system i kortform, er vist i Figur 1. For å koda ei hemmeleg melding treng me to objekt inn: den hemmelege meldinga og eit dekkbilete. Resultatet kaller me eit *steganogram* eller stego-bilete. Dekkbiletet vert modulert for å representera den hemmelege meldinga, i prinsippet til liks med tradisjonell telekommunikasjon der ei berebølgje vert modulert.

Det seier seg sjølv at det må vera uråd å skilja eit steganogram frå eit umodifisert dekkbilete (reint bilete), men ordet *skilja* er ope for ein viss diskusjon. Dersom Wendy inspiserer biletet manuelt og visuelt, er det fyrst og fremst viktig at steganogrammet *ser ut som* eit naturleg bilete. Når biletet vert analysert maskinelt, må steganogrammet ha den same *statistiske* fordelinga som dekkbiletet, fordi ein maskinell statistisk analyse lett kan peika på skilnader som ikkje er synlege for eit menneskeauga.

Det er verd å merkja seg at, til skilnad frå andre formar for *data hiding*, er dekkbiletet i steganografi ikkje viktig. Det vert ikkje brukt på mottakarsida. Det har fleire fylgjer. Det gjer ingenting om Wendy kan skilja dekkbiletet og steganogrammet frå kvarandre når ho ser dei side om side, for det gjer ho likevel ikkje, og steganogrammet skal aldri brukast i staden for dekkbiletet. Dekkbiletet treng heller ikkje vera inndata til kodinga. Sjølv om dei er sjeldne, so finst steganografisystem som syntetiserer eit bilete eller tekstdokument som ein del av kodinga (steganografi ved dekksyntese). Når ho bruker steganografi ved modifikasjon, har Alice høve til å velja eit dekkbilete som gjer jobben særleg vanskeleg for Wendy. Dette valet kan vera avhengig eller uavhengig av meldinga.

Det er velkjent at somme dekkbilete gjer steganografi vanskelegare å oppdaga enn andre, men det er skrive forbløffande lite om korleis me kan dra nytte av det. Sjølv om dekkval er eit openbert viktig tema i steganografi, finst der få praktiske løysingar.

Steganalyse

Steganalyse er eit typisk klassifiseringsproblem. Der finst to klasser av bilete, nemleg steganogram og naturlege bilete (dekkbilete). Oppgåva åt Wendy er å avgjera, for eit vilkårleg bilete I, kva klasse det høyrer til.

Dei dominerande steganalyseteknikkane i seinare år er baserte på maskinlæring eller *pattern recognition*. Eit typisk system er vist i figur 2. I treningsfasen lagar systemet ein modell som seinare kan brukast til å klassifisera objekt med ukjend klasse. Objekta er normalt for store til å handsamast direkte, og difor vil ein alltid trekkja ut ein so-kalla *feature*-vektor som vert brukt i staden for objektet.

Ein *feature* er ein funksjon $f : \mathfrak{I} \to \mathbb{R}$, der $\mathfrak{I}$ er mengda av alle bilete. Funksjonsverdien $f(I)$ for eit bestemt bilete $I \in \mathfrak{I}$ vil me kalla ein *feature*-verdi. Dersom me har fleire features $f_1, f_2, \ldots, f_d$, so kallar me $(f_1(I), f_2(I), \ldots, f_d(I)) \in \mathbb{R}^d$ for ein *feature*-vektor. Ein *feature* kan vera einkvan utreknbar funksjon som helst. Middelverdi og statistiske moment av pixelverdiar eller wavelet-koeffisientar er vanlege *features*.

Systemet i figur 2 er det same for alle klassifikasjonsproblem. Der er eit par vanlege klassifikasjonsalgoritmar, medan valet av algoritme stort sett er ei enkel avveging mellom køyretid og presisjon og er uavhengig av klassifikasjonsproblemet. I dette arbeidet har me brukt *støttevektmaskiner* (SVM) som er enkle å bruka, med få konfigureringsopsjonar [6], og med fornuftig køyretid opp til nokre tusen *features* og fleire tusen treningsbilete. *Features* må derimot utviklast spesielt for den aktuelle klassifikasjonen, og utgjer dermed hovudutfordringa i steganalyse.

For å vurdera nøyaktigheita på ein nytrent klassifikator, må han testast på eit *testsett* som er uavhengig av treningssettet. Heilt enkelt klassifiserer ein kvart objekt frå testsettet og tel kvar gong klassifikatoren svarer feil. Ein kan skilja mellom falske positivar, der klassifikatoren seier «stego» for eit bilete som eigentleg er reint, og falske negativar der klassifikatoren seier «reint» for eit steganogram. Me vil referera til tre ytingsmål seinare i artikkelen:

$$\text{nøyaktigheit} \qquad A = 1 - \frac{\mathrm{FP} + \mathrm{FN}}{n_1 + n_0}, \tag{1}$$

$$\text{FP-rate} \quad e_{\mathrm{FP}} = \frac{\mathrm{FP}}{n_0}, \tag{2}$$

$$\text{FN-rate} \quad e_{\mathrm{FN}} = \frac{\mathrm{FN}}{n_1}; \tag{3}$$

der FP og FN er talet på falske positivar og negativar høvesvis, og n_0 og n_1 er talet på høvesvis reine bilete og steganogram i testsettet.

3 Kjende løysingar

Steganografi

Steganografiske meldingar kan i prinsippet modulerast i einkvan biletrepresentasjon, anten i pixmap, i *wavelet*-transformasjonar, eller i blokk-DCT-domenet (t.d. JPEG). Eit bredt utval av programvareimplementasjonar har sirkulert på nettet i alle fall sidan midten av 90-talet, i tillegg til algoritmar formelt skildra i litteraturen. Stort sett byggjer alt på dei same grunnprinsippa. Den klassiske teknikken i pixmap er LSB (*least significant bit*) som går ut på å erstatta den minst signifikante bitten i kvar pixel med ein meldingsbit. Dekoding er simpelthen reduksjon modulo 2. Ei forbetring er LSB$\pm$ som nyttar same dekodingsalgoritme, men som legg til ± 1 tilfeldig når ein pixel må endrast, i staden for å

byta ut den minst signifikante bitten og halda resten konstant. Det er vanleg å lesa pixlane i slumpmessig rekkjefylgje, der frøet til slumptalsgeneratoren vert ein hemmeleg nykel for Alice og Bob. Både LSB og LSB± er rekna for relativt lette å oppdaga, men det gjeld fyrst og fremst ved lange meldingar.

Eit relativt nytt system er Hugo [13], som kombinerer modulasjonsteknikken frå LSB saman med kodeteori. Kodeteknikkane byggjer på dei som er kjende for avgrensa minne (*constrained memory*), og gjer at ein bruker fleire pixlar per meldingsbit men gjer færre endringar. Med færre endringar vert meldinga vanskelegare å oppdaga. I tillegg er kodinga organisert slik at dei fleste pixelendringane skjer der dei er verst å oppdaga.

Steganalyse

Ein av dei mest kjende *feature vectors* for pixmap-steganografi er SPAM-848 [11]. Idéen er å utnytta den statistiske fordelinga av pixelgrupper. Fyrst dannar ein differansematrisa D ved ta pixmap-matrisa M og trekkja verdien av nabopixelen frå kvar pixel. Vassrette, loddrette og diagonale differansar gjev opphav til fire separate differansematriser. Deretter modellerer ein D som ein markovkjede der ein les elementa i same retning som ein tok differansen. Overgangssannsyna i fyrste og andre ordens markovkjeder vert brukte som *features*. Dette gjev *features* som avheng av grupper på totalt tre og fire pixlar, og me kan kalla dei tredje- og fjerdeordens *features*.

Hugo vart designa for å minimera verknaden opp til og med fjerde orden. For å oppdaga Hugo, vart HOLMES [4] utvikla med *features* av endå høgare orden. Den enklaste løysinga er å bruka tredje og høgare ordens markovkjeder, men dette gjev raskt ein overflod av *features*. I tillegg kan ein bruka høgare ordens differansematriser, t.d. andre ordens $\Delta x_0 = 2x_0 - x_{-1} - x_{+1}$ der x_{-1} og x_{+1} er nabopixlane på kvar side av x_0. Kombinasjonen av andre ordens differansar og tredje ordens markovkjede gjev sjette ordens *features*. HOLMES bruker ei lang rekkje variantar av slike høgordens *features*, med ein total dimensjon om lag 25 000.

Dekkval

Bilethandsaming er ein grein av signalprosessering, og det er svært nyttig å sjå på pixelverdiane som eit signal som kan studerast i eit todimensjonalt frekvensdomene. Modulasjonen i eit steganografisystem kan sjåast som eit høgfrekvent støysignal $\mathbf{x}$ som vert addert til dekkbiletet $\mathbf{c}$. Dvs. steganogrammet $\mathbf{s}$ er gjeve som $\mathbf{s} = \mathbf{x} + \mathbf{c}$. Det er den høge frekvensen frå $\mathbf{x}$ som vert fanga opp av dei fleste steganalysesystem. Det fungerer godt når dekkbiletet $\mathbf{c}$ er prega av monotone og jamne fargeovergangar, som har låg frekvens. Diverre er det stor variasjon i naturlege bilete, og det er vanskeleg i steganalyse å skilja mellom høgfrekvent støy, og høgfrekvente, informasjonsberande signal som høyrer naturleg heime i biletet. Det seier seg sjølv av steganografen kan oppnå mykje ved selektivt å velja dekkbilete som naturleg har mykje høgfrekvent informasjon som kan gje skjul for løyndomen.

Kharrazi *et al* [9] var blant dei fyrste til å diskutera dekkval i steganografi. Alle teknikkane dei gjennomgjekk baserer seg på ein eller annan målbar heuristikk som skal minimerast eller maksimerast for det optimale dekkbiletet. Slik *dekkvalsheuristikkar* kan enkelt innarbeidast i eksisterande stego-system, på ein av fleire ulike måtar. Dersom steganografisystemet automatisk vel eit tilfeldig dekkbilete frå ein database, so kan ein bruka heuristikken til å filtrera bort bilete som er venta å vera dårlege. Dersom brukaren må gje dekkbiletet manuelt som inndata til programmet, kan ein skriva ut ein åtvaring dersom heuristikken gjev grunn til pessimisme. Kva terskelverdiar ein bør velja, både i

filteret og for åtvaring, avheng av mange faktorar, som risikovilje, meldingslengd, og kva bilete ein har tilgang til.

Kharrazi *et al* [9] skilte mellom dekkbaserte og dekk-stego-baserte teknikkar. Fyrstnemnde vel dekkbiletet heilt uavhengig av meldinga og stegoteknikken som skal brukast, medan sistnemnde testar stegokoding i kvart dekkbilete og vel det beste steganogrammet. Det er rimeleg å tru at dekk-stego-baserte heuristikkar vil vera meir effektive fordi dei dreg nytte av meir informasjon, og det kan aldri vera ei ulempe. Derimot vil reine dekkbaserte teknikkar normalt vera raskare, sidan ein kan analysera moglege dekkbilete éin gong for alle og byggja ein database av egna dekkbilete. Dekk-stego-teknikkar vurderer dekkbileta på nytt kvar gong noko skal sendast. Av plassomsyn vil me sjå bort frå dekk-stego-teknikkar i resten av artikkelen.

To reine dekkbaserte heuristikkar vart nemnde av Kharrazzi *et al*, nemleg talet på modifiserbare koeffisientar og kvalitetsfaktoren i JPEG. Sistnemnde er sjølvsagt berre relevant for komprimerte bilete, og i stor grad gjeld det òg det fyrste. Dei fleste stegoteknikkar for pixmap-bilete kan modifisera alle pixlar. I JPEG, derimot, er det vanleg ikkje å røra koeffisientar som i utgangspunktet er 0, og talet på modifiserbare koeffisientar avheng av komprimeringsfaktoren og av teksturnivået i biletet.

Sajedi og Jamzad [15] føreslo ein kapasitetsbasert teknikk som ikkje passar inn i dei to kategoriane åt Kharrazi *et al*. Dei freista å estimera *sikker kapasitet* i kvart dekkbilete ved å testa ulike meldingslengder mot kjende steganalyseverkty. Den sikre kapasiteten er definert som den største meldinglengda som ikkje vert oppdaga. Når ei gjeven melding skal sendast, vel ein eit tilfeldig dekkbilete med kapasitet større enn meldinga. Eit mogleg problem med denne teknikken er at han er bunden til kjende steganalyseteknikkar og det er vandt å forutseia kapasiteten i høve til ukjende teknikkar som Wendy kan nytta.

Der finst òg ein del arbeid som integrerer dekkval som ein del av steganografialgoritmen [8, 14]. Det vil seia at Alice kan dra nytte av dekkval, men me får ikkje umiddelbart ein gjenbrukbar dekkvalsalgoritme som kan kombinerast med det ypparste av modulasjon.

Me kjenner til eitt arbeid [19] som ser på ein generell heuristikk for dekkval, og som ikkje føreset bestemte modulasjons- eller steganalyseteknikkar. Dei bruker korrelasjonskoeffisienten mellom par av tilstøytande pixlar.

4 Metodologi

Metodologien vår [16] fylgjer det som er standard i steganalyselitteraturen for å testa kvar kombinasjon av steganografialgoritme, dekkval og steganalyseteknikk. Trenings- og testbilete er henta frå BOSS-konkurransen [1] som inneheld 10 000 ukomprimerte gråtonebilete i 512×512. Desse vert delt inn tilfeldig i trenings- og testsett på 5000 bilete kvar. Innanfor kvart sett er helvten brukt som reine bilete og resten som steganogrammar. Separate trenings- og testsett er danna for kvar steganografialgoritme og meldingslengd, men slik at dei same dekkbileta er brukt til steganogramma i alle setta. Som klassifiseringsalgoritme har me brukt SVM og fylgt metodologien åt Hsu *et al* [6]. Som *features* har me hovudsakleg brukt SPAM-848 [12].

Me har testa tre ulike stego-system, nemleg Hugo simulert vha. forfattarane sin programvare, samt LSB og LSB$\pm$. Me oppgjev meldingslengda i prosent av kapasitet, dvs. som bit per pixel. Me har lagt størst vekt på 40% Hugo og 10% LSB$\pm$.

Hovudutfordringa vår har vore å halda styr på store mengder utrekna *features* saman med dekkvalsheuristikkar, og gjera ulike analyser utan å måtte gjenta utrekningane. Køyretida på *feature*-utrekningane er opp mot eitt CPU-minutt per bilete. Dette krev ein form for parallellisering og systematisk organisering av resultat. Me har brukt ein

SQL-database for å halda orden på *features*, og me har òg laga eit enkelt køsystem i den same databasen. Slik kan me køyra uavhengige og parallelle klientar som reknar ut *features*. Både dekkvalsheuristikkane og klassifiseringsheuristikkane frå SVM vert rekna som *features* og lagt i den same databasen. Det gjer det enkelt å gjera statistisk analyse på ulike kombinasjonar av heuristikkar og *features*. Programvaren er skriven i python og publisert under GPL [17].

5 Heuristikk

Tekstur i biletet har som kjent stor innverknad på steganalyse, og det er fornuftig å sjå etter heuristikkar som kan måle teksturnivå. Tekstur tyder at der ofte er stor skilnad mellom nabopixlar, dermed bør der vera lite korrelasjon. Korrelasjonskoeffisienten mellom to stokastiske variablar X og Y er definert som

$$\rho = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}}. \tag{4}$$

For å få ein dekkvalsheuristikk kan me lata (X,Y) vera par av nabopixlar. Me har testa både horisontale og vertikale par, og skriv høvesvis ρ_h og ρ_v for korrelasjonskoeffisientane.

Eit anna mål for tekstur vart innført av [2]. Dei såg på ein *wavelet*-transform av biletet (eitt nivå med Haar-*wavelet*) og modellerte fordelinga i den diagonale komponenten med ei generalisert Gauss-fordeling, som er definert ved

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp -(|x-\mu|/\alpha)^\beta, \tag{5}$$

der Γ er gammafunksjonen. Parameteren β vert gjerne kalla formparameteren. For $\beta = 2$ har me normalfordelinga, medan $\beta < 2$ gjev tyngre halar og $\beta > 2$ tynnare halar. Van de Wouwer *et al* [2] argumenterte for at β-parameteren er eit mål for tekstur, der høge verdiar for β indikerer mykje tekstur.

Van de Wouwer *et al* [2] skriv at β kan estimerast som

$$\hat{\beta} = F^{-1}\left(\frac{\hat{m}_1^2}{\hat{m}_2}\right) \quad \text{der} \quad F(x) = \frac{\Gamma^2(2/x)}{\Gamma(1/x)\Gamma(3/x)} \tag{6}$$

og m_1 og m_2 er fyrste og andre ordens absolutte statistiske moment, dvs.

$$m_1 = \frac{1}{N}\sum_{i=1}^{N}|x_i|, \quad m_2 = \frac{1}{N}\sum_{i=1}^{N}x_i^2,$$

for N observasjonar x_i.

Det er kjent at ρ kan brukast som dekkvalsheuristikk [19]. Det er òg vist at steganalyse er lettare for låg β enn for høg [10], men me kjenner likevel ikkje til noko arbeid på β som dekkvalsheuristikk. Høge verdiar for β og låge verdiar for ρ indikerer bilete godt egna for steganografi. Tidlegare forfattarar har brukt relativt små biletsamlingar på 3000–5000 bilete og lange meldingar frå 25% og oppover. Me vil stadfesta at ρ er egna som dekkvalsheuristikk for nye stego-teknikkar, kortare meldingar og andre *features*, og me skal sjå at β-parameteren i fleire tilfelle er vesentleg betre.

Feature	Gjennomsnitt	Varians	Skeivheit	Kurtosis
ρ_h	0.967	0.00137	-2.78	13.2
ρ_v	0.961	0.00171	-3.03	16.8
β	0.529	0.0527	1.51	2.80
$\Delta_{\text{HUGO}}\rho_h$	$-7.9 \cdot 10^{-05}$	$7.7 \cdot 10^{-09}$	-8.7	151
$\Delta_{\text{HUGO}}\rho_v$	$-7.9 \cdot 10^{-05}$	$6.9 \cdot 10^{-09}$	-7.0	119
$\Delta_{\text{HUGO}}\beta$	$3.5 \cdot 10^{-03}$	$7.6 \cdot 10^{-05}$	5.6	45.4
$\Delta_{\text{LSB}\pm}\beta$	$2.1 \cdot 10^{-02}$	$8.3 \cdot 10^{-04}$	3.9	18.3

Tabell 1: Statistiske moment for dei ulike heuristikkane.

6 Analyse og resultat

Intuitivt veit me at bilete med mykje tekstur, og som difor er godt egna som dekkbilete, vil ha høg β og liten ρ. Tabell 1 gjev eit statistisk overblikk over desse heuristikkane. Det er verd å merka seg at ρ har svært låg varians, og samstundes er gjennomsnittet svært nær den maksimale verdien på 1. Det kan tyda på at det er vanskeleg å skilja mellom gode og mindre gode dekkbilete. Variansen for β er ikkje veldig høg, men likevel vesentleg betre enn for ρ. Skeivheita, som er negativ for ρ og positiv for β, fortel oss at bileta hopar seg opp i den enden av spekteret som er dårleg egna for steganografi.

Det er òg interessant å sjå på forskjellen mellom eit steganogram og tilhøyrande dekkbilete. Me definerer $\Delta f(I)$ for ein gjeven statistikk f som $f(I) - f(C)$ der C er dekkbiletet som vart brukt for å laga steganogrammet I. Tabellen viser svært låge verdiar i gjennomsnitt og varians for Δf, noko som fortel oss at heuristikkane våre vert svært lite påverka av 40% HUGO. Dermed kan Wendy rekna ut heuristikkane og få praktisk talt den same verdien som Alice. Dette kan brukast som ein indikasjon på kor god klassifiseringa hennar kan ventast å vera. Stegosystem som er enklare å oppdaga kan òg gjera meir signifikante endringar i β, og tabellen viser eit døme for 20% LSB$\pm$. Korrelasjonskoeffisienten er derimot meir stabil enn β.

Me kan òg merka oss korrelasjonen mellom β og ρ:

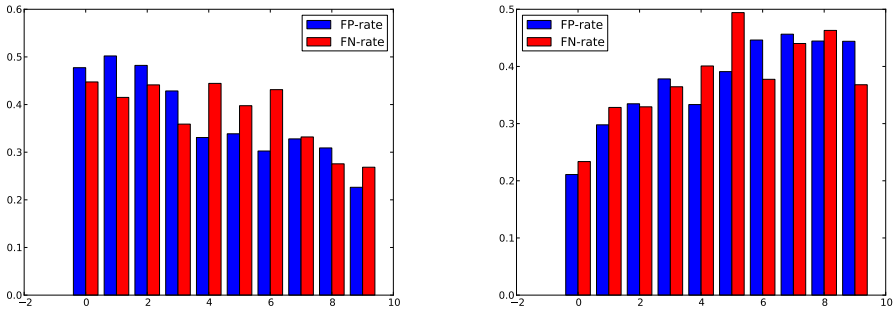
$$\text{cor}(\beta, \rho_h) = -0{,}2147, \tag{7}$$
$$\text{cor}(\beta, \rho_v) = -0{,}2768. \tag{8}$$

Til samanlikning har me $\text{cor}(\rho_h, \rho_v) = 0{,}8694$. Utrekninga er basert på 5000 bilete frå BOSS. Dermed kan me venta at ρ og β vil ha ulike eigenskapar ved dekkval.

Dekkvalsheuristikkar

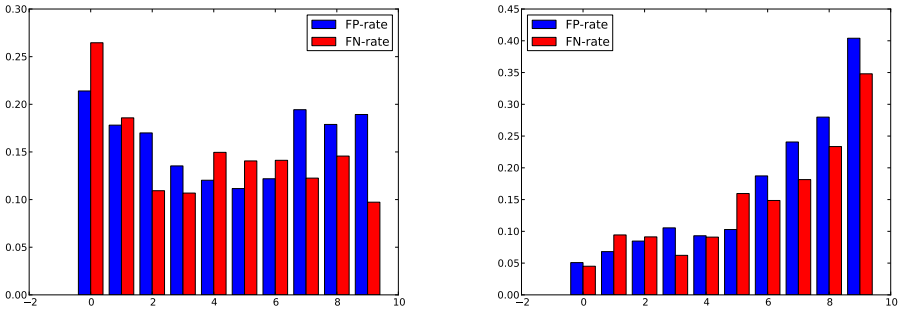
Me har testa heuristikkane ved å dela testbileta i ti jamnstore grupper basert på kvar heuristikk. Klassifikatoren er so testa på kvar gruppe kvar for seg. Figur 3 viser feilratane for 40% HUGO som eit stolpediagram. Me ser tydeleg at feilratane, i alle fall om ein ser falske positivar og negativar under eitt, varierer nokonlunde monotont med heuristikken, frå rundt 20% til over 45%. Det viser at SPAM-848 er svært følsam for dekkval.

Figur 4 viser eit tilsvarande plott for 10% LSB$\pm$. Her ser me at β-parameteren har ein endå meir éintydig verknad på feilratane. Igjen er feilraten over 40% for velvalde dekkbilete. Korrelasjonskoeffisienten på den andre sida har ein mindre tydeleg effekt. Bortsett frå for dei aller lågaste verdiane, so er der ikkje nokon klar tendens. Ei heller ser me feilratar over 25%.



(a) Korrelasjonskoeffisient (b) β-parameteren

Figur 3: Samanlikning av feilratane for ulike dekkgrupper for 40% simulert HUGO. Dekkbileta er inndelte i ti jamnstore grupper basert på dekkvalsheuristikken.



(a) Korrelasjonskoeffisient (b) β-parameteren

Figur 4: Samanlikning av feilratane for ulike dekkgrupper for 10% LSB±. Dekkbileta er inndelte i ti jamnstore grupper basert på dekkvalsheuristikken.

Ein kan spørja seg om dekkval er like effektivt dersom treningssettet for klassifikatoren vert avgrensa med same kriterium. Me har testa det ved å avgrensa både treningssettet og testsettet til $\beta > 0.669$, evt. $\rho < 0.948$, noko som svarer til den beste 20-prosentilen. Det avgrensa testsettet har derimot vesentleg dårlegare nøyaktigheit på klassifikatoren med avgrensa treningssett enn med den opprinnelege klassifikatoren. For 10% LSB± og $\beta > 0.669$ t.d. går nøyaktigheita frå 68,6% til 62,8%. Storleiken på treningssettet kan medverka til dette.

Testane på BOSS-datasetta vert til dels stadfesta av testar på ein tilsvarande biletbase frå BOWS2, men BOWS2 gjev gjennomgåande fleire klassifikasjonsfeil og gjev difor knapt signifikante resultat. Grunnen til dette er truleg at BOWS2-bileta generelt er noko meir teksturerte, både med lågare ρ og høgare β. Andre *feature*-vektorar gjev òg liknande resultat. Me har testa Farids 72 *features* og ein versjon av WAM-27 [5]. Den einaste signifikante skilnaden er at WAM-27 får svært høg FP-rate og låg FN-rate på teksturerte bilete. Likeeins oppfører 10% LSB og 20% LSB og LSB± omlag som 10% LSB±.

Variasjon mellom dekkbilete

Det er verd å merkja seg at feilsannsynet avheng meir av dekkbiletet enn av den skjulte meldinga eller stego-algoritmen. Me køyrde ein test med seks steganogram basert på

	0	1	2	3	4	5	6
0	158	30	21	15	13	21	37
1	69	15	14	12	9	13	26
2	46	18	20	15	12	20	38
3	58	10	7	18	8	15	52
4	70	15	15	8	22	23	39
5	89	18	27	24	26	24	112
6	626	236	197	203	267	356	1813

Tabell 2: Tal på korrekte klassifikasjonar per dekkbilete for 10% LSB± (rekkjer) og 40% Hugo (søyler).

kvart av dei 5000 dekkbileta i testsettet vårt, både for 10% LSB± og for 40% Hugo. Klassifikatoren vart testa for kvart steganogram, og feila talt opp for kvart dekkbilete. Resultatet er vist i tabell 2. Me ser at 36% av dekkbileta gjev korrekt klassifikasjon i 12 av 12 tilfelle. Det er rett nok få bilete (litt over 3%) som vert konsekvent feilklassifisert, men det er venteleg med den låge feilraten for LSB±. Me ser òg stor korrelasjon mellom dei to stego-algoritmane når me samanliknar feiltal per dekkbilete. Dette styrkar inntrykket av at eigenskapar ved dekkbiletet er vel so viktige som algoritmen.

Dei dekkbileta som vert feilklassifiserte for LSB± har markert høgare β enn dei som vert konsekvent rett klassifiserte: $\beta = 0{,}48$ for 6/6, 0,58 for 5/6, og 0,63–0,68 for dei fire siste gruppene. Variansen er høvesvis 0,36, 0,44, og 0,51–0,69. Dekkvalsheuristikken kan altso ikkje aleine identifisera konsekvent feilklassifiserte bilete, men kan venteleg identifisera grupper med nær 50% feilsannsyn.

7 Oppsummering og opne spørsmål

Me har vist at enkle dekkvalsteknikkar kan gjera eksisterande steganografisystem vesentleg sikrare. Til og med ein teknikk som LSB± som vert rekna som usikker og passé får feilratar over 40% ved 10% meldingslengd når me bruker dei beste tiandedelen av dekkbileta. Vidare har me vist at dekkval ikkje er uavhengig av steganografisystemet. Med HUGO får me dei beste resultata med korrelasjonskoeffisienten, medan LSB og LSB± får dei beste resultata ved dekkval med β.

Artikkelen reiser fleire spørsmål enn han gjev svar. Det mest umiddelbare spørsmålet for vidare gransking er kor effektiv dekkvalet vil vera mot steganalyse med HOLMES-*features*. På grunn av dimensjonaliteten vil det krevja treningssett om lag ti gongar større enn det me har brukt her, og det fører til ein serie tidkrevjande utfordringar. Ei anna openberr utfordring er å utvikla nye steganalyseteknikkar for høgteksturerte bilete.

Analysa vår har gått ut i frå at bileta vert brukte og analyserte enkeltvis. Wendy gjer soleis vurderinga si basert på eitt einskild bilete, og Alice har berre eitt bilete til å skjule det ho har å seia. I satssteganografi får Alice senda ein straum med bilete, og Wendy overvaker straumen og kan aggregera stastiske data over tid. Andrew Ker [7] har vist at både steganografi og steganalyse vert meir komplekse problem i satssteganografi. Han har t.d. vist at Alices optimale strategi ofte vil vera å fordela løynmeldinga ujamnt over dekkbileta. Her kan dekkval fortelja oss kor stor del av meldinga ein skal leggja i kvart dekkbilete.

Det er òg interessant å vurdera nye dekkvalsheuristikkar. Sjølv om dei som me har studert er høveleg gode, har det vore sopass lite forsking på dekkval totalt sett at me

neppe har funne optimale løysingar enno. Resultata våre viser at Alice sannsynlegvis kan identifisera dekkbilete som nesten heilt sikkert vil gå uoppdaga.

Referansar

[1] Patrick Bas, Tomás Filler, and Tomás Pevný. break our steganographic system": The ins and outs of organizing boss. In Filler et al. [3], pages 59–70.

[2] G. Van de Wouwer, P. Scheunders, and D. Van Dyck. Statistical texture characterization from discrete wavelet representations. *IEEE Trans. on Image Proc.*, 8(4), April 1999.

[3] Tomás Filler, Tomás Pevný, Scott Craver, and Andrew D. Ker, editors. *Information Hiding - 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers*, volume 6958 of *Lecture Notes in Computer Science*. Springer, 2011.

[4] Jessica J. Fridrich, Jan Kodovský, Vojtech Holub, and Miroslav Goljan. Steganalysis of content-adaptive steganography in spatial domain. In Filler et al. [3], pages 102–117.

[5] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, January 2006.

[6] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[7] Andrew D. Ker. Batch steganography and pooled steganalysis. In Jan Camenisch, Christian S. Collberg, Neil F. Johnson, and Phil Sallee, editors, *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2006.

[8] Z.Z. Kermani and M. Jamzad. A robust steganography algorithm based on texture similarity using gabor filter. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, pages 578–582, December 2005.

[9] Mehdi Kharrazi, Husrev T. Sencar, and Nasir Memon. Cover selection for steganographic embedding. In *IEEE International Conference on Image Processing*, 2006.

[10] Qingzhong Liu, Andrew H. Sung, Bernardete Ribeiro, Mingzhen Wei, Zhongxue Chen, and Jianyun Xu. Image complexity and feature mining for steganalysis of least significant bit matching steganography. *Inf. Sci.*, 178(1):21–36, 2008.

[11] Tomáš Pevný, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. In *MM&Sec '09: Proceedings of the 11th ACM workshop on Multimedia and security*, pages 75–84, New York, NY, USA, 2009. ACM.

[12] Tomás Pevný, Patrick Bas, and Jessica J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, 2010.

[13] Tomás Pevný, Tomás Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. 2010.

[14] Hedieh Sajedi and M. Jamzad. Cover selection steganography method based on similarity of image blocks. pages 379–384, July 2008.

[15] Hedieh Sajedi and Mansour Jamzad. Secure cover selection steganography. In Jong Hyuk Park, Hsiao-Hwa Chen, Mohammed Atiquzzaman, Changhoon Lee, Tai-Hoon Kim, and Sang-Soo Yeo, editors, *ISA*, volume 5576 of *Lecture Notes in Computer Science*, pages 317–326. Springer, 2009.

[16] Hans Georg Schaathun. *Machine Learning in Image Steganalysis*. Wiley - IEEE. John Wiley & Sons, 2012.

[17] Hans Georg Schaathun. pysteg – a python library for steganography and steganalysis, 2012. `http://www.ifs.schaathun.net/pysteg/`.

[18] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983.

[19] Yifeng Sun and Fenlin Liu. Selecting cover for image steganography by correlation coefficient. *Education Technology and Computer Science, International Workshop on*, 2:159–162, 2010.