

Електротехнички факултет
Универзитет у Бањој Луци
Катедра за општу електротехнику

СЕМЕСТРАЛНИ РАД ИЗ ДИГИТАЛНЕ ОБРАДЕ СЛИКЕ

Примјена ентропије у раздвајању текстура на слици

Студенти:

Дамјановић Нешо
Милановић Ђорђије

УВОД

Појам текстуре и сегментације

Када погледамо неку фотографију, наше око јасно може разазнати одређене објекте на слици, поједине регионе који имају исту боју или исти ниво сивог, те одређене структуре које садрже мање објекте који се понављају на региону (нпр. плочице у купатилу и сл.). Овај последњи примјер јако добро илуструје појам једне текстуре.

Иако не постоји формална дефиниција, текстуру можемо схватити као понављање одређених узорака или узорка на неком региону. Узорак може да се понавља егзактно, или као сет варијација на тему зависно од неке функције или позиције. Други, исто тако битни, аспекти текстура који не смију бити запостављени су: величина, облик, боја и оријентација узорака (текстона). Понекад је разлика између двије текстуре садржана само у степену варијација или у статистичкој расподјели сродних текстона. Наравно, и схватање текстура може да буде различито, зависно од региона од интереса. На примјер, скуп свих фотографија на једном листу може се протумачити као једна текстура или свака од њих засебно као посебне текстуре.

Неки од примјера природних и вјештачких текстура приказани су на сљедећој слици:



Регион са одређеном текстуром неминовно мора бити довољно велики да би могао представити своју периодичну структуру. Ако је регион мали, или ако су његови текстони малени дана одређени начин не могу бити детектовани, онда се његова текстура и не може приказати у правом облику. Из овога се јасно назире проблем скале, који уноси додатне проблеме у, и онако компликовано рјешавање проблема текстура. Заиста, једна иста текстура у двије одвојене скале ће бити приказана као различите текстуре, ако обезбиједимо да су скале довољно различите.

Што скале постају ближе, текстуре ће се све теже разликовати, док се на крају не стопе у једну јединствену текстуру.

У овом раду ћемо се позабавити текстурама у циљу проучавања могућности сегментације региона више него самим особинама текстура уопште. Наравно, јасно је да ћемо, бар дјелимично, морати да се упознамо са неким особинама и величинама које на најбољи начин описују неку текстуру. Овдје ћемо покушати раздвојити који регион на одређеној слици има текстуру А, а који текстуру Б, те да направимо што је могуће јаснију границу између њих. Резултат би требао бити слика, или слике које у неким даљим примјенама могу послужити као маске, на којој ће регион текстуре А бити обојен једном бојом (рецимо црном), а регион текстуре Б другом (бијелом). Битно је напоменути да је граница између региона са различитим текстурама често нејасна и на оригиналној слици и да је више функција интуитивна него математичка.

Из свега претходно наведеног да се закључити да је мало вјероватно да постоји било какав једноставан поступак или операција која може дати добре резултате у сегментацији региона са различитим текстурама у свим случајевима. С друге стране, могуће је да нека комбинација операција може дати разумно добру сегментацију за широки опсег текстура.

ОСНОВНИ ПАРАМЕТРИ ТЕКСТУРА

Средњи ниво сивог и стандардна девијација

Регион одређене текстуре који посједује широк дијапазон нивоа сивог мора имати нека својства на основу којих би их људски визуелни систем могао опазити. С обзиром на то да је скуп свих текстура бесконачан немогуће је дати некакав алгоритам на основу кога би се сваки регион са неком текстуром поредио са унапријед познатим текстурама из библиотеке и на тај начин вршило раздвајање региона. Много је вјероватније да сличности и разлике које су видљиве голим оком, те да се на тај начин интуитивно могу предвидјети поступци или операције које ће разграничити да ли су текстурни региони различити или не.

Један очигледан начин да се раздвоје региони јесте да се прикажу различитим бојама или различитим нивоом сивог. Код сегментације региона црно-бијелих слика сваки пиксел у једном текстурном региону може бити средњи ниво сивог на неком мањем региону унутар постојеће текстуре. Овај регион, кога ћемо убудуће звати прозором, може варирати у својој величини зависно од скале која је потребна да би се текстони неке текстуре могли уочити и бити правилно репрезентовани.

Употреба облика прозора је произвољна све док је у интересу сама текстура, а не поједини пиксели. Као резултат, граница између текстурних региона може бити одређена у дужини од W пиксела, гдје је W ширина прозора.

Поступак употребе средњег нивоа сивог састоји се у превлачењу прозора $W \times W$ преко оригиналне слике тако што ће централном пикселу који се види кроз прозор додијелити средњу вриједност свих пиксела обухваћених прозором. Затим се погодним одабиром прага врши подјела слике на два различито обојена региона. Из поступка се јасно уочава да се употреба средњег нивоа сивог може примијенити само на слике које има два различита текстурна региона и то са видљивом разликом у нивоу сивог.

Употреба стандардне девијације примјеном малих прозора даје знатно боље резултате. То би у суштини значило да већина корисне информације лежи у промјенама интензитета пиксела, а не у вриједностима самих пиксела понаособ. Стандардна девијација у малом региону нам даје корисне информације о томе колико пиксела тог региона припада текстурним, а колико позадини.

И средњи ниво сивог и стандардна девијација познати су као статистички моменти. Усредњавање је повезано са моментом првог реда, док је стандардна девијација у директној вези са моментом другог реда. Такође се могу дефинисати и моменти вишег реда (закренутост, куртозис итд). Математички, момент n -тог реда дат је са:

$$M_n = \frac{\sum (x - \bar{x})^n}{N}, \text{ гдје је } n \text{ ред момента, а } N \text{ број одмјерака.}$$

Закренутост (нагнутост на једну страну) се дефинише као:

$$S_k = \frac{1}{N} \sum \left(\frac{x - \bar{x}}{\sigma} \right)^3, \text{ а куртозис као } Kurt = \frac{1}{N} \sum \left(\frac{x - \bar{x}}{\sigma} \right)^4 - 3.$$

И ови моменти се могу употријебити на сличан начин, с назнаком да се преферирају нешто већи прозори.

Расподјеле нивоа сивог

Средња вриједност нивоа сивог, као ни стандардна девијација јесу једноставне за оперисање, али не носе било какве информације о понављању природе одређене текстуре.

Матрица расподјеле нивоа сивог (ГЛЦМ) садржи информације о позицијама пиксела који имају сличне вриједности нивоа сивог. Идеја се састоји у томе да се слика скенира тако да се задржи информација о томе колико се често понављају пиксели чија се вриједност разликује за Δz на одређеном фиксном растојању d у једној линији.

Природно је да је и правац кретања између два пиксела такође од интереса што изискује конструисање вишеструких матрица, по једну за сваки правац кретања од интереса. Обично су од интереса четири правца: хоризонтални, вертикални, те двије дијагонале. Стога, за сваку вриједност d имамо 4 матрице, сваку димензија 256×256 , ако оригинална слика има 256 нивоа сивог. То проузрокује огроман број података, често знатно више него што их посједује оригинална слика. Уобичајен поступак се састоји у анализи тих матрица и рачунању неколико једноставних бројних вриједности које садрже вриједне информације. Те бројне вриједности се често називају дескрипторима, и њих пет ће бити приказано у овом раду, а на могућностима употребе једне од њих ћемо се посебно задржати.

Ако број нивоа сивог расте фактором 2, димензије GLCM матрица расту фактором 4. То значи да веома брзо постаје тешко употребљавати матрице директно. Зато ћемо их радије карактеризовати низом нумеричких дескриптора него да рачунамо вриједности матрица директно. Средња и стандардна девијација могу бити употребљене као дескриптори за информације на слици, али исто тако могу се употријебити једнако добро и да опишу ГЛЦМ матрице. Међутим, многе дескрипторе треба испробати, јер неки раде боље од других:

Максимална вјероватноћа

Представља једноставно највећу вриједност у матрици и у директној је вези са највећом осјетљивошћу или правцем највеће промјене. Она може бити највећа вриједност у било којој од матрица или максимум по свакој од њих; уствари, садржи корисну информацију у одређивању која од матрица садржи максималну вриједност, пошто индукује потребни правац у којем ће текстура бити испитивана.

Моменти

Момент диференција степена k може се дефинисати са:

$$Mom_k = \sum_i \sum_j (i - j)^k M[i, j]$$

Овај дескриптор има мале вриједности у случају када највећи елементи матрице M припадају главној дијагонали; ово је био случај када смо анализирали слику 4.1а. Супротан ефекат може бити постигнут употребом инверзног момента, који се израчунава по формули:

$$Mom_k^{-1} = \sum_i \sum_j \frac{M[i, j]}{(i - j)^k}, i \neq j.$$

Контраст

Процјена контраста дата је сљедећим изразом:

$$C(k, n) = \sum_i \sum_j |i - j|^k M[i, j]^n$$

Када је $k=n=1$ израчунава се очекивана вриједност разлике између два пиксела.

Хомогеност

Овај дескриптор је дат са:

$$G = \sum_i \sum_j \frac{M[i, j]}{1 + |i - j|}$$

Мала вриједност G значи да велике вриједности M леже близу главне дијагонале. G је врло сличан Mom^{-1} .

Ентропија

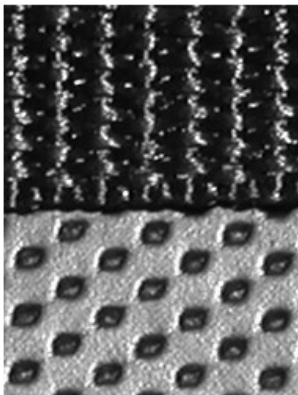
Ентропија се рачуна по следећој формули:

$$H = - \sum_i \sum_j M[i, j] \log(M[i, j])$$

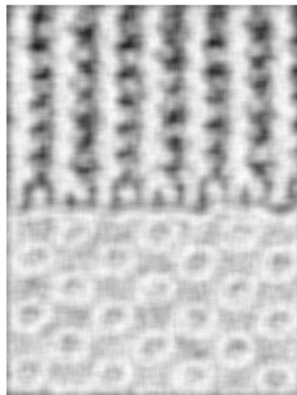
Она представља мјеру информације која је садржана у M . Велики број празних мјеста носе мало информација, док збркани региони носе велику количину информација.

ПРИМЈЕНА ЕНТРОПИЈЕ У СЕГМЕНТАЦИЈИ ТЕКСТУРНИХ РЕГИОНА

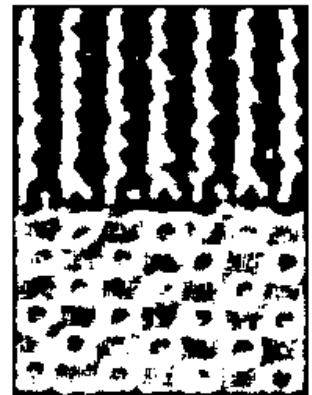
Један примјер употребе ентропије у сегментацији текстурних региона дат је у демосу напреднијих MATLAB-а. Задржаћемо се укратко и на том примјеру и дати основне идеје у сегментацији региона од интереса.



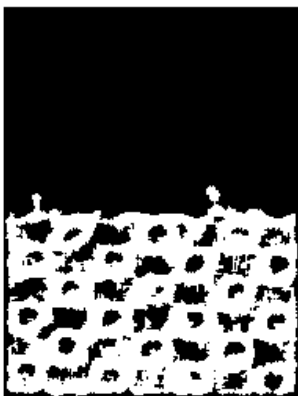
Слика1



Слика2



Слика3



Слика4



Слика5



Слика6

На слици 2 приказана је оригинална слика нарушена ентропијом и на резултујућој слици се види да су истакнути детаљи гдје су јаче изражени прелази са свијетлог на тамно и обратно (тзв. збркани региони). То се постиже функцијом „entropyfilt“ која је имплементирана у новијим верзијама MATLAB-а због све веће примјене у сегментацији. Тако су добијене израженије „шаре“, које су битне за природу саме структуре. Претварањем слике у бинарну, узимањем интуитивног прага 0.8, једноставно се добија слика 3. Истицањем отворених објеката, те њиховим затварањем и попуњавањем добијају се слике 4, 5 и 6 респективно. Последња слика представља сегментован горњи регион на слици са текстуром која се може доста јасно издвојити.

Могућности ентропије у издвајању текстуалних региона

Примјенићемо сличан поступак и код слика које су настале скенирањем текстуалних докумената. За различите примјене, показало се да је битно испитати одређене регионе од интереса на скенираном документу. У првом реду се издвајају маргине, фотографије и текст боксови као региони од нарочитог интереса. Проблем издвајања региона у којима се налази текст може се посматрати као скуп региона са сличном или скоро истом, додуше јако нехомогеном, текстуром. Јавља се такође и проблем скале, гдје ће сигурно текстуални блокови са различитом величином фонта бити протумачени као различите текстуре.

Наш задатак, оvdје, ће бити да испитамо могућности ентропије за издвајање текстуалних региона. Идеја се састоји у сљедећем:

- једноставним програмчићем можемо издвојити маргине на слици. Идеја ће се састојати у томе да ћемо израчунавати суму вриједности по маргинским врстама и колонама, те да ћемо претпоставити да се ту и тамо појави понеки црни пиксел који ће настати као резултат слабог скенирања или евентуално лоше одређеног прага. Тај програм ради добро на свим испитиваним сликама и дат је под насловом „MARGINE“.
- примјена ентропије састоји се у одређивању текстура фотографија на слици. Израчунавање прага смо аутоматизовали Otsuovim алгоритмом, јер интуитивно израчунавање прага за сваку фотографију посебно нема смисла, мада понекад даје боље резултате. Такође смо користили истицање отворених објеката, затварање и попуњавање коришћењем имплементираних MATLAB-ових функција *bwareaopen*, *imclose* и *imfill* као што је и приказано у коду. Димензије кернела (у наредби *true*), те вриједност 2000, узете су интуитивно.
- на крају тако добијену слику можемо искористити као маску да из оригиналне слике излучимо само текст.

MARGININE

```
clc, clear all, close all
```

```
%Pretvaranje slike u binarnu i izracunavanje dimenzija
```

```
I = imread('Slika1.jpg'); imshow(I)
I = mat2gray(I);
prag = graythresh(I);
Bin = im2bw(I, prag); figure, imshow(Bin);
[m,n] = size(Bin);
B = zeros(m,n);
C = zeros(m,n);
```

```
%ODREDJIVANJE MARGINA
```

```
%Odredjivanje desne margine
```

```
for j=n:-1:1
if sum(Bin(:,j))>m-3
    for i=1:m
        B(i,j)=1;
    end
else break
end
end
```

```
%Odredjivanje lijeve margine
```

```
for j=1:n
if sum(Bin(:,j))>m-3
    for i=1:m
        B(i,j)=1;
    end
else break
end
end
figure, imshow(B)
```

```
%Odredjivanje donje margine
```

```
for i=m:-1:1
if sum(Bin(i,:))>n-3
    for j=1:n
        C(i,j)=1;
    end
else break
end
end
```

```
%Odredjivanje gornje margine
```

```
for i=1:m
if sum(Bin(i,:))>n-3
    for j=1:n
        C(i,j)=1;
    end
else break
end
end
```

```
figure, imshow(C)
M=B+C;
figure, imshow(M)
```



```
clc, clear all, close all,
```

```
%Entropija slike i njeno pretvaranje u binarnu Otsuovim pragom
```

```
I = imread('slika2.jpg');
I = rgb2gray(I); figure, imshow(I)
I = im2double(I);
I = mat2gray(I);
J = entropyfilt(I); figure, imshow(J)
J1 = mat2gray(J); figure, imshow(J1);
prag = graythresh(J1);
Bin = im2bw(J1, prag); figure, imshow(Bin);
```

```
% Isticanje otvorenih objekata
```

```
Bin1 = bwareaopen(Bin,5000); figure, imshow(Bin1);
```

```
% zatvaranje i popunjavanje
```

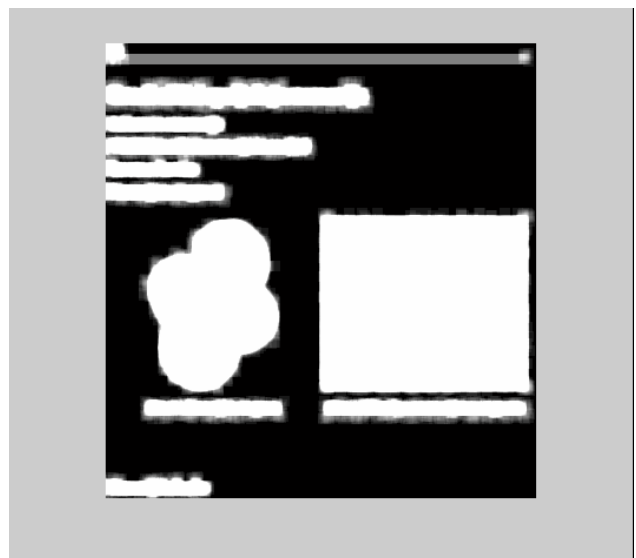
```
nhood = true(9);
closeBin1 = imclose(Bin1,nhood); figure, imshow(closeBin1)
roughMask = imfill(closeBin1,'holes'); figure, imshow(roughMask);
```

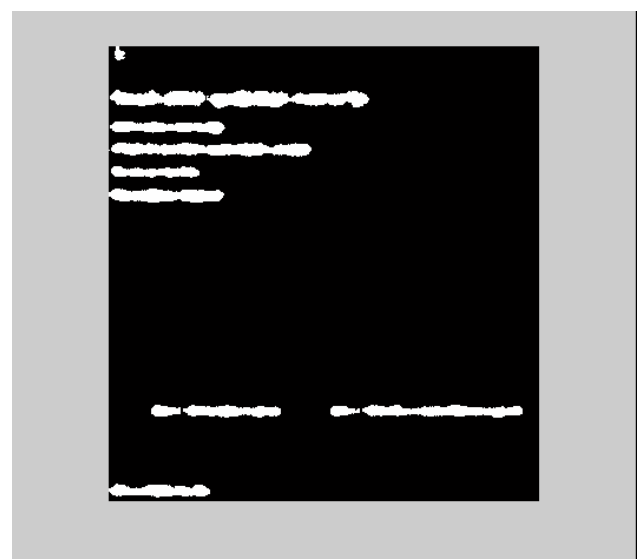
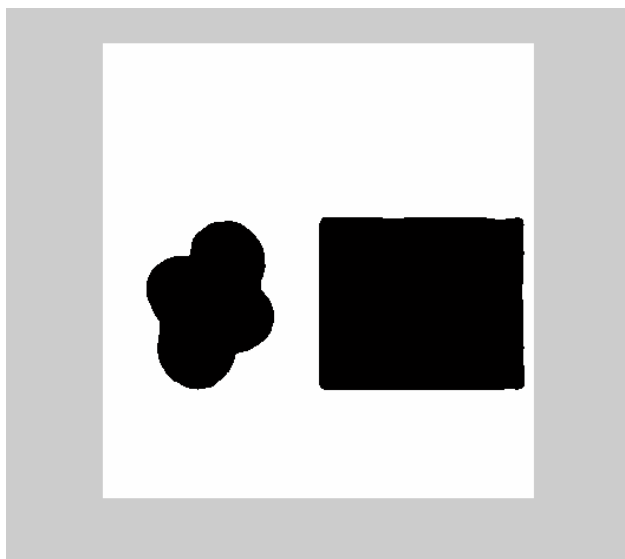
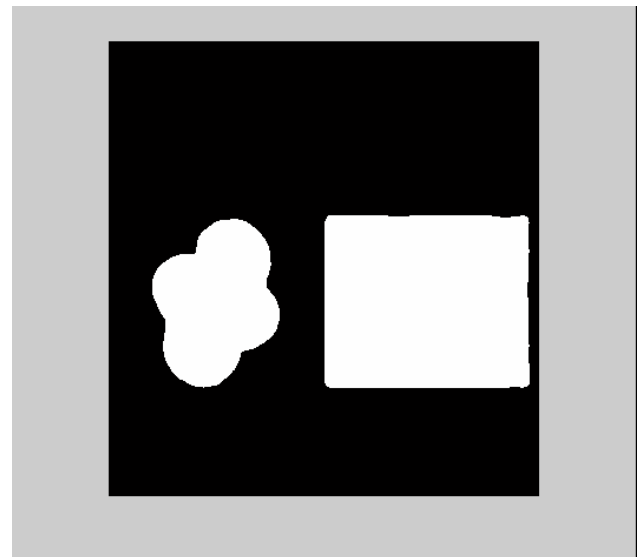
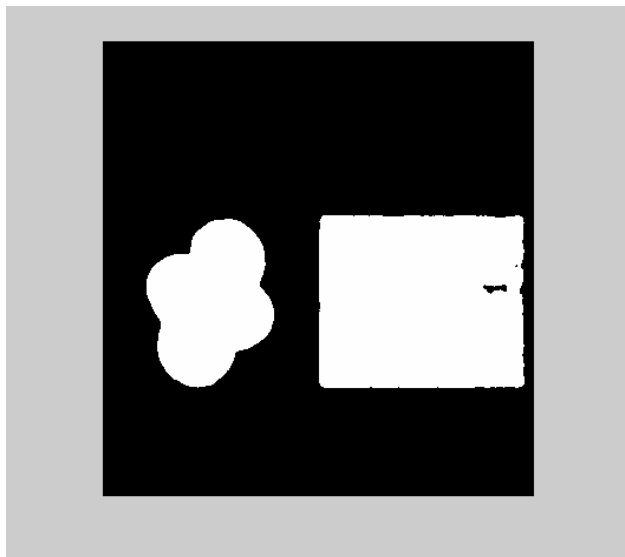
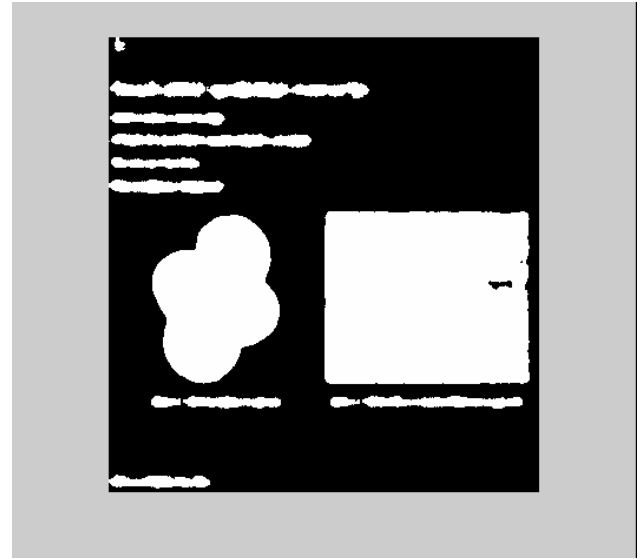
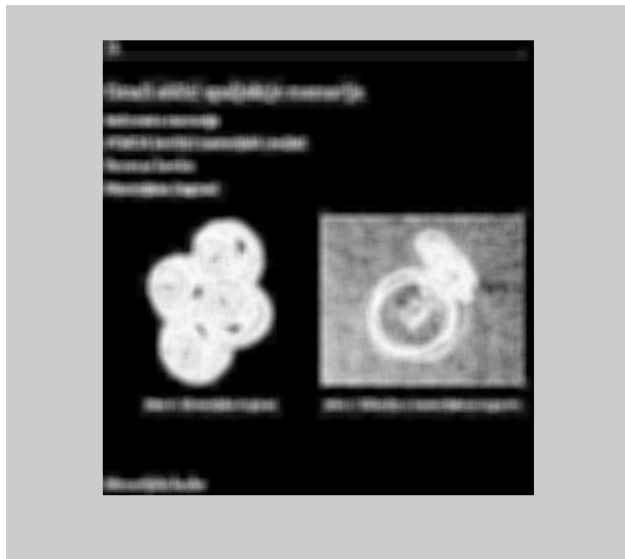
```
% kreiranje maske za izdvajanje teksta
```

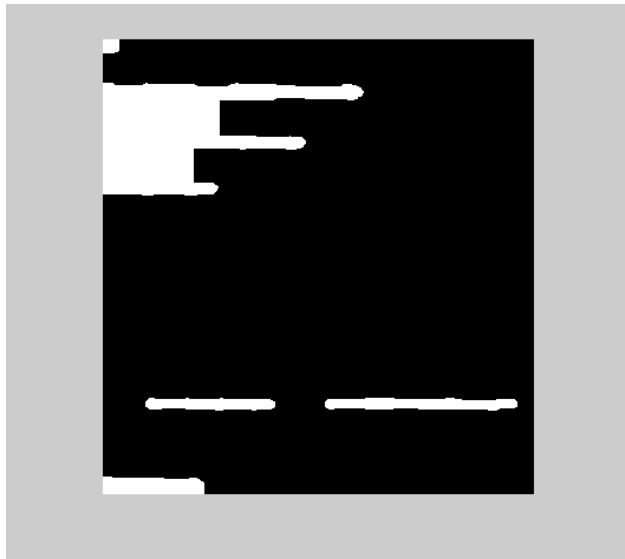
```
Maska = 1 - roughMask; figure, imshow(Maska)
Tekst = Bin .* Maska; figure, imshow(Tekst)
se = strel('square',20);
OPEN = imclose(Tekst,se); figure, imshow(OPEN)
OPEN = im2double(OPEN);
```

```
% Finalna slika
```

```
I = 1 - I;
Fin = I .* OPEN; figure, imshow(1-Fin)
```

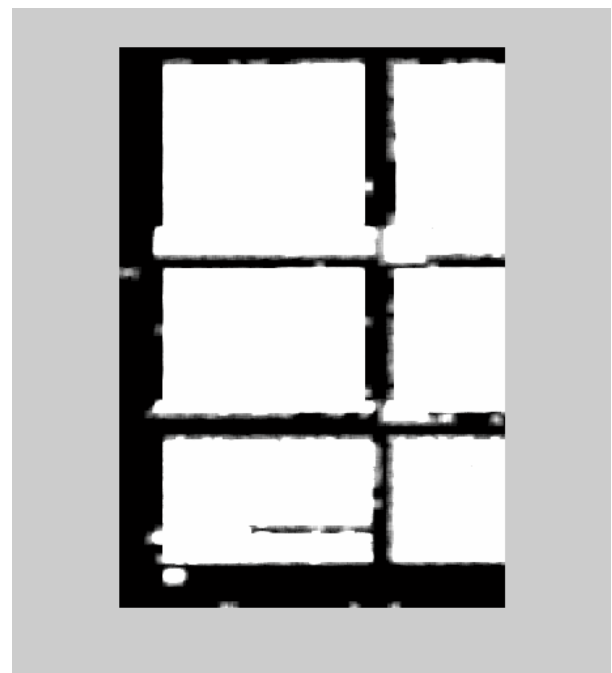


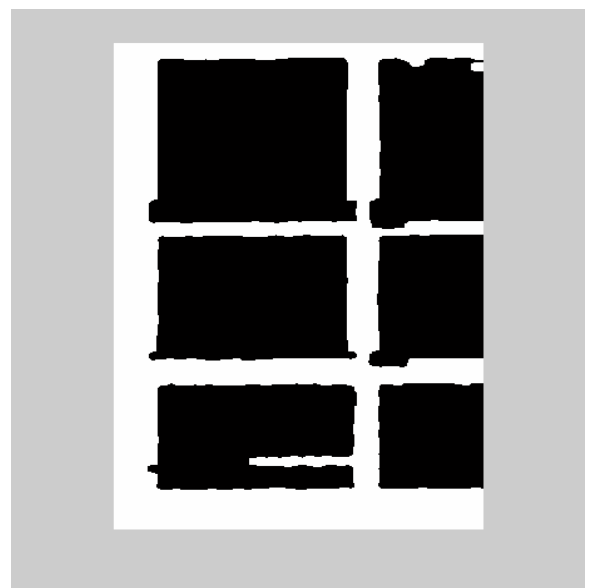
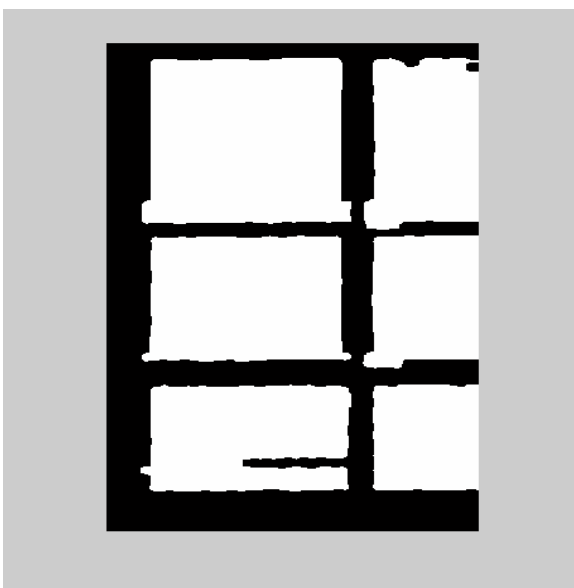
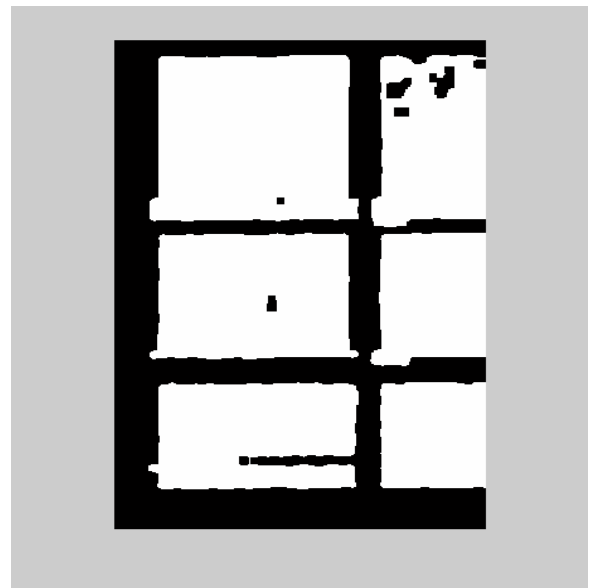




Из претходних слика видимо да поступак заснован на ентропији може дати добре резултате. Међутим, када је текст ситан и проред између слова мали, ентропијом само добијамо додатно замућење које ће као исход сигурно дати препознавање текст боксова као слике. У томе и лежи главни недостатак ентропије као поступка при одређивању тексуалних текстура.

Један примјер на коме поступак ентропије даје слабе резултате илустрован је на сљедећим сликама. Код, наравно, остаје потпуно исти, само је слика која се испитује другачија. Такође је могуће манипулисати праговима и величинама прозора, али резултати и даље остају лоши.





Са последње слике се јасно може видјети да су текстуални боксови груписани уз слике, не могу се одвојити неким промјенама у програму, што наводи на закључак да ентропија остаје немоћна код скенираних докумената гдје је текст писан ситним фонтом.

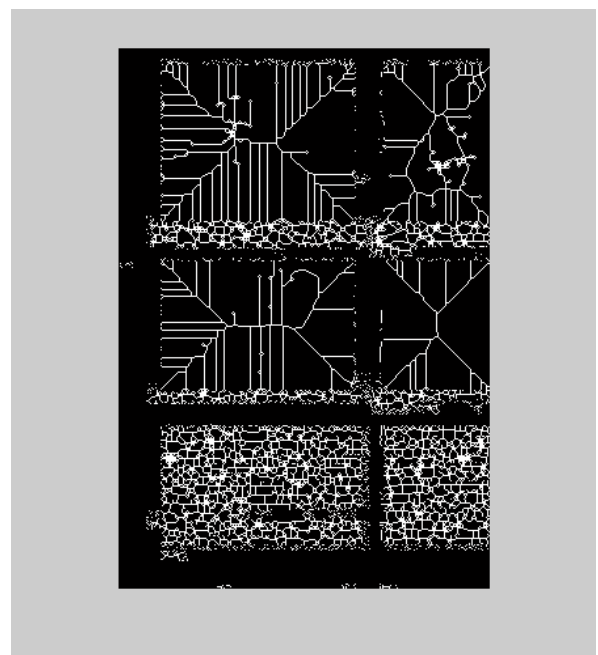
Даћемо, на крају, још и правац у коме смо отишли док смо завршавали наш рад. Чини се да се добри резултати у одређивању и лоцирању тексура текста могу добити примјеном скелетона. Као што ће се у наредним примјерима видјети, сви боксови са текстом имају карактеристичну мрежасту текстуру и доста се лако могу разабрати од текста користећи једноставне морфолошке операције на скелетонима.

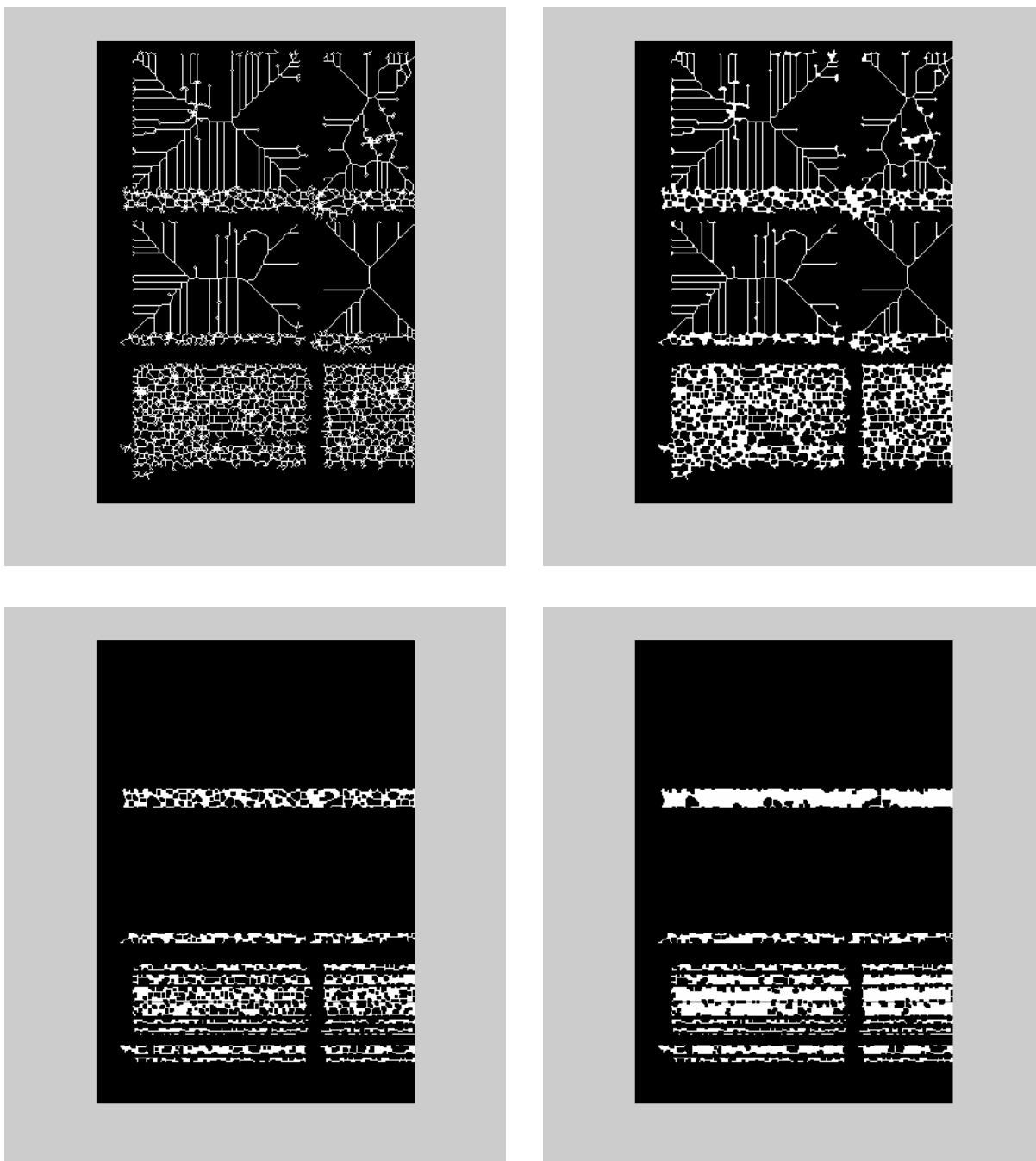
```

clc, clear all, close all,
I = imread('petal.jpg');
I = rgb2gray(I); figure, imshow(I)

I = im2double(I);
I = mat2gray(I);
I=1-I;
I = bwmorph(I,'skel', Inf); figure, imshow(I)
Bin1 = bwareaopen(I,30); figure, imshow(Bin1);
nhood = true(3);
skel = imclose(Bin1,nhood); figure, imshow(skel)
[m, n]= size(skel);
for i=1:m
if sum(skel(i,:))<n/3
for j=1:n
skel(i,j)=0;
end
end
end
end
figure, imshow(skel)
roughMask = imfill(skel,'holes'); figure, imshow(roughMask)

```





Последње двије слике могу дати корисне информације о регионима у којима се налази текст.

Напоменућемо још на крају да смо се нешто дубље упустили и у анализу изложеног проблема помоћу других параметара, али због опширности нисмо сада давали никакве резултате. У сваком случају остаје да се овим проблемом позабавимо и у будућности тражећи општи алгоритам који ће се моћи примјењивати на сваком скенираном текстуалном документу са једнако добрим резултатима.