

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET

PREPOZNAVANJE OBLIKA I OBRADA SLIKE

Projektni Zadatak br. 2

Studenti:

MUFTIĆ *Belma*, 1423/17260

LEMEŠ *Lamija*, 1474/17070

KRUPALIJA *Ehlimana*, 1431/17461

Odgovorni asistent:

MoE SUMEJJA PORČA

December, 2018

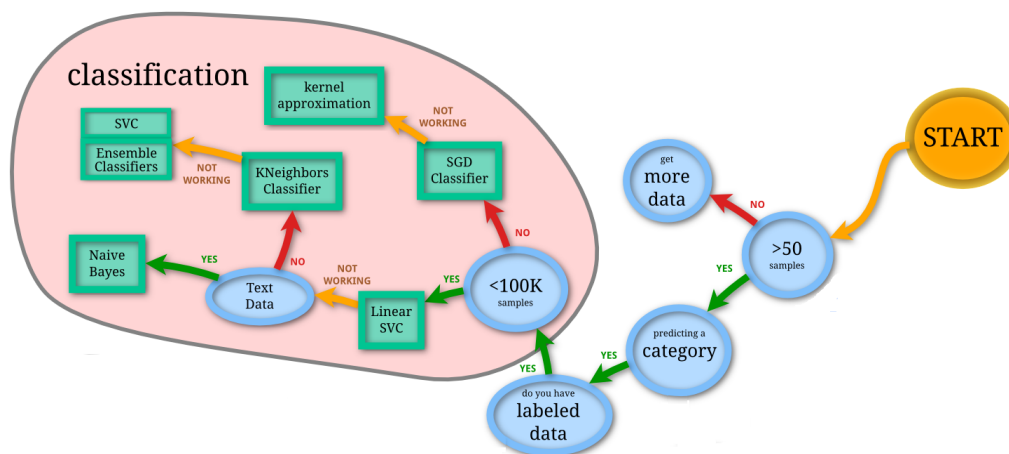
Sadržaj

1	Izbor modela za prepoznavanje	1
2	Izbor i kreiranje deskriptora	2
3	Izbor metoda poboljšanja	3
3.1	Poboljšavanje kontrasta	3
3.2	Poboljšavanje osvjetljenja	4
3.3	Ujednačavanje histograma	5
4	Testiranje modela	6
5	Poboljšavanje performansi modela za prepoznavanje	7
5.1	Izmjena parametara modela	7
5.2	Drukčija podjela podataka	7
5.3	Izbacivanje <i>outlier</i> -a slika	7

1 Izbor modela za prepoznavanje

Postoji šest najčešće korištenih klasifikacijskih modela (što je prikazano na Slici 1.):

1. SGD (*Stochastic Gradient Descent*) klasifikacija;
2. *Kernel Approximation* klasifikacija;
3. Linearna SVC (*Support Vector Classification*);
4. SVC (*Support Vector Classification*);
5. KNN (*K-Nearest Neighbors*) klasifikacija;
6. *Naive Bayes* klasifikacija.



Slika 1: Klasifikacijski modeli i vršenje njihovog odabira

Budući da *dataset* ima manje od 100,000 slika, te postoje tekstualni podaci o različitim kategorijama i ROI, koristiti će se ***Naive Bayes*** klasifikacijski model.

2 Izbor i kreiranje deskriptora

Odabran je **HuMoments** deskriptor, kako on opisuje oblik željenog objekta, a prepoznavanje ćelije se oslanja prvenstveno na njen oblik. U svrhu toga je kreirana skripta `maskiranje.py` koja izloira samu ćeliju (dakle bez pozadine), jer se *HuMoments* oslanja na analizu slika sa jednim kanalom.

Maskiranje je zasnovano na izdvajanju piksela koji spadaju u određeni rang vrijednosti, nakon čega se radi otvaranje da bi se uklonili "zalutali" pikseli, te se radi i dilatacija da bi se uključile i granice ćelije koje prethodno nisu bile u zadatom rasponu. Nakon maskiranja je pozvana funkcija **HuMoments** koja vraća niz brojeva koji opisuju oblik prethodno dobivene binarne slike. Nakon normalizacije niza, vrijednosti se zapisuju u CSV datoteku, te ako su u pitanju *Train* slike, upisuje se i kojoj klasi slike pripadaju.

Navedene funkcionalnosti implementirane su na sljedeći način:

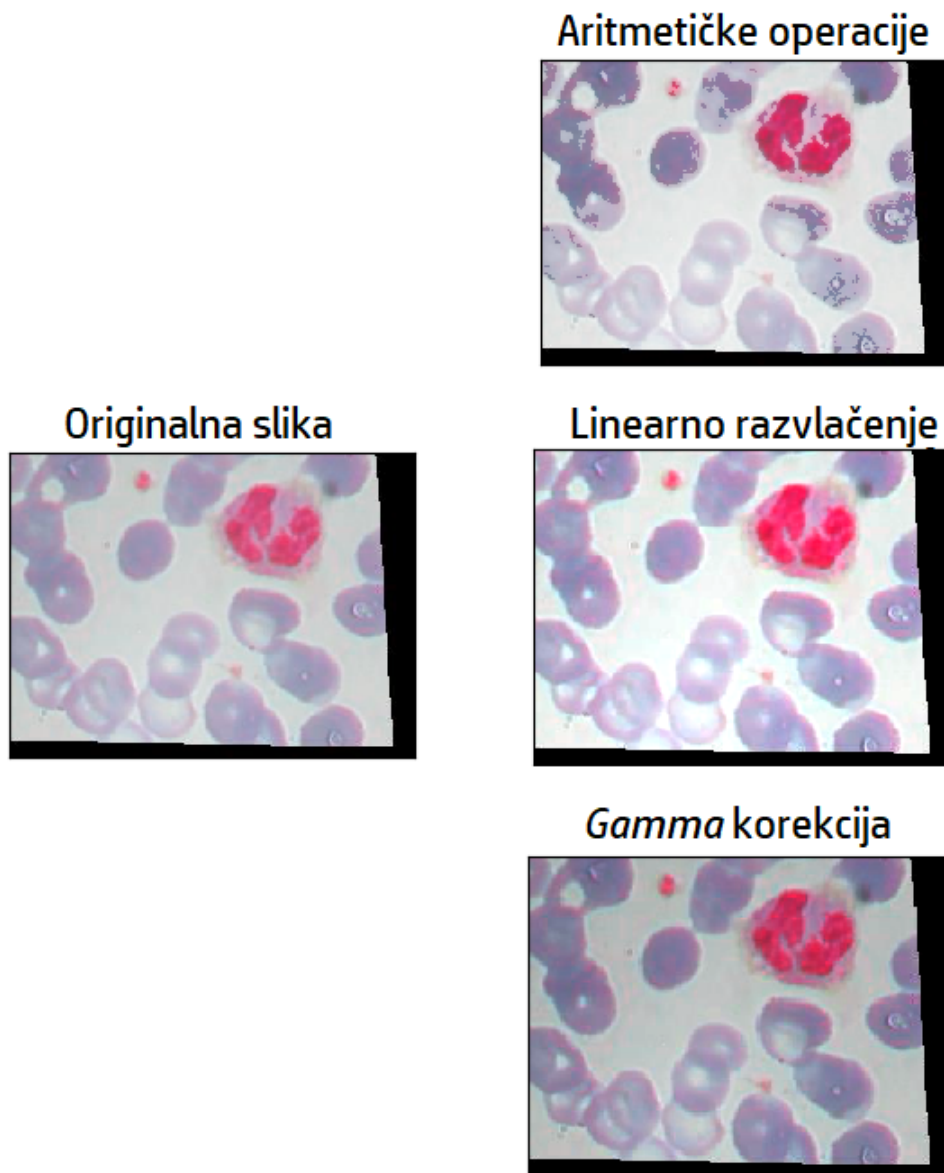
```
for i in range(0, 90):
    try:
        Slika = cv2.imread('./Train/CroppedROI{}.jpg'
                           .format(i + 1))
        B = cv2.inRange(Slika, lowerR, upperR)
        if (i > 29):
            klasa = 2
        if (i > 59):
            klasa = 3
        print("{} , {} \n".format(i, klasa))
        shape = cv2.getStructuringElement
                     (cv2.MORPH_ELLIPSE, (2, 2))
        nm = cv2.morphologyEx(B, cv2.MORPH_OPEN, shape)
        shape2 = cv2.getStructuringElement
                     (cv2.MORPH_ELLIPSE, (3, 5))
        NewMask = cv2.dilate(nm, shape2)
        B = NewMask
        huMoments = cv2.HuMoments(cv2.moments(B))
                           .flatten()

        for i in range(0, 7):
            huMoments[i] = -1 * m.copysign(1.0,
            huMoments[i]) * m.log10(abs(huMoments[i]))
        f = open("./DeskriptorTrain.csv", "a")
        f.write("{} , {} , {} , {} , {} , {} , {} , {} \n"
        .format(huMoments[0], huMoments[1], huMoments[2],
        huMoments[3], huMoments[4], huMoments[5],
        huMoments[6], klasa))
```

3 Izbor metoda poboljšanja

3.1 Poboljšavanje kontrasta

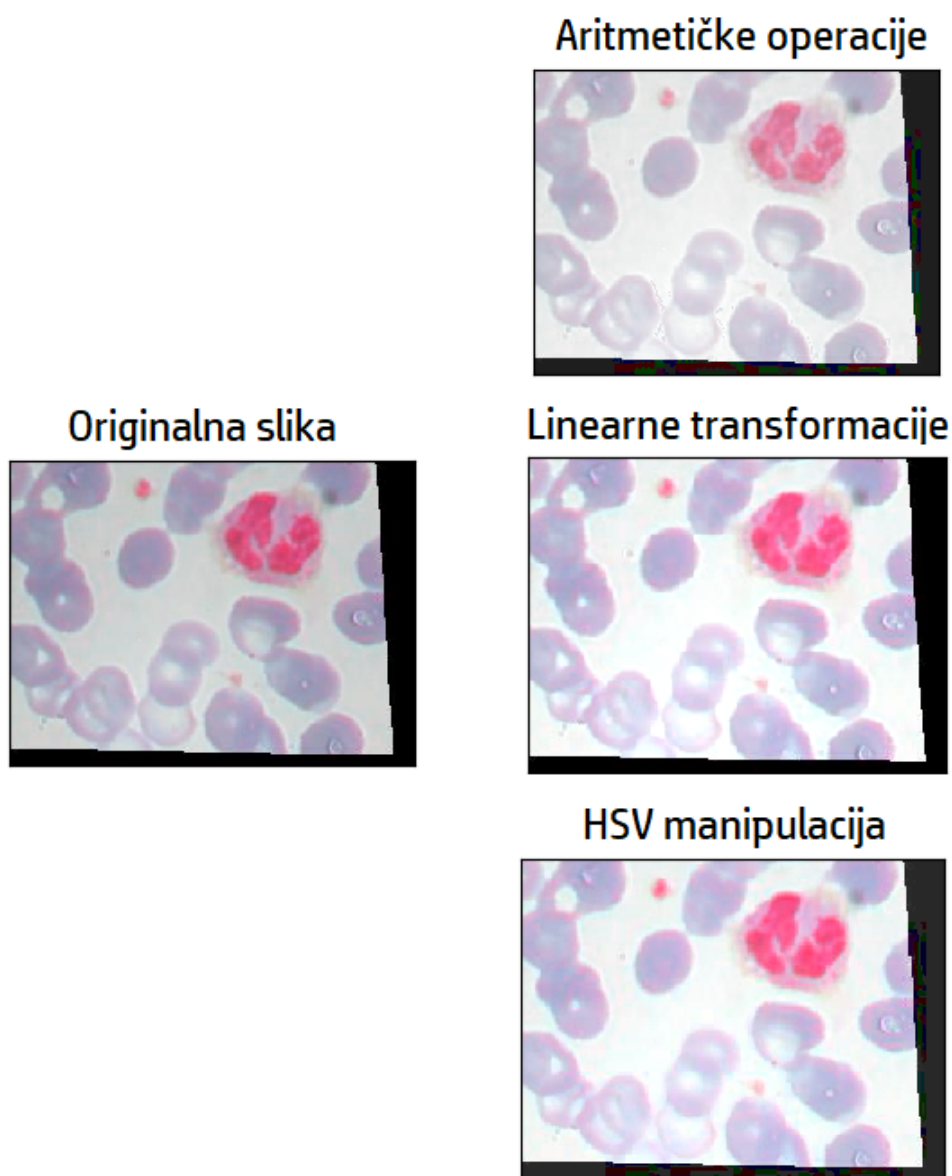
Od tri implementirane metode, najbolje rezultate pokazala je metoda **linearnog razvlačenja kontrasta**. Metoda aritmetičkih operacija previše je narušila strukturu slike, dok metoda *gamma* korekcije nije izvršila dovoljnu modifikaciju kontrasta. Iz ovog razloga metoda linearnog razvlačenja kontrasta koristiti će se za poboljšavanje slika prije njihove dalje obrade.



Slika 2: Rezultati korištenja različitih metoda za poboljšavanje kontrasta

3.2 Poboljšavanje osvjetljenja

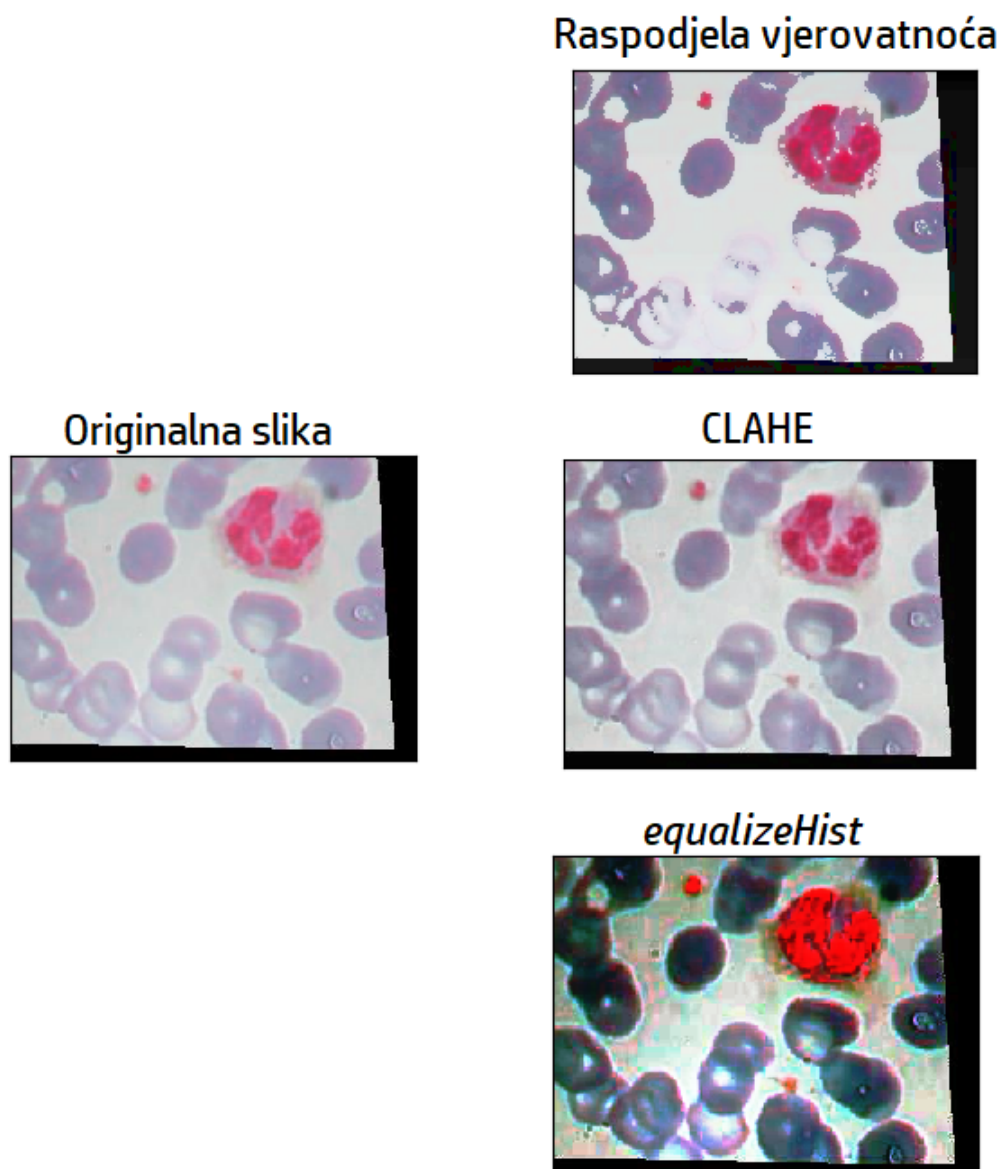
Od tri implementirane metode, najbolje rezultate pokazala je metoda **linearnih transformacija**. Metoda aritmetičkih operacija nije izvršila dovoljnu promjenu osvjetljenja, dok je metoda manipulacije HSV slikom izvršile preveliko povećanje osvjetljenja, te će se iz ovog razloga metoda linearnih transformacija koristiti za poboljšavanje slika prije njihove dalje obrade.



Slika 3: Rezultati korištenja različitih metoda za poboljšavanje osvjetljenja

3.3 Ujednačavanje histograma

Od tri implementirane metode, najbolje rezultate pokazala je metoda **CLAHE**. Metoda raspodjele vjerovatnoća previše je degradirala strukturu slike (pri čemu su neke ćelije potpuno nestale sa slike, što je nedopustivo), dok je metoda `equalizeHist` degradirala strukturu slike u pogledu boje. Iz ovog razloga metoda CLAHE koristiti će se za poboljšavanje slika prije njihove dalje obrade.



Slika 4: Rezultati korištenja različitih metoda za ujednačavanje histograma

4 Testiranje modela

Kreirani deskriptori koriste se za treniranje i testiranje modela, pri čemu veoma važan aspekt predstavlja varijabla `confusionMatrix`, koja sadrži sve relevantne informacije o performansama modela. Na Slici 5. vidljivo je da je tačnost modela oko **76 %**, te da najveću specifičnost pokazuje **klasa 1**, dok najveću senzitivnost pokazuje **klasa 2**. Najmanje vrijednosti specifičnosti i senzitivnosti ima **klasa 3**.

```
Accuracy sa accuracy_score za citav model: 0.7619047619047619
```

```
Confusion matrix:
```

```
[5, 0, 2]
```

```
[0, 7, 0]
```

```
[0, 3, 4]
```

```
Klasa 1
```

```
Acc: 0.7619047619047619
```

```
Sen: 0.7142857142857143
```

```
Spec: 1.0
```

```
Klasa 2
```

```
Acc: 0.7619047619047619
```

```
Sen: 1.0
```

```
Spec: 0.75
```

```
Klasa 3
```

```
Acc: 0.7619047619047619
```

```
Sen: 0.5714285714285714
```

Slika 5: Rezultati pokretanja modela za prepoznavanje

5 Poboljšavanje performansi modela za prepoznavanje

5.1 Izmjena parametara modela

5.2 Drukčija podjela podataka

5.3 Izbacivanje *outlier*-a slika