

Universitatea Tehnica din Cluj-Napoca
Facultatea de Automatica si Calculatoare
Departamentul Calculatoare

PROCESAREA DE IMAGINI

Identificarea pieselor de pe tabla de șah

Profesor îndrumător: Răzvan Itu

Sorina - Rodica Iojă
Grupa 302310

Contents

Introducere	2
Descrierea problemei abordate.....	2
Metoda propusa.....	2
Rezultate experimentale.....	4
Concluzii	7

Introducere

Descrierea problemei abordate

Data fiind o imagine a unei table de șah se cere identificarea pieselor și printarea lor într-un format standard. [Given an image of a board from above and the template images of the pieces determine the position of each piece and output the table configuration in a standard format]. Cerinta problemei poate fi consultata [aici](#).

Metoda propusa

Vom stoca piesele posibile intr-un vector de matrice, imaginile au fost descarcate de [aici](#). După cum se poate observa acestea au o dimensiune de 60 x 60 de pixeli, asadar vom folosi o tabla de 480 x 480 de pixeli.

Dupa deschiderea imaginii (tablei de sah) vom decupa pe rand fiecare patrat al tablei de sah utilizand coordonatele pixelului din stanga sus. Obtinem astfel un patrat de 60x60 de pe o anumita pozitie.

```
Mat croppedMat(int x, int y, Mat image) {
    int height = image.cols;
    int width = image.rows;

    Mat cropped = Mat(60, 60, CV_8UC1);
    int xx = x + 60;
    int yy = y + 60;

    int k = 0;
    int l = 0;
    for (int i = x; i < xx; i++) {
        for (int j = y; j < yy; j++) {
            //printf("%d  %d\n", i, j);
            cropped.at<uchar>(k, l) = image.at<uchar>(i, j);
            l++;
        }
        k++;
        l = 0;
    }
    return cropped;
}
```

Urmatorul pas este acela de a compara imaginea obtinuta prin decupare cu piesele din memorie, dar mai intai ne vom asigura ca in locul respective exista o piesa, daca spatiul e gol vom printa spatiu alb sau spatiu negru in functie de acesta. Pentru a determina daca un patrat e alb sau negru (culoarea tablei) vom folosi urmatoarea metoda. Ne folosim de o norma (adunam diferentele la patrat) si la final daca este sun un anumit prag atunci intregul patrat are culoarea alb/ culoarea tablei.

```

bool black(Mat img) {
    int s = 0;
    for (int i = 0; i < img.rows; i++) {
        for (int j = 0; j < img.cols; j++) {

            Vec3b pixel = img.at<uchar>(i, j);
            //valore obtinuta din mai multe rulari si printari
            s += (pixel[0] - 62) * (pixel[0] - 62);

        }
    }

    if (s < 720)
        return true;
    return false;
}

```

Daca in imaginea decupata avem o piesa atunci pentru a obtine id-ul acesteia parcurgem vectorul de piese din memorie calculand numarul de pixeli comuni pentru fiecare. Vom returna indicele piesei care are numarul de pixeli comuni mai mare.

```

int getChessPieceId(int x, int y, Mat image) {
    int ind=0;
    int max_value = 0;
    int value = 0;

    vector<Mat> piese_vect = piese();

    Mat croppedImage = croppedMat(x,y,image);

    if (white(croppedImage))
        return 12;
    if (black(croppedImage))
        return 13;

    for (int i = 0; i < piese_vect.size(); i++)
    {
        //printf("scor: %d cu %d\n", value, i);

        int value = comparatie(croppedImage, piese_vect[i]);
        //printf("scor: %d cu %d\n", value, i);
        if (value > max_value) {
            max_value = value;
            ind = i;
        }
    }

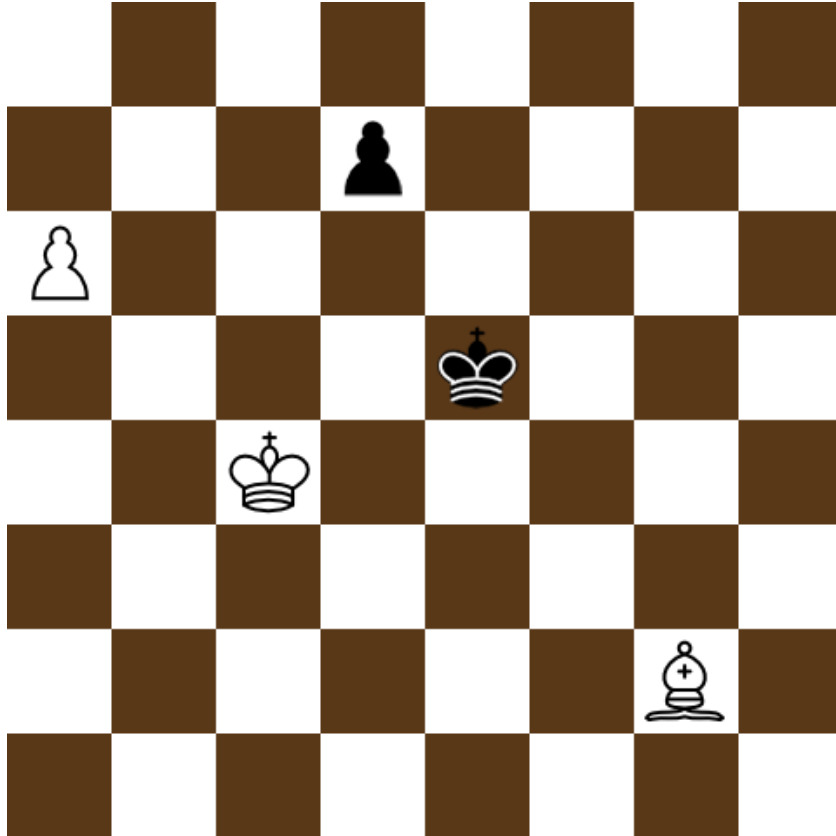
    return ind;
}

```

Acesti pasi vor fi repetati pentru toate cele 64 de patrate ale tablei de sah.

Rezultate experimentale

1. Identificarea unor pise



Rezultat:

[(0, 0) - spatiu alb][(0, 1) - spatiu negru][(0, 2) - spatiu alb][(0, 3) - spatiu negru][(0, 4) - spatiu alb][(0, 5) - spatiu negru][(0, 6) - spatiu alb][(0, 7) - spatiu negru]
[(1, 0) - spatiu negru][(1, 1) - spatiu alb][(1, 2) - spatiu negru][(1, 3) - pion negru][(1, 4) - spatiu negru][(1, 5) - spatiu alb][(1, 6) - spatiu negru][(1, 7) - spatiu alb]
[(2, 0) - pion alb][(2, 1) - spatiu negru][(2, 2) - spatiu alb][(2, 3) - spatiu negru][(2, 4) - spatiu alb][(2, 5) - spatiu negru][(2, 6) - spatiu alb][(2, 7) - spatiu negru]
[(3, 0) - spatiu negru][(3, 1) - spatiu alb][(3, 2) - spatiu negru][(3, 3) - spatiu alb][(3, 4) - rege negru][(3, 5) - spatiu alb][(3, 6) - spatiu negru][(3, 7) - spatiu alb]
[(4, 0) - spatiu alb][(4, 1) - spatiu negru][(4, 2) - rege alb][(4, 3) - spatiu negru][(4, 4) - spatiu alb][(4, 5) - spatiu negru][(4, 6) - spatiu alb][(4, 7) - spatiu negru]
[(5, 0) - spatiu negru][(5, 1) - spatiu alb][(5, 2) - spatiu negru][(5, 3) - spatiu alb][(5, 4) - spatiu negru][(5, 5) - spatiu alb][(5, 6) - spatiu negru][(5, 7) - spatiu alb]
[(6, 0) - spatiu alb][(6, 1) - spatiu negru][(6, 2) - spatiu alb][(6, 3) - spatiu negru][(6, 4) - spatiu alb][(6, 5) - spatiu negru][(6, 6) - nebun alb][(6, 7) - spatiu negru]
[(7, 0) - spatiu negru][(7, 1) - spatiu alb][(7, 2) - spatiu negru][(7, 3) - spatiu alb][(7, 4) - spatiu negru][(7, 5) - spatiu alb][(7, 6) - spatiu negru][(7, 7) - spatiu alb]

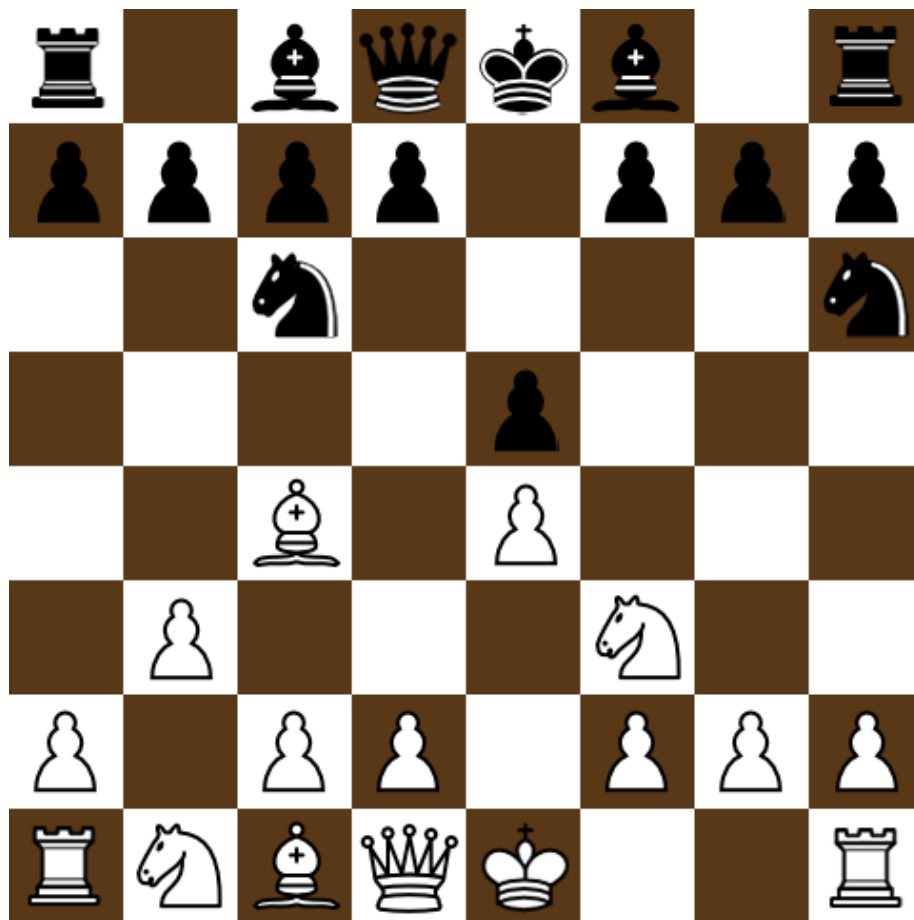
2. Identificarea tuturor piselor



Rezultat:

[(0, 0) - tura neagra [(0, 1) - cal negru [(0, 2) - nebun negru [(0, 3) - regina neagra [(0, 4) - rege negru [(0, 5) - nebun negru [(0, 6) - cal negru [(0, 7) - tura neagra]
 [(1, 0) - pion negru [(1, 1) - pion negru [(1, 2) - pion negru [(1, 3) - pion negru [(1, 4) - pion negru [(1, 5) - pion negru [(1, 6) - pion negru [(1, 7) - pion negru]
 [(2, 0) - spatiu alb [(2, 1) - spatiu negru [(2, 2) - spatiu alb [(2, 3) - spatiu negru [(2, 4) - spatiu alb [(2, 5) - spatiu negru [(2, 6) - spatiu alb [(2, 7) - spatiu negru]
 [(3, 0) - spatiu negru [(3, 1) - spatiu alb [(3, 2) - spatiu negru [(3, 3) - spatiu alb [(3, 4) - spatiu negru [(3, 5) - spatiu alb [(3, 6) - spatiu negru [(3, 7) - spatiu alb]
 [(4, 0) - spatiu alb [(4, 1) - spatiu negru [(4, 2) - spatiu alb [(4, 3) - spatiu negru [(4, 4) - spatiu alb [(4, 5) - spatiu negru [(4, 6) - spatiu alb [(4, 7) - spatiu negru]
 [(5, 0) - spatiu negru [(5, 1) - spatiu alb [(5, 2) - spatiu negru [(5, 3) - spatiu alb [(5, 4) - spatiu negru [(5, 5) - spatiu alb [(5, 6) - spatiu negru [(5, 7) - spatiu alb]
 [(6, 0) - pion alb [(6, 1) - pion alb [(6, 2) - pion alb [(6, 3) - pion alb [(6, 4) - pion alb [(6, 5) - pion alb [(6, 6) - pion alb [(6, 7) - pion alb]
 [(7, 0) - tura alba [(7, 1) - cal alb [(7, 2) - nebun alb [(7, 3) - regina alba [(7, 4) - rege alb [(7, 5) - nebun alb [(7, 6) - cal alb [(7, 7) - tura alba]

3. Test 3



Rezultat:

```

[(0, 0) - tura neagra ][(0, 1) - spatiu negru ][(0, 2) - nebun negru ][(0, 3) - regina neagra ][(0, 4) - rege negru ][(0, 5) - nebun negru ][(0, 6) - spatiu alb ][(0, 7) - tura neagra ]
[(1, 0) - pion negru ][(1, 1) - pion negru ][(1, 2) - pion negru ][(1, 3) - pion negru ][(1, 4) - spatiu negru ][(1, 5) - pion negru ][(1, 6) - pion negru ][(1, 7) - pion negru ]
[(2, 0) - spatiu alb ][(2, 1) - spatiu negru ][(2, 2) - cal negru ][(2, 3) - spatiu negru ][(2, 4) - spatiu alb ][(2, 5) - spatiu negru ][(2, 6) - spatiu alb ][(2, 7) - cal negru ]
[(3, 0) - spatiu negru ][(3, 1) - spatiu alb ][(3, 2) - spatiu negru ][(3, 3) - spatiu alb ][(3, 4) - pion negru ][(3, 5) - spatiu alb ][(3, 6) - spatiu negru ][(3, 7) - spatiu alb ]
[(4, 0) - spatiu alb ][(4, 1) - spatiu negru ][(4, 2) - nebun alb ][(4, 3) - spatiu negru ][(4, 4) - pion alb ][(4, 5) - spatiu negru ][(4, 6) - spatiu alb ][(4, 7) - spatiu negru ]
[(5, 0) - spatiu negru ][(5, 1) - pion alb ][(5, 2) - spatiu negru ][(5, 3) - spatiu alb ][(5, 4) - spatiu negru ][(5, 5) - cal alb ][(5, 6) - spatiu negru ][(5, 7) - spatiu alb ]
[(6, 0) - pion alb ][(6, 1) - spatiu negru ][(6, 2) - pion alb ][(6, 3) - pion alb ][(6, 4) - spatiu alb ][(6, 5) - pion alb ][(6, 6) - pion alb ][(6, 7) - pion alb ]
[(7, 0) - tura alba ][(7, 1) - cal alb ][(7, 2) - nebun alb ][(7, 3) - regina alba ][(7, 4) - rege alb ][(7, 5) - spatiu alb ][(7, 6) - spatiu negru ][(7, 7) - tura alba ]

```

Concluzii

Problema functioneaza corespunzator, dar ar putea fi desigur optimizat prin a nu parcurge toate imaginile.