SECONDARY SCHOOL
High School Slovenska Bistrica

AUTHORS
Barbara Ferčec, Peter Fekonja, Marko Fekonja

SUPERVISORS
Marko Žigart, Damijana Gregorič,
Mateja Petrič Podvršnik, Ivana Moškotevc, Nejc Ravnjak

An Application for Comparing Two Satellite Images

A group of students decided to write an application using the Java Programming Language, which would compare two satellite images and show differences in a new perspective. It actually does so by indicating patches in shades of red in the new image. We have clearly succeeded in doing so.

Problem overview

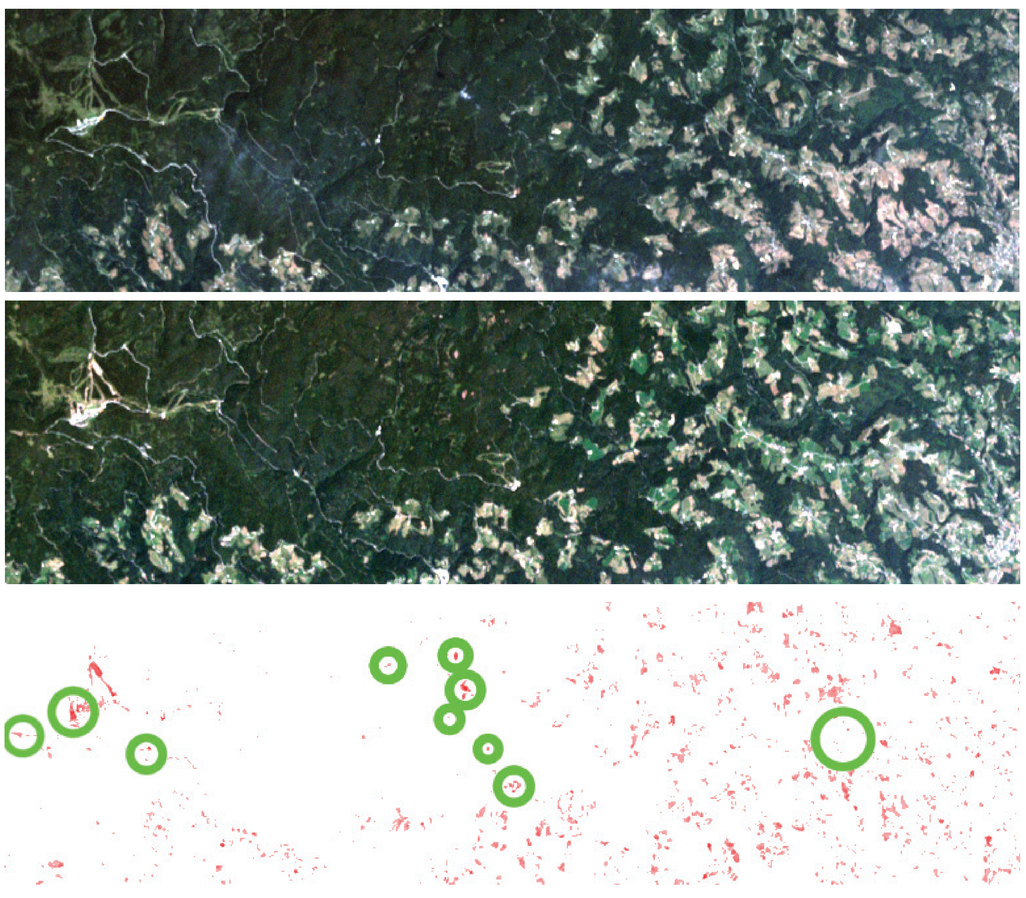
The objective of our group was to create an application, which compares two images and shows the differences between them in shades of red.

Data

- Landsat 5 images of Pohorje (year 2003 and 2011).

Methods

The application was created in the Java Programming Language for Android. It functions by reading the two images inserted, breaking them down to three basic colours and comparing the pixels by subtracting their colour values. The dissimilarities are then denoted by shades of red – the stronger the colour, the bigger the difference.



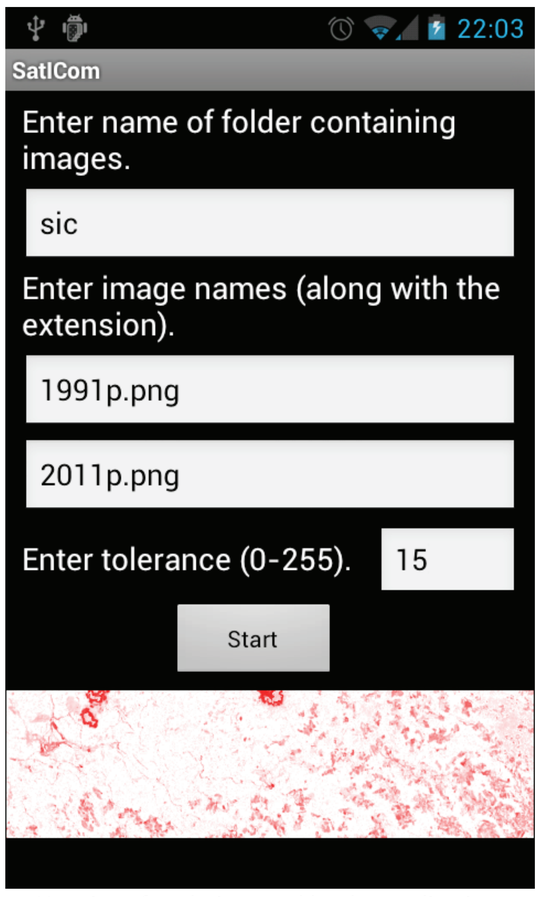
Original satellite images from 2003 and 2011 along with the image produced by the mobile application, showing differences between the original images.

This QR code is linked to:
http://www.mediafire.com/?8y1zl71m5bv225u, where the installation file for the Android application can be downloaded.

This QR code is linked to
http://www.mediafire.com/?7jyca8awc5oi1u0, where some sample satellite images can be downloaded.

Conclusions

Our results, produced by the application, were confirmed by the second team, which was tasked with solving the same problem but using manual comparison. The combined results were confirmed with a trip to the actual location of the clear-cuts.



User interface being used by the application.

```
int color1, color2, a, r, g, b, c;

for (int x = 0; x < widthMain; x++) {
      for (int y = 0; y < heightMain; y++) {
            if (x < width1 && y < height1)
            {
        color1 = mBitmap1.getPixel(x, y);
            }
        else
        {
          color1 = Color.BLACK;
        }

        if (x < width2 && y < height2)
        {
          color2 = mBitmap2.getPixel(x, y);
        }
        else
        {
          color2 = Color.BLACK;
        }

        a = 255;
        r = 255;
        c = (Math.abs(Color.red(color1) - Color.red(color2)) +
                Math.abs(Color.green(color1) - Color.green(color2)) +
                Math.abs(Color.blue(color1) - Color.blue(color2))) /  3;

        if (c < tolerance)
    {
      g = 255;
      b = 255;
    }
        else
        {
      g = 255-c;
      b = 255-c;
        }

        mBitmapOut1.setPixel(x, y, Color.argb(a, r, g, b));
    }
}
```

Main part of the code, used to calculate colour differences between the two pixels of the original image.