GREĐEVINSKI FAKULTET

UNIVERZITET U BEOGRADU

KATEDRA ZA GEODEZIJU I GEOINFORMATIKU

PREDMET: PROGRAMIRANJE GEOINFORMACIONIH SISTEMA

**OPIS PROJEKTA IZ PREDMETA PROGRAMIRANJE GEOINFORMACIONIH SISTEMA**

**STUDENT:** \_Mirko Milanović 1526/15\_

**TEMA:** \_Filtri bazirani na gradijentu drugog reda za izdvajanje ivica na rasterskim slikama u QGIS-u \_

**1. KONCEPT**

**1.1 OPIS**

Pri analizi objekata na slici neophodno je razgraničiti “objekte od interesa” od ostatka slike. Tehnika koja izdvaja objekte od interesa se naziva segmentacija. Dve najčešće korišćene tehnike segmentacije su pronalaženje ivica (izdvajanje samo onih piksela koji pripadaju rubovima objekata) i trešholding (izdvajanje celog objekta od pozadine pridružujući piksele čiji je intenzitet ispod nekog praga pozadini a ostale objektu ili obrnuto). Ne postoji univerzalna tehnika segmentacije koja će raditi na svim slikama niti je i jedna tehnika segmentacije savršena.

Osnovni zadatak tehnika za pronalaženje ivica regiona je da odrede (zatvorene) konture oko objekata od interesa. Postoje slike koje su jednostavne za segmentaciju, kao i druge gde je automatska segmentacija veoma težak zadatak.

Za automatsko izdvajanje ivica koriste se pre svega procedure zasnovane na gradijentu. Određivanje ivica ovom metom se svodi na traženje maksimuma amplituda gradijenta. Umesto derivacije slike, koriste se aproksimacije derivacije. Neki od najpoznatijih gradijentnih filtara su Robertsov, Prewittov i Sobelov gradijentni filtar. Za objekte sa velikim odnosom signal/šum poželjno je koristiti glačanje za suzbijanje šuma pre primene gradijenta.

U novije vreme više od gradijentnih filtara se koristi derivacijski filtar drugog reda, odnosno Laplasijan, kod koga prolasci kroz nulu određuju poziciju ivica. Koristi se Laplasijan, ili LoG filtar, gde se prvo vrši glačanje Gausovim filterom pa zatim Laplasijan. Kod LoG filtra umesto da se prvo primeni konvolucija sa Gausovim filtrom, a zatim Laplasijan, isti rezultat se može postići ako se konvolucioni kernel formira kao odmereni Laplasijan Gausove funkcije. Lokacija ivica je određena prolazima kroz nulu. Dobijeni dvodimenzionalni konvolucioni kernel se naziva “Meksikanski šešir”.

**1.2 IDEJA REŠENJA**

U ovom projektnom zadatku je potrebno izvršiti automatsko izdvajanje ivica na rasterskim slikama. Program koji se pritom koristi je QGIS Desktop. Ideja je da se kreira plug-in dodatak programu QGIS, isprogramiran u python programskom jeziku, koji će obezbediti izdvajanje ivica nad izabranim unetim rasterskim lejerom. Ivice je moguće izdvojiti jednom od dve metode za izdvajanje ivica, odnosno jedan od dva derivacijska filtra drugog reda, Laplasian i LoG filtar (Laplasian Gausa). Kao ponuđene opcije stoje dva filtra:

* Laplasijan i
* Gausov filtar

Laplasijan omogućava izdvajanje ivica direktno, dok se Gausov filtar koristi za glačanje. Pomenuti LoG filtar se dobija prvo glačanjem Gausovim filterom a zatim Laplasijanom. Pored izbora korišćenog filtra, postoji mogućnost i za izbor korišćenog kernela. Kernel je korisnički definisan, odnosno za oba pomenuta filtera se mogu uneti parametri za kernel:

* Mod (kvadrat ili krug),
* Standardna devijacija i
* Radijus.

Default vrednosti treba da stoje upisane u odgovarajućim poljima, ali se one mogu menjati.

Kao rezultat operacije za primenjeni filter, potrebno je da se formira novi rasterski lejer koji se istovremeno pamti u TIFF formatu na zadatom direktorijumu računara.

**1.3 KORISNICI**

Korisnici su sve zainteresovane institucije i organizacije koje se bave digitalnom obradom slika u funkciji obrade prostornih podataka; podaci mogu biti od koristi u fotogrametriji, kod digitalizacije snimljenog sadržaja, kod prostornog planiranja, za različite vrste daljih istraživanja i analiza prostornih podataka i u drugim oblastima itd. Korisnici su i svi zainteresovani gradjani Republike Srbije.

**2. METODOLOGIJA RADA**

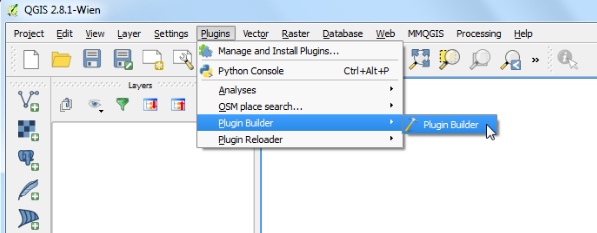
**2.1 ULAZNI PODACI**

Za izradu projekta nije potrebno obezbediti nikakve posebne podatke. Funkcionalnost izdvajanja ivica je potrebno omogućiti nad rasterskim slikama bilo kakvog sadržaja u grayscale formatu (0-255 vrednosti). QGIS pre svega radi sa rasterskim slikama u TIFF formatu, pa je bilo kakav grayscale TIFF rasterski lejer pogodan za testiranje izdvajanja ivica, odnosno za testiranje rada plug-in dodatka programu.

**2.2. METODOLOGIJA OBRADE I ANALIZA ULAZNIH PODATAKA**

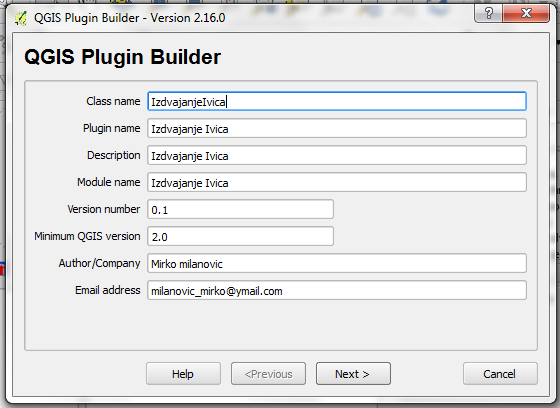
Za kreiranje plug-in-a u QGIS Desktop programu, prvo je potrebno kreirati “kostur” plug-in dodatka, odnosno, prvo se kreira osnova za plug-in koju čine svi potrebni fajlovi od kojih se plug-in sastoji. Kasnije je potrebno te fajlove dopuniti python kodom za potrebne funkcionalnosti i za korisnički interfejs plug-in-a.

Za kreiranje plug-in-a u QGIS Desktop softveru se bira opcija ***Plugins/Plugin Builder/Plugin Builder*** (Slika 1).



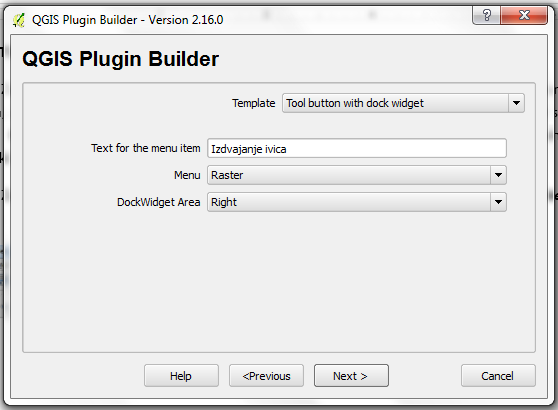
*Slika 1:* QGIS: Pozivanje Plugin Builder-a

U Plugin Builder dijalogu se popunjavaju tražena polja o nazivima plug-in-a, klase, opisi, podaci o autoru i sl. (Slika 2).



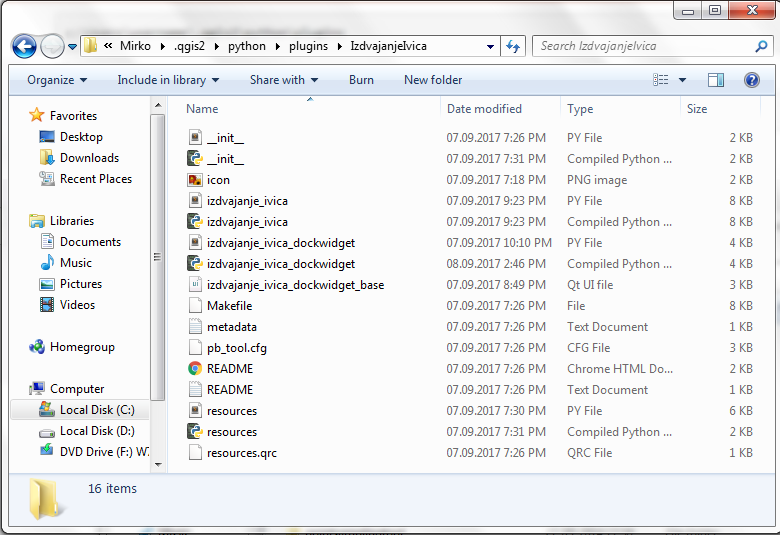
*Slika 2:* Plugin Builder, unos naziva i autora

U sledećem koraku dijaloga se unosi naziv plug-in prozora, i gde će se prozor nalaziti. Izabrano je da plug-in bude u sastavu Raster menija, a da se pojavljuje na desnoj strani sidebar-a a ne kao klasičan popup prozor (Slika 3).



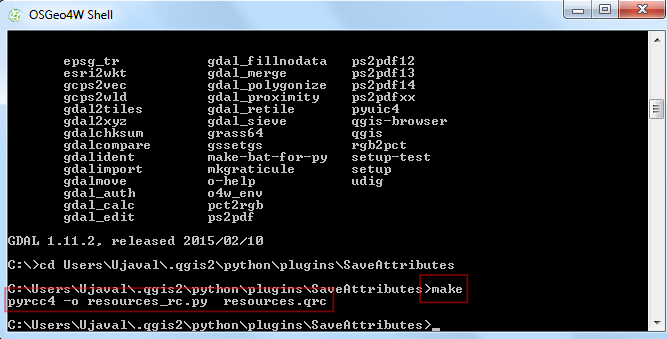
*Slika 3:* Plugin Builder, unos mesta za smeštanje plug-in-a u radnom okruženju QGIS-a

U nastavku kreiranja plug-in dodatka popunjavaju se Publication polja gde se unose linkovi ka github repozitorijumu gde će se smeštati programski kod i gde će se pratiti greške. Uneti su linkovi ka kreiranom github repozitorijumu: <https://github.com/MirkoMilanovic/Izdvajanje-Ivica>. Na kraju se obeležava opcija ***Flag the plugin as experimental***, i kreira se osnovni set fajlova plug-in-a na izabranom direktorijumu (Slika 4).



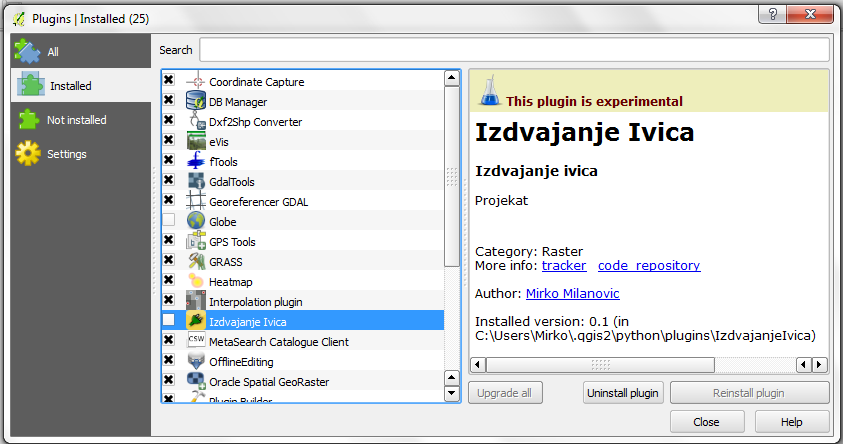
*Slika 4:* Kreirane datoteke pri kreiranju QGIS plud-in-a

Nakon kreiranja, potrebno je izvršiti kompajliranje kreiranog plug-in fajla ***resources.qrc***. To se vrši tako što se za Windows platformu pokreće **OSGeo4W Shell**. Prelazi se na direktorijum gde je plug-in smešten (cd C:\Users\Mirko\.qgis2\python\plugins\IzdvajanjeIvica) a zatim se kuca make. Kao rezultat se pokreće pyrcc4 komanda Qt bindings, koji je prethodno instaliran (Primer na slici 5).



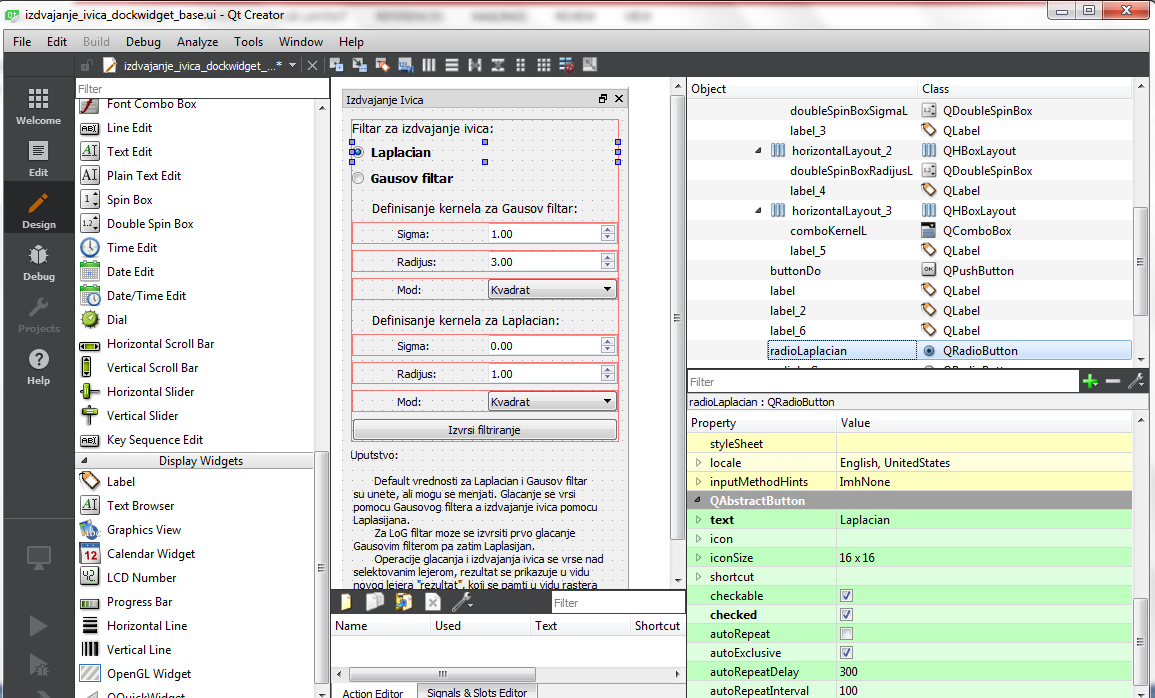
*Slika 5:* OSGeo4W Shell – kompajliranje plug-in fajla resources.qrc

Dalje što je potrebno uraditi je da se plug-in importuje u QGIS Desktop okruženje. To se radi u QGIS meniju ***Plugins/Manage and Install plugins***, u odeljku ***Settings*** se obeležava ***Show Experimental*** ***Plugins***, dok se u odeljku ***All*** traži izrađen plug-in “Izdvajanje Ivica” i instalira se. U odeljku ***Installed*** se pojavljuje instaliran plug-in sa podacima o autoru i linkovima ka online repozitorijumu i instalacionom direktorijumu gde je instaliran na lokalnom računaru (Slika 6).



*Slika 6:* Plugins meni – lista instaliranih plug-in-ova

Kreiran plug-in je blanko, odnosno nema još uvek nikakve funkcionalnosti. Pre svega treba osmisliti interfejs i izraditi ga. Sve potrebne opcije u vidu mesta za unose, dugmića, padajućih lista za izbor opcija, ispisa, default vrednosti i slično se dodaju na jednostavan način korišćenjem programa **Qt Creator**. Otvara se fajl ***izdvajanje\_ivica\_dockwidget\_base.ui*** i kreira se interfejs plug-in-a (Slika 7).

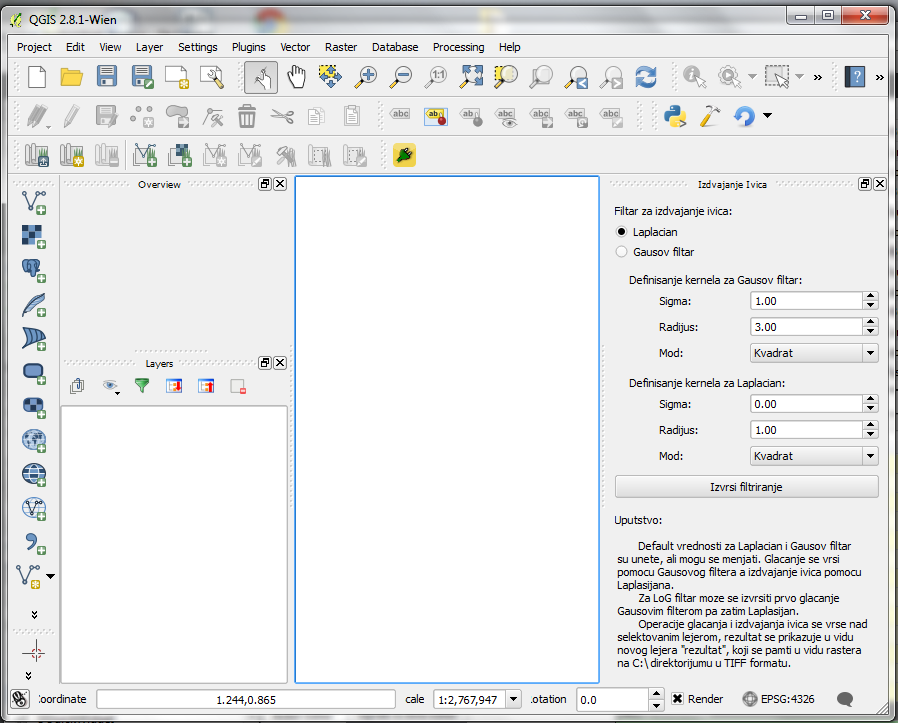
 *Slika 7:* Qt Creator – kreiranje interfejsa

Kao što je rečeno, u Qt Creator-u se ceo interfejs formira korišćenjem, za to se koriste intuitivne opcije u levom sidebar-u programa. Međutim, u desnom sidebar-u se dodaju i default vrednosti za sva polja i izabrane funkcije, ali se i za svaki kreiran objekat kreiranog prozora – interfejsa može promeniti ime u desnom sidebar-u. Promena imena kreiranim objektima je poželjna kako bi samo python programiranje bilo intuitivnije. Na primer:

* kućica za izbor filtera Laplacian se zove “radioLaplacian”, izabrana je po defaultu,
* definisanje vrednosti sigma kernela Gausovog filtera je nazvan “doubleSpinBoxSigmaG”,
* izbor moda Gausovog filtera je “comboKernelG”,
* dugme za izvršavanje filtriranja se zove “buttonDo” i sl.

Na adekvatnim mestima su dodati i prikladni ispisi i ispis uputstva, kako bi se na najlakši i najbrži način novi korisnici upoznali sa kreiranim plug-in-om.

U QGIS Desktop okruženju je moguće videti kako plug-in sa kreiranim interfejsom vizuelno izgleda, pa se eventualno može vratiti na Qt Creator radi promene interfejsa. Otvaranje plug-in-a se vrši izborom opcije ***Plugin/Plugin Reloader/Choose a plugin to be reloaded***. Plug-in se dodaje u desnom sidebar-u kako je prvobitno izabrano tokom kreiranja (Slika 8). Kao poslednja alatka u paleti sa alatkama se dodaje ikonica za dodat plug-in.



*Slika 8:* Dodat plug-in u okruženje QGIS

Nakon kreiranja interfejsa, potrebno je isprogramirati sve funkcionalnosti plug-in-a. U nekom od text editora se može izvršiti programiranje. Korišćen je text editor **Sublime text** za otvaranje datoteke ***izdvajanje\_ivica\_dockwidget.py*** u kojoj je izvršeno programiranje. Neki deo koda datoteke je po defaultu kreiran.

Pomenuta datoteka počinje iskomentiranim podacima o autoru i sl.

# -\*- coding: utf-8 -\*-

"""

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IzdvajanjeIvicaDockWidget

A QGIS plugin

Izdvajanje ivica

-------------------

begin : 2017-09-07

git sha : $Format:%H$

copyright : (C) 2017 by Mirko Milanovic

email : milanovic\_mirko@ymail.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* This program is free software; you can redistribute it and/or modify \*

\* it under the terms of the GNU General Public License as published by \*

\* the Free Software Foundation; either version 2 of the License, or \*

\* (at your option) any later version. \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

"""

Naredni deo koda je delom automatski generisan, ali deo je potrebno dodati. Prvo se unose korišćeni paketi biblioteka. **PyQt4** je paket neophodan za kreiranje interfejsa, iz paketa **qgis-core** se importuje biblioteka za rad sa rasterskim podacima, importuju se i funkcije *random* i *processing* koje će se kasnije koristiti (random za izbor random broja a processing za izabrane filtere). Nakon toga, dodate se konstante (polja). U automatski kreiranu glavnu klasu *IzdvajanjeIvicaDockWidget* se dodaje prepoznavanje izvršavanja klikom na dugme *buttonDo*, odnosno povezuje se sa metodom process\_layer.

import os

from PyQt4 import QtGui, uic

from PyQt4.QtCore import pyqtSignal

import random

import processing

from qgis.core import QgsRasterLayer

from PyQt4.QtCore import QFileInfo

FORM\_CLASS, \_ = uic.loadUiType(os.path.join(

os.path.dirname(\_\_file\_\_), 'izdvajanje\_ivica\_dockwidget\_base.ui'))

# Constants

LAPLACIAN = 1

LOG = 2

class IzdvajanjeIvicaDockWidget(QtGui.QDockWidget, FORM\_CLASS):

closingPlugin = pyqtSignal()

def \_\_init\_\_(self, parent=None):

"""Constructor."""

super(IzdvajanjeIvicaDockWidget, self).\_\_init\_\_(parent)

# Set up the user interface from Designer.

# After setupUI you can access any designer object by doing

# self.<objectname>, and you can use autoconnect slots - see

# http://qt-project.org/doc/qt-4.8/designer-using-a-ui-file.html

# #widgets-and-dialogs-with-auto-connect

self.setupUi(self)

# polja

self.activeLayer = None

self.iface = None

self.buttonDo.clicked.connect(self.process\_layer) # Povezuje dogadjaj click dugmeta sa metodom process\_layer

def closeEvent(self, event):

self.closingPlugin.emit()

event.accept()

Narednim kodnim zapisom se definišu metode koje povezuju adekvatne objekte interfejsa:

def getKernelModL(self):

return self.comboKernelL.currentIndex()

def getSigmaL(self):

return self.doubleSpinBoxSigmaL.value()

def getRadijusL(self):

return self.doubleSpinBoxRadijusL.value()

def getKernelModG(self):

return self.comboKernelG.currentIndex()

def getSigmaG(self):

return self.doubleSpinBoxSigmaG.value()

def getRadijusG(self):

return self.doubleSpinBoxRadijusG.value()

Sledi kodni zapis napravljenih metoda *getFilter* za prepoznavanje čekiranog filtera i *process\_layer* u kojem se definiše ispis u Python Console, zatim koja će se funkcija vršiti u zavisnosti od selekcije i dodavanje rasterskog lejera po završetku operacije filtriranja:

def getFilter(self):

if self.radioLaplacian.isChecked():

return LAPLACIAN

elif self.radioLoG.isChecked():

return LOG

return None # null u javi -> None

def process\_layer(self):

self.layer = self.iface.activeLayer()

print("Izdvajanje ivica. Filter {0}".format(self.getFilter()))

if self.getFilter() == LAPLACIAN:

result = self.laplacian(self.layer)

else:

result = self.gauss(self.layer)

self.iface.addRasterLayer(result["RESULT"], "rezultat")

Na kraju, kodni zapis za funkcije koje se pokreću u zavisnosti od izabranog filtera (laplacian() i gauss()), zajedno sa iskomentiranim objašnjenjima parametara slede u nastavku:

def laplacian(self, layer):

# PARAMETRI ZA LAPLACIAN

# Grid [raster]

# <put parameter description here>

# Method [selection]

# <put parameter description here>

# Options:

# 0 — [0] standard kernel 1

# 1 — [1] standard kernel 2

# 2 — [2] Standard kernel 3

# 3 — [3] user defined kernel

# Default: 0

# Standard Deviation (Percent of Radius) [number]

# <put parameter description here>

# Default: 0

# Radius [number]

# <put parameter description here>

# Default: 1

# Search Mode [selection]

# <put parameter description here>

# Options:

# 0 — [0] square

# 1 — [1] circle

# Default: 0

sigma = self.getSigmaL()

radius = self.getRadijusL()

mode = self.getKernelModL()

fileName = "C:\laplace{0}.tif".format(random.randint(0, 1000000))

return processing.runalg('saga:laplacianfilter', self.layer, 3, sigma, radius, mode, fileName)

def gauss(self, layer):

# PARAMETRI ZA GAUSOV FILTAR

# Grid [raster]

# <put parameter description here>

# Standard Deviation [number]

# <put parameter description here>

# Default: 1

# Search Mode [selection]

# <put parameter description here>

# Options:

# 0 — [0] Square

# 1 — [1] Circle

# Default: 0

# Search Radius [number]

# <put parameter description here>

# Default: 3

sigmaG = self.getSigmaG()

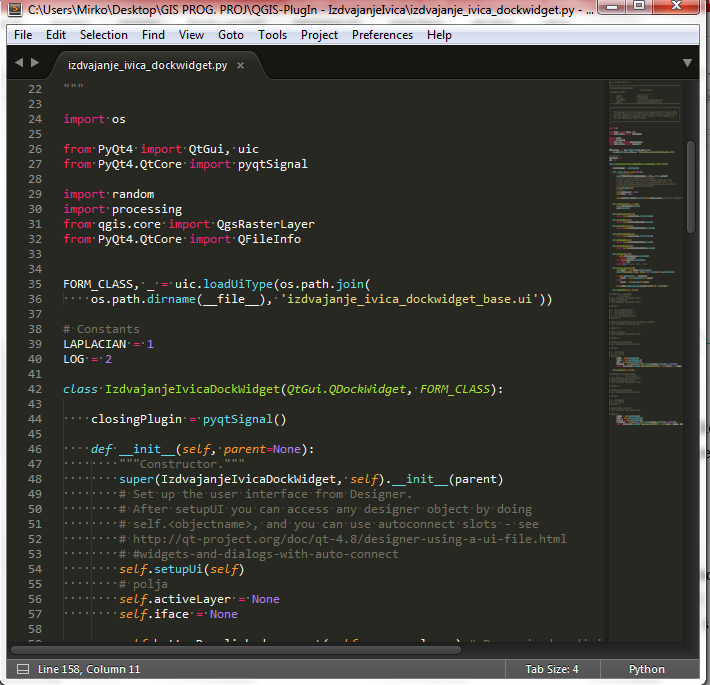
modeG = self.getKernelModG()

radiusG = self.getRadijusG()

fileNameG = "C:\gauss{0}.tif".format(random.randint(0, 1000000))

return processing.runalg('saga:gaussianfilter', self.layer, sigmaG, modeG, radiusG, fileNameG)

Sublime Text je jedan od najpopularnijih text editora iz razloga što omogućava pregledno porgramiranje, takođe omogućava i određeno ukazivanje na eventualne greške (Slika 9).

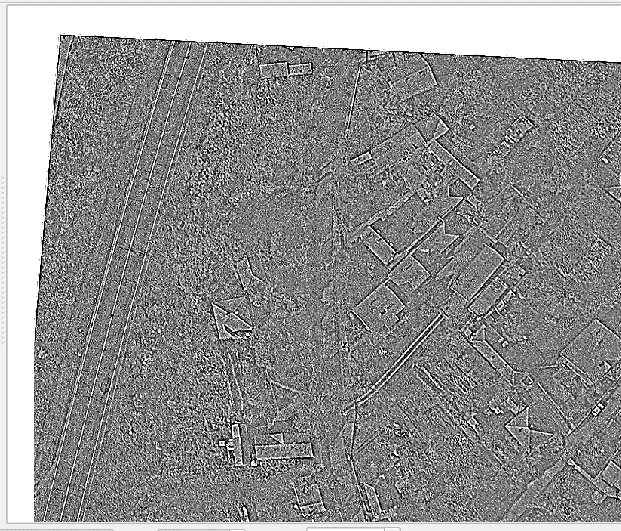
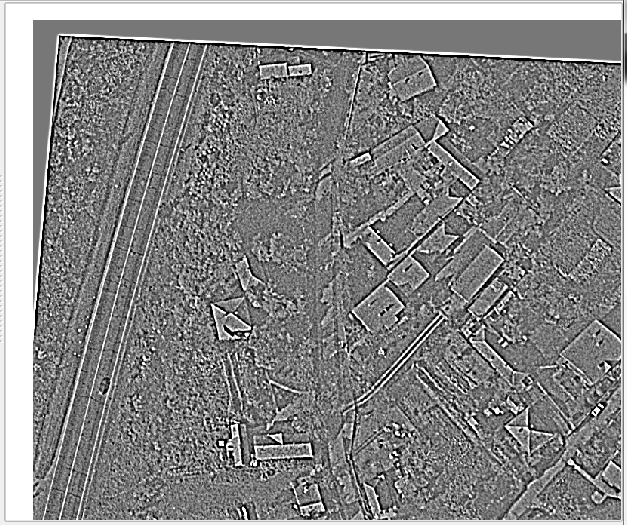


*Slika 9:* Sublime text

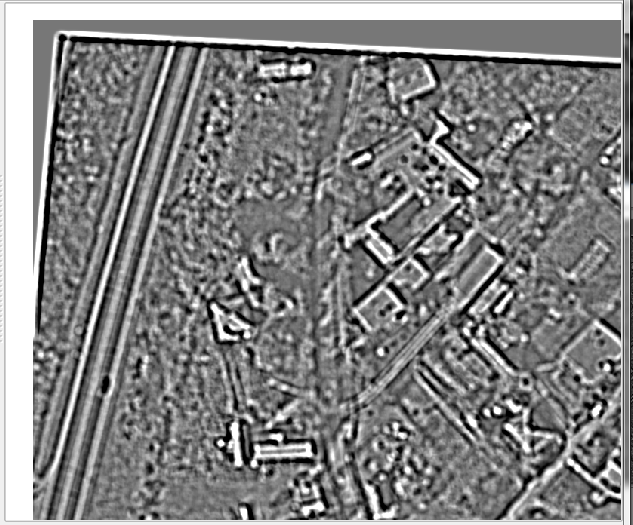
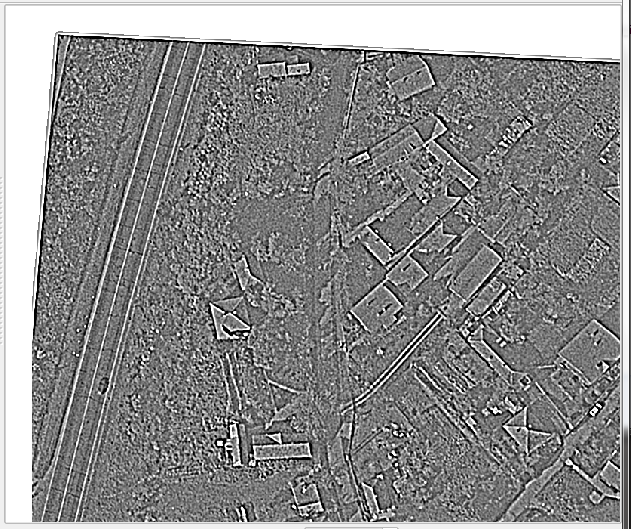
Tokom programiranja plug-in-a, a posebno nakon njegovog završetka, potrebno je ispitati njegov rad. U QGIS Desktop okruženju se unose grayscale slike TIFF formata i vrši se filtriranje sa oba filtera. Primer je dat na slici aerofoto (Slika 10). Uočava se da LoG daje bolje rezultate od Laplasijana, međutim, ako se pretera sa glačanjem Gausovim filterom, rezultat LoG filtriranja postaje zamućen.



Gaus (kvadratni kernel: sigma=1, radijus=3) Gaus (kvadratni kernel: sigma=4, radijus=6)

Laplasijan (kvadratni kernel: sigma=0, radijus=1) Laplasijan (kvadratni kernel: sigma=3, radijus=4)



LoG (kerneli definisani kao slike iznad) LoG (kerneli definisani kao slike iznad)

*Slika 10:* Rezultati filtriranja: Gausov filtar, Laplasijan, Laplasijan Gausa – LoG (odozgo na dole)

**3. ZAKLJUČAK**

Python je programski jezik visokog nivoa opšte namene. Podržava, u prvom redu imperativni, objektno-orijentisan i funkcionalni stil programiranja. Sintaksa jezika Python omogućava pisanje veoma preglednih programa. Jezik se brzo i lako uči.

Kao jezik opšte namene, python je veoma pogodan i za rad sa prostornim podacima, tako da se lako implementira i u QGIS Desktop okruženje, jednog od najviše korišćenih programa za rad sa prostornim podacima.

Izrađen dodatak programa QGIS Desktop je pokazao kako se korišćenjem python programskog jezika, pored programiranja osmišljenih funkcionalnosti, te funkcionalnosti mogu implementirati i u praksi na konkretnim zadacima. Konkretno, omogućeno je da se korišćenjem adekvatnog interfejsa izvedu operacije filtriranja radi izdvajanja ivica na intuitivan i brz način nad rasterskim lejerima. Pri tom, omogućen je unos i svih potrebnih parametara za izvođenje pomenutih filtriranja.

Pošto izrađeni plug-in, nakon pokretanja operacija, rezultate izbacuje brzo u vidu rasterskih lejera i pamti ih kao TIFF rasterske datoteke, konačni rasteri izvršenih filtriranja se brzo mogu pregledati. Pregledom tih rezultujučih rastera se uočava kako primena Gausovog filtera za glačanje pre primene Laplasijana za izdvajanje ivica daje bolje rezultate nego kada se odmah primeni samo Laplasijan. Drugim rečima, LoG filtar daje bolje rezultate od Laplasijana, pošto je glačanjem smanjen uticaj šuma a suvišne ivice zbog prevelike detaljnosti se ne izdvajaju.

**4. LITERATURA**

QGIS Tutorials and Tips (2016). Building a Python Plugin - Ujaval Gandhi [online]. Dostupno na:

<http://www.qgistutorials.com/en/docs/building_a_python_plugin.html>

QGIS Tutorials and Tips (2016). Using Plugins - Ujaval Gandhi [online]. Dostupno na:

<http://www.qgistutorials.com/en/docs/using_plugins.html>

Wikipedia (2017). Pajton [online]. Dostupno na:

<https://sr.wikipedia.org/wiki/Пајтон_(програмски_језик)>

Wikipedia (2017). Gaussian function [online]. Dostupno na:

<https://en.wikipedia.org/wiki/Gaussian_function>

Documentation for QGIS testing (2017). Grid filter [online]. Dostupno na: <https://docs.qgis.org/testing/en/docs/user_manual/processing_algs/saga/grid_filter.html#user-defined-filter>

Homepages (2003). Laplacian/Laplacian of Gaussian - R. Fisher, S. Perkins, A. Walker and E. Wolfart [online]. Dostupno na:

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

Fred's ImageMagick Scripts (2015). LAPLACIAN - Fred Weinhaus [online]. Dostupno na:

<http://www.fmwconcepts.com/imagemagick/laplacian/index.php>

Geographic Information Systems (2013). How to get field names in pyqgis 2.0 – gene [online]. Dostupno na:

<https://gis.stackexchange.com/questions/76364/how-to-get-field-names-in-pyqgis-2-0>