USING SIMPLECV FOR SEED METADATA EXTRACTION INTO XML DOCUMENT

Sergio Silva Ribeiro

Universidade Estadual de Ponta Grossa (UEPG) – Ponta Grossa – PR – Brasil
professor@sergioribeiro.com.br

Abstract. *Computing approaches have been used in agriculture problems. Seeds information like shape, size, texture. color, etc are important for agriculture traits resulting in quality and market price. The purpose of this work was used digital image processing with metadata techniques to generate data from seeds using SimpleCV framework and programming language Python. The result shows that this combination can be used efficiently for this purpose.*

Keywords: Computer Vision, Python, Agriculture.

1. INTRODUCAO

Today, practically all phones and computers have cameras, images, and videos can be generated anytime and anywhere. Online on the internet have them in the billions and it is not uncommon for people to have many gigabytes of photos and videos. It is necessary algorithms for understanding what is in theses images and converts them into information, this is the field of computer vision that powers applications like image search, robot navigation, medical image analysis, image processing, etc [1].

Fundamental steps of digital image processing in a computer vision system are: (I) Image acquisition, that could be as simple as being given an image that is already in digital form; (II) Image enhancement, the idea is to bring out detail that is obscured, or simply to highlight certain features of interest in an image; (III) Image restoration, that deals with improving the appearance of an image; (IV) Color image processing, include color modeling and processing in a digital domain, etc; (V) Wavelets and multiresolution processing are the foundation for representing images in various degrees of resolution; (VI) Compression , reducing the storage required to save or transmit an image; (VII) Morphological processing, extract image components useful in the representation and description of shape; (VIII) Segmentation, divides an image into its constituent parts or objects; (IX) Representation and description transforms raw data into a form suitable for subsequent computer processing; (X) Object recognition is the process that assigns a label into object based on its descriptor; (XI) Knowledgebase indicates regions of an image where the information of interest is known to be located (FIGURE 1) [9][13].

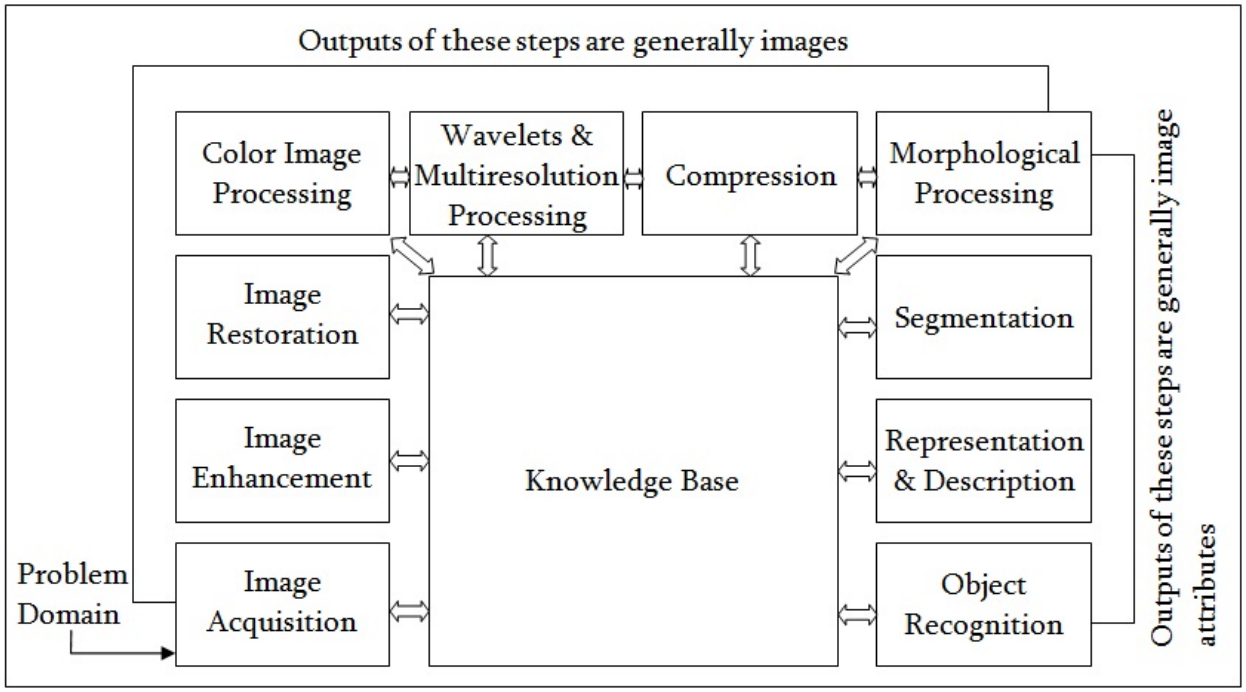

Figure 1 - Fundamental steps of digital image processing [9]

A programming language is required for developing of digital image processing. Python is a powerful computer programming language [10]. Its was optimized to be productive and to have readable codes with high quality [2]. SimpleCV is a framework for use with Python and as its name implies was designed to be simple. SimpleCV is a specific framework to computer vision uses that powers Python for this purpose [3]. Also, can add a metadata and XML document to improve a digital image program solution.

Metadata could be defined like data about data, that is an old definition, however moderns authors argue for a broader definition [4]. Metadata is very important helping us to find information quickly because the information is organized and structured. In a metadata information are not only labeled as well managed [5]. XML (Extensible Markup Language) is a metalanguage used to define marks and a tree structural relationship or integration between them using linear syntax. XML documents could be understood a powerful kind of metadata [4].

Yield are affected by quality of seeds and it is associated with its aspects like shape, size, color, texture, etc (FIGURE 2). Thus, this kind of information are the most important agronomic traits because affect yield, quality and market price [6]. A study was developed by [7] and the objective was to get systematic and complete information on important seed morphology of maize like size, shape, texture and color. Basics reviews were presented by [8], which are contributing to improving insight of seed morphology, seed image analysis in terms of quality, germination, varietal identification, characterization, moisture, grading, sorting, etc.

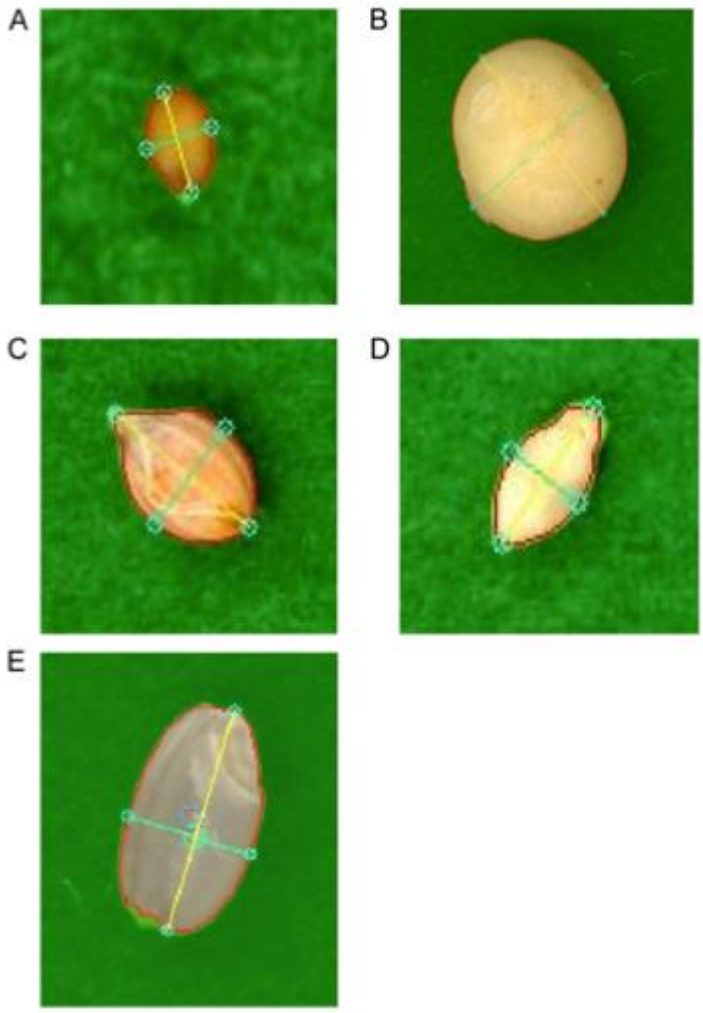

Figure 2 - Seed detection and measurement. (A) Arabidopsis, (B) Soybean, (C) *S. italica*, (D) *S. viridis* e (E) cv Koshihikari rice [6]

This work's purpose is present how to use SimpleCV framework on Python to processing seed images and extracting information into a metadata based on XML structure.

2. MATERIAIS AND METHODS

For this work was used a Notebook ACER Aspire, model 5733-6663, with processor Intel Core i3 and 4GB of memory, Linux UBUNTU 12.04 LTS version. The programming language used was Python 2.7.3 version [10]. The framework used was SimpleCV 1.3.0 version [3]. To write and execution of algorithms was used (SPE) Stani's Python Editor, which is a Python IDE wrote in Python available in <http://pythonide.stani.be/>.

Images acquisition was made by scanner SAMSUNG, model SCX-4600. The software used to acquisition was Xsane Imaging Sanner Program 0.998 version with following configuration: File type: png; Background: plane; Color: all; Resolution: 300 dpi; Size: A4 and location os scanner area: x top left value of 0.5 cm, y top left value of 0.5 cm, x bottom right value 21.0 cm, and y bottom right value 29.0 cm. Was used which background blue EVA attached on the scanner lid, and for grain position was used EVA matrix with 88 holes sizing 1.5 cm (width) x 2.0 cm (height) approximately.

After put grain in the holes of EVA matrix, this last one was removed, resting only grains in its positions to proceed scanning. For this work was used 50 seeds divided into 4 groups: rice,

corn, bean, chickpea and lentil (FIGURE 3).
In the Algorithm 1 is shown the program used to processing images of seeds, detection of each one seed present on images, segmentation and extraction of features from seed.

Algorithm 1 – Source code of program do processing image and extracting metadata informations

```
 1:  from SimpleCV import Image,Color, Display, np
 2:  import glob, math, os, sys, time
 3:  import cv
 4:  import numpy as np
 5:  from lxml import etree
 6:
 7:  color  = Color.BLUE
 8:  item   = 1
 9:  files  = glob.glob("graos-300-dpi-rice.png")
10:  files += glob.glob("graos-300-dpi-cariocabeans.png")
11:  files += glob.glob("graos-300-dpi-chickpea.png")
12:  files += glob.glob("graos-300-dpi-lentil.png")
13:  files += glob.glob("graos-300-dpi-corn.png")
14:
15:  for file in files:
16:
17:      img = Image(file)
18:      img  = img.crop(5,5,img.width-10,img.height-5).medianFilter()
19:      mask = img.colorDistance(color).medianFilter().binarize().erode(1).invert()
20:      seed = img-mask.invert()
21:      dist = seed.colorDistance(color).invert()
22:      blobs = seed.findBlobs(minsize=1000,maxsize=30000)
23:
24:
25:      if( blobs is not None ):
26:          for blob in blobs:
27:              hist = blob.blobImage().toGray().histogram(256)
28:              huehist = blob.blobImage().hueHistogram(256)
29:
30:
31:              root = etree.Element("grain", number=str(item) )
32:
33:              t_area = etree.SubElement(root, "group")
34:              t_area.text = file.split('-', 3 )[3].split('.')[0]
35:
36:              t_area = etree.SubElement(root, "area")
37:              t_area.text = str(blob.area())
38:
39:
40:              t_radius = etree.SubElement(root, "radius")
41:              t_radius.text = str(blob.radius())
42:
43:
44:              t_radius = etree.SubElement(root, "perimeter")
45:              t_radius.text = str(blob.perimeter())
46:
47:              t_radius = etree.SubElement(root, "contour")
48:              t_radius.text = str(blob.contour())
49:
50:              t_radius = etree.SubElement(root, "gray-histogram")
51:              t_radius.text = str(hist)
52:
53:              t_radius = etree.SubElement(root, "hue-histogram")
54:              t_radius.text = str(huehist)
55:
56:              f2save = file.split('-', 3 )[3].split('.')[0]+"-"+str(item)+".xml"
57:              f2w = open(f2save,"w")
58:              f2w.write( etree.tostring(root, pretty_print=True) )
                 f2w.close()

                 item += 1
```

The first five lines of Algorithm are responsible for importing python classes to be used in the program like SimpleCV. Background color is defined in line 7, this variable will be used to difference calculation from background and seeds. Lines from 8 to 13 are created variables for image files reading. Lines from 24 to 50 each seed detected (blob) is processed and its characteristics, number, group, area, radius, perimeter, contour, gray histogram and hue histogram are extracted and stored in appropriated variables. Finally in lines from 52 to 55 each XML document file is created within its characteristics.

3. RESULTS AND DISCUSSIONS

In Figure 3 is shown acquisition results which 50 seeds divided into 5 groups. Each image resulting in a PNG format file with 300 dpi of resolution.

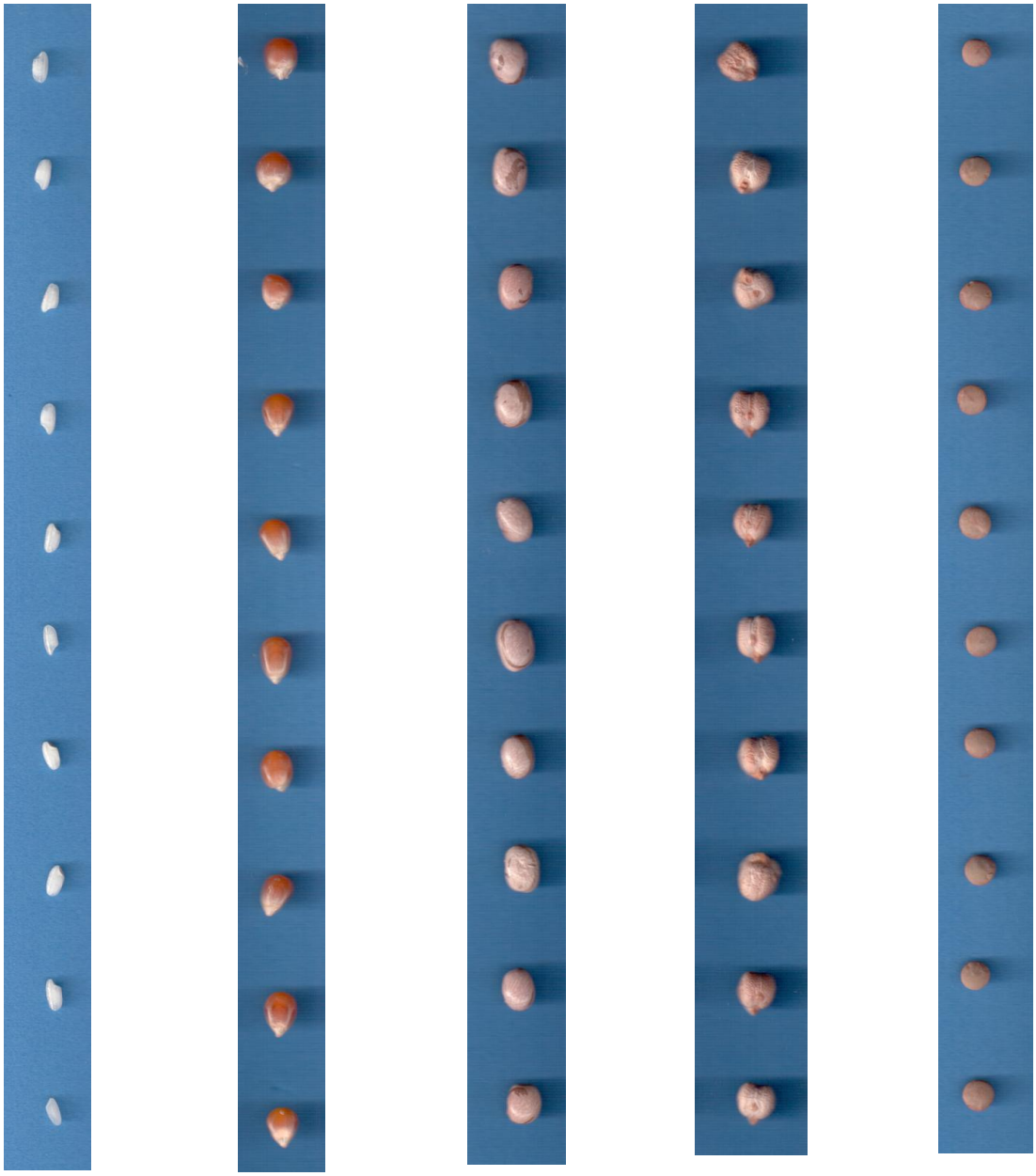

Figure 3 - Seeds used: (A) rice, (B) corn, (C) bean, (D) chickpea and (E) lentil.

Seeds from Figure 3 was detected by Algorithm using digital image process techniques and each seed object was separated from the background for next process step. Finally for each seed metadata information, group, area, radius, perimeter, contour, gray histogram and hue histogram were extracted into XML document file.

Following, in Figure 4 is shown XML file result from an extraction process. In this

picture, the XML file structure contains main attributes and are the number, group, area, radius, perimeter, contour, gray histogram and hue histogram.


```
 number="1">
p>rice</group>
>1906.5</area>
us>25.249661147</radius>
meter>187.923879504</perimeter>
our>[(113, 2808), (112, 2809), (112, 2810), (111, 2811), (111, 2813),
     (109, 2815), (109, 2816), (108, 2817), (108, 2819), (106, 2821),
     (106, 2822), (105, 2823), (105, 2824), (104, 2825), (104, 2826),
     (103, 2827), (103, 2828), (101, 2830), (101, 2835), (102, 2836),
     (102, 2838), (103, 2839), (103, 2842), (104, 2843), (104, 2845),
     (105, 2846), (105, 2848), (106, 2849), (106, 2850), (107, 2851),
     (107, 2852), (108, 2853), (108, 2855), (109, 2856), (109, 2857),
     (110, 2858), (110, 2859), (112, 2861), (112, 2862), (113, 2863),
     (113, 2864), (116, 2867), (116, 2868), (118, 2870), (118, 2871),
     (124, 2877), (125, 2877), (127, 2879), (129, 2879), (130, 2880),
     (134, 2880), (135, 2879), (136, 2879), (137, 2878), (138, 2878),
     (140, 2876), (140, 2875), (141, 2874), (141, 2871), (142, 2870),
     (142, 2861), (141, 2860), (141, 2853), (140, 2852), (140, 2849),
     (139, 2848), (139, 2845), (138, 2844), (138, 2842), (137, 2841),
     (137, 2839), (136, 2838), (136, 2836), (135, 2835), (135, 2833),
     (134, 2832), (134, 2830), (133, 2829), (133, 2828), (132, 2827),
     (132, 2826), (131, 2825), (131, 2824), (130, 2823), (130, 2821),
     (129, 2820), (129, 2819), (127, 2817), (127, 2816), (126, 2815),
     (126, 2814), (125, 2813), (125, 2812), (122, 2809), (121, 2809),
     (120, 2808)]</contour>
-histogram>[1078, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 2, 0, 3, 1, 0, 1, 2, 1, 2, 0, 0, 5,
            2, 4, 3, 2, 0, 0, 1, 7, 5, 6, 0, 5, 12, 5, 5, 6, 0, 8, 10, 9, 8, 4, 0, 7, 2, 12, 5, 6, 0, 2,
            5, 10, 5, 5, 0, 3, 4, 5, 12, 5, 6, 0, 11, 12, 8, 13, 10, 0, 7, 6, 7, 14, 12, 0, 16, 18, 30,
            51, 47, 0, 52, 50, 47, 66, 79, 0, 89, 67, 61, 45, 52, 0, 44, 59, 68, 56, 64, 0, 66, 61, 46,
            58, 51, 0, 29, 46, 26, 27, 25, 0, 34, 29, 25, 13, 18, 0, 18, 14, 11, 5, 5, 2]</gray-
am>
histogram>[1079    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              0    0    2    1    0    3    0    8    0   19   35    0  104    0  182
            293    0  282    0  301    0  237  181    0  123    0   47   57    0   32
              0   22   13    0    7    0    5    0    0   17    0    0    0    0    1
              0    1    0    3    0    1    0    0    1    0    2    3    0    0    0
              0    1    0    0    0    0    0    0    0    0    0    0    0    1    0
              0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
              1]</hue-histogram>
>
```


Figure 4 - XML document generated from rice seed numer 1

These XML files with metadata information can be used for any applications to process seeds characteristics extracted. It is can be very helpful for any purposes.

4. CONCLUSION

This study demonstrated that proposed method can be used to generate metadata information in XML document format. This kind of the metadata document can be used in various application because XML is a standard type widely used in the world.

SimpleCV framework can be used efficiently for seed image processing to read, analyse and information extraction. Python have been used widely for computer vision area, and in this works was enough to object proposed.

For future works more, seeds information can be extracted and this method can be tested with seeds from others yields.

5. REFERENCES

[1] J. E. Solem. (2012). Programming Computer Vision with Python: Tools and algorithms for analyzing images. O'Reilly Media, Inc.
[2] Lutz, M. (2013). Learning python. O'Reilly Media, Inc.
[3] K. Demaagd; A. Oliver; N. Oostendorp; K. Scott. (2012). Practical Computer Vision with SimpleCV. O'Reilly Media, Inc.
[4] L. F. de Barros Campos (2007). Metadados digitais: revisão bibliográfica da evolução e tendências por meio de categorias funcionais 10.5007/1518-2924.2007 v12n23p16. Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação, 12(23), 16-46.
[5] R. S. Ikematu (2001). Gestão de metadados: sua evolução na tecnologia da informação. DataGramaZero-Revista de Ciência da Informação, 2(6).
[6] Tanabata, T., Shibaya, T., Hori, K., Ebana, K., & Yano, M. (2012). SmartGrain: high-throughput phenotyping software for measuring seed shape through image analysis. Plant physiology, 160(4), 1871-1880.
[7] Hao, J. P., Yang, J. Z., Du, T. Q., Cui, F. Z., & Sang, S. P. (2008). A study on basic morphologic information and classification of maize cultivars based on seed image process. Scientia Agricultura Sinica, 41(4), 994-1002.
[8] Sandeep Varma, V., Kanaka Durga, K., Keshavulu, K. (2013) Seed Image Analysis: Its Applications in Seed Science Research, International Research Journal of Agricultural Science, Vol. 1(2), pp. 30-36.
[9] R. C. Gonzalez; R. E. Woods. (2008) Digital image processing. 3 ed. Cloth : Prentice Hall.
[10] PYTHON SOFTWARE FOUNDATION. (2012). The python programming language. Available at <http://www.python.org>.
[11] O. Marques Filho; H. Vieira Neto. (1999) Processamento Digital de Imagens. Rio de Janeiro: Brasport.