Comparison of texture recognition algorithms

Anonymous Submission

*Anonymous Afﬁliation*

Abstract

Texture plays an important role in human everyday life. Through textures humans can distinguish different types of objects. Texture recognition has important application in fields like remote sensing, industrial surface inspection and biomedical image analysis. Vision is the pivotal way in which textures are recognised. Most existing texture identification algorithms use greyscale images to detect textures. In this paper a texture recognition approach based on the use of Gabor filters and Local binary patterns is used in conjunction with an SVM. An existing texture identification dataset is used to perform comparative analysis and results demonstrate that local binary pattern approach has superior performance.

**Keywords:** Texture recognition, Local binary pattern, Gabor filters, Support Vector Machines

# 1 Introduction

Texture is a visual pattern of repeated elements that have some amount of variability in element appearance and relative positions. Texture is a useful cue for many image processing applications including detection and segmentation, classification, inferring object shape/orientation/deformation and synthesizing new images. Textures are normally grouped as regular, stochastic or a combination of these. Stochastic textures have high levels of variability in appearance of individual elements and in how they are placed in the image whereas regular textures have less or no variability in appearance of individual elements and in how they are placed in the image. The analysis of texture comes in various forms, namely texture estimation, texture categorisation, texture segmentation and shape from texture. Some of the general problems faced during texture identification are the semantic gap (the difference in representation of the image), viewpoint variation and illumination changes. Texture identification has numerous applications in various fields including robot grasping [1], material inspection [2], medical image analysis [3], document processing [4] and remote sensing [5]. In this work, we compare the performance of different feature extraction methods combined with machine learning for texture classification. Two feature extraction methods being evaluated are the Gabor filters and Local binary patterns approach. A Support Vector Machine (SVM) is used to train a classifier using a selection of kernels in order to classify various textures based on their features. The performance of the approaches is compared using an existing texture identification dataset.

# 2 Background and related work

An ideal texture detection system should be able to discriminate between a large range of different textures. The idea that two textures are not distinguishable if they had identical second order statistics (i.e. the distribution of intensities over pairs of pixels was identical) was proposed in [6]. The statistical theory of textons [7] replaced this idea; textons are fundamental texture primitives composed of features such as corners, intersections, line terminators, etc. Using this theory it was proven that two textures, having different texton densities, are easily discriminable. The drawbacks of the theory of textons are: a) how to formalise a list of universal textons and b) how to generalise the theory to greyscale images. Meanwhile, Bergen et al. [8] used filter banks for texture analysis to predict the difference between two textures using the responses of size tuned centre-surround filters. The advantages of filter banks are that they can be used to analyse greyscale images. Some filter banks used for texture analysis include Law’s filter masks [9], Ring and Wedge filters [10], dyadic Gabor filter bank [11], wavelet transform [12], discrete cosine transform [13], and quadrature mirror filters [14]. Leung and Malik [15] attempted to classify 3D textures under varying viewpoint and illumination however their approach was not robust to viewpoint changes. Additionally, Suen et al. [17] used correlation functions to obtain the texture across multiple colour bands. A multi-resolution greyscale and rotation invariant texture classification was performed in [18] using local binary patterns and the approach was computationally simple and satisfied the criteria of illumination invariance.

Standard texture analysis tasks such as texture recognition can also successfully use object recognition techniques. Typically, an object recognition algorithm, uses a pooling encoder to take a local descriptors of an image *x* and represents this as a single feature vector. Some of the best-known pooling encoders used are bag of visual words [16], locality constrained linear coding [19], vector of locally aggregated descriptors [20], fisher vector [21] and spatial pyramid pooling [22]. Recently, the use of CNN (convolutional neural networks) for texture classification has outperformed conventional texture analysis approaches and some of the standard CNN based methods used for texture classification are presented in [23].

# 3 Texture Feature Extraction

In order to classify textures, it is often useful to extract the underlying image features. In this section we review two commonly used texture feature extraction methods.

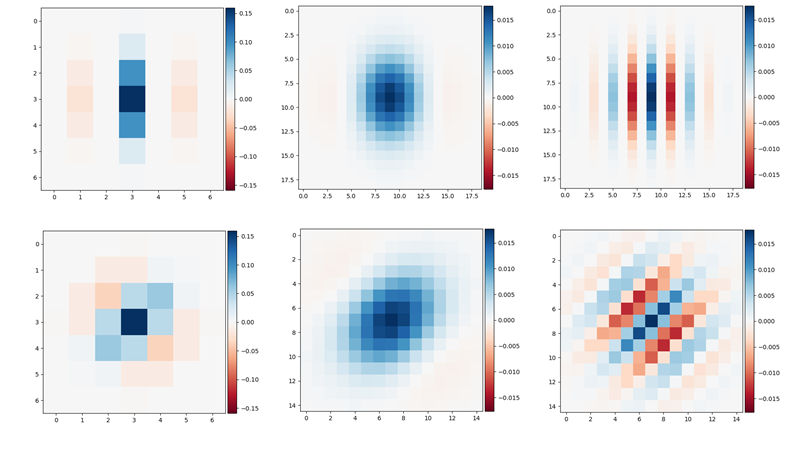
## 2.1 Gabor filtering

Gabor filtering is often claimed to have a lot of resemblance to visual processing in the human visual system [24] though there is no empirical evidence of this. Gabor filters are often used for texture analysis, which means that they basically analyse whether frequency content in the image has specific directions localized around a particular area. By applying the Gabor filter along different directions, features are smoothed in the respective direction keeping the information intact in other directions. A Gabor filter which is a Gaussian function modulated with a complex sinusoid is given as:

(1)

Equation (1) where the centre frequencies are *U* and *V* and variance is and respectively.

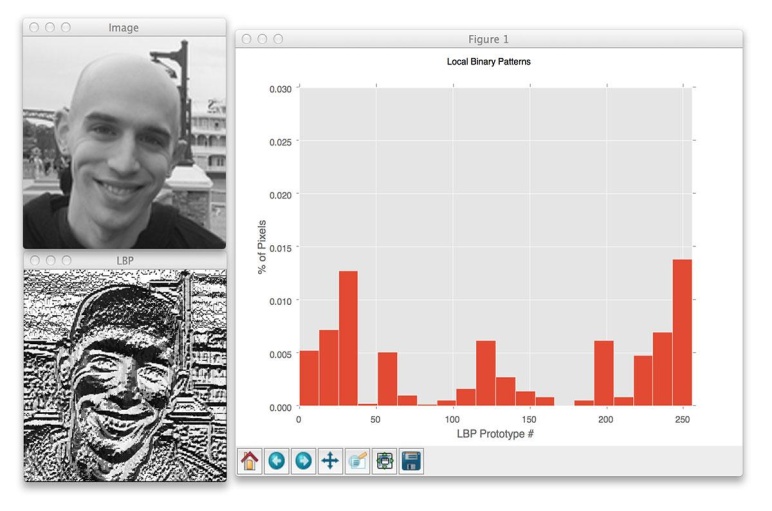
Due to the presence of the Gaussian term in Equation (1), smoothing is performed when a Gabor filter is applied to an image. The Gabor filter kernel used in this paper uses frequencies of 0.05, 0.15 and 0.25 and of 0, 0.785, 1.57, and 2.355 respectively. Here, images are filtered using Gabor filter kernels and the mean and variance of the filtered images are then used as features for the texture classification stage. The sample of Gabor filters used in the experiments is shown in Figure 1.



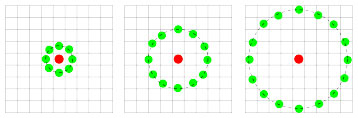
**Figure 1: Gabor filters**

## 2.2 Local binary pattern

Local Binary Patterns (LBP), compare each pixel with a surrounding neighbourhood of pixels. In LBP, the image is converted to grayscale and a neighbourhood is chosen around a centre pixel. For a given neighbourhood, if the intensity of the centre pixel is greater than the neighbouring pixel the value it is set to 1 and otherwise it is set to zero. The binary array is obtained in a clockwise or anti-clockwise direction around the centre pixel and the results are stored in an 8-bit array which is often converted to decimal. The LBP value is then stored in an array in decimal format. A histogram may then be created using these decimal values and used as an input vector to machine learning algorithms. In the experiments here, we use the histogram vector as an input into an SVM where the LBP descriptor uses 24 pixels with radius of 8 pixels from the centre pixel. Figure 2 illustrates an example image, the obtained LBP and the resulting histogram and Figure 3 illustrates the LBP radius and pixels used.



**Figure 2: Local binary pattern and resulting histogram example [25].**

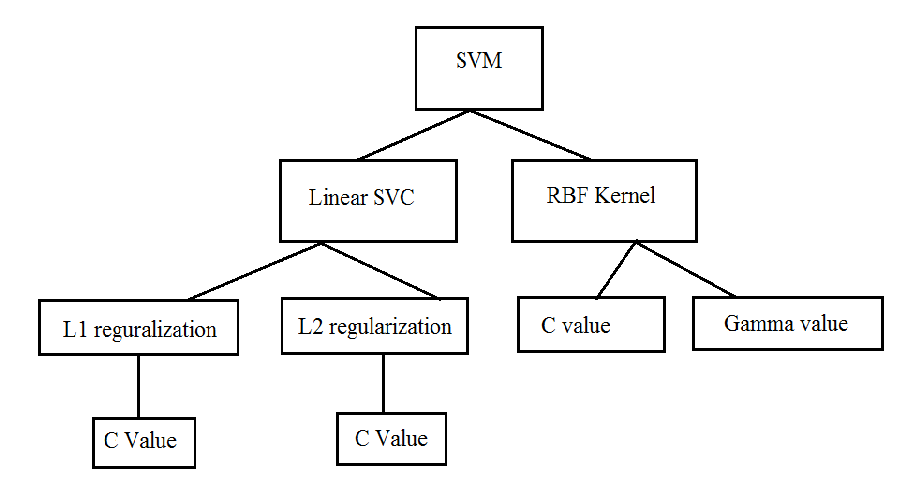


**Figure 3: Visualisation of radius and pixels used to compute LBP [26]**

# 4 Texture Classification

Here we classify the texture image data (consisting of training images and testing images) using the label lists. Each image is first converted to grayscale and the features are computed using either the Gabor filters or the LBP approach. An SVM is trained on the image feature vectors. The model is fitted using the training data and label. The training set score is then calculated. Once trained the test texture images are used and the label is predicted.

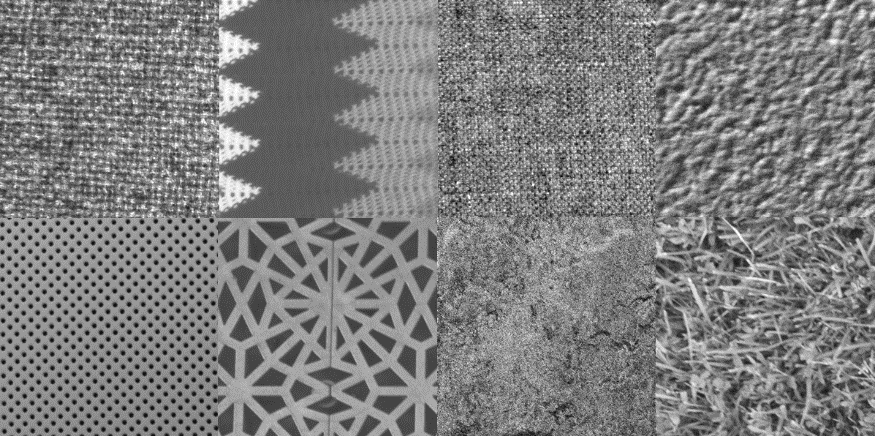
In the experiments here both Linear and Radial basis function (RBF) kernels are used. Regularisation explicitly restricts a model to avoid overfitting and this can be adjusted using the penalty parameters L1 and L2 where L2 regularisation gives sparse outputs and L1 gives non-sparse outputs. The trade-off parameter C determines the strength of the regularisation. Higher value of C fit the training data correctly whereas lower value of C adjusts to the majority of data points. Similarly, in the SVM RBF, the Γ parameter determines the scale of how close together the boundary parameters are. Small Γ values reflect smooth decision boundaries and large Γ values reflect large variance in the decision boundaries. The figure below illustrates the setup of SVM.



**Figure 4: The SVM setup**

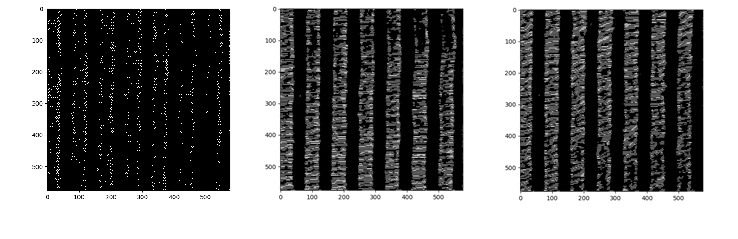
# 5 Performance Evaluation

The well-known Kylberg texture dataset which contains 28 classes of texture is used to compare performance. Some of the classes include blanket, rice, scarf and different variations of stone. Each class in the dataset contains 40 varying images of the texture. From this, 30 images are used for training and 10 images are used for testing respectively. Some samples from the 36 textures in the Kylberg database are illustrated in Figure 5.

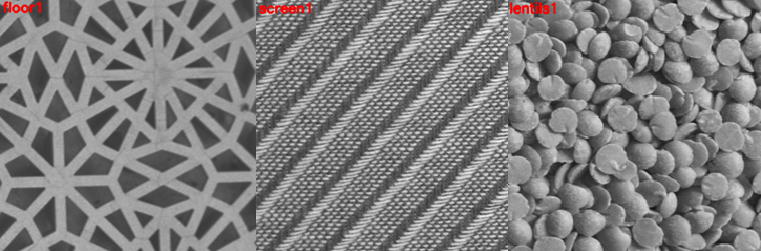


**Figure 5: Example textures from the Kylberg database (Kylberg, 2011).**

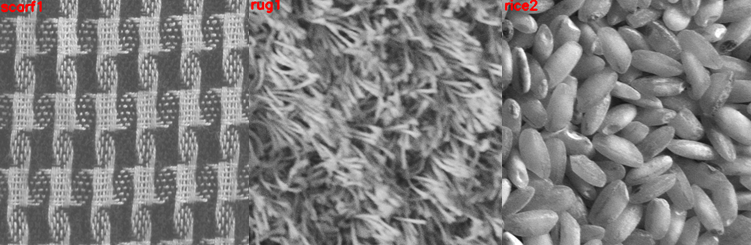
When a Gabor filter is applied to an image, for example the Scarf Image in Figure 6, it is applied at various frequencies and theta values as outlined in Section 3.1**.** Example output images obtained after Gabor filtering are illustrated in Figure 6. Similarly, the output LBP images are illustrated in Figure 7.



**Figure 6: Example Gabor filtered images.**



**Figure 7: Output images of Gabor filter**



**Figure 8: Output images of LBP**

A Support Vector Machine (SVM) is used to classify the Gabor and LBP images. In addition, the original greyscale image which has no features extracted is also used in the experiments. Both Linear kernel SVM and Radial basis function (RBF) SVM are used. In the Linear SVM, the *C* value is varied to be 100,500 and 1000 using the L2 configuration. L2 configuration known as a penalty parameter in linear support vector machines produces non-sparse outputs. Similarly, in the SVM RBF, the *C* value is varied to be 100,500 and 1000 and for each *C* value, the Γ is varied to be 0.5,0.8 and 1.0. In the experiments here the SVM with parameter *C*=1000 and Γ= 0.8 using an RBF kernel provided the highest testing accuracy of 76% and training accuracy of 92%. The Gabor filter feature extraction approach with *C*=1000 using a Linear SVM and L2 regularisation achieves the highest testing accuracy of 80.3%, with a training accuracy of 92.7%.

Experiments show that the highest testing accuracy of 98.5 % (training accuracy of 99.9%) is obtained when using LBP with Γ= 1.0 and *C*= 1000. Now using the 576\*576 dimensional configuration of the image, the SVM is trained but during testing instead of giving the texture (image) with same dimension, image with 400\*300 pixel (lower dimension) of the same texture is given and the results are checked. The testing accuracy reduces to 88.2% (training accuracy still remains 99.9%). This shows that the training is extremely specific and not generalizable. The result would not be satisfactory even if a more detailed view of the same texture is used and also if the object is present in different lighting conditions. The most common error in all the texture recognition algorithms is the recognition between sand and stone. With human eye, one could detect the difference because humans perceive texture in 3D and are able to interpret in 2D but computer systems perceive in 2D and have to interpret everything in 2D. Figure 8 shows the testing data obtained using Gabor filter, image data and local binary pattern using the different parameters of support vector machine.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Image Data** |  |  | **C=100** | **C=500** | **C=1000** |
|  |  | **C=100** | **Γ=0.5** | **Γ=0.5** | **Γ=0.5** |
| **Training data** |  | 0.726 | 0.851 | 0.893 | 0.911 |
| **Testing data** |  | 0.621 | 0.735 | 0.739 | 0.753 |
|  |  | **C=500** | **Γ=0.8** | **Γ=0.8** | **Γ=0.8** |
| **Training data** |  | 0.483 | 0.861 | 0.910 | 0.925 |
| **Testing data** |  | 0.464 | 0.742 | 0.75 | 0.760 |
|  |  | **C=1000** | **Γ=1.0** | **Γ=1.0** | **Γ=1.0** |
| **Training data** |  | 0.373 | 0.871 | 0.913 | 0.931 |
| **Testing data** |  | 0.371 | 0.742 | 0.75 | 0.739 |
|  |  |  |  |  |  |
| **Gabor Filter** |  |  | **C=100** | **C=500** | **C=1000** |
|  |  | **C=100** | **Γ=0.5** | **Γ=0.5** | **Γ=0.5** |
| **Training data** |  | 0.929 | 0.973 | 0.994 | 0.999 |
| **Testing data** |  | 0.764 | 0.771 | 0.771 | 0.767 |
|  |  | **C=500** | **Γ=0.8** | **Γ=0.8** | **Γ=0.8** |
| **Training data** |  | 0.929 | 0.982 | 0.999 | 0.999 |
| **Testing data** |  | 0.796 | 0.753 | 0.757 | 0.753 |
|  |  | **C=1000** | **Γ=1.0** | **Γ=1.0** | **Γ=1.0** |
| **Training data** |  | 0.927 | 0.993 | 0.999 | 0.999 |
| **Testing data** |  | 0.803 | 0.735 | 0.75 | 0.785 |
|  |  |  |  |  |  |
| **LBP** |  |  | **C=100** | **C=500** | **C=1000** |
|  |  | **C=100** | **Γ=0.5** | **Γ=0.5** | **Γ=0.5** |
| **Training data** |  | 0.954 | 0.994 | 0.999 | 0.999 |
| **Testing data** |  | 0.935 | 0.982 | 0.978 | 0.985 |
|  |  | **C=500** | **Γ=0.8** | **Γ=0.8** | **Γ=0.8** |
| **Training data** |  | 0.946 | 0.995 | 0.999 | 0.999 |
| **Testing data** |  | 0.935 | 0.978 | 0.982 | 0.985 |
|  |  | **C=1000** | **Γ=1.0** | **Γ=1.0** | **Γ=1.0** |
| **Training data** |  | 0.731 | 0.996 | 0.999 | 0.999 |
| **Testing data** |  | 0.732 | 0.978 | 0.982 | 0.985 |

**Table 1: Experimental results of image data, Gabor filter and local binary pattern.**

# 6 Conclusions

This work has studied comparison of texture recognition algorithm using vision and machine learning. It uses support vector machine as a machine learning technique with texture recognition algorithms like Gabor filter and local binary pattern. It has also applied support vector machines on original image data. The experimental conclusion is that the local binary pattern has the most promising results for parameters *C*=1000 and Γ= 1.0 using a Radial basis function (RBF) kernel which has the highest testing result of 98.5%, with a training of 99.9%.

# References

[1] C. Glowania, et al., “Smooth at one end and rough at the other: influence of object texture on grasping behaviour”, *Experimental Brain Research*, vol. 235, 2017

[2] P. Dewaele, L. Van Gool, A. Wambacq and A. Oosterlinck, "Texture inspection with self-adaptive convolution filters," *[1988 Proceedings] 9th International Conference on Pattern Recognition*, Rome, 1988

[3] R. N. Sutton and E. L. Hall, "Texture Measures for Automatic Classification of Pulmonary Disease," in *IEEE Transactions on Computers*, vol. C-21, no. 7, pp. 667-676, July 1972.

[4] D. Wang and S. N. Srihari, “Classification of Newspaper Image Blocks Using Texture Analysis,” *Computer Vision, Graphics, and Image Processing*, 47, pp. 327-352, 1989.

[1] C. Glowania, et al., “Smooth at one end and rough at the other: influence of object texture on grasping behaviour”, *Experimental Brain Research*, vol. 235, 2017

[5] E. Rignot and R. Kwok, “Extraction of Textural Features in SAR Images: Statistical Model and Sensitivity,” *In Proceedings of International Geoscience and Remote Sensing Symposium, pp. 1979-1982, Washington,* DC, 1990.

[6] B. Julesz, E. N. Gilbert, L. A. Shepp, and H. L. Frisch, “Inability of humans to discriminate between visual textures that agree in second-order statistics – revisited “.1973

[7] B. Julesz,” Textons, “the elements of texture perception,and their interactions””. 1981

[8] J. R. Bergen, and E. H. Adelson, “Early vision and texture perception”. 1988

[9] K.I. Laws, “Rapid Texture Identification,” *Proc. SPIE Conf. Image Processing for Missile Guidance*, pp. 376–380, 1980.

[10] J.M. Coggins and A.K. Jain, “A Spatial Filtering Approach to Texture Analysis,” *Pattern Recognition Letters*, 3(3), pp. 195–203, 1985.

[11] A.K. Jain, F. Farrokhnia, “Unsupervised Texture Segmentation Using Gabor Filters,” *Pattern Recognition*, 24(12), pp.1,167–1,186, 1991.

[12] S.G. Mallat, “A Theory for Multiresolution Signal Decomposition:The Wavelet Representation,” *IEEE Trans. Pattern Analysis and MachineIntelligence*, vol. 11, pp. 674–693, July 1989.

[13] I. Ng, T. Tan, and J. Kittler, “On Local Linear Transform and Gabor Filter Representation of Texture,” *Proc. Int’l Conf. Pattern Recognition*,pp. 627–631. Int’l Assoc. for Pattern Recognition, 1992.

[14] T. Randen and J.H. Husøy, “Multichannel Filtering for Image Texture Segmentation,” *Optical Eng.*, vol. 33, pp. 2,617–2,625, Aug.1994.

[15] T. Leung, and J.Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons”. 2001. *International Journal of Computer Vision*,

[16] P. Suen, and G. Healey, “The analysis and reconstruction of real-world textures in three dimensions”. 2000, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):491-503.

[17] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, Jul 2002.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, "Locality-constrained Linear Coding for image classification," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010

[19] H. Jégou, M. Douze, C. Schmid and P. Pérez, "Aggregating local descriptors into a compact image representation," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010

[20] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007

[21] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006

[22] F. H. C. Tivive and A. Bouzerdoum, "Texture Classification using Convolutional Neural Networks," TENCON 2006 - 2006 IEEE Region 10 Conference, Hong Kong, 2006

[23] V. Andrearczyk and P. F. Whelan, “Texture segmentation with fully convolutional networks,” arXiv preprint arXiv: 1703.05230, 2017.

[24] Daugman, J.G. (1980), "Two-dimensional spectral analysis of cortical receptive field profiles", Vision Res., 20 (10): 847–56

[25] <https://www.pyimagesearch.com/2015/12/07/local-binary-paerns-with-python-opencv/>