# pyHIVE, a Health-related Image Visualization and Engineering system using Python

Ruochi Zhang1, Ruixue Zhao1, Xinyang Zhao1, Di Wu2, Weiwei Zheng1, Xin Feng1, Fengfeng Zhou1,\*.

1 College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin, China, 130012, 2 College of Software, Jilin University, Changchun, Jilin, China, 130012.

\*To whom correspondence should be addressed at: FengfengZhou@gmail.com or ffzhou@jlu.edu.cn. Web site: http://www.healthinformaticslab.org/.

## Supplementary Text 1 – Implementation details

pyHIVE implemented five widely-used image feature extraction algorithms, *i.e.*  Histogram of Oriented Gradient (HOG), Local Binary Pattern (LBP), Gray-level Co-occurrence Matrix (GLCM), Hessian Matrix (HEM) and Canny (CAN), using the programming language Python. HOG, LBP and GLCM are three widely-used image features to describe textures [1-4]. HEM and CAN work well on object topology detections in images [5, 6]. A post-processing module using Principal Component Analysis (PCA) was also provided. The eigenvalues of the aforementioned five algorithms generate different value scales, and the user may choose to normalize the data using an embedded normalization module.

pyHIVE also parallelized the calculations of the aforementioned image feature extraction algorithms. The calculation of image features has an inherent nature of O(*n*×*m*) time complexity, where *n* and *m* are the width and height of an image, respectively. Some image feature extraction algorithms have been implemented using interpretive languages like MatLab, and may run for minutes or even hours to extract features of a large image. It may become intolerable in real-time analysis situations or for thousands of biomedical images. Python was chosen to implement pyHIVE for its fast running and a large repository of data mining modules. pyHIVE may fully utilize the computing power of multi-core architecture of the modern servers. Due to that the Windows operation system does not provide a strong support for the parallel programming interface in Python, the Windows version of pyHIVE does not support parallelized calculation. It has a time cost to switch between computing processes in the same CPU core, so it is recommended that the number of parallel tasks may be set to the number of CPU cores in the user server.

pyHIVE has been conceived and implemented as a user-friendly image feature extraction system. So pyHIVE has been tested using both Python v2 and Python v3 in all the three main operation systems, *i.e.* Linux, Mac OS and Windows. pyHIVE accepts 30+ image file formats as input, *e.g.* BMP, EPS, GIF, JPG, PNG, PPM, and SGI, etc. The calculated features may be saved as one or more choices of the six supported file formats, *i.e.* CSV, PICKLE, JSON, EXCEL, TXT and SQL. The user may decide the number of fractional digits of the decimal values of the calculated image features. A larger file size and a longer saving time will be needed if a larger number of fractional digits is chosen.

Due to the many parameters that a user may want to manipulate, pyHIVE defined a configuration file to help the user tune how pyHIVE works for different experiment requirements. All the aforementioned functionalities may be tuned by parameters in the configuration file. A default value was also provided for these parameters, so that a user may run pyHIVE with a minimum parameters.

## Supplementary Text 2 – Dataset details

We demonstrated how to use pyHIVE by a public dataset of gastrointestinal endoscopic images. The endoscopic images are publicly available at the El Salvador Atlas of Gastrointestinal Video Endoscopy, as similar in [7]. 243 images were randomly captured from the 16 endoscopic videos of normal participants, and each is 1280×720 pixels in size. These were regarded as the negative images. There are 17 gastric polyp videos, 26 gastric ulcer videos and 10 gastritis videos available at the database. A random image capturing step generated 158, 99 and 74 images from these three groups of videos, respectively. These 331 (=158+99+74) images were regarded as the positive images, and are 352×240 pixels in size. The negative images were scaled to the same size of the positive ones, which is a user-defined parameter of pyHIVE.

## References

1. Adetiba E, Olugbara OO: **Lung cancer prediction using neural network ensemble with histogram of oriented gradient genomic features**. *The Scientific World Journal* 2015, **2015**.

2. Unay D, Ekin A, Cetin M, Jasinschi R, Ercil A: **Robustness of local binary patterns in brain MR image analysis**. In: *Engineering in Medicine and Biology Society, 2007 EMBS 2007 29th Annual International Conference of the IEEE: 2007*. IEEE: 2098-2101.

3. Nanni L, Lumini A, Brahnam S: **Local binary patterns variants as texture descriptors for medical image analysis**. *Artificial intelligence in medicine* 2010, **49**(2):117-125.

4. Vallieres M, Freeman C, Skamene S, El Naqa I: **WE-C-WAB-02: Joint FDG-PET/MR Imaging for the Early Prediction of Tumor Outcomes**. *Med Phys* 2013, **40**(6Part29):477.

5. Tang Z, Zhang J, Gui W: **Selective Search and Intensity Context Based Retina Vessel Image Segmentation**. *J Med Syst* 2017, **41**(3):47.

6. Yang D, Wang X, Duan Y, Tan J, Mutic S: **SU-E-J-76: 3D Soft Tissue Boundary Detection for Automatic Verification of Deformable Image Registration**. *Med Phys* 2013, **40**(6Part7):167.

7. Tursi T: **Use of ss-blocker therapy to prevent primary bleeding of esophageal varices**. *J Am Acad Nurse Pract* 2010, **22**(12):640-647.