

Slice The Image And Predict

- In some large images, it is necessary to crop them into small ones before making predictions, which will increase the accuracy of the prediction. In this example, it shows how to crop the image first, predict the results, and then put the results back in the original large image.
- In this example, the slicing is done equally in N rows and N columns. In addition, you can implement the desired slicing operation yourself. The function interface is as follows

```
int dividePicture(vector<smallPic *> &iMat, cv::Mat& image, int row, int col)
```

1 : Class small Pic

```
class smallPic
{
public:
    int row_begin, col_begin;           // the position in the big picture.
    cv::Mat image;                      // the image
    int height, width;                  // the height and width of the image
    vector<DetectionResult> res;        // the result after prediction
    smallPic(int row, int col, cv::Mat img, int height, int width)
    :row_begin(row), col_begin(col), image(img), height(height), width(width) {};
    bool setDetectionResult(vector<DetectionResult> result)
    {
        res = result;
        return true;
    }
};
```

- This class describes the small image information after a large image is segmented, including the actual position of the small image in the large image

2 : Class BigPic

```
class BigPic
{
public:
    string fileName;                   // the fileName of the big picture
    cv::Mat imageMat;                  // the image Mat object
    vector<smallPic *> child;           // all of his child
    vector<DetectionResult> res;       // the result after prediction
    BigPic(string file_name, cv::Mat image_mat)
    {
```

```

        this->fileName = file_name;
        this->imageMat = image_mat;

    }
    BigPic(string file_name, cv::Mat image_mat, vector<smallPic *> child)
    {
        this->fileName = file_name;
        this->imageMat = image_mat;
        this->child = child;
    }
};

```

- This class describes a large image information, including all of its children and the predicted results of this image.
- After all children are predicted, the predicted results in all children's images will be restored in the large image.

3 : dividePicture

- This is the core of slice-and-dice prediction, which in this example is done using N rows and N columns of uniform slices.
- In addition to equalizing, you can use your own way of slicing, such as by the size of the pixel values, or at 1/3. 1/5.

```

int dividePicture(vector<smallPic *> &iMat, cv::Mat& image, int row, int col)

```

Appendix

1 : Load Error Code Definitions:

status	Definitions
0	Success
1	Configuration file not found
2	Batch parameter not found
3	dete_thres parameter not found
4	class parameter not found
5	Missing required DLL for execution
6	ARG - NVIDIA graphics card error
7	Not supported
8	Out of bounds
9	Insufficient memory

status	Definitions
10	Missing model-related files for loading
11	Load failed
12	Status count
50	Model type error or unable to read model.conf file

2 : Predict Error Code Definitions:

status	Definitions
0	Success
1	Configuration file not found
2	Batch parameter not found
3	dete_thres parameter not found
4	class parameter not found
5	Missing required DLL for execution
6	ARG - NVIDIA graphics card error
7	Not supported
50	Authorization file does not exist or permission to read is denied
51	Insufficient memory
52	Missing model-related files for loading
53	Load failed
54	Status count
55	Model type error or unable to read model.conf file
56	No write permission for the authorization file
57	Unknown error related to authorization
58	Authorization file version is different from SDK version
59	Exceeded the limit of the number of labels