Algorithm 1: The calculating and hyperparameter adjusting algorithm of A_r.

Given Edge detection results $D^0 = \{X_1, X_2, \dots, X_i, \dots\}$, $X_i \in \boldsymbol{R}^{N\times C\times H\times W}$, where D^0 is an image dataset, N denotes the batch axis, C denotes the channel axis, H and W represent the spatial height and width axes, respectively;
The binary image X_i' is generated by using the threshold α to binarize X_i, where α is a hyperparameter;
for j←1 **to** number of images in D^0 **do**
    Perform two erosion operations on X_j' using the Conv operation to eliminate small areas. Subsequently, apply an expansion operation to fill any remaining empty regions. Finally, identify two outlines in the image and record their exact locations using two arrays $[]_1$ and $[]_2$.
    Define S_1, S_2 as the area based on first and second contour circle, respectively. To calculate A_r←$S_2/S_1 \times 100\%$ (S_1 or S_2=cv2.contourArea($[]_1$ or $[]_2$);
end
The annotation information of X_i and A_r calculated by X_i' were compared, and four hyperparameters were defined as the judgment threshold, they are a, b, c, d, respectively. The adjustment of the hyperparameters and pre-training are started.
for j←1 **to** number of images in D^0 **do**
    if A_r of X_j < a **then**
        Write tuyere state represented by X_j as "Pulverized coal lower";
    elseif A_r of X_j < b **then**
        Write tuyere state represented by X_j as "Normal";
    elseif A_r of X_j < c **then**
        Write tuyere state represented by X_j as "Leaking";
    elseif A_r of X_j < d **then**
        Write tuyere state represented by X_j as "Hanging slag";
    else
        Write tuyere state represented by X_j as "Irrigation slag";
    endif
end
Compare the status value of the subsequently writing with the annotation content, repeat steps 8~13, until the accuracy is greater than 90%.