

```

def CVIONS_FAST(image, LoG_sigma=1.5, threshold=0.3):
    # some computer vision algorithms
    image_log = gaussian_laplace(image, LoG_sigma)
    image_log = 1 - (image_log - np.min(image_log)) / (np.max(image_log) - np.min(image_log))
    image_bn = np.where(image_log > threshold, 1, 0)
    image_lb = label(image_bn)

    # label ions
    lb_num = np.max(image_lb)
    para_num = 4
    para_cv = np.zeros((lb_num, para_num))
    for lb in range(1, lb_num + 1):
        count = image[image_lb == lb]
        wgt = count / np.sum(count)
        y, x = np.where(image_lb == lb)
        max_count = np.max(count)
        x0 = np.average(x, weights=wgt)
        y0 = np.average(y, weights=wgt)
        s = np.count_nonzero(image_lb == lb)
        para_cv[lb - 1, :] = [max_count, x0, y0, s]

    # calculate ion positions
    order = np.argsort(para_cv[:, 1])
    for i in range(para_num):
        para_cv[:, i] = para_cv[order, i]
    return para_cv

```