Skoltech

Final project

CORE MICROPROBE ANALYSIS

made by
Elizaveta Gladchenko and Ramilya Sharifullina

Why is it important (education)?

Porosity is one of the key parameters of reservoir rock that should be identified for calculation of the amount of oil or gas that can be found in the field and then obtained by different production techniques. Besides, not only the value of porosity that matters.

What's the expected value of our CV algorithm?

Identification of tiny pore spaces between grains of the rock, estimation of their form and type of porosity in the sample

Dataset description

Microprobes of reservoir rock samples and real values of porosity

	porosity in each of five images
img1.jpg	6.80
img2.jpg	2.00
img3.jpg	0.41
img4.jpg	28.70
img5.jpg	5.04

The first method – mask by “hsv” + calculate by pixels

Step 1 - original image

Step 2 – create the mask

```python
# function to estimate pore space
def por_by_pixels1(image):
    por_pred = []
    blackpixels = 0
    whitepixels = 0
    for col in range(0, img.shape[0]):
        for row in range(0, img.shape[1]):
            if image[col,row] == 1:
                whitepixels += 1
            else:
                blackpixels += 1
    porosity = (whitepixels/(img.shape[0]*img.shape[1]))*100
    por_pred.append(porosity)

    return por_pred
```

```python
porosity_1 = por_by_pixels1(mask)
```

Step 3 – calculating the pore space

The second method – mask by “hsv” + contours

Step 1 - original image

Step 2 – contours

```
def por_by_contours(image, mask):
    por_pred = []
    mask_int = mask.astype(np.uint8)
    contours, hierarchy = cv2.findContours(mask_int, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    img_cnt = image.copy()
    cv2.drawContours(img_cnt, contours, -1, (255,0,0), 4)

    allporeareas = []

    for contour in contours:
        area = cv2.contourArea(contour)
        allporeareas.append(area)

    pore_area = np.array(allporeareas).sum()
    porosity = (pore_area/(img.shape[0]*img.shape[1]))*100
    por_pred.append(porosity)
    return por_pred
```

```
porosity_2 = por_by_contours(orig, mask)
```

Step 3 – calculating the pore space

The second method – pore type

```python
def get_pore_shapes(contours):
    pore_shapes = []
    for cnt in contours:
        perimeter = cv2.arcLength(cnt, True)
        epsilon = 0.01 * perimeter
        approx = cv2.approxPolyDP(cnt, epsilon,True)
        if len(approx) == 3:
            shape = "triangle"
        elif len(approx) == 4:
            area = cv2.contourArea(approx)
            perimeter = cv2.arcLength(approx, True)
            ar = area * 16 / perimeter**2
            shape = "square" if ar >= 0.95 and ar <= 1.05 else "rectangle"
        else:
            shape = "circle"
        pore_shapes.append(shape)
    return pore_shapes
```

```python
shape = get_pore_shapes(contours)
```

Figure – count plot from seaborn

The third method – mask by “hsv” + contours

Step 1 - original image

Step 2 – markers

Step 3 – segmentation

Step 4 – calculation pore space (like in the first method)

Estimate quality metrics (“rmse”) and comparing

the first method

the second method

the third method

```python
def get_rmse(real_porosity, por_pred):
    real_porosity = [real_porosity]
    rmse = mean_squared_error(real_porosity, por_pred)
    return rmse
```

Code for “rmse”

➡

For the img2.jpg we have:

	rmse
mask	1.782258
contours	3.133840
segmentation	0.038179

Results

CV pipeline suitable to solve the problem – pipeline with contours detection

- good results
- extra useful information

Motivation for each step of the pipeline

- hsv color space – clear pore spaces
- the mask - more precise contours
- contours - calculation of the area of pores
- approximating contours of pores, dividing them into different shapes and getting the most common one - type of porosity

The same for another image – good one

the first method

the second method

the third method

For the img3.jpg we have:

	rmse
mask	0.511847
contours	0.187712
segmentation	0.047610

Bad example

the first method

the second method

the third method

For the img1.jpg we have:

	rmse
mask	0.150602
contours	13.273196
segmentation	0.019823

What else we have tried

1. CNN

2. Working with images in gray colored space

original image

result