Адаптация изображений аншлифов, полученных в разных условиях съемки, в задаче сегментации минералов

Индычко Олеся Игоревна

Научный руководитель:
к.ф-м.н., м.н.с.
А.В. Хвостиков

https://imaging.cs.msu.ru/

Laboratory of Mathematics Methods of Image Processing
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University

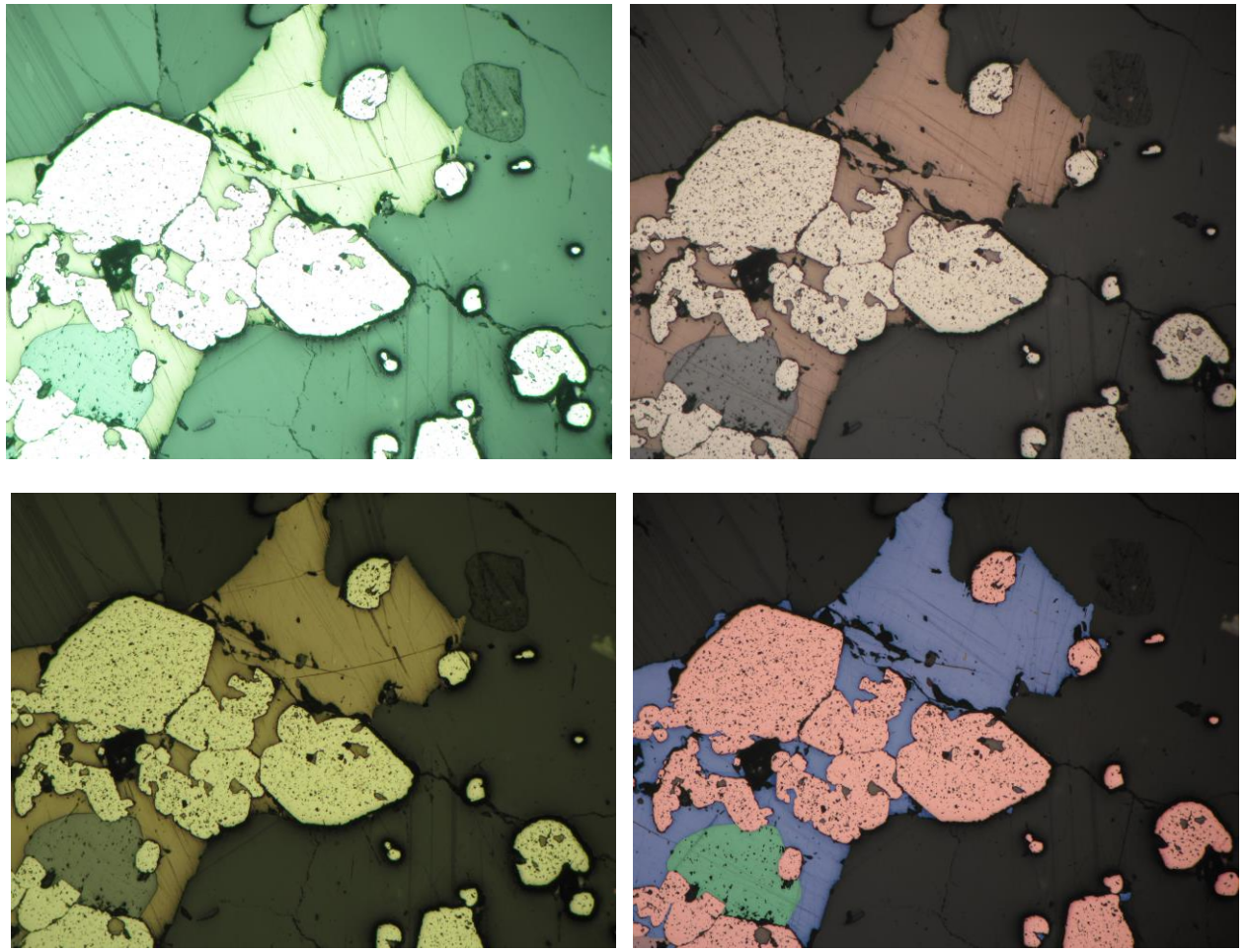
2022

Предметная область

Аншлиф – специально подготовленный образец горной породы.

Задача компьютерного зрения – **автоматическая сегментация минералов**.

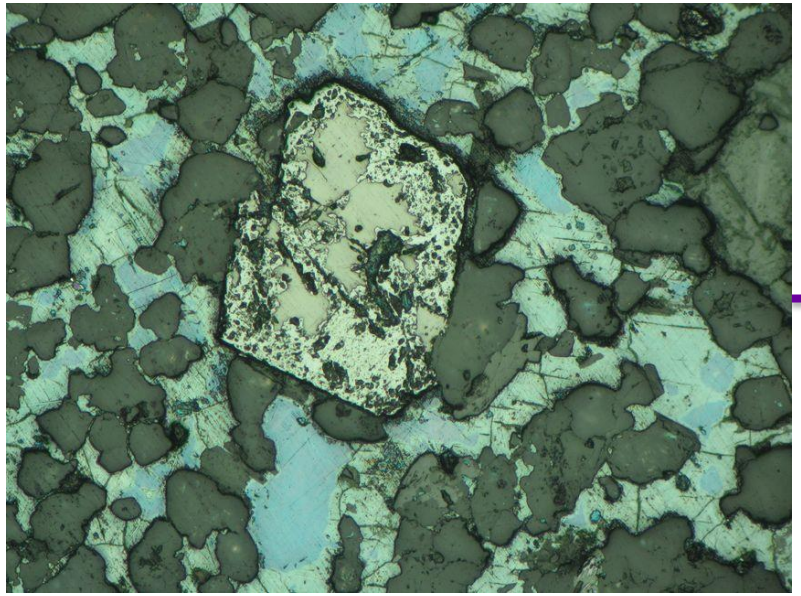
Сложность задачи: нестабильная работа алгоритма на изображениях, сильно отличающихся от обучающей выборки.



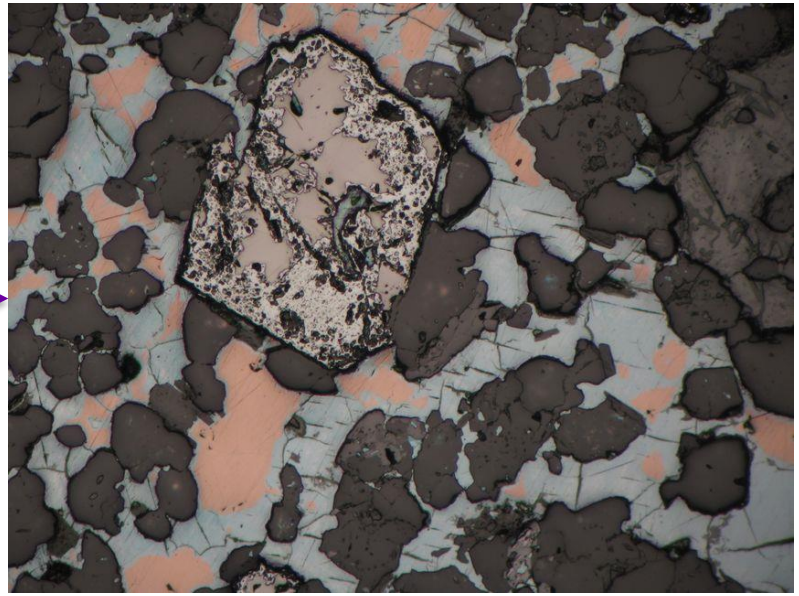
Примеры изображений из набора данных LumenStone

Постановка задачи

Разработать **алгоритм приведения изображений аншлифов**, полученных с разных микроскопов, камер и сделанных при разных условиях съемки, **к референсному виду** изображений (т.е. тех, на которых обучался алгоритм сегментации).



Входное изображение

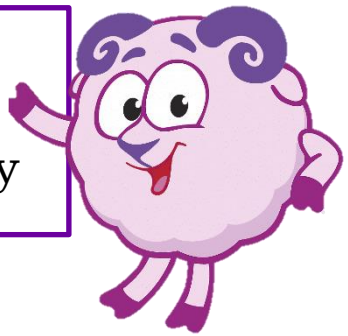


Изображение, преобразованное к референсному виду

План работы

1. Получение **набора данных**, содержащего размеченные референсные изображения и вариации этих изображений в различных условиях съемки

2. Разработка **алгоритма**, реализующего
   - совмещение изображения с референсным,
   - приведение цветовых распределений изображения к референсному

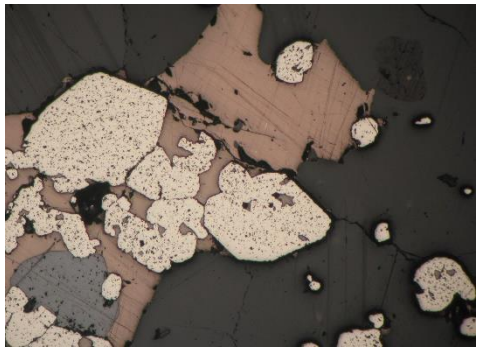


3. Адаптация алгоритма для **software калибровки**

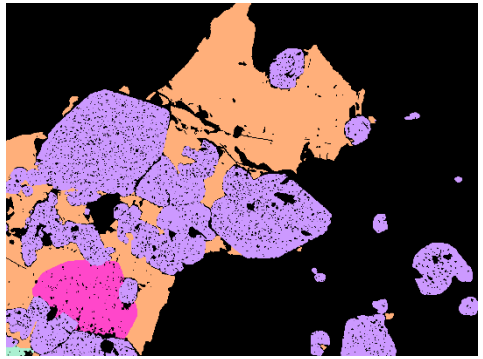
4. Адаптация алгоритма для **hardware калибровки**

Идея работы алгоритма адаптации

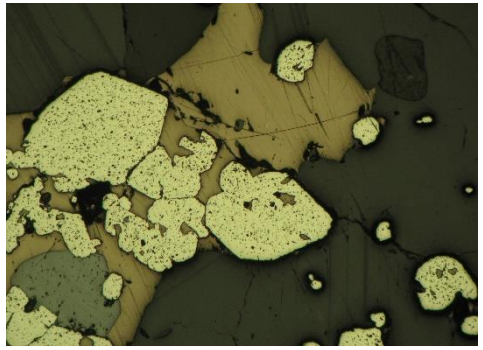
Входные данные:



Референсное изображение

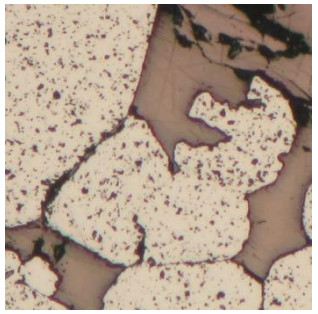


Маска сегментации

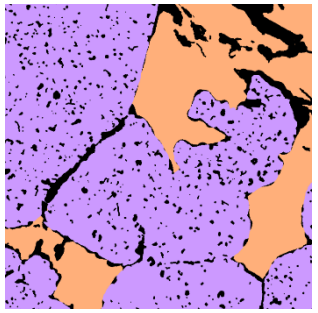


Искаженное изображение

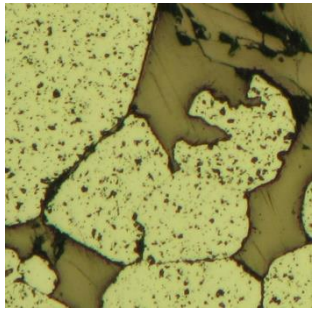
Совмещение фрагментов:



Фрагмент референсного



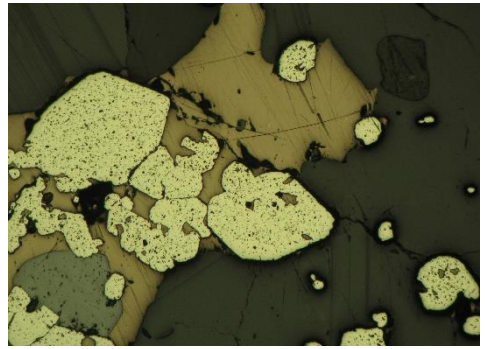
Фрагмент маски



Фрагмент искаженного

Получение цветовых распределений

Цветовая коррекция:



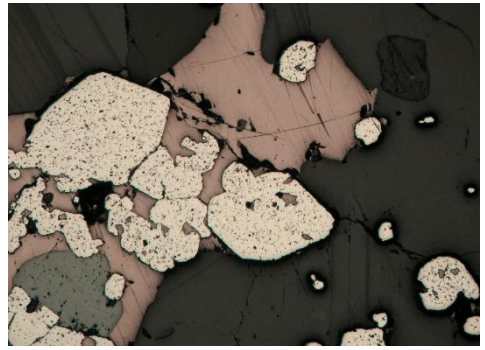
Искаженное изображение

×

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Матрица цветовой коррекции

=



Адптированное изображение

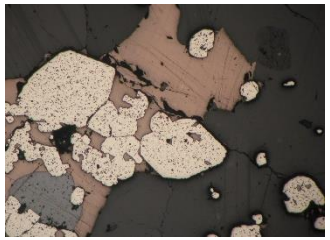
Совмещение фрагментов

Входные данные

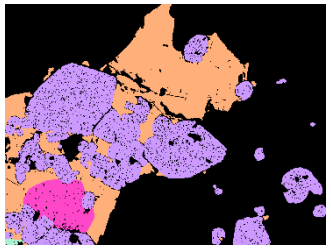
Выбор фрагмента 800x800, содержащего наибольшее число минералов

Совмещение фрагмента референсного с искаженным изображением

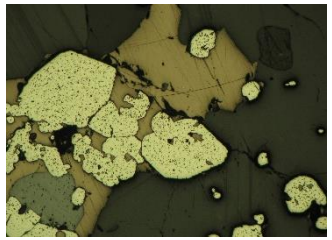
Применение преобразований и кадрирование изображений



Референсное изображение

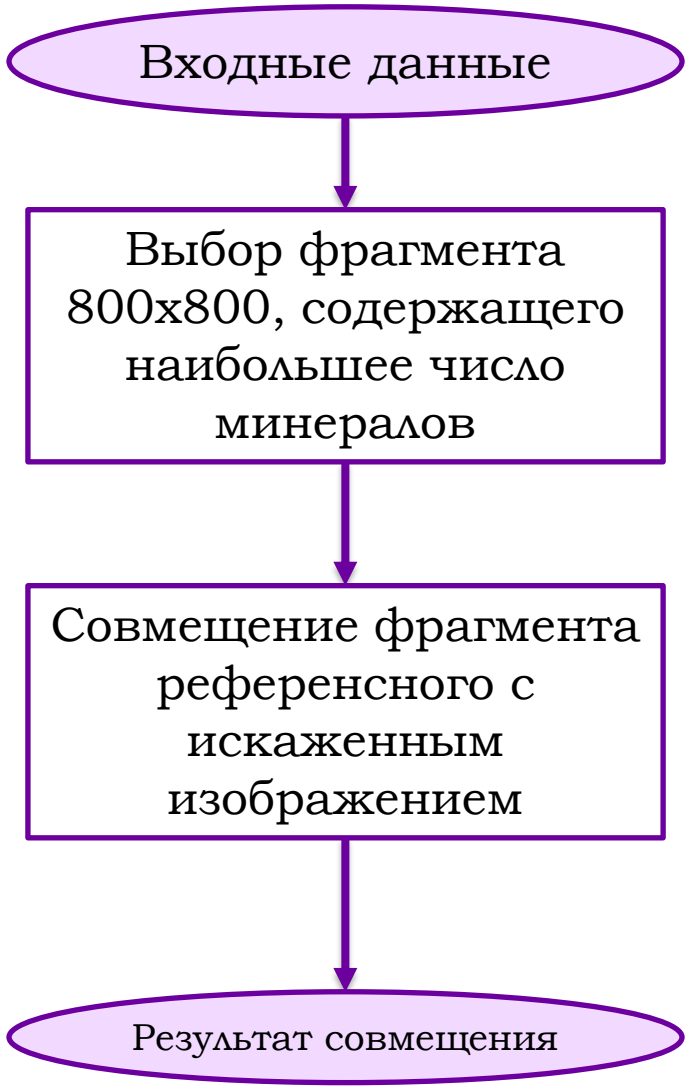


Маска сегментации

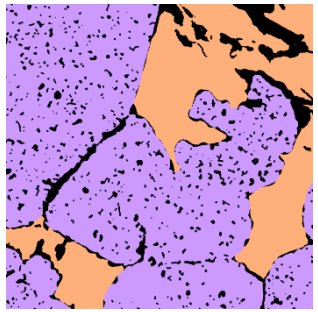


Искаженное изображение

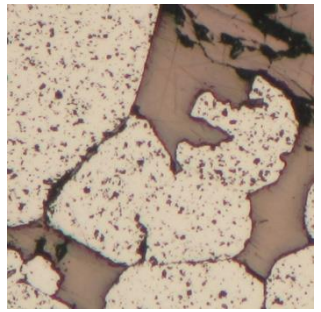
Совмещение фрагментов



По маске сегментации ищем фрагмент, на котором наибольшее число минералов и вырезаем его на маске и референсном изображении

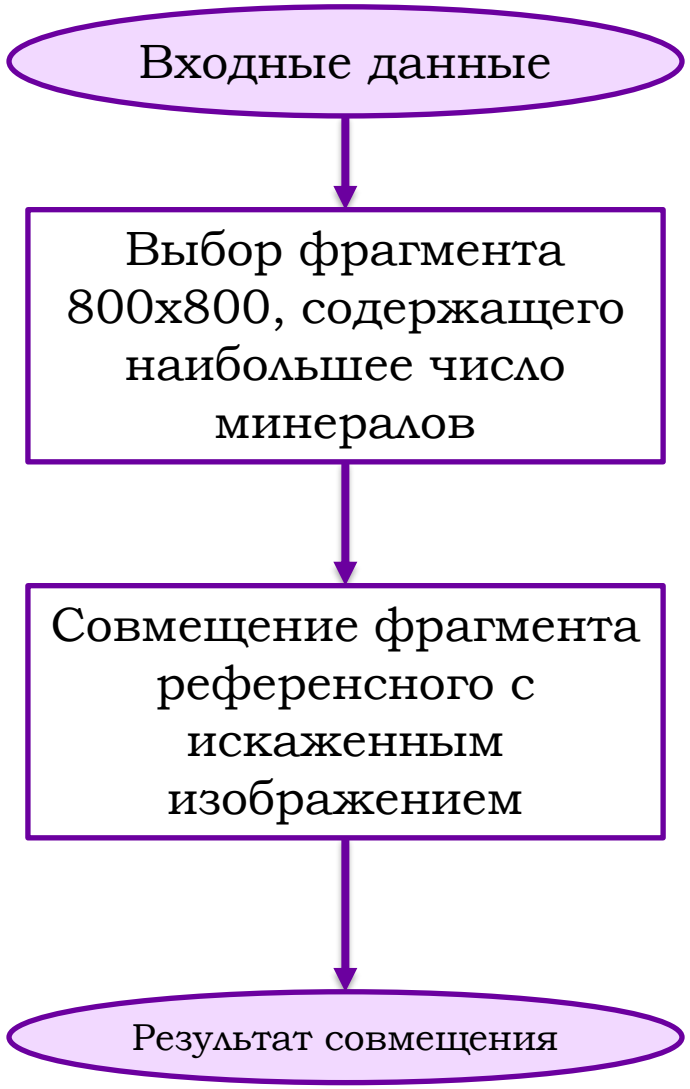


Фрагмент маски

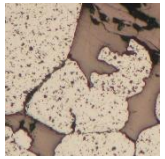


Фрагмент референсного

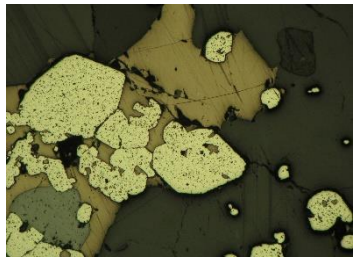
Совмещение фрагментов



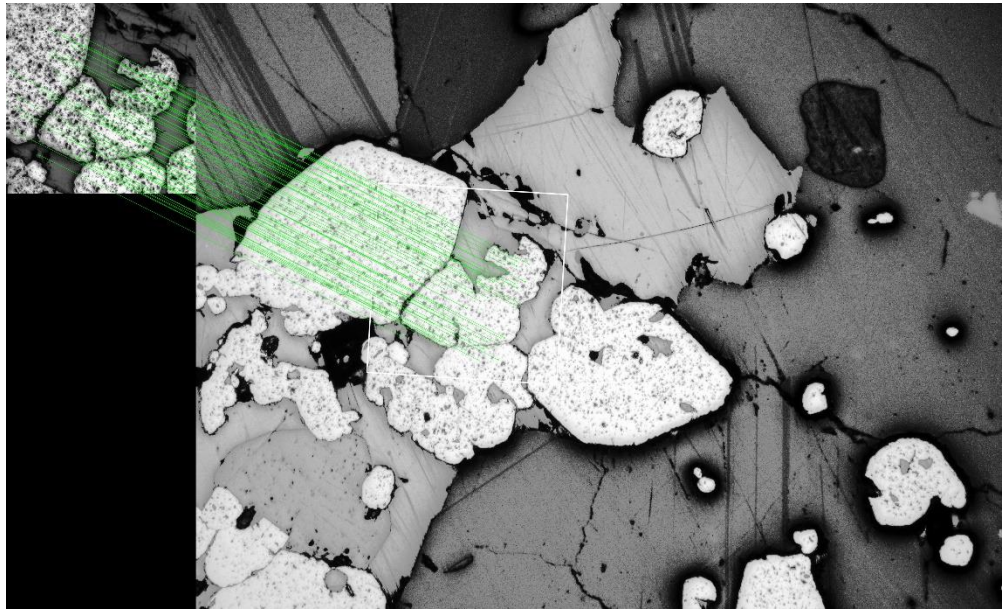
Используя **SIFT** для поиска ключевых точек и **FlannBasedMatcher** для их сопоставления, совмещаем фрагмент референсного с искаженным



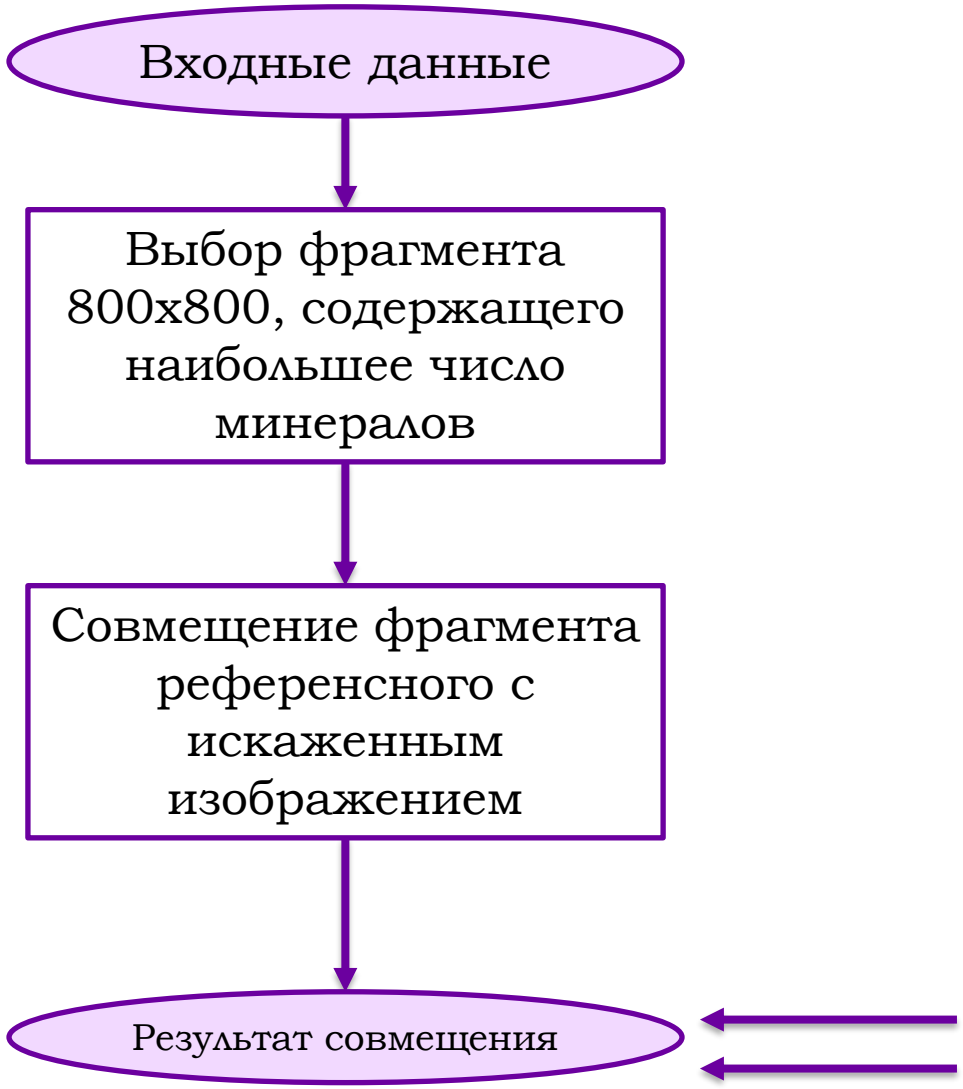
Фрагмент референсного



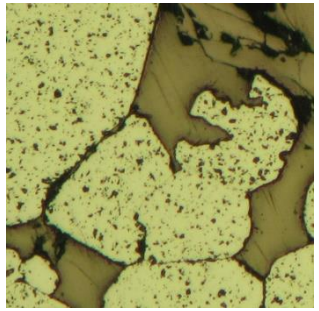
Искаженное изображение



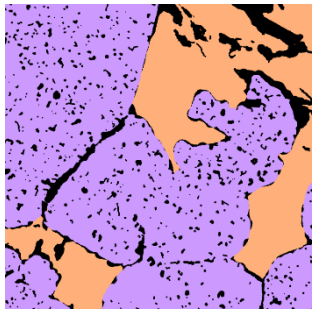
Совмещение фрагментов



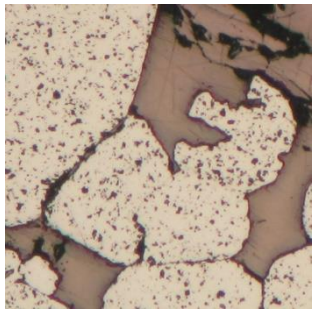
Применение преобразований и кадрирование изображений



Фрагмент искаженного

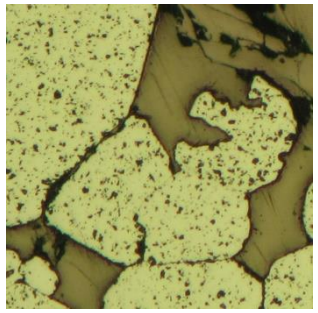


Фрагмент маски

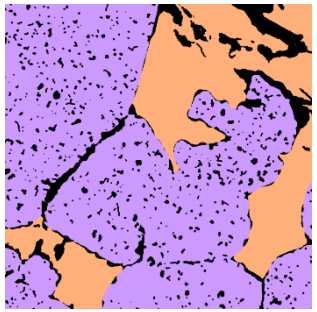


Фрагмент референсного

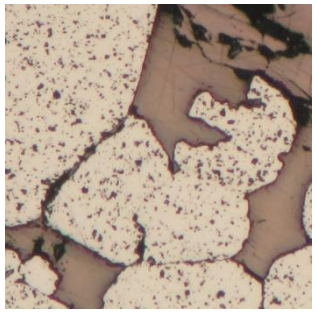
Получение цветовых распределений



Фрагмент искаженного



Фрагмент маски



Фрагмент референсного

$$CS' = \begin{bmatrix} R'_1 & G'_1 & B'_1 \\ R'_2 & G'_2 & B'_2 \\ R'_3 & G'_3 & B'_3 \end{bmatrix}$$

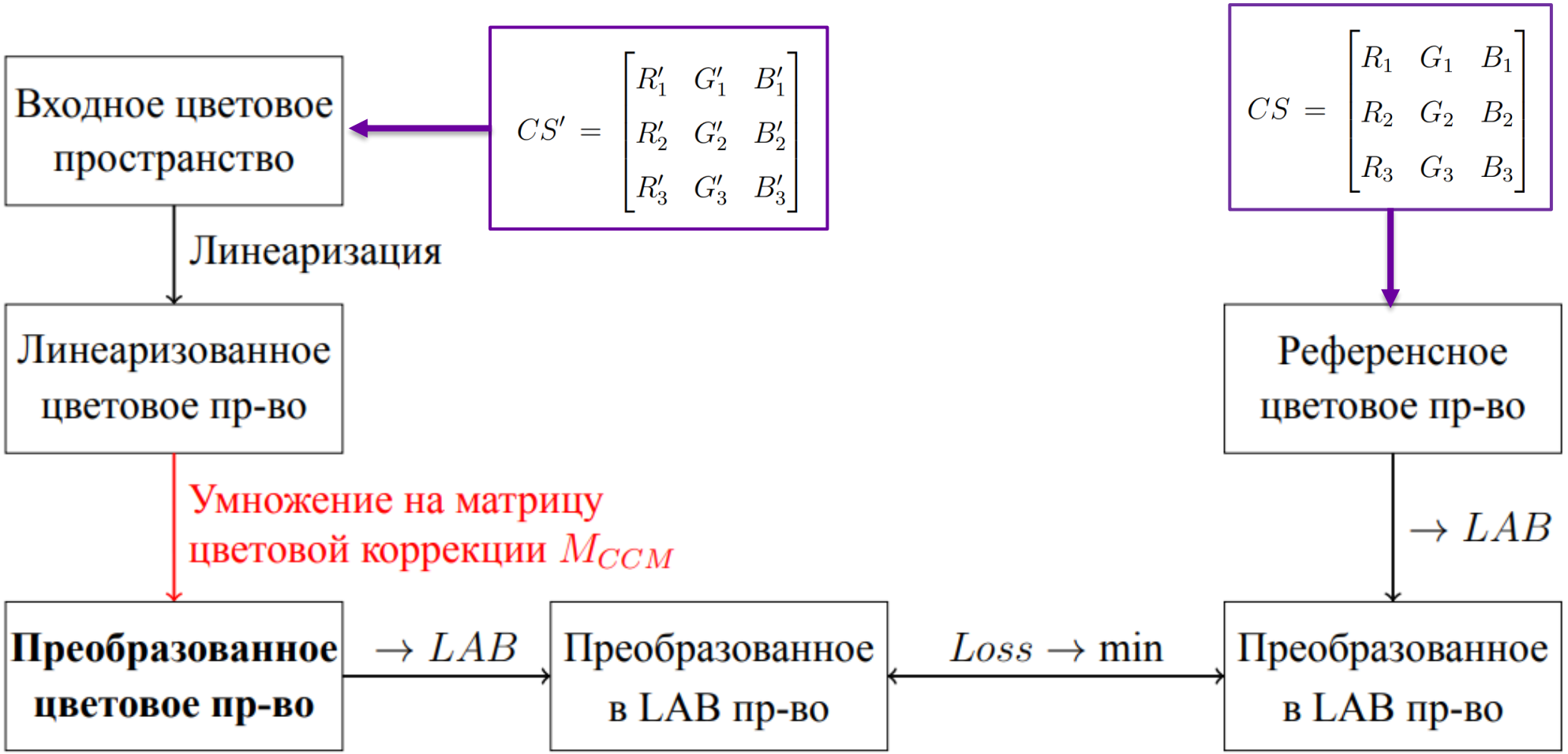
$$CS = \begin{bmatrix} R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 \\ R_3 & G_3 & B_3 \end{bmatrix}$$

1. По фрагменту маски получаем количество N минералов. Искомое цветовое пространство — матрица размера $N \times 3$.

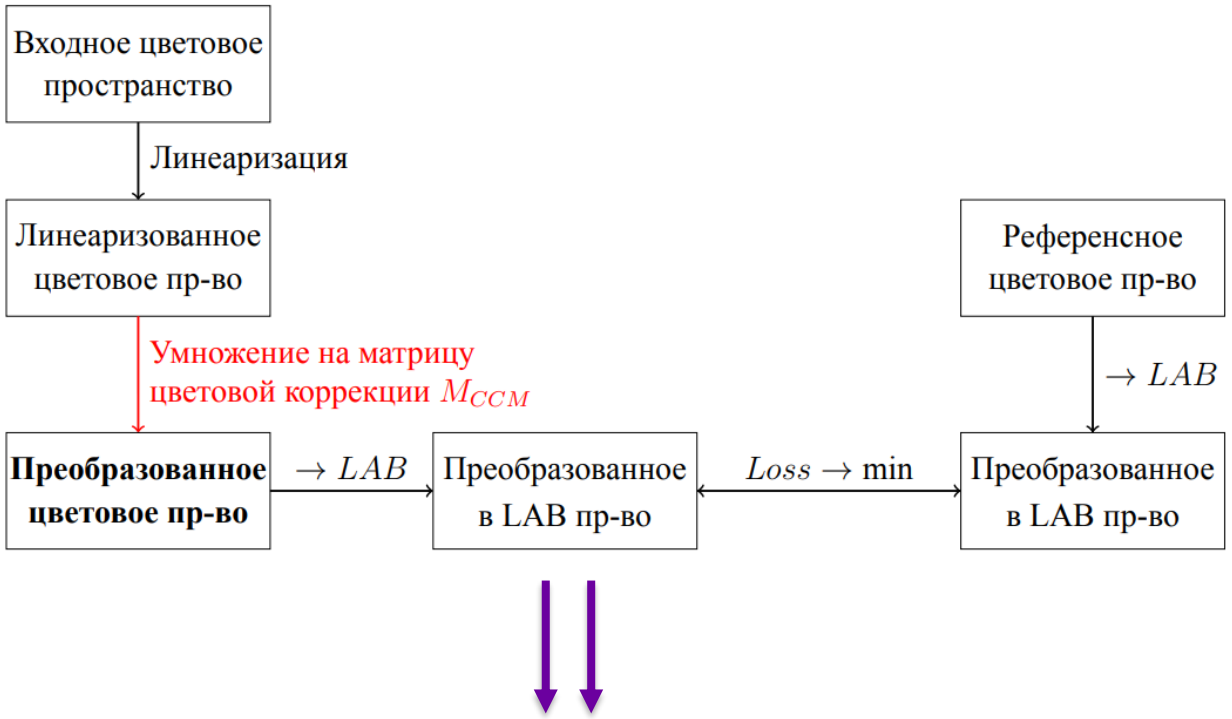
2. Для каждого минерала получаем три значения R_i, G_i, B_i — среднее значение красной, зеленой и синей компонент.

3. Извлекаем цветовые пространства из искаженного и референсного фрагментов.

Калибровка алгоритма цветовой коррекции



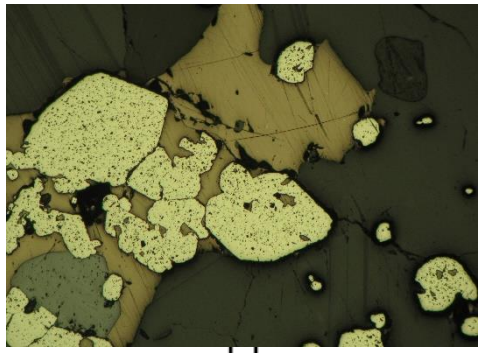
Калибровка алгоритма цветовой коррекции



Получили матрицу цветовой коррекции,
решив задачу минимизации

$$M_{CCM} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Применение алгоритма цветовой коррекции



$$M_{CCM} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Входное цветовое пространство

Линеаризация

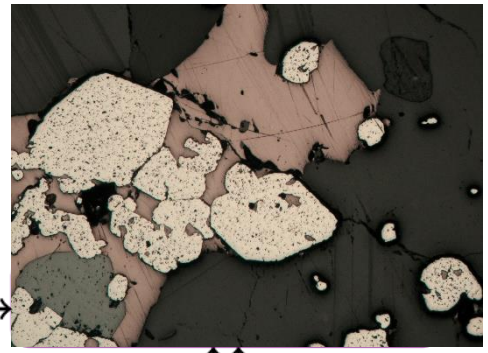
Линеаризованное цветовое пр-во

$\times M_{CCM}$

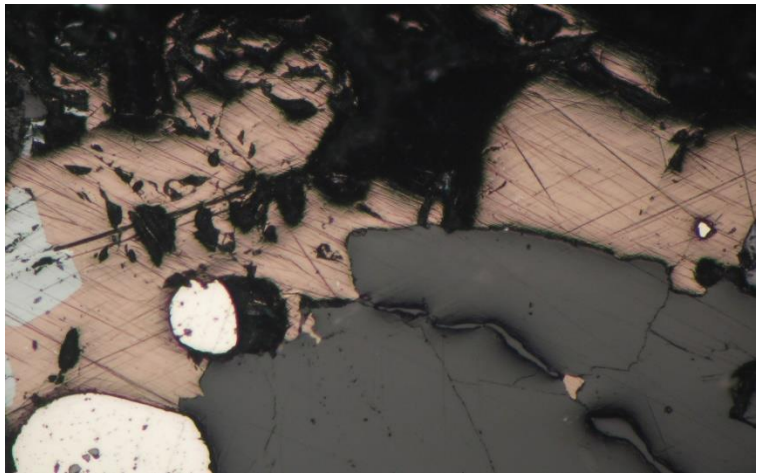
Преобразованное цветовое пр-во

Делинеаризация

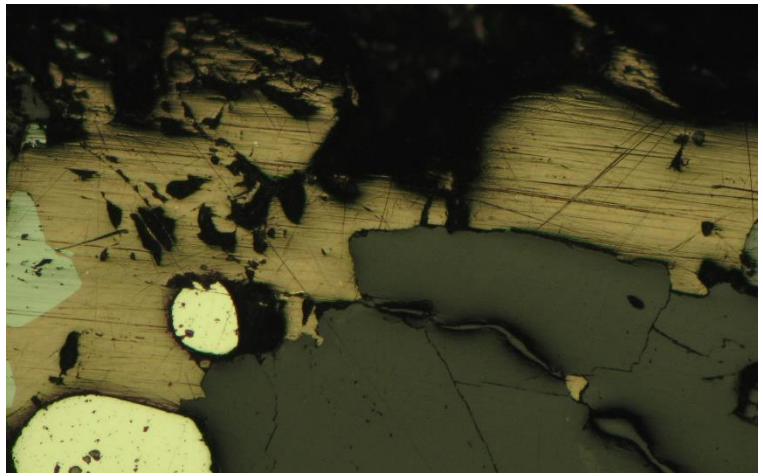
Выходное цветовое пр-во



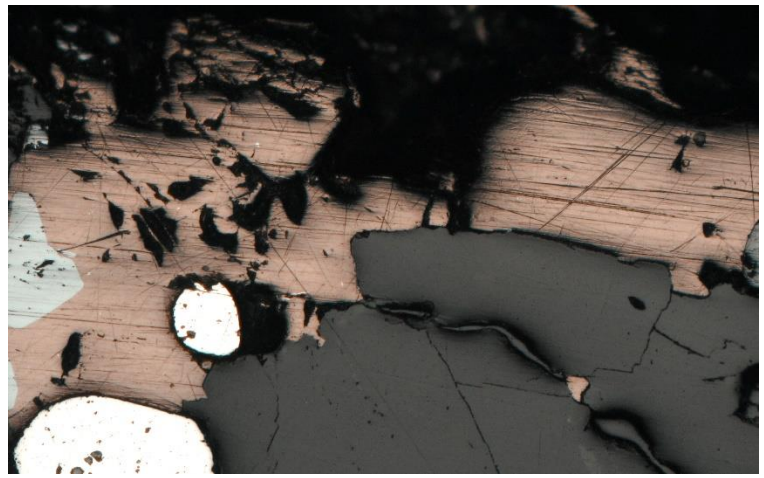
Примеры работы алгоритма адаптации



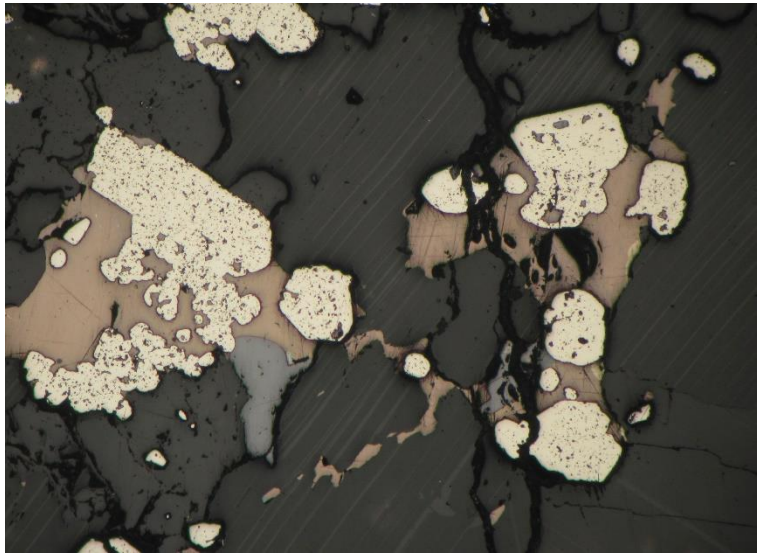
Референсные изображения

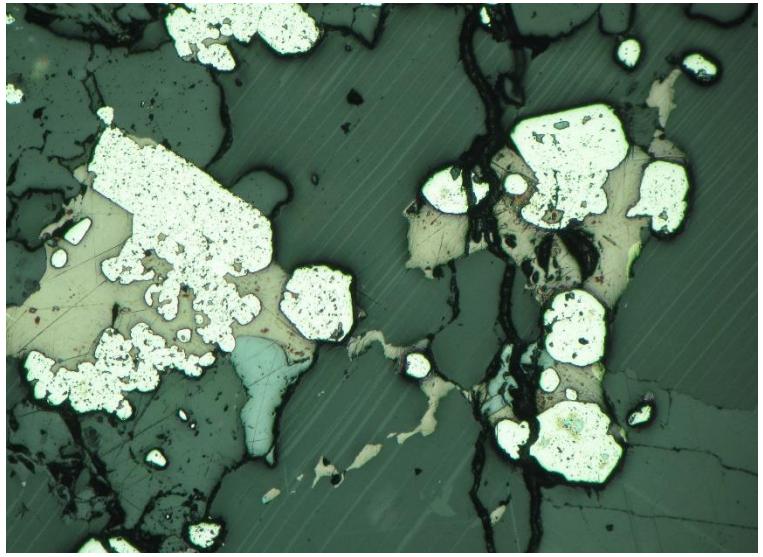


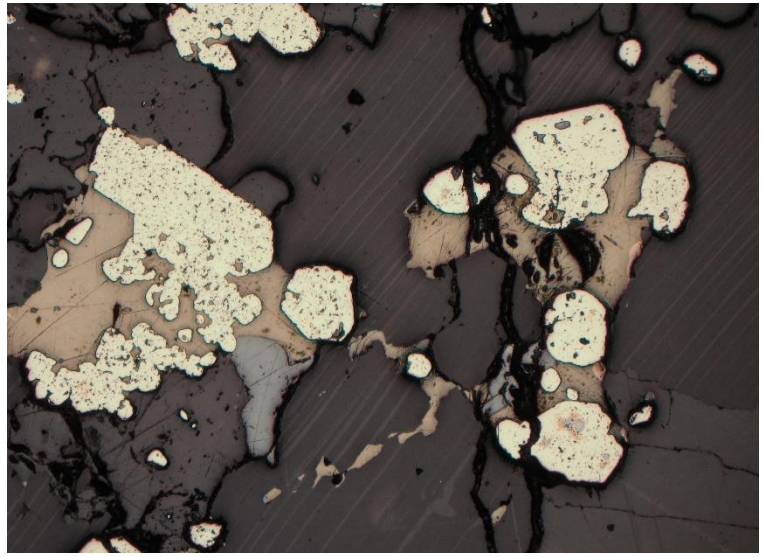
Искаженные изображения



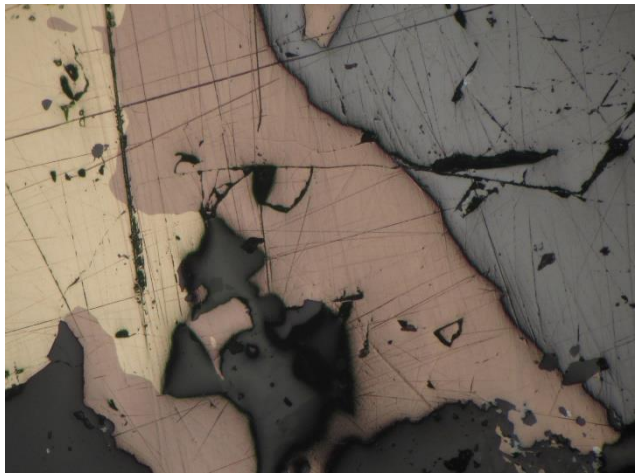
Адаптированные изображения



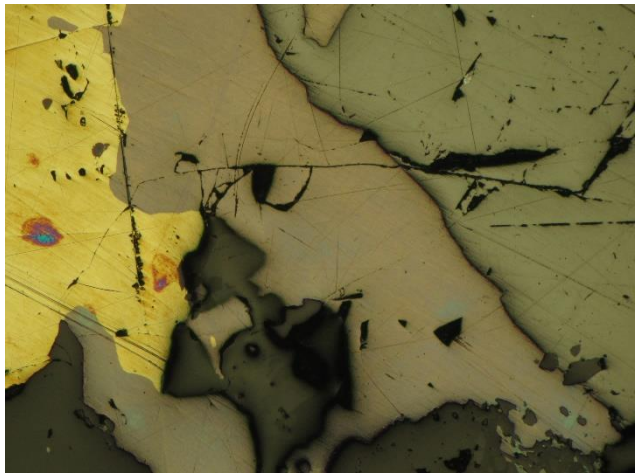




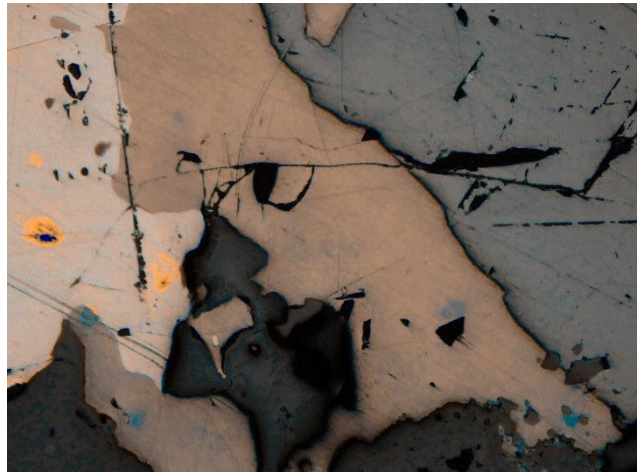
Примеры работы алгоритма адаптации



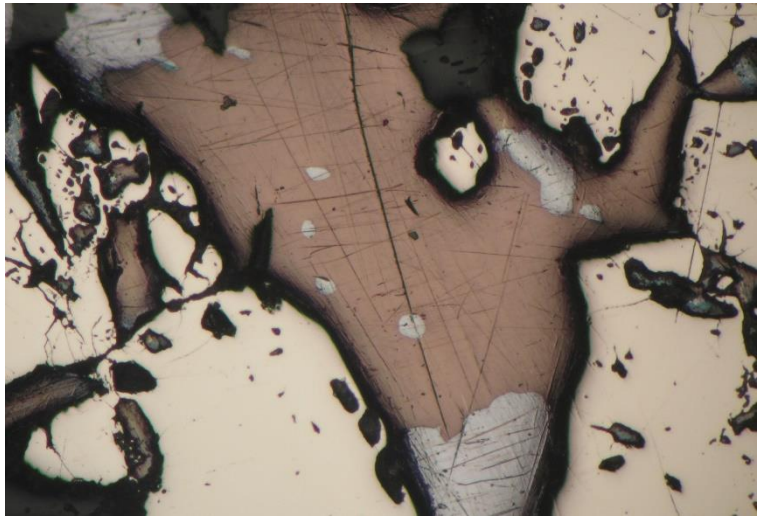
Референсные изображения

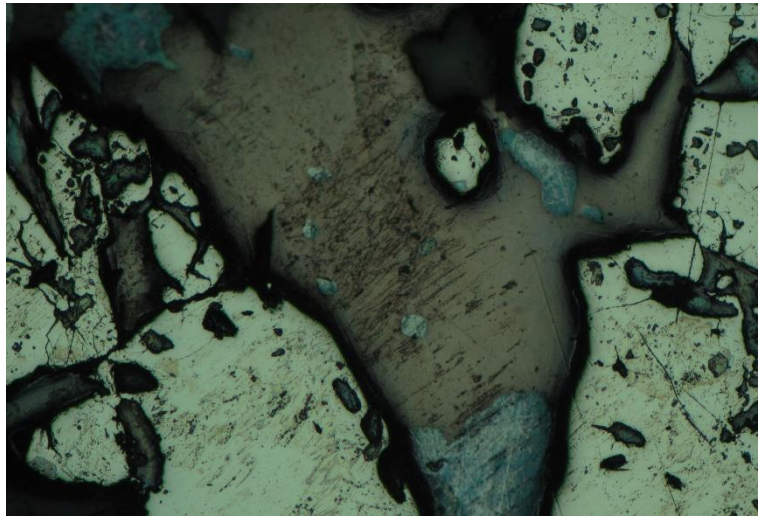


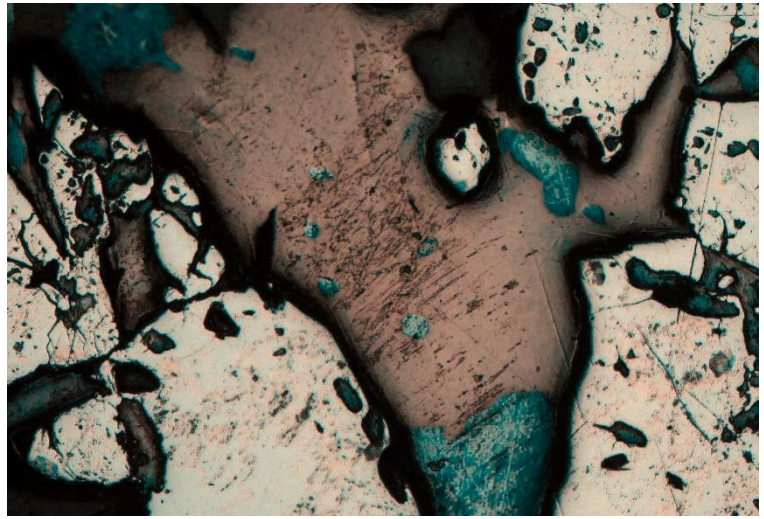
Искаженные изображения



Адаптированные изображения

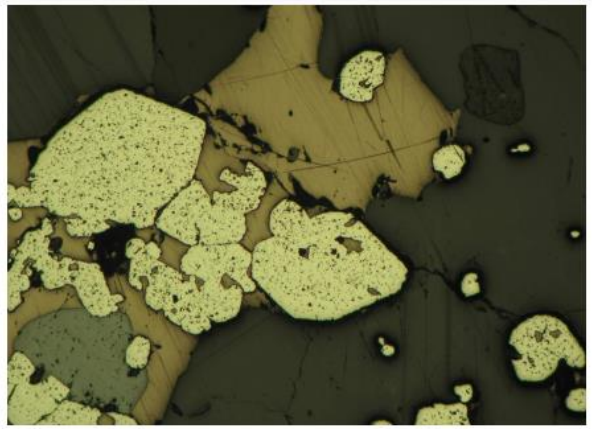




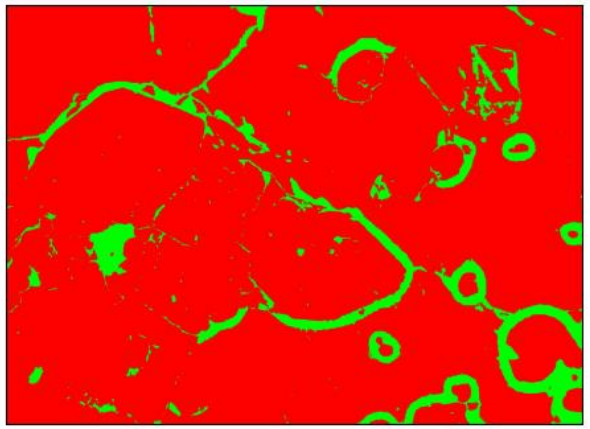


Оценка работы алгоритма

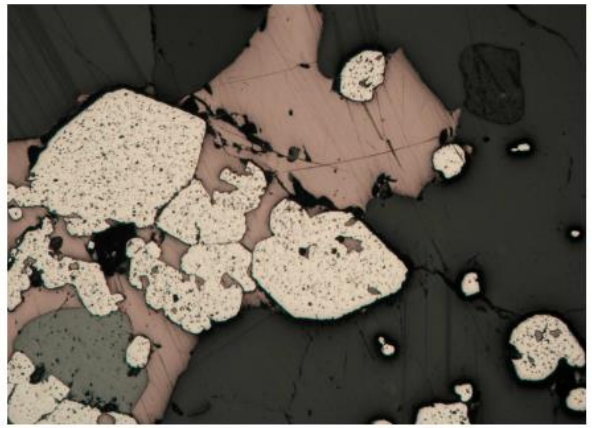
Используется следующая **метрика**: чем ближе точность автоматической сегментации минералов на адаптированных изображениях к значениям на референсном наборе данных, тем лучше работает алгоритм адаптации.



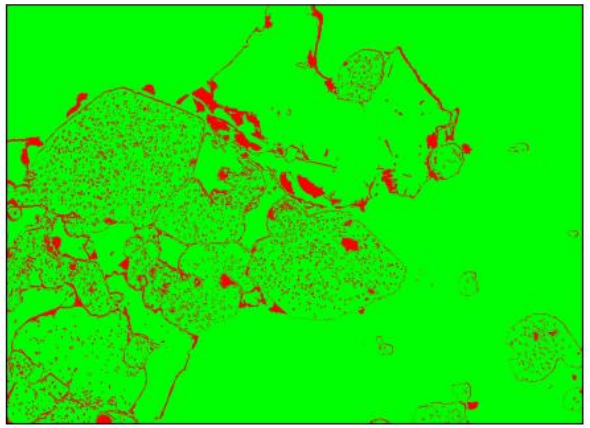
Искаженное изображение



Карта ошибок сегментации

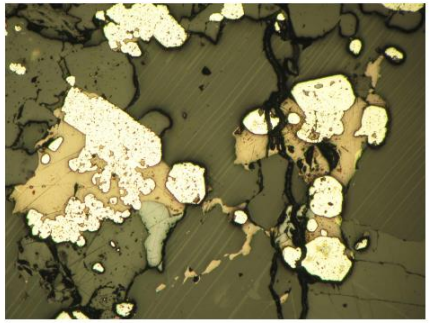


Адаптированное изображение

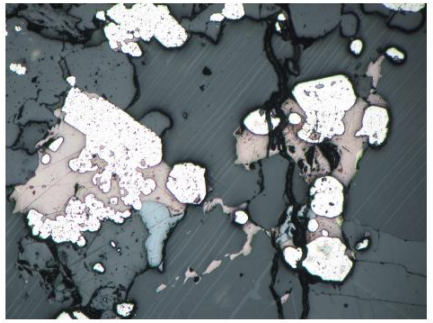


Карта ошибок сегментации

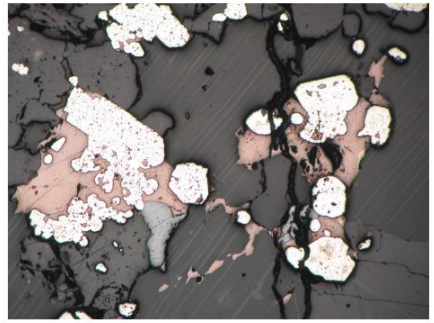
Оценка работы алгоритма



«Желтое» искажение



«Синее» искажение



«Светлое» искажение

Тип	Точность	Intersection over Union (IOU)						
		BG	Ccp	Gl	Brt	Py/Mrc	Sph	Tnt/Ttr
Референсное	0.9504	0.9351	0.8585	0.8560	0.9140	0.9270	0.6988	0.5565
Искаженное	0.2482	0.3770	0.0408	0.0510	0.0228	0.1886	0.0027	0.0340
Адаптированное	0.6029	0.5667	0.2262	0.1831	0.5173	0.5207	0.1019	0.1503
Желтое искаж.	0.2053	0.2294	0.0398	0.0000	0.0000	0.3768	0.0000	0.0000
Желтое адапт.	0.5262	0.6042	0.0691	0.1341	0.4472	0.2250	0.2089	0.0274
Синее искаж.	0.2911	0.5136	0.0661	0.0572	0.0456	0.0210	0.0031	0.0372
Синее адапт.	0.4858	0.4601	0.2182	0.1293	0.2436	0.6065	0.0104	0.3904
Светлое искаж.	0.4552	0.5004	0.1120	0.1094	0.1517	0.5031	0.0251	0.0008
Светлое адапт.	0.7495	0.6337	0.7334	0.4252	0.6877	0.6800	0.1950	0.0117

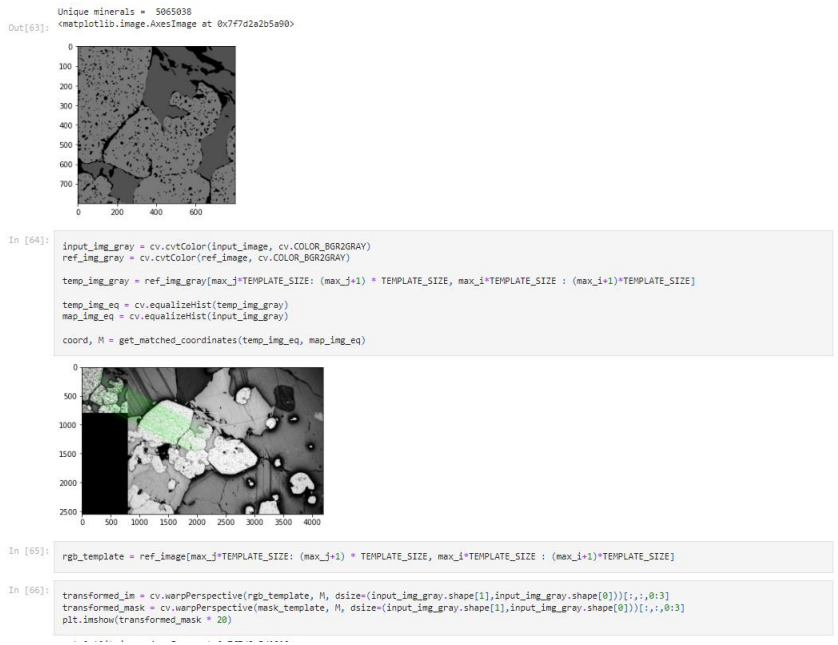
Программная реализация

Был разработан алгоритм адаптации изображений аншлифов, полученных в разных условиях съемки. Предложенный метод был программно реализован на языке **Python3** с использованием библиотек **scikit-image, OpenCV и NumPy**. Программная реализация представлена в интерактивной среде Jupyter Notebook.

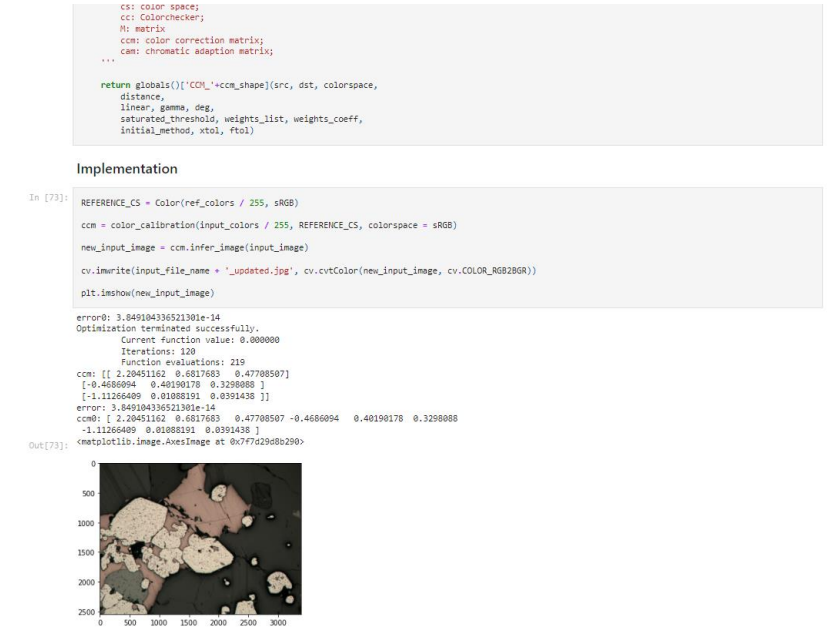
```
In [50]:
ref_image = cv.imread('ref.jpg')
ref_image = cv.cvtColor(ref_image, cv.COLOR_BGR2RGB)

input_file_name = 'input'
input_image = cv.imread(f'{input_file_name}.jpg')
input_image = cv.cvtColor(input_image, cv.COLOR_BGR2RGB)

ref_mask = cv.imread('mask.png')

In [51]:
f, axarr = plt.subplots(1,3, figsize=(15,15))
axarr[0].imshow(ref_image)
axarr[1].imshow(input_image)
axarr[2].imshow(ref_mask * 20, 'gray')

Out[51]:
<matplotlib.image.AxesImage at 0x7f7d49301510>
```

Template matching

```
In [17]:
MIN_MATCH_COUNT = 2

def get_matched_coordinates(temp_img, map_img):
    """
    Gets template and map image and returns matched coordinates in map image

    Parameters
    ----------
    temp_img: image
        image to be used as template

    map_img: image
        image to be searched in

    Returns
    ---------
    ndarray
        an array that contains matched coordinates

    """

    # initiate SIFT detector
    sift = cv.xfeatures2d.SIFT_create(nfeatures = 0,
                                      nOctaveLayers = 3,
```



```
Unique minerals =  5065038
Out[63]:
<matplotlib.image.AxesImage at 0x7f7d2a2b5a90>

In [64]:
input_img_gray = cv.cvtColor(input_image, cv.COLOR_BGR2GRAY)
ref_img_gray = cv.cvtColor(ref_image, cv.COLOR_BGR2GRAY)

temp_img_gray = ref_img_gray[max_j*TEMPLATE_SIZE: (max_j+1) * TEMPLATE_SIZE, max_i*TEMPLATE_SIZE : (max_i+1)*TEMPLATE_SIZE]

temp_img_eq = cv.equalizeHist(temp_img_gray)
map_img_eq = cv.equalizeHist(input_img_gray)

coord, M = get_matched_coordinates(temp_img_eq, map_img_eq)

In [65]:
rgb_template = ref_image[max_j*TEMPLATE_SIZE: (max_j+1) * TEMPLATE_SIZE, max_i*TEMPLATE_SIZE : (max_i+1)*TEMPLATE_SIZE]

In [66]:
transformed_im = cv.warpPerspective(rgb_template, M, dsize=(input_img_gray.shape[1],input_img_gray.shape[0]))[:,:,0:3]
transformed_mask = cv.warpPerspective(mask_template, M, dsize=(input_img_gray.shape[1],input_img_gray.shape[0]))[:,:,0:3]
plt.imshow(transformed_mask * 20)
```



```
        cs: color space;
        cc: Colorchecker;
        M: matrix
        ccm: color correction matrix;
        cam: chromatic adaption matrix;
    '''

    return globals()['CCM_'+ccm_shape](src, dst, colorspace,
        distance,
        linear, gamma, deg,
        saturated_threshold, weights_list, weights_coeff,
        initial_method, xtol, ftol)
```

Implementation

```
In [73]:
REFERENCE_CS = Color(ref_colors / 255, sRGB)

ccm = color_calibration(input_colors / 255, REFERENCE_CS, colorspace = sRGB)

new_input_image = ccm.infer_image(input_image)

cv.imwrite(input_file_name + '_updated.jpg', cv.cvtColor(new_input_image, cv.COLOR_RGB2BGR))

plt.imshow(new_input_image)

error0: 3.849104336521301e-14
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 120
         Function evaluations: 219
ccm: [[ 2.20451162  0.6817683   0.47708507]
 [-0.4686094   0.40190178  0.3298088 ]
 [-1.11266409  0.01088191  0.0391438 ]]
error: 3.849104336521301e-14
ccm0: [ 2.20451162  0.6817683   0.47708507 -0.4686094   0.40190178  0.3298088
 -1.11266409  0.01088191  0.0391438 ]
Out[73]:
<matplotlib.image.AxesImage at 0x7f7d29d8b290>
```

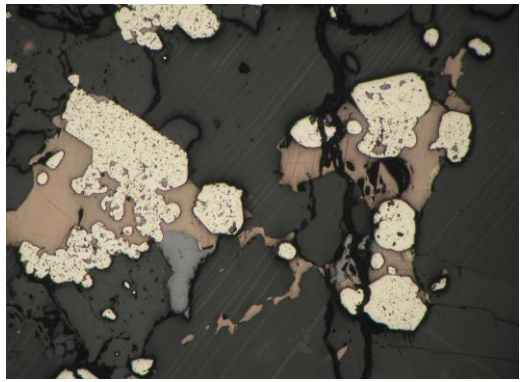
https://github.com/luseno4ek/geology_image_adaptation

Результаты

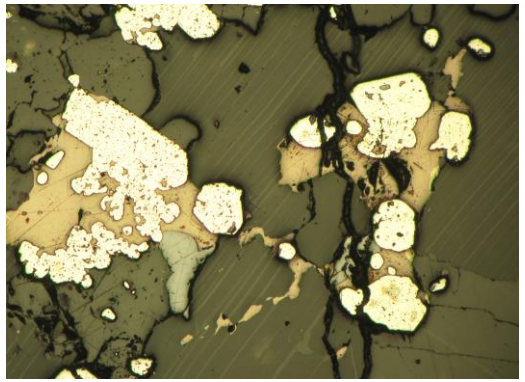
1. Был разработан и реализован на языке **Python3** алгоритм адаптации изображений аншлифов, полученных в разных условиях съемки.

2. Алгоритм был протестирован на наборе данных **LumenStone** и показал увеличение качества автоматической сегментации более чем в два раза по сравнению с точностью на искаженных изображениях.

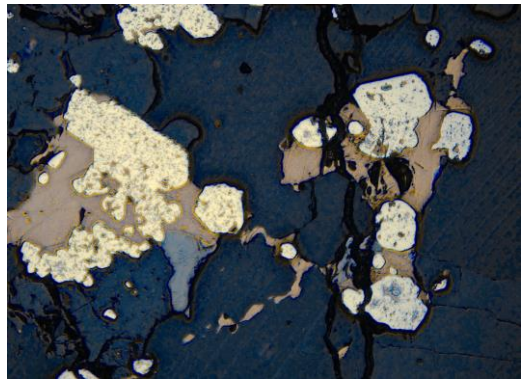
3. Выявлена следующая проблемы алгоритма — недостаточность извлекаемой информации для цветовой коррекции:



Референсное изображение



Искаженное изображение



Адаптированное изображение

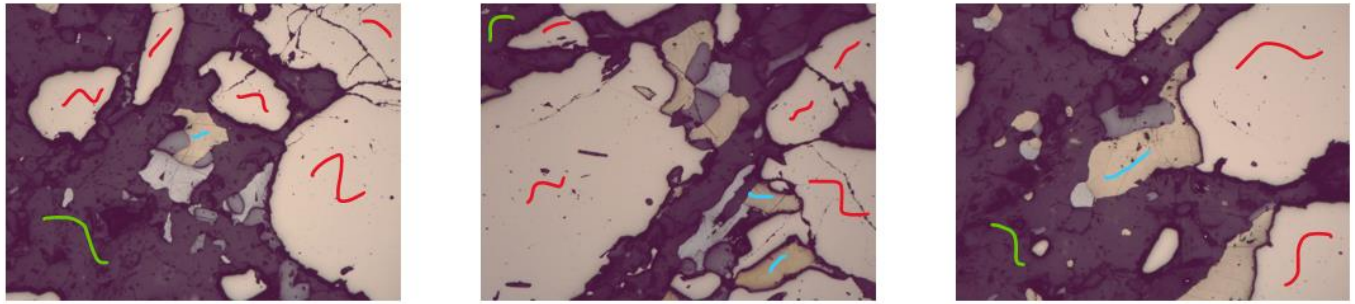
→ Доработка алгоритма заключается в увеличении размера совмещаемого
→ фрагмента (сейчас 800x800 пикселей).

Дальнейшее развитие

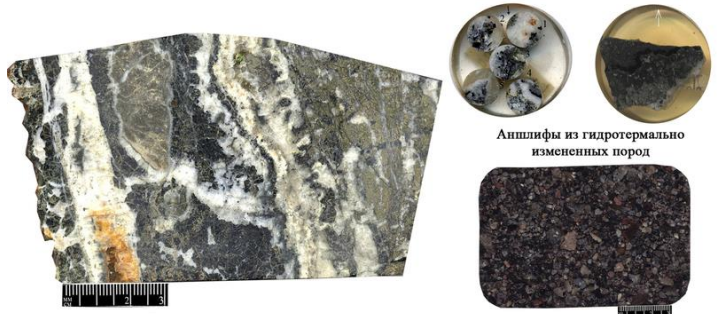
Реализация **интерактивного приложения**, которое позволило бы заменить полную маску сегментации на некоторую обратную связь от пользователя.

Варианты получения информации от пользователя:

1. Пользователь предоставляет частичную разметку характерных минералов с помощью штрихов или точек входного изображения, сделанного в новых условиях;



2. Пользователь предварительно калибрует алгоритм к своим условиям съемки, предоставляя снимок специально подготовленного аншлифа, для которого имеется полная маска сегментации.



Спасибо за внимание!