Data Processing for Automatic Classification of Spheroidite Microstructure using Deep Learning Based on FCNNs

Rania Elbana¹, Rania Mostafa², Ahmed Elkeran²
¹Basic science Department, Mansoura Higher Institute for Engineering & Technology, Mansoura, Egypt
²Production Engineering and Mechanical Design Department, Faculty of Engineering, MU, Egypt

* Corresponding Author (Email: ranmos75@mans.edu.eg)

Abstract-- The microstructure of a material is one of the main influences on its mechanical properties, and therefore it participates in determining its uses and possible applications in the field of manufacturing. The automatic recognition which used for microstructure classification is a major challenge. Nowadays, Deep learning is the most exciting method that used to classify the microstructure of matter automatically. In this work, six outstanding Fully Convolutional Neural Networks (FCNNs) architectures were used to study their capabilities in classifying Spheroidite microstructural images into classes (6classes and 3classes). Three tasks were accomplished to test the classification of different classes of Spheroidite micrographs that were divided into based on different annealing conditions. The constructed Datasets were comprised of the images that are taken over a range of magnifications. The six networks were compared to assess their performance during the supposed tests. The comparison includes the all possible combinations of training datasets size, the learning rate, the cropping method of images, number of classes and the magnification scale. Results showed that the transfer learning approach can represent the microstructure image data very well by using the pretrained models with validation accuracies about 90.83%, 98.33% for 6classes and 3classes, respectively. When fine-tuning approach was used, the validation accuracy reached about 96.67%, 100% for 6classes and 3classes, respectively.

Index Term-- Deep learning; Spheroidite morphology; FCNNs; Automatic Classification; Material Microstructure.

1. INTRODUCTION

The microstructure of materials affected by the mechanical and thermal processes applied to it, which defines their mechanical properties. Therefore, the microstructural analysis is very important to the design and manufacture processes [1]. A lot of research that are depending on information technology and data science methods has been concerned lately with the field of material informatics. The microstructural images were taken a great attention in the field of material science [2-8]. A big effort has been directed by the researchers towards the deep understanding of materials microstructure. Corresponding to Material Science, the finding of reliable characterization and classification of the structures, that control the microstructural image is enabling the studying of microstructures [9]. Earlier, Microstructural images were evaluated by a traditional method based on human experts to understand the micrographs and to make a connection between these images and the processing conditions and their features outcomes [10]. After that, New researches have begun to use the applications of modern computer vision approach to construct the microstructure representations that appropriate for use in the machine learning and the systematic computational analysis of microstructure tasks [11-12].

Deep Learning methods have been recently gripped great attention from scientists It can extract and classify the features in only one step unlike traditional methods. Deep Learning methods are based on artificial neural networks such as Convolutional Neural Networks (CNNs) [13]. Nowadays, many researches have been applied these methods successfully with respect to computer vision problems [14-15].

The challenge of automated recognition of metals microstructure have been presented by number of researches using Computer vision and machine learning classical methods. Francesco Iacoviello. [16] made a classification of ductile cast iron specimens based on the image processing evaluation of global features of the specimens and a support vector machine classifier to classify the specimens with regular nodules from the category of specimens with irregular nodules. Pattan Prakash.[17] employed a classification of graphite grains of cast iron using spectral and spatial features and a neural network classifier. Jason S. Van Sluytman. [18] was analyses the precipitate shapes in nickel-based superalloys. Additionally, Brian L. DeCost and Elizabeth A. Holm.[19] used the Visual Bag of Words feature extraction method with a Support Vector Machine classifier for multi-class classification of microstructures. Aritra Chowdhury and Elizabeth Kautz. [20] used various feature extraction, feature selection, and classification methods to classify tasks between micrographs of dendrites of alloys varying Sn–Ag–Cu compositions. Other researchers applied bilinear CNN representations to artificial lamellar structures [21-23]. Brian L. DeCost. [24] used the UHCS dataset to compare state-of-the art CNN-based image texture representations with the classic bag of visual words (BoW) representation. Seyed Majid Azimi. [25] employs pixel-wise segmentation via Fully Convolutional Neural Networks (FCNNs)
accompanied by a max-voting scheme to classify steel image dataset.

In this work, Deep learning technique based on Fully Convolutional neural network (FCNNs) pretrained models are applied on specific microstructural constituents of ultra-high carbon steel (UHCS). These pretrained models are used for the task of micrograph recognition for different spheroidite datasets. Spheroidite microstructure images are very similar and its classification is very confusing for human experts, so its automated classification is a great challenge. The evaluation of these several competitive computer vision techniques is made. The comparison between these networks includes the effect of the number of classes that are classified, the learning rate value changing, the number of trained images, and the effect of the image magnification scale. Their points of strength and weakness for a range of real-world microstructure analytics tasks are discussed. Also, the effect of the random cropping trend is studied. At the end of this work there is a recommendation for the most proper classification model for the task at hand.

2. Material and Methods

2.1. UHCS Dataset

The open source (UHCS) dataset [26] is used as the case study in this paper. All morphologies involved by this dataset can be summarized in Fig.1. While Table I illustrates the chemical composition of UHCS, the annealing temperatures, annealing duration and quenching methods are illustrated in Tables 2, 3, and 4 respectively. In UHCS dataset which has 961 micrographs, Spheroidite images (374 micrographs) which are resulting from twenty-three distinct annealing schedules, are very similar even that their classification based on human expert is very confusing. So, the classification task in this work is focused on spheroidite morphology. The automated recognition is a great challenge especially if these images have the same specifications in terms of annealing condition and magnification scale. To obtain a balanced classification dataset, seven annealing conditions classes were selected with thirteen micrographs with the same magnification scale for each class within the spheroidite subset. The Annealing Conditions (AC) are illustrated in Table 5.

In order to make a real evaluation for each model, different datasets with different number of classes and magnification scales were constructed.Datasets are classified to three Classes (3C) with 10 and 20 microns (μm) as a magnification scale and six Classes (6C) with 10 microns. To construct the required datasets each original image is cropped to (224*224) pixel sub-images with two different methods. In the first cropping method, the original image will be cropped from its center to four sub-images for the 6C and 3C classification test. Also, the original image will be cropped overlapped to eight sub-images for both 6C and 3C classification tests. In the second method, the original image will be cropped randomly to 8, 16, 32 and 64 sub-images for both 6C and 3C to get a huge dataset. Table 6 contains the numbers of images in each dataset for both classes of classification test under the two cropping methods. The size of datasets is varied from 120-3840 training image and 30-960 validation image.

Table I

<table>
<thead>
<tr>
<th>C</th>
<th>Si</th>
<th>Mn</th>
<th>Cr</th>
<th>Ni</th>
<th>Mo</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.02</td>
<td>0.65</td>
<td>0.72</td>
<td>3.86</td>
<td>1.45</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>Annealing temperature</th>
<th>700</th>
<th>750</th>
<th>800</th>
<th>900</th>
<th>970</th>
<th>1000</th>
<th>1100</th>
<th>(Mostly as-cast) N/A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>11</td>
<td>4</td>
<td>149</td>
<td>60</td>
<td>344</td>
<td>14</td>
<td>16</td>
<td>363</td>
<td>961</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>Annealing duration</th>
<th>5M</th>
<th>1H</th>
<th>90M</th>
<th>3H</th>
<th>8H</th>
<th>24H</th>
<th>48H</th>
<th>85H</th>
<th>(Mostly as-cast) N/A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>89</td>
<td>16</td>
<td>173</td>
<td>100</td>
<td>69</td>
<td>115</td>
<td>5</td>
<td>31</td>
<td>363</td>
<td>961</td>
</tr>
</tbody>
</table>
Table IV
Quenching methods list [26]

<table>
<thead>
<tr>
<th>Quench method</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Air cooled</td>
<td>20</td>
</tr>
<tr>
<td>FC</td>
<td>Furnace cooled</td>
<td>73</td>
</tr>
<tr>
<td>Q</td>
<td>Quench</td>
<td>489</td>
</tr>
<tr>
<td>650 1H</td>
<td>650 °C for 1 h</td>
<td>16</td>
</tr>
<tr>
<td>N/A</td>
<td>Mostly as-cast</td>
<td>363</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>961</td>
</tr>
</tbody>
</table>

Table V
The selected Annealing Conditions (AC) description with Quench method (Q) [26]

<table>
<thead>
<tr>
<th>(AC)</th>
<th>AC1</th>
<th>AC2</th>
<th>AC3</th>
<th>AC4</th>
<th>AC5</th>
<th>AC6</th>
<th>AC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann. Temp.</td>
<td>800°C</td>
<td>970°C</td>
<td>970°C</td>
<td>970°C</td>
<td>800°C</td>
<td>970°C</td>
<td>800°C</td>
</tr>
<tr>
<td>Ann. duration</td>
<td>85H</td>
<td>8H</td>
<td>24H</td>
<td>90M</td>
<td>8H</td>
<td>3H</td>
<td>24H</td>
</tr>
</tbody>
</table>

Fig. 1. Primary microstructure constituents in the UHCS dataset: (a) Pearlite, (b) Spheroidite, (c) Cementite network, (d) Pearlite containing Spheroidite, (e) Widmanstatten cementite, and (f) Martensite.

Table VI
The details of Spheroidite datasets.

<table>
<thead>
<tr>
<th>No. of classes</th>
<th>Datasets name</th>
<th>No. of train images</th>
<th>No. of validation images</th>
<th>Classes name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6classes (6C)</td>
<td>Dataset240</td>
<td>240</td>
<td>60</td>
<td>AC1&amp;AC3</td>
</tr>
<tr>
<td></td>
<td>Dataset480</td>
<td>480</td>
<td>120</td>
<td>&amp;AC4&amp;AC5</td>
</tr>
<tr>
<td></td>
<td>Dataset480r</td>
<td>480</td>
<td>120</td>
<td>&amp;AC6&amp;AC7</td>
</tr>
<tr>
<td></td>
<td>Dataset960r</td>
<td>960</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset1920r</td>
<td>1920</td>
<td>480</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset3840r</td>
<td>3840</td>
<td>960</td>
<td></td>
</tr>
<tr>
<td>3classes (3C)</td>
<td>Dataset120</td>
<td>120</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset240</td>
<td>240</td>
<td>60</td>
<td>The classes name for 10µ images are (AC1&amp;AC3 &amp;AC4) and for 20µ (AC2&amp;AC3 &amp;AC6)</td>
</tr>
<tr>
<td></td>
<td>Dataset240r</td>
<td>240</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset480r</td>
<td>480</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset960r</td>
<td>960</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dataset1920r</td>
<td>1920</td>
<td>480</td>
<td></td>
</tr>
</tbody>
</table>

(r) subscripts to datasets produced from random crop method
2.2. FCNNs Pretrained models

FCNNs have newly appreciated a great success in large-scale image and thus, due to the large public image repositories. ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has played an important role in the field of deep visual recognition, it acts as a testbed for some generations of large-scale image classification systems, from high-dimensional shallow feature encodings to deep Convolutional networks ConvNets [27]. Maybe it’s possible to model any function using just a single hidden layer theoretically, but that required an enormous number of neurons and thus making the network difficult to train, so a deep network with many hidden layers which try to learn different features at different layers are used to overcome these challenges. These deep networks contain a huge number of unknown parameters (in millions) and to find the optimum parameters the network is training using the training data and thus required a lot of data (in millions) to obtain the accurate values for these parameters not approximate values. The problem is that it is very difficult to get such huge labeled datasets for training the network especially with respect to microstructure morphology and even this data could be obtained it takes a large amount of time, money and effort to train such network successfully.

Luckily, many research groups share the models they have trained for competitions like ILSVRC to the open-source community. These models are trained on millions of pictures and for many hours on powerful GPUs. Most often these models can be used as a starting point for a new function training process, instead of training the new model from scratch.

In this work Python3.6 and the pretrained ConvNets models that are comes bundled with Keras deep learning library with Tensorflow backend are used to apply on the spheroidite microstructure images classification. This is called Transfer Learning [28]. Transfer Learning can done by one of the two scenarios: The first one is using the Pretrained models as an initialization or a fixed feature extractor for the task of interest, the second scenario is Fine-tuning the ConvNet by replace and retrain the classifier on top of the ConvNet on the new dataset, and also fine-tune the weights of the pretrained network by continuing the backpropagation. The selected models fully integrated into the Keras core will be VGG16 and VGG19 [27], ResNet50 [29], MobileNet [30], InceptionV3 [31], Xception [32]. All the mentioned models are used as pretrained models and the fine-tuning approach is applied only to VGG16 model. The task of fine-tuning a network is to tweak the parameters of the previously trained network in order that it adapts to the new task at hand. Thus, fine-tuning avoids both the limitations associated with training the network from scratch.

The amount of data required for training is not much because of we are not training the entire network and the part that is being trained is not trained from scratch. Since the parameters that require to be updated is smaller amount, the amount of time needed will also be less.

The following steps are applied for Fine-tuning VGG16 model:

- **First**, load a VGG16 model without the top layer (which consists of fully connected layers).
- **Second**, freeze the required layers.
- **Third**, create a new model by add a classifier on top of the convolutional base. A fully connected layer followed by a softmax layer with 6 outputs were added.
- **Fourth**, Setup the data generators by using ImageDataGenerator available in Keras to read images in batches directly from the “train” and “validation” folders and perform data augmentation.
- **Finally**, Train the model and check out the performance. The optimizer and the learning rate are specified then the training is start.

3. Results and Discussion

This section reports the results from applying multiple classification models to the task of micrograph recognition for different spheroidite datasets. Three tests were applied to study the attitude of each network with respect to number of classes and magnification scale. In each test different dataset size and different Learning Rate (L.R) values were used. Dataset size varied according to the way of cropping the sub-images from the original one, as it mentioned previously. The pretrained model’s classification and validation accuracies mainly affected by the learning rate (L.R) value. The large (L.R) value may decrease the Validation Accuracy (V.A). The value of 0.01 as a (L.R) gave about 16.6 % and 33.33% V.A for 6C and 3C classification test, respectively. So, the three smaller values of L.R (10^{-3}, 10^{-4} and 10^{-5}) are selected to apply for each test.

3.1. Test1: Classification of 6C / Spheroidite micrograph / 10µm

Figure 4 shows the effect of the change in the value of L.R on the validation accuracies for all selected datasets in this case (6C/10µm). By noticing the graphs in Figs. 4a, b and c, it found that reducing the value of L.R (10^{-3}, 10^{-4} and 10^{-5}) has a clear effect on raising the validation accuracy percentage of most of selected datasets with all six classification networks models. Upon careful observation of the results, it is noted that the V.A of ResNet50 model does not exceed 62.5% with smallest L.R value (10^{-5}) and this is the least value achieved in this test. It was also noted clearly that the VGG16 reached about 87.5 % with L.R value (10^{-5}) for dataset480. The V.A of VGG19 model has reached about 90.83% for dataset480 at L.R value (10^{-4}). It is noticeable that the MobileNet model shows a good performance, especially with the low learning rate (10^{-5}), as the accuracy reaches 90.5 % for dataset3840r. Also, Inception -v3 and xception models reached about 87% V.A with L.R values (10^{-4} and 10^{-5})
respectively, this V.A value was achieved for training data (dataset3840r).

Figure 5 illustrates the effect of adding the batch normalization (B.N) [33] and Fine tuning to VGG16 model under different L.R values ($10^{-3}$, $10^{-4}$ and $10^{-5}$). B. N approach has added to VGG16 model instead of the Dropout layer to study how it can affect the classification process. From Fig. 5 it can be observed that the B.N approach doesn’t improve the V.A of VGG16 model except in three cases as the following:

1. With dataset240 and L.R ($10^{-4}$), the V.A changed from 71.67% to 75%.
2. With dataset240 and L.R ($10^{-5}$), the V.A changed from 78.33% to 80%.
3. With dataset480 and L.R ($10^{-5}$), the V.A changed from 87.5% to 88.33%.

Observing the three graphs in Fig. 5 to know the extent of improvement that can happen by VGG16 Fine-tuning, it was found that the low L.R value ($10^{-5}$) does not have any effect on the model, while if the value is reduced to the least ($10^{-4}$ and $10^{-5}$), the effect is clearly evident in improving the validation accuracy, which reaches its peak (about 96.6%) with the $10^{-5}$ as a L.R value for dataset 480.

The best V.A obtained by all the pretrained models with respect to different datasets and learning rate values are summarized in table.7. By reading the summary of results from the table, it becomes clear how the impact of the decreasing in the learning rate value. It must be observed that each network achieved its best V.A based on the L.R value and dataset size together. VGG19, ResNet50, VGG16, and VGG16 variation (fine-tuning and batch normalization) have achieved the highest V.A for dataset480. While, the other three networks are very compatible with large dataset size (dataset 3840r).

5.2. Test2: Classification of 3C / Spheroidite micrograph / 10\(\mu m\)

Test2 studied the classification task of 3C of spheroidite images with 10\(\mu m\) as a magnification scale, the two scenarios that mentioned before will be applied. In this test the applied networks have achieved V.A much greater than the results that achieved in the first test with respect to nearly all datasets and all L.R values. Again let’s study the effect of decreasing the L.R value, increasing the trained dataset size and the way of preparing the dataset. At the end of this test result’s analysis, the optimal classification parameters can be chosen for 3C classification task. By observing the Figs.6 and 7, it becomes clear that the decreasing of L.R value from $10^{-3}$ to $10^{-4}$ and $10^{-5}$ has a different effect for each network with respect to the dataset size.
The least V.A obtained in this test was 85% by ResNet50 model, while VGG16, Mobilenet, Inception-v3, and Xception pretrained models achieved higher accuracy—96.67%. The maximum validation accuracies achieved in this test for each dataset and different learning rate values are concluded in table 8, while the maximum values are bolded and underlined. In this case, the VGG19, ResNet50, VGG16 and Mobilenet models achieved the best V.A values with dataset240, so there is no need to the random cropping. For Inception-v3, Xception, and B.N to VGG16 models, a random cropping is necessary. These models achieved their highest V.A with dataset240r, dataset480r, and dataset1920r respectively. Most of networks used in this test reached its best V.A values by using L.R (10^-4) excluding Inception-v3 and Mobilenet networks they preferred L.R (10^-5). Fine-
tuning achieved the maximum V.A in this test (100%) when using dataset240 and L.R (10^{-4}).

<table>
<thead>
<tr>
<th>Test</th>
<th>Network</th>
<th>Dataset240</th>
<th>Dataset480</th>
<th>Dataset480r</th>
<th>Dataset960r</th>
<th>Dataset1920r</th>
<th>Dataset3840r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test1</td>
<td>VGG16</td>
<td>78.33</td>
<td>87.5</td>
<td>80</td>
<td>78.75</td>
<td>83.96</td>
<td>83.54</td>
</tr>
<tr>
<td>Test1</td>
<td>VGG19</td>
<td>83.33</td>
<td>90.83</td>
<td>74.17</td>
<td>79.17</td>
<td>80.62</td>
<td>81.46</td>
</tr>
<tr>
<td>Test1</td>
<td>ResNet50</td>
<td>45</td>
<td>62.5</td>
<td>38.33</td>
<td>45</td>
<td>53.13</td>
<td>56.04</td>
</tr>
<tr>
<td>Test1</td>
<td>Mobilenet</td>
<td>80</td>
<td>86.67</td>
<td>83.33</td>
<td>86.67</td>
<td>89.79</td>
<td>90.52</td>
</tr>
<tr>
<td>Test1</td>
<td>Inception-v3</td>
<td>78.33</td>
<td>80</td>
<td>81.67</td>
<td>76.25</td>
<td>84.79</td>
<td>87.19</td>
</tr>
<tr>
<td>Test1</td>
<td>Xception</td>
<td>81.67</td>
<td>79.17</td>
<td>79.17</td>
<td>81.67</td>
<td>84.79</td>
<td>87.92</td>
</tr>
<tr>
<td>Test1</td>
<td>VGG16 Finetuning</td>
<td>86.67</td>
<td>96.67</td>
<td>91.67</td>
<td>87.08</td>
<td>93.12</td>
<td>94.17</td>
</tr>
<tr>
<td>Test1</td>
<td>B.N to VGG16</td>
<td>80</td>
<td>88.33</td>
<td>74.17</td>
<td>78.75</td>
<td>80</td>
<td>81.77</td>
</tr>
</tbody>
</table>

5.3. Test3: Classification of 3C / Spheroidite micrograph / 20 μm

Test3 was done for images with high magnification scale (20μm) where the grains boundaries are not sufficiently clear. This test explained how the variation of the dataset size and learning rate value affect the accuracy of the classification mission. Fig.8 and Fig.9 show the different V.A results that achieved by each network for each dataset and L.R value. The maximum values that obtained for each model with respect to each dataset are summarized in table9. Each model achieves its maximum V.A at specific condition as the following:

- VGG16 model achieves 98.33% maximum V.A with dataset240r and L.R (10^{-4}).
- VGG19 model achieves 94.58% maximum V.A with dataset960r and L.R (10^{-5}).
- ResNet50 model achieves 80.42% maximum V.A with dataset960r and L.R (10^{-5}).
- Mobilenet model achieves 94.58% maximum V.A with dataset1920r and L.R (10^{-5}).
- Inception-v3 model achieves 89.17% maximum V.A with dataset1920r and L.R (10^{-5}).
- Xception model achieves 90% maximum V.A with dataset960r and L.R (10^{-5}).
- VGG16 Fine-tuning achieves 100% maximum V.A with dataset120 & dataset240r and L.R (10^{-4}, 10^{-5}) respectively.
- VGG16 with B.N achieves 96.67% maximum V.A with dataset120 and L.R (10^{-5}).

Based on the previous results, it can be concluded that the pretrained models that used to classify such difficult microstructure images (with high magnification scale) need relatively large dataset size, except for VGG16 and its modifications.
5.4. Networks Evaluation

5.4.1. VGG16 model

By observing the results, it found that the VGG16 model is the most suitable model for classification tasks of 3C. In such model at the third test, the higher V.A was recorded as 98.33% and the error rate was 1.66% with dataset240r and L.R ($10^{-3}$). According to these results, the Confusion Matrix (CM) of (VGG16 model) is constructed in table 10. The constructed matrix shows the number of samples of each class that are predicted by VGG16 model. It can be noticed from table 10, that the network produces only one misclassification as a part of one of the original images belongs to AC2 class. The misprediction obtained here represent just 12.5% of the original image pixels while that original image is correctly predicted by 87.5%. Therefore,
original image can be considered as successful predicted image and the V.A of VGG16 model can be taken as 100%.

### Table VIII

<table>
<thead>
<tr>
<th>Dataset120</th>
<th>Dataset240</th>
<th>Dataset240r</th>
<th>Dataset480r</th>
<th>Dataset960r</th>
<th>Dataset1920r</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VGG16</strong></td>
<td>80</td>
<td><strong>96.67</strong></td>
<td>91.67</td>
<td>87.5</td>
<td>91.25</td>
</tr>
<tr>
<td><strong>VGG19</strong></td>
<td>80</td>
<td>95</td>
<td>88.33</td>
<td>89.17</td>
<td>86.67</td>
</tr>
<tr>
<td><strong>ResNet50</strong></td>
<td>70</td>
<td><strong>85</strong></td>
<td>58.33</td>
<td>60</td>
<td>65.42</td>
</tr>
<tr>
<td><strong>Mobilenet</strong></td>
<td>86.67</td>
<td><strong>96.67</strong></td>
<td>95</td>
<td><strong>96.67</strong></td>
<td>92.92</td>
</tr>
<tr>
<td><strong>Inception-v3</strong></td>
<td>80</td>
<td>93.33</td>
<td><strong>96.67</strong></td>
<td>93.33</td>
<td>92.08</td>
</tr>
<tr>
<td><strong>Xception</strong></td>
<td>86.67</td>
<td>88.33</td>
<td>95</td>
<td><strong>96.67</strong></td>
<td>92.92</td>
</tr>
<tr>
<td><strong>VGG16 Fine-tuning</strong></td>
<td>86.67</td>
<td>100</td>
<td>96.67</td>
<td>98.33</td>
<td>98.33</td>
</tr>
<tr>
<td><strong>B.N toVGG16</strong></td>
<td>76.67</td>
<td>91.67</td>
<td>91.67</td>
<td>90.83</td>
<td><strong>91.67</strong></td>
</tr>
</tbody>
</table>

#### 5.4.2. VGG19 model

VGG19 model is very useful for 3C classification task. Its higher V.A was achieved in test2 as 95% when the training data size is 240 images from the center (dataset240), and the L.R is ($10^{-4}$). The CM is constructed in table11. VGG19 model produce 5% error rate, it’s mis predict 3/60 sub-images from the validation data. The three sub-images belong to two different original images, it’s mean that these two original images were correctly predicted by 75% and 87.5% assurance, so the validation data may be considered as 100% correctly predicted.

#### 5.4.3. ResNet50 model

This network is not suitable for the microstructural classification work. It gives the lowest V.A when compared to the other networks. Its best result was achieved at test2, its V.A recorded about 85% and the error rate was 15% for dataset240 at L.R ($10^{-4}$). Its CM is shown in table 12.

#### 5.4.4. Mobilenet model

Mobilenet network has achieved impressive results in this micrograph recognition experiments. In test2, it recorded the highest V.A result (~96.67%) and the error rate was 3.33% with dataset240 at L.R ($10^{-3}$) and with dataset480r at L.R ($10^{-3}$). Table 13 &14 illustrate its CM. From table13, the network predicted two sub-images from class AC4 as AC1 and they are belonging to the same original image, therefore, that original image is correctly classified by 75% assurance. Table 14 also illustrates that for dataset480r (a validation set of 120 images). The system mis predicted only 4 sub-images that belonging two different original images from two different classes. In this case, it’s clearly that these two original images were classified correctly by 87.5% confidence. At the end, it can be assumed that Mobilenet model can classified and validated the spheroidite images correctly by 100% accuracy.
Fig. 8. Validation accuracy results from Test3 using pretrained models and different learning rates.

Fig. 9. Validation accuracy results from Test3 using VGG16 variation and different learning rates.
Table IX
Test3 (3C/20 µm): The V.A for all networks for selected Datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>VGG16</th>
<th>VGG19</th>
<th>ResNet50</th>
<th>Mobilenet</th>
<th>Inception-v3</th>
<th>Xception</th>
<th>VGG16 Fine-tuning</th>
<th>B.N to VGG16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset120</td>
<td>90</td>
<td>93.33</td>
<td>93.33</td>
<td>86.67</td>
<td>80</td>
<td>85</td>
<td>100</td>
<td>96.67</td>
</tr>
<tr>
<td>Dataset240</td>
<td>93.33</td>
<td>91.67</td>
<td>73.33</td>
<td>86.67</td>
<td>80.42</td>
<td>85</td>
<td>96.67</td>
<td>93.33</td>
</tr>
<tr>
<td>Dataset240r</td>
<td>90</td>
<td>94.17</td>
<td>75</td>
<td>91.67</td>
<td>94.58</td>
<td>88.33</td>
<td>95.42</td>
<td>93.75</td>
</tr>
<tr>
<td>Dataset480r</td>
<td>94.17</td>
<td>94.58</td>
<td>69.17</td>
<td>91.67</td>
<td>94.58</td>
<td>88.33</td>
<td>95.42</td>
<td>94.58</td>
</tr>
<tr>
<td>Dataset960r</td>
<td>94.58</td>
<td>94.58</td>
<td>80.42</td>
<td>86.25</td>
<td>94.58</td>
<td>88.33</td>
<td>95.42</td>
<td>96.25</td>
</tr>
<tr>
<td>Dataset1920r</td>
<td>95.42</td>
<td>95.42</td>
<td>88.33</td>
<td>96.25</td>
<td>95.42</td>
<td>94.58</td>
<td>95.42</td>
<td>96.25</td>
</tr>
</tbody>
</table>

5.4.5. Inception-v3 model

Inception-v3 model has classified 3C of spheroidite images with 10µm magnification scale at (test2) by the V.A about 96.67% and 3.33% as an error rate for the dataset240r at the L.R. (10^5). Table 15 indicates that two sub-images from class AC1 and AC4 were mis predicted by Inception-v3 model. Again, the mis predicted sub-image is represent only 12.5% from each original image, so the two original images are 87.5% correctly classified. The V.A of this model is nearly considered as 100%.

5.4.6. Xception model

This network follows suit the Inception-v3 model where its V.A also being 96.67% for test2 classification while the dataset was Dataset480r at L.R (10^5). Its CM is shown in table 16. Based on the previous concept that was discussed with the other models, the V.A for this model may considered as 100% instead of 96.67% because the 4 sub-images that were mis classified represent 25% only of the original image.

5.4.7. VGG16 Fine-tuning

In general, fine tuning for VGG16 has improved its result, specially at test1. It increased the V.A of the VGG16 model in test1 from 87.5% to 96.67% while the highest V.A was achieved by the pretrained models was about 90.5% by VGG19 and Mobilenet models. Fine tuning to VGG16 reached 96.67% as a V.A and the error rate was 3.33% with dataset480 at L.R (10^5). The CM in this case is illustrated in table17. Fine-tuning approach achieved 100% V.A results when it is used for 3C classification tasks.

5.4.8. VGG16 with Batch normalization

This approach reaches its higher V.A result (~ 96.67%) at the test3 with the dataset120 and the L. R (10^-5). Table18 introduces its CM. From that table, it can be noted there is a one section of original image that belong to AC2 class was mis predict, so it can be considered that this original image is correctly classified by 75% confidence and that error can be neglected. This approach can classify 3C spheroidite images with 20µm magnification scale with nearly 100% validation accuracy.

Table X
CM for VGG16 model.

<table>
<thead>
<tr>
<th>Actual class label</th>
<th>AC2</th>
<th>AC3</th>
<th>AC6</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC2</td>
<td>19 (95%)</td>
<td>0%</td>
<td>0%</td>
<td>95%</td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>20 (100%)</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>AC6</td>
<td>1 (5%)</td>
<td>0%</td>
<td>20 (100%)</td>
<td>100%</td>
</tr>
<tr>
<td>Class recall</td>
<td>95%</td>
<td>100%</td>
<td>100%</td>
<td>Total accuracy</td>
</tr>
</tbody>
</table>
### Table XI
CM for VGG19 model.

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>20 (100%)</td>
<td>0%</td>
<td>3 (15%)</td>
<td>86.96%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>20 (100%)</td>
<td>0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>0%</td>
<td>0%</td>
<td>17 (85%)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>100%</td>
<td>100%</td>
<td>85%</td>
<td>Total accuracy 95%</td>
<td></td>
</tr>
</tbody>
</table>

### Table XII
CM for Resnet50 model.

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>16 (80%)</td>
<td>0%</td>
<td>5 (25%)</td>
<td>76.19%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>1 (5%)</td>
<td>20 (100%)</td>
<td>0%</td>
<td>95.24%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>3 (15%)</td>
<td>0%</td>
<td>15 (75%)</td>
<td>83.33%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>80%</td>
<td>100%</td>
<td>75%</td>
<td>Total accuracy 85%</td>
<td></td>
</tr>
</tbody>
</table>

### Table XIII
CM for Mobilenet model with dataset240 and LR (10^{-5}).

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>20 (100%)</td>
<td>0%</td>
<td>2 (10%)</td>
<td>90.9%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>20 (100%)</td>
<td>0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>0%</td>
<td>0%</td>
<td>18 (90%)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>100%</td>
<td>100%</td>
<td>90%</td>
<td>Total accuracy 96.67%</td>
<td></td>
</tr>
</tbody>
</table>

### Table XIV
CM for Mobilenet model with dataset480r and LR (10^{-3}).

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>38 (95%)</td>
<td>0%</td>
<td>2 (5%)</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>1 (2.5%)</td>
<td>40 (100%)</td>
<td>0%</td>
<td>97.56%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>1 (2.5%)</td>
<td>0%</td>
<td>38 (95%)</td>
<td>97.44%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>95%</td>
<td>100%</td>
<td>95%</td>
<td>Total accuracy 96.67%</td>
<td></td>
</tr>
</tbody>
</table>

### Table XV
CM for Inception-v3 model.

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>19 (95%)</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>20 (100%)</td>
<td>1 (5%)</td>
<td>95.24%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>1 (5%)</td>
<td>0%</td>
<td>19 (95%)</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>95%</td>
<td>100%</td>
<td>95%</td>
<td>Total accuracy 96.67%</td>
<td></td>
</tr>
</tbody>
</table>

### Table XVI
CM for Xception model.

<table>
<thead>
<tr>
<th>Predicted class label</th>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>40 (100%)</td>
<td>0%</td>
<td>4 (10%)</td>
<td>90.91%</td>
<td></td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>40 (100%)</td>
<td>0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>AC4</td>
<td>0%</td>
<td>0%</td>
<td>36 (90%)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Class recall</td>
<td>100%</td>
<td>100%</td>
<td>90%</td>
<td>Total accuracy 96.67%</td>
<td></td>
</tr>
</tbody>
</table>
6. Conclusion

This work validates the suitability of the deep learning methods for the mission of microstructure classification without need to the separate segmentation and the feature extraction. Three tests are applied using different FCNNs pretrained models to the task of micrograph recognition for different spheroidite datasets. The transfer learning approach is utilized here with two different scenarios (using the pretrained model directly and the fine-tuning).

By comparing and evaluating the selected FCNNs, it was found that the ResNet50 network isn’t appropriate for microstructure classification task. In contrast the Mobilenet model achieves surprisingly results in these classifications despite of the application it was founded for (mobile and embedded vision). This model achieves about 90.5% as a V.A for 6C and achieves 96.67% and 94.58% as a V.A for 3C at 10 ㎛ & 20 ㎛ magnification scale values, respectively. But, it important to mention that Mobilenet model often needed a large training data to reach such satisfied results.

Also, the random cropping method is useful to overcomes the lack of details per image when the image taken by high magnification scale and it’s useful also for some models that need large training data.

Based on the tests results, we can say that the all classes of spheroidite micrograph can be identified very well and they can be classified using pretrained models with highly validation accuracy. At test 3, VGG16 recorded the higher V.A (~ 98.33%) while the error rate was 1.66% and L.R (10⁻⁴) with a small dataset (Dataset240r). So, the VGG16 model is the appropriate pretrained model that can be used for microstructural images classification at the large magnification scale value (20㎛) and the small data size (Dataset240). Also, the VGG19 model is convenient for 6C at magnification scale 10 ㎛. It recorded the highest V.A in test1(~ 90.83%) while the training data was dataset240, and

<table>
<thead>
<tr>
<th>Actual class label</th>
<th>AC1</th>
<th>AC3</th>
<th>AC4</th>
<th>AC5</th>
<th>AC6</th>
<th>AC7</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC1</td>
<td>20 (100%)</td>
<td>0%</td>
<td>1 (5%)</td>
<td>0%</td>
<td>0%</td>
<td>1 (5%)</td>
<td>90.91%</td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>20 (100%)</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>AC4</td>
<td>0%</td>
<td>0%</td>
<td>19 (95%)</td>
<td>0%</td>
<td>1 (5%)</td>
<td>0%</td>
<td>95%</td>
</tr>
<tr>
<td>AC5</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>20 (100%)</td>
<td>0%</td>
<td>1 (5%)</td>
<td>95.24%</td>
</tr>
<tr>
<td>AC6</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>19 (95%)</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>AC7</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>18 (90%)</td>
<td>100%</td>
</tr>
<tr>
<td>Class recall</td>
<td>100%</td>
<td>100%</td>
<td>95%</td>
<td>100%</td>
<td>95%</td>
<td>90%</td>
<td>Total accuracy 96.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actual class label</th>
<th>AC2</th>
<th>AC3</th>
<th>AC6</th>
<th>Class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC2</td>
<td>9 (90%)</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>AC3</td>
<td>0%</td>
<td>10 (100%)</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>AC6</td>
<td>1 (10%)</td>
<td>0%</td>
<td>10 (100%)</td>
<td>90.91%</td>
</tr>
<tr>
<td>Class recall</td>
<td>90%</td>
<td>100%</td>
<td>100%</td>
<td>Total accuracy 96.67</td>
</tr>
</tbody>
</table>
the L.R is \((10^{-3})\). The fine-tuning approach contributed to raising the validation accuracy of VGG16 the three tests. It was recorded 96.67% for 6C and 100% for 3C while the L.R was \(10^{-3}\).

**REFERENCES**


