Coke characterization: Segmentation of pores and constituents from microscopic images of Coke

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science

by

Luna Biswas
CS2005

under the guidance of

Professor Dipti Prasad Mukherjee
Electronics and Communication Sciences Unit
Indian Statistical Institute

Indian Statistical Institute
Kolkata - 700108 , India

Department of Electronics and
Communication Sciences Unit
Indian Statistical Institute, Kolkata
India - 700108

CERTIFICATE

This is to certify that we have examined the thesis entitled **Coke characterization: Segmentation of pores and constituents from microscopic images of Coke**, submitted by **Luna Biswas** (Roll Number: *CS2005*) a postgraduate student of **Department of Electronics and Communication Sciences Unit** in partial fulfillment for the award of degree of Master of Technology. We hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfillment for the Post Graduate Degree for which it has been submitted. The thesis has fulfilled all the requirements as per the regulations of the Institute and has reached the standard needed for submission.

22/07/2022

Professor Dipti Prasad Mukherjee
Electronics and Communication Sciences Unit
Indian Statistical Institute
Kolkata - 700108
India

ACKNOWLEDGEMENTS

I would like to show my highest gratitude to my advisor, Prof. Dipti Prasad Mukherjee, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. I have learnt tremendously on every field from him. He has taught me how to do good research, and motivated me with great insights and innovative ideas.

I would also like to thank Dr. Swagatam Das, Professor, Indian Statistical Institute, Kolkata, for his valuable suggestions and discussion during the mid-term evaluation and for teaching machine learning and throwing torch lights on the door ways of the latest developments in deep learning and for providing valuable references.

I would like to thank Dr. Pradipta Maji, Professor, Indian Statistical Institute, Kolkata, for teaching image processing, without which I would not have the courage to take up this dissertation, and for his valuable suggestions and discussions.

My deepest thanks to all the teachers of Indian Statistical Institute, for their priceless suggestions, discussions and teaching which added an important dimension to my research work. I would like to acknowledge Aditya Panda, Suman Ghosh, Sankarsan Seal and all other seniors at the lab for their constant guidance. Finally, I am very much thankful to my parents and family for their everlasting support. Last but not the least, I would like to thank all of my friends for their help and encouragement. I thank all those, whom I have missed out from the above list.

Luna Biswas
Indian Statistical Institute
Kolkata - 700108
India
Date:

ABSTRACT

Coke is mainly used in steel industry as a fuel and a reducing agent for melting iron in the blast furnace, since it generates intense heat but little smoke. The quality of the coke material (like porosity, wall thickness, texture etc., as seen in a microscopic image of coke) affects the performance of blast furnace impacting the profit/loss of the industry. Therefore it is important to determine the structure and porosity of coke on a large scale. Manual process of coke characterisation is costly and slow. Automation of coke characterization, from microscopic images of cokes, is beneficial for the steel industry. An attempt has been made to calculate porosity of coke from the images, and produce semantic segmentation of the coke images into different types of metallurgical textures like inert, incipient, circular, lenticular etc. A shallow convolutional neural network (CNN) was trained with annotated coke images using cross entropy loss (between the probability distributions of the predictions out of the CNN and the target as per annotation, for different classes). A new contrastive loss function has been written, that maximises entropy between the probability distribution of a training sample with another sample belonging to a different class, in addition to minimising entropy loss between the probability distributions of the predictions and the target. This new loss function enables faster learning, and useful when quantity of annotations for training a model, is less. A shallow CNN model obtained higher accuracy in prediction of class for each pixel of the coke images, and the granularity of semantic segmentation was reduced when trained using this novel loss function.

Keywords: automated coke characterization, semantic segmentation, contrastive training, loss, image processing, convolutional neural network.

Contents

1 Introduction **1**

1.1 Coke carbon forms 1

1.2 Problem statement 5

1.3 Our solution 5

1.3.1 Identification of pores and calculation of porosity 6

1.3.2 Segmentation of non-pore regions 6

2 Related works **9**

3 Methods **13**

3.1 Identification of pores and calculation of porosity 13

3.1.1 Identification of pores 13

3.1.2 Calculation of porosity 14

3.2 Collecting sample patches for training a shallow CNN [4] model . . . 16

3.3 Determining class of a sample patch 19

3.4 Choice of the activation function used for non linearity in the shallow CNN [4] model 21

3.5 Choice of model hyper-parameters 23

3.5.1 Choice of input patch size and number of intermittent channels 23

3.5.2 Choice of optimizer and learning rate 25

3.5.3 Loss function 25

3.5.3.1 Cross Entropy Loss (CE) 26

3.5.3.2 Custom Cross Entropy Loss 27

3.5.4 Deciding optimum number of epochs for training 29

3.6 Training procedure 30

4 Results **33**
4.1 Performance metrics 33
4.2 Qualitative results 35

5 Conclusions **43**

Bibliography **44**

List of Figures

1.1 Size specifications of coke forms 2
1.2 Sample inert, incipient, circular, lenticular and pore regions 3
1.3 Inerts. Porous inerts display a blackish shade, and non-porous inerts display whitish shade in a microscopic image. 4
1.4 When scientists take photos under the microscopes, light gets reflected from bottom of the pores. These reflections are visible in the microscopic images as lighter shade within dark pores. Pores can be distinguished from minerals by inspecting such lighter shaded objects within the large black regions. 5
1.5 Flow chart for the process of identification of pores and calculation of porosity from coke images. 6
1.6 Flow chart for the segmentation of non-pore regions from coke images. 7

2.1 Baseline CNN model. 10
2.2 Semantic segmentation of non-pore regions of coke images into incipient, inert, circular and lenticular classes. 11
2.3 Examples of 49 x 49 patches extracted from the image. 12

3.1 Images out of each step of the porosity calculation process. 15
3.2 Porosity histogram with pore distances from Fig. 3.1f 16
3.3 Automated patch collection process. 17
3.4 A patch is added to the training data, if it belongs to a single class, or else it is rejected. 18
3.5 5-Fold cross validation. 20
3.6 Confusion matrix. 21
3.7 Selu activation fuction. 22

3.8 The Baseline CNN model parameters : input patch size and number of channels in between two convolution layers. 24
3.9 Cross entropy loss minimises entropy between predicted and target distributions. 27
3.10 Contrastive cross entropy loss maximises entropy between distributions of prediction and a target belonging to a different class; in addition to minimising entropy between predicted and target distributions. 28
3.11 CNN model (Fig. 2.1) performance with 1000 epochs of training. . . . 30
3.12 The training process. 32

4.1 Loss, accuracy, precision, recall on train data after training the models for 150 epochs. 34
4.2 Precision and recalls obtained on test data using the models trained with different losses. 35
4.3 (a) Original image, (b) binary image showing pores in black color and non-pores in white, (c) manual annotations (ground truth), Color code: grey (inert), red (circular), green (lenticular)(d) the segmented image using the CNN model, with Cross Entropy loss. Color code: black (pores and incipients), blue (inert), red (circular), green (lenticular), (e) the segmented image using the CNN model, with the Contrastive Cross Entropy loss (Sub), (f) polygonal approximation of pores in green lines, and the distances between pores in yellow, (g) porosity histogram with pore distances calculated from (f). 36
4.4 (a) original image, (b) segmented image by CCE-Sub. The inert regions, marked with green border, are classified accurately. But the ones marked in red border, are the failure cases, where inert regions are not identified fully as inert, but as a grainy segmentation, consisting of inert, circular and lenticular regions. 38
4.5 (a) original, (b) segmented by CCE-Sub. Incipient regions are correctly identified (marked with green border). Other regions are properly segmented too, except that the whole segmented image is grainy 40

4.6 (a) original image, (b) binary image with pores identified in black, (c) segmentation by CCE-Sub. Some of the pore regions are not marked as pore by the pore identification process, due to high intensity regions touching the border. These regions are wrongly identified as inerts by the CNN model, due to their uniform texture. Some inert regions are marked as pore, by the pore identification process, which could have been classified as inert by the CNN otherwise. This coke image contains maximum of the non-pore regions as incipient, which are correctly classified by the CNN model. 41
4.7 (a) original image, (b) segmented image from CCE-Sub. Majority of the non-pore region is incipient, and classified accurately. Inerts are also identified properly (marked with green border). But some pores (red border) and a small portion of incipient in the left bottom portion of the image are classified as inert. 42

Chapter 1

Introduction

The steel making industry relies on the quality of coke in terms of [1] its ability to (i) produce heat for melting the ore, to reduce the ore to metal, (ii) to import permeability to facilitate the reactions and (iii) to bear the load of the charge in the blast furnace. [2] The carbonisation/coking of coals produces cokes that exhibit a variety of microscopic textures whose optical behavior in polarized light aids in their characterization. The word texture relates to the carbons' optical properties, while structure relates to the amount and size of coke pores and walls. Many properties of carbons such as their graphitizability, their electrical resistivity, reactivity to CO_2 at elevated temperatures and strength are related to the optical properties of the coke carbon forms. To measure the characteristics of coke materials (like porosity, strength, wall thickness, classification of binder-phase carbon form), coke micro structures are studied under microscopes. The binder phase carbon forms can be classified into different forms like inerts, isotropic, incipient, circular, lenticular, ribbon structures.

1.1 Coke carbon forms

The specifications of different carbon forms are tabulated in Fig. 1.1 [3].

TABLE 7.1
System of coke microscopy

Coke Binder-Phase Carbon Form Classification			
Binder Phase	*Width (in Microns)*	*Length (L) to Width (W) Relation*	*Parent Coal Vitrinoid Type*
Isotropic	0.0	None	6, 7
Incipient (anisotropic)	0.5	$L = W$	8
Circular (anisotropic)			
Fine circular	0.5–1.0	$L = W$	9
Medium circular	1.0–1.5	$L = W$	10
Coarse circular	1.5–2.0	$L < 2W$	11
Lenticular (anisotropic)			
Fine lenticular	1.0–3.0	$L \geq 2W$, $L < 4W$	12
Medium lenticular	3.0–8.0	$L > 2W$, $L < 4W$	13
Coarse lenticular	8.0–12.0	$L > 2W$, $L \leq 4W$	14
Ribbon (anisotropic)			
Fine ribbon	2.0–12.0	$L > 4W$	15
Medium ribbon	12.0–25.0	$L > 4W$	16
Coarse ribbon	25.0+	$L > 4W$	17, 18

Coke Filler-Phase Carbon Form Classification		
Filler Phase	*Size (Microns)*	*Precursors*
Organic inerts		
Fine	<50	Micrinite, macrinite, inertodetrinite
Coarse	>50	Semifusinite, fusinite durain
Miscellaneous inerts		
Oxidized coal (coke)		Oxidized and brecciated coal
Brecciated coal (coke)		
Noncoking vitrinite (coke)		Vitrinite too high or low in rank
Inorganic inerts		
Fine	<50	Mineral matter and coal bone
Coarse	>50	

*Miscellaneous categories, including carbon additives, depositional carbons, and green and burnt coke, may be quantified.
Source: *Organic Geochemistry* 17, R. J. Gray, "Some petrographic applications to coal, coke, and carbons," 535–555, copyright 1991, with permission from Elsevier.

Figure 1.1: Size specifications of coke forms

Some examples of incipient, inert, circular, lenticular and pore regions are shown in Fig. 1.2. Incipient means beginning to develop. As described in [2], incipient carbon forms are < 0.5 microns, and appear faintly textured when rotated in polarized light. Circular anisotropic domains are relatively circular in outline and increase in size from 0.5 to 2.0 microns. The binder phase carbons produced from medium volatile coals, are lenticular in shape having widths that range from 1.0 to 12.0 microns, with a length (L) to width (W) ratio of 2 to 4.

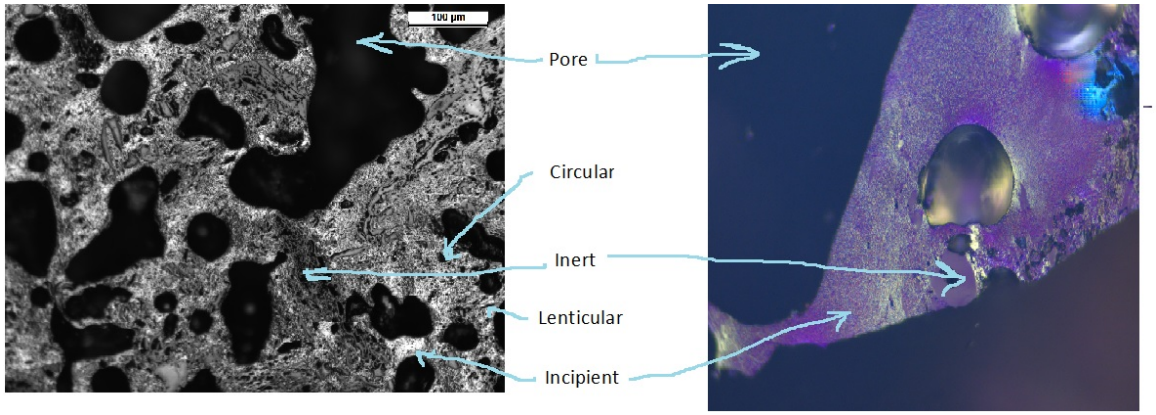


Figure 1.2: Sample inert, incipient, circular, lenticular and pore regions

Inert regions have solid white and black structures (which are minerals inside the inertinites). If the region is porous, then it is in more black shade than the surroundings, otherwise, if not porous, then it is whiter. Examples of porous and non-porous inerts are shown in Fig. 1.3.

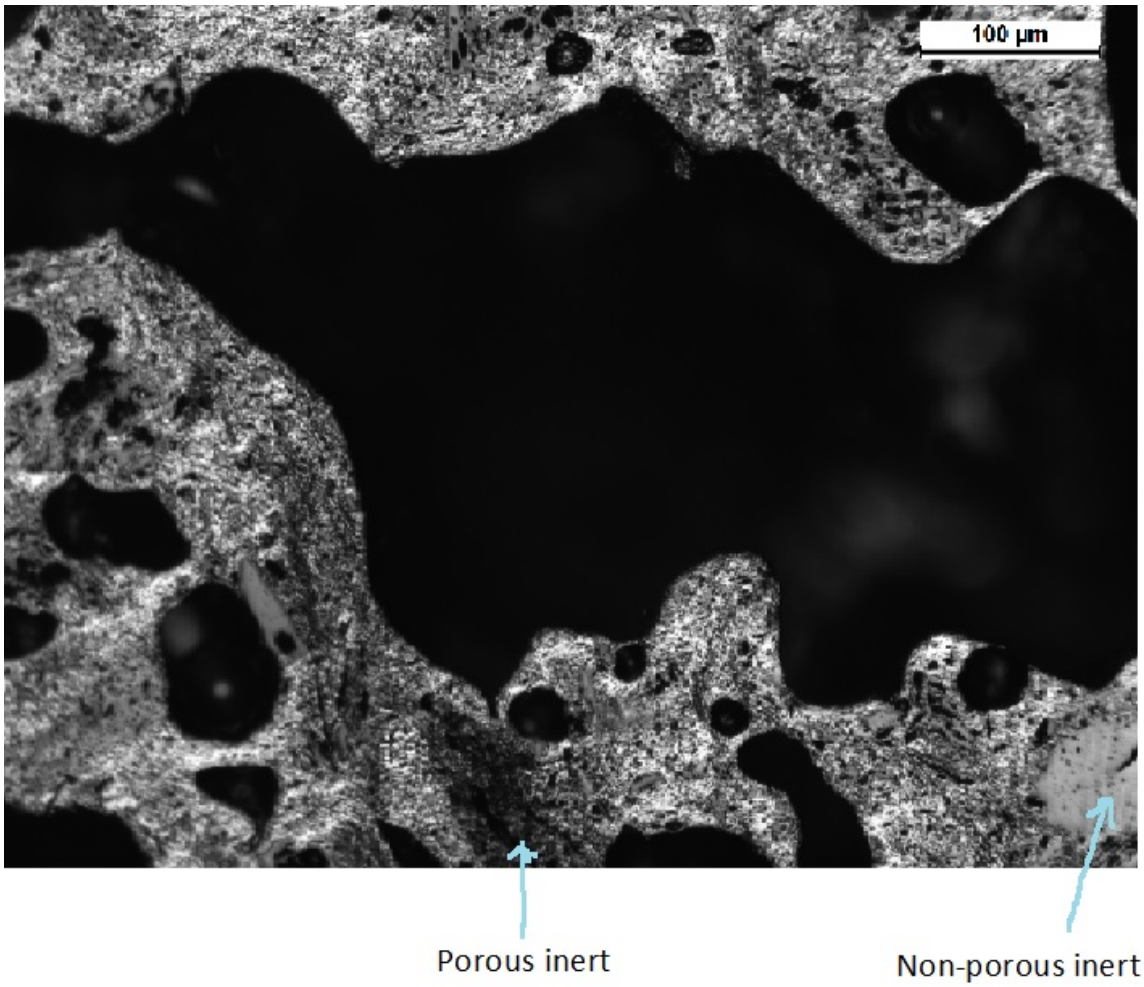


Figure 1.3: Inerts. Porous inerts display a blackish shade, and non-porous inerts display whitish shade in a microscopic image.

Pores are black or brown, round or oval (if multiple round pores are joined together). Pores should have floating, pale-white regions visible inside, which are reflections of light in the microscope from bottom of the pores (shown in Fig. 1.4.).

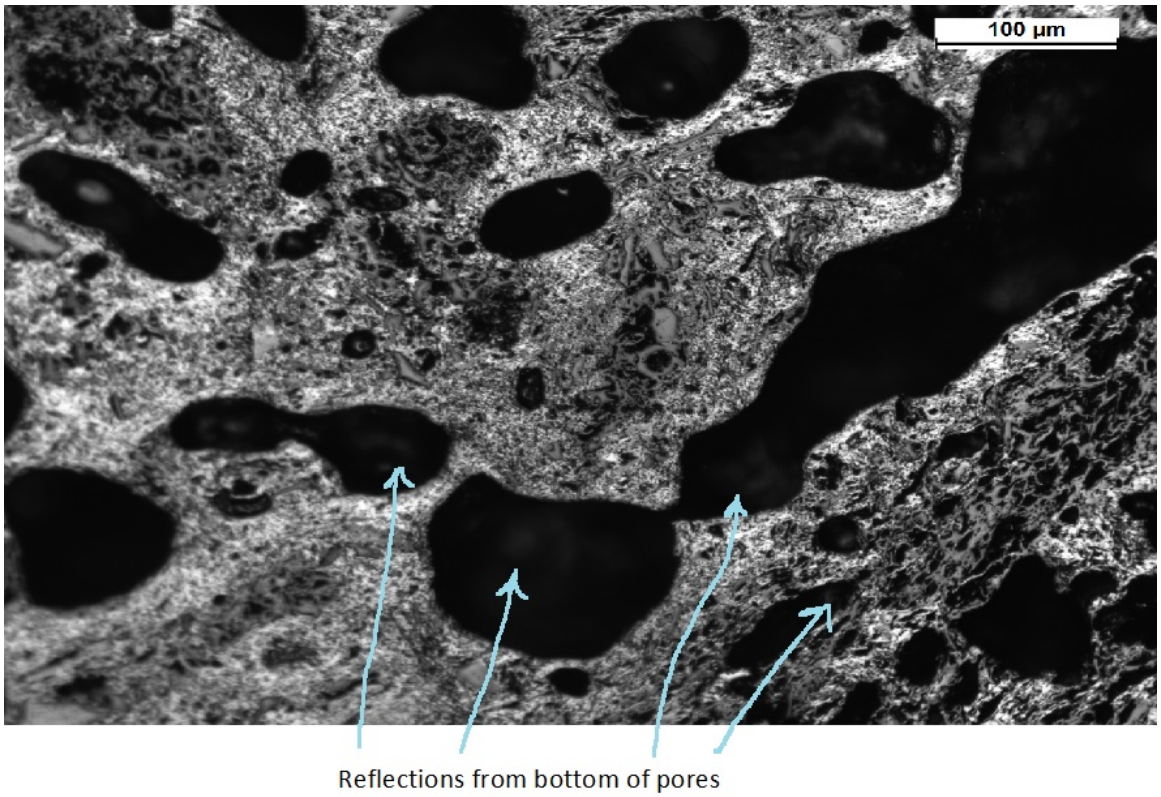


Figure 1.4: When scientists take photos under the microscopes, light gets reflected from bottom of the pores. These reflections are visible in the microscopic images as lighter shade within dark pores. Pores can be distinguished from minerals by inspecting such lighter shaded objects within the large black regions.

1.2 Problem statement

The goal of this thesis is to automate (1) calculation of porosity, and (2) semantic segmentation based on binder phase carbon form, from the microscopic images of coke.

Since the hue and intensity values of pores and minerals are similar in the coke images, the segmentation problem is hard.

1.3 Our solution

Our approach is to first separate out the pores from carbon regions and then measure distance between pores to calculate porosity; and pass the non-pore regions to a

convolutional neural network (CNN) [4] for further segmentation into inert, incipient, circular and lenticular regions.

1.3.1 Identification of pores and calculation of porosity

A flow chart for the porosity calculation process is added in Fig. 1.5. First blacker regions are identified from the original coke images, using Otsu threshold [5]. Then pores are selected from the identified black region, based on their bigger size and reflections (lighter shaded regions) within them (as shown in Fig. 1.4). After pores identification process, to calculate porosity of the coke, distance between each pair of pores are measured and reported in a histogram.

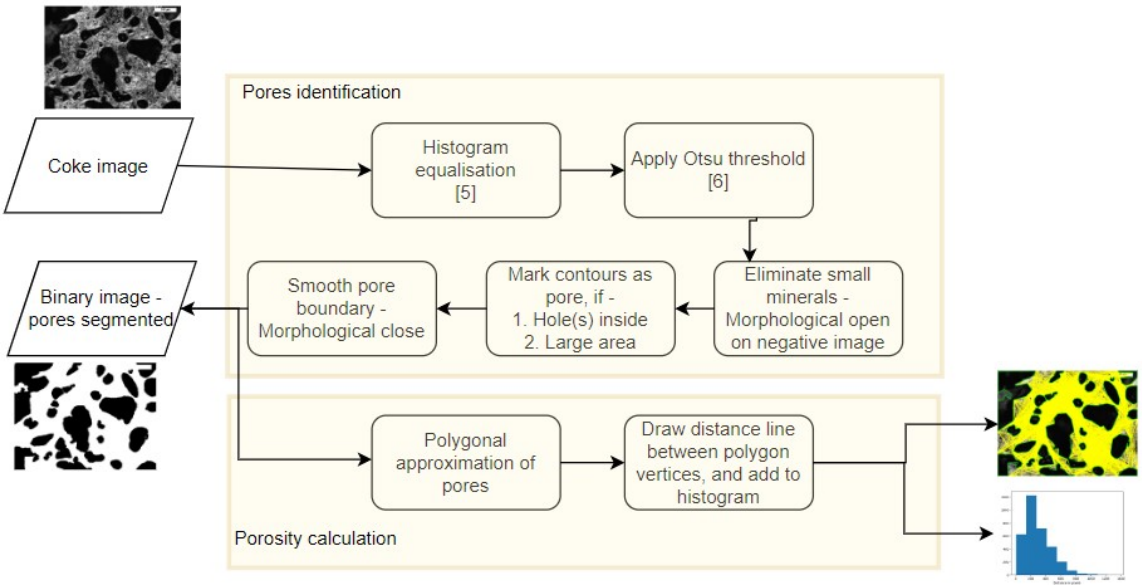


Figure 1.5: Flow chart for the process of identification of pores and calculation of porosity from coke images.

The details process is explained in Section 3.1.

1.3.2 Segmentation of non-pore regions

Fig. 1.6 explains the approach to segment non-pore regions further. First a few regions of the coke images are annotated manually. Then 49×49 patches and corresponding targets are extracted from the images and a shallow CNN [4] model is trained using

them. For inference, patches from non-pore regions are extracted and their classes are predicted using the trained model.

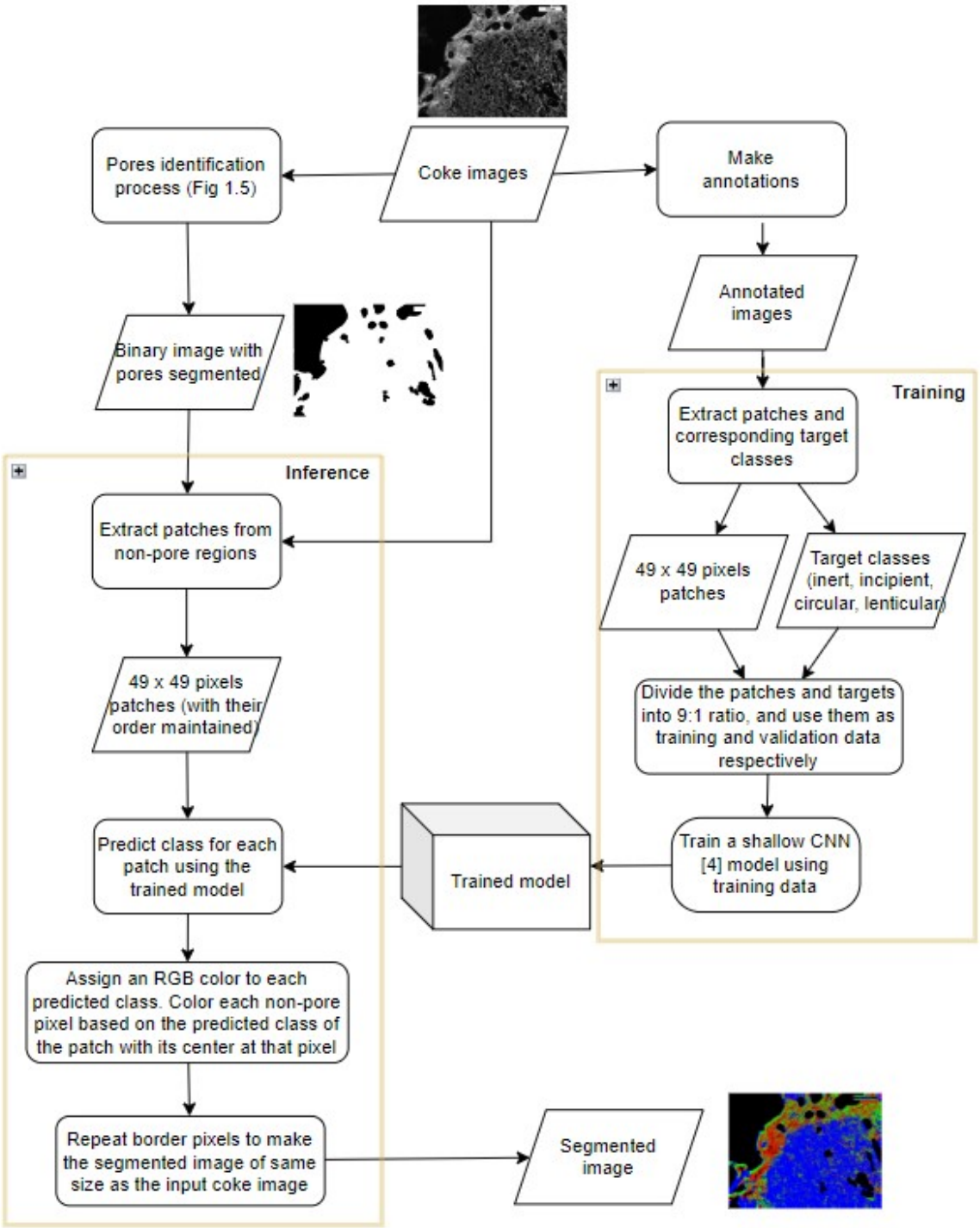


Figure 1.6: Flow chart for the segmentation of non-pore regions from coke images.

Next we present related work in Chapter 2, followed by the details of methodology in Chapter 3. Finally experimental results and its analysis are shown in Chapter 4, followed by conclusions in Chapter 5.

Chapter 2

Related works

One of the major challenges for segmentation of coke images, is that the intensity levels of pores and minerals are similar, and the whole images are blackish. Based on different parameters, an automated tool [6] have been published by Anderson et al. to characterise coke images, however it takes a binary image in the input with pores as black and carbon as white. Our method takes a color image with three different channels (red, green, blue) in the input and automatically identifies pores, and measures the porosity and produces a histogram. Based on different parameters, [7], [8] identify pores from coke images and characterize coke micro structures, but the images used as input to their methods have different intensities for the pores than non-pore regions, and identification of pores were straight forward. Moreover no paper has been published to segment out regions further in classes like circular, lenticular etc. based on shapes of the carbon mineral.

A deep learning convolutional neural network [4] can learn the texture representations of the non-pore regions, if tens of thousands of training data is available. But we have only 21 coke images available. There have been plenty of work done on zero or few shot image classification, [9]–[13] are only a few of them. All of them use annotated data to create a latent space, and then some criteria about the new class (with no or very few annotations) is used to map the new class in the latent space. During training, this latent space is learnt, and during prediction, the input image data is mapped to the learnt latent space, and classified based on distance from the identified classes in the latent space. To build the latent space during training, enough data of existing available classes were used. But in our case, only a few annotations were available.

To mitigate the challenge of less data, the images are broken into small overlapping patches (small enough to contain a single class of texture in it, and large enough to capture the characteristics of the texture for the class). One image of size 1360×1024 bytes can be broken into $12,80,512$ overlapping patches of size 49×49 pixels. A box of size (49×49) can slide across the image, $1360 - 2(\frac{49-1}{2}) = 1,312$ times on a single column and $1024 - 2(\frac{49-1}{2}) = 976$ times on a single row, resulting in $1312 \times 976 = 12,80,512$ sliding positions. A patch can be extracted from each of these sliding box positions. In order to train a model, a small portion of data is annotated and patches were extracted from the annotated regions only.

A shallow convolutional neural network model [4] is used as the baseline model, as shown in Fig. 2.1. The first layer convolutes the 49×49 patches with 21 numbers of 7×7 filters with stride 4, and transforms the patches to 11×11 files. 21 such files are created, matching the number of filters used. Non linearity in the transformation is introduced using scaled exponential linear unit (Selu) [14] activation function. More details of this activation function is given in Section 3.4 of Chapter 3. The second layer consists of 4 filters of size 7×7 pixels, applied with strides 5, they transform the 11×11 input files to 4 number of 1×1 output data. These 4 numbers represent the weights corresponding to the 4 output classes. Finally a softmax [15] is applied to determine the predicted class, having maximum weight among all.

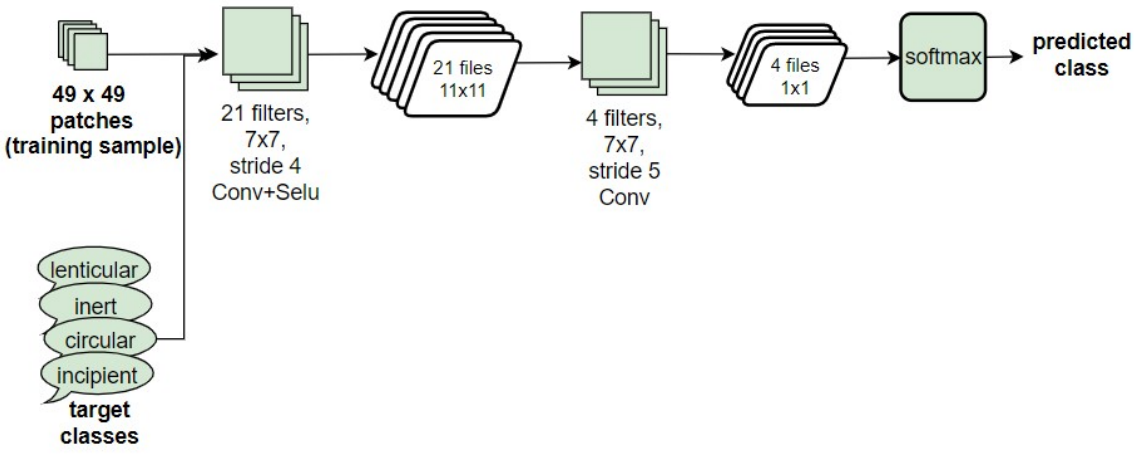


Figure 2.1: Baseline CNN model.

The whole process of extracting training patches, training and inference is described in Fig. 2.2.

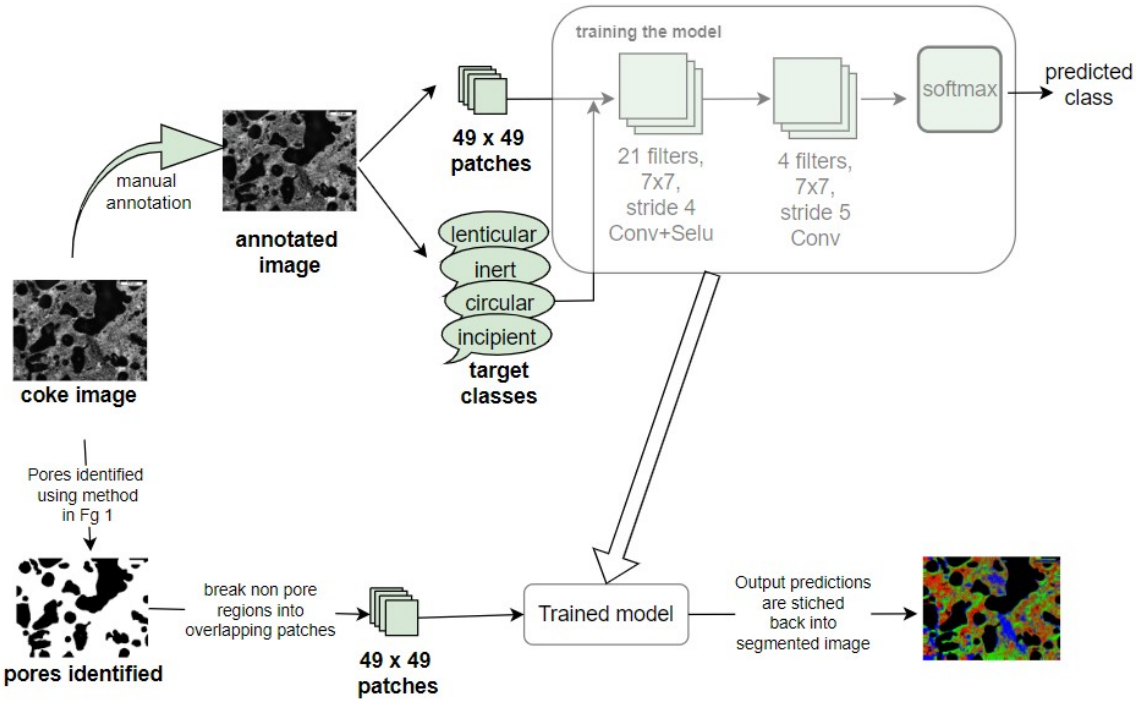
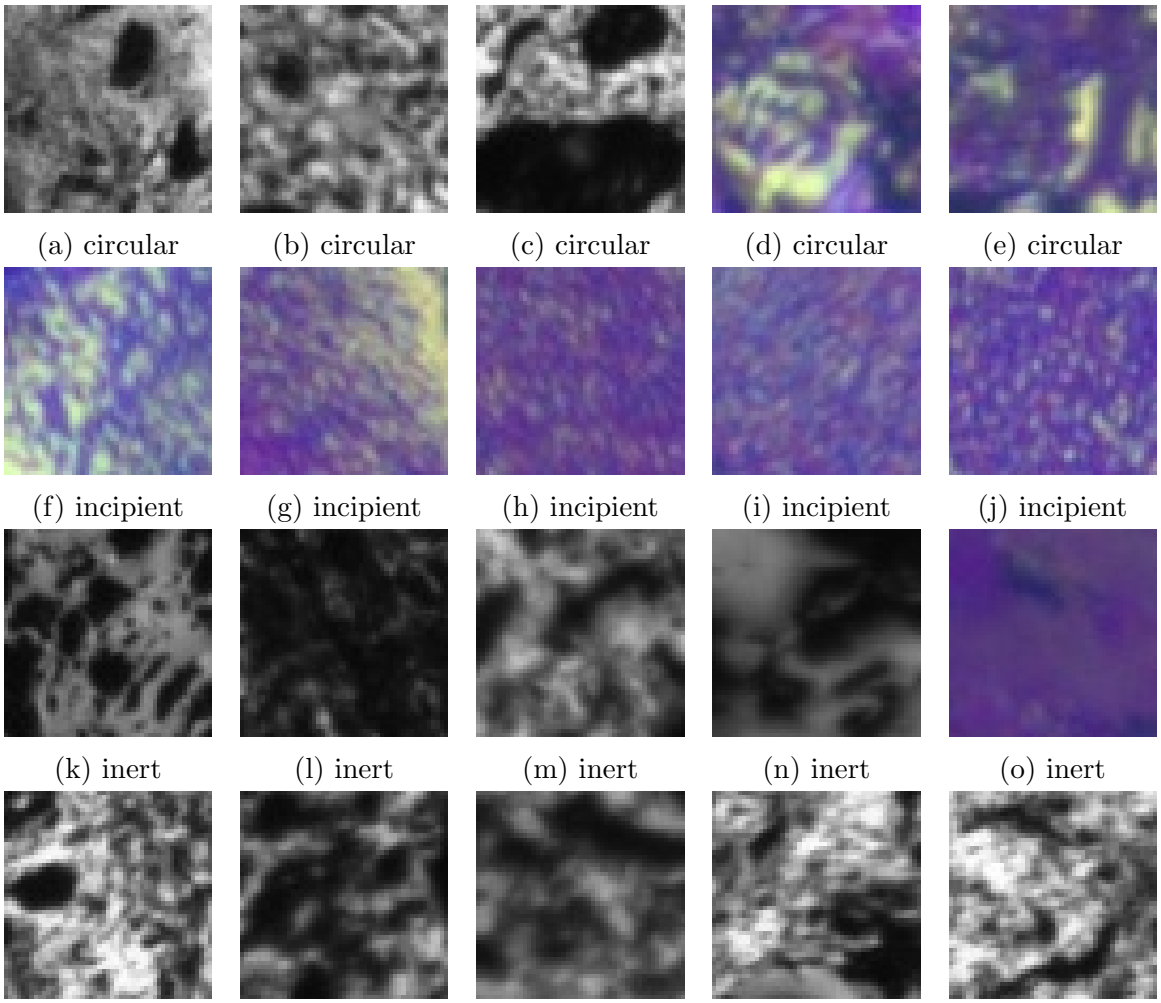


Figure 2.2: Semantic segmentation of non-pore regions of coke images into incipient, inert, circular and lenticular classes.

Sample training patches (examples in Fig. 2.3) are passed through the model and predictions are compared with targets for training. During inference, for each non-pore pixel in the image, a 49×49 patch is extracted keeping that pixel in its center. Classes for the non-pore patches are predicted using the trained model, and a semantically segmented image is created at the output accordingly.



(a) circular (b) circular (c) circular (d) circular (e) circular

(f) incipient (g) incipient (h) incipient (i) incipient (j) incipient

(k) inert (l) inert (m) inert (n) inert (o) inert

(p) lenticular (q) lenticular (r) lenticular (s) lenticular (t) lenticular

Figure 2.3: Examples of 49 x 49 patches extracted from the image.

Chapter 3

Methods

3.1 Identification of pores and calculation of porosity

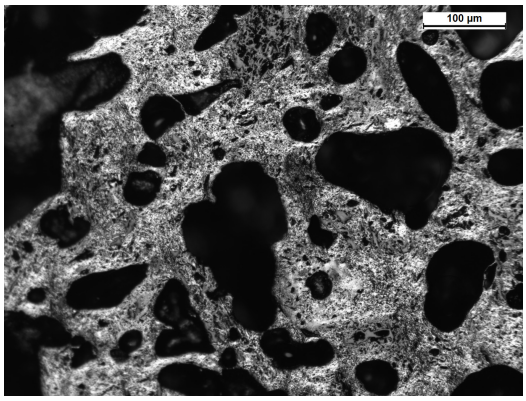
The porosity calculation process is shown in Fig. 1.5. First the pores are identified, and segmented from the coke image, and then porosity of the coke is calculated.

3.1.1 Identification of pores

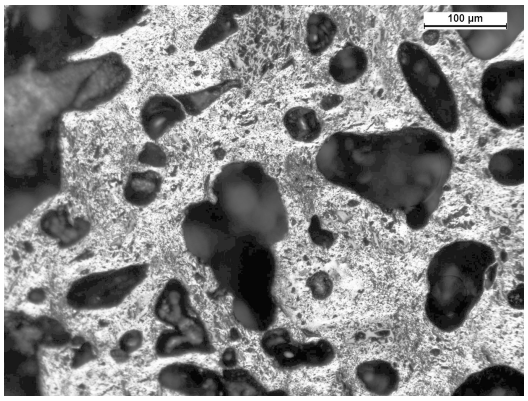
To identify pores, in the beginning, histogram equalisation [16] is performed on the input coke image to improve contrast, then Otsu's threshold [5] is applied on intensity levels to separate out pore-like black regions. Since pores are usually bigger than minerals, a morphological open operation [17] is performed on the negative of the binary image, to eliminate smaller regions, representing minerals. As shown in Fig. 1.4, pores are large, and consist of reflections of lighter shade inside them. They can be distinguished as large contours with holes inside them. So contours having hole(s) inside, are first identified as pores. Then the size of other contours (with no hole inside), are checked. If the area of the contour is more than 0.19% of the whole image, then the contour is chosen as a pore, or else it is identified as a non-pore region. The threshold (0.19%) is fixed based on manual inspection of the available coke images. A binary image, with pores segmented (in black colour) from non-pore regions (in white colour), is shown in Fig. 3.1e. All images obtained from each of the internal steps, as discussed above, are shown in Fig. 3.1.

3.1.2 Calculation of porosity

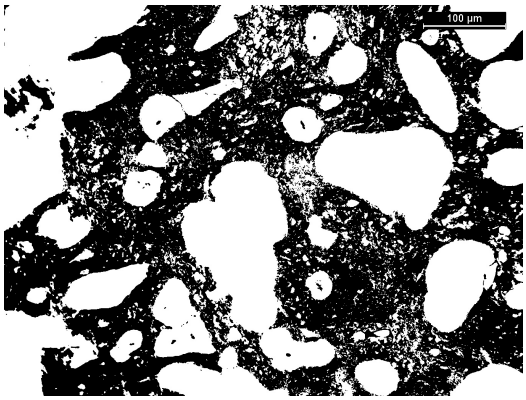
To calculate porosity of the coke, thickness of each pore wall is calculated as the distance between every pair of pores. Pores are first approximated as polygons. Accuracy for polygonal approximation [18], is taken as 0.1, ensuring that the maximum distance from the approximated polygon to the actual pore border curve, is 0.1% of the perimeter of the pore [19]. The polygons are drawn on the original image, in green colour, as shown in Fig. 3.1f. A yellow line is drawn joining each vertex of every pairs of polygons. If the yellow line is crossing any pore, then that line is discarded and not drawn. The length of these yellow lines measure the distances between every pair of pores. These distances are reported using a histogram (in Fig. 3.2). This histogram portrays the porosity and wall thickness of the coke.



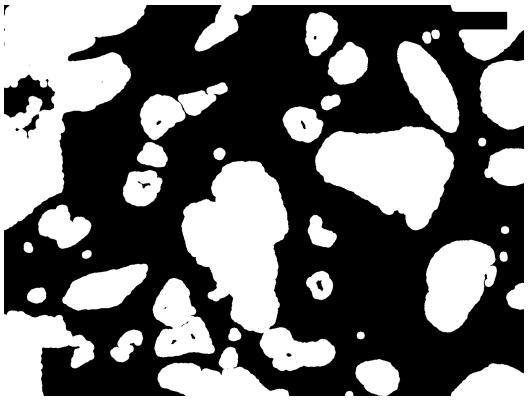
(a) Original coke image



(b) Improved contrast after histogram equalisation

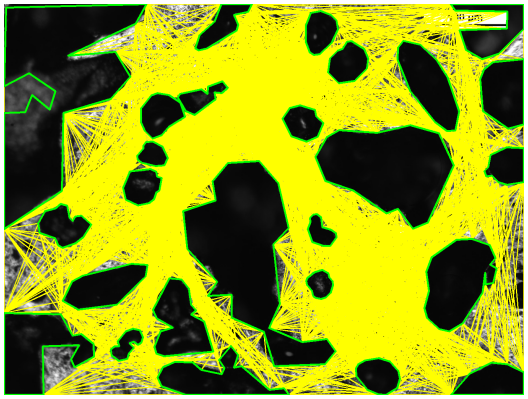


(c) Otsu's threshold applied



(d) Smaller regions removed after morphological open

(e) Pores identified as large contours and medium ones with hole inside



(f) Polygonal approximation of pores in green, and distances between polygon vertices in yellow

Figure 3.1: Images out of each step of the porosity calculation process.

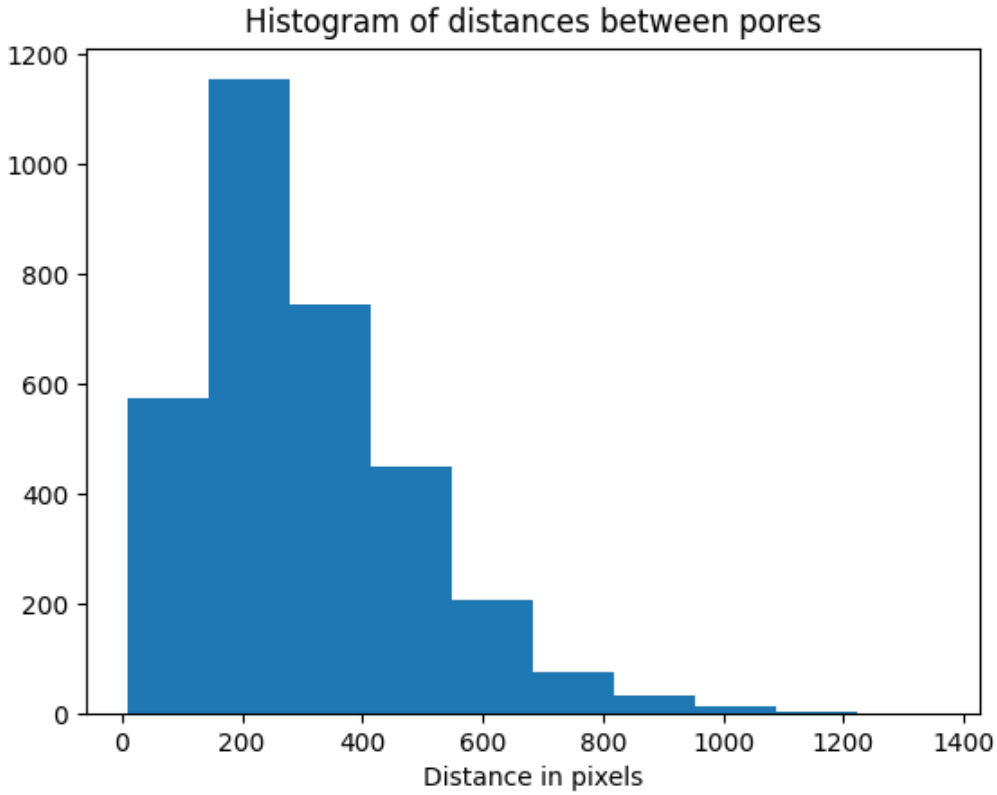


Figure 3.2: Porosity histogram with pore distances from Fig. 3.1f

3.2 Collecting sample patches for training a shallow CNN [4] model

Patches are collected from the annotated regions of the images only. Within the annotated regions, random pixels are selected. Annotations for all pixels that are within a 49×49 pixels box, keeping the initially selected pixel in the middle of the box, are checked. If all these 49×49 pixels belong to same class as per the annotations, then that 49×49 pixels patch and its class is collected as a training sample. Refer to Fig. 3.3 and Fig. 3.4. If all pixels inside the 49×49 box are not assigned a class in the annotation, or if any of them belong to a different class, then that patch is rejected and another random pixel position is checked. While collecting the sample patches, it is ensured that same (or similar) number of patches are collected from each of the annotated training images. Also, a tab is kept on the higher limit of patches

to be collected for a single class, to make sure that the total number of samples for each class is same (or similar). Total $10,000$ training patches are obtained, using this automated patch collection process, out of which $2,500$ samples belong to each of the incipient, inert, circular and lenticular classes.

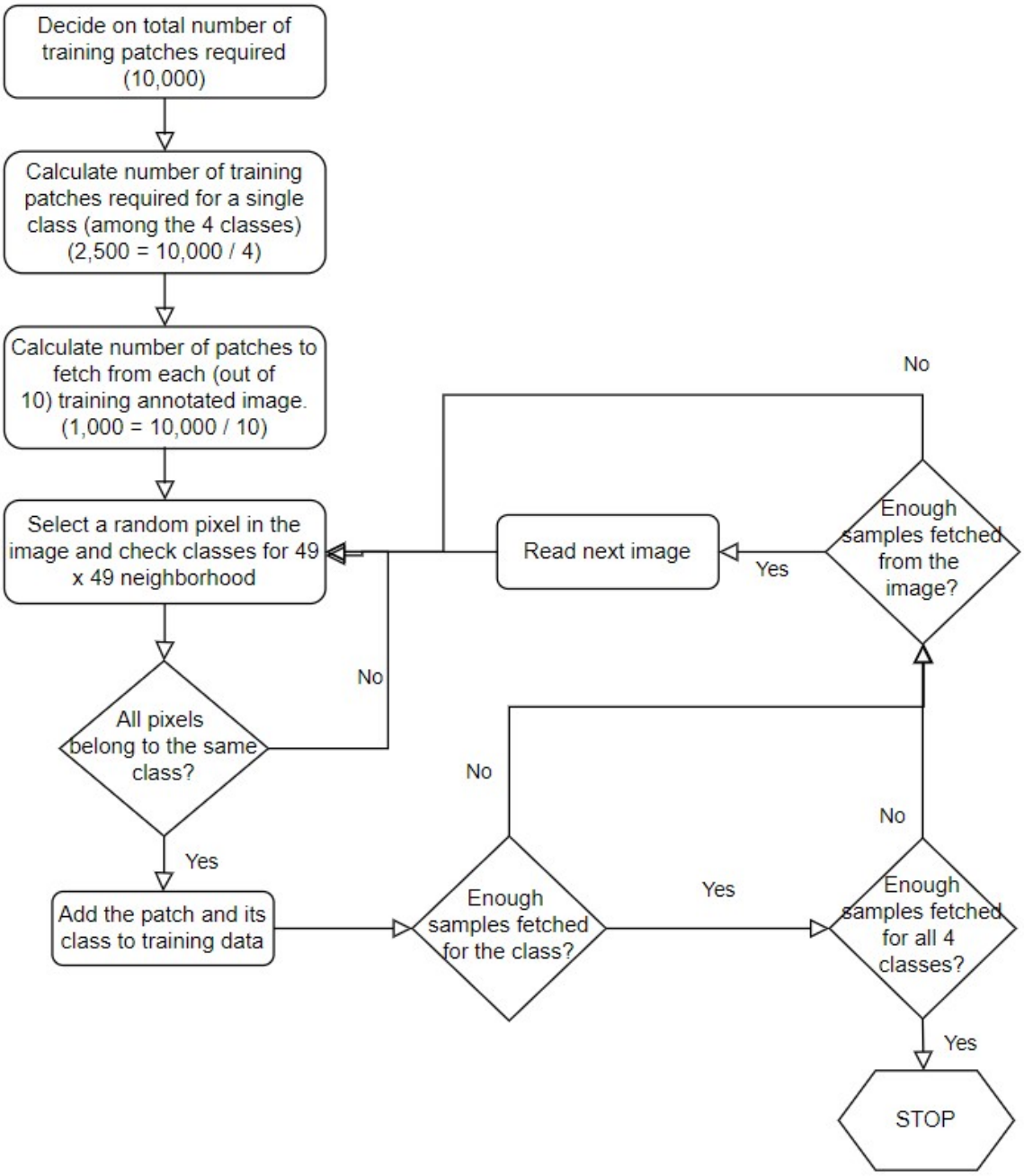


Figure 3.3: Automated patch collection process.

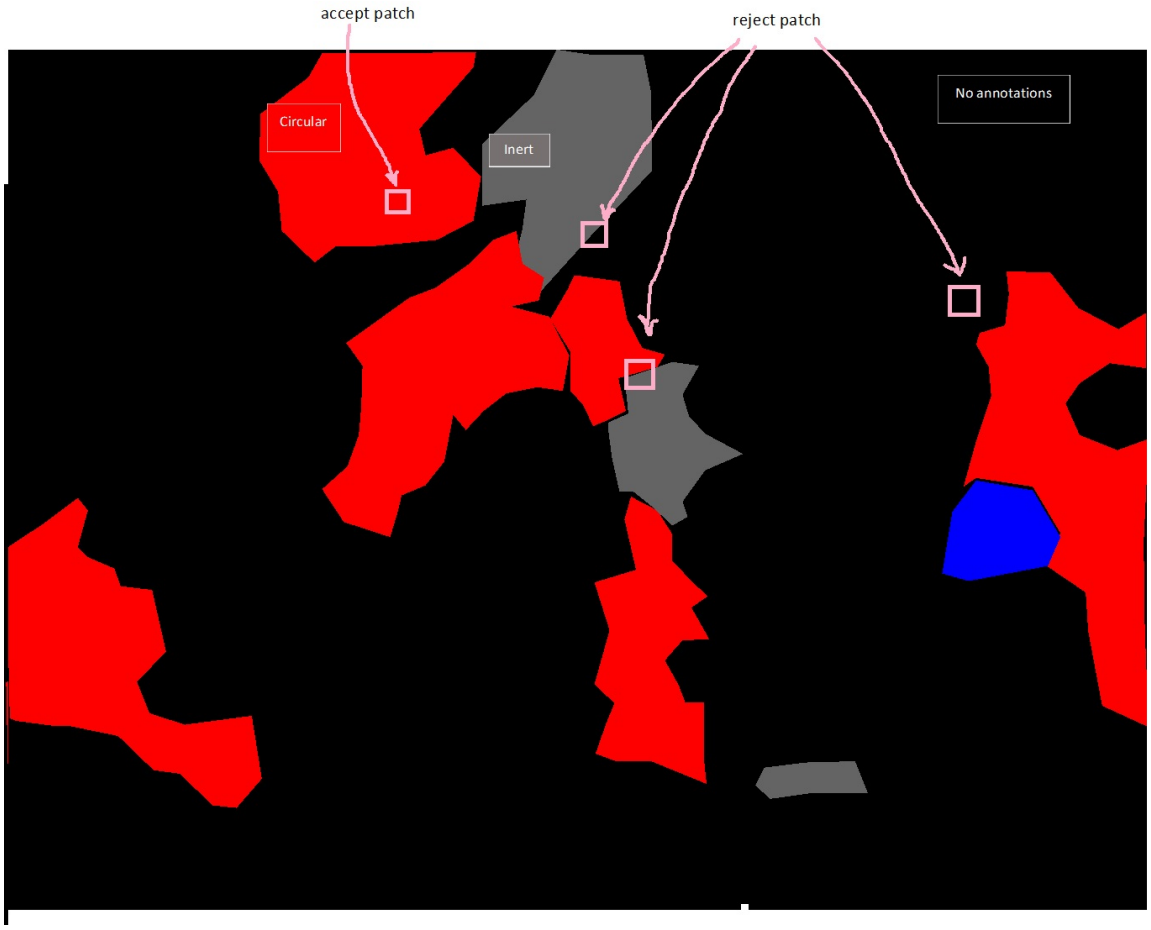


Figure 3.4: A patch is added to the training data, if it belongs to a single class, or else it is rejected.

3.3 Determining class of a sample patch

During the process of patch collection from the annotated coke images (refer to Section 3.2), for training the CNN model (in Fig. 2.1), the target class of the sample patch could be determined in two ways -

1. If all pixels, in the 49×49 neighborhood in the annotated image, belong to the same target class, then take that class.

2. Take the class on which majority of the pixels, in the 49×49 neighborhood in the annotated image, are assigned to.

Using K-fold cross validation, the accuracy of the model (in Fig. 2.1) is obtained using above two approaches of patch collection, and compared in Table 3.1.

K is chosen as 5. The training data of $10,000$ patches and their target classes are randomly split into 5 equal parts. As shown in Fig. 3.5, 5 folds are created sequentially and the CNN models (with different hyper-parameters described in next paragraph) are trained using train data ($8,000$ patches and their target classes) and validated using validation data ($2,000$ patches and their target classes). Fold 0 takes the 1^{st} split as validation data, and $2^{nd} - 5^{th}$ split as training data, Fold 1 takes the 2^{nd} split as validation data, and $1^{st}, 3^{rd} - 5^{th}$ split as training data, and so on.

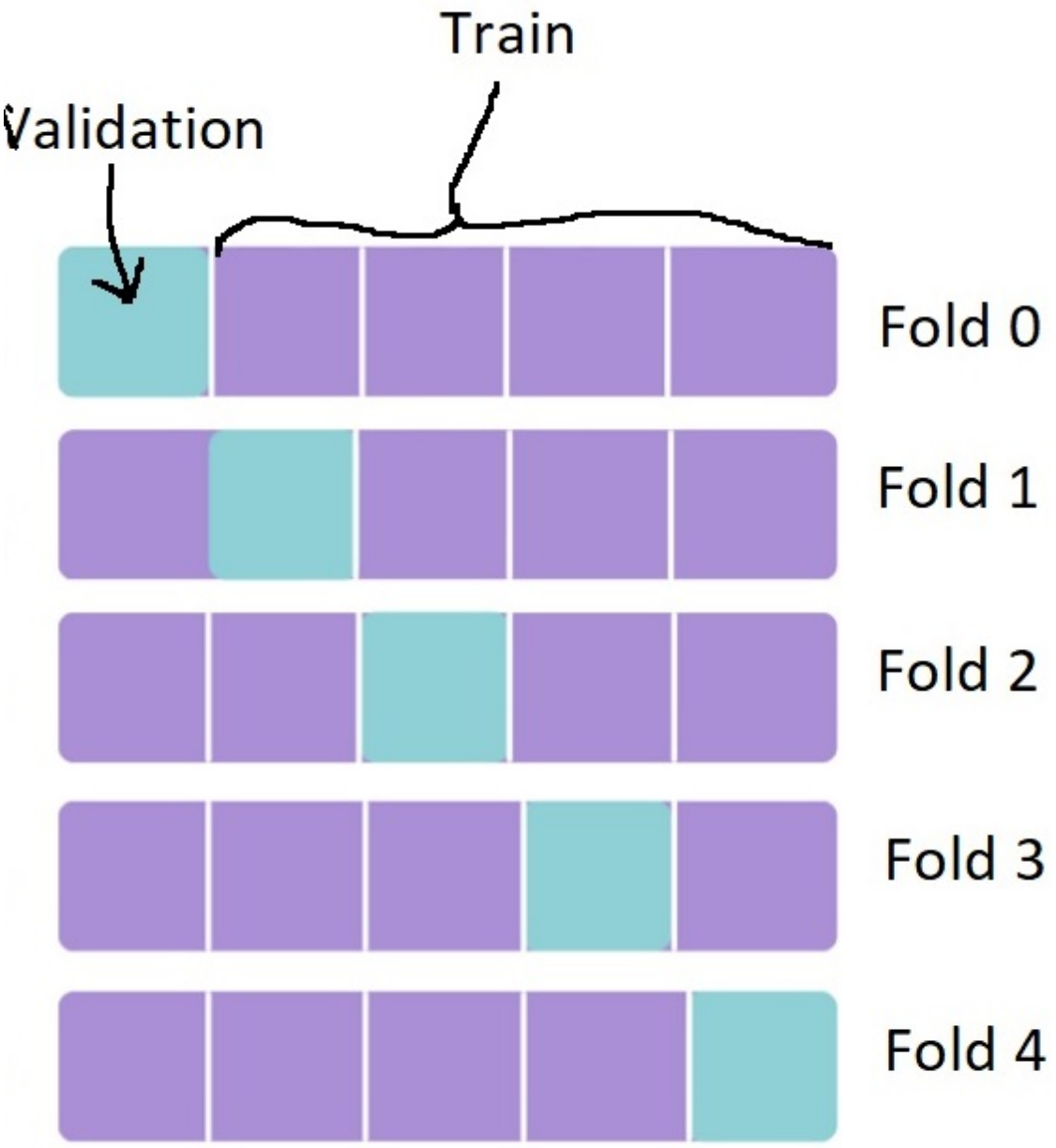


Figure 3.5: 5-Fold cross validation.

The average is calculated as the average of all folds (Fold 0 to Fold 4). The accuracy values obtained by approach 1 above, is denoted as "Same" in the 1^{st} row, and the accuracy values obtained from approach 2 are denoted as "Majority" in the 2^{nd} row.

Accuracy is defined as the number of 'true positive's (as described in Fig. 3.6) for

all four classes, divided by the total number of samples predicted.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 3.6: Confusion matrix.

Based on observed data, it is decided that approach 1 is giving better performance. Therefore all pixels are checked for the training patches, and the patch is selected as a training data, only if all the pixels belong to a same valid class, in the annotated image.

Table 3.1: Accuracy for 5-Fold cross validation using two types of train patches: patches with same label in all pixels; and patches with label same as majority of pixels

	Accuracy					
	Average	**Fold 0**	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**
Same	**0.67**	**0.67**	**0.65**	**0.66**	**0.67**	**0.68**
Majority	0.62	0.62	0.60	0.61	0.64	0.62

3.4 Choice of the activation function used for non linearity in the shallow CNN [4] model

SELU (scaled exponential linear units) activation function is proposed by Klambauer et. al, in [14].

$$selu(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \leq 0 \end{cases} \quad \text{.....(1)}$$

This activation function is created to construct self-normalising neural networks, with pre-defined constants $\alpha = 1.67, \lambda = 1.05$.

Selu activation function has following properties, please refer to Fig. 3.7 for the graph of Selu activation function.

1. Negative and positive values at the output, for controlling the mean to be near zero, hence normalising the output
2. Saturation regions (derivatives approaching zero) to dampen the variance, for negative net inputs, hence diminish the vanishing/exploding gradient effect
3. A slope larger than one to increase the variance for positive inputs
4. A continuous curve ensuring a fixed point, where variance damping is equalized by variance increasing

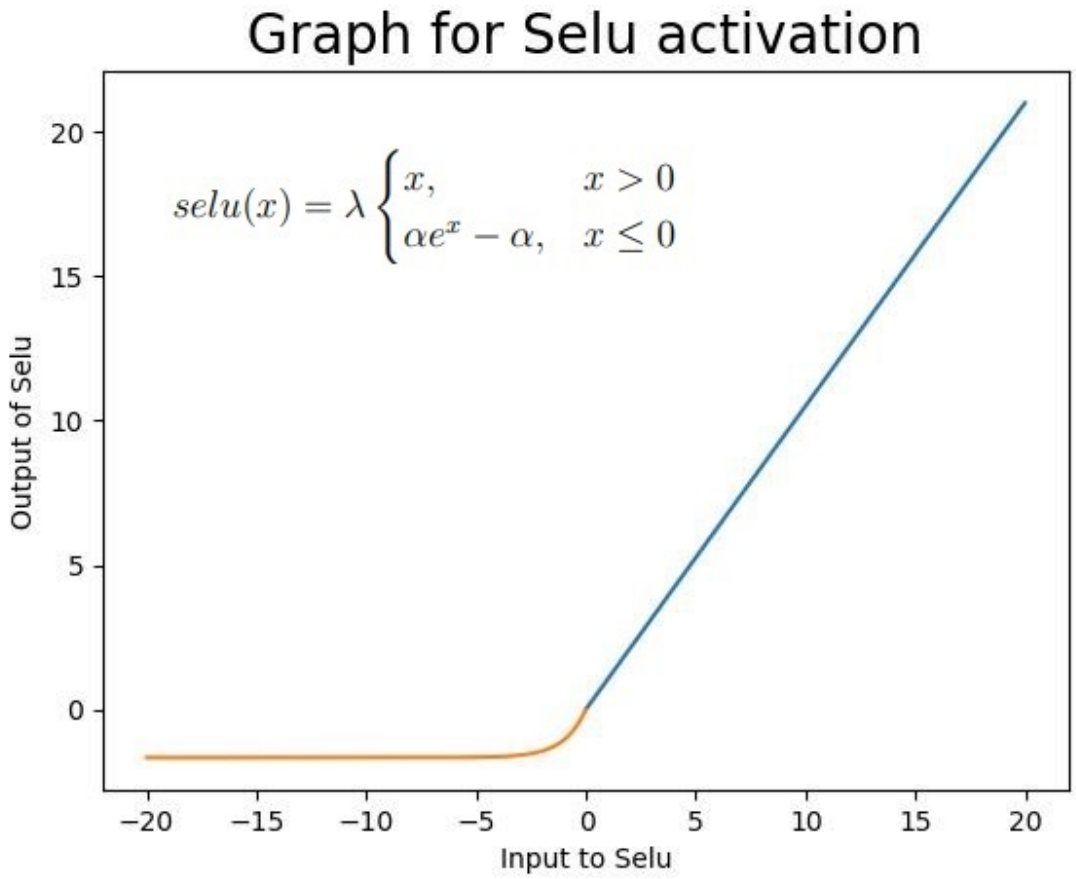


Figure 3.7: Selu activation fuction.

To compare performances of Relu and Selu activation units, the model in Fig. 2.1 has been trained for 5 epochs, using Selu [14] and Relu [20] activation functions. Based on the results documented in Table 3.2, it is decided that the model is learning faster with Selu activation function.

Table 3.2: Losses when a shallow CNN model [4] is trained with Selu and Relu.

Losses					
	epoch 1	epoch 2	epoch 3	epoch 4	epoch 5
Selu	17.56	19.09	18.99	13.46	11.49
Relu	29.23	25.04	22.37	19.91	15.19

3.5 Choice of model hyper-parameters

3.5.1 Choice of input patch size and number of intermittent channels

The optimum size of patches for the train samples, and the number of channels in between the two CNN layers in a shallow CNN model, are decided by K-fold cross validation, as described in Section 3.3.

Fig. 3.8 is the same diagram of the CNN model in Fig. 2.1, with arrows pointing at the parameters being decided by this K-fold cross validation.

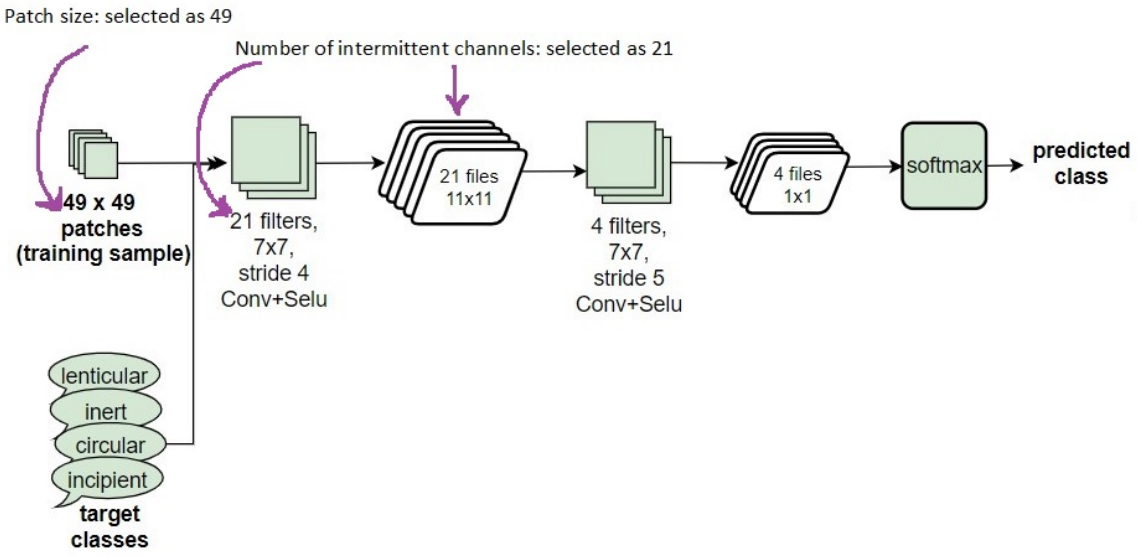


Figure 3.8: The Baseline CNN model parameters : input patch size and number of channels in between two convolution layers.

To estimate the best performing hyper-parameters, first the input patch size is kept at 49, number of channels of the intermittent file in between two convolution layers of the model (in Fig. 3.8) are varied, and the accuracy (defined in Section 3.3) is checked for each fold (Fold 0 to Fold 4) to find out an optimum value. The hyper-parameters of the model, for which accuracy is maximum and consistent among each folds, are chosen. It is observed that, increasing intermittent channels beyond 12 does not result in much improvement in the accuracy. Please refer to Table 3.3 for the accuracy of each fold for different hyper-parameters set. Then keeping the number of channels in intermittent files as 12, patch size of the training samples are varied. For square patches of side above 49 pixels, observed accuracy across different folds are not uniform and accuracy is decreasing for patch size below 49. Variance among accuracy across folds, for patch sizes 61 and 11, are large (7.691 and 6.419 respectively), whereas variance of the accuracy highlighted in green colour is only 1.353. Based on this experimental data, patch size is chosen as 49 and the number of intermittent channels between two convolution layers, is kept as 21.

Table 3.3: List of accuracy obtained by a shallow CNN model as in Fig. 3.8, with varying model parameters (patch size, number of internal channels between 2 CNN layers), for 5 different folds of the 5-Fold cross validation

Patch size	**Int. chnls**	**Accuracy**						**Comments**
		Average (%)	**Fold 0 (%)**	**Fold 1 (%)**	**Fold 2 (%)**	**Fold 3 (%)**	**Fold 4 (%)**	
49	3	25.44	23.25	27.56	25.87	24.80	25.72	
49	7	30.36	36.23	23.42	24.69	32.92	34.56	Increasing
49	12	39.17	36.64	31.49	42.13	41.41	44.17	Increasing
49	**21**	**41.38**	**40.27**	**39.78**	**43.97**	**40.95**	**41.92**	
61	12	40.26	43.89	23.54	44.44	46.29	43.15	Unstable
35	12	38.45	37.24	44.81	42.75	35.61	31.86	Reduced
11	12	33.56	25.00	24.91	39.47	38.79	39.63	Unstable

3.5.2 Choice of optimizer and learning rate

Machine learning involves using an algorithm to learn and generalize from test data, in order to make predictions on new test data. The machine learning algorithm defines a parameterized mapping function and an optimization algorithm is used to find the values of the parameters (model weights) that minimize the error (loss) of the function when used to map inputs to outputs. Every time the machine learning algorithm is fit on a training data set, it solves an optimization problem.

The learning rate controls how quickly the model is learning. Smaller learning rates require more training epochs given the smaller changes made to the weights each update, whereas larger learning rates result in rapid changes and require fewer training epochs.

Different optimizers (Stochastic Gradient descent (SGD) [21], Adam [22], RProp [23], SGD with momentum [24] etc.) and learning rates are used to train the model in Fig. 2.1. It is observed that maximum train accuracy is obtained using an Adam optimiser [22] with learning rate of 0.0001.

3.5.3 Loss function

Neural Networks are trained using optimization process (refer to Section 3.5.2) that tries to reduce the loss. This loss depicts the difference of the predicted output from the expected (target output). Since the output predictions from the model are probability distributions for the four output classes, cross entropy loss is used

that measures the loss in terms of the entropy between the predicted and target distributions.

3.5.3.1 Cross Entropy Loss (CE)

The existing cross entropy loss minimises the distance between the predicted class distribution with its target distribution. Refer to Fig. 3.9. The loss function minimises the distance between the probability distributions of the predicted data and corresponding distribution of one-hot encoding of the target class. Cross entropy is calculated as

$$CE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} t_{ik} \ln y_{ik} \quad \text{.....(2)}$$

where where y_{ik} is the soft assignment (prediction percentage probability) of the i^{th} patch of the batch to the k^{th} class; and t_{ik} is the k^{th} component of the one-hot encoding distribution of the target class of the i^{th} patch, i.e. $t_{ik} = 1$, if the i^{th} patch belongs to k^{th} class, otherwise it is 0.

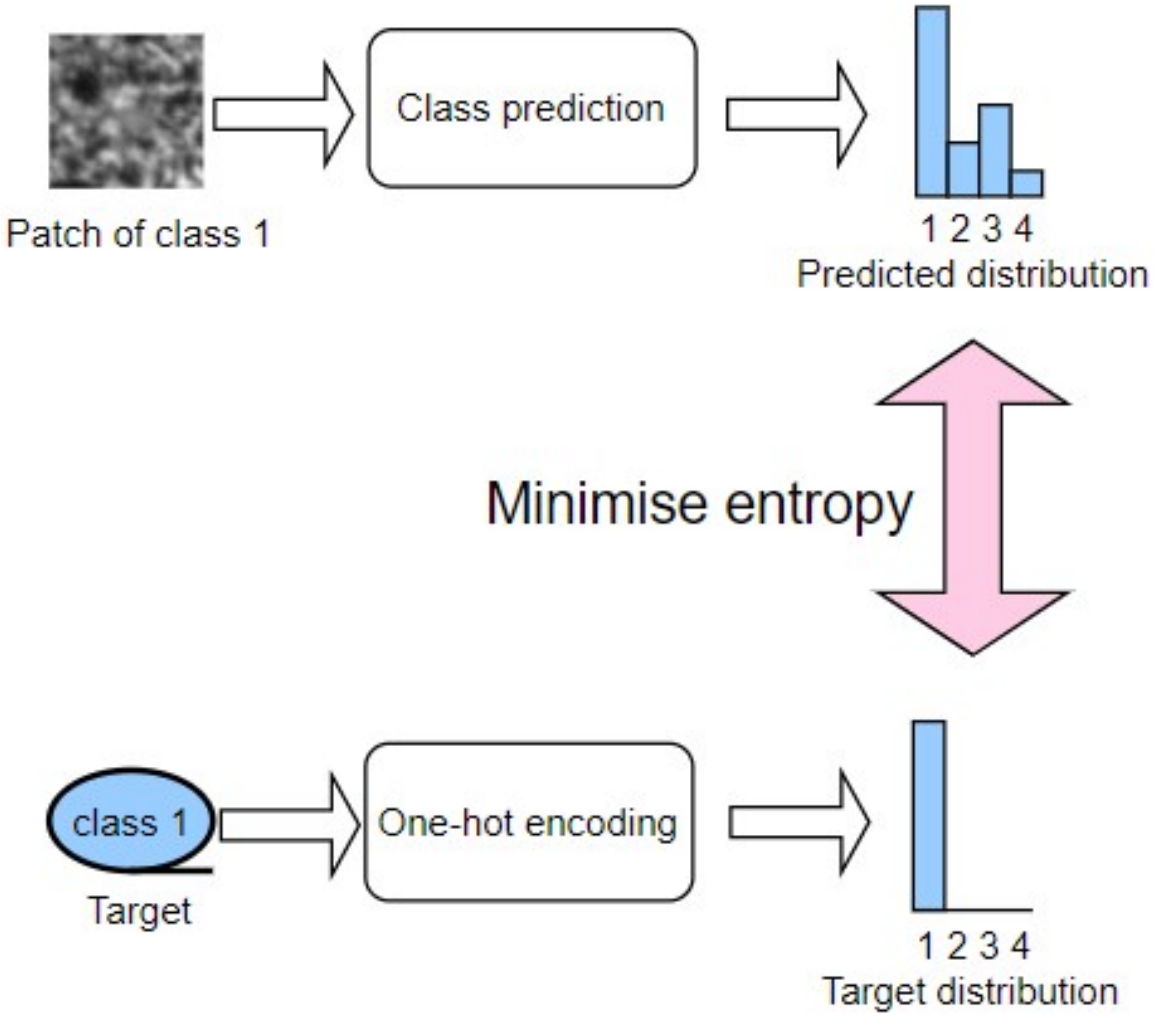


Figure 3.9: Cross entropy loss minimises entropy between predicted and target distributions.

3.5.3.2 Custom Cross Entropy Loss

As less annotations were available on the coke images, three new losses are defined, which make use of contrastive learning, by maximising the distance between distributions of class predictions of patches belonging to different classes, in addition to minimising the distance between distribution of prediction of a patch and its target class. Refer to Fig. 3.10. The concept of 'minimizing intra-class spread and maximizing inter-class distances' is already used in Fisher Rao discrimination [25] since the first half of 20^{th} century, but it is used in deep learning, recently. These losses are

described below.

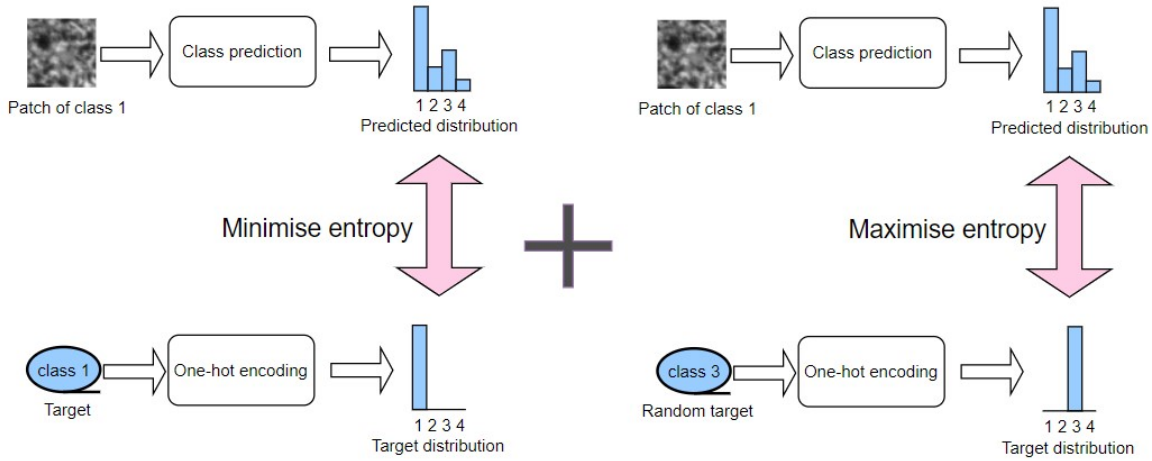


Figure 3.10: Contrastive cross entropy loss maximises entropy between distributions of prediction and a target belonging to a different class; in addition to minimising entropy between predicted and target distributions.

Custom cross entropy - subtraction (CCE-Sub) The custom cross entropy subtraction loss is defined as

$$CCE - Sub = -\frac{1}{n}\sum_{i=1}^{n}(\sum_{k=1}^{c} t_{ik} \ln y_{ik} - \sum_{i \neq j, l=1}^{c} t_{jl} \ln y_{il}) \quad \text{.....(3)}$$

where j is a random patch present in the batch (hopefully) belonging to a different class than the i^{th} sample in the batch. t_{jl} are the one-hot encoding of the j^{th} target to l^{th} class i.e. $t_{jl} = 1$, if patch j belongs to the l^{th} class, otherwise 0.

Custom cross entropy - division (CCE-Div) The custom cross entropy division loss is defined as

$$CCE - Div = \frac{1}{n}\sum_{i=1}^{n}(\frac{\sum_{k=1}^{c} t_{ik} \ln y_{ik}}{\sum_{i \neq j, l=1}^{c} t_{jl} \ln y_{il}}) \quad \text{.....(4)}$$

where j is a random patch present in the batch (hopefully) belonging to a different class than the i^{th} sample in the batch. t_{jl} are the one-hot encoding of the j^{th} target to l^{th} class i.e. $t_{jl} = 1$, if patch j belongs to the l^{th} class, otherwise 0.

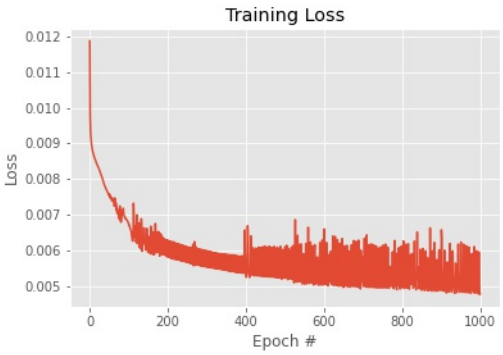
Custom cross entropy - power (CCE-Pow) The custom cross entropy power loss is defined as

$$CCE - Pow = \frac{1}{n}\sum_{i=1}^{n}(-\sum_{k=1}^{c} t_{ik} \ln y_{ik})^{\sum_{i\neq j, l=1}^{c} t_{jl} \ln y_{il})} \quad \text{.....(5)}$$

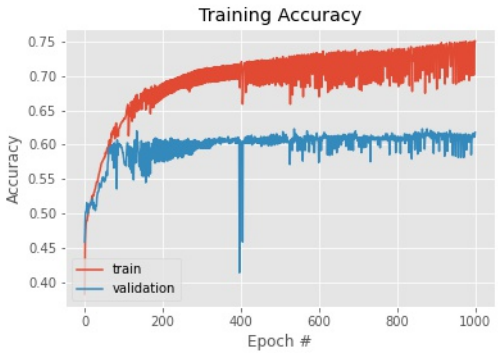
where j is a random patch present in the batch (hopefully) belonging to a different class than the i^{th} sample in the batch. t_{jl} are the one-hot encoding of the j^{th} target to l^{th} class i.e. $t_{jl} = 1$, if patch j belongs to the l^{th} class, otherwise 0.

3.5.4 Deciding optimum number of epochs for training

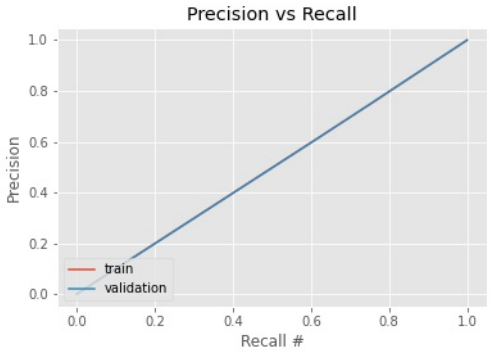
The training procedure of the CNN model (Fig. 2.1) is explained in detail in Section 3.6. The plot of the train and validation accuracy are shown in Fig. 3.11b. This plot illustrates that the accuracy for validation data is saturated after 150 epochs, after which validation accuracy is not increasing along with rising train accuracy.



(a) Train loss



(b) Accuracy



(c) Precision vs Recall

```
------------------------------

1000 epochs executed.

------------------------------

Train loss: 0.0047632201421487184

Train accuracy: 75%

Train precision: 74%

Train recall: 75%


Validation accuracy: 62%

Validation precision: 61%

Validation recall: 62%
```

(d) Execution log

Figure 3.11: CNN model (Fig. 2.1) performance with 1000 epochs of training.

3.6 Training procedure

The training procedure is shown in Fig. 3.12. Total 15 partially annotated coke images were available. Out of them, 10 images were used for training and validation, and rest 5 were kept aside, as data unseen by the model, for testing. $10,000$ patches of size 49×49 pixels were collected from the 10 training annotated images, using the process described in Section 3.2. A 5-Fold cross validation is performed using this training data (with $10,000$ patches and their targets) to determine hyper-parameters so that the model (as shown in Fig. 2.1 and using cross entropy loss as in equation (2)) is robust and stable (refer to Section 3.3 and Section 3.5.2). After that, the highest

performing model from this 5-Fold cross validation (the model obtained from Fold 3 with accuracy 43.97% in Table 3.1) is taken, and is trained further with almost full training data, to increase performance.

The training data (with $10,000$ patches and their targets) is randomly divided into $9:1$ ratio, and used as training and validation data respectively. It is validated that both training and validation data contains data from all four classes.

With above $9:1$ training and validation data, first the CNN model (in Fig. 2.1) is trained for 1000 epochs using cross entropy loss, as in equation (2). Then again the best performing model (obtained from Fold 3 of 5-Fold with accuracy 43.97% in Table 3.1) is taken and trained with above $9:1$ training and validation data, for 1000 epochs using custom cross entropy subtraction (CCE-Sub) loss, as in equation (3). Then similar training is performed using the other two loss functions (CCE-Div and CCE-Pow), as shown in equations (4) and (5). The accuracy, precision and recall are provided in Chapter 4.

Accuracy is defined in Section 3.3. Precision is defined as number of true positive / number of predicted positive, i.e. (true positive + false positive). And recall is defined as number of true positive / number of actual positive, i.e. (true positive + false negative). Please refer to Fig. 3.6 for definition of 'true positive', 'false positive', 'false negative' etc.

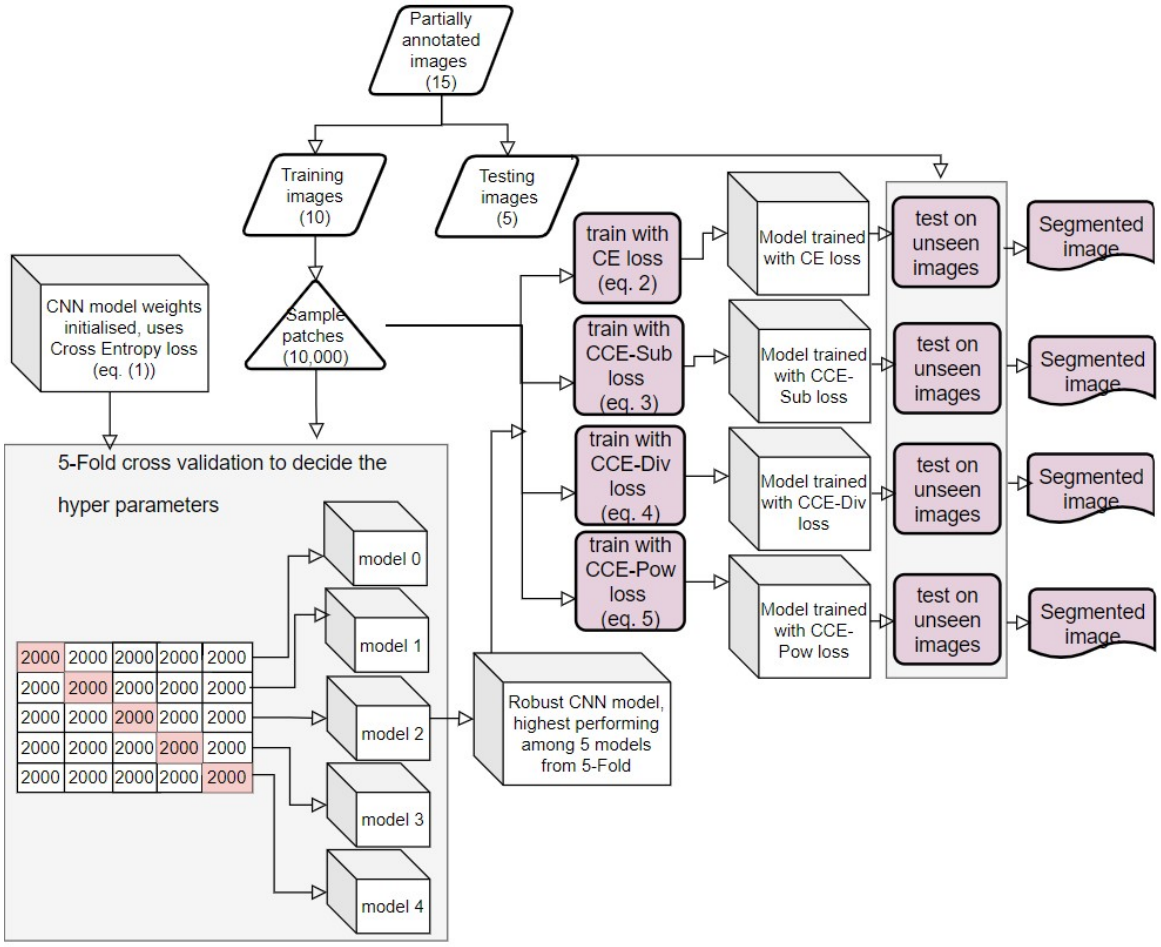
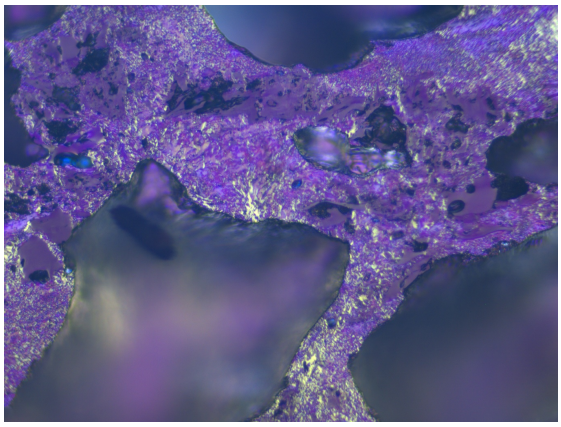


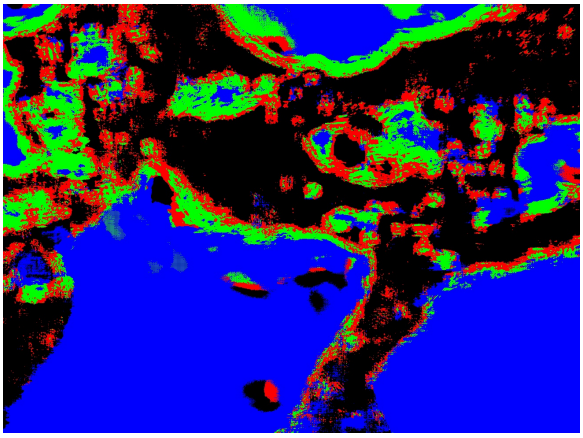
Figure 3.12: The training process.

Fig. 4.6 and Fig. 4.7 display correct identification of incipients, which are majority in that specimen images. However the pore identification process failed to identify some portions of the pores, due to high intensity regions touching the border of the image (since a contour inside a contour, is not considered inside it, if it is touching the image border). And those pore regions are passed to the CNN (process described in Fig. 2.2), and classified as inerts wrongly.



(a) Original

(b) Pores identified



(c) Segmented with CCE-Sub loss

Figure 4.6: (a) original image, (b) binary image with pores identified in black, (c) segmentation by CCE-Sub. Some of the pore regions are not marked as pore by the pore identification process, due to high intensity regions touching the border. These regions are wrongly identified as inerts by the CNN model, due to their uniform texture. Some inert regions are marked as pore, by the pore identification process, which could have been classified as inert by the CNN otherwise. This coke image contains maximum of the non-pore regions as incipient, which are correctly classified by the CNN model.

Chapter 4

Results

4.1 Performance metrics

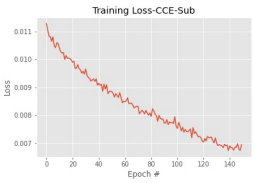
When the shallow CNN model (described in Fig. 2.1) is trained with same data (using procedure explained in Section 3.6, and in Fig. 3.12) and same number of epochs, but different loss functions as discussed in Section 3.5.3), it is observed that the model trained using custom cross entropy loss (defined in equations (3)) is learning faster than the model trained using cross entropy loss (CE) (as in equation (2)). The precision, recall and accuracy values obtained after training the models for 150 epochs are reported in Table 4.1 and Fig. 4.1.

Performance of these models on test data, are shown in Table 4.2 and Fig. 4.2.

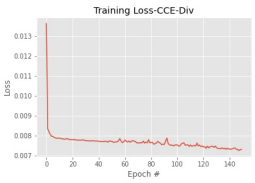
Table 4.1: Precision, recall and accuracy values obtained after training the models for 150 epochs.

Training	Overall			Circular		Incipient		Inert		Lenticular	
150 epoch	Accuracy (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)
CE	54	53	54	48	45	61	**85**	56	44	47	41
CCE-Sub	**57**	**56**	**57**	**53**	**47**	**65**	84	58	**54**	**49**	**44**
CCE-Div	43	45	43	35	36	44	58	**62**	39	40	40
CCE-Pow	38	39	38	29	43	52	58	39	24	35	26

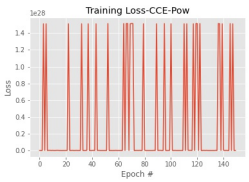
(a) CE loss



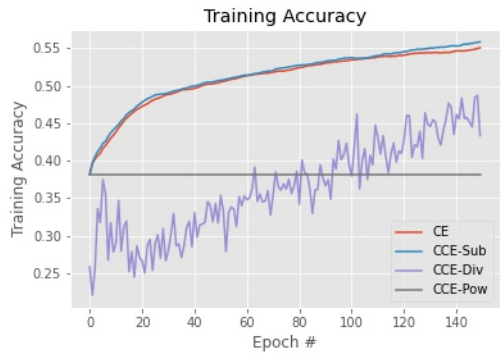
(b) CCE-Sub loss



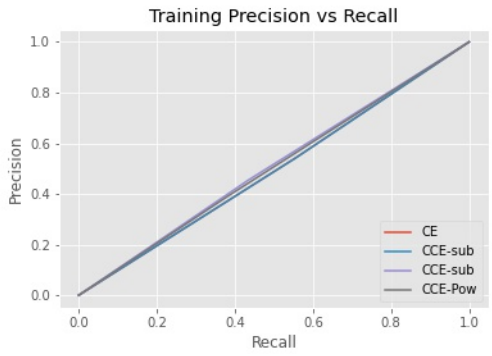
(c) CCE-Div loss



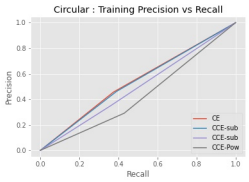
(d) CCE-Pow loss



(e) Accuracy



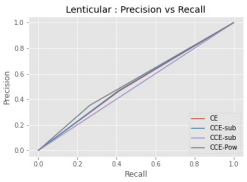
(f) Precision vs. recall



(g) Circular

(h) Incipient

(i) Inert

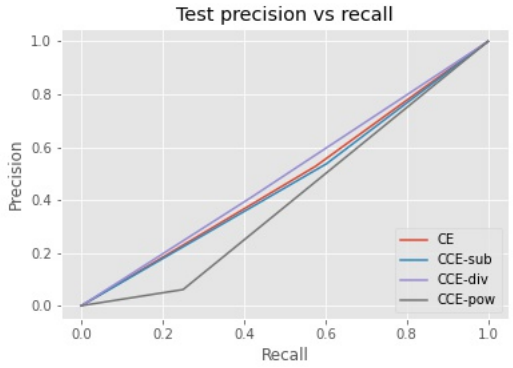


(j) Lenticular

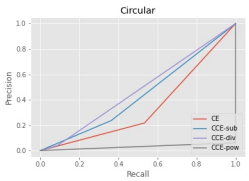
Figure 4.1: Loss, accuracy, precision, recall on train data after training the models for 150 epochs.

Table 4.2: Performance of the models on test data.

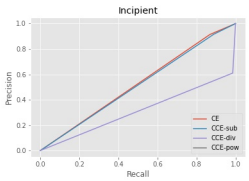
Test	Overall			Circular		Incipient		Inert		Lenticular	
	Accrcy (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)	Prec. (%)	Rec. (%)
CE	68.2	53	58	22	**53**	91	87	97	63	01	27
CCE-Sub	**71.5**	**54**	**60**	**24**	36	**91.5**	**89**	**98**	**69**	01	**47**
CCE-Div	63.6	41	42	4	9	61	99	98	57	0.1	2
CCE-Pow	6	6	25	6	1	0	0	0	0	0	0



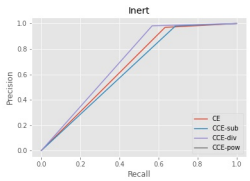
(a) Precision vs. recall



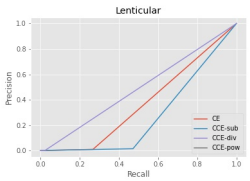
(b) Circular



(c) Incipient



(d) Inert



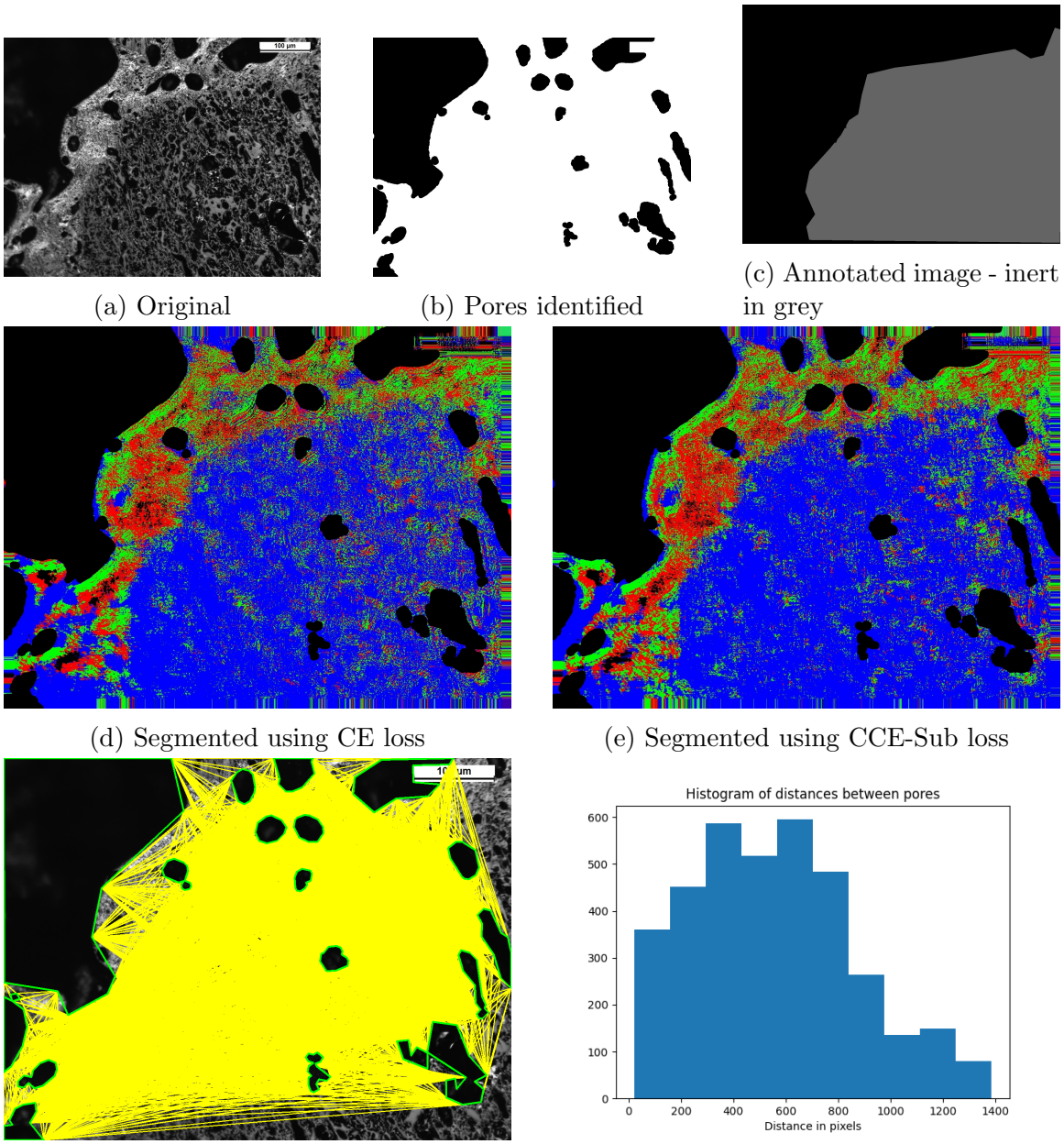
(e) Lenticular

Figure 4.2: Precision and recalls obtained on test data using the models trained with different losses.

4.2 Qualitative results

Porosity calculation results and semantic segmentation of unseen images are shown in this section.

In Fig. 4.3, the segmented image produced from the model trained using CCE-Sub loss function (described in equation (3)), is more solid and less grainy than the segmentation created out of the model trained using CE loss (shown in equation (2)). The inert regions are identified accurately in blue color, and rest of the regions are classified as pore (black), circular (red) or lenticular (green), based on the texture. The pore distances and their histogram is also shown.



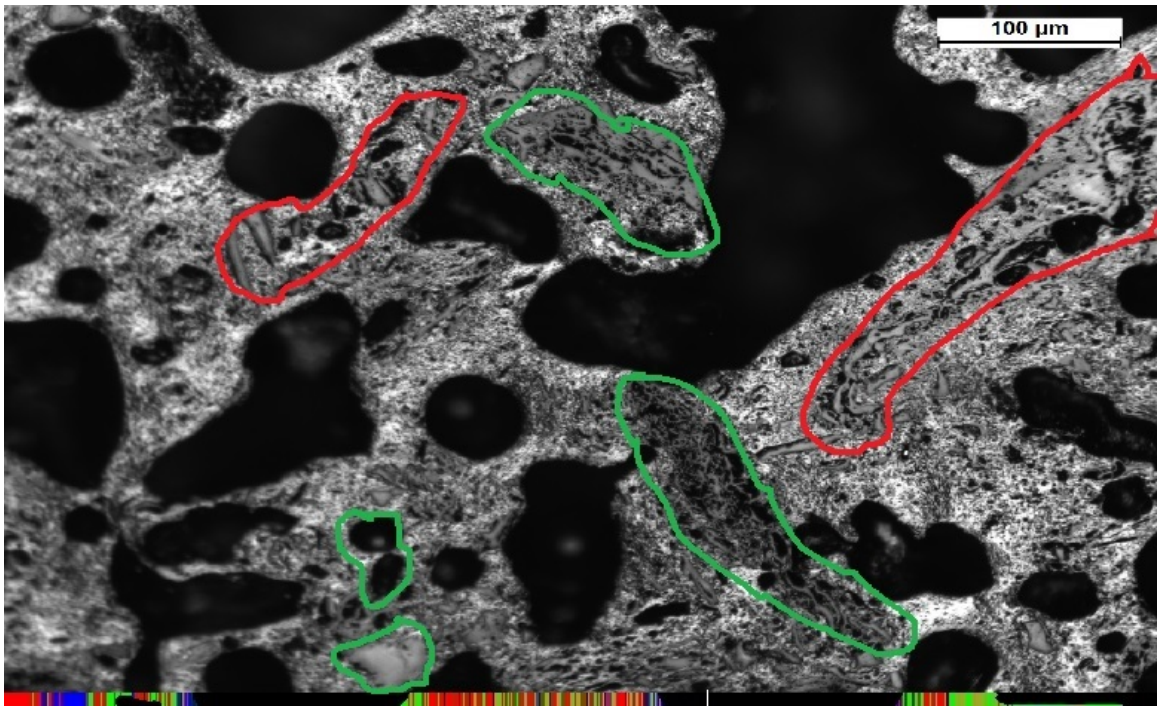
(a) Original (b) Pores identified (c) Annotated image - inert in grey

(d) Segmented using CE loss (e) Segmented using CCE-Sub loss

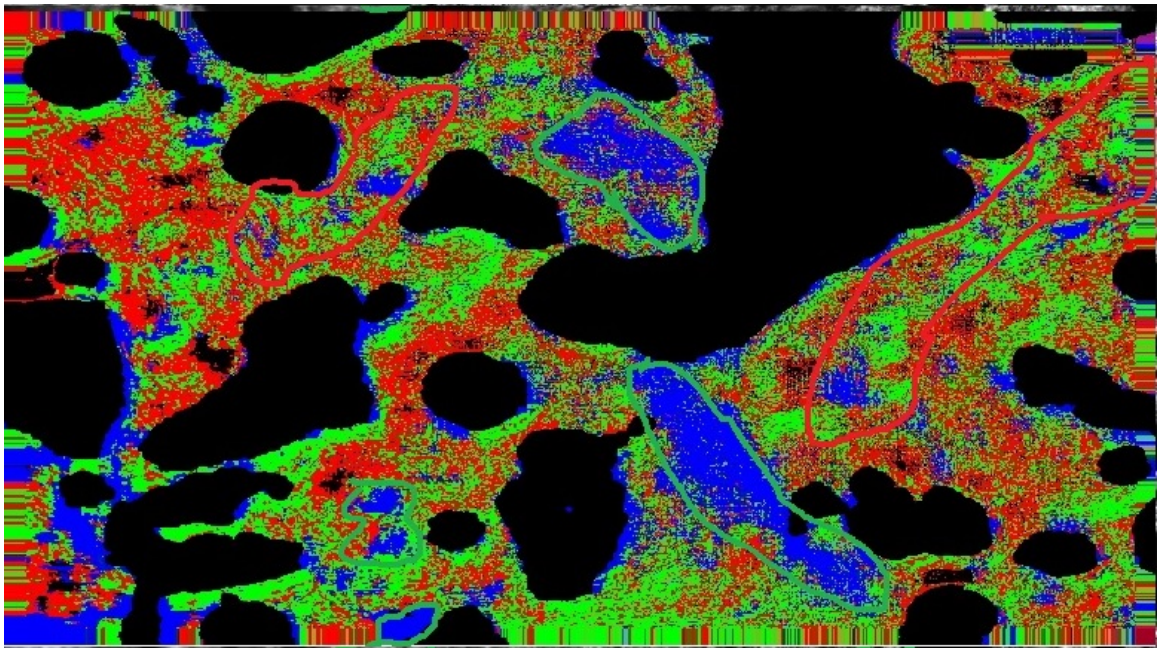
(f) Pore distances (g) Porosity histogram

Figure 4.3: (a) Original image, (b) binary image showing pores in black color and non-pores in white, (c) manual annotations (ground truth), Color code: grey (inert), red (circular), green (lenticular)(d) the segmented image using the CNN model, with Cross Entropy loss. Color code: black (pores and incipients), blue (inert), red (circular), green (lenticular), (e) the segmented image using the CNN model, with the Contrastive Cross Entropy loss (Sub), (f) polygonal approximation of pores in green lines, and the distances between pores in yellow, (g) porosity histogram with pore distances calculated from (f).

Fig. 4.4 shows some accurate and some incomplete identification of inert regions in semantic segmentation created by the model trained using CCE-Sub loss (equation (3)). The failure cases are the inerts having comparatively narrow structure than other inerts. This type of narrow inert structures were not present in the annotations used for training. So the model was unable to identify them as inerts.



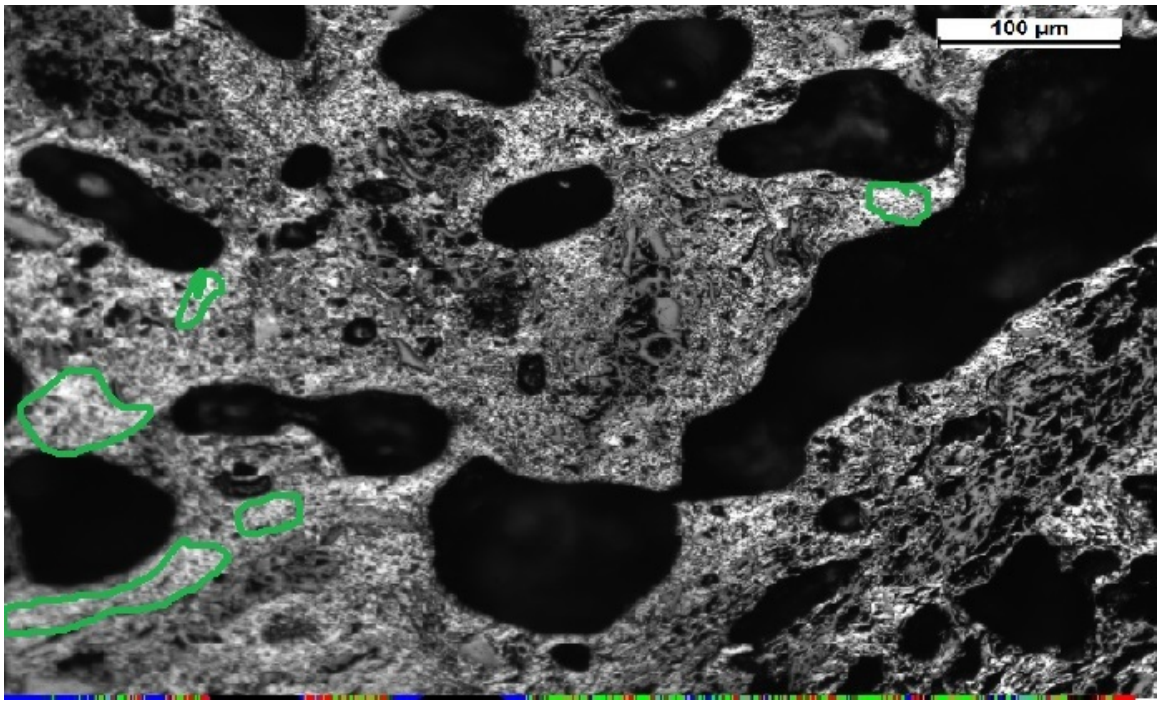
(a) Original



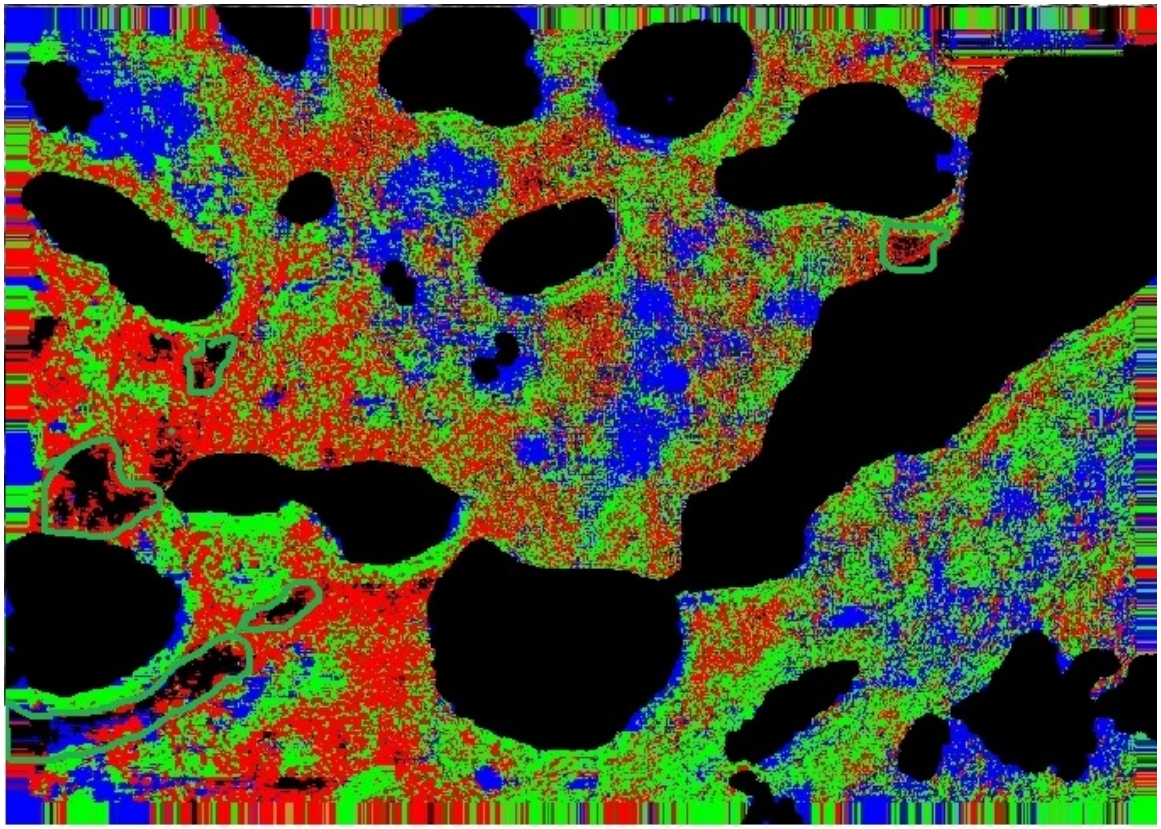
(b) Segmented with CCE-Sub loss

Figure 4.4: (a) original image, (b) segmented image by CCE-Sub. The inert regions, marked with green border, are classified accurately. But the ones marked in red border, are the failure cases, where inert regions are not identified fully as inert, but as a grainy segmentation, consisting of inert, circular and lenticular regions.

In Fig. 4.5, almost all regions are classified properly, except the fact that the whole segmentation is grainy. Some incipient regions, correctly identified, are marked in green border.

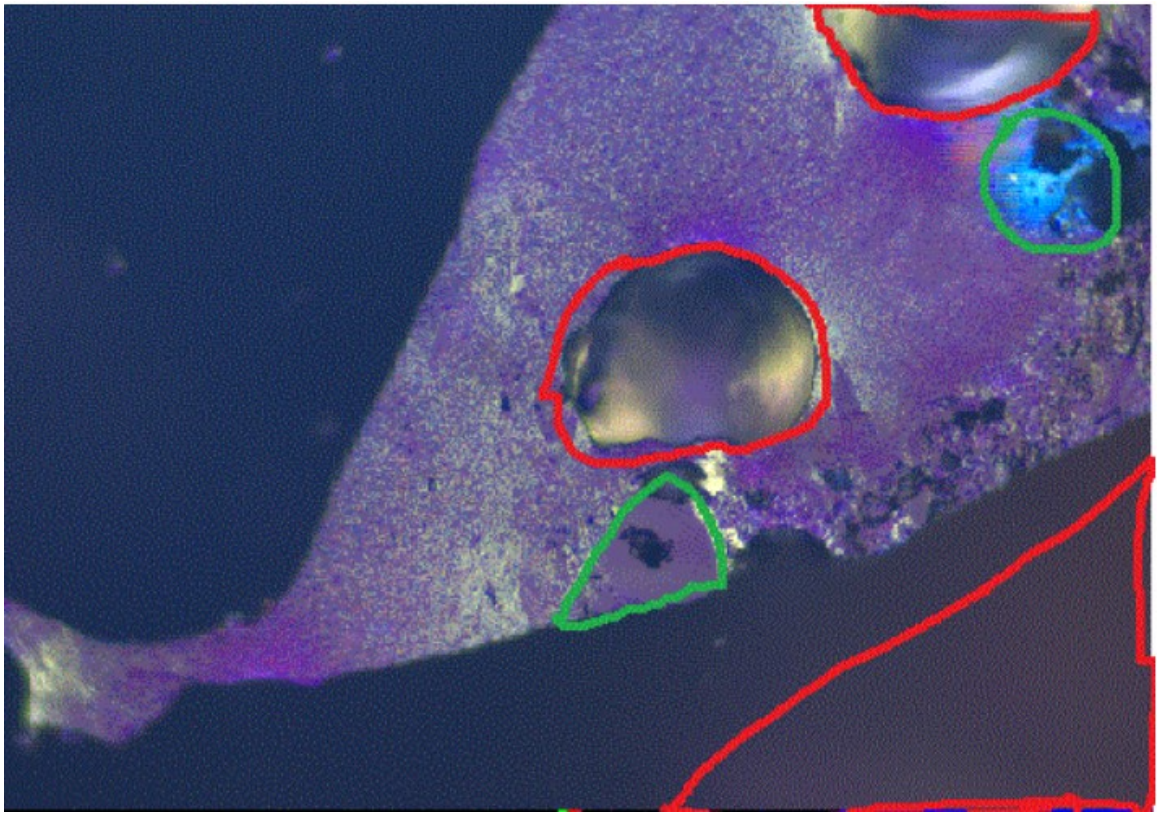


(a) Original

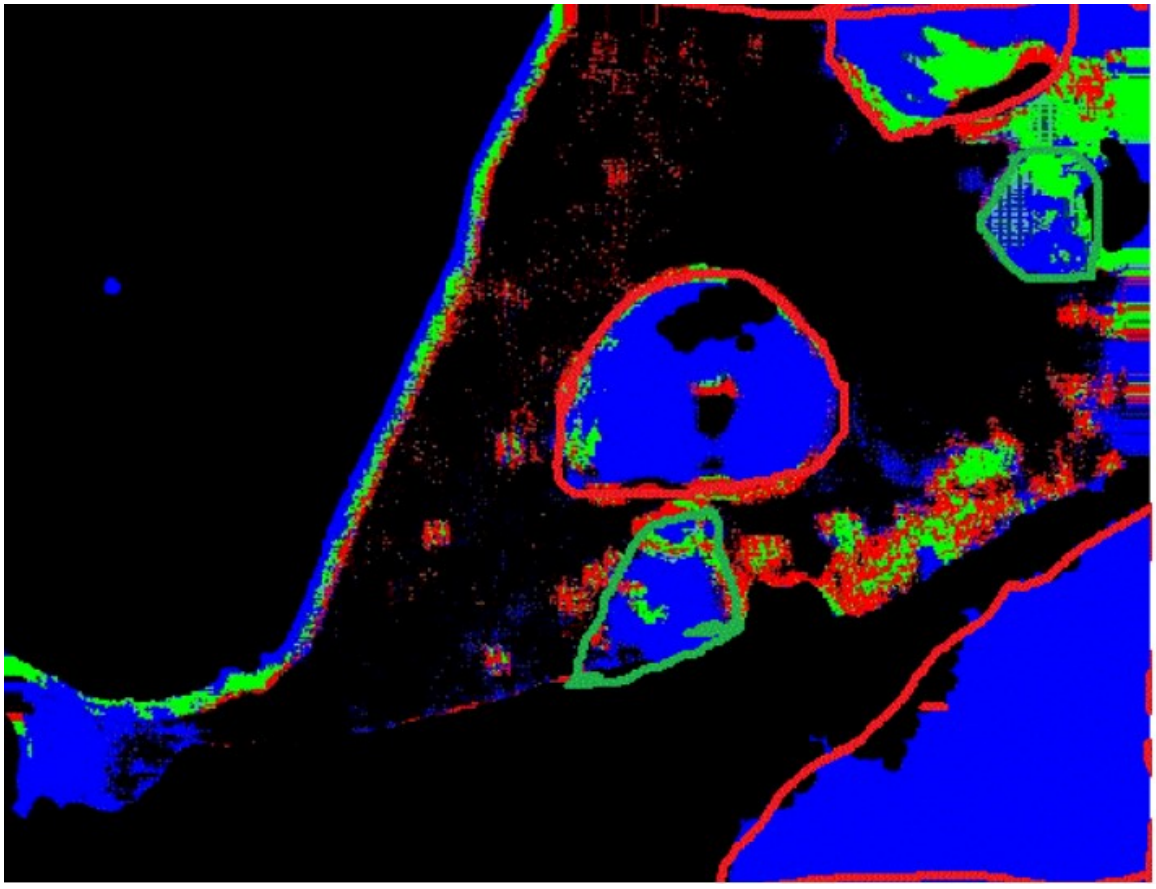


(b) Segmented with CCE-Sub loss

Figure 4.5: (a) original, (b) segmented by CCE-Sub. Incipient regions are correctly identified (marked with green border). Other regions are properly segmented too, except that the whole segmented image is grainy



(a) Original



(b) Segmented with CCE-Sub loss

Chapter 5

Conclusions

As the number of training samples were increased by collecting small patches, the CNN model trained using CCE-Sub loss (described in equation (3)) had learnt the class representations reasonably well. Using the contrastive approach in the loss function has helped the model learn representations well, provided the scarcity of annotated data. However the segmented images were still grainy. That is because the test precision for the circular and lenticular classes could not be increased to more than 24%. With a more accurate model, the grains could be reduced. Also, classification was done for only four broad classes, but there were other classes, like ribbon, isotropic; moreover each of the circular, lenticular and ribbon classes could be sub-divided into fine, medium and coarse structures. Annotations for these finer divisions and ribbon structures were not available. So these classes could not be not used for segmentation of the images.

Bibliography

[1] N. Choudhury, D. Mohanty, P. Boral, S. Kumar, and S. K. Hazra, "Microscopic evaluation of coal and coke for metallurgical usage," *Current Science*, vol. 94, no. 1, pp. 74–81, 2008.

[2] Southern Illinois University. "Crelling's petrographic atlas of coals and carbons." (), [Online]. Available: `https://coalandcarbonatlas.siu.edu/` (visited on 07/21/2022).

[3] R. J.Gray, "Some petrographic applications to coal, coke and carbons," *Organic Geochemistry*, vol. 17, no. 4, pp. 535–555, 1991.

[4] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *AT'I&'T Bell Laboratories*, 1989. [Online]. Available: `http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf` (visited on 07/21/2022).

[5] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[6] A. A. Agra, A. Nicolodi, B. D. Flores, *et al.*, "Automated procedure for coke microstructural characterization in imagej software aiming industrial application," *Fuel*, vol. 304, pp. 121–374, 2021.

[7] E. Donskoi, A. Poliakov, M. R. Mahoney, and O. Scholes, "Novel optical image analysis coke characterisation and its application to study of the relationships between coke Structure, coke strength and parent coal composition," *Fuel*, vol. 208, pp. 281–295, 2017.

[8] F. Meng, S. Gupta, D. French, P. Koshy, C. Sorrell, and Y. Shen, "Characterization of microstructure and strength of coke particles and their dependence on coal properties," *Powder Technology*, vol. 320, pp. 249–256, 2017.

[9] Y. Zheng, J. Wu, Y. Qin, F. Zhang, and L. Cui, "Zero-Shot Instance Segmentation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. DOI: `10.1109/CVPR46437.2021.00262`.

[10] L. Bo, Q. Dong, and Z. Hu, "Hardness Sampling for Self-Training Based Transductive Zero-Shot Learning," *Conference: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, DOI: 10.1109/CVPR46437.2021.01623.

[11] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive Embedding for Generalized Zero-Shot Learning," *Conference: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, DOI: 10.1109/CVPR46437.2021.00240.

[12] M. F. Naeem, Y. Xian, F. Tombari, and Z. Akata, "Learning Graph Embeddings for Compositional Zero-shot Learning," *Conference: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, DOI: 10.1109/CVPR46437.2021.00101.

[13] A. Ben-Cohen, N. Zamir, E. B. Baruch, I. Friedman, and L. Z. Manor, "Semantic Diversity Learning for Zero-Shot Multi-label Classification," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, DOI: 10.1109/ICCV48922.2021.00068.

[14] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 972–981, 2017. DOI: https://dl.acm.org/doi/abs/10.5555/3294771.3294864.

[15] J. s. Bridle, "Training Stochastic Model Recognition Algorithms as Networks can lead to Maximum Mutual Information Estimation of Parameters," *Advances in Neural Information Processing Systems 2 (NIPS 1989)*, 1989. DOI: https://proceedings.neurips.cc/paper/1989/hash/0336dcbab05b9d5ad24f4333c7658a0e-Abstract.html. (visited on 07/21/2022).

[16] Y. C. Hum, K. W. Lai, M. Salim, and M. Irna, "Multiobjectives bihistogram equalization for image contrast enhancement," *Complexity*, vol. 20, no. 2, pp. 22–36, 2014.

[17] OpenCV: Open Source Computer Vision. "Morphological transformations." (), [Online]. Available: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html (visited on 07/21/2022).

[18] U. RAMER, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244–256, 1936.

[19] OpenCV: Open Source Computer Vision. "Approxpolydp." (), [Online]. Available: https://opencv-laboratory.readthedocs.io/en/latest/nodes/imgproc/approxPolyDP.html (visited on 07/22/2022).

[20] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological Cybernetics*, vol. 20, pp. 121–136, 1975.

[21] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–400, 1951.

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2015. DOI: https://doi.org/10.48550/arXiv.1412.6980. (visited on 07/21/2022).

[23] M. Riedmiller and H. Braun, "Rprop - A Fast Adaptive Learning Algorithm," *International Symposium on Computer and Information Science*, vol. VII, 1992.

[24] G. Nakerst, J. Brennan, and M. Haque, "Gradient descent with momentum — to accelerate or to super-accelerate?" *arxiv.org*, 2020. DOI: arXiv:2001.06472. (visited on 07/21/2022).

[25] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. II, pp. 179–188, 1936.