PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Richard Bryan Magalhães Santos

Use of Deep Convolutional Neural Networks in Automatic Recognition and Classification of Coal Macerals

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia de Materiais e de Processos Químicos e Metalúrgicos of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia de Materiais e de Processos Químicos e Metalúrgicos.

Advisor: Prof. Sidnei Paciornik

Coadvisors: Dra. Karen Soares Augusto
Dr. Julio César Álvarez Iglesias

Rio de Janeiro
August 2022

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Richard Bryan Magalhães Santos

Use of Deep Convolutional Neural Networks in Automatic Recognition and Classification of Coal Macerals

Thesis presented to the Programa de Pós–graduação em Engenharia de Materiais e de Processos Químicos e Metalúrgicos of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia de Materiais e de Processos Químicos e Metalúrgicos. Approved by the undersigned Examination Committee.

Prof. Sidnei Paciornik
Advisor
Departamento de Engenharia Química e de Materiais - PUC-Rio

Dr. Karen Soares Augusto
Co-advisor
Departamento de Engenharia Química e de Materiais - PUC-Rio

Dr. Julio César Álvarez Iglesias
Artificial Intelligence & Data, Deloitte Touche Tohmatsu

Prof. Joan Sharon Esterle
School of Earth and Environmental Sciences- UQ

Prof. Sandra Paulo Rodrigues
School of Earth and Environmental Sciences- UQ

Prof. Guilherme Lucio Abelha Mota
Departamento de Informática e Ciência da Computação-UERJ

Dr. Marcos Henrique de Pinho Maurício
Departamento de Engenharia Química e de Materiais - PUC-Rio

Rio de Janeiro, August the 29th,2022

Richard Bryan Magalhães Santos

Graduated in Materials and Nanotechnology Engineering at PUC-Rio in 2016. Master's Degree in Materials Engineering and Chemical and Metallurgical Processes at PUC-Rio in 2017, specializing in Materials Science, Digital Microscopy, and Image Processing and Analysis.

Bibliographic data

Santos, Richard Bryan Magalhães.

Use of deep convolutional neural networks in automatic recognition and classification of coal macerals / Richard Bryan Magalhães Santos; advisor: Sidnei Paciornik; co-advisors: Karen Soares Augusto, Julio César Álvarez Iglesias. – 2022.
116 f. : il. color. ; 30 cm

Tese (doutorado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Química e de Materiais, 2022.
Inclui bibliografia

1. Engenharia Química e de Materiais – Teses. 2. Redes neurais convolucionais. 3. Refletância de vitrinita. 4. Petrografia de carvão. 5. Análise de macerais. 6. Processamento digital de imagens. I. Paciornik, Sidnei. II. Augusto, Karen Soares. III. Álvarez Iglesias, Julio César. IV. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Química e de Materiais. V. Título.

CDD: 620.11

Acknowledgements

To professor Sidnei Paciornik, who encouraged me to pursue this PhD.

To professors Joan Esterle and Sandra Rodrigues, for taking me in in Australia and teaching me so much about coal.

To Doctors Karen Augusto and Julio Cesar, for co-advising during this time.

To family, for all support and understanding given to me during the last 4 years.

To all my friends from the lab, that helped make the work lighter and more tolerable during the pandemic.

To CNPq and PUC-Rio, for the financial support necessary to finish this thesis.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior- Brasil (Capes) - Finance Code 001.

Abstract

Santos, Richard Bryan Magalhães; Paciornik, Sidnei (Advisor); Augusto, Karen Soares (Co-advisor); Iglesias, Julio Cesar Alvarez (Co-advisor)**. Use of Deep Convolutional Neural Networks in Automatic Recognition and Classification of Coal Macerals.** Rio de Janeiro, 2022. 116p. Tese de Doutorado – Departamento de Engenharia Química e de Materiais, Pontifícia Universidade Católica do Rio de Janeiro.

Unlike most other rocks, coal is a sedimentary rock composed primarily of organic matter derived from plant debris that accumulated in peat mires during different geological periods. Coal is also an essential economic resource in many countries, having been the main driving force behind the industrial revolution. Coal is still widely used industrially for many different purposes: carbonization and coke production, iron/steel making, thermal coal to generate electricity, liquefaction, and gasification. The utility of the coal is dictated by its properties which are commonly referred to as its rank, type, and grade. Coal composition, in terms of its macerals, and its rank determination are determined manually by a petrographer due to its complex nature. This study aimed to develop an automatic method based on machine learning capable of maceral segmentation at group level followed by a module for rank reflectance determination on petrographic images of coal that can improve the efficiency of this process and decrease operator subjectivity. Firstly, a Mask R-CNN-based architecture deep learning approach was developed to identify and segment the vitrinite maceral group, which is fundamental for rank analysis, as rank is determined by collotelinite reflectance (one of its individual macerals). Secondly, an image processing method was developed to analyze the vitrinite segmented images and determine coal rank by associating the grey values with the reflectance. For the maceral (group) segmentation, five samples were used to train the network, 174 images were used for training, and 86 were used for testing, with the best results obtained for the vitrinite, inertinite, liptinite, and collotelinite models (89.23%, 68.81%, 37.00% and 84.77% F1-score, respectively). Those samples were used alongside another eight samples to determine the rank results utilizing collotelinite reflectance. The samples ranged from 0.97% to 1.8% reflectance. This method should help save time and labor for analysis if implemented into a production model. The root mean square calculated between the proposed method and the reference reflectance values was 0.0978.

Keywords

Digital Image Processing; Vitrinite Reflectance; Convolutional Neural Networks; Coal Petrography; Maceral Analysis.

Resumo

Santos, Richard Bryan Magalhães; Paciornik, Sidnei (Orientador); Augusto, Karen Soares (Coorientadora); Iglesias, Julio Cesar Alvarez (Coorientador)**. Uso de Redes Neurais Convolucionais Profundas para Reconhecimento e Classificação Automáticas de Macerais de Carvão.** Rio de Janeiro, 2022. 116p. Tese de Doutorado – Departamento de Engenharia Química e de Materiais, Pontifícia Universidade Católica do Rio de Janeiro.

Diferentemente de muitas outras rochas, o carvão é uma rocha sedimentar composta principalmente de matéria orgânica derivada de detritos vegetais, acumulados em turfeiras em diferentes períodos geológicos. O carvão é um recurso econômico essencial em muitos países, tendo sido a principal força motriz por trás da revolução industrial. O carvão é amplamente utilizado industrialmente para diversos fins: carbonização e produção de coque, produção de ferro/aço, carvão térmico para gerar eletricidade, liquefação e gaseificação. A utilização do carvão é ditada pelas suas propriedades que são geralmente classificadas como sua composição, rank e grau. A composição do carvão, em termos dos seus macerais, e a sua classificação são determinadas manualmente por um petrógrafo, devido à sua natureza complexa. Este estudo almejou desenvolver um método automático baseado na aprendizagem de máquina para segmentação automática de macerais a nível de grupo e um módulo para determinação de rank por refletância em imagens petrográficas do carvão que pode melhorar a eficiência deste processo e diminuir a subjetividade do operador. foi desenvolvida uma abordagem de aprendizagem profunda da arquitetura baseada na Mask R-CNN para identificar e segmentar o grupo de maceral vitrinite, o qual é fundamental para a análise do rank, uma vez que a classificação é determinada pela reflectância da collotelinite (maceral desse grupo). Em segundo lugar, foi desenvolvido um método de processamento de imagem para analisar as imagens segmentadas de vitrinite e determinar a classificação do carvão, associando os valores cinzentos à reflectância. Para a segmentação de maceral, foram utilizadas cinco amostras para treinar a rede, 174 imagens foram utilizadas para treino, e 86 foram utilizadas para validação, com os melhores resultados obtidos para os modelos de vitrinite, inertinita, liptinita e colotelinita (89,23%, 68,81%, 37,00% e 84,77% F1-score, respectivamente). Essas amostras foram utilizadas juntamente com outras oito amostras para determinar os resultados de classificação utilizando a reflectância de collotelinite. As amostras variaram entre 0,97% e 1,8% de reflectância. Este método deverá ajudar a poupar tempo e mão-de-obra para análise, se implementado num modelo de produção. O desvio médio quadrático entre o método proposto e os valores de reflectância de referência foi de 0,0978.

Palavras-Chave

Processamento Digital de Imagens; Refletância de Vitrinita; Redes Neurais Convolucionais; Petrografia de Carvão; Análise de Macerais.

Summary

List of Figures

List of Tables

1
Introduction

Coal is still used as a raw material for many manufacturing processes, especially in the steel industry, such as pig iron production, ferroalloy production, and cement manufacturing. The macerals, the individual constituents of organic matter, influence the quality of the coals; they can be recognized by microscopic analysis. Thus, the microstructural characterization of coal by optical microscopy is a fundamental tool, allowing the identification of the macerals by reflectance, color, morphology, anisotropy, size, and relief or polishing hardness. The most important parameters in coal analysis are its **rank** and **maceral composition**. The macerals originate from different plant tissues after many physico-chemical changes due to heat and pressure during the coalification process. Coalification refers to the process of the peat turning into coal as it changes its chemical composition and increases its carbon content. Rank is a measurement of how advanced the coal is in the coalification process, one of the most common ways to evaluate it is by measuring the reflectance of the vitrinite macerals group, which increases in a well-behaved manner during coalification (ICCP 1998, 2001; SUAREZ-RUIZ & CRELLING, 2008; TAYLOR et al., 1998).

The analysis of these macerals is traditionally performed by a trained coal petrographer, who inspects the polished surface of a specimen under an optical microscope (TAYLOR ET AL., 1998). A percentage of each maceral is estimated based on counting a statistical number of points, sweeping the entire sample surface. The procedures for maceral analysis follow the recommendations of the ASTM D2799-11 (2011), ISO 7404-3 (2009), or AS 2856.2 (1998). It is important to note that, while coal petrology refers to the study of coal, coal petrography is limited to the techniques of evaluation of coal rank and type.

The most significant limitations of the manual approach are the dependency on the subjectivity and level of training of different petrographers, errors due to human fatigue, and, most importantly, the fact that it is a relatively lengthy process. The reason why it is done manually lies in the fact that maceral classification is an extraordinarily complex task, as coal itself is a complex material. Macerals are identified by their size, shape, color, reflectivity, texture, fungal presence, degradation, degree of gelification, fluorescence, and relative position to other macerals (TAYLOR et al., 1998). No clear set of easily extractable parameters can be used to define a maceral, rendering traditional image analysis and machine learning ineffective.

This motivates the use of deep learning methods, which have proven to be highly efficient for classifying and recognizing patterns. Unlike traditional neural networks, where it is necessary to extract the relevant attributes previously,

convolutional neural networks (CNNs) (BOUVRIE, J., 2006) can learn to extract the relevant characteristics of the image and make a decision as to the classification at the same time.

The three major maceral groups, vitrinite, inertinite, and liptinite, are derived from relatively well-preserved lignin-rich tissues, degraded or oxidized tissues, and lipid-rich plant matter, respectively (ICCP 1998, 2001; PICKEL et al., 2017). These main groups have different optical properties as well as different morphologies, texture and degrees of gelification. Out of all the optical properties, however, the most important one for maceral classification at group level is their reflectance in the reflected light microscope. Liptinite displays a dark grey to brown color, vitrinite displays an intermediate grey color, and inertinite shows a light grey to white color. However, as the rank increases, the maceral groups experience an increase in reflectance (SMITH & COOK, 1980), making it difficult to distinguish between them as they converge. The most consistent changes occur within the vitrinite group macerals throughout coalification. The reflectance of vitrinite, particularly that of the smooth maceral collotelinite derived commonly from woody cell walls (ICCP 1993, 1998), is typically used as a measure of the sample's rank.

The current work aimed to develop an automatic maceral segmentation system that eventually settled for maceral group-level segmentation and a method to determine coal rank reflectance based on the collotelinite segmentation results. A dataset containing 260 images with coals ranging from 0.97% to 1.2% reflectance was built with over 17000 annotations being hand-made. The annotations were made at maceral group, maceral subgroup and individual maceral level; different models were trained considering different maceral groupings and their uses and limitations were discussed.

2
Problem Description and Objectives

Coal is widely used material industrially, its applications ranging from iron production to the electricity and heat generation. Coal's usage is heavily dictated by its properties, of importance to this work being its maceral composition and rank. Maceral is to coal what mineral is to rock, that is, the individual chemical components that compose coal. The three main groups of macerals ae: vitrinite, inertinite and liptinite. Coal rank is a measurement of its maturation stage in the coalification process. There are many ways to evaluate coal rank, this work concerned itself with rank evaluation by means of vitrinite reflectance measurement.

The main objectives of this thesis can then be summarized as: developing a convolutional neural network (CNN) system capable of automatically identifying, classifying, and segmenting macerals in reflected light optical microscopy (RLOM) coal images at maceral group level and possibly individual maceral.

Specific Objectives:

- Build up an extensive databank consisting of as many images as possible of each industrially relevant maceral class.

- Develop a reproducible method of capturing RLOM images from the macerals.

- Annotate extensively and classify the maceral groups present in these images.

- Train the neural network and optimize architecture and parameters for maceral segmentation.

- Select the most appropriate segmentation model for rank reflectance recognition.
- Select the most appropriate segmentation models for maceral composition determination
- Develop a methodology to determine rank reflectance using the segmented images from the model above.

3
Bibliographic review

3.1 What is coal?

Coal is a sedimentary rock that, unlike the majority of other rocks, is composed mostly of organic matter derived from plant debris that accumulated in peat mires during different geological periods (TAYLOR et al., 1998). As a result, coal is chemically composed of carbon and hydrogen, sulfur, oxygen, and nitrogen compounds (HAENEL, 1992). The plant debris underwent decomposition in the peat phase, followed by low-grade metamorphism from increasing temperature and pressure in the subsurface as layers of waste were buried in the subsurface. These layers became the coal seams that are extracted today. The different ranks of coal can be grouped into the following classes (ISO 11760:2018, 2018) (Figure 1 and Figure 2):

- Lignite (often referred to as brown coal) and sub-bituminous coals are used chiefly for fuel and electricity generation.

- Bituminous coal (often referred to as black coal), which is the most abundant type of coal and is also used for electricity generation, but more importantly for coking and steel making (being, therefore, the focus of this work)

- Anthracite is used mainly for residential and commercial space heating.

The utility of the coal is dictated by its properties which are commonly referred to as its rank, type, and grade. This work concerns itself with only the first two, which will be presented below.

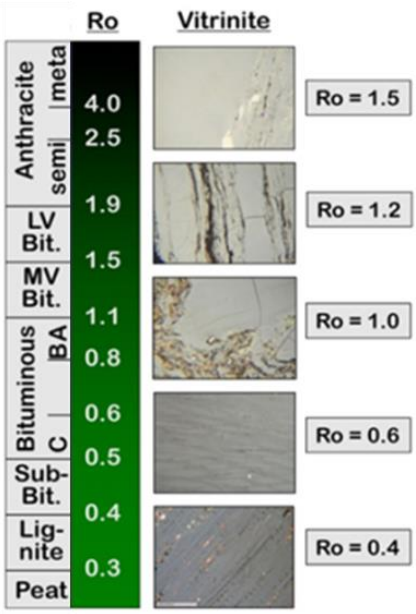

Figure 1 - Increase in vitrinite reflectance with increasing rank from peat to anthracite. From https://geology.com/rocks/coal.shtml.

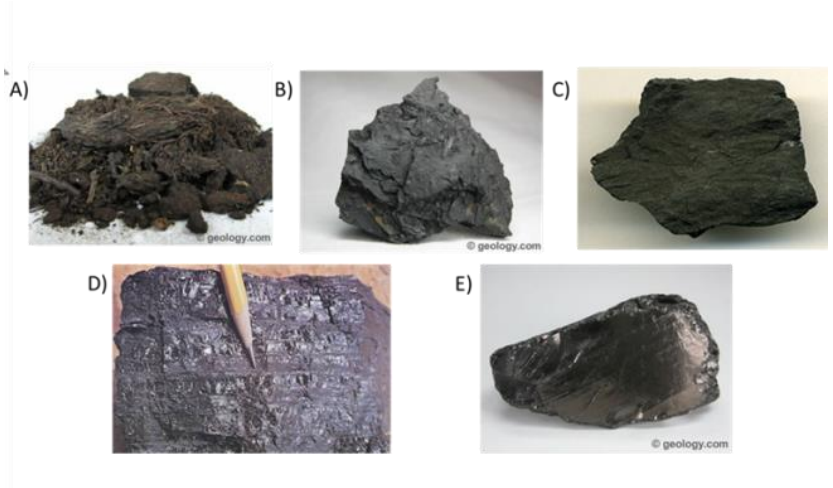

Figure 2 - A-Peat, B-Lignite, C-Sub-bituminous coal, D-Bituminous coal, E-Anthracite. Note that the pencil in D also shows the different bands of vitreous and dull coal, most evident in bituminous rank coals. From https://geology.com/rocks/coal.shtml.

3.2 The coalification processes and coal rank

The plant material undergoes geological processes which convert it into coal, and these processes are called peatification and coalification. During the peat phase, biochemical decay processes occur, led by fungi and aerobic bacteria (TEICHMULLER & TEICHMULLER, 1979). During coalification, a series of physical and chemical transformations emerge promoted by compaction, pressure, and heat with prolonged burial at depths of up to several kilometers and over several hundred million years (TEICHMULLER & TEICHMULLER, 1979). The

coalification process can be evaluated as the coal rank, measured by the decrease in the volatile matter, hydrogen, oxygen, and moisture, and a consequent increase in the carbon content. Accordingly, porosity decreases, and the amount of light that reflects off the coal surface, or reflectance, increases. These transformations can be seen macroscopically in hand specimens (the coals change from brown to black and show an increase in sheen) and under the optical microscope, where the plant material referred to as macerals or recognizable macerated plant fragments, will have different reflectivity.

The vitrinite maceral group (especially the collotelinite maceral) has the most consistent and stable increase during coalification. Because of that, vitrinite reflectance is often used as a good measure of the coal rank.

3.3 Peat accumulation processes leading to coal type

The coal deposits originated in different geological periods, notably in the late Carboniferous and Permian periods, from which most coking quality coals derive. For peat (the precursor of coal) to be formed, two conditions had to be met. There should be enough organic matter deposited as the plants die, and some of this matter would have to remain preserved throughout the peatification and coalification processes. As partial preservation is achieved by means of burial, the burial rate under later deposits must exceed the decomposition rate of organic matter decomposition. The ideal conditions for peatification are a slowly subsiding area or basin, anoxic conditions to slow decay, minimal flooding by sediment-laden water, and humid conditions. These are found in wetlands, and peat will accumulate as long as the groundwater table maintains a stable relationship with the peat deposits, that is, not rising fast enough to flood the deposits and not receding too quickly to expose organic matter and already formed peat to erosion and decomposition (TAYLOR et al., 1998). For the peat to become coal, it is eventually buried underneath layers of sediment that amass as the geological basin subsides.

The plants growing in wetlands today and accumulating peat vary from trees with dense wood and bark in their boughs, trunks, and root systems to smaller shrubs and non-woody plants such as ferns and mosses, among others. Plant tissues have different chemical compositions with varying pathways of decomposition that lead to different coal types. Two critical processes occur during peatification. The first one is the decomposition of hydrolyzable and nitrogen-rich substances of plants, such as cellulose, hemicellulose, starch, and proteins (giving off liquids and gases like water, methane, carbon dioxide, and ammonia). The more resistant substances like lignins and tannins become thus more concentrated. The second and most important process is humification, where substances like cellulose, and mainly lignin, are transformed into humic substances (TAYLOR et al., 1998). Humic substances are complex structures containing carbon, oxygen, hydrogen, and nitrogen; they have an aromatic nucleus and hydroxyl (-OH) and carboxyl (COOH) functional groups. These humic

substances remain and undergo coalification to originate the huminite (brown coal rank) and vitrinite (black coal rank) group macerals. Plant tissues that are rich in long chain aliphatics such as suberin-rich bark, leaf cuticles, resins, and spores, or those which were burnt to charcoal or oxidized enzymatically by fungus are the constituents of the peat that do not undergo humification and have their chemical composition mostly untouched during this process (TAYLOR et al., 1998). These form the maceral groups, liptinite, and inertinite.

In hand specimens of coal, the proportion of the different plant tissues preserved in the peat phase is recognizable as stratified layers or bands. The preserved wood and bark bands are observable as such in lower rank lignites, but with the increased compaction and loss of moisture and porosity in higher rank black coals, they appear shiny or vitreous (like glass) and hence were called vitrain bands by Marie Stopes (STOPES, 1935). The more decomposed or macerated and gelified plant detritus is compacted, but its micro-granular texture is slightly more porous and appears duller. It is also mechanically tougher or harder than the vitrain bands and was called durain. Charcoal in coal has a fibrous luster and is called fusain. Different proportions of the end member vitrain and durain bands are used to characterize coals layers into different lithotypes. Thus, coal seams are composed of different lithotypes that often show a vertical stratigraphic pattern or profile reflecting the succession of other plants responding to the rising (or falling) water tables in the peat phase. These different megascopic lithotypes directly relate to the microscopic maceral composition, which is presented later.

3.4 Maceral definition and criteria for recognition

This works focuses on the approach proposed by Marie Stopes – the Stopes-Heerlen system (SH) – which views coal as a rock (STOPES, 1935), but more specifically on the ICCP classification. Stopes proposed the term maceral from the Latin *macerare*, which means to soften, to describe the individual organic components of coal. This would then be to coal analogous to the term mineral to other rocks. Stopes proposed the classification of these macerals into three main groups: vitrinite, inertinite, and liptinite (previously exinite). Although the Stopes system is still maintained, the individual maceral classification system has been modified over the years, resulting in a group, sub-group, and individual maceral subdivision. The maceral names, classification, sub-group names and their overall grouping can vary depending on which of the many standards is being considered. Table 1 shows the maceral classification used in this work; it was inspired by both the International Committee for Coal and Organic Petrology (ICCP) standards and the Australian standard (AS) for maceral classification (AS, 1986; ICCP, 1998, 2001; PICKEL et al., 2017; SYKOROVA et al., 2005).

The ICCP subdivision can be understood as follows:

-Maceral group is defined by reflectance.

-Maceral subgroup is defined by the degree of destruction.

-Individual maceral is defined by morphology and degree of gelification.

Table 1- ICCP classification of coal macerals.

Maceral Group	Maceral Subgroup	Maceral
Vitrinite	Telovitrinite	Collotelinite
		Telinite
	Detrovitrinite	Collodetrinite
		Vitrodetrinite
	Gelovitrinite	Corpogelinite
		Gelinite
Inertinite	Teloinertinite	Fusinite
		Semifusinite
		Funginite
	Geloinertinite	Macrinite
		Secretinite
		Micrinite
	Detroinertinite	Inertodetrinite
Liptinite	X	Sporinite
		Cutinite
		Resinite
		Alginite
		Liptodetrinite

3.4.1 The Vitrinite Group

The **vitrinite group** receives its name from *vitrum*, Latin for glass (STOPES, 1935). Generally, it is the most abundant of the groups being the leading maceral coal group; it is defined by having an intermediary reflectance between the other two groups over the rank range where they can still be distinguished.

Vitrinite reflectance ranges from 0.5% in low-rank coals to over 4% in high-rank coals, going from a dark grey tone to a very light grey or even white. Liptinite reflectance surpasses that of vitrinite in upper medium rank (around 1.5% max reflectance) coals, and both overtake inertinite in high-rank coals.

Vitrinite is soft and usually does not show any relief related to other macerals. Vitrinite has considerably higher oxygen content than the other groups, its chemical composition, more specifically its elemental composition, varies during coalification. Its carbon content increases, whereas its oxygen content decreases steadily alongside hydrogen content. Vitrinite is a significant source of natural gas, and, in medium-rank coals, it fuses during carbonization, influencing hydrogenation and combustion (ICCP, 1998).

Vitrinite can be subdivided into three subgroups: telovitrinite (tela from Latin meaning tissue), gelovitrinite (gelu from Latin meaning frost), and detrovitrinite (detritus from Latin meaning abrasion) (ICCP, 1998).

The Telovitrinite Subgroup is the first and technologically the most important of the vitrinite subgroups. It is meant to include all vitrinite that preserves its cellular structure regardless of whether it can still be seen under reflected light. This subgroup originates from tissues such as bark, roots, stems, and leaves, which are lignin- and cellulose-rich. It is comprised of two main macerals: collotelinite and tellinite.

Collotelinite is the most abundant of the individual macerals (Figure 3). It is characterized mainly by its homogeneity, having no visible cell structure. Its reflectance is the primary rank parameter used for coal and is usually higher than the reflectance of collodetrinite. Collotelinite goes from a dark gray in lower ranks to bright gray or even white at higher ranks (Figure 1). It's composed elementally mainly of carbon and becomes more "closed" and aromatized during coalification which is the cause for its observed reflectance increase (ICCP, 1998).

It is a reactive maceral important to many technological applications, including carbonization and liquefaction for fuel production. The collotelinite's grey value is usually chosen to represent the rank of the sample due to its consistent increase during the coalification process.

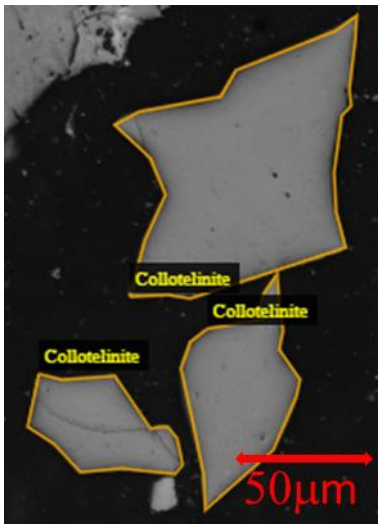

Figure 3 - Collotelinite maceral from the telovitrinite subgroup and vitrinite group (1.20% R_R).

Telinite is an individual maceral characterized by having its cellular structure visible (Figure 4). Despite variations in size and shape, it preserves the appearance of a cell lumen, usually filled by other macerals such as corpogelinite and resinite, sometimes even mineral matter (ICCP, 1998).

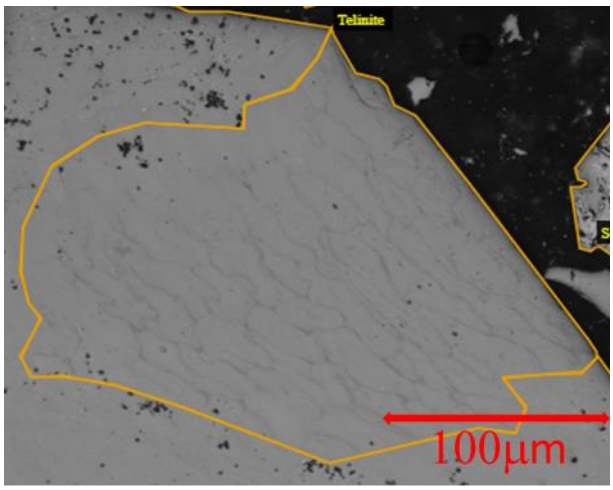

Figure 4 - Telinite maceral from the telovitrinite subgroup and vitrinite group with a corpogelinite filling. (1.20% R_R).

Telinite can only be identified when it has different reflectance from its filling inclusions. Most of these inclusions are brighter, but resinite is usually darker than the tellinite cell walls. It is less abundant than collotelinite, and, due to the reflectance convergence of vitrinite macerals, at high ranks, it can be indistinguishable from collotelinite if it has a vitrinitic filling (ICCP, 1998).

The Detrovitrinite subgroup is characterized by vitrinized plant remains; it has the same origin as the telovitrinite subgroup but undergoes further chemical/physical degradation. It is subdivided into two macerals, namely: vitrodetrinite and detrovitrinite (ICCP, 1998).

Vitrodetrinite is an individual maceral composed of small vitrinitic fragments smaller than 10 µm in the maximum direction. Vitrodetrinite originates from extensive comminution from parenchymatous and woody tissues of roots, stems, and leaves composed of cellulose and lignin. its gelification process can occur either before transportation and deposition, or after sedimentation. It is also considered a rare occurring maceral in coal unless the sample surface is etched beforehand to reveal it (ICCP, 1998). Only three small instances of vitrodetrinite were found in this work.

Collodetrinite differs from collotelinite by being usually slightly darker, but more importantly, by playing the role of a matrix that contains and binds together pieces and fragments of other macerals and possibly mineral matter (Figure 5). Unlike vitrodetrinite, the vitrinite here appears as a matrix, not being possible to identify individual vitrinite particles. It has a similar formation to the collotelinite, differing only because of the foreign particles trapped in the vitrinite before the gelification process (ICCP, 1998).

Alongside collotelinite, it is one of the most abundant macerals found in coal, sometimes even more common than collotelinite itself; due to its high amount of volatile matter, it degasses considerably during carbonization, making it a significant factor in the coal's caking and coking properties. Collodetrinite usually displays a slightly lower reflectance (0.05-.0.10%) than collotelinite. This difference, however, tends to disappear as the rank increases (ICCP, 1998).

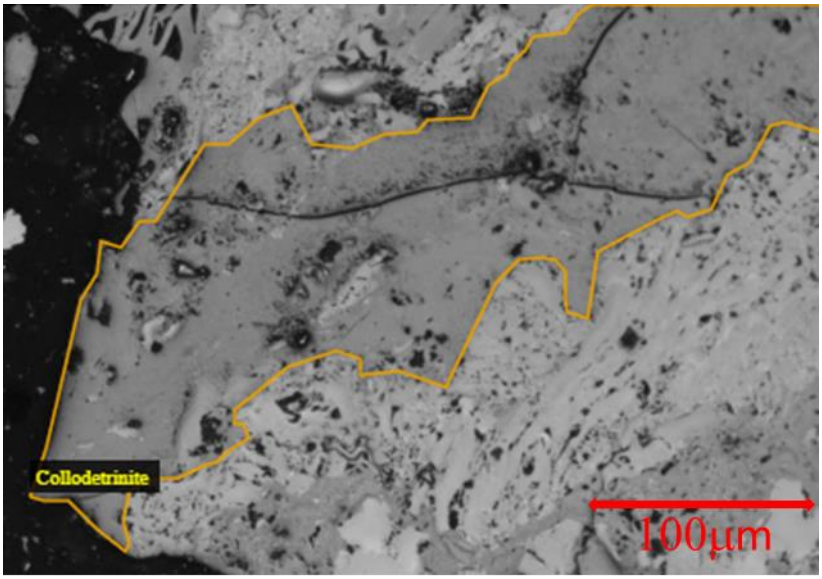

Figure 5 - Collodetrinite maceral from the detrovitrinite subgroup and vitrinite group.

The Gelovitrinite subgroup is a vitrinite maceral subgroup comprising macerals of many possible origins. Still, it is mainly assumed to be formed from decayed plant tissues that precipitated as gels within cavities in other available macerals, hence the name gelo. Its main macerals are corpogelinite and gelinite (ICCP, 1998).

Corpogelinite receives its name from "body" (corpus meaning body, in Latin) and has the appearance of small structureless bodies usually found as infillings in cavities from other macerals (Figure 6). They can be spherical but elongated and more oval with a variable size. It is usually slightly brighter than other vitrinites of the same rank, contrasting with the typically collotelinitic walls surrounding it (ICCP, 1998).

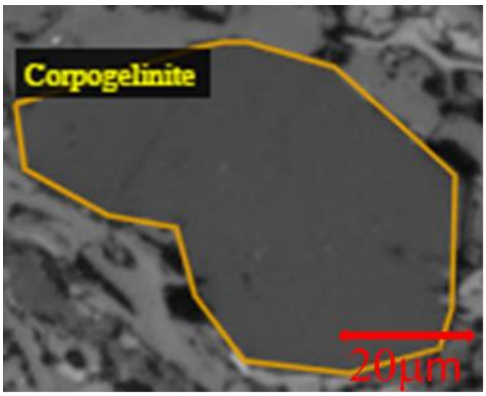

Figure 6 - Corpogelinite maceral from the gelovitrinite subgroup and vitrinite group.

Gelinite is a maceral consisting of colloidal gels filled in cracks and voids in other macerals. It is structureless and homogeneous and does not have the body-like shape presented by corpogelinite. Like corpogelinite, it has a higher reflectance than other vitrinites. It is the most uncommon vitrinite maceral, more likely found in coals exhibiting high degradation (ICCP, 1998). Only two possible instances of gelinite were found in this work

3.4.2 The Inertinite Group

The inertinite group receives its name from inertia, Latin for inactivity, for they are considered more inert (non-reactive) than the vitrinite macerals in most technological processes involving coal. They are defined by having higher reflectance than vitrinite for medium to low-rank coals (<5%Rmax of vitrinite). Its color varies from light grey to white and even yellowish white at high ranks.

Inertinite macerals can vary broadly in shape, reflectance, and texture; some lower reflectance inertinites with less defined structure can be almost indistinguishable from vitrinites. This is especially true between collotelinite, collodetrinite and semifusinite, which can be exacerbated when both are present in the same particle and display a smooth transition.

Inertinites have high carbon and low hydrogen and oxygen content, having the least pronounced reflectance increase over the coalification process. Inertinite may derive from tissues (from plants or fungi), detrital fragments, gelified amorphous material, and cell secretions. It may preserve highly intricate cell walls as in some fusinites or exhibit almost completely amorphous structure as some semifusinites (ICCP, 2001). The inertinite group can be subdivided into subgroups according to the presence of cell structure (with telo-inertinite presenting it and gelo-inertinite not) and whether they are classified as detritus (in that case being detro-inertinites) (AS, 1986).

Fusinite (fusus meaning spine in Latin) is a structured **teloinertinite subgroup** maceral characterized by its easily recognizable, well-preserved cell

structure (Figure 7). Fusinite has a very high carbon content that increases even further with rank.

Fusinite is highly brittle and, when finely dispersed, can improve coke strength; it cannot be used for caking capacity and acts as an inert aggregate material in coal blends. Due to its high carbon content, fusinite is mostly inert and does not fuse during combustion.

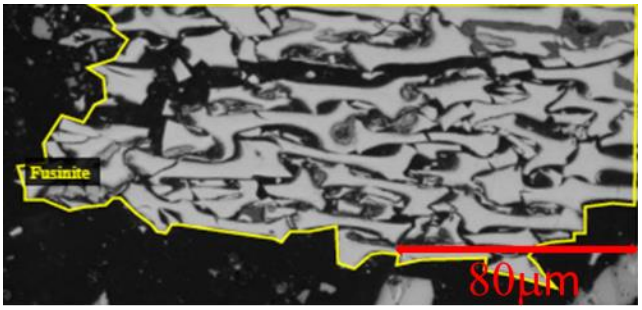

Figure 7 - Fusinite maceral from the teloinertinite subgroup and inertinite group.

Semifusinite is a teloinertinite subgroup maceral defined by displaying intermediary reflectance and structure compared to collotelinite and fusinite (Figure 8). The maceral can be slight to highly amorphous, with its cell walls barely recognizable and cavities filled with other macerals or minerals. Its reflectance can range from just above that of collotelinite of the same rank to just below that of fusinite. On the extreme ends of this range, it can be hardly distinguishable from the other two macerals and, like both, is a ubiquitous coal component worldwide (ICCP, 2001).

Semifusinite is partially reactive during conversion processes, with a portion behaving as fusinite and a portion like collotelinite. This property makes it strictly necessary for a good classification system to correctly identify, quantify and distinguish it from the collotelinite and fusinite. Its potential similarity to those two macerals represents one of the biggest obstacles identified here to proper maceral classification (ICCP, 2001).

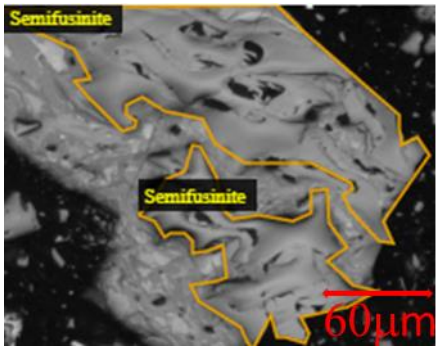

Figure 8 - Semifusinite maceral from the teloinertinite subgroup and inertinite group.

Funginite is a teloinertinite subgroup maceral defined by being derived from fungal tissue such as fungal spores, sclerotia, mycelia, and other fungal tissue (Figure 9). Like other inertinites, it has a reflectance slightly above that of vitrinite over the range they can be distinguished. Unlike fusinite, however, its reflectance becomes indistinguishable from that of vitrinite relatively early, at around 1.6%R_{max}; however, it still retains a very distinct structure. Due to the small quantities of this maceral in coal, there is no definitive conclusion about its influence on the coal properties (ICCP, 2001).

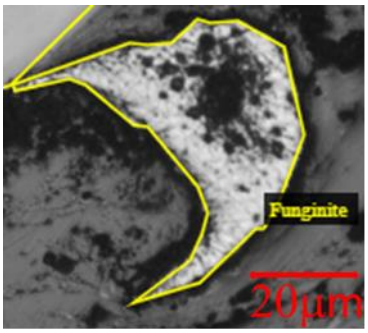

Figure 9 - Funginite maceral from the teloinertinite subgroup and inertinite group.

Secretinite is a **geloinertinite subgroup** maceral that has a possible origin in the secretory ducts of seed ferns (Figure 10). Its origins and chemical composition are, however, not clearly defined. Its appearance is that of small round to oval bodies possessing one or more slits on its surface, its most striking feature. Secretinite may have reflectance slightly above that of vitrinite up to even above that of fusinite. It is distinguishable from macrinite due to its slits and relief (ICCP, 2001).

Like fusinite, secretinite doesn't fuse during coking and, when finely dispersed, enhances coke strength.

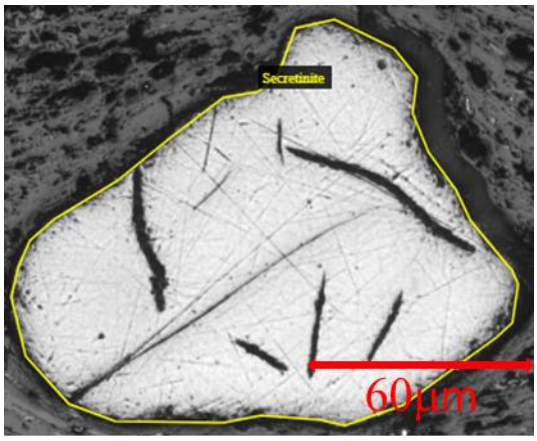

Figure 10 - Secretinite maceral from the geloinertinite subgroup and inertinite group.

Macrinite is a geloinertinite subgroup maceral that comprises macerals of varying shapes and reflectance. Macrinite can look spherical/oval but also like an amorphous matrix (Figure 11). When it has small oval shapes, it can be distinguished from secretinite by the absence of slits and by its smoother edges. When it resembles a more extensive particle, it can be distinguished from semifusinite by an even more pronounced lack of structure and smooth surface and from vitrinite due to its higher reflectance.

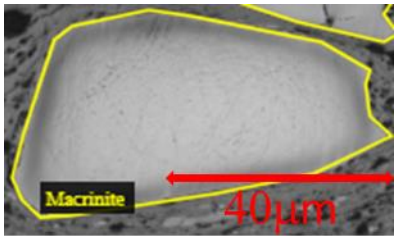

Figure 11 - Macrinite maceral from the geloinertinite subgroup and inertinite group.

Its technological importance is related to its reflectance; the lower the reflectance, the more reactive like the vitrinite it is; the higher the reflectance, the less reactive it is (ICCP, 2001).

Micrinite is a geloinertinite subgroup maceral characterized by small inertinitic fragments with an upper size limit of 2 µm. Micrinite can be grey to white and becomes indistinguishable from vitrinite at around 1.4%R_{max}. Due to its small size, it can be challenging to identify with a 20x objective lens (OL), making it challenging to train an automated system. It is believed to be reactive, but its importance to technological processes hasn't been fully assessed due to its small size and proportion. Because this work captured images with a 20X OL, the magnification was insufficient to identify micrinite correctly; no micrinites images were captured.

Inertodetrinite is a **detro-inertinite subgroup** maceral that consists of inertinitic fragments between 2 and 10 µm. Due to being composed of pieces from other macerals, its properties vary widely depending on its origin (Figure 12).

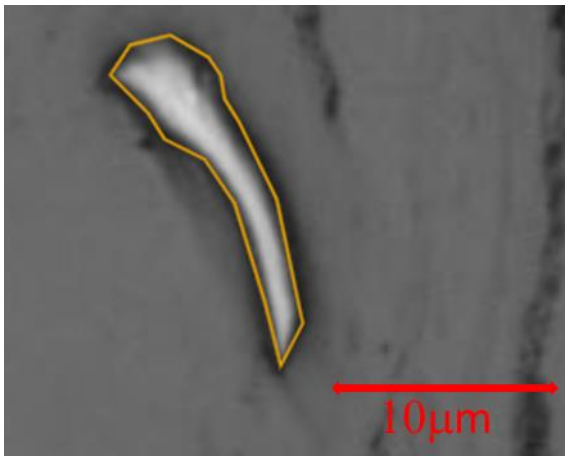

Figure 12 - Inertodetrinite maceral from the detro-inertinite subgroup and inertinite group

3.4.3 The Liptinite Group

Liptinite is a maceral group characterized by its high hydrogen content and lower reflectance than vitrinite in the range where they can be distinguished (until around 1.3%R_{rvitr}). Previously called exinites, they stem from non-humifiable plant matter such as cuticles, resins, spores, and fats. Liptinites yield the highest amount of by-products during coking, also a vital petroleum source. Liptinites rarely exist as particles in the resin and are usually found embedded in other macerals, mainly vitrinite (PICKEL, W. et al., 2017).

Liptinites are not classified in subgroups but directly in the individual macerals: sporinite, cutinite, resinite, suberinite, exsudatinite, chlorophyllinite, alginite, liptodetrinite, and bituminite. This work focuses on the most relevant (due to being the most frequent) liptinites in the 0.7%< R_{rvitr} <1.4%, sporinite, and cutinite.

Sporinites originate from spores and pollen; they have a greyish dark brown color and may look like small thin lines, with the format of its original spore still visible. However, some might stem from mega spores, making them harder to distinguish from mineral matter without fluorescence (PICKEL, W. et al., 2017) (Figure 13).

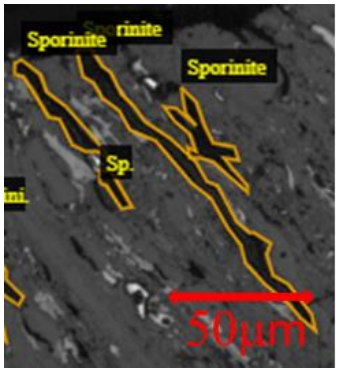

Figure 13 - Sporinite maceral from the liptinite group.

Cutinites originate from cuticles and stems; they appear as long thin lines and can be very similar to sporinites. Cutinites, however, are slightly lighter than sporinites of the same rank and have a serrated aspect to their appearance (Figure 14).

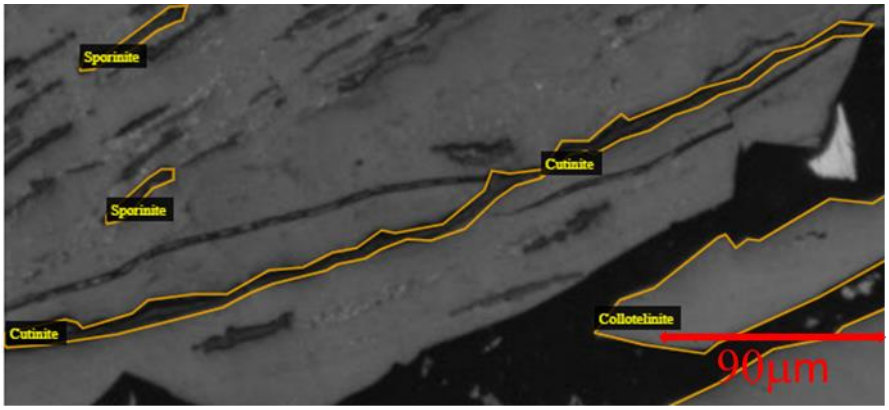

Figure 14 - Cutinite maceral from the liptinite group

3.5 Changes in maceral reflectance with increasing coalification or rank

As stated previously, coal changes physically and chemically during rank evolution, measured by bulk assay and optically. The gross calorific value (MJ/kg) can distinguish low-rank coals like lignites and high volatile bituminous coals. Volatile matter loss (weight loss when heating to 900°C) works well for higher rank anthracite coals and lower rank bituminous coals. The carbon content (dry and ash free) can be used for lower rank coal lignites to medium volatile bituminous coal. Hydrogen content is one of the best parameters for high-rank anthracites. Measuring the atomic O/H ratio is also possible when working with low-rank coals (TAYLOR et al., 1998).

The most widely used parameter nowadays, however, is the reflectance of the vitrinite maceral group (collotelinite specifically (ICCP, 1998), if possible). All maceral groups experience a reflectance increase during coalification (Figure 15), but vitrinite group macerals, particularly collotelinite, have the most stable increase. That, coupled with the fact that it is the most abundant and common maceral in coking coals, makes it one of the most used parameters industrially and academically for coal analysis. Figure 15 shows a comparison between reflectance and the other parameters as the rank evolves, illustrating how the vitrinite reflectance has the simplest, most stable, and reliable increase throughout ranks (ASTM D2798, 2011) (ASTM D2799, 2011) (TAYLOR et al., 1998).

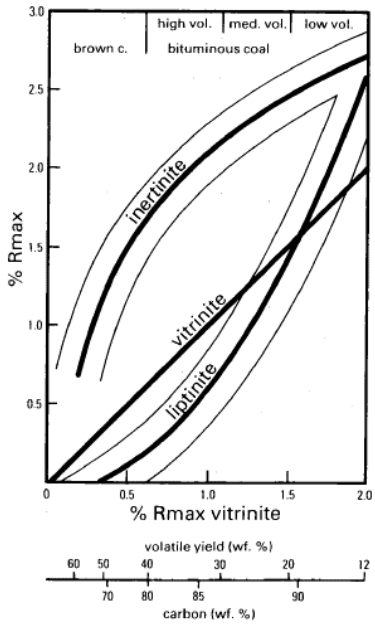

Figure 15- Relative progression of liptinite and inertinite reflectance compared to the progress of vitrinite reflectance as the coal rank increases (SMITH & COOK, 1980).

Being the only of the aforementioned parameters that can be easily extracted from an optical microscopy image makes reflectance (especifically random reflectance) the most important parameter for this thesis work, as the network can use it to help distinguish between maceral groups. However, due to the structural and morphological difference between macerals in the same groups, reflectance could never be used as the sole parameter to distinguish individual macerals (SMITH & COOK,1980) (TAYLOR et al., 1998). As the microscope captures light from the sample surface, the grey tone of objects in the image is directly related to their ability to reflect light, that is, their reflectance. By using a standard of well-known reflectance for calibration it is possible to measure the reflectance of any of the images' pixels afterwards. And as long as the lighting conditions are preserved between samples, their grey tones will always be comparable, which is essential for the network's training (Section 4.3).

3.6 Traditional Characterization

3.6.1 Maceral Analysis

In order to achieve large scale reproducibility and precision in both industrial and academic settings, many international standards were established for coal sample preparation (ASTM D 2797/D2797M, 2020; ISO 7404-2, 2009; AS 2856.1, 2000), maceral composition analysis (ASTM D 2799-05, 2021; ISO 7404-3, 2009; AS 2856.2, 1998) and coal rank reflectance determination (ASTM D 2798, 2021; ISO 7404-5, 2009; AS 2856.3, 2000). Coal samples have been prepared as polished particulate blocks as they are easier to prepare than thin sections (STOPES, 1935; TAYLOR et al., 1998). Microscope systems have also

been changing. Recently, automated and digital systems equipped with high-resolution cameras are referred to in the literature as the traditional photometer systems (e.g., BARBOLINI et al., 2019; PETERSEN et al., 2013; SANEI & ARDAKANI, 2016, among many others). These new systems are also included in the abovementioned standards, particularly for measuring reflectance.

Coal petrographers developed a widely used method for determining coal compositions in terms of its macerals, called point counting. Coal samples are first homogenized by passing through a 1 mm sieve and are embedded in a resin, which allows the samples to be used for both maceral and vitrinite reflectance analyses. For maceral composition determination, the samples are examined under a reflected light microscope with a mechanical stage capable of moving in increments or steps big enough so that negligibly few particles receive more than one count. The procedure should also be performed with an immersion objective of magnification between 25x and 60x with an eye piece of magnification between 8x and 12x. On each step, the point directly below the intersection of the cross-hairs must be analyzed, and if the point belongs to a maceral group, the counter increases for said group. If the point belongs to the resin or a hole/pore inside a maceral, it is skipped. The operator decides whether to include the mineral matter in the analysis; if that isn't the case, any point belonging to the mineral matter is also ignored. The procedure continues until at least 500 points have been collected (Figure 16) (ASTM-D2799, 2011; ISO 7404-3, 2009; SUAREZ-RUIZ & CRELLING, 2008).

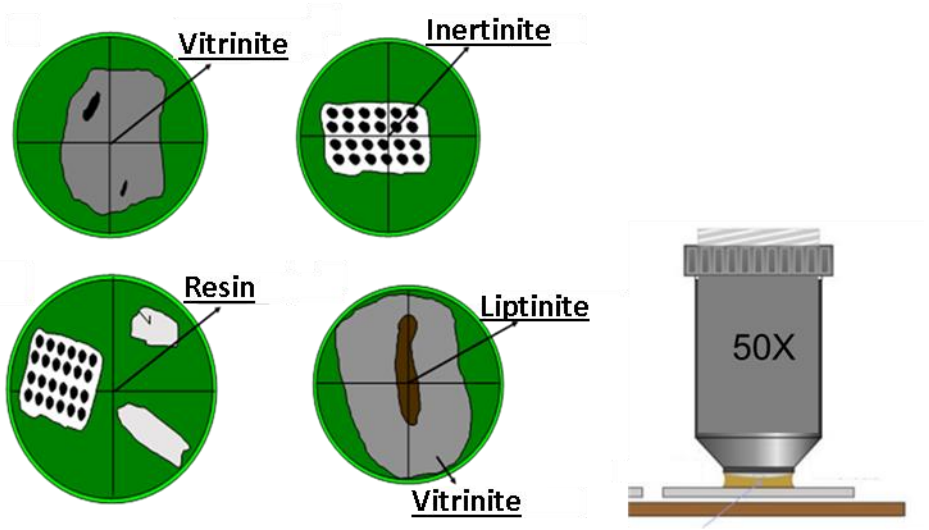

Figure 16 - Example of a typical point counting procedure.

The point percentage of the groups is understood to represent the volume fraction of the maceral groups in a sample, which means:

$$V_V+V_I+V_L=100 \quad (1)$$

where V_V is the volume fraction of vitrinite, V_I is inertinite, and V_L is the volume fraction of liptinite. The same procedure can be applied while classifying the

points in their maceral-subgroups or individual macerals. In the last case, the volume fractions represent the individual macerals, and the equation (1) will be adapted accordingly.

3.6.2 Collotelinite Reflectance Analysis

Coal rank by vitrinite reflectance petrographic analysis will be covered here according to the ASTM-D2798 standard. As described in the previous section, a microscope can also be used for rank reflectance determination; however, the mechanical stage must allow for a manual 360º rotation if maximum mean reflectance measurements are desired.

At least two standard mirrors, whose well-known reflectances cover the range of vitrinite reflectance expected to be measured, are used for the lighting calibration. The relation between the grey level measured and the reflectance is presumed to be linear in that range after calibration unless a different mode of capture is used.

There are two possible measurements for vitrinite reflectance: maximum mean reflectance and random mean reflectance. After calibration, the sample is placed on the stage for analysis to perform the measurements. The sample is covered in steps similar to maceral analysis. For each section where a measurement will be made, the location to be measured must be scratch free, away from borders, transitions to other macerals and very reflective areas (such as pyrite) and of uniform appearance. The reflectance value of the chosen point is collected in the case of random mean reflectance; for maximum mean reflectance, the circular stage is rotated and the maximum reflectance value observed is collected. The entire procedure is repeated until at least 100 measurements have been collected. For blends of different coal ranks, it is advised to collect at least 150 points.

It is essential to state that random reflectance measurements are the more relevant ones to this work. The methodology proposed in this work does not include rotating the sample during capture, which makes the results not directly comparable to maximum reflectance measurements.

Once the points are collected, a distribution can be plotted in both cases, resulting in mean, mode, and standard deviation values. The value reported as the reflectance corresponds to the mean value from these distributions. The histogram bins can be reported as V bins, which can be read following the example: V4 class meaning 0.4%, V12 meaning 1.2%. Figure 17 shows a typical histogram from rank reflectance analysis of a sample with 1.12% random mean vitrinite reflectance and 0.08 standard deviation.

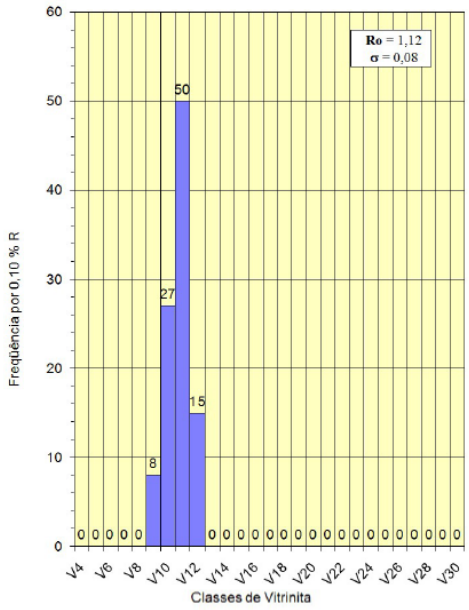

Figure 17 - Example of a typical reflectance histogram from the traditional method. Classes can be read following the example: V4 class meaning 0.4%, V12 meaning 1.2%.

The reproducibility of this method is defined by the maximum difference expected from two measurements made on the same sample by different operators in different labs and equipment on different days. The reproducibility reported in the standard is 0.06% reflectance. More details on this method can be found in the standard ASTM D2798.

3.7 Deep Leaning and Convolutional Neural Networks (CNNs)

Machine learning can be described as using algorithms to analyze data, learn from it, and determine or predict something related. Instead of manually coding software routines with specific instructions to perform a particular task, the algorithm can learn from the data how to solve a task without being explicitly programmed. It is considered a subset of artificial intelligence studies.

Deep learning, in turn, is a subset of machine learning (Figure 18), in which the algorithms are created at many levels, each level providing a different interpretation of the transmitted data and having a further depth of abstraction. These interconnected algorithms form an artificial neural network resembling the human brain's neural connections. Deeps learning’s most crucial characteristic is feature learning, that is, the ability to learn which features are essential for pattern recognition; traditional machine learning relies on an operator to determine which features are crucial and devise a way to extract them and feed them to the network. However, unlike a biological brain, in which any neuron can connect to any other neuron at a certain physical distance, these artificial neural networks have different layers, connections, and directions of data propagation. This

characteristic is responsible for deep learning methods' consistent and rapid rise in use for image identification and classification worldwide.

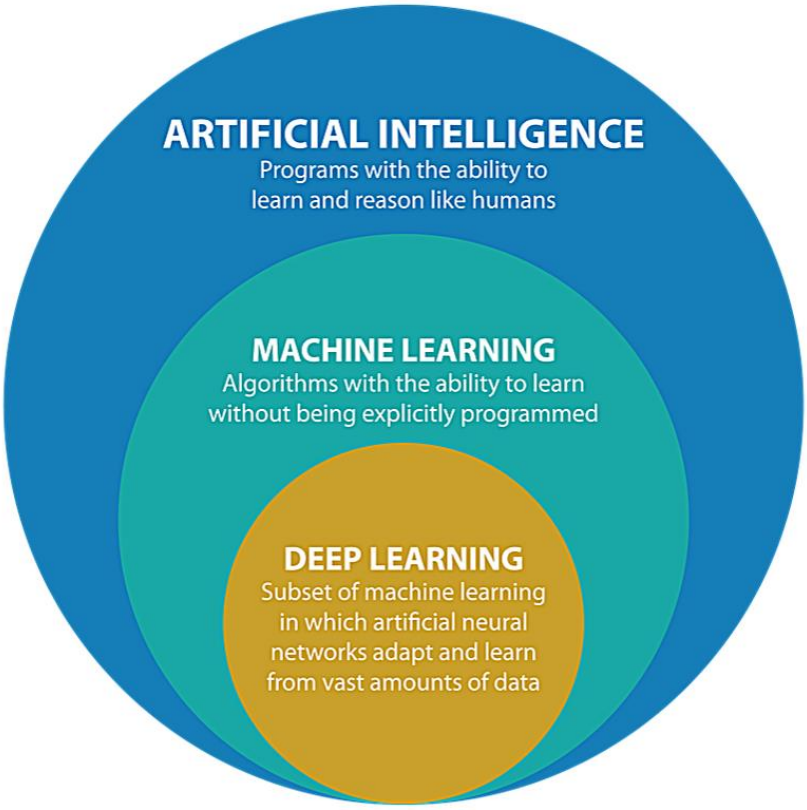

Figure 18 - Artificial intelligence, machine learning, and deep learning (CLAIRE, 2019).

The central nervous system inspires the idea for an artificial neural network, which is composed of mathematical functions responsible for extracting or analyzing features fed to it. These mathematical functions are called artificial neurons (Figure 19). These neurons are, therefore, the basic units in an artificial neural network.

Neurons located at the beginning of the neural network receive input directly from the analyzed data, whereas every other neuron receives input from previous layer neurons. Each input received is multiplied by a particular value, called weight (w), which establishes the connection between any two neurons. Each neuron sums each multiplied input it receives and applies an activation function, resulting in an output fed to one or more of the subsequent layer neurons. When properly trained, that means having appropriate weights, the output given by the last layer neurons can be used to solve the problem it was trained for (IGLESIAS et al., 2019). This can be seen in the following equation:

$$\sum_{i=0}^{n} w_i x_i + b \tag{2}$$

where x_i would be the input received from the previous neuron, w_i would be the weights, and b, the bias. (IGLESIAS et al., 2019)

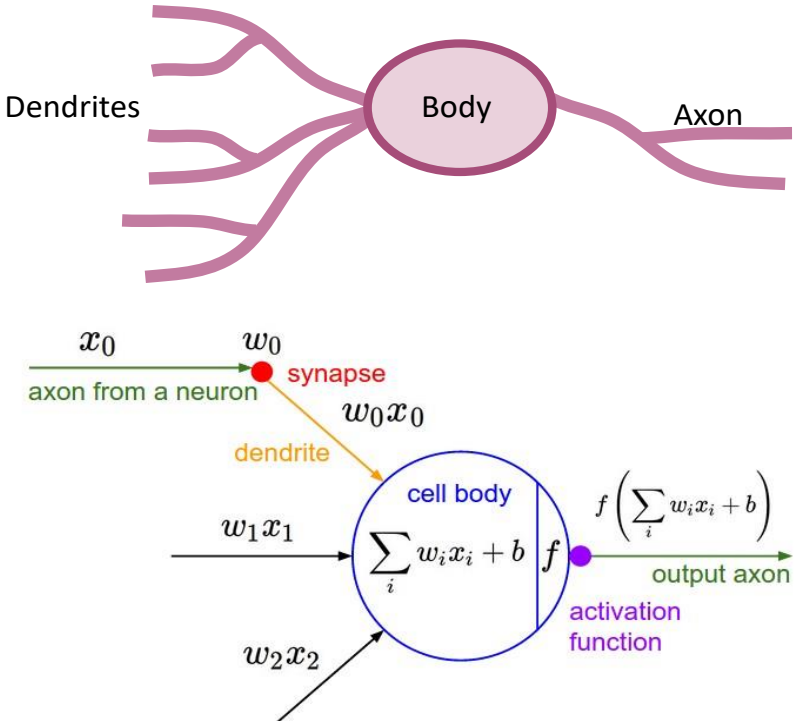

Figure 19 - Biological model of a neuron and its mathematical equivalent. w is the weight connecting the neuron to one of the previous neurons, x is the input received from the last neuron, and f is the activation function that results in the output of the neuron.

This work's network training approach is called supervised training. For this approach, it is initially necessary to create a dataset consisting of delineated objects with their respective classes properly labeled by a specialist. These annotations are traditionally called ground truth; these images are split between training and test sets. The first is used to train and update the network weights, while the second is used merely to validate and assess the model's performance on images that have not been used to train the network. Hence the term supervised.

Most deep learning architectures for image analysis problems involve multiple convolutional layers called convolutional neural networks (CNNs). This architecture can receive images as input and doesn't require pre-processing to extract features. In this case, x_i (equation (2)) becomes the pixel values of the previous matrix. The convolution operation can be written as:

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da \tag{3}$$

where the function x is referred to as the input, w is the convolution kernel, and the output s is sometimes referred to as the feature map (IGLESIAS et al., 2019). A visual representation of this operation can be seen in Figure 20, as the kernel "slides" across the image while multiplying with their pixel, each step and multiplication generates a single pixel value seen in the resulting image to the right.

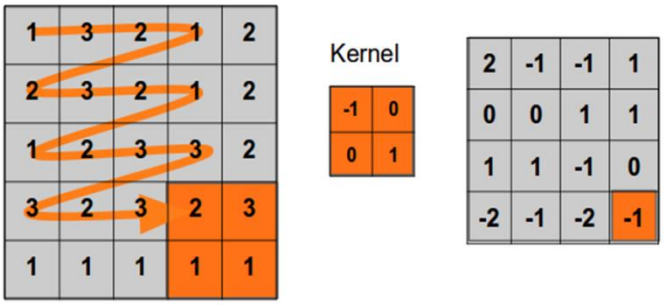

Figure 20 - Mathematical and matrix representation of the convolution operation involving two images.

Each neuron in a CNN has a random values kernel at first associated with it, equivalent to the weights from traditional networks, which means that at first they extract random features. These kernels sweep the input image resulting in a new matrix. As for the standard multi-layer networks, a non-linear activation function is applied to the numbers resulting from the sweep. Otherwise, the result of the network output would be linearly dependent on the input data, preventing the network from solving more complex cases. The typical recommendation nowadays is to use the ReLU (Rectified Linear Unit) function (JARRETT et al., 2009), as it maintains a relatively simple relation (not computationally heavy) while removing the linear dependency by making any negative values have a zero output (Figure 21). It is defined by the equation (4):

$$g\,(z) = \ max\,\{0, z\} \tag{4}$$

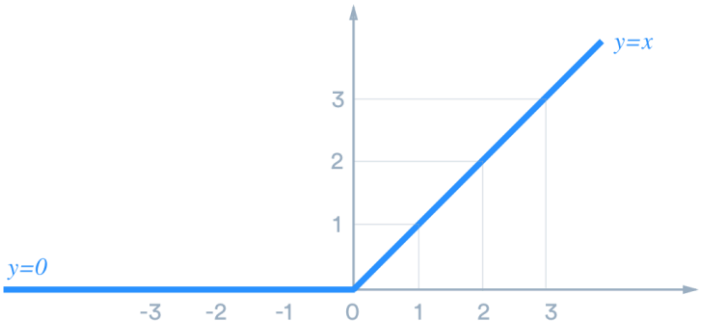

Figure 21- Rectified Linear Unit (ReLU) graphical representation.

The kernels can be understood as feature detectors and the resulting image as a feature map, which identifies the areas in the original image containing the kernel's feature (areas containing the feature have brighter pixels). In traditional digital image analysis, this can be understood as applying a filter to the image to highlight a feature, such as a border (Figure 22).

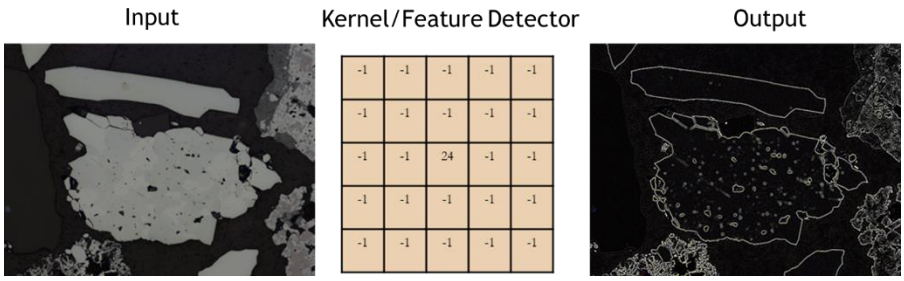

Figure 22 - Feature detector used to highlight borders in the original image.

Convolutional layers are usually intertwined with subsampling layers called pooling. This pooling aims to reduce the number of parameters, and redundancy and add a degree of spatial variance to the process. It also converts all the values of a block of size n by n, along its spatial dimensions, into a single pixel with a value that can be the minimum (min-pooling), the maximum (max-pooling), or the average (average-sharing) of the block values (Figure 23). The output image will be n times smaller across both spatial dimensions. Since higher pixel values in a feature map indicate the presence of the feature, the most common type of pooling is max-pooling.

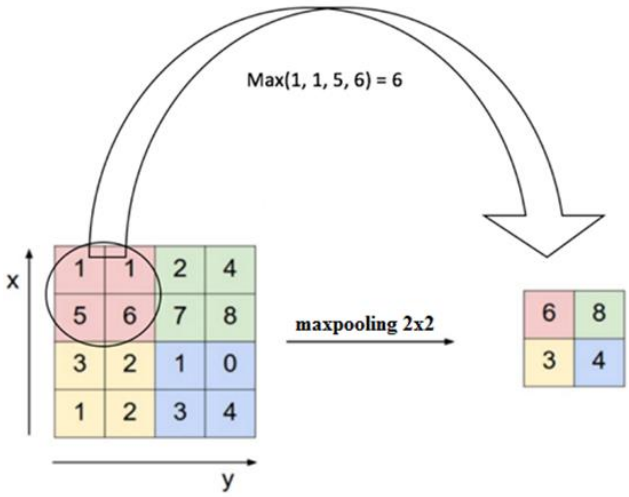

Figure 23 - Max-pooling operation applied to a matrix and its result.

The disposition of the convolutional and pooling layers depends on the architecture used, and they are usually applied multiple times. Each convolutional layer can have hundreds of neurons, and coupled with pooling layers, they make up the feature extraction portion of a convolutional neural network.

A generic chart of this conventional CNN can be seen in Figure 24.

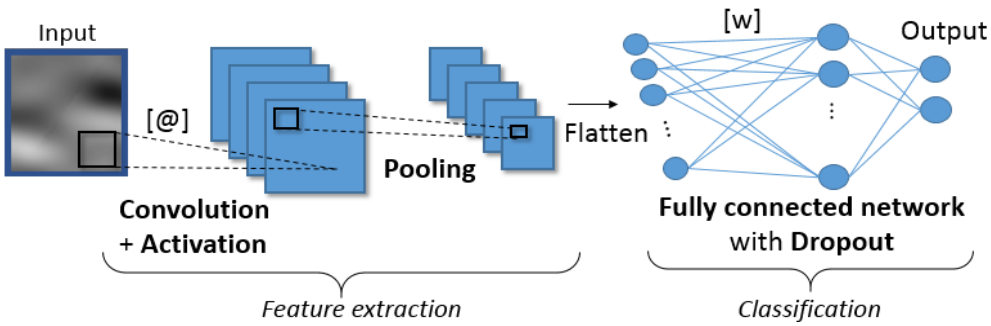

Figure 24 - Generic architecture of a CNN with its feature extraction and classification sections.

The extracted features are then fed to the classification portion of the network (Fully Connected Network in the case of Figure 24), which varies depending on the architecture used. The Mask R-CNN (Regional Convolutional Neural Network) section will describe that in more detail.

However, for a network to make a correct classification, it needs to have the appropriate weights in all of its layers to extract the useful features and make the correct classification. A supervised training achieves this by having a databank with previously classified images to train the network. The network is initialized with random weights and is run through the databank. A loss function is used to measure the error, in other words, the distance between what the network does and what would be the correct answer as a function of the weights in the network. A popular loss function is the cross-entropy loss function:

$$Cross - Entropy\ Loss\ Function = -\sum_x y(x)\log\big(p(x)\big) \tag{5}$$

Where y(x) is the binary class label, which results in 1 if the correct class, 0 otherwise, and p(x) is the probability of each class.

Employing a minimization technique (like gradient descent) and a back-propagation algorithm (RUMELHART et al., 1986), the weights of both the convolutional and fully connected layers are updated to reduce the error. This process is called Backpropagation. The adjustment of errors in the CNN during the learning process occurs in both portions of the network: in the feature extraction layers (convolution), where filters' weights are adjusted, so the network learns which features are meaningful to the task; and in the classification layer, where the weights are changed, so the network learns how to correctly segment/classify the image based on the extracted features.

The gradient descent is performed based on the cost function gradient ($\frac{\partial C}{\partial w_{ij}}$). The weights ($w_{ij}$) are updated to decrease the loss function output by increasing or decreasing accordingly, with a learning rate (α) (KIEFER and WOLFOWITZ, 1952):

$$w_{ij}^{new} = w_{ij}^{old} - \alpha\frac{\partial C}{\partial w_{ij}} \tag{6}$$

The learning rate is responsible for the magnitude of the weight update, also called step size ($\alpha \frac{\partial C}{\partial w_{ij}}$). It is usually adaptative, meaning that it is bigger at the beginning of the training and becomes smaller as the training advances. The bigger learning rate, in the beginning, allows for a quicker convergence of the weight, while the smaller step size allows for a final fine-tuning around the minimum value of the Cost function. An example of the process can be seen in Figure 25.

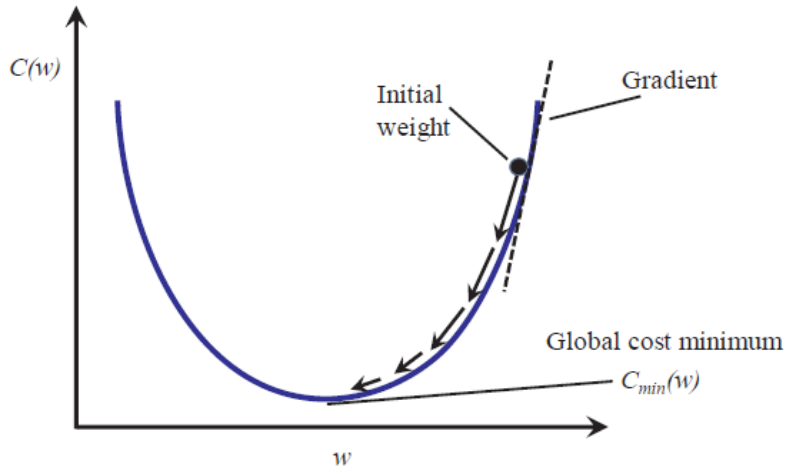

Figure 25 - The steepest descent algorithm for minimizing loss is responsible for the network's weight/learning convergence (IGLESIAS et al., 2019).

However, in a real case scenario, the network might get "stuck" in a local minimum where the error is still very high. For that reason, the weights are randomly initialized, as a new initial weights configuration would allow the weights to converge to a new minimum (Figure 26).

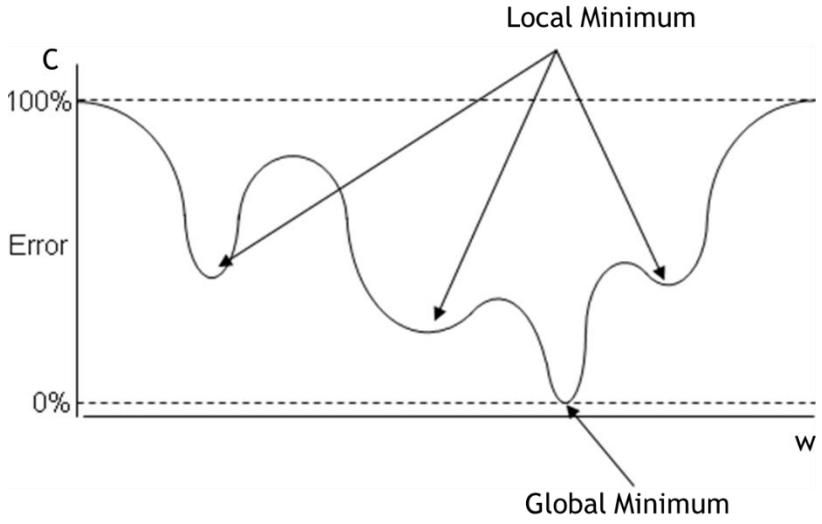

Figure 26 - A 2D example of how the weight space can have multiple of local minima, which will not provide the lowest possible error.

Another problem is updating the weight an appropriate amount of times. If the training is stopped too early, the model won't be complex enough to represent the data (underfitting). If the training is stopped too late, the model will be more complex than the data suggests, being more likely to fail when dealing with new data (overfitting). Figure 27 shows an example of underfitting, or fitting and an adequate fitting (KIEFER, and WOLFOWITZ, 1952).

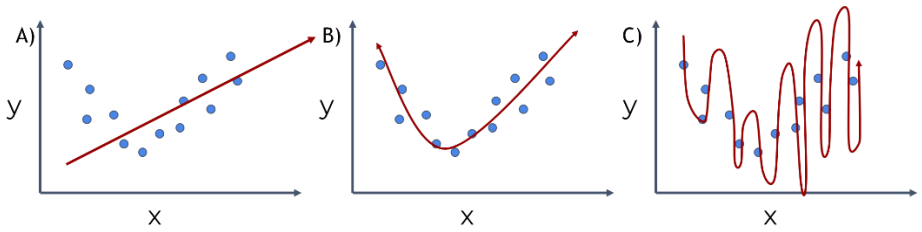

Figure 27 – A mathematical representation of model (red line) attempting to fit a dataset (blue dots). A) shows the case for underfitting (model too simple), B) shows the case for an adequate fitting, C) shows the case for overfitting (model too complex).

One solution is to split the training set each epoch into a further cross-validation set and actual training set. By assessing the errors in both sets at the end of each epoch, it is possible to know when to stop the training. By stopping the training when the cross-validation error starts to increase, the model will have been trained an adequate amount of epochs. This method for avoiding under- or overfitting is called early stopping (Figure 28) (KIEFER and WOLFOWITZ, 1952).

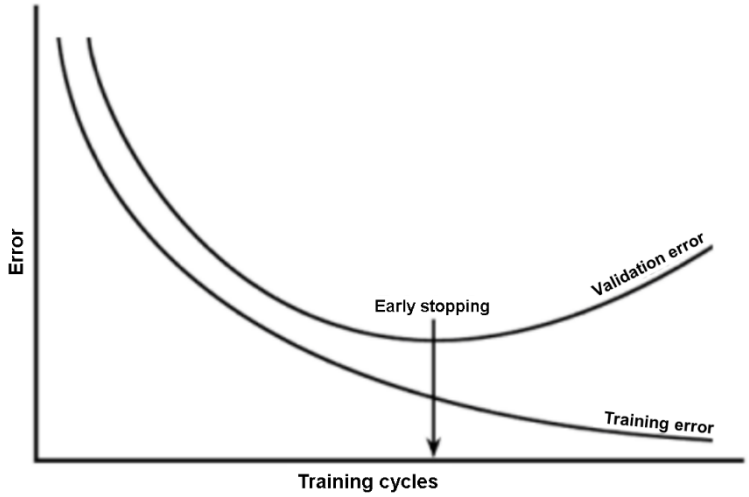

Figure 28—Early stopping method. By determining the number of training cycles after which the cross-validation error occurs, both underfitting and overfitting are avoided.

The last layer of neurons outputs numbers related to the probability of each class. For these numbers to reflect an actual probability, they are normalized with a softmax function (7):

$$softmax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (7)$$

where Z_i is the output of each of the last layer neurons.

The role of the softmax function can be seen in Figure 29.

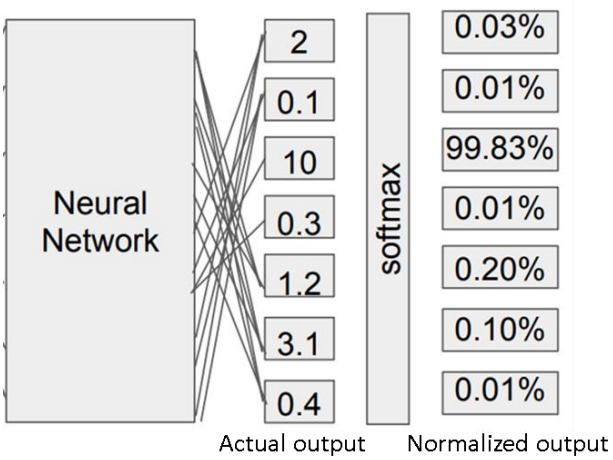

Figure 29 - Example of the softmax converting the network output into a probability output.

The error function slowly converges to a minimum each time the network runs through the databank (an epoch), and all its weights are updated to minimize the loss function. That ensures that the network slowly learns how to extract features relevant to the problem, and then use them for the classification/segmentation/detection. However, to achieve better results it is necessary that the training set is sufficiently large and representative of the actual case scenario the network will use later. The test set can then be used to validate these results.

Neural networks, however, are not limited to the classification of images. Depending on how the network is built, it can offer different kinds of output; the four main image interpretation algorithms (despite it not being limited to these) are as follows (Figure 30) (BHARATI and PRAMANIK, 2019):

- Classification algorithms recognize an image's content and do not offer the possibility of locating the detected element, in other words, make a single decision for the whole image. However, they are swift and efficient.
- Object detection algorithms are trained to locate objects and identify classes simultaneously. The location is approximated by a “bounding box,” a rectangle with two coordinates (top left and bottom right), significantly simplifying the location process.

- Semantic segmentation (CORDTS et al., 2016) algorithms classify each image pixel as belonging to a class, thus generating Thematic Maps. These mask algorithms present objects drawn in polygons and can more accurately detect the presence of the element or not. In technical terms, they are a second-class specialization through segmentation algorithms. Naturally, as they have more free parameters to adjust, they require a more extensive training set and significantly greater computational power.
- Instance segmentation algorithms are trained to locate, classify and segment objects.

This work aims to train a network that not only classifies images but provides as output the original image with individual objects from different classes correctly identified and segmented, also known as instance segmentation. Thus, a more sophisticated CNN is needed; the algorithm chosen for this work is called Mask R-CNN (HE, et al., 2018), which is a state-of-the-art network architecture for instance segmentation.

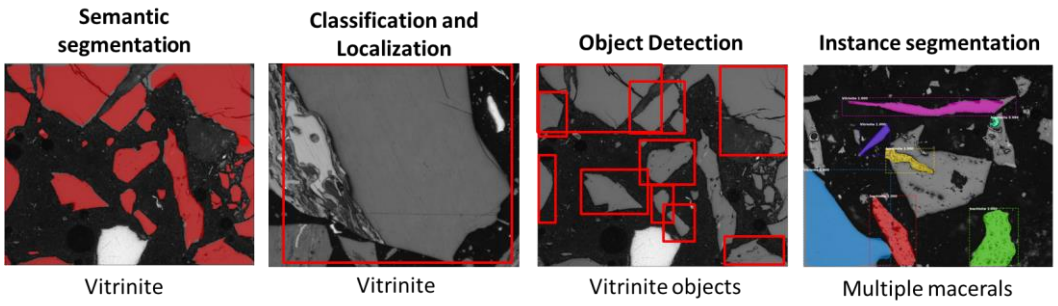

Figure 30 - Examples of semantic segmentation of pixels from objects, classification, and localization of an object in an image, object detection, and instance segmentation.

3.8 Mask R-CNN

A Mask R-CNN (HE et al., 2018) is a very elaborate, recent architecture of neural network that is currently the state-of-art for instance segmentation. Mask R-CNN, Faster R-CNN (REN et al., 2018), and Fast R-CNN (GIRSHICK, 2015) are all derivatives of the original R-CNN (BHARATI and PRAMANIK, 2019) (Region-based Convolutional Neural Network). R-CNN received its name because instead of offering an input image directly to a convolution layer for feature extraction, it uses a region proposal algorithm that narrows down the relevant areas of the input image for convolution. Fast R-CNN implemented changes in the training algorithm that improved both speed and accuracy of results by jointly learning to classify object proposals and refine their spatial locations.

Faster R-CNN improved that even further by fusing the convolutional and region proposal stages into a single stage called RPN (Region Proposal Network). All of these are dedicated to object detection. Mask R-CNN (Figure 31), in turn, adds a mask segmentation branch parallel to the established bounding box and class recognition branch from Faster R-CNN.

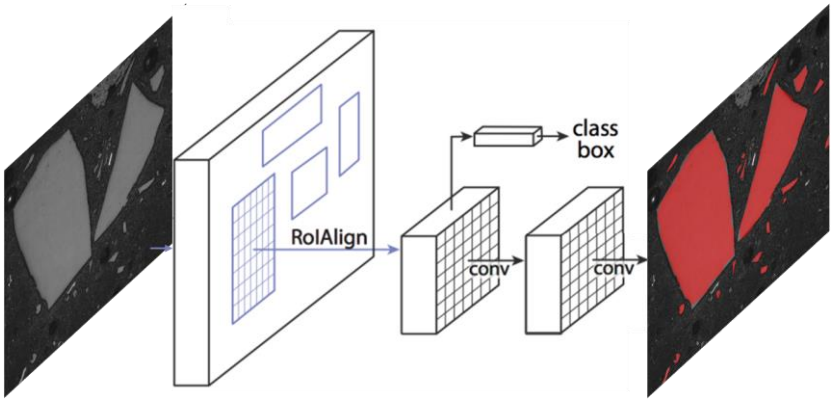

Figure 31 - Generic architecture of a mask R- CNN, feature extraction, RoI proposal, and instance segmentation sections (HE et al., 2018).

Mask R-CNN can be understood as consisting of two different stages. The first one scans the image and proposes possible bounding boxes for identified objects (Regions of Interest or ROIs), which are then refined and fed to the second stage. The second stage analyzes the individual bounding boxes and can choose which pixels belong to the object and which ones belong to the background, with the idea that a bounding box contains only a single whole object, even when it overlaps with a different one (HE et al., 2018).

These stages can be subdivided into smaller modules discussed in the following.

For the first stage, the first module of the network is called the backbone. It consists of a standard convolutional layer, like previously discussed, whose objective is to extract features from the input image. The early layers detect low-level features (edges and corners), and later layers detect higher-level features. A traditional convolutional neural network may be chosen to act as backbone.

The backbone may also receive an additional extension called Feature Pyramid Network (FPN) (Figure 32). It is an additional pyramid of CNN, whose convolutional layers receive input from each convolutional layer of the backbone. This allows the network to access features of all levels, improving its ability to deal with objects at different scales. Therefore, instead of the output being a single feature map, it is now a set of feature maps for each level of the original backbone; the map used is chosen dynamically depending on the object's size (LIN et al., 2017).

Figure 32 - Feature pyramid network schematic. It is responsible for extracting different level features from the initial convolutional layers (LIN et al., 2017; REN et al., 2018).

This work employs a ResNet 101 (101 convolutional layers) in its backbone. A high number of convolutional layers allows for more abstract features to be appreciated and better performance in complex problems. A high number of layers, however, makes the training slower as the loss must be back propagated to the beginning of the network, because with the gradient vanishing, there is a loss of magnitude in the weight updates. A ResNet CNN avoids this problem by having “shortcuts, " in which the loss from deeper layers can be directly propagated back, allowing more than 100 layers to be trained efficiently (Figure 33).

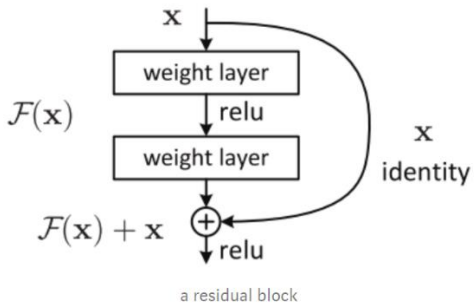

Figure 33-Schematic of the shortcuts for loss backpropagation

The second module is a Region Proposal Neural Network (RPN), which scans the selected backbone feature map and creates a series of regions, called anchors, that may contain objects (Figure 34).

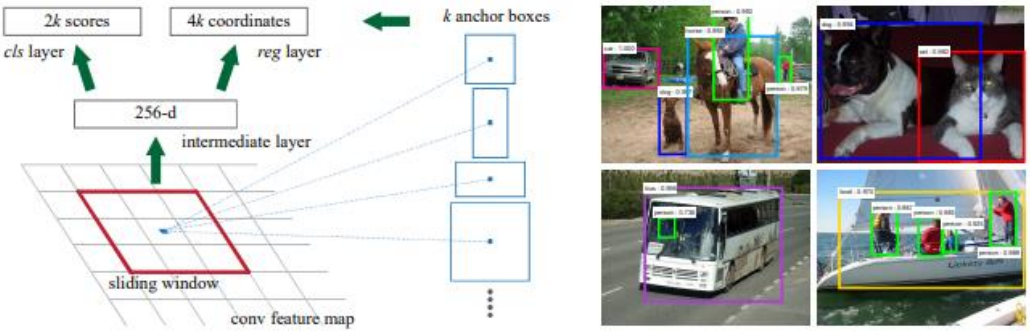

Figure 34 - RoI detection schematic. This network section scans the image and proposes bounding boxes for probable objects (GIRSHICK et al., 2014, 2015; HE et al., 2018).

The following module is the bounding box classifier, which analyzes the individual bounding boxes, and assigns a class to them or recognizes it as background (HE et al., 2018; REN et al., 2018).

The last module is the segmentation mask, which analyzes each of the bounding boxes and segments the whole object that is supposed to be inside them (HE et al., 2018; REN et al., 2018).

To the author's knowledge, this is the first time this particular architecture has been applied to maceral analysis and rank reflectance determination.

3.8.1 Metrics

Metrics are the way to assess the performance of a trained network. Most of them compare the results to the ground truth. The ground truth can be defined as a reference with the expected correct result. This section will present a few of the more common metrics in problems of classification and location of objects.

3.8.1.1 Confusion matrix

The confusion matrix allows for a quick analysis of an algorithm's performance. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in a real class or vice versa (Figure 35). The name of the confusion matrix derives from the fact that it makes it easy to see if the system is mixing up two classes.

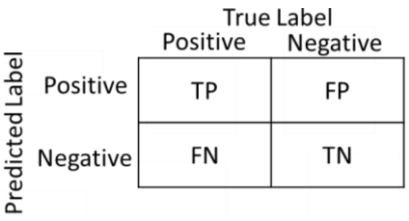

Figure 35 - Confusion matrix.

The meaning of the confusion matrix labels are:

-TP (True Positive) when the model's prediction is positive, and the ground truth is also positive;

-TN (True Negative) when the model's prediction is negative, and the ground truth is also negative;

-FP (False Positive) when the model prediction is positive, and the ground truth is negative;

-FN (False Negative) when the model prediction is negative, and the ground truth is positive.

In Figure 36a, we can see an example of TP, as both the ground truth (red rectangle) and the prediction (green rectangle) coincide. In other words, the RoI predicted by the model coincided with the appointment of the specialist. Figure 36b and Figure 36c are examples of FP. In the first case, the prediction does not match the ground truth, and in the second case, there is no ground truth. However, the model makes a prediction. Figure 36d is an example of FN, in which the model could not make a prediction, with the image having a ground truth.

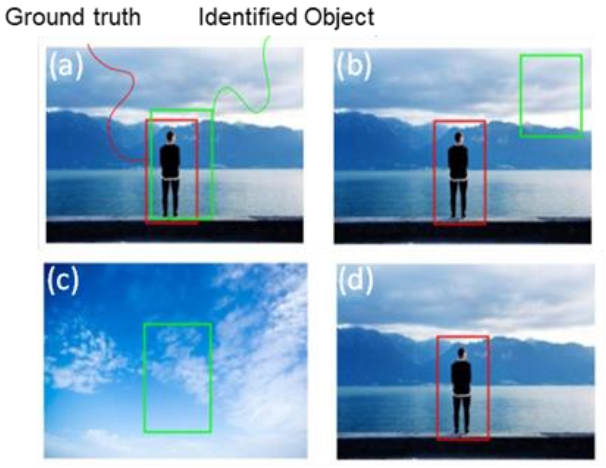

Figure 36 - Examples of TP (a), FP (b and c), and FN (d) in object detection.

An example of TN was not shown since it would be the case where the image lacks ground truth and prediction. In addition, when calculating the object detection metrics discussed in the following, true negatives are not necessary..

In Figure 36a, we can see that a prediction doesn't always (in fact, rarely ever) match the ground truth perfectly. Therefore, it is usually necessary to evaluate how good of a match they are, so the prediction can be considered correct (TP) or wrong (FP).

3.8.1.2 Intersection over union

IoU (Intersection over Union) is an essential concept in object detection problems, also referred to in the literature as Jaccard's Index. IoU consists of an evaluation metric for detecting objects in images, which assesses the similarity between the ground truth and the RoI inferred by the model. The closer this metric is to 1 indicates that the inferred RoI is closer to the real object in the field.

Thus, the IoU is the ratio between the intersection and the union of ground truth with the prediction made, as shown in Figure 37.

$$IoU = \frac{segmented\ object \cap ground\ truth}{segmented\ object \cup ground\ truth} \quad (8)$$

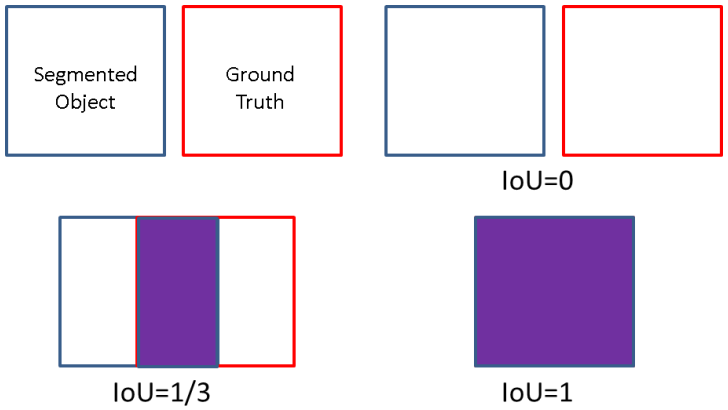

Figure 37 - IoU = Ratio between the intersection and the union of ground truth with the prediction made.

This metric is more critical during the network training than during the test per se, as only predictions achieving a minimum pre-established IoU can be considered correct for error estimation and weight updates. IoU is usually used during test to confirm whether a prediction is a TP, an FN or an FP, as explained above (The most common value being 0.5). However, due to the complexity of the macerals involved in this work, it is nearly impossible to annotate them as objects consistently. In other words, their actual border during delineation can be pretty arbitrary or subjective.

Another critical point is that maceral composition is reported in terms of area fraction and not the number of objects, as the size and relevance of these

objects would vary quite a lot. Consequently, the following metrics were calculated regarding the total area segmented, not individual objects. That bypassed the need to use the IoU to validate the TPs.

3.8.1.3 Precision

The precision indicates how much the objects detected by the models belong to or approach the classes specified in the test data, that is, out of all detections made, precision quantifies how many were correct:

$$Precision = \frac{TP}{TP+FP} = \frac{area\ of\ the\ objects\ detected\ correctly}{all\ detections\ made} \quad (9)$$

3.8.1.4 Recall

Recall, also called sensitivity, is the rate of true positives in the model, which measures the probability of objects belonging to the ground truth being correctly detected. In practice, it indicates the relationship between the existing objects in the ground truth that were correctly detected, that is:

$$Recall = \frac{TP}{TP+FN} = \frac{area\ of\ the\ objects\ detected\ correctly}{all\ ground\ truth} \quad (10)$$

Where false negatives (FN) are the ground truth annotations not detected by the model, thus, the recall is inversely proportional to the amount of FN.

It is essential to highlight the difference that exists between recall and precision. A model can have outstanding precision, indicating the suitable class, but recognize few objects or "detect" few elements. Depending on the problem, one might be more important than the other.

3.8.1.5 F1 - score

F1-score simply evaluates both aspects of recall and precision and can be mathematically defined as the harmonic mean between the two:

$$F1 - score = \frac{2\times(Recall\times Precision)}{Recall+Precision} \quad (11)$$

3.9 Previous works on Maceral Analysis and Instance Segmentation

Due to the commercial interest in coal, many attempts have been made in the past decades to automate coal analyses, saving labor and time. One of the first attempts to automatize the process was by CHAO et al. (1982). They were the first to implement a form of digital microscopy that allowed the computer to capture the selected points and register its maceral label (according to the operator's choice) and its reflectance based on the calibration of the microscope. They were then able to group those points in bins and build the first automated reflectance histograms, which became the basis of many other reflectance-based approaches for maceral group identification. Similarly, KUILI et al. (1988) also worked on an automatic “fingerprint” histogram collection for maceral group composition and vitrinite reflectance.

Many attempts have been made at automating maceral analysis throughout the years; one of the earliest attempts was made by AGUS et al., 1994, in which optical microscopy was used to extract three different texture parameters automatically:

- Angular second momentum (ASM) - Measure of image homogeneity (grey tones transitions).
- Contrast (CON) - Measure of local variations present in the examined area.
- Correlation (COR) - Measurement of linear grey scale dependencies in an image.

They were measured in areas of the image identified as belonging to a single maceral in an attempt to characterize them (Figure 38).

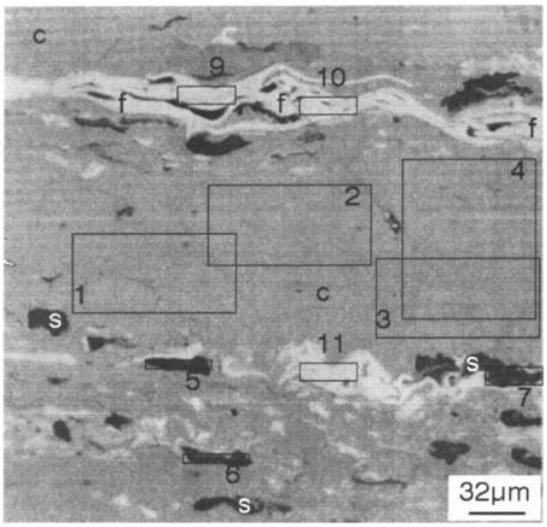

Figure 38 - Areas were manually chosen to extract the texture parameters; chosen to encompass only a single maceral (AGUS et al., 1994).

After plotting the results for the parameters for the macerals collotelinite, sporinite, inertodetrinite, and semifusinite (Figure 39), they were able to verify

that there is indeed a texture difference between them, even though the standard deviation didn't allow for consistent discrimination (AGUS et al., 1994).

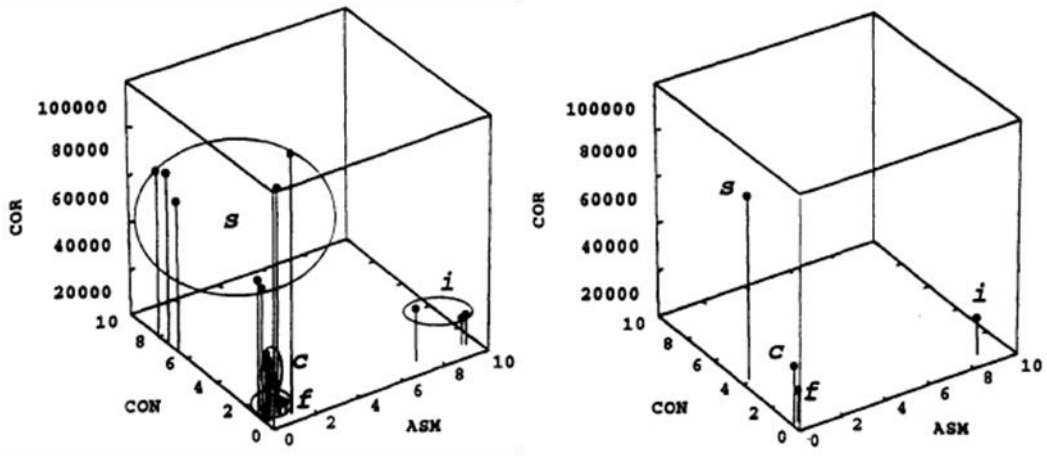

Figure 39 - 3D graph plotting the three texture parameters: Angular second momentum (ASM), Contrast (CON), and Correlation (COR) for the macerals semifusinite s, collotelinite c, fusinite f, and inertodetrinite i. To the left the dispersion of the measurements is shown, while to the right, the mean measurement of each group was plotted (AGUS et al., 1994).

This was one of the first works to conclude that only a system capable of considering a wide array of abstract features would be able to characterize the maceral analysis problem successfully. As the neural network does precisely that, it lends credence to the proposal that it could be used to solve maceral classification.

Another attempt at automating the maceral composition and rank reflectance determination was (O'BRIEN G. et al., 2003). It took advantage of the different reflectance ranges for the three main maceral groups to classify them. Two hundred images from 40 different coals ranging from 0.48% to 2.13% mean vitrinite reflectance were used. By collecting random reflectance information from coal sample images pre-processed to remove resin and plotting it against their cumulative frequency, the authors obtained a full maceral reflectogram (FMR), which allowed them to estimate the amount of the three different groups and isolate the pixels referring to the vitrinite group, which allowed for an analysis of the vitrinite reflectance. It is understood that a change in slope in the FMR shows the transition between classes, as it reflects the three different population in the reflectance distribution. Liptinite, however, can't be discriminated from mineral matter using this approach due to overlapping reflectance (Figure 40).

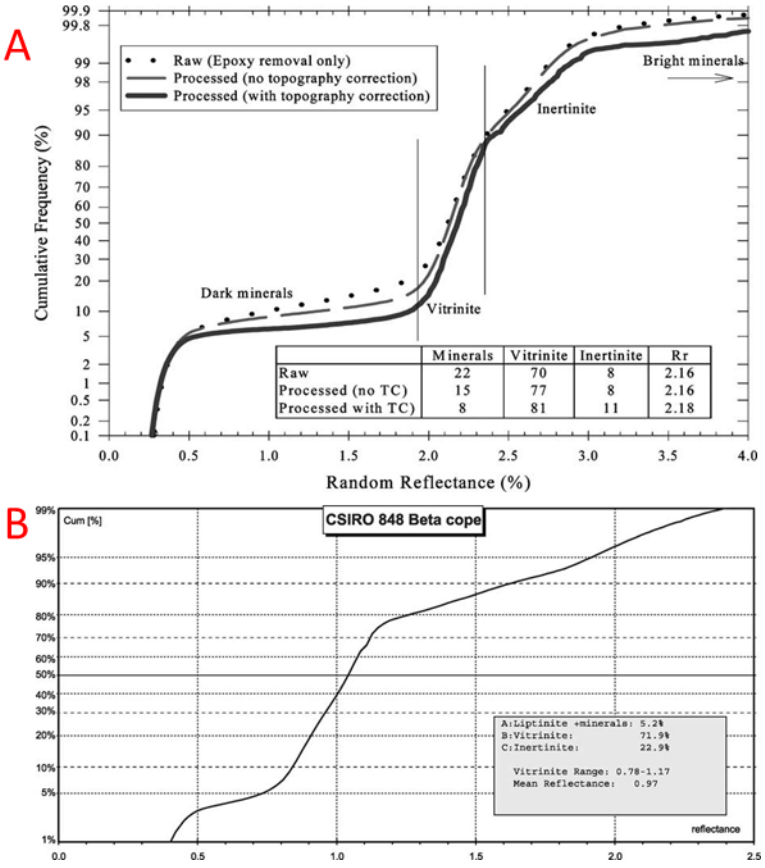

Figure 40 - Full maceral reflectogram (FMR) correction for maceral composition determination (A). Corrected FMR for a 0.97% reflectance sample(B). Adapted from (O'BRIEN et al., 2003).

Even the first works mentioned here already faced many of the same challenges faced by the later ones, including i) the reflectance distinction between the three maceral groups is more pronounced in lower rank coals and becomes harder to resolve as they converge in higher ranks; ii) some degree of overlap in the lower and higher reflectance ranges of the maceral groups, and iii) difficulty to distinguish liptinite macerals from some minerals, as liptinite shares its reflectance, and sometimes even shape, with some dark minerals. Most of these issues would require a human operator to intervene, either correcting the reflectance ranges of the maceral groups visually or verifying if the images chosen are suitable to have their texture parameters extracted. These issues prevented these earlier methods from avoiding operator intervention entirely and becoming fully automated.

The works mentioned so far seem to have shared two main ideas for approaching maceral classification: i) Using reflectance values to determine maceral groups (CHAO et al., 1982; KUILI et al., 1988; O'BRIEN et al., 2003; PETERSEN et al., 2013) or ii) using pre-defined texture parameters (AGUS et al., 1994; CHAO et al., 1982), sometimes alongside reflectance as a further aid to distinguish maceral groups. Machine learning-based methods have the advantage of naturally exploring both approaches, built upon a self-learning algorithm. Their

most significant advantage lies in the sheer number of image features they can extract and work with (ranging from texture, shape, and size to color/grey value features). A combination of such features is what allows these methods to solve highly abstract problems (HE et al., 2018). However, one main drawback of machine learning methods is their current inability to "open" the network to assess which features were learned and used. However, their state-of-the-art performance on many image-related problems has inspired their recent use in coal petrography.

A recent attempt at maceral segmentation using machine learning (TIWARY et al., 2020) used a random forest algorithm. The authors extracted features related to micro-structures and fed them to the random forest-based algorithm, which then was able to segment the three maceral groups and mineral matter. Vitrinite and inertinite were segmented to approximately 90% success rate, while liptinite had a lower success rate, which traditionally is the case for most automated maceral classification systems (Figure 41).

Instead of annotating the maceral groups, the authors relied on an experienced coal petrographer to delineate triangular areas in the images that only contained one of the four main classes: vitrinite, inertinite, liptinite, and resin. By doing that, their model was able to extract correct features that described these classes and were able to reach high success rates.

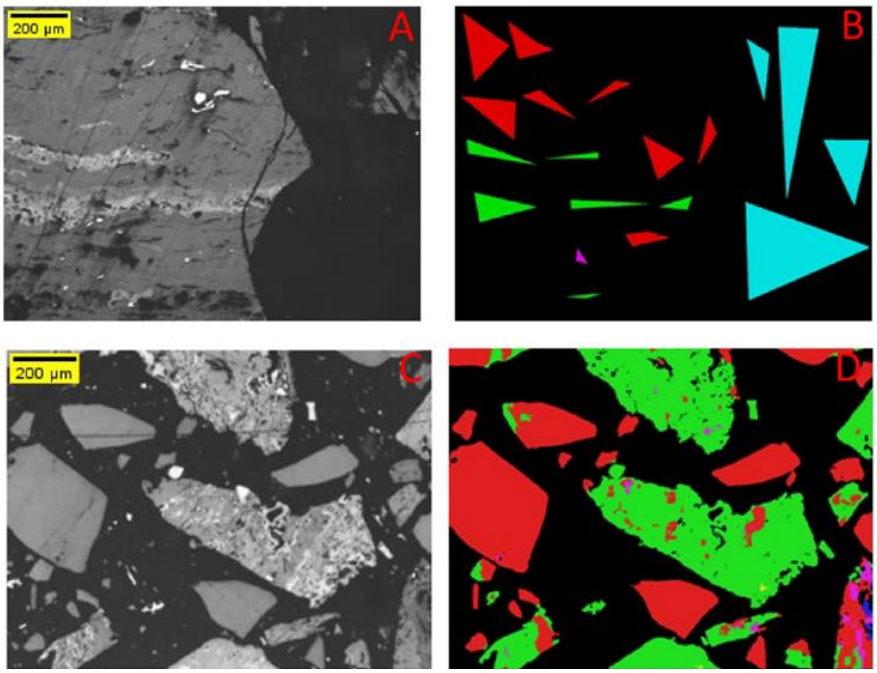

Figure 41 - Example image used for the ground truth (A). Ground truth for image A (B). Example of image for the model's test (C). Random forest models result for image C (D). Red, green, blue, magenta, and cyan represent vitrinite, inertinite, liptinite, mineral matter, and resin, respectively. Adapted from (TIWARY et al., 2020).

Another interesting attempt at maceral identification at a low rank reflectance (0.5%-0.8%) was by (IWASZENKO, S. and RÓG, L., 2021), which used a deep learning-based algorithm with a U-net architecture. With the aid of a

professional petrographer, the authors manually generated a ground truth for the three maceral groups and used them to train the network. The authors obtained success rates of over 90% for all three groups (Figure 42 and 43).

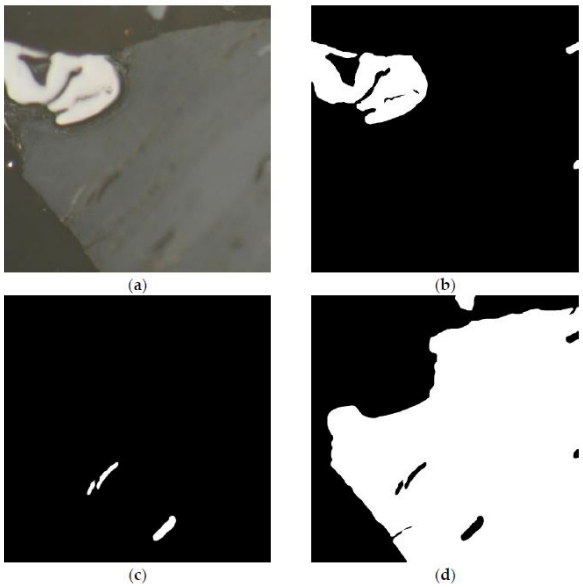

Figure 42 - Original grey level image (a). Inertinite ground truth mask (b). Liptinite ground truth mask (c). Vitrinite ground truth mask (d) (IWASZENKO and RÓG, 2021).

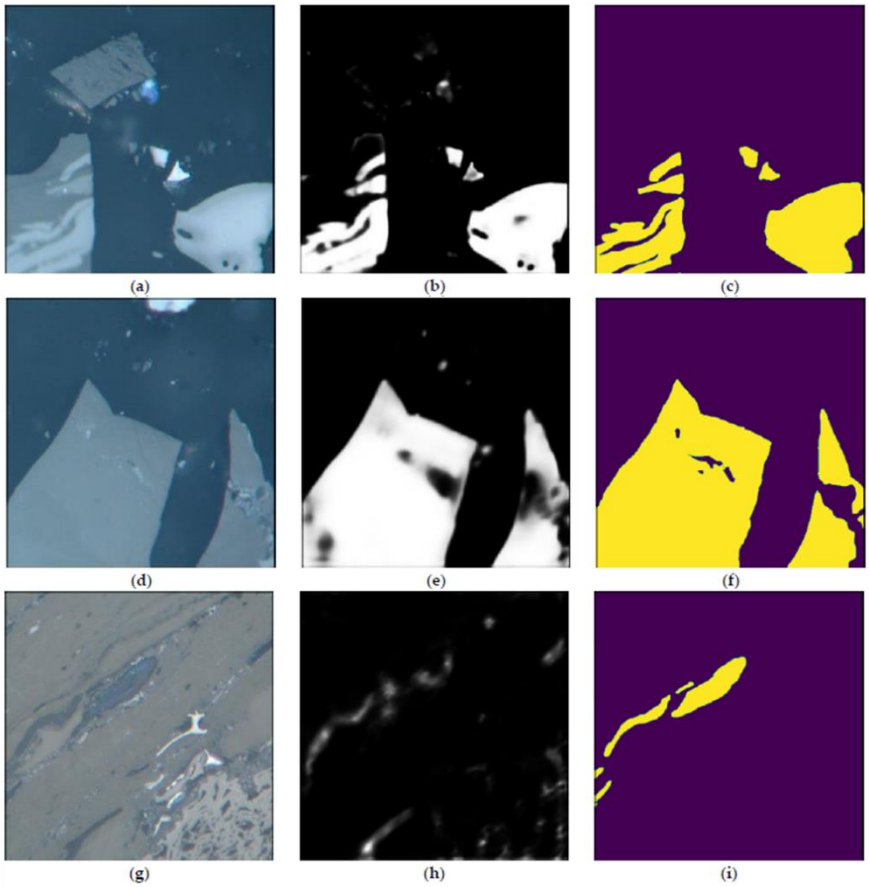

Figure 43- Input images (a,d,g).Segmentation results for inertinite, vitrinite, and liptinite, respectively (b,e,h).Ground truth for inertinite, vitrinite, and liptinite, respectively (c,f, i). Adapted from (IWASZENKO and RÓG, 2021).

WANG et al. (2019) performed an attempt at both maceral segmentation and rank reflectance determination. The authors built a dataset of 78 coal images in the 0.7%-1.79% rank range. They employed an adaptative k-means clustering to identify maceral groups and then utilized comprehensive features and a support vector machine to identify vitrinite regions. These regions were then analyzed with a random forest algorithm to extract information on the maximum mean reflectance (Figure 44).

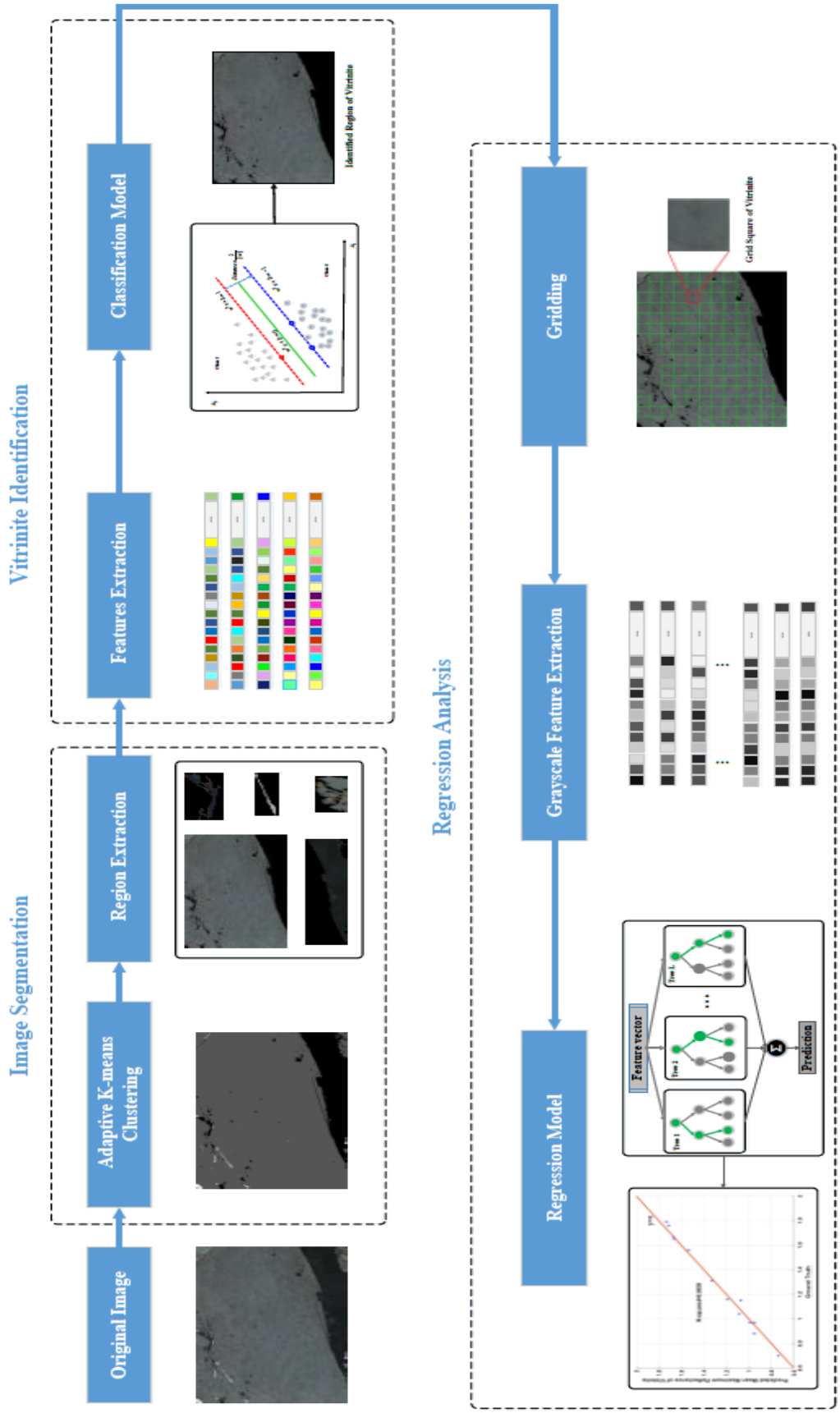

Figure 44 – Flowchart for WANG et al. (2019)'s method for maceral segmentation and vitrinite reflectance determination.

Their model reached a root mean square error of 0.0424 for the vitrinite reflectance determination, by comparing the reflectance estimated with their method and the reference values they had.

This thesis uses a single algorithm to perform classification and segmentation but works on 20X images instead of 50X. While the usage of 50X images provides their method with more detail for segmentation, 20X images allow the current approach to be more flexible with sample preparation as it is easier to have homogeneously focused images at lower magnification and provide for a faster scan of the sample.

While no work was found that uses the proposed CNN model (Mask R-CNN) for maceral segmentation, this model has found use in other areas of image analysis. Mask R-CNN (He et al., 2018) is a state-of-the-art neural network for instance segmentation, which has experienced a considerable boom since 2018 for many different applications. SALAU and KRIETER (2020), for example, were able to use a Mask R-CNN model to detect and segment cows in security camera images. They annotated the cows using the VIA software (DUTAA & ZISSERMAN, 2019), also used in this work (Figure 45). They manually delineated each cow and were able to ascribe a “standing” or “lying” feature to each cow.

Figure 45 - Annotated image of cows using the VIA software. “standing” and “lying” information was also recorded but not used in that work (SALAU and KRIETER, 2020).

They could annotate 575 images (479 and 96 for training and testing, respectively). They also proved that the dataset size is one of the most crucial points in this methodology. Once their annotation work was over, they plotted the

validation loss and their metric for different states of the dataset (Figure 46), proving that an increase in size does benefit the model.

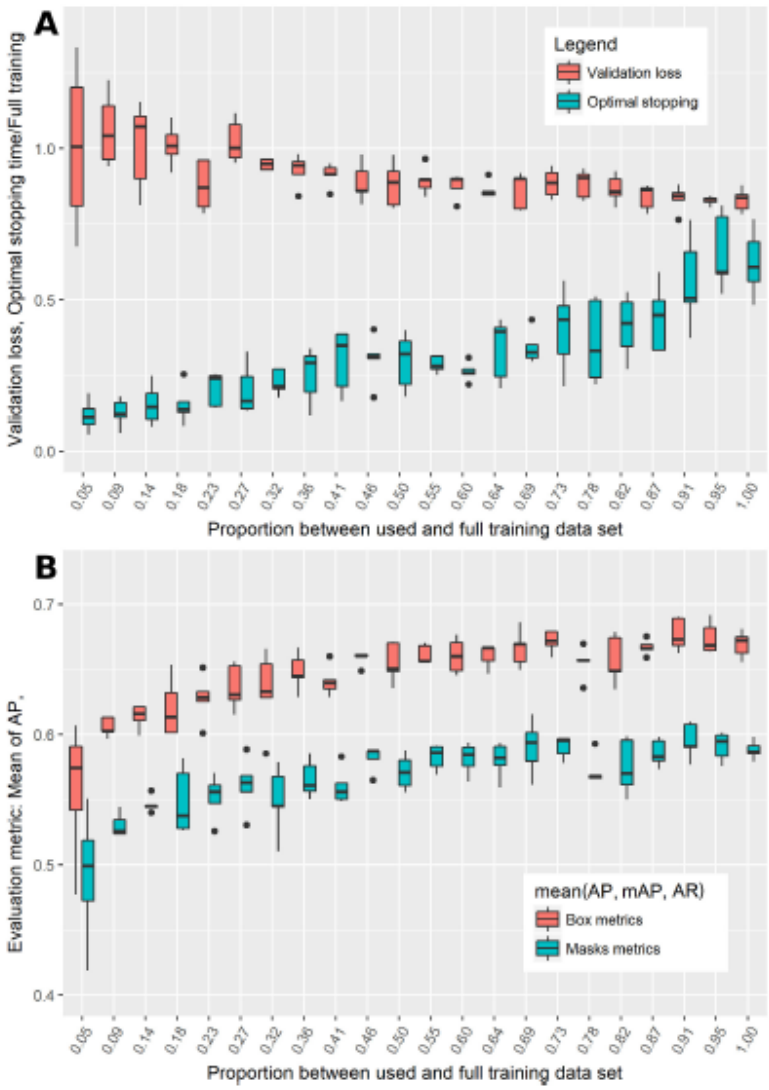

Figure 46- Validation loss and optimal stopping (A) and bounding box metrics and mask metrics (B) plotted against the proportion of the full dataset used (SALAU and KRIETER, 2020).

Some other interesting applications were HE et al. (2022), which used a modified Mask R-CNN to identify and extract oil sites from satellite multi-sensor remote sensing images and LAI et al. (2022), which attempted to use a specifically improved Mask R-CNN to correctly and precisely segment coal gangues, so that accurate shape and location information could be obtained. LAI et al. (2022) also compared such results to those obtained from YOLO v4, CenterNet, U-Net, and Deeplab v3+, concluding that the best results were obtained for their optimized Mask R-CNN. GU et al.(2022) was an extensive review of 2D instance segmentation, comparing different CNN architectures, which concluded that two-stage methods (such as Mask R-CNN), that is, those

that first detect objects and then segment them, dominate the current frontier of general instance segmentation.

This review makes the author confident that machine learning can be a powerful tool for maceral segmentation, aiding petrographers and making coal analyses more time efficient, especially the Mask R-CNN architecture. However, to the author's knowledge, there has not been a deep learning-based approach focused only on collotelinite aiming to automate its reflectance determination and no other work has attempted to use Mask R-CNN for maceral segmentation. Another huge differential of this work was the vast number of images used for the dataset, with 260 images (and over 17000 annotated objects) for five samples ranging from 0.97%-1.20%. These images were captured at random positions, allowing the dataset to represent an actual analysis.

While 260 images is usually a very small number of images for a deep learning approach, it is considerably larger than the number of images used in other related works in this area (WANG et al. ,2019; TIWARY et al., 2020; which used 78 and 162 images respectively). The reason for that is the complexity of manually preparing the dataset, which seriously hinders the number of images that can be prepared in a reasonable time.

4
Methodology

4.1 Sample preparation and image acquisition

For this study, 13 samples with random reflectance (Rr%) range of 0.97%-1.8% (Table 2) were prepared and five of them were annotated and used to train the network. They have different origins and correspond to either single seams (e.g., Samples C and D) or commercial coals (e.g., Sample B). Even though the initial interest is to develop a method for natural coals, the sample B, a commercial sample, was also included. This was done because sample B presents a considerable amount and variety of inertinites that could help train the models. However, due to a blend of different coal ranks, the test results without it will also be presented.

Table 2 - Samples, random mean reflectance values (Rr%) as received, and basin origin (* samples used for network training).

Samples	Rr%	Basin/Origin
Sample A*	0.97%	East coast USA
Sample B*	1.03%	Kuzbass, Russia
Sample C	1.03%	Bowen Basin, Australia
Sample D*	1.04%	Bowen Basin, Australia
Sample E*	1.20%	Moatize Basin, Mozambique
Sample F*	1.20%	Moatize Basin, Mozambique
Sample G	1.15%	Moatize Basin, Mozambique
Sample H	1.26%	China
Sample I	1.42%	Brookwood, Alabama
Sample J	1.47%	China
Sample K	1.50%	China
Sample L	1.51%	East Coast, USA
Sample M	1.80%	Bowen Basin, Australia

The sample preparation followed the traditional ISO 7404-2 (2009) standard; each sample was crushed and made to pass a 1 µm sieve. Each sample was mixed with the resin (Epofix) and hardener and stirred until it acquired consistency and solidified. Afterward, the sample was polished, following the same standard.

The images were captured on a Leica DM6000 optical microscope using the Diskus-Fossil program from Hilgers Technisches Buero. The microscope was calibrated to a standard with known reflectance (YAG, 0.908%) and a black standard (0.0%), and images were captured as 8-bit, with 256 grey levels. For the same lighting condition to be used for all samples, which range from 0.97 to 1.8% reflectance, it was necessary to utilize a non-linear Look-Up Table (LUT). Applying a LUT to an image is a typical image processing procedure in which the pixel values of an image are recalculated to new ones, following an equation. Suppose the lighting conditions were adjusted and the images captured with a linear LUT. In that case, lower rank images could become too dark, or higher rank images could become too bright (over or under-saturated). The Diskus-Fossil system, however, only allows for a single other LUT called enhanced.

The images were captured with a 20X objective (0.5µm/pixel resolution), though vitrinite reflectance can be carried out using 32x or 50x objectives (ISO 7404-5, 2009). The reason is that it was difficult to capture homogeneously focused images with a 50X objective consistently. Besides that, using 20X lenses allows for a faster capture of the same area, as each image covers a larger field of view of the sample.

4.2 Maceral Analysis

4.2.1 Dataset Creation

The dataset's images were annotated with the VGG (Visual Geometry Group) Image Annotator (VIA). VIA allows for a very efficient point-by-point delineation of objects in an image, allowing the operator to ascribe one or multiple classes for a single object, making it possible to record each object at maceral and maceral group level alongside its coordinates in the image (Figure 47). This annotation provides the ground truth for training and test (DUTAA & ZISSERMAN, 2019).

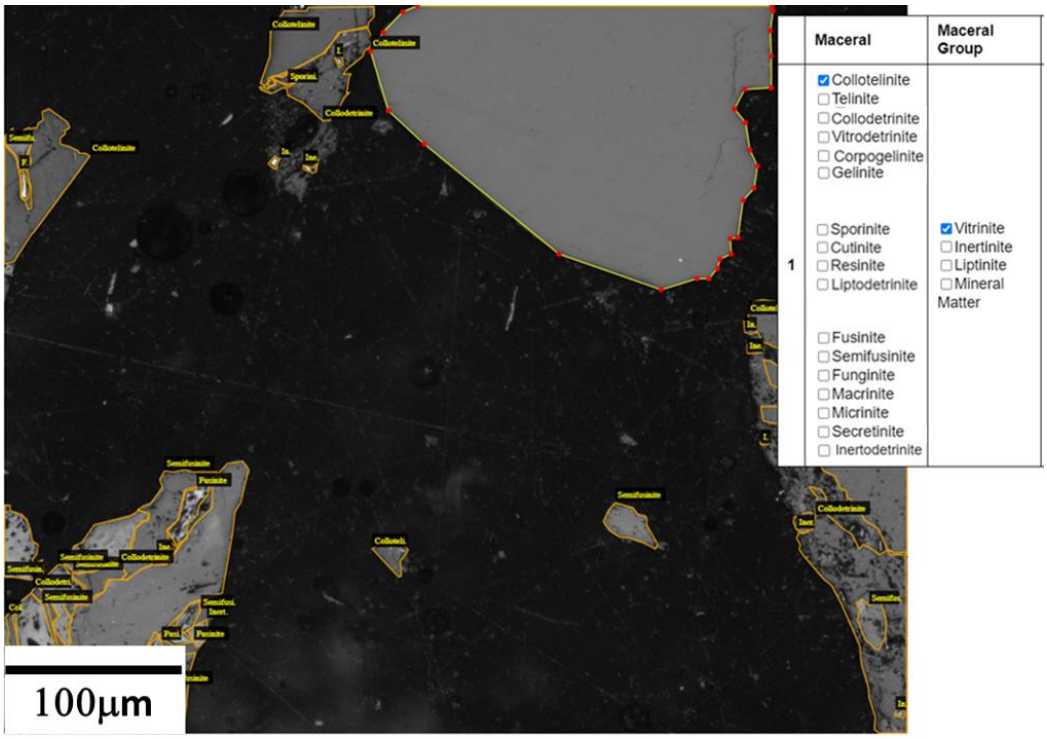

Figure 47- Example image annotated with VIA; it showcases the annotation that can be recorded with this software.

A total of 400 images were taken for each of the 13 samples presented in Table 1. For training and test of the model, 52 images for each of the following samples were taken: Sample A, Sample B, Sample D, Sample E, and Sample F, covering a rank reflectance range of 0.97%-1.2%. With the help of a certified petrographer, the 260 images were analyzed and manually delineated at the maceral level and classified at maceral and maceral group levels (according to ICCP, (1998, 1993, 2001)) and used to train the network.

The remaining 348 images for each of these five samples were used for the collotelinite reflectance determination tests alongside the 400 images for the nine samples not used in training. The total number of objects annotated in terms of individual macerals and macerals groups can be seen in Table 3.

The images were split into a training and a test set; this split was made to preserve approximately a 70%/30% ratio for the annotated objects between the training and the test sets, respectively.

Table 3 - Number of annotations made when considering the individual macerals.

Maceral Groups	Macerals	Training set	Test set
Vitrinite	Collotelinite	3037	1318
	Collodetrinite	566	317
Inertinite	Fusinite	818	396
	Semifusinite	2115	1208
	Inertodetrinite	1799	1037
Liptinite	Sporinite	3132	1611
	Cutinite	201	79
	Resinite	13	4
	Liptodetrinite	22	6

Vitrodetrinite, telinite, corpogelinite, gelinite, macrinite, funginite, and secretinite did not reach more than ten total annotations each.

The individual macerals were annotated and then classified with the aid of certified petrographer Dr. Sandra Rodrigues from the University of Queensland.

4.2.2 Generic Description of Training and Inference Parameters of the Models

The Mask R-CNN used a ResNet 101 (HE et al., 2015) as a backbone, which is an efficient and widely tested network. The following is a generic description of the parameters used to train the many models that will be presented. Any specific modification done to a model will be addressed in its description, but unless explicitly mentioned, the parameters used will be the ones described in this section.

During training, errors are calculated individually for the bounding box, the segmentation mask, and the class labeling for the mask R-CNN and the bounding box and class errors for the Region Proposal Network stage (RPN). The cost function used to calculate the mask error was the binary cross-entropy, the cost function used to calculate the bounding box error for the RPN was the L1 loss function, while the other errors were calculated with the categorical cross-entropy function (HE et al., 2022).

Training was performed with cross-validation, in which a percentage of random images of the training set is held out each epoch (in this case, 20%) to follow the cross-validation error evolution. The learning algorithm used to minimize error and update the weights was the stochastic gradient descent.

Training: a learning rate of 0.001 was used, with four graphical processing units (GPUs) and one image trained on each GPU at a time. Batch size was calculated as images per GPU x GPU count = 1 x 4 = 4. The number of steps per epoch was taken as: number of images / batch size = 174 / 4 = 43 steps. An ROI is considered positive concerning the ground truth when it presents IoU ≥ 0.5. Input images were resized to be 1344 x 1344 pixels square, Learning momentum was kept at 0.9, and weight decay was kept at 0.0001. The RPN anchor scales were kept at the default values: (32, 64, 128, 256, and 512). The training was executed for 1300 epochs, approximately 48h on an AWS EC2 P3 instance: p3.8xlarge with 4 GPUs Tesla V100, 32 vCPUs, and 244 GiB of RAM (Amazon, 2022), as the results reached an apparent plateau afterward.

Inference: All overlapping parameters between training and inference were kept the same. The model kept the threshold for displaying a segmented instance at a default > 0.50 certainty score. The output was taken as both segmentation masks over the image for visual evaluation and a binary image of segmented pixels that could be used later for image processing and subsequent collotelinite reflectance determination.

4.2.3 Dataset grouping and models trained

Due to the inherent maceral annotations imbalance found within the dataset, many different grouping approaches were attempted when training the network. The complex hierarchical annotation provided by VIA made it possible to easily group and regroup the many different macerals into their groups.

Many attempts at grouping the macerals were made, but for the sake of simplicity, only a small selection of them will be discussed here. Among the many different trained models, the following ones stand out:

- Model 1 - one single model for the most relevant and abundant macerals (collotelinite, collodetrinite, sporinite, fusinite, semifusinite, and inertodetrinite).
- Model 2 - one single model for the three main maceral groups plus the semifusinite maceral.
- Model 3 - encompassing the vitrinite maceral group.
- Model 4 - encompassing the inertinite maceral group.
- Model 5 - encompassing the liptinite maceral group.
- Model 6 - encompassing the liptinite maceral group but trained with different anchors.
- Model 7 - a model for identifying inertinite but distinguishing semifusinite as a distinct class.

- Model 8 - a model focused on identifying the most critical/abundant macerals in the inertinite group (fusinite, semifusinite, and inertodetrinite).
- Model 9 - a model focused on identifying the semifusinite maceral.
- Model 10 - a model focused on identifying the collotelinite maceral.

A non-extensive list of the tests not mentioned here follows:

- Varying the number of epochs in training
- Training only the heads layers of the network
- Altering the rpn_nms_threshold parameter, which controls the fraction of RPNs suppressed during the RPN proposals stage,
- Altering the post_nms_rois_training parameter, which controls the number of RoI's kept after the suppression stage during training,
- Altering the post_nms_rois_inference parameter, which controls the number of RoI's kept after the suppression stage during inference,
- Altering the image_resize_mode (square vs. Pad 64) parameter, which controls how the images are resized to "fit" the network, either resizing it to be a square or padding it with zeros,
- Altering the image_min_dim parameter, which controls the minimum dimension size an image can have after being resized,
- Altering the image_max_dim parameter, which controls the maximum dimension size an image can have after being resized,
- Altering the train_rois_per_image parameter, number of RoI's per image that is fed to the classifier,
- Altering the max_gt_instances parameter is the maximum number of ground truth instances used in an image.
- Altering the loss_weights parameters, which govern the weights given to the different losses from the various stages of the network (loss from the RPN stage vs. loss from the MRCNN stage as well as losses related to the bounding box accuracy, correct classification, and the segmentation masks)
- And others...

However, one parameter not altered was the score of certainty of the shown segmented objects. Alongside every single-segmented object, the network

evaluates its confidence, and this score will be seen in some of the following segmentation images. The score is also a parameter used during inference by establishing a score threshold above which the corresponding segmentation will not be discarded. The score threshold was maintained as a default 0.5 certainty.

By increasing the score threshold, it would be possible to eliminate segmented objects that the network isn't as sure about, including, most likely, mostly FPs, but also unavoidably a few TPs. The practical effect would be improving precision while slightly lowering or, at best, not changing the recall. Since most models discussed here showed relatively high precisions and low recalls, there was no advantage in increasing the score threshold. As nearly all objects segmented by the network were observed to have a>0.7 score, there was also no reason to lower the value. Thus, no reason was found to alter that parameter.

An illustration of segmented objects with their scores for model 2 can be seen in Figure 48.

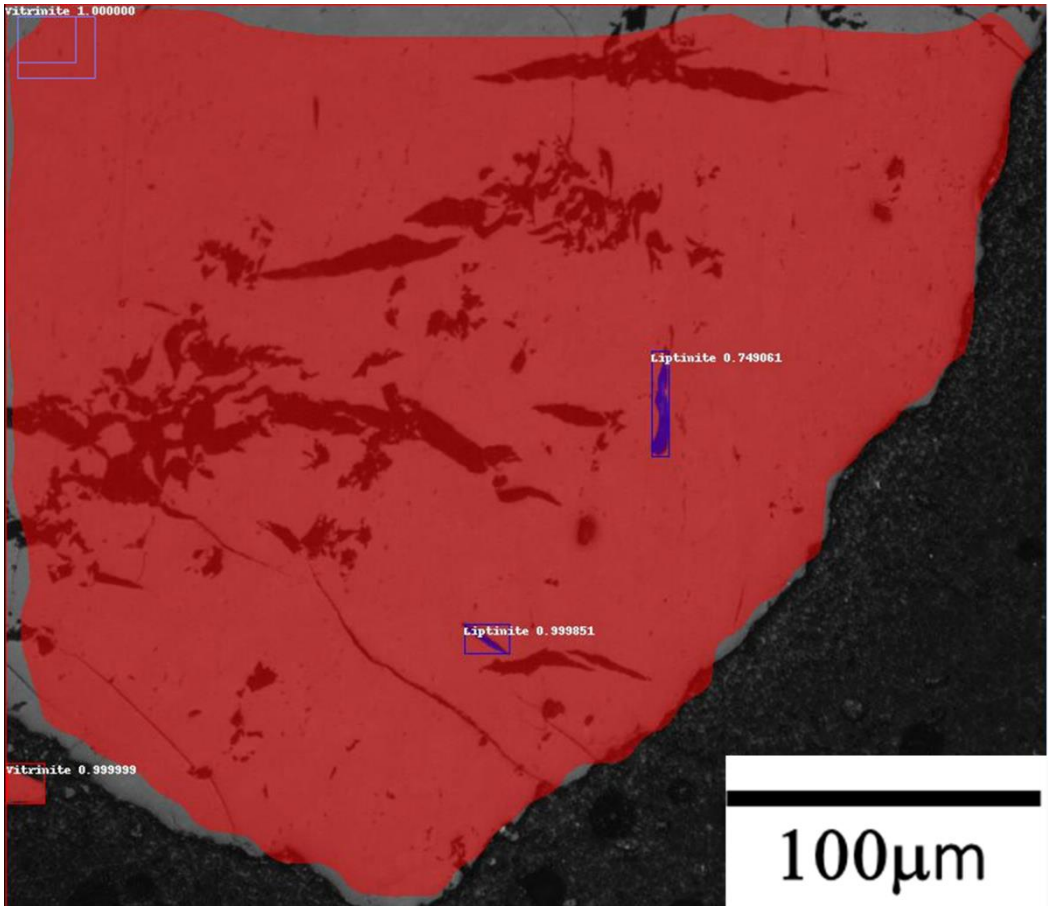

Figure 48 – Segmented image for model 2. Notice how the vitrinite objects have very high scores (0.99-1.0), while liptinite objects have more varying scores (0.74- 0.99).

4.3 Collotelinite Reflectance Determination

To properly determine vitrinite reflectance, it was necessary to standardize image capture by calibrating the microscope and determining the grey level-reflectance relation.

Because the current work focused on analyzing in the rank reflectance range of 0.97%-1.8% Rr, the 1.80% sample was used to determine the best capture conditions. The reason for that is because it features both the lowest reflectance pixels (the Epofix resin) and the highest reflectance pixels expected (the vitrinite and inertinite pixels). Therefore, lighting conditions capable of capturing both vitrinite and inertinite pixels in this sample, while avoiding light saturation, would ensure that no such saturation would occur for any macerals in the desired rank reflectance range.

To discover the relationship between the enhanced mode grey levels and the reflectance values, a one pixel-wide line selection was used to collect pixels belonging to the resin, vitrinite, and inertinite (Figure 49). This is necessary because the Hilgers system is calibrated and can calculate the reflectance values automatically; the captured images were processed outside the Diskus Fossil software environment.

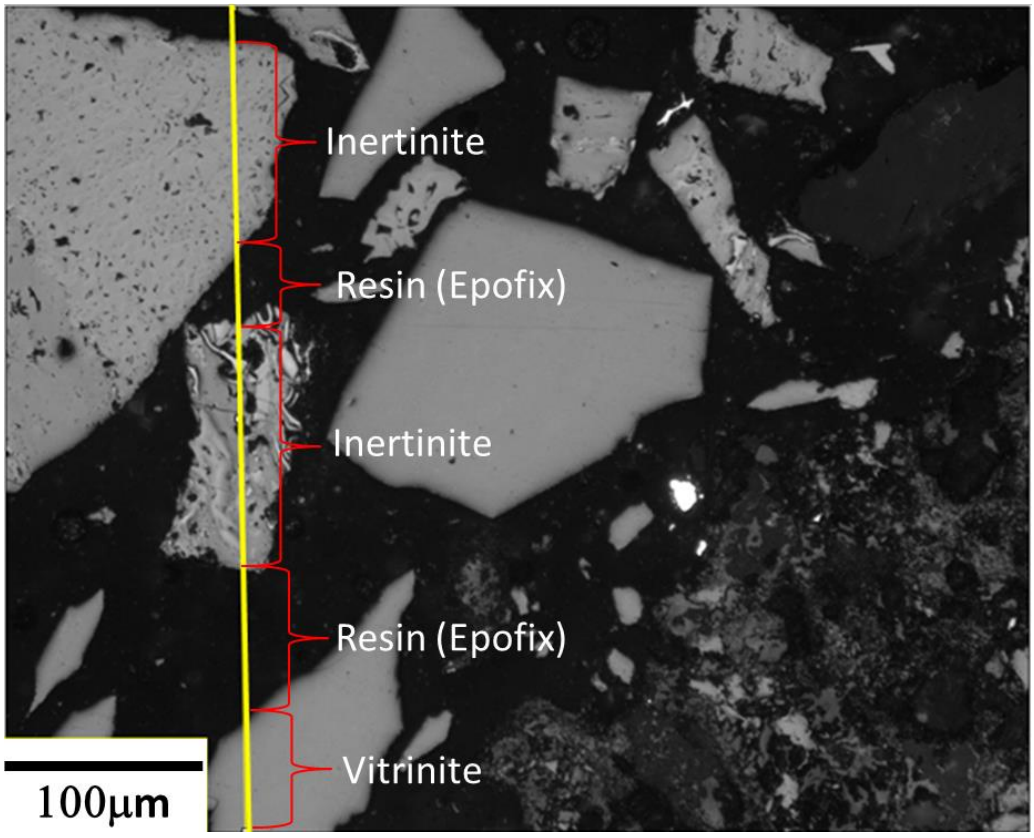

Figure 49 – Sample M image used for calibration. The yellow line collected the grey levels and reflectance values of the vitrinite and inertinite groups and of the resin.

The enhanced grey levels and their corresponding reflectances were extracted, and an exponential curve was used to fit them (Figure 50) and determine the equation that correlates with them (seen in equation (12)):

$$y=y_0+Ae^{R_0X} \quad (12)$$

where y is the reflectance value, y_0 is 0.82 %, A is 0.75%, R_0 is 0.00883 (enhanced grey level)$^{-1}$, and X is the enhanced grey level. The fitting had a correlation coefficient r^2 of 0.9979.

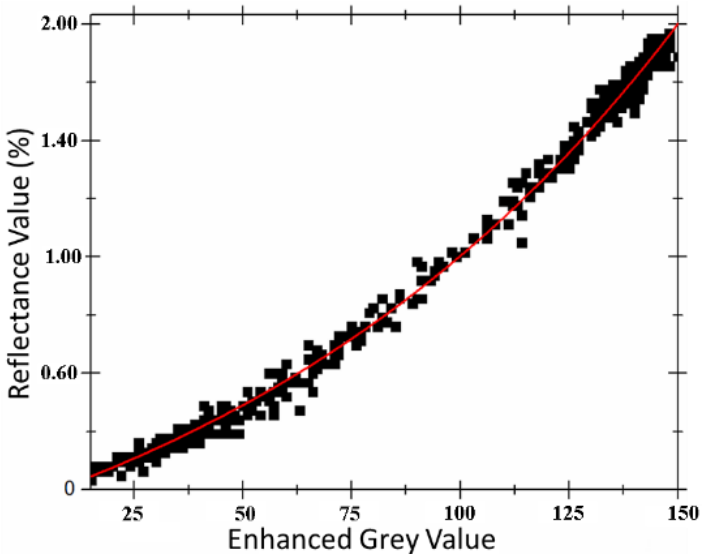

Figure 50 - Enhanced grey values and their corresponding reflectance alongside the exponential curve (in red) that fitted the data points.

With the equation determined, it is then possible to analyze the collotelinite segmented by the Mask R-CNN and determine the reflectance from the grey value, creating a calibration scale for any image captured in *enhanced* mode and aforementioned lighting conditions.

This is a critical step for this methodology. If it is to be employed by another operator, they would need to settle on lighting condition they can work with, determine their equivalent of equation (2), and update it in the rank reflectance module (Section 5.3). Once this one-time calibration is performed, the operator would be able to capture as many images and samples as necessary under the chosen lighting conditions.

5
Results and Discussion

5.1 Macerals and Maceral Groups

5.1.1 Model 1 (collotelinite, collodetrinite, sporinite, fusinite, semifusinite and inertodetrinite)

The first model trained focused on segmenting all the most relevant and abundant macerals found within the dataset: collotelinite and collodetrinite (belonging to the vitrinite group); fusinite, semifusinite, and inertodetrinite (belonging to the inertinite maceral group); and sporinite (main maceral of the liptinite group). Figure 51 and Figure 52 show the model's attempt at segmentation in a 1.2% and a 0.97% vitrinite reflectance samples.

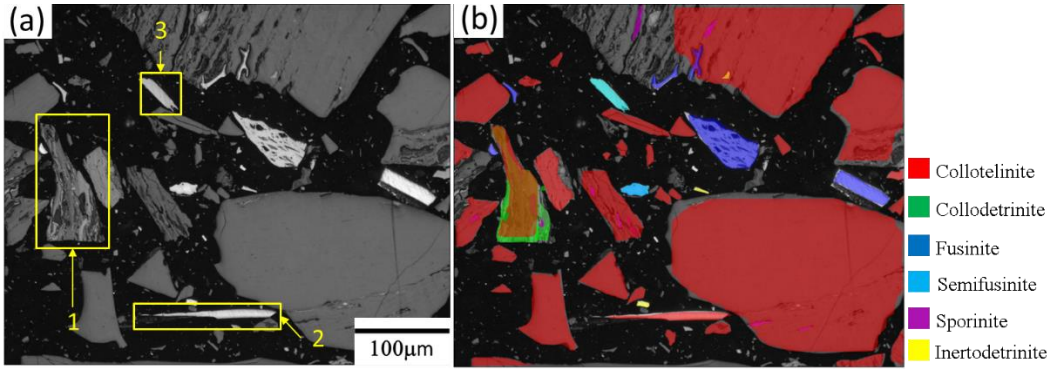

Figure 51 - Original grey scale image for a 0.97% sample (a). Corresponding segmentation results for model 1 (b).

Figure 51 also shows the three more common segmentation errors found in this model. Error 1 is an "indecision" of the model between collotelinite and collodetrinite that seems to occur occasionally in more complex particles (with many intertwined macerals). Error 2 is an infrequent misclassification of fusinite as collotelinite; it seems to happen more frequently when the fusinite is remarkably smooth and homogeneous. Error 3 is a misclassification of fusinite as semifusinite, it is the hardest ones to assess, as it usually occurs when it can be challenging even for an operator to make the distinction. It is also important to acknowledge a degree of "randomness" to these errors, especially the last 2; Figure 51b shows other similar fusinite objects that were properly classified.

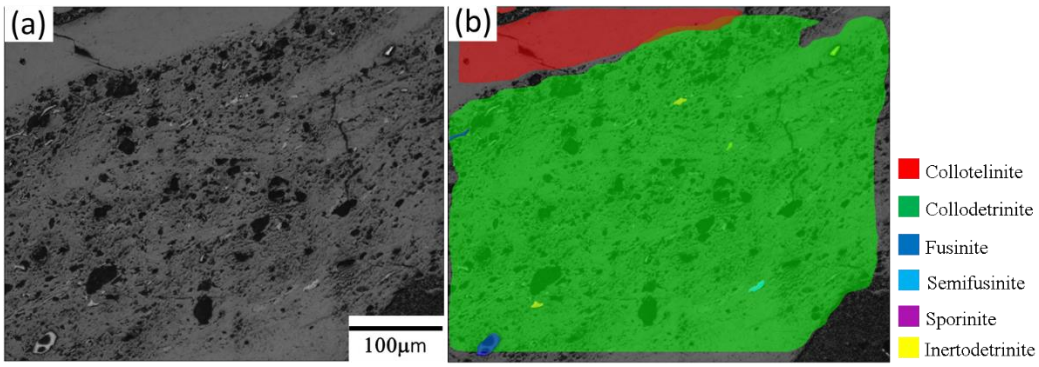

Figure 52 - Original grey scale image for a 1.2% sample (a). Corresponding segmentation results for model 1 (b).

However, due to the high number of images, “randomness” of the errors. and the complexity of maceral segmentation, it isn't easy to assess the model’s performance based only on a visual analysis of the images. It is more appropriate to use the validation metrics.

Table 4 shows the validation metrics (Precision, Recall, and F1-score) for Model 1. The model’s greatest strength was in segmenting collotelinite. A recall of over 75% indicates that the model correctly identified over 75% of all collotelinite annotated. However, the model fails considerably when identifying collodetrinite (7.97% recall). It is possible that is due to the common presence of other macerals inside the collodetrinite and the fact that it has a highly varying structure (sometimes being hard to distinguish from semifusinite). That marks an abysmal performance overall for the vitrinite group; despite collotelinite being usually the most abundant maceral, collodetrinite also occurs frequently (compare 883 collodetrinite objects to the 4355 collotelinite objects in this dataset according to Table 3).

Table 4 - Validation metrics for model 1.

Class	Precision (%)	Recall (%)	F1-score (%)
Collotelinite	63.71	75.06	68.92
Collodetrinite	18.03	7.97	11.06
Fusinite	60.00	32.43	42.11
Semifusinite	38.16	20.02	26.26
Inertodetrinite	43.94	18.53	26.07
Sporinite	41.60	16.51	23.64

An even worse problem can be seen in the inertinite group, where the most abundant maceral (semifusinite with 2115 instance objects annotated) had a meager 20.02% recall rate.

This leads the author to infer that any trained model would have difficulty doing a direct individual maceral segmentation on the images due to the

complexity of maceral segmentation. Therefore, for the successive attempts, the macerals were grouped in their maceral groups according to ICCP, and both collective and individual models were trained for each group plus semifusinite.

It is also interesting to discuss that Figure 52 highlighted a prevalent segmentation issue; the segmentation is more likely to fail at the edges of the image due to how the kernels "scan" them. All models presented that issue to a varying degree; the vitrinite/collotelinite models presented it more obviously due to those macerals being larger and closer to the edges. This work didn't concern itself with fixing this issue because i) for reflectance determination, it is only necessary to have a representative number of pixels, the segmentation failure at the edges doesn't hinder the process when hundreds of images were used; and ii) for area fraction results, the amount of area lost is small compared to the rest of the image. Despite that, it would be possible to fix this issue by automatically pre-adding 0 value pixels around the image before segmenting it or, more simply, cropping the borders after segmentation, analyzing only the inner area.

It is important to highlight that the author doesn't think all three metrics should be given the same weight. When trying to determine maceral composition, it is essential detect as much of the macerals as possible, which makes recall the vital metric in that regard. Because of that, special emphasis will be given to the recall metric for all of the following models, except model 10, which will be discussed later.

5.1.2 Model 2 (vitrinite, inertinite, liptinite, and semifusinite)

This model attempted to group the macerals into the three maceral groups but keep the semifusinite apart from the other inertinites. This is because semifusinite is partially reactive, unlike other inertinite macerals (≈30%). That fact alone makes it technologically interesting for a model to be able to evaluate the semifusinite apart from the other macerals. Collotelinite, telinite, collodetrinite, gelinite, and corpogelinite were grouped as vitrinite. Fusinite, funginite, secretinite, macrinites, and inertodetrinite were grouped as inertinite. Sporinite, cutinite, resinite, and liptodetrinite were grouped as liptinite. Semifusinite was kept as its class apart from the other inertinites.

The segmentation results can be seen in Figure 53. One of the most recurring identified segmentation mistakes was that the model struggled to distinguish the semifusinite from the other inertinites, also seen in Figure 53a and 50b.

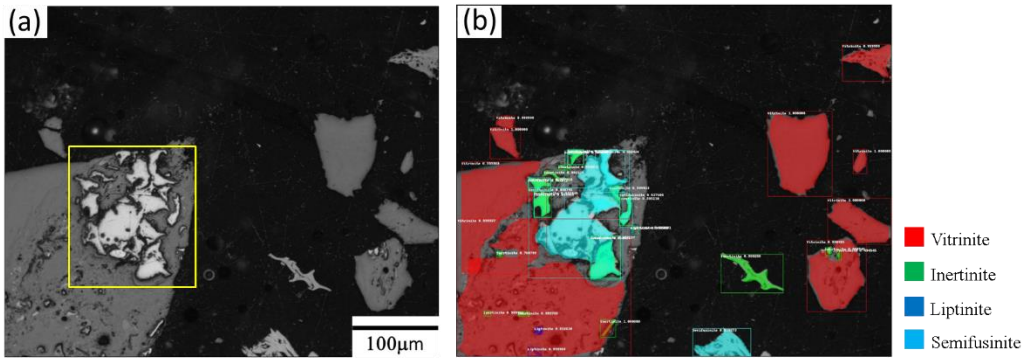

Figure 53 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 2 (b). Note the big fusinite object in (a) being partially misclassified as semifusinite.

The metrics for the model can be seen in Table 5.

Table 5 - Validation metrics for model 2.

Class	Precision (%)	Recall (%)	F1-score (%)
Vitrinite	57.58	68.86	62.72
Inertinite	60.19	23.86	34.17
Liptinite	34.73	10.79	16.46
Semifusinite	31.52	17.16	22.22

Grouping the collotelinite and collodetrinite was beneficial to the vitrinite group. The recall obtained (68.86%) for vitrinite was expectedly lower than the one obtained from collotelinite in model 1 (75.06%) but much higher than the recall obtained for collodetrinite (7.97%), indicating an overall improvement for the vitrinite group (considering vitrinite encompasses 3037 collotelinite and 566 collodetrinite objects).

Grouping the fusinite and the inertodetrinite did not seem as beneficial to the inertinite group. The recall obtained (23.86%) for inertinite was expectedly lower than the one obtained for fusinite in model 1 (32.43%) but higher than the recall obtained for inertodetrinite (18.53%). Despite being numerous, inertodetrinites are much smaller than fusinites, thus making it inconclusive if this intermediary recall was an improvement. Being kept separated from the other inertinites did not improve the semifusinite results either, as the model struggled to distinguish it from the other inertinites, As seen in Figure 53.

Grouping the liptinites also did not improve the metrics for the group. Due to the high complexity of the macerals, it is possible that the models would benefit from focusing on learning single classes instead of many. Therefore, individual models for vitrinite, inertinite, and liptinite were tested next.

5.1.3 Model 3 (vitrinite)

This model was trained by grouping all individual vitrinite macerals (collotelinite, telinite, collodetrinite, gelinite, and corpogelinite) as vitrinite only and all the rest as background (with a total of 5238 instance objects annotated according to Table 3). An example of segmentation can be seen in Figure 54.

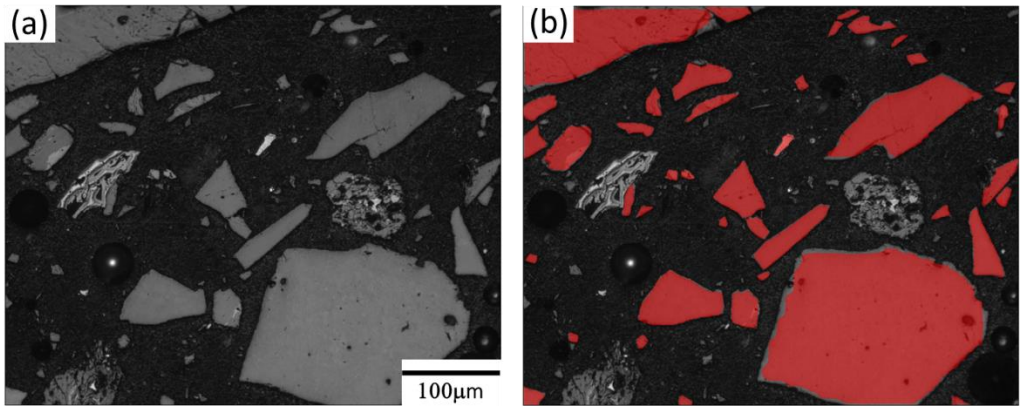

Figure 54 - Original grey scale image for a 1.2% sample (a). Corresponding segmentation results for model 3 (vitrinite) (b).

Table 6 shows a considerable improvement for all metrics compared to both collotelinite and collodetrinite from model 1. Possibly because the individual macerals have more in common with each other than with macerals from other groups, and because the model doesn't occupy itself with the other macerals, this "lessened" the burden during the model's training.

Table 6 - Validation metrics for model 3.

Class	Precision (%)	Recall (%)	F1-score (%)
Vitrinite	81.07	88.42	84.59

Because vitrinite is the maceral group that is most likely to contain other macerals/mineral matter inside it, it is possible to see in Figure 55 that the model is expected to include pixels of these inclusions as vitrinite. This indicates that these results would have to be refined for an area fraction or rank reflectance determination analysis.

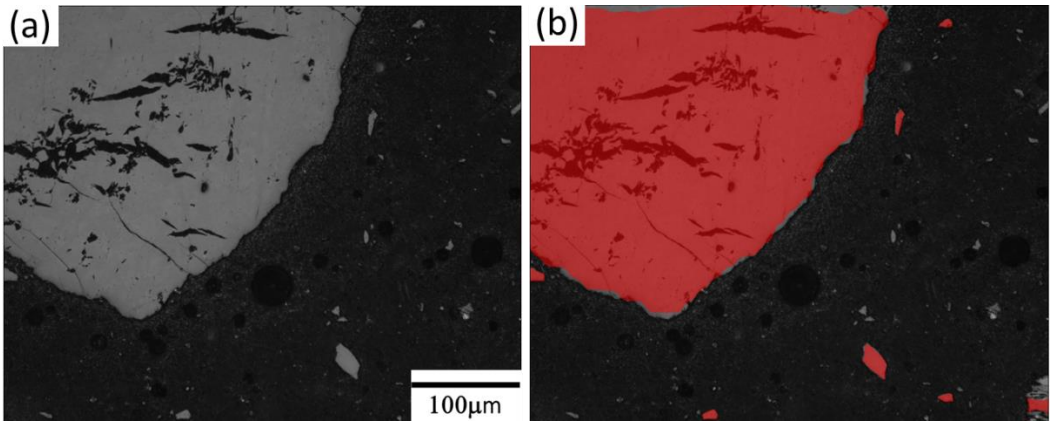

Figure 55 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 3 (vitrinite) (b). Notice how the cracks and mineral matter pixels were segmented alongside the actual vitrinite.

5.1.4 Model 4 (inertinite)

This model was trained by grouping all individual inertinite macerals (fusinite, semifusinite, funginite, secretinite, macrinites, and inertodetrinite) as inertinite only (with a total of 7373 instance objects annotated according to Table 3). An example of segmentation can be seen in Figure 56 and Figure 57.

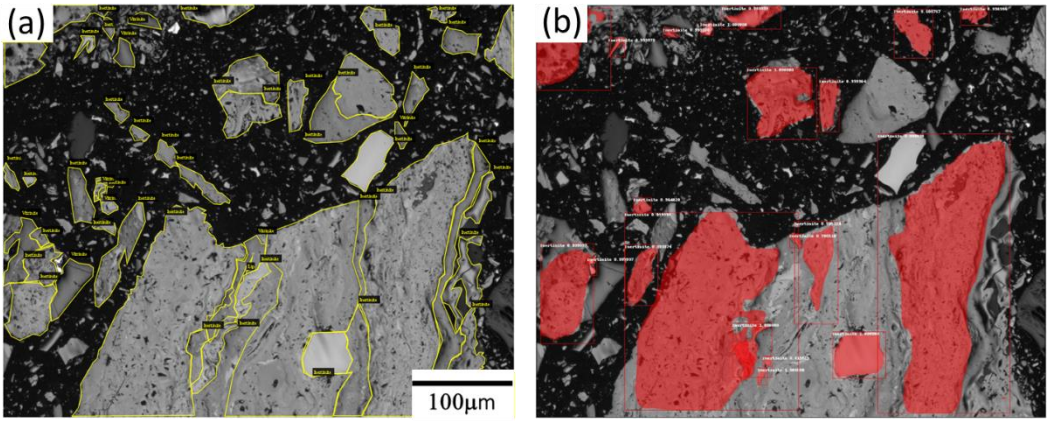

Figure 56 - Original grey scale image for a 1.03% sample (a). Corresponding segmentation results for model 4 (inertinite) (b).

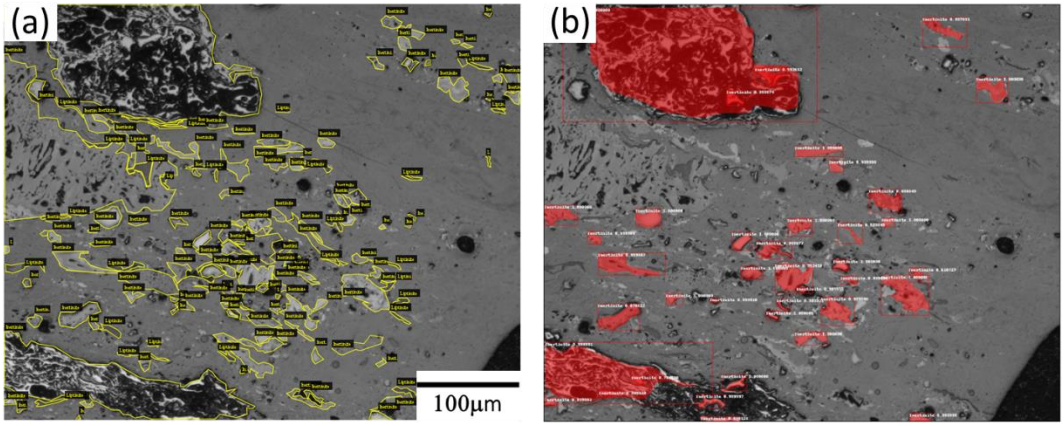

Figure 57 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 4 (inertinite) (b).

Table 7 shows the test results for model 4 and, like model 2, also offers a considerable improvement on all metrics compared to model 1.

Table 7 - Validation metrics for model 4.

Class	Precision (%)	Recall (%)	F1-score (%)
Inertinite	90.81	42.11	57.54

Even though this model achieved a very high precision of 90.81%, it still has a very low recall of 42.11%. A low recall and high precision indicate that the model correctly identifies nearly everything it tries to segment, but it cannot segment as many objects. This can be easily seen in Figure 56 and Figure 57.

5.1.5 Model 5 (liptinite)

This model was trained by grouping all individual liptinite macerals (sporinite, cutinite, resinite, and liptodetrinite) as liptinite only (with a total of 5068 instance objects annotated according to Table 3). An example of segmentation can be seen in Figure 58 and Figure 59.

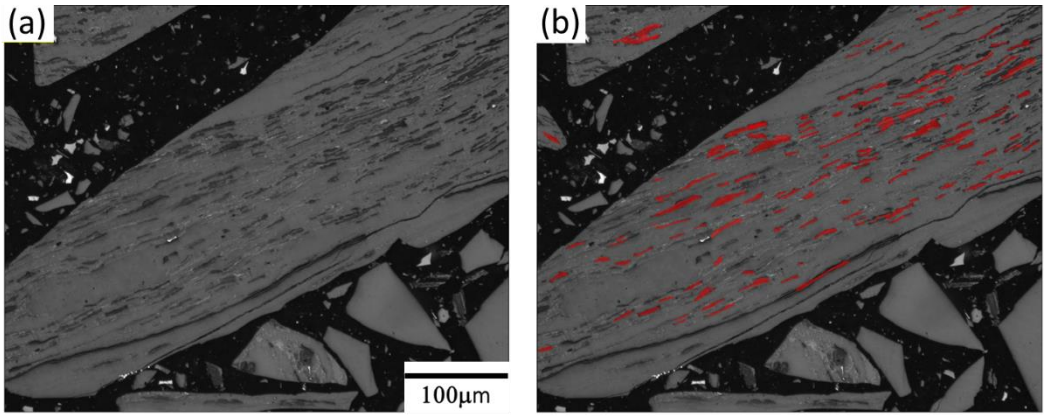

Figure 58 - Original grey scale image for a 0.97% sample (a). Corresponding segmentation results for model 5 (liptinite) (b).

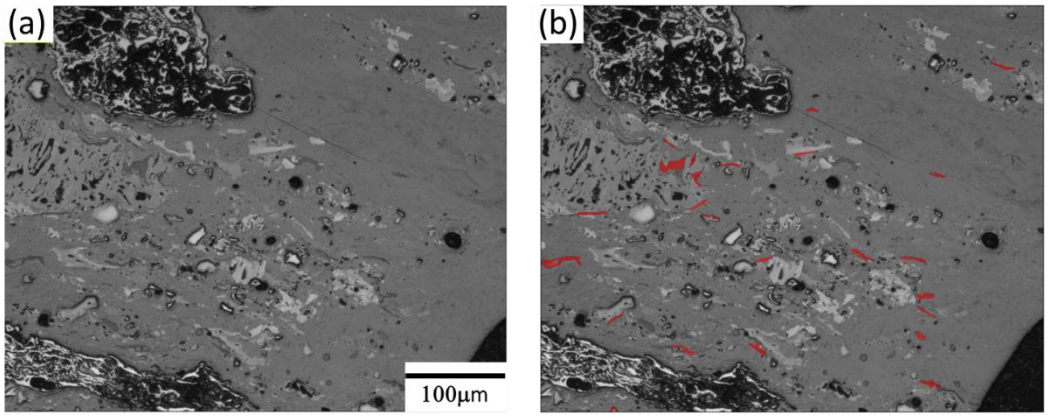

Figure 59 Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 5 (liptinite) (b).

Table 8 shows the validation metrics for model 5. This model improved over model 1, though not nearly as much as models 3 and 4. Since 4743 out of the 5068 liptinites were sporinite, it is believed that the improvement seen here was mainly due to the model being able to focus solely on learning the liptinite features and not really because of the grouping of the sporinite and cutinite objects.

Table 8 - Validation metrics for model 5.

Class	Precision (%)	Recall (%)	F1-score (%)
Liptinite	67.46	22.53	33.78

Because of the massive number of tiny liptinites in a picture, the operator couldn't annotate every one of them. That causes an issue with the precision metric; whenever the model identifies a liptinite present in the image that wasn't annotated, it will be incorrectly classified as a False Positive (FP). This means that

the liptinite model's precision and F1-score will probably always be lower than their actual values, and recall becomes the most critical metric in this case.

Similar to model 4, this model still experiences a low recall and relatively high precision.

5.1.6 Model 6 (liptinite, different anchors)

This model was trained by grouping all individual liptinite macerals (sporinite, cutinite, resinite, and liptodetrinite) as liptinite only (with a total of 5068 instance objects annotated according to Table 3).

This is the only model with parameters slightly different from the ones described in section 4..2.2. In this case, the altered parameter was the RPN anchors, which are used by the network to establish the size and shape of the ROI that will be segmented. Due to the wide variety of sizes and shapes the macerals exhibit, it is usually not worth it to vary the RPN anchors, and they were left at their default values of (32, 64, 128, 256, 512) pixels. However, since this model is only concerned with liptinites, which are consistently smaller than the other macerals, the RPN anchors were adapted to suit them better.

The anchors chosen were (8,24,48,64,152) as most liptinites could be found within the 0-150 pixels diameter (Feret) range (Figure 60). Since the distribution is skewed towards lower values, the anchors were chosen to favor the 0-64 pixels diameter range. The distribution was calculated using the same hand-drawn annotations used as ground truth (Figure 47).

One important note is that all sizes here are referred to in pixels, as the model works on the image in terms of its pixels and not its real scale.

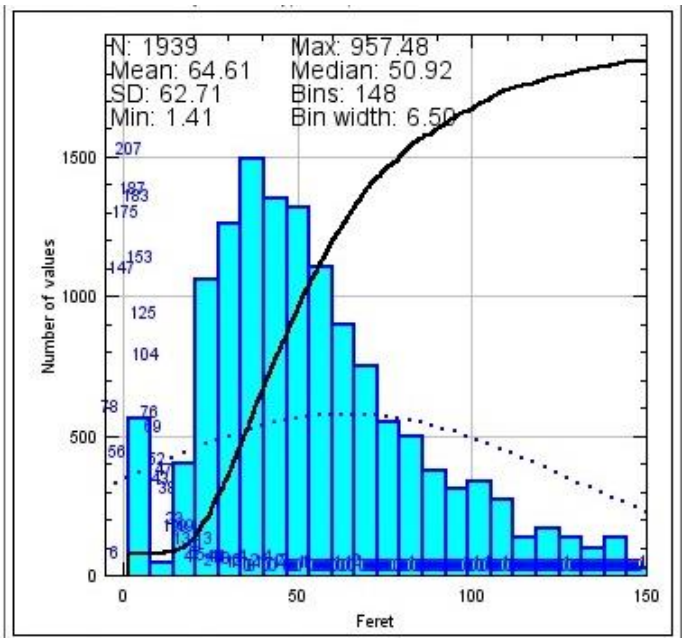

Figure 60 - Feret distribution for all liptinites manually annotated in the dataset.

An example of segmentation can be seen in Figure 61 and Figure 62.

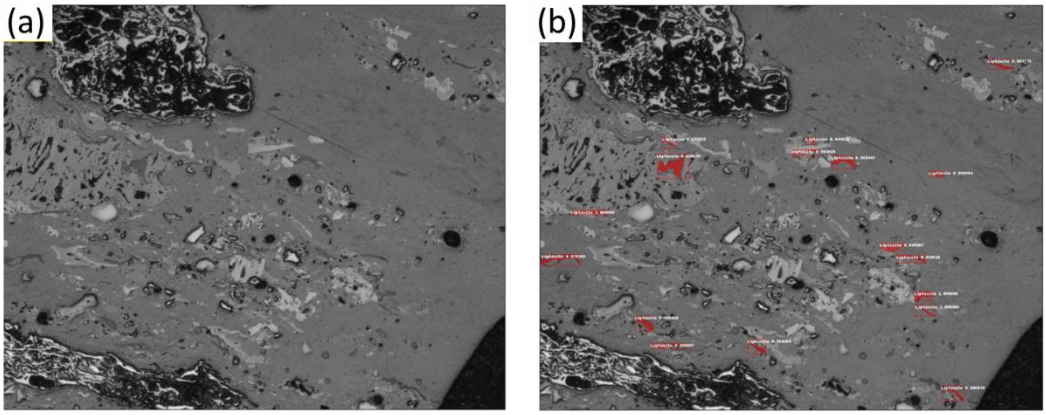

Figure 61- Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 6 (liptinite) (b).

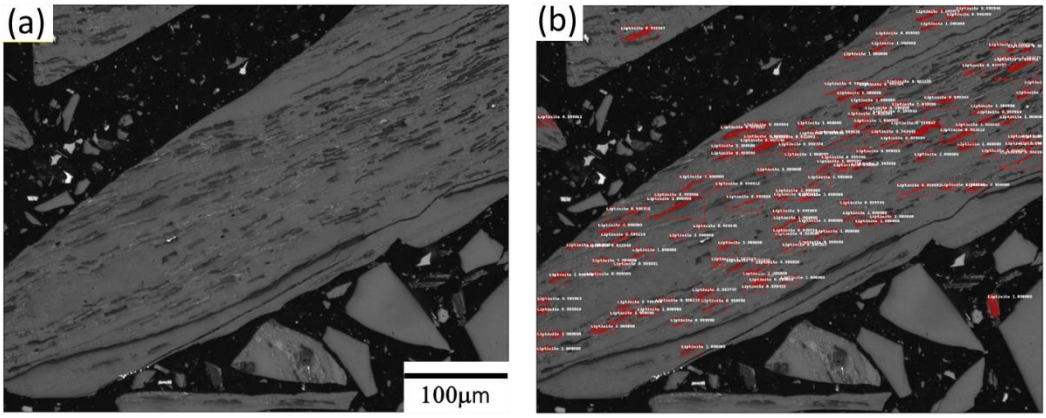

Figure 62 - Original grey scale image for a 0.97% sample (a). Corresponding segmentation results for model 6 (liptinite) (b).

Table 9 shows the validation metrics for model 6. It can be seen that altering the anchors positively impacted recall, which improved from 22.53% to 25.83%. Since recall is considered the primary metric for liptinite, this model can be regarded as superior to model 5, despite the loss in precision. Altering the size of the anchors to be smaller and more suitable to liptinites had, therefore, a positive impact on the model.

Table 9 - Validation metrics for model 6.

Class	Precision (%)	Recall (%)	F1-score (%)
Liptinite	64.57	25.83	36.85

5.1.7 Model 7 (inertinite and semifusinite)

This model was trained by grouping all individual inertinite macerals (fusinite, funginite, secretinite, macrinites, and inertodetrinite) as inertinite (with a total of 4050 instance objects annotated according to Table 3), except for the semifusinite maceral, which was kept as a distinct class. An example of segmentation can be seen in Figure 63 and Figure 64.

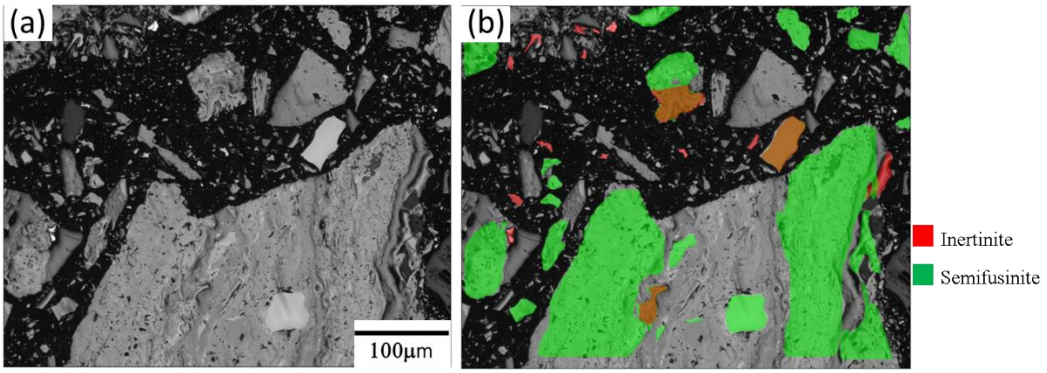

Figure 63 - Original grey scale image for a 1.03% sample (a). Corresponding segmentation results for model 7 (b).

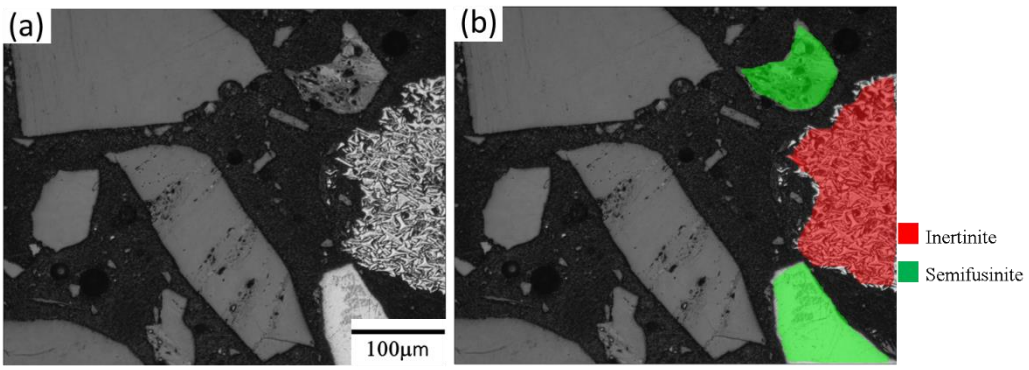

Figure 64 - Original grey scale image for a 1.2% sample (a). Corresponding segmentation results for model 7 (b).

Table 10 shows the validation metrics for model 7. Despite what was expected, splitting the semifusinite from the other macerals did not result in any performance gain. Precision and recall plummeted for both classes compared to the sole inertinite model. Therefore, another approach will be necessary if semifusinite is to be segmented.

Table 10 - Validation metrics for model 7.

Classes	Precision (%)	Recall (%)	F1-score (%)
Inertinite	65.60	18.19	28.48
Semifusinite	26.96	15.19	19.44

5.1.8 Model 8 (fusinite, semifusinite and inertodetrinite)

This model was trained for all three main individual inertinite macerals (fusinite, semifusinite, inertodetrinite). The motivation for this training was to achieve better results for all three macerals, as they aren't as imbalanced as the macerals in the other two groups. An example of segmentation can be seen in Figure 65, Figure 66, and Figure 67.

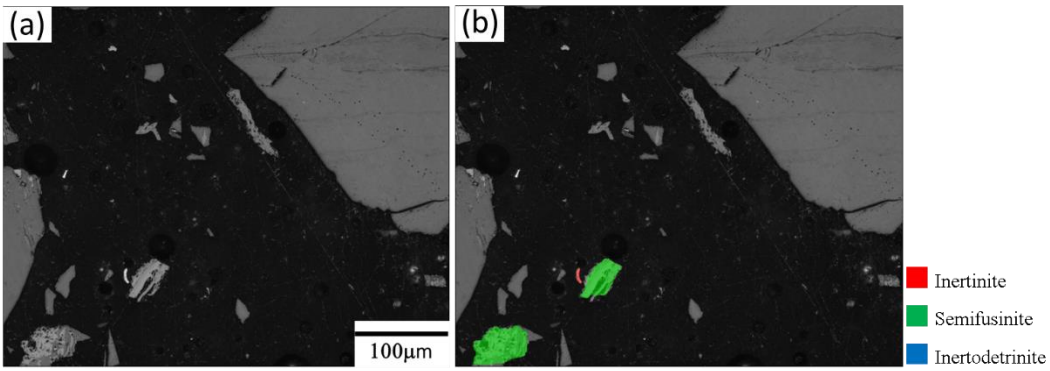

Figure 65 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 8 (b).

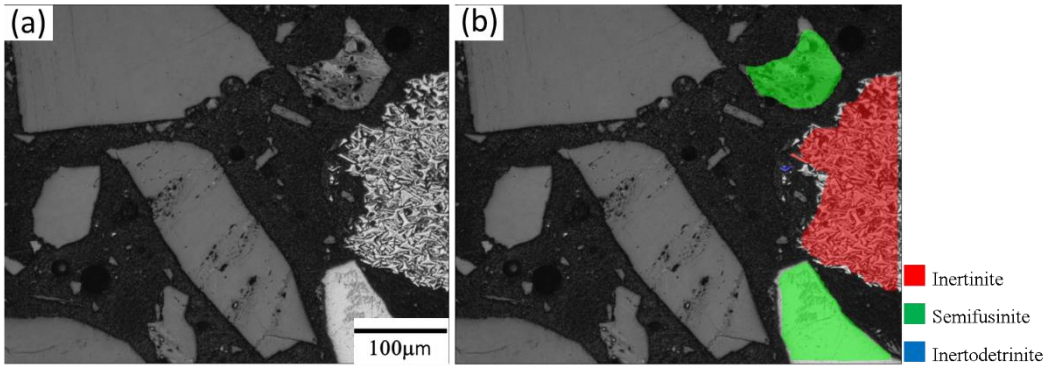

Figure 66 - Original grey scale image for a 1.2% sample (a). Corresponding segmentation results for model 8 (b).

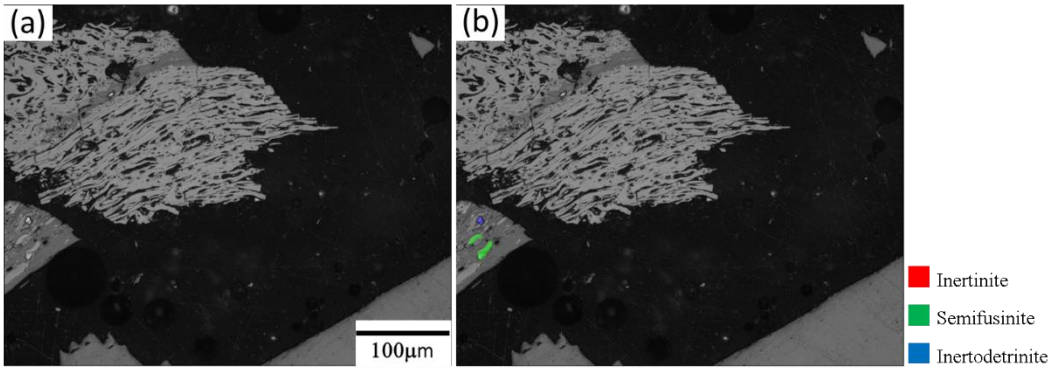

Figure 67 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 8 (b).

Table 11 shows the validation metrics for model 8. As with model 7, splitting the inertinite group into three macerals did not translate into any perceivable gain for the model's overall performance. Model 4 remains the best model for inertinite segmentation.

Table 11 - Validation metrics for model 8.

Classes	Precision (%)	Recall (%)	F1-score (%)
Fusinite	54.78	34.05	42.00
Semifusinite	39.19	12.76	19.25
Inertodetrinite	49.13	12.03	19.33

5.1.9 Model 9 (semifusinite)

This model was trained to focus on the single maceral semifusinite, which belongs to the inertinite maceral group (with a total of 3323 instance objects annotated according to Table 3). An example of segmentation can be seen in Figure 68 and Figure 69. In Figure 68, it can be seen that the model is misclassifying a huge fusinite as semifusinite. In Figure 69, it can be seen that the model also cannot identify many of the present semifusinites.

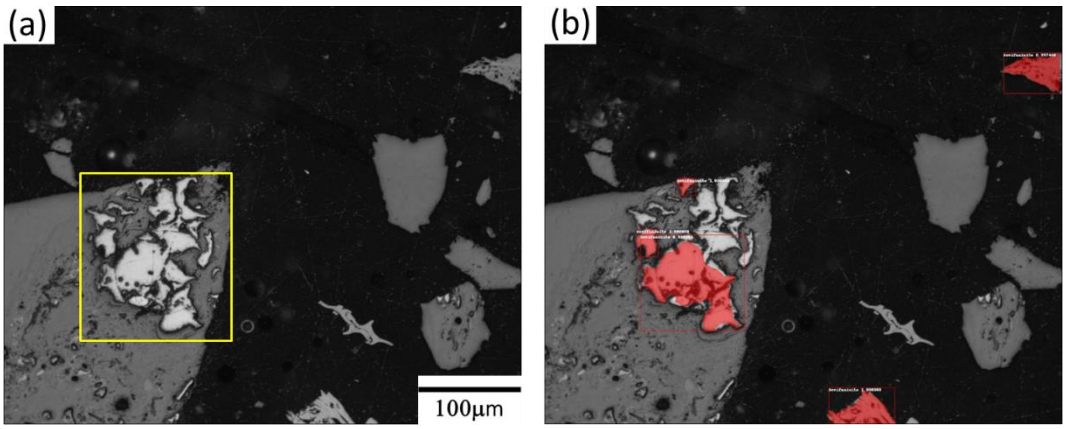

Figure 68 - Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 9 (b). Note the big fusinite object in (a) being partially misclassified as semifusinite.

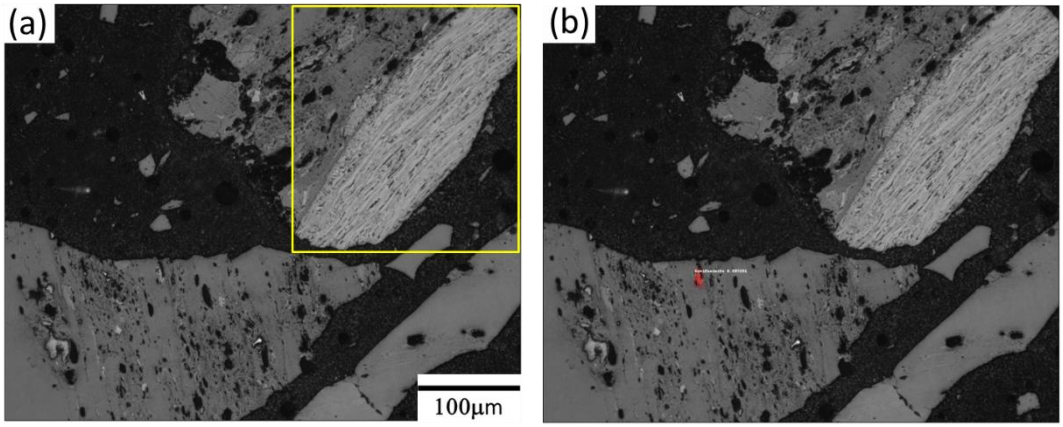

Figure 69 - Original grey scale image for a 1.20% sample (a). Corresponding segmentation results for model 9 (b). Note the big semifusinite object in (a) being completely ignored by the model.

Both mistakes above reflect the model's poor metrics, as seen in Table 12. These metrics seem even lower than the ones obtained for semifusinite in model 1 (Table 4). That means the semifusinite model didn't benefit from focusing on a single class, unlike the others. It might be that not giving the model any other option during training might be hampering the learning. Since semifusinite has a very high intra-class variance and can occasionally be very similar to vitrinite macerals or fusinite, it might be that not allowing the model to learn the other two options is reducing its ability to distinguish between them.

Table 12 - Validation metrics for model 9.

Class	Precision (%)	Recall (%)	F1-score (%)
Semifusinite	45.15	10.23	16.68

This is a setback, as an efficient semifusinite model could provide a way to quantify it separately from the total inertinite. One possible way to address this issue would be to improve the inertinite model and train an individual inertinite model similar to model 8 only on the segmented results. It is possible that, by training a model identical to 8 only on the segmented inertinite results, better results could be achieved as there would be less "noise" present in the images. Such images would have less of an intraclass variety, making them easier to be dealt by the network.

Alternatively, it is possible that a single class model could work for semifusinite. Due to its intrinsic complexity, a much higher number of annotated objects would be necessary to achieve good results.

5.1.10 Model 10 (collotelinite)

This model was trained to focus on the single maceral collotelinite, which belongs to the vitrinite maceral group (with a total of 4355 instance objects annotated according to Table 3). An example of segmentation can be seen in Figure 70.

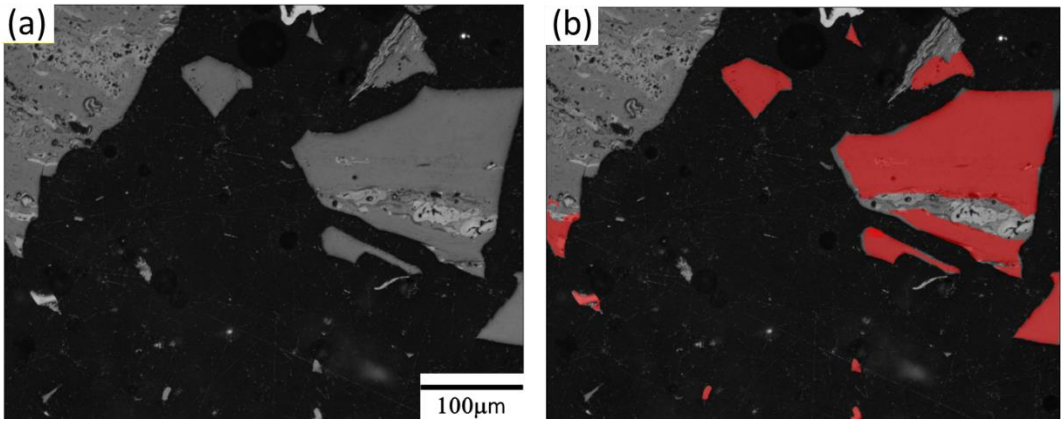

Figure 70-Original grey scale image for a 1.04% sample (a). Corresponding segmentation results for model 10 (collotelinite) (b).

As with the vitrinite model, it is possible to see in Figure 70 that the model is likely to include pixels inclusions not belonging to collotelinite. This indicates that these results would also have to be refined for rank reflectance determination analysis.

Table 13 shows that the model had similar success rates to the vitrinite model (Table 6). Coal rank is traditionally assessed by measuring the maximum or mean random vitrinite reflectance. However, the standards advise making the measurement specifically on the collotelinite individual maceral (ASTM D 2798, 2011; ISO 7404-5, 2009; AS 2856.3, 2000). Due to being, by definition, a very homogeneous maceral (free from other maceral inclusions/“clean” and smooth), its reflectance exhibits a more homogeneous distribution across the coalification model, having “less noise.” Since both the collotelinite and vitrinite models have similar success rates and the collotelinite, by definition, would have less noise in its reflectance distribution, the collotelinite model is the more appropriate one to be used for rank reflectance determination.

Table 13 - Validation metrics for model 10.

Class	Precision (%)	Recall (%)	F1-score (%)
Collotelinite	81.67	85.25	83.42

It should be highlighted that in this case, precision is the more relevant metric, as this model should be used for vitrinite reflectance determination. Unlike maceral composition, it is important that the vitrinite reflectance determined represents the sample. Because of that, it is more important to make correct

predictions than it is to detect every collotelinite particle available. As the network can be run over hundreds of images in minutes, it is not important that it detects every collotelinite, even if it only detects a few, it will eventually have a statistically relevant population. Therefore, in terms of vitrinite reflectance analysis, precision is a more relevant metric than recall, and model 10 does have higher recall than model 3 (vitrinite).

5.1.11 Comparative discussion of the models

Upon inspecting the validation metrics for the previous models, it was clear that the best segmentation results were obtained for models 3 (vitrinite), 4 (inertinite), 5 (liptinite), and 10 (collotelinite). Table 14 shows the results for the best segmentation models. Despite being necessary for the training due to the high amount of inertinites, sample B is a blended coal, so it is interesting to evaluate the models, including only the samples of natural unprocessed single component coals. Those results are also included in Table 14.

Table 14 - Validation metrics for models 2, 3, and 5, considering and disregarding sample B images.

Classes	Precision (%)	Recall (%)	F1-score (%)
Vitrinite	81.07	88.42	84.59
Vitrinite (no sample B)	89.42	89.04	89.23
Collotelinite	81.67	85.25	83.42
Collotelinite (no sample B)	83.72	85.84	84.77
Inertinite	90.81	42.11	57.54
Inertinite (no sample B)	94.81	54.01	68.81
Liptinite	64.57	25.83	36.85
Liptinite (no sample B)	62.95	26.20	37.00

The vitrinite model performed the best out of the 3, possibly due to the maceral group being considerably more abundant than the other 2, and because the vitrinite might have an easier structure to define (as it is simpler morphologically). The model had both high precision and high recall, which indicates that the model is correctly identifying most of the vitrinite in the samples. Since vitrinite can contain other macerals, the segmentation results could be refined further by subtracting the inertinite and liptinite results from it and adding a threshold segmentation step. However, this refining would remain limited due to the low recall from the other two models.

However, due to the collotelinite objects containing/being adjacent to other objects (other macerals or noise), it is necessary to post-process the resulting images to clean the collotelinite objects for subsequent reflectance determination.

Unfortunately, it would not be possible to avoid this issue even with dataset expansion, as it is impossible to delineate a collotelinite for the dataset while automatically excluding the inclusions it may contain. If the inclusions were delineated and deleted, the resulting black pixels would change the appearance of the collotelinite, hampering the ability of the network to segment it. There is currently no way to deal with this issue completely. Other works have tried different approaches from this one. TIWARY et al. (2020) annotated triangular maceral group areas that belonged to a single group. This might make it easier for the network to learn the features belonging to the maceral group, but it would also carry the drawback of adding arbitrarily chosen areas that might not represent the real variety within the class. Moreover, using arbitrary areas as ground truth means the metrics evaluated will not reflect the performance of the model.

It was decided to extensively annotate the maceral objects as accurately as humanly possible so the network would be "used" to dealing with realistic, complex situations. Also, by having a more accurately annotated ground truth, the validation metrics would be more reflective of the actual performance of the model.

To determine rank reflectance, the model doesn't need to identify all of the vitrinite; it is only required for most of its segmentation attempts to be correct. That means that precision is the more relevant parameter of the two for rank reflectance determination. Even though the vitrinite model had a higher precision than the collotelinite model, the collotelinite does present a more well-behaved reflectance distribution across the coalification process (as it is less likely to have inclusions like collodetrinite has). Because of that, it remains an adequate model for rank reflectance determination (ASTM D 2798, 2011; ICCP, 1998).

On the other hand, when analyzing maceral composition, it is more critical to correctly identify as many of the macerals as possible. For this reason, recall is a more adequate metric than precision when considering models for area fraction determination.

The inertinite model suffered, as mentioned before, from very high (over 90%) precision and relatively low recall (around 40%). That means the model was correct in most segmentation attempts but couldn't segment all the inertinites present (observed to happen to fusinite, semifusinite and inertodetrinite). Many other training attempts and parameter changes, not shown here, were performed to try to improve the model's performance, but none of them was able to outperform the one presented here.

Deep learning methods have the distinct ability to continuously improve their performance with a continuous expansion of the dataset used for training (LAQTIB et al., 2019). Since no change in the training parameters improved the model's performance, the most logical step would be to expand the dataset with new images and samples rich in inertinite, expanding into non-coking coal samples that could have more inertinite.

The liptinite model faced a similar issue, and all that was said about the inertinite model also applies to it. It should be noted, however, that the liptinite model does have a few distinct issues of its own. Due to the sheer number of liptinites present in an image, it is almost impossible to annotate every single instance of liptinite available. The non-annotated liptinites become background and hinder the model's learning. Another issue is that liptinite is the easiest macerals group to mistake for other objects (dirt, scratches, or mineral matter). During traditional coal petrography, the operator has access to fluorescence that can be used to identify liptinites in the most complex cases (PICKEL et al., 2017), which is a luxury this model never has, as it works solely on RLOM images. Also, because it is impossible to annotate every single liptinite present, any liptinite detected by the models that weren't annotated will be considered a false positive. This means that the recall and the precision are very likely underestimated in this case.

As is the case for the inertinite model, extensive work on expanding the dataset with lower-rank samples (lower than 1.0% vitrinite reflectance) rich in liptinites is the most logical way to improve the recall numbers for this macerals group.

Unfortunately, the semifusinite model was not as successful as the collotelinite model (Table 13), despite both singling out very abundant individual macerals. This is possible because, despite both being abundant, semifusinite has a more complex and variable structure and would need to be more represented in the dataset to be adequately learned by the network. An annotation effort primarily focused on semifusinite would be fundamental to improve those results, which, in turn, could be combined with the inertinite results to achieve a quantitative evaluation that distinguishes both.

Mask R-CNN makes its segmentation based on a multitude of different features of varying levels of abstraction. Even though it is not currently possible to extract which features were precisely used and their weights, they are sure to involve reflectance and texture features based on heterogeneity. The features are certain to be relevant to the collotelinite segmentation because the network was trained on actual collotelinite objects, allowing the network to learn its distinctive characteristic features. Compared to the earliest attempts (AGUS et al., 1994) at defining the macerals using texture parameters, CNN-based approaches have the advantage of considering thousands of different features at the same time and, due to the automated learning algorithm, determine which features it will use, leading to relevant but highly abstract features that a human operator would not be able to imagine or define.

The focus on the dataset creation in a 0.97%-1.8% reflectance range reflects its industrial interest, even though it is understood that maceral segmentation should be easier in lower ranks, as the difference between the macerals groups is more pronounced. Other works have attempted maceral segmentation in lower rank ranges (IWASZENKO and RÓG, 2021)., which suggests that deep learning

can be a valuable and effective tool for semantic segmentation in a wide rank range.

While both IWASZENKO and RÓG, 2021 and WANG et al., 2019 had better segmentation results than the ones presented in this work, this could be explained by several factors. The tagging performed in this work was much more extensive and reflective of the real ground truth than in IWASZENKO and RÓG, 2021, which makes it more likely to results in lower metrics, while the images were captured in random positions, avoiding simpler images. In the case of WANG et al, 2019, the dataset focused on a lower 0.5-0.8% reflectance range, where the distinctions between the maceral groups are more pronounced and th distinction would be easier to make.

Data augmentation techniques could also be implemented in the future in order the improve performance. However, it must be noted that maceral segmentation faces a few distinct problems. Due to the already very complex structures that macerals may present, it would be advisable to focus on data augmentation methods that merely flip or rotate the images, any variation that deforms the macerals is likely to be impact the training negatively. Another issue is that class imbalance would still be very difficult to correct, each maceral's natural frequency in nature is very difficult. Since they will rarely appear by themselves in a single image, it is not possible to focus data augmentation on individually selected images to augment only desired classes. Thus, the maceral imbalance would be preserved as the data is augmented.

Some preliminary data augmentation was done in this work; however, it surprisingly didn't result in any performance improvement. This phenomenon is still being studied and data augmentation remains a test to be performed in the future.

In summary, the vitrinite, inertinite, and liptinite group-level models were the more adequate ones to be used for the area fraction tests. In contrast, the collotelinite one was the most appropriate one to be used for the rank reflectance determination tests.

5.2 Area fraction Results and Discussion

Despite exciting results for maceral segmentation, most of the maceral analyses performed in coal petrography focus on obtaining maceral composition, not segmentation. Therefore, a method inspired by traditional point counting was developed to estimate the macerals composition based on the segmentation results.

Once the main maceral groups have been segmented, it is possible to generate binary masks for each class, which allows for traditional image processing measurements, such as area fraction (Figure 71). But because vitrinite is annotated as a whole object and often contains other macerals, it is necessary to subtract the inertinite and liptinite from the vitrinite so as not to overestimate the

latter. Furthermore, a new segmentation can be performed by simply thresholding the vitrinite results in the 85 - 145 grey level range (corresponding to approximately 0.85%-1.85% reflectance range, slightly broader than the 0.97%-1.8% range this work was concerned with) to eliminate undue residues (Figure 72). The grey level range will be automatically determined in other lighting conditions depending the grey level- reflectance calibration done for it. These steps, therefore, should minimize vitrinite overestimation even further.

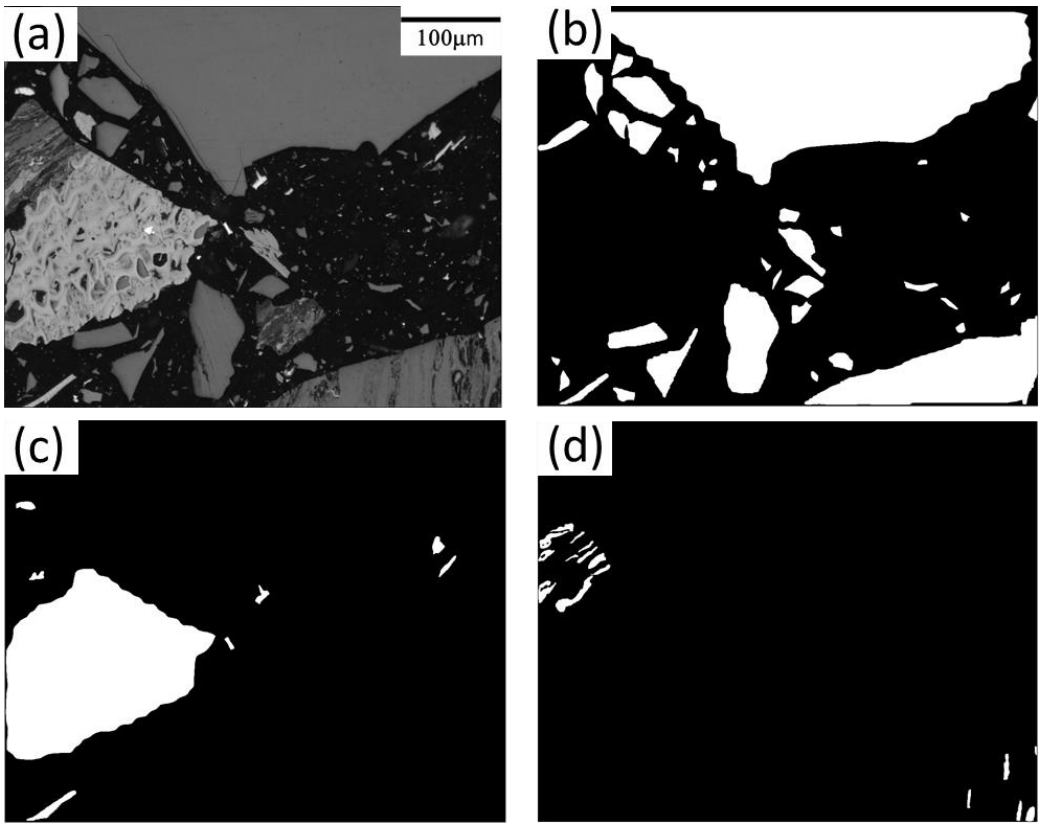

Figure 71 - Original grey image for a 0.97% sample (a), original vitrinite segmentation results before processing (b), inertinite segmentation results (c), and liptinite results (d).

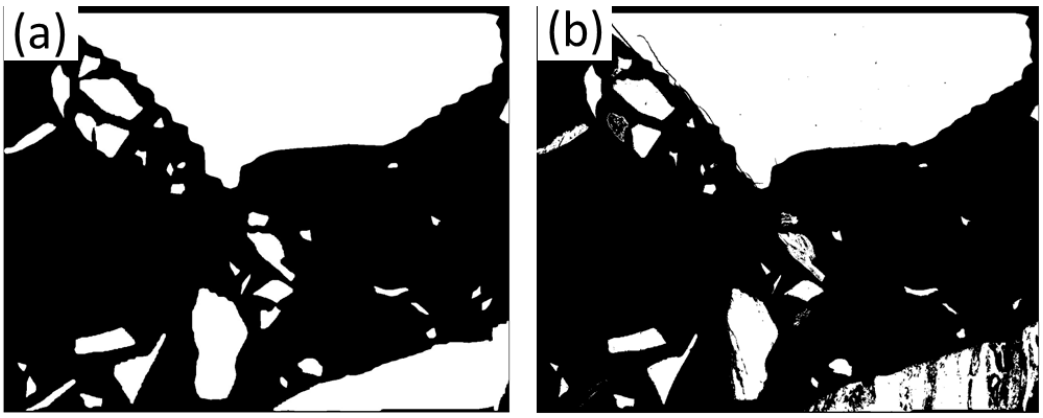

Figure 72 - Vitrinite segmentation results before (a) and after (b) being segmented in the 85-145 grey level range.

Once the images have been processed, they can be used for area fraction measurements. A method was developed to emulate traditional point counting, to compare the results obtained here with what would be obtained by a specialist. For each binary image to be counted, a grid with 500 x 500 equidistantly distributed points was created and overlapped with the maceral group results (an example for vitrinite can be seen in Figure 73). This is done for all images of a sample and individually for each maceral group. In the end, the point fraction for each maceral group is understood as a representative of its area fraction.

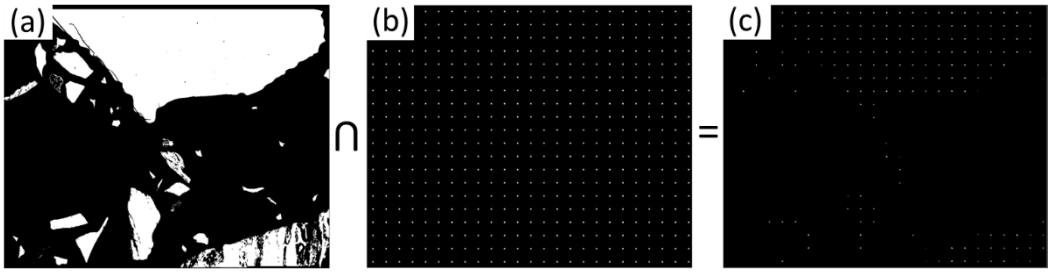

Figure 73- Vitrinite results after processing (a) intersected with a 500x500 points grid (b), resulting in numbers the points representative of the vitrinite composition (c). This was performed on both the ground truth (annotated images) and the results (segmented post-processed images).

The ground truth for comparison was obtained directly from the annotations made on the images. The results obtained from a direct area measurement with no grid points were also calculated. The results can be seen in Table 15.

Table 15 - Maceral group composition obtained by the models compared with that obtained from the dataset annotations.

Sample/Point fraction	Vitrinite (%)	Inertinite(%)	Liptinite(%)
Sample E (annotation)	93.25	6.71	0.03
Sample E (model/grid points)	97.54	2.28	0.18
Sample E (model/ total area)	96.13	3.67	0.20

Sample F (annotation)	85.68	14.32	0.00
Sample F (model/grid points)	97.44	2.48	0.08
Sample F (model/ total area)	95.31	4.58	0.11
Sample D (annotation)	76.40	22.81	0.78
Sample D (model/grid points)	88.97	10.60	0.42
Sample D (model/ total area)	88.69	10.82	0.49
Sample A (annotation)	84.77	7.15	8.08
Sample A (model/grid points)	92.08	5.53	2.40
Sample A (model/ total area)	90.10	6.84	3.06

The expected results reflect the test metrics of the inertinite and liptinite models. Due to their low recall, they cannot completely identify the maceral groups, which systematically underestimates both groups compared to the vitrinite. This method's results can't be quantitively compared to traditional point counting. However, it should still have internal consistency in that it will still point out which samples are richer in which maceral groups, even if the actual numbers can't be used for comparison.

However, it can be seen that emulating the point counting method had minimal impact, as the area fraction is very similar to the one measured directly from the post-processed mask. The results obtained without grid points were slightly more accurate, as they underestimated the liptinite and inertinite less than the grid point method. This is probably due to vitrinite having a larger area on average, which makes the grid points favor it.

An expansion of the dataset aiming at better recall rates for inertinite and liptinite remains the most logical solution to improve these results. A more oriented effort on annotating semifusinite and improving that model would allow it to be "split" from the total inertinite here and be quantified separately.

5.3 Collotelinite reflectance Determination

To have a system capable of dealing with single component coals in the range of 0.97%-1.8%, it is necessary to refine the results obtained from the vitrinite model. As the microscope was calibrated and the image capture procedure was standardized for all samples, it is possible to translate the reflectance range into a grey level range. The first step is to intersect the mask generated by the model with the original image (Figure 74). The second step is to segment the resulting image's pixels by a threshold between 85 and 145 grey levels, the same as in the area fraction analysis. This range corresponds to a slightly broader reflectance range than the one this work concerned itself with (corresponding to approximately 0.85%-1.85%), this is done to account for the low and high standard deviation of grey level values in the samples. This procedure filters out noise, scratches, some transition/border pixels, and, potentially, even a few other segmentation mistakes.

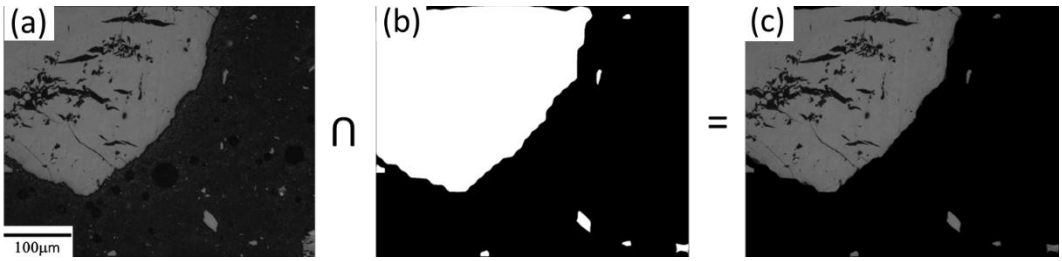

Figure 74 - Original grey image (a) intersected with the binary vitrinite result (b), resulting in a grey image containing only the model's segmented vitrinite(c). Notice how the dark inner pixels of mineral matter were excluded in the process.

That process is performed for all the images of each sample, so it is possible to obtain the grey level distribution of the collotelinite and convert it to a reflectance distribution using equation 12.

The results for the mean and mode of the reflectance distributions and their comparison to the traditional method can be seen in Table 16 and Table 17.

Table 16 - Results of the random collotelinite reflectance (Rr%) values, comparing the traditional and proposed (Mask R-CNN) methods for the four samples used to train the model.

	Traditional Method (Mean)	Mask R-CNN (Mean)	Mask R-CNN (Mode)
Sample E	1.20%	1.07% +- 0.14%	1.07%
Sample F	1.20%	1.05% +- 0.15%	1.07%
Sample D	1.04%	1.02% +- 0.13%	1.02%
Sample A	0.97%	0.98% +- 0.13%	0.97%

Table 17 - Results of the random collotelinite reflectance (Rr%) values, comparing the traditional and proposed (Mask R-CNN) methods for the samples used to test the model.

	Traditional Method (Mean)	Mask R-CNN (Mean)	Mask R-CNN (Mode)
Sample M	1.80%	1.60% + - 0.25%	1.84%
Sample L	1.51%	1.32% + - 0.24%	1.39%
Sample K	1.50%	1.42% + - 0.22%	1.51%
Sample J	1.47%	1.28% + - 0.18%	1.26%
Sample I	1.42%	1.28% + - 0.24%	1.41%
Sample H	1.26%	1.31% + - 0.24%	1.17%
Sample G	1.15%	1.34% + - 0.30%	1.04%
Sample C	1.08%	1.17% + - 0.26%	1.07%

One of the first things to notice in these results is how the mean and the mode seemed to be in better agreement with each other for the samples used in training and had a smaller standard deviation (Table 16). Even though this was performed in images for those samples that were not used in training, it is still understandable that the network would perform better on samples that it was more familiar with.

Another vital thing to notice is how the mode agrees better with the traditional method than the mean. That is noteworthy because the reflectance measurements standards (ASTM D 2798, 2011; ISO 7404-5, 2009; AS 2856.3, 2000) advise the use of the mean instead of the mode. And yet, for this method, the mode seems closer to the traditional mean.

To better understand why that would be the case, it was necessary to plot the reflectance distribution for all the samples. Figure 75 shows an example of such distribution for a 1.08% reflectance sample.

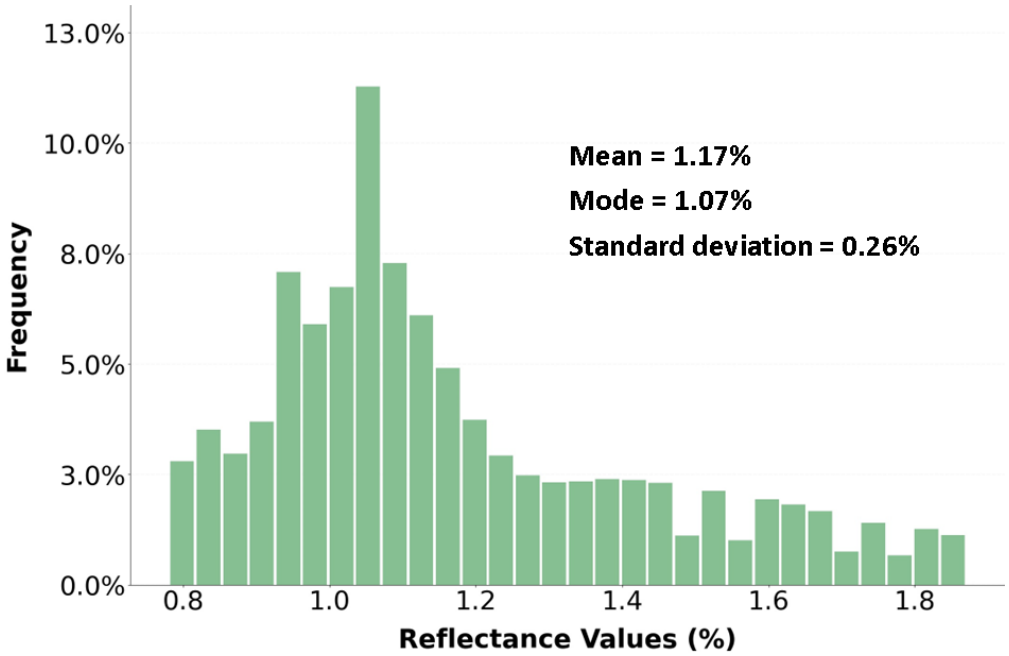

Figure 75 - Reflectance distribution for the 1.08% vitrinite reflectance sample C.

The distribution behavior exhibited was the typical one for all samples. They always have a single peak related to the segmented vitrinite and what seems to be a variable noise baseline, both below and above the vitrinite peak reflectance when compared to what is expected from a traditional petrographer analysis (Figure 17). The traditional petrographer usually analysis yields a histogram for a single component coal that only contains the vitrinite peak. To better understand this baseline, it was necessary to inspect the segmentation results of all the samples to find the patterns of segmentation errors that could be responsible for it.

Upon analyzing hundreds of images for different samples, the two major segmentation mistakes responsible for the broad reflectance distribution were finally identified. Figure 76 shows the rare case in which both error types were found in the same image. The first type of error (Figure 76c 1) is when the model segments a large collotelinite particle that might intercalate or juxtapose with other macerals, such as liptinites or small inertodetrinites. In this case, these macerals are segmented alongside the collotelinite object, contributing to the broadening of the reflectance distribution. The second type of error (Figure 76c 2) can be described as misclassified inertinite objects, which seems to happen occasionally, especially when inertinite particles are exceptionally homogeneous and have a smooth surface (which is the case of Figure 76c 2) and have a just slightly higher reflectance to that of vitrinite.

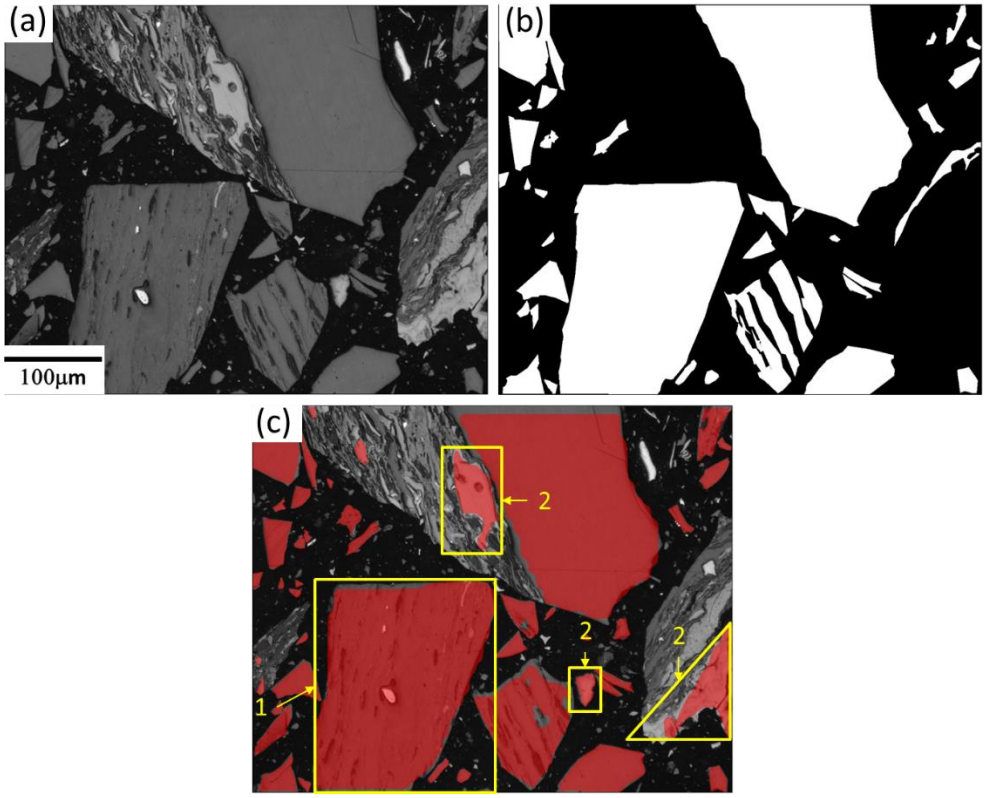

Figure 76 – Original image (a), binary ground truth for the collotelinite class (b), and the segmented collotelinite (c) for A. Figure 76c showcases the most common segmentation errors identified. Error 1 refers to maceral and mineral matter objects segmented alongside the collotelinite, and error 2 refers to inertinite objects misclassified as collotelinite.

Type 1 errors could be fixed in a post-processing procedure similar to the area fraction results by subtracting from the collotelinite the inertinite and liptinite segmentation results. Unfortunately, since neither model achieved a high recall, their effect on improving the collotelinite results would be limited for now. However, that remains a possibility in the future.

Type 2 errors can only be fixed upon improving the model, either by refining training parameters or expanding the dataset.

The transition pixels, border pixels, scratches, dirt, mineral matter, and liptinite associated with type 1 errors are probably responsible for the low reflectance baseline of noise that can be seen below the vitrinite peak in Figure 75. On the other hand, bright mineral matter, small inertodetrinites, and misclassified inertinite associated with both type 1 and 2 errors can be responsible for the high reflectance baseline of noise that can be seen after the vitrinite peak.

There would be many ways to treat the data found in Figure 75 to extract a representative reflectance value. One way would be to find the vitrinite peak automatically and extract its mean and mode. Another would be to model the noise baseline and subtract it from the distribution. However, the simplest way to obtain a representative value was to take the distribution mode over the mean. The

mode, unlike the mean, is impervious to the influence of the noise/segmentation errors and (as seen in Table 16 and Table 17) is in better agreement with the traditional method. Therefore, taking the mode as representative of the collotelinite reflectance distribution was chosen as the best and simplest way to handle this data.

Plotting the distribution histogram and extracting the mode is one of the many ways to treat the data. Other techniques, including from other works in the literature, could improve collotelinite reflectance determination even further. Using a Full Maceral Reflectogram (FMR) (O'BRIEN et al., 2003) on these results to extract the collotelinite reflectance could prove beneficial, as the segmentation step already filters out most noise from other macerals groups and other vitrinite macerals. This would combine the strengths of both approaches, and it could be an interesting future test.

It should be mentioned, however, that even if the segmentation were flawless, a perfect match between this method and the traditional method would still not be expected. The traditional method is a point-counting method, while this method segments areas. The conventional method can avoid borders, transition pixels, scratches, dirt, and relief effects, which allows its reflectance distribution to be much narrower than this method could ever achieve. By segmenting areas, a more heterogeneous and broader distribution will always be expected for this method.

Once the mode was chosen as the best representation of the collotelinite reflectance, it was possible to focus on evaluating how much it differed from the traditionally obtained. The metric chosen to evaluate that was the root mean square error (RMSE), which is a classical measurement of how much two different sets of points differ. The RMSE of 0.0978% was calculated for all samples in Table 16 and Table 17, and it can be considered outstanding as it is close to the reproducibility of the standard method, 0.06% (ASTM D2798, 2011). Reproducibility means that if two operators measure the same sample on different days in different labs and other equipment, their discrepancy is acceptable if below that value. As this method uses an entirely different approach, any result within 0.06% around the reference reflectance should be considered outstanding.

Random mean reflectance was the reference reflectance used, as it is the only one comparable to the developed methodology. Maximum mean reflectance measurements involve rotating the sample, which does not happen in this approach.

One possible limitation of this method is that it is more likely to "fail" or yield less reliable results in complex inertinite-rich samples if vitrinite and inertinite are harder to differentiate even by a human operator (usually, collotelinite vs. semifusinite). However, in those cases, the inertinite reflectance is expected to be similar to that of the vitrinite, which should limit any rank reflectance deviation.

As mentioned previously, this approach assumes a unimodal distribution of grey levels, which makes it unsuitable for dealing with blends containing coals with different ranks. But an analysis of grey level histogram should still indicate the presence of vitrinites of various ranks and allow for an estimation.

The use of 200X images in the current work allows for easier and faster acquisition of focused images, making an automatic image acquisition step easier to implement in the future. However, unlike other works that elected to acquire 500X images (WANG et al., 2019), this does come at the cost of reducing the information level contained in an image. It is likely that by acquiring 200X images, correct identification of fine macerals like micrinite might never be achievable.

As the results obtained here are directly related to the quality of the segmentation, improving it is crucial to refining these results. While the expansion of the dataset is critical for any improvement, an expansion focusing on inertinites and liptinites seems to be the most beneficial to reflectance results. Once inertinites and liptinite models with a high recall have been obtained, they could be used to severely decrease the occurrence of type 1 and 2 errors (Figure 76), refining the reflectance distribution by minimizing the baseline noise (Figure 75).

It is also essential to highlight the vast number of images used for training the models, with 260 images for five samples. Also, unlike many other works (CHAO et al., 1982; KUILI et al., 1988; O'BRIEN et al., 2003; TIWARY et al., 2020; WANG et al., 2019) that attempted to determine rank reflectance, this method focused on segmenting the collotelinite maceral instead of the whole vitrinite maceral group, which provides a more accurate rank reflectance measurement according to the standards for vitrinite reflectance measurements (ASTM D 2798, 2011; ISO 7404-5, 2009; AS 2856.3, 2000).

Another advantage to highlight is that the images were captured with a motorized stage microscope which stops at random positions in the samples according to the initial input for the stage setting, allowing the dataset to be as representative of an actual analysis as possible. Another important feature was that the root mean square of 0.0978% was obtained over 12 samples in the 0.97%-1.8% range, even though the collotelinite model was trained with only five samples in the 0.97%-1.20% range. That indicates the future potential of this approach, as it could predict the vitrinite reflectance, even outside of the vitrinite reflectance range it was trained in. Therefore, it is expected that expansion and improvement of the dataset will be essential to refine the results found herein even further and to make this system more universally appliable.

A summarized workflow detailing all the procedures performed on the results of the trained models can be seen in Figure 77.

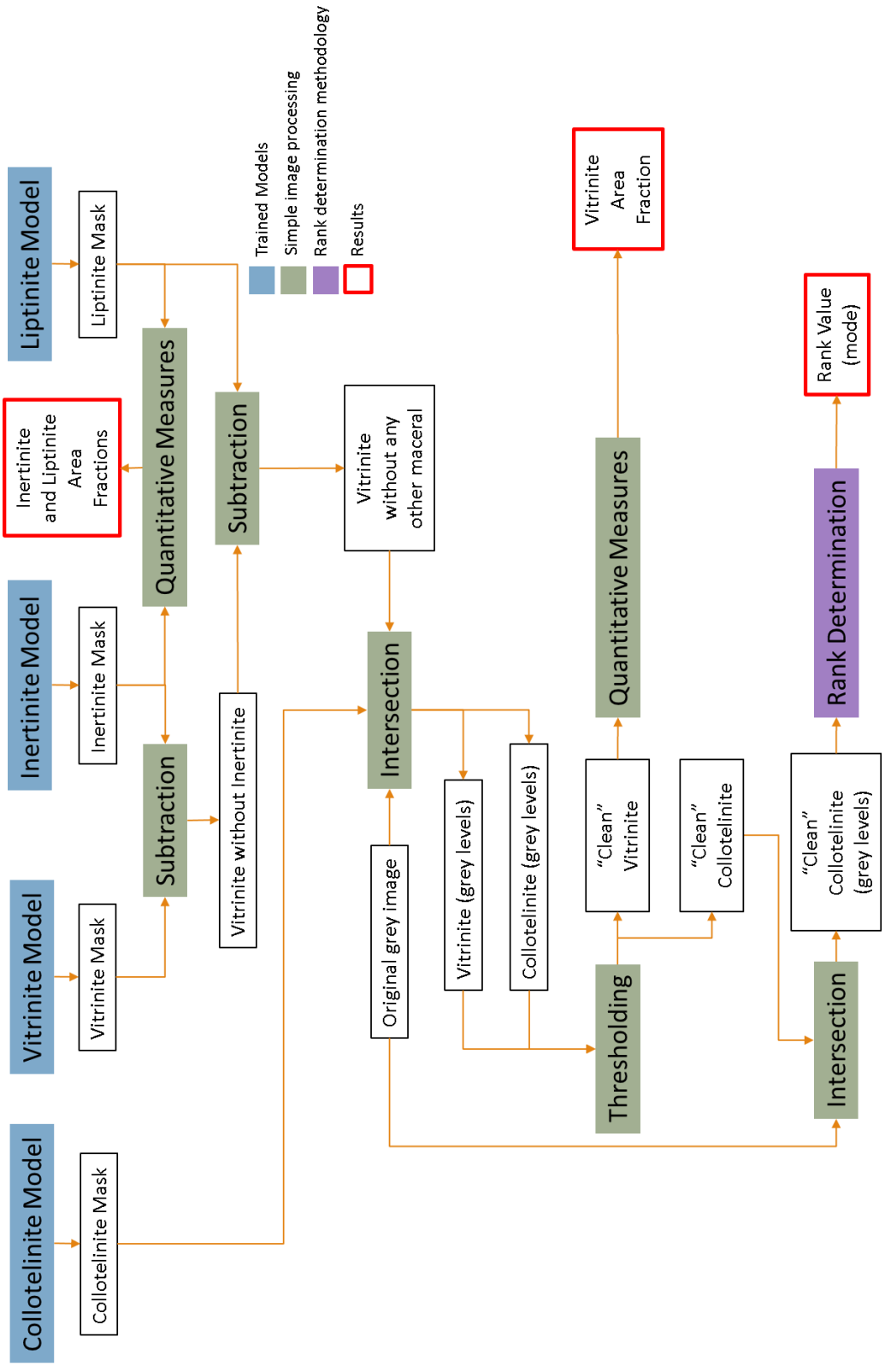

Figure 77 - Workflow detailing how the results of the models were processed to obtain area fraction and rank reflectance results.

6
Conclusions

This work aimed to create an automated system capable of identifying and segmenting macerals and maceral groups. An extensive dataset was built and used to train many different models. The best results were obtained for the vitrinite, inertinite, liptinite maceral groups, and collotelinite individual maceral (89.23%, 68.81%, 37.00%, 84.77% F1-score; respectively; 89.42%, 94.81%, 62.95%, 83.72% precision; respectively; 89.04%, 54.01%, 26.20%, 85.84% recall; respectively). While optimization of the training parameters for all models, the liptinite model had a special optimization. For the liptinite model, the pixel size anchors the network uses to identify regions of interest were adapted to better encompass the observed size distribution of the liptinites.

The results were considered satisfactory for both the vitrinite and collotelinite models but still underwhelming for the other two maceral groups. It is believed that a continuous and consistent dataset expansion is the most logical step toward improving those models.

These results are also reflected in the area fraction results, as inertinite and liptinite remain underestimated due to their models' low recall. This approach consistently overestimates the vitrinite while underestimating the other two maceral groups. This makes it impossible to be compared with the traditional point-counting method. However, comparing different samples within this method should still be possible.

It was not possible in this study to develop a methodology to satisfactorily quantify the semifusinite, as the model had a very low recall and precision. Possibly due to its complexity and similarity to other macerals, the only way to improve that would be an extensive annotation of even more semifusinite objects.

The collotelinite results, however, were good enough that they could be used to develop an additional rank determination method. After a short image processing step to refine these results, coal rank was assessed by selecting appropriate collotelinite pixels and selecting the mode of the grey values.

The root mean square error (RMSE) between the reference rank values and the ones determined by this method was evaluated to be 0.0978%. This is a good agreement despite the inherent differences between both methodologies. More time invested in expanding the dataset of coal sample images could improve the overall performance of this approach even further. Nevertheless, this method proves to be a reliable tool for efficiently determining coal rank in the 0.97-1.8% range and could be a powerful tool to save time for coal petrography analyses, as

the computers used in this work could examine up to 400 images in about 10 minutes, which would help coal petrographers do faster coal rank assessments.

Many future steps are still available to improve this work. Besides the tagging for expansion of the dataset, developing an automatic method for treating the reflectance distribution and find the collotelinite peak could lead for the post processing no longer being necessary. That would also allow the rank reflectance determination module to deal with samples containing multiple vitrinite populations (peaks), that is, coal blends. As the vitrinite model and the collotelinite models had good metrics, it would be interesting to extract a size distribution of the particles, as it would take advantage of the instance segmentation performed with the models.

Another possibility is using a more traditional method for maceral group identification, such as Coal Grain Analysis or Full Maceral Reflectogram, and focus the models on discriminating the macerals groups and subgroups afterwards. That would have the advantage of combining simpler reflectance-based methods for the maceral group segmentation (which is maceral-based) with more sophisticated morphology-based method (Machine Learning) for an individual maceral segmentation (as individual macerals are defined based on their morphology).

One more possibility is experimenting with other network architectures, possibly ones that focus on semantic segmentation. A simpler network for semantic segmentation might not require such an expansive dataset to produce better results.

7
Bibliographic references

AGUS, M.; BONIFAZI, G.; MASSACCI, P. **Image Texture Analysis Based Procedure to Characterize and Recognize Coal Macerals**, Minerals Engineering, vol 7, n:9, p. 1127-1147, 1994.

AMERICAN SOCIETY FOR TESTING & MATERIALS (ASTM). **Standard Test Method for Microscopical Determination of The Vitrinite Reflectance Of Coal, D2798** – 11a, 2011.

AMERICAN SOCIETY FOR TESTING & MATERIALS (ASTM). **Standard Test Method for Microscopical Determination of The Maceral Composition Of Coal, D2799** – 11a, 2011.

AUSTRALIA ST&ARDS (AS). **Coal - Maceral analysis**, AS 2856-1986, first edition, 1986.

BENJAMINI, Y.; HOCHBERG, Y. **Controlling the false Discovery rate: a practical and Power-full approach to multiple testing,** Journal of the Royal Statistics Society, v.57, p. 289-300, 1995.

BHARATI, P.; PRAMANIK, A. **Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey,** Computational Intelligence in Pattern Recognition Advances in Intelligent Systems and Computing, p. 657-668, 2019.

BOUVRIE, J. **Notes on convolutional neural networks[R],** Massachusetts Institute of Technology: Center for Biological and Computational Learning, Technical report, p. 38, 2006.

CHAO, E. C. T.; MINKIN, J. A.; THOMPSON, C. L. **Application of Automated Image Analysis to Coal Petrography**, International Journal of Coal Geology vol.2, p. 113-150, 1982.

CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; SCHIELE, B. **The cityscapes dataset for semantic urban scene understanding,** In: IEEE Conference on Computer Vision and Pattern Recognition (CTPR), 2016.

CLAIRE, D. **The Difference Between Artificial Intelligence, Machine Learning, And Deep Learning**. Last updated: 25/03/2019. [online] https://blog.digitalogy.co/the-difference-between-artificial-intelligence-machine-learning-and-deep-learning/.

DUTAA, A.; ZISSERMAN, A. **The VIA Annotation Software for Images, Audio, and Video,** Proceedings of the 27th ACM International Conference on Multimedia, four pages. https://doi.org/10.1145/3343031.3350535, 2019.

GIRSHICK, R. **Fast R-CNN**, arXiv: 1504.08083v2, 2015.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. **Region-based Convolutional Networks for Accurate Object Detection and Segmentation**, arXiv:1311.2524v5, 2014.

GU, W.; BAI, S.; KONG, L. **A review on 2D instance segmentation based on deep neural networks.** Image and Vision Computing 120. https://doi.org/10.1016/j.imavis.2022.104401, 2022.

HAENEL, M. W. **Recent progress in coal structure research,** Fuel, vol 71, p. 1211–1222, 1992.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques: Concepts and Techniques,** The Morgan Kaufman Series in Data Management Systems. Elsevier Science, 2011.

HE, H.; XU, H.; ZHANG, Y.; GAO, K.; LI, H.; MA, L.; LI, J. **Mask R-CNN based automated identification and extraction of oil well sites.** International Journal of Applied Earth Observation and Geoinformation 112. https://doi.org/10.1016/j.jag.2022.102875, 2022.

HE, K.; DOLLA, P.; GIRSHICK, R.; GKIOXARI, G. **Mask R-CNN,** arXiv:1703.06870v3 (2018), https://arxiv.org/abs/1703.06870, accessed on 25th September 2020

IGLESIAS, J. C. A.; SANTOS, R. B. M.; PACIORNIK, S. **Deep learning discrimination of quartz and resin in optical microscopy images of minerals,** Minerals Engineering, vol. 138, p.79–85, 2019.

INTERNATIONAL COMMITTEE FOR COAL & ORGANIC PETROLOGY (ICCP). **The new vitrinite classification (ICCP System 1994),** PII: SOO16-2361(97)00161-O, Fuel Vol. 77, No. 5, p. 349-358, (1998).

INTERNATIONAL COMMITTEE FOR COAL & ORGANIC PETROLOGY (ICCP). **The new inertinite classification (ICCP System 1994),** Fuel vol. 80, p. 459–471, 2001.

INTERNATIONAL ORGANIZATION FOR ST&ARDIZATION (ISO), **Methods for Petrographic Analysis of Coals- Part 3: Method of determining maceral group composition**, ISO 7404-3:2009, third edition. (2009).

INTERNATIONAL ORGANIZATION FOR ST&ARDIZATION (ISO), **Classification of coals**, ISO 11760:2018, second edition, 2018.

JARRETT, K.; KAVUKCUOGLU, K.; RANZATO, M. A.; LECUN, Y. **What is the best multi-stage architecture for object recognition?**, International Conference on Computer Vision (ICCV'09), 2009.

KIEFER, J; WOLFOWITZ, J. **Stochastic Estimation of the Maximum of a Regression Function,** Ann. Math. Statist. Vol. 23, Number 3, p. 462-466, 1952.

KUILI, J.; JIAN, X.; DUOHU, H. **The Use of Automated Coal Petrography in Determining Maceral Group Composition and the Reflectance of Vitrinite**, International Journal of Coal Geology vol. 9, p. 385-395, 1988.

LAI, W.; HU, F.; KONG, X.; YAN, P.; BIAN, K.; DAI, X. **The study of coal gangue segmentation for location and shape predicts based on multispectral**

and improved Mask R-CNN. Powder technology vol 407. ttps://doi.org/10.1016/j.powtec.2022.117655, 2022.

LAQTIB, S.; YASSINI, K. E.; HASNAOUI, M. L. **A deep learning method for intrusion detection systems-based machine learning in MANET,** SCA '19: Proceedings of the 4th International Conference on Smart City Applications Article No.: 43, p. 1–8 https://doi.org/10.1145/3368756.3369021. October 2019.

LIN, T. Y.; DOLLA, P.; GIRSHICK, R.; HE, K.; HARIHARAN B.; BELONGIE S. **Feature Pyramid Networks for Object Detection,** arXiv:1612.03144v2. 2017.

O'BRIEN, G.; JENKINS, B.; ESTERLE, J.; BEATH, H. **Coal characterization by automated coal petrography**, Fuel vol 82, p.1067–1073, 2003.

PETERSEN, H. I.; SCHOVSBO; N. H.; NIELSEN, A. T. **Reflectance measurements of zooclasts and solid bitumen in Lower Paleozoic shales, southern Scandinavia: Correlation to vitrinite reflectance.** International Journal of Coal Geology 114, p. 1-18, 2013.

PICKEL, W.; KUS, J.; FLORES, D.; KALAITZIDIS, S.; CHRISTANIS, K.; CARDOTT, B.J.; MISZ-KENNAN, M.; RODRIGUES, S.; HENTSCHEL, A.; HAMOR-VIDO, M.; CROSDALE, P.; WAGNER, N. **Classification of liptinite – ICCP System 1994,** International Journal of Coal Geology vol. 169, p. 40–61, 2017.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**, arXiv: 1506.01497v3, 2016.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. **Learning representations by back-propagating errors,** Nature vol. 323, p. 533–536, 1986.

SALAU, J.; KRIETER, J. **Instance Segmentation with Mask R-CNN Applied to Loose-Housed Dairy Cows in a Multi-Camera Setting**, Animals vol.10, 2102, doi:10.3390/ani10122402, 2020.

SMITH, G. C.; COOK, A. C. **Coalification paths of exinite, vitrinite, and inertinite**, Fuel vol. 59, p. 641-646, 1980.

STOPES, M. C. **On the Petrology of Banded Bituminous Coal**, vol 14, p. 4-12 ,1935.

SUÁREZ, I. R.; CRELLING, C. **Applied Coal Petrology-The Role of Petrology in Coal Utilization,** DOI- https://doi.org/10.1016/B978-0-08-045051-3.X0001-2, 2008.

SUÁREZ, I. R.; FLORES, D.; MENDONÇA, J. G. F.; HACKLEY, P. C. **Review and update of the applications of organic petrology: Part 1, geological applications,** International Journal of Coal Geology vol. 99, p. 54-112, 2012.

SYKOROVA, I.; PICKEL, W.; CHRISTANIS, K.; WOLF, M.; TAYLOR, G. H.; FLORES, D. **Classification of Huminite- ICCP System 1994,** International Journal of Coal Geology, vol 62, p. 85–106, 2005.

TAYLOR, G. H.; TEICHMÜLLER, M.; DAVIS, A.; DIESSEL, C. F. K.; LITTKE, R.; ROBERT, P. **Organic Petrology,** 14th edition, 1998.

TEICHMÜLLER, M.; TEICHMÜLLER, R. **Diagenesis of Sediments and Sedimentary Rocks- Diagenesis of Coal (Coalification)**, Elsevier, Amsterdam, p. 207–246, DOI: 10.1016/S0070-4571(08)71074-4, 1979

TIWARY, A. K.; GHOSH, S.; SINGH, R.; MUKHERJEE, D. P.; SHANKAR, B. U.; DASH, P. S. **Automated Coal Petrography Using Random Forest**, International Journal of Coal Geology 232 103629, 2020.

WANG, H.; LEI, M.; LI, M.; CHEN, Y.; JIANG, J.; ZOU, L. **Intelligent Estimation of Vitrinite Reflectance of Coal from Photomicrographs Based on Machine Learning**, Energies, 12, 3855 doi:10.3390/en12203855, 2019.

ZHOU, Y.; CHELLAPPA, R. **Computation of optical flow using a neural network**, In: IEEE International Conference on Neural Networks, p. 71-78, 1988.

8
Appendix

The author would like to add here the python code used for training the Mask R-CNN for the collotelinite model for future reference:

```
# -*- coding: utf-8 -*-
"""
Created on Wed Sep 11 17:22:55 2019

@author: julio
"""
import os
import json
import skimage.io
import numpy as np
import imgaug as ia
import skimage.draw
import skimage.color
import skimage.transform
from mrcnn.utils import Dataset
from mrcnn.config import Config
import imgaug.augmenters as iaa
from mrcnn.model import MaskRCNN

# Class that defines and loads the dataset
class MyDataset(Dataset):
    datadict = dict()
    debug = False
    def load_image(self,image_id):
```

```
    if image_id in self.datadict:
        if self.debug:
            print('load image ram dict :' + str(image_id) + '\n')
        return self.datadict[image_id]
    else:
        if self.debug:
            print('load image from disk :' + str(image_id) + '\n')
        return super().load_image(image_id)
# Load the dataset definitions
def load_dataset(self, dataset_dir, subset, memory_buffer = True):
    '''
    Load a subset of the dataset.

    :param  dataset_dir:    Root directory of the datasete
    :param  subset:         Subset to load: train or val
    :return:                "Load a subset of the dataset"
    '''
    # Define the classes
    self.add_class("dataset", 1, "Collotelinite")
    # Train or validation dataset?
    assert subset in ["train", "val"]
    # Define data locations
    dataset_dir = os.path.join(dataset_dir, subset)
    dataset_annots = dataset_dir + '/annots'
    annotations = json.load(open(os.path.join(dataset_annots,
                                 "via_export_json.json")))
    annotations = list(annotations.values())  # don't need the dict keys
    # The VIA tool saves images in the JSON even if they don't have any
    # annotations. Skip unannotated images.
    annotations = [a for a in annotations if a['regions']]
```

```
            # Find all images

            # Add images

            for a in annotations:

                # Get the x, y coordinaets of points of the polygons that make up

                # the outline of each object instance. There are stores in the
shape_attributes.

                class_id = [sub['id'] for sub in self.class_info] # find all class ids

                class_name = [sub['name'] for sub in self.class_info] # find all class
names

                shape_points = [r['shape_attributes'] for r in a['regions']]

                objects = [s['region_attributes'] for s in a['regions']]

                # Finds the corresponding class id in the json

                flag = False

                k = 0

                class_ids, polygons = [], []

                for n in objects:

                    k = k + 1

                    #position = list(n['Position'].keys()) # get the dict key = position
of the maceral

                    #if len(position) > 0 and position[0] == 'Matrix': # check only the
not empty position equal to "matrix"

                    try:

                        # Maceral and maceral group

                        #maceral_subgroup = list(n['Maceral Sub-Group'].keys()) # get
the dict key = the major maceral sub-group

                        #maceral_group = list(n['Maceral Group'].keys()) # get the dict
key = the major maceral group

                        maceral = list(n['Maceral'].keys()) # get the dict key = the major
maceral

                        if maceral is not None and len(maceral) > 0 and maceral[0] ==
'Collotelinite': # check only the not empty class and if the name of the maceral
group matches one of the possible classes

                            flag = True
```

```
                    polygons.append(shape_points[k - 1]) # get only the
polygons that satisfy the previous conditions
                    ids = class_id[1] # get maceral group class id
                    class_ids.append(ids) # get all maceral group class ids


                except KeyError:
                    print('Image name:', a['filename'])
                    print('A Maceral is missing!')
            # Load_mask() needs the image size to convert polygons to masks.
            image_path = os.path.join(dataset_dir + '/images', a['filename'])
            image = skimage.io.imread(image_path)
            height, width = image.shape[:2]
            # Add to dataset
            if flag:
                self.add_image(
                    "dataset",
                    image_id = a['filename'],  # use file name as a unique image id
                    path = image_path,
                    width = width, height = height,
                    polygons = polygons,
                    class_ids = class_ids)
                '''# Add image to dictionay
                if memory_buffer and image_id not in self.datadict:
                    self.datadict[image_id] = self.load_image_path(img_path)'''
        def load_image_path(self, img_path):
            '''Load the specified image and return a [H,W,3] Numpy array.
            '''
            # Load image
            image = skimage.io.imread(img_path)
            # If grayscale. Convert to RGB for consistency.
            if image.ndim != 3:
```

```
            image = skimage.color.gray2rgb(image)
        # If has an alpha channel, remove it for consistency
        if image.shape[-1] == 4:
            image = image[..., :3]
        return image

    # load the masks for an image
    def load_mask(self, image_id):
        '''
        Generate instance masks for an image.

        :param  image_id:       Image ID
        :return masks:          A bool array of shape [height, width, instance count] with one mask per instance
        :return class_ids:      A 1D array of class IDs of the instance masks
        '''
        # Get details of image
        info = self.image_info[image_id]
        class_ids = info['class_ids']
        # Convert polygons to a bitmap mask of shape
        # [height, width, instance_count]
        mask = np.zeros([info["height"], info["width"], len(info["polygons"])], dtype=np.uint8)
        for i, p in enumerate(info["polygons"]):
            # Get indexes of pixels inside the polygon and set them to 1
            rr, cc = skimage.draw.polygon(p['all_points_y'], p['all_points_x'])
            mask[rr, cc, i] = 1
        # Return mask, and array of class IDs of each instance.
        return mask, np.asarray(class_ids, dtype = 'int32')

    # Load an image reference
```

```
        def image_reference(self, image_id):
            '''
            Return the path of the image.

            :param  image_id:       Image ID
            :return info['path']:   Image path
            '''
            info = self.image_info[image_id]
            return info['path']


    # Define a configuration for the model
    class ModelConfig(Config):
        # Define the name of the configuration
        NAME = "Coal_cfg"
        # Number of classes (background + classes)
        NUM_CLASSES = 1 + 1
        # NUMBER OF GPUs to use. When using only a CPU, this needs to be set to 1
        GPU_COUNT = 4
        # Number of images to train with on each GPU
        IMAGES_PER_GPU = 1
        # Total number os images
        TOTAL_IMAGES = 172
        # Number of training steps per epoch
        # STEPS_PER_EPOCH = TOTAL_IMAGES / BATCH_SIZE where BATCH_SIZE = IMAGES_PER_GPU * GPU_COUNT
        STEPS_PER_EPOCH = int(TOTAL_IMAGES/(IMAGES_PER_GPU*GPU_COUNT))
        # Learning rate
        LEARNING_RATE = 0.001
        # Gradient norm clipping
        GRADIENT_CLIP_NORM = 5.0
```

```
    # Minimum probability value to accept a detected instance
    # ROIs below this threshold are skipped
    DETECTION_MIN_CONFIDENCE = 0 # was 0.7
    # Non-maximum suppression threshold for detection
    DETECTION_NMS_THRESHOLD = 0.3
    # Non-max suppression threshold to filter RPN proposals.
    # You can increase this during training to generate more propsals.
    RPN_NMS_THRESHOLD = 0.9
    # How many anchors per image to use for RPN training
    RPN_TRAIN_ANCHORS_PER_IMAGE = 512 # was 256
    # Maximum number of ground truth instances to use in one image
    MAX_GT_INSTANCES = 400 # was 100
    # Max number of final detections
    DETECTION_MAX_INSTANCES = 450 # was 100
    # Image mean (RGB)
    MEAN_PIXEL = np.array([64.0, 64.0, 64.0])
    # Input image resizing
    # Image size must be dividable by 2 at least 6 times
    # to avoid fractions when downscaling and upscaling.
    # For example, use 256, 320, 384, 448, 512, ... etc.
    IMAGE_RESIZE_MODE = "square"
    IMAGE_MIN_DIM = 1024
    IMAGE_MAX_DIM = 1344


def main():
    # Construct the argument
    path_data = '/mnt/panowin/hd/UsersPanoramix/Richard/python/script/Dataset/Coal/via_tagging/maceral_group/Vitrinite'
    path_model = '/mnt/panowin/hd/UsersPanoramix/Richard/python/pre-trained_models/mask_rcnn_coco.h5'
    from_zero = True
```

```
save_period = 5
epochs = 1300
layers = 'all'
# heads: The RPN, classifier and mask heads of the network
# all:   All the layers
# 3+:    Train Resnet stage 3 and up
# 4+:    Train Resnet stage 4 and up
# 5+:    Train Resnet stage 5 and up

# Prepare train set
print ('Loading dataset.......')
train_set = MyDataset()
train_set.load_dataset(path_data, 'train')
train_set.prepare()
print('Train: %d' % len(train_set.image_ids))
# Prepare val set
val_set = MyDataset()
val_set.load_dataset(path_data, 'val')
val_set.prepare()
print('Test: %d' % len(val_set.image_ids))
# Prepare config
config = ModelConfig()
config.display()
# Define the model
print ('Loading Model.......')
model = MaskRCNN(mode = 'training', model_dir =\
   '/mnt/panowin/hd/UsersPanoramix/Richard/python/script/',    config    =
config)
# Load weights (mscoco) and exclude the output layers
if from_zero:
   print ('Loading weights from zero.......')
```

```
            model.load_weights(path_model, by_name = True,
                               exclude = ["mrcnn_class_logits",
                                          "mrcnn_bbox_fc", "mrcnn_bbox",
"mrcnn_mask"])
        else:
            print ('Loading weights from last checkpoint.....')
            model.load_weights(model.find_last(), by_name = True)
        # Train weights
        print ('Start trainning session .....')
        model.train(train_set, val_set, learning_rate =
config.LEARNING_RATE, epochs = epochs,
                    layers = layers, chkpoint_save_period = save_period)


    if __name__ == '__main__':
        main()
```