steel research international
www.steel-research.de

Classification of Cast Iron Alloys through Convolutional Neural Networks Applied on Optical Microscopy Images

Marta Bárcena, Lara Lloret Iglesias, Diego Ferreño, and Isidro Carrascal*

Classification of cast iron alloys based on graphite morphology plays a crucial role in materials science and engineering. Traditionally, this classification has relied on visual analysis, a method that is not only time-consuming but also suffers from subjectivity, leading to inconsistencies. This study introduces a novel approach utilizing convolutional neural networks—MobileNet for image classification and U-Net for semantic segmentation—to automate the classification process of cast iron alloys. A significant challenge in this domain is the limited availability of diverse and comprehensive datasets necessary for training effective machine learning models. This is addressed by generating a synthetic dataset, creating a rich collection of 2400 pure and 1500 mixed images based on the ISO 945-1:2019 standard. This ensures a robust training process, enhancing the model's ability to generalize across various morphologies of graphite particles. The findings showcase a remarkable accuracy in classifying cast iron alloys (achieving an overall accuracy of 98.9 ± 0.4%—and exceeding 97% for all six classes—for classification of pure images and ranging between 84% and 93% for semantic segmentation of mixed images) and also demonstrate the model's ability to consistently identify and graphite morphology with a level of precision and speed unattainable through manual methods.

1. Introduction

Cast irons represent a significant classification among ferrous alloys, characterized by carbon contents exceeding 2.14 wt%. However, practical observations reveal that these alloys commonly contain carbon percentages between 3.0 and 4.5 wt%, along with other alloying elements. The study of the iron–iron carbide phase diagram shows that alloys within this range become completely liquid at temperatures approximately between 1150 and 1300 °C. These values are significantly lower than those observed for steels, thereby simplifying the melting process and favoring casting. Notably, certain types of cast irons exhibit pronounced brittleness, which, in turn, positions casting as an optimal fabrication technique for these materials.[1,2] The thermomechanical properties of cast iron are directly influenced by the morphology of its graphite particles. Factors such as fracture toughness and ductility are strongly linked to the shape of these graphite particles. Particles exhibiting a nodular shape enhance these properties, while particles that are more elongated or have irregular contours can negatively impact the material due to points of stress concentration. Consequently, the classification of cast iron is primarily determined by the morphologies of its graphite particles.[3]

Iron cast alloys can exhibit various microstructures depending on their composition and the specific conditions under which they are processed. The main types of microstructures found in these alloys are ferrite, pearlite, martensite, austenite, and graphite in various forms (such as flake, nodular, and compacted). Ferrite is a form of iron with a body-centered-cubic crystal structure. It is soft, ductile, and has low strength and hardness. The presence of ferrite can be increased in cast iron by annealing or normalizing the alloy.[1,2] Pearlite is a two-phased, lamellar (layered) structure composed of alternating layers of ferrite and cementite (Fe_3C). The lamellar structure gives pearlite its characteristic appearance under the microscope. Its properties are intermediate between ferrite and cementite.[2,4] Martensite is a hard, brittle structure that is formed when austenite is rapidly cooled (quenched). The rapid cooling traps carbon atoms within the iron lattice, leading to a distorted, body-centered-tetragonal structure.[2,5] Austenite is a form of iron with a face-centered-cubic crystal structure. It is stable at high temperatures but can transform to other microstructures such as ferrite, pearlite, or martensite under different cooling rates.[2,6] The structure of graphite in cast irons can take different forms. Flake graphite iron has thin, flat flakes of graphite. Nodular or ductile iron has small, spherical nodules of graphite. Compacted graphite iron (CGI) has intermediate shapes between flake and nodular graphite. The morphology of the graphite greatly affects the mechanical properties of the alloy.[2,7]

M. Bárcena, D. Ferreño, I. Carrascal
LADICIM (Laboratory of Science and Engineering of Materials Division)
University of Cantabria. E.T.S. de Ingenieros de Caminos
Canales y Puertos, Av. Los Castros 44, 39005 Santander, Spain
E-mail: ferrenod@unican.es

L. Lloret Iglesias
IFCA (Instituto de Física de Cantabria)
University of Cantabria - CSIC
Av. Los Castros s/n, 39005 Santander, Spain

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/srin.202400120.

DOI: 10.1002/srin.202400120

Iron cast alloys are differentiated based on their composition and the presence of certain microstructures, especially the morphology of graphite. The primary types are described hereafter. Gray iron is the most common type of cast iron and is so named due to the gray color of the fracture surface. This is because the carbon is present in the form of flake graphite or lamellae. The presence of the flake graphite makes the iron relatively soft and brittle but with good damping capacity and high thermal conductivity.[1,2,7] In white iron the carbon is combined chemically as iron carbide or cementite, which is hard and brittle. As a result, white cast iron is also hard and brittle, but it is highly resistant to wear. It gets its name from the white, crystalline appearance of the fracture surface.[1,2] Nodular iron, also known as ductile iron or spheroidal graphite iron, has the carbon present in the form of small spheres or nodules. This gives nodular iron high ductility and impact strength, and it can be heat treated to further improve wthese properties.[1,2,8] Malleable iron is initially cast as white iron and then heat treated (annealed) over several days to convert the brittle cementite into more ductile iron and graphite clusters. This process gives malleable iron good ductility and toughness.[1,2,9] CGI has graphite present in a compacted, vermicular (worm-like) form. The properties of CGI are intermediate between those of gray iron and nodular iron; it has higher strength and thermal conductivity than nodular iron, but lower ductility.[1,2,10]

The standard ISO 945-1:2019 (Microstructure of cast irons—Part 1: Graphite classification by visual analysis)[11] specifies a method of classifying the microstructure of graphite in cast irons by means of a comparative visual analysis. The reference images given in the standard provide a basis for visually classifying graphite forms. As of today, classification of microstructures and identification of their constituents has primarily relied on the expertise of human analysts, due to the observed diversity in morphological patterns within microstructure imagery. Consequently, the inherent subjectivity of this method poses significant limitations to the repeatability and precision of the process. The implementation of stereological principles in microscopic image analysis has driven the evolution of quantitative metallography. Fundamental parameters such as the shape and spatial distribution of microstructure elements have proven important in predicting certain macroscopic properties.[12–14]

Recent advancements in object morphology characterization, based on automated methodologies and artificial intelligence (AI), have significantly propelled the progress in recognition and classification of microstructure images. Recent[15–18] provides examples of the application of deep learning (DL) and machine learning (ML) techniques in local feature extraction and classification. In a recent publication, Iacoviello et al.[19] introduced a hybrid methodology that integrates the conventional method for microstructural classification of ductile cast irons with ML algorithms. Initially, image segmentation analysis was employed to identify individual nodules, followed by the measurement of numerous morphological properties. Subsequently, a support vector machine (SVM) classifier was trained in the second phase to categorize each specimen according to the standard ASTM A247-16a (Standard Test Method for Evaluating the Microstructure of Graphite in Iron Castings).[20] The objective of this study is to establish a distinct signature for each type of specimen, facilitating automatic and objective data classification.

Recent findings[21,22] confirm that convolutional neural networks (CNNs) demonstrate substantial benefits over traditional ML algorithms, like SVMs or ensembles of trees, in image classification tasks. The adoption of CNNs in this study is motivated by their superior ability to analyze and classify image data, especially within the complex, high-dimensional domain of optical microscopy images of cast iron alloys. CNNs distinguish themselves as the leading methodology in image processing through several mechanisms: their capacity for autonomous feature learning allows for the extraction of intricate patterns and structures directly from raw pixel data, circumventing the need for manually defined features. This is crucial for identifying complex morphological characteristics in materials science. Additionally, their architectural design promotes the development of hierarchical spatial representations, facilitating an understanding of both local and global spatial relationships within images, essential for tasks such as identifying various graphite spot morphologies. Finally, their use of parameter sharing and local connectivity grants them translation invariance, ensuring consistent feature detection across varying positions and orientations within samples, thus bolstering their robustness and generalization in analyzing microscopy imagery.[21,22] The paper by Che et al.[23] serves as a valuable reference for the use of CNNs in microstructure analysis of alloy materials. It highlights the significance of deep learning techniques in accurately classifying microstructures and performing image segmentation tasks, which align with our utilization of CNNs for categorizing graphite spots in cast iron alloys. Additionally, it addresses the broader implications and challenges of deep learning in alloy material applications, providing essential context for our research. In addition, the study conducted by Szatkowski et al.[24] explores the efficacy of ML techniques for classifying cast iron microstructures through optical microscopy. Addressing the challenge of classifying microstructures with varying shapes, sizes, and orientations, the research compares the performance of traditional classifiers like SVMs and random forests with advanced CNNs, specifically Faster R-CNN and Mask R-CNN. The study's findings underscore the superiority of CNNs in handling the complexity and high dimensionality inherent in microstructure images, demonstrating their ability to learn and classify features more accurately than traditional methods. However, despite these advantages, maximizing CNN potential necessitates large labeled training datasets, a requirement often challenging to meet in the context of cast alloys' microstructures. Data augmentation methods have been successfully implemented by Sarrionandia et al.[25] who proposed a framework that combines classical machine vision for feature extraction with deep learning for the objective classification of microstructural images of nodular cast iron. Their study introduces data augmentation techniques to enhance the training of deep learning models, thereby addressing the challenge of limited and imbalanced datasets commonly encountered in metallography. The classification carried out by these authors was conducted through two pretrained VGG16 and VGG19 models (trained on the ImageNet dataset) and a third one with a custom architecture based on a net used to classify letters and numbers.

To the authors' understanding, no prior research has focused on creating a CNN-based classifier for graphite microstructures in cast irons. This study tackles the challenge of insufficient

labeled datasets by generating a substantial volume of synthetic samples. This is achieved through the segmentation and data augmentation of reference micrographs provided by the standard ISO 945-1:2019.[11]

The remainder of this article is organized as follows. Section 2: Methodology outlines the procedural framework of our research. It begins with a detailed explanation of the ISO 945-1:2019[11] standard for graphite classification in cast irons, followed by an overview of the key features of artificial neural networks (ANNs) used for image classification, with a particular focus on CNNs. We then delve into the specific analytical tasks performed during the study, including data preprocessing, augmentation, and the development and training of the CNN models. In Section 3: Results, we present the outcomes of our analysis. We first describe the results of using the pretrained MobileNet CNN for the classification of pure images. Next, we discuss the findings from the semantic segmentation approach using the U-Net model, which classifies each pixel within the images. Detailed accuracy metrics and performance evaluations are provided for both models. Section 4: Discussion interprets the results, comparing them with existing literature and highlighting the significance of our findings in the context of cast iron classification. We discuss the implications of using deep learning models for this task, the advantages over traditional methods, and potential limitations and future research directions. The article concludes with a summary of the main achievements and contributions of our research in Section 5: Conclusion. We reiterate the effectiveness of the CNN models developed, their accuracy, and their potential impact on the field of materials science and engineering. By following this structured approach, we aim to provide a comprehensive and coherent narrative of our research, from the foundational methods to the significant findings and their broader implications.

2. Experimental Section

This section presents an outline of the methodologies adopted in this research, with an emphasis on the reproducibility of its results. It comprises a depiction of the ISO 945-1:2019[11] standard for graphite classification in cast irons (Section 2.1), an overview of the principal variants and attributes of ANNs for image classification, with a focus on the superiority of CNNs (Section 2.2), and a detailed description of the analytical tasks performed in the course of the study (Section 2.3).

2.1. The Procedure of Standard ISO 945-1:2019

Graphite classification via visual analysis is a well-regarded approach employed by the foundry industry for the rapid determination of the microstructure in cast iron castings. This procedure involves the examination of a representative area of polished samples under a microscope. The graphite's form is determined through a comparative analysis with the reference images provided in ISO 945-1:2019.[11] The advised magnification for this evaluation was x100, primarily to assess the form and distribution of graphite. However, this magnification can be modulated as per necessity to closely align with the corresponding images before initiating the classification of the graphite form and distribution. The graphite was categorized by its form, utilizing Roman numerals from I to VI for designation. As mandated by the standard, the evaluation of the results procured from this analysis should be performed by an operator proficient in this specific metallographic technique.

As previously noted, a set of reference images demonstrating graphite microstructures were utilized in the classification process of graphite form in cast iron. For the purposes of clarity and to support subsequent analysis, these images are reproduced in **Figure 1**. The classification system consists of six distinct classes, described as follows.[3] 1) Class I particles are indicative of gray cast iron. 2) Class II particles, known as crab or spiky due to their morphology, are not attributed to a particular type of cast iron. These particles originate from the degradation of Class VI particles under conditions of impurity presence, excess nodulizing constituents during manufacturing, or rapid cooling of hypereutectic gray irons. 3) Class III, encompassing CGI, represents graphite particles exhibiting a form intermediate to Classes I and VI or equivalently, between gray and nodular cast iron. 4) Class IV, V, and VI particles are called, respectively, irregular nodular, indistinct nodular, and regular nodular or spheroidal. While Classes IV and V are associated with malleable cast iron, Classes V and VI typify nodular cast iron.

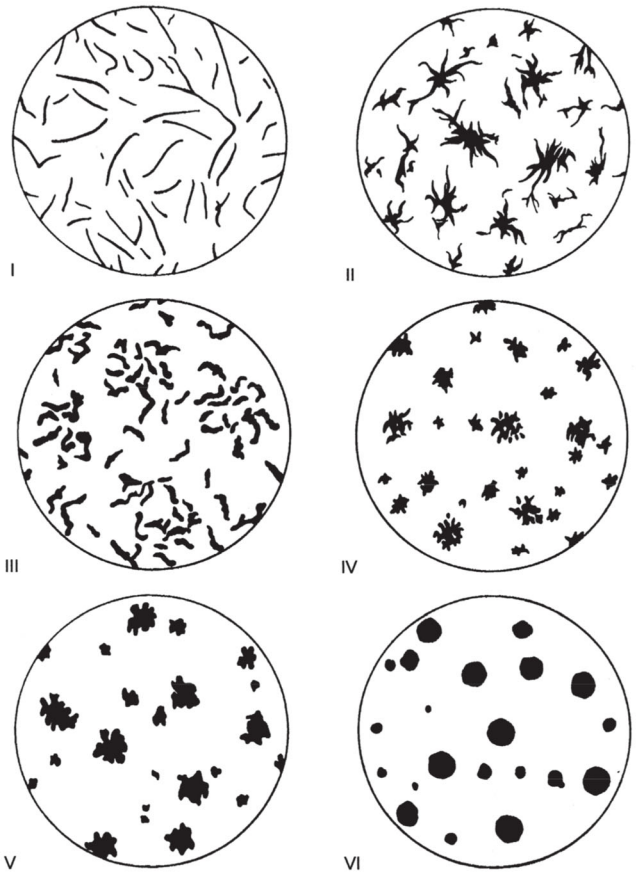

Figure 1. Reference images for the principal graphite forms in cast-iron materials as established by the standard ISO 945-1:2019.[1] The recommended magnification of these pictures is x100.

2.2. Artificial Neural Networks for Image Classification

ANNs are a subset of ML algorithms inspired by the human brain. They aim to replicate the brain's ability to learn from and interpret sensory data through a process of ML and pattern recognition. ANNs are structured in layers made up of interconnected nodes, or artificial neurons, which are inspired by biological neurons. Each connection between nodes can transmit a signal from one neuron to another, and the receiving neuron processes the signal and signals to downstream neurons. There are several types of ANNs, each with their unique architectures and use-cases. The most prominent types are summarized.[21] 1) Multilayer perceptron (MLPs) is the most basic type of neural network. In this architecture, information moves in only one direction—forward—from the input layer, through the hidden layers, to the output layer. 2) CNN is a class of deep, feed-forward ANNs specifically designed for processing grid-structured data, such as images, where spatial relationships between the data points matter. CNNs have proven very effective in image recognition and classification tasks. 3) Recurrent neural networks (RNNs) have connections that form directed cycles. This creates a form of internal memory which allows them to be very effective when dealing with sequential data like time series, speech, or text. Long short-term memory networks are a subtype of RNN that have a special mechanism that helps them learn long-term dependencies, making them particularly well-suited for tasks involving sequences of data with important long-range temporal context, such as language translation or handwriting recognition. 4) Generative adversarial networks (GANs) consist of two neural networks: the generator, which generates new data instances, and the discriminator, which evaluates them for authenticity. GANs can learn to create new data that is similar to the input data.

As mentioned above, CNNs are primarily used for analyzing visual imagery. The architecture of a CNN is designed to take advantage of the 2D structure of an input image. This is achieved with the use of a special kind of layer that performs a convolution. In summary, a convolution is a mathematical operation that slides a filter or kernel over the input data and performs element-wise multiplication and summing to produce a different representation of the input, often reducing dimensionality and capturing local dependencies in the data.

CNNs exhibit several advantages[21,22] over conventional ML algorithms (such as SVMs or ensembles of trees) in image classification tasks. 1) Hierarchical feature learning: CNNs automatically learn hierarchical representations. Lower layers of the network learn to detect simple features such as edges, while higher layers compose these lower-level features into more complex representations. Conventional methods, on the other hand, often require manual feature extraction or engineering. 2) Translation invariance: Due to their architecture, CNNs have a degree of translational invariance, meaning they can recognize a feature regardless of its location in the image. Traditional ML methods often treat input features independently and may fail to recognize a feature if its location in the image changes. 3) Performance: CNNs generally outperform conventional algorithms on complex image classification tasks, especially when large labeled datasets are available for training. 4) End-to-end learning: With CNNs, an end-to-end learning is possible. They can take raw pixel data as input and output class labels, eliminating the need for preprocessing or feature extraction steps that are often required with conventional algorithms. 5) Robustness to overfitting: With the use of techniques like dropout, data augmentation, and early stopping, CNNs can be more robust to overfitting compared to conventional algorithms, particularly on high-dimensional image data.

However, it's also important to note that CNNs require more computational resources and data to train compared to traditional ML methods. They can also be less interpretable than simpler models, which can be a disadvantage in applications where understanding the model's decision-making process is important.

CNNs often contain three types of layers: convolutional layers, pooling (downsampling) layers, and fully connected layers.[22] 1) Convolutional layers apply a specified number of convolution learnable filters (matrices of weights) to the image. For each subregion, the layer performs a set of mathematical operations to produce a single value in the output feature map. Convolutional layers then typically apply a rectified linear unit (ReLU) activation function to the output to introduce nonlinearities into the model. 2) Pooling layers downsample the image data extracted by the convolutional layers to reduce the dimensionality of the feature map in order to decrease processing time. A commonly used pooling algorithm is max pooling, which extracts subregions of the feature map (e.g., 2×2-pixel tiles), keeps their maximum value, and discards all other values. 3) Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional MLP neural network.

Spatial feature loss is the great drawback of MLPs as compared to CNNs for image recognition. Before feeding an image to the hidden layers of an MLP, the image matrix must be flattened to a 1D vector throwing away all the 2D information contained in the image. CNNs do not require a flattened image; rather, a raw image matrix of pixels is fed to a CNN network, and the CNN will understand that pixels that are close to each other are more heavily related than pixels that are far apart.

2.2.1. Activation Function

An activation function in a neural network defines the output of a neuron given a set of inputs. Biologically inspired by activity in human brains, where different neurons are activated by different stimuli, these functions are used to add complexity to the learning models of an ANN. By applying a nonlinear transformation, activation functions allow neural networks to learn from complex patterns. There are several types of activation functions used in neural networks, each with its own use case and properties. These include step functions, sigmoid functions, hyperbolic tangent (tanh), ReLU, and softmax functions, among others. The activation functions of the CNNs developed in this study are ReLU for the internal layers while the last one uses softmax, recommended for multilabel classification, that turns numbers into probabilities that sum to one. The class corresponding to the maximum probability is identified with the prediction ("winner takes all").

2.2.2. Loss Function

A loss function is used to compute the discrepancy between the predicted outcome of a ML model and the actual, true outcome. The goal of training a model is to find parameters that minimize this loss function. There are various types of loss functions used in ML, including, among others, mean squared error for regression problems, binary cross-entropy for binary classification problems, and categorical cross-entropy for multiclass classification problems, as in this study. Given a prediction probability distribution and the true distribution, the categorical cross-entropy loss calculates the average number of bits needed to identify an event (a class label) from a set of possibilities, if a coding scheme is used based on the prediction probabilities as opposed to the true distribution.

2.2.3. Optimizer

During the optimization process, a neural network iteratively processes the dataset over multiple epochs. Neuron weights, initially set at random, are updated following each epoch to reduce the error. This procedure ceases once the loss function is minimized or upon reaching a predetermined number of epochs. Weight updates are typically managed by gradient descent-based optimizers. The learning rate, which governs the magnitude of each update, is balanced to optimize computational cost and convergence. Efficiency is further improved through the use of mini-batch gradient descent, where gradients are computed on dataset subsets specified by the batch size. This batch size is generally determined by memory limitations.

In the current study, we employed two optimization algorithms commonly used in neural network models, namely RMSProp and Adam. RMSProp, or root mean squared propagation, is a gradient descent-based algorithm intended to expedite the optimization process by applying distinct learning rates for each parameter. Conversely, the Adam algorithm combines features of stochastic gradient descent and RMSProp. It borrows the weight updating mechanism based on the training batch size from the former while adopting the variable learning rate per parameter from RMSProp. This combination results in an enhancement in model performance, making Adam the prevailing choice for optimization in neural network models due to its rapid convergence rate.

2.2.4. Transfer Learning

Transfer learning is a ML technique where a pretrained model, typically developed for a large-scale task such as image classification on a dataset like ImageNet, is utilized as a starting point for a similar but typically smaller and more specific task. The pretrained model is often fine-tuned on the new task, adjusting the pretrained weights slightly to adapt them to the specific features of the new task. This concept is based on the notion that knowledge gained while solving one problem can be applied to a different but related problem. Transfer learning has several advantages compared to conventional learning. 1) Efficiency: Transfer learning can significantly reduce the computational resources and time required to train models, as the initial layers of the model have already been trained on a large dataset and do not require additional training from scratch. 2) Lower data requirement: Since the pretrained model has already learned useful features from a large dataset, transfer learning can be particularly useful when the new task has limited training data. 3) Improved performance: Transfer learning often leads to better performance in tasks with limited data, as the model benefits from features learnt from a large-scale task.

Several pretrained models using CNNs are popular for tasks such as image classification, object detection, and segmentation. Among others, a few widely used are VGG (developed at the University of Oxford), ResNet (Microsoft Research), InceptionV3, Inception-ResNet, MobileNet, EfficientNet, and Xception (Google). These models were pretrained on the ImageNet dataset, a large-scale, diversified database of images with 1000 classes, and are often used as a starting point in transfer learning for new tasks.

In the context of transfer learning, "frozen layers" refer to the layers of a pretrained model that are not updated, or trained, during the training process on a new task. The rationale behind freezing the initial layers of the model is based on the hierarchical feature learning nature of deep neural networks. In tasks such as image classification, the earlier layers often learn low-level features such as edges and color blobs, which are generally useful across different tasks. The later layers of the network, on the other hand, learn high-level features that are more task-specific. When adapting a pretrained model for a new task, the high-level, task-specific features of the original task might not be applicable to the new task. Thus, these layers are often replaced or reinitialized and then trained on the new task. The lower layers, which learnt general features, are often left as is, or "frozen", to take advantage of the features they have already learnt. This technique is particularly recommended for situations where the dataset is small and the images are very different from the original set on which the base model was trained, as in the article.

2.2.5. Semantic Segmentation

Semantic segmentation refers to the process of partitioning an image into multiple segments where each segment corresponds to a specific class or label. Unlike simple image classification, where the entire image is assigned a single label, semantic segmentation assigns a label to each pixel in the image, resulting in a detailed, pixel-level classification. In this context, a "mask" refers to an array or an image where each pixel is assigned a label indicating the class to which that pixel belongs. Each unique label represents a different class, so all pixels in the image that belong to a certain class have the same label in the mask. Masks are the output of semantic segmentation tasks. They provide a visual way to represent the result of the segmentation, where each class can be assigned a different color for visualization. When overlaid with the original image, these masks show precisely where each object (class) is located within the image. The masks are in turn transformed into semantic labels, in which each pixel is assigned a label according to the class to which it belongs, and each class is represented by an integer number. **Figure 2** shows the difference between a mask and a semantic label for one of the images used

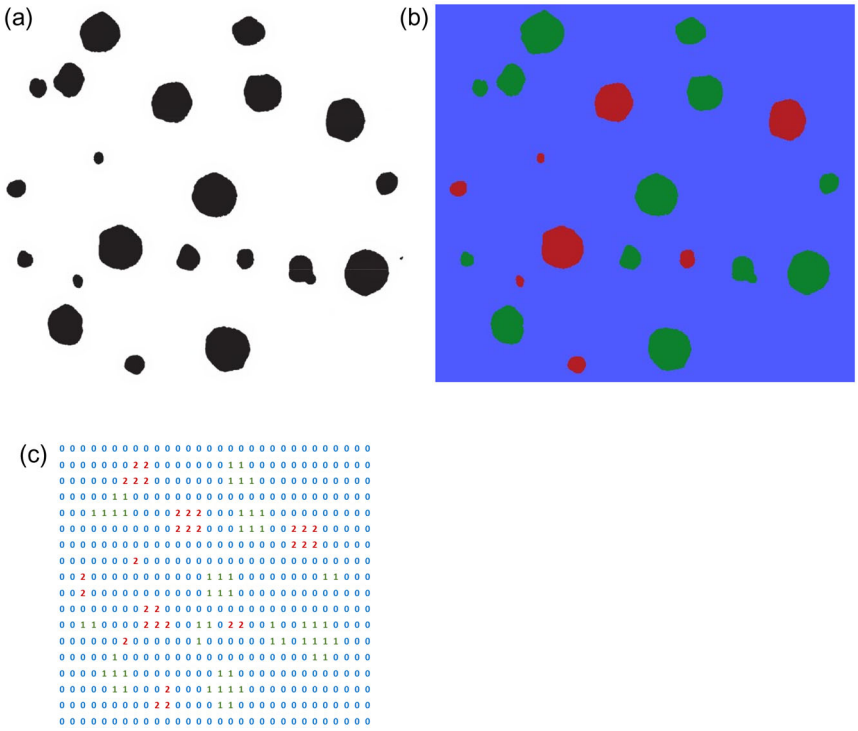

Figure 2. a) The black and white input image, b) its mask, and c) the corresponding semantic label. Note that the image has three classes, the blue background (class 0) and two classes of spots (class 1 and 2, respectively).

in this study (for simplicity, the semantic label represents a smaller number of pixels than the original image).

Semantic segmentation allows for a more detailed and nuanced analysis than image classification or object detection because it identifies the class and location of every single pixel in an image. It has many practical applications, including autonomous driving (for identifying road, vehicles, pedestrians, etc.), medical imaging (for identifying different tissues, anomalies, etc.), and image editing. Prominent techniques and architectures for semantic segmentation include fully convolutional networks (FCNs), SegNet, U-Net, DeepLab, and Mask R-CNN, among others. These networks typically use a series of convolutional layers to extract features from images, followed by upscaling layers to generate a full-size output image where each pixel is assigned a class label.

2.3. Analytical Scope

This section delineates the analytical scope of the study, providing a detailed overview of the methods employed, the data utilized, the specific steps taken, and the technical tools employed to develop the CNNs for classification of cast irons.

2.3.1. Schematic Description

In this research, two parallel CNNs were developed for categorizing primary carbon morphologies in cast iron specimens, grounded on the six reference micrographs provided by ISO 945-1:2019[11] and displayed in Figure 1. **Figure 3** outlines the flowchart followed in this study. A central challenge ML is the conflict between optimization and generalization. Optimization concerns the procedure of refining a model to achieve the maximum feasible performance on the training

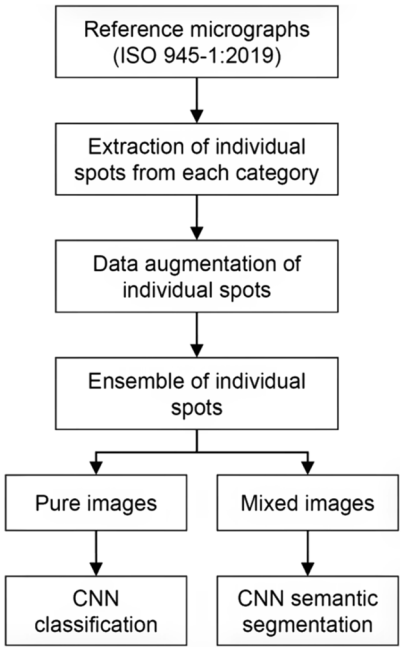

Figure 3. Schematic workflow showing the main steps for the development of the CNNs for classifying the graphite microstructure in cast irons.

data, and failure to accomplish this results in underfitting, where the model inadequately fits the training data due to its simplicity against the complexity of the data. Conversely, generalization refers to the performance of the trained model on previously unseen data. Overfitting arises when the model ideally fits the training dataset but is inept at generalizing to new, unseen data samples. This situation may arise in this study considering that, in principle, the dataset only contains the six images reproduced in Figure 1. In comparison to human learners who can identify and recognize objects after a few exposures, machines require comprehensive training data, varying from hundreds to millions of samples, particularly for more complex objects.[22] To bridge this gap, in this study a considerable dataset of synthetic samples, extracted from the reference images reproduced in Figure 1, was constructed. As illustrated in Figure 3 and detailed in Section 2.3.2, this process entails three phases: first, isolating individual spots from each of the reference images provided by ISO 945-1:2019[11]; second, manipulating each spot via a range of geometric transformations for data augmentation to create synthetic spots; and final, randomly assembling these synthetic spots to generate a sufficient quantity of synthetic samples. Two distinct datasets were built: one consisting of "pure images," or pictures composed of synthetic spots from the same category of ISO 945-1:2019[11] and another combining variable proportions of synthetic spots from two sequential categories, referred to as "mixed images." Each of these datasets was modeled using a specific CNN. The first model was trained to classify pure images using a pretrained CNN, whereas the second CNN was trained on the dataset of "mixed images" with the objective of classifying each pixel within the image through a semantic segmentation approach.

2.3.2. Preprocessing of the Reference Micrographs Provided in the Standard ISO 945-1:2019

CNNs demand that input images possess uniform dimensions, requiring prior preprocessing and scaling to identical widths and heights. In this study, the input pure images were all 224×224 pixels while mixed images were 128×128. As illustrated in Figure 1, each of the reference images comprised numerous individual spots. The initial phase of generating the synthetic sample dataset involved identifying and segregating each spot corresponding to a particular category. The extraction of each of the individual spots was performed in a Python script with the OpenCV library. This script took each of the images in the standard and identified the contour of the spots, which were essentially sets of dark pixels. Afterward, each of the spots was saved in a new image. Spots in contact with edges of the images were removed from the dataset to avoid introducing spurious information. The spots surrounded by blue boxes in **Figure 4** are examples of spots removed for this reason.

Subsequently, data augmentation—a preprocessing technique—was employed, which entailed the creation of synthetic spots which were varied versions of the original individual after being subjected to reliable geometric transformations. This not only expanded the dataset but also introduced the neural network to a multitude of image variations,[22] thereby enhancing the CNN's classification proficiency and mitigating the likelihood of overfitting. **Table 1** describes the six types of augmentation methods applied and their respective ranges while Figure 4 shows an example of the transformations to which a specific spot was subjected.

Upon transformation, the individual synthetic spots were randomly consolidated into two sets of synthetic samples, forming

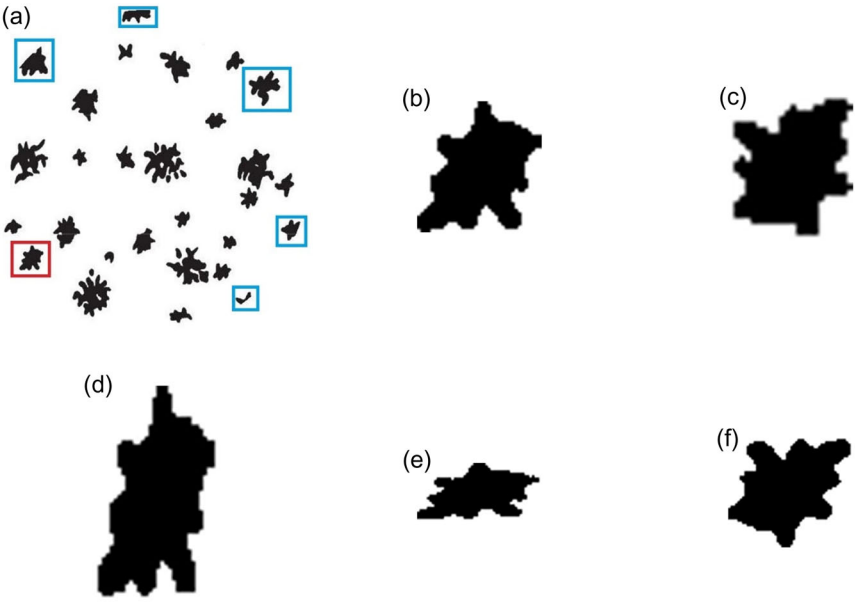

Figure 4. Schematic example showing the geometric transformation experienced by one specific spot (surrounded by a red box in a)) belonging to the reference micrograph corresponding to category IV according to the standard ISO 945-1:2019.[11] b–f) Examples of the transformed spot. The spots surrounded by a blue box are those that have been removed since they intersect the edges of the image.

Table 1. Data augmentation transformations applied to the individual spots for the generation of synthetic images. Each transformation is applied using a coefficient that takes a random value within the range specified in each case.

Transformation	Range
Image width	[0.5, 1.5]
Image height	[0.5, 1.5]
Horizontal axis inversion	True/False
Vertical axis inversion	True/False
Rotation	(0, 2π]
Shear range	[0, 0.4]

the dataset for training the CNNs. The first dataset, consisting of 2400 pure 224 × 224 synthetic images, was used to develop a pretrained CNN to classify the graphite microstructures according the categories of the standard ISO 945-1:2019.[11] The second dataset comprised 1500 128 × 128 mixed images, each of them including synthetic spots coming from two consecutive reference categories (I and II, II and III, etc.), the fraction of spots from each category in the image being randomly selected. This dataset will feed the CNN developed for the semantic segmentation-based analysis that was trained to produce a pixel-level classification, that is, to assign a label to each of the pixel in the image.

2.3.3. Description of CNNs and Datasets

This section details the development and implementation of the two CNNs designed for classifying graphite microstructures in cast iron based on the ISO 945-1:2019[11] standard. The reference images provided by ISO 945-1:2019[11] were preprocessed to extract individual spots corresponding to different categories of graphite morphology. Using a Python script with the OpenCV library, the contours of these spots were identified and segmented. Spots touching the image edges were excluded to avoid introducing noise. Two distinct datasets were created. 1) Pure-image dataset: This dataset contained 2400 synthetic images, each with spots from a single category, with each of the six classes contributing 400 images. 1800 out of these were used for training, 300 for validation, and the remaining 300 for testing. This dataset was used to train the first CNN for classifying pure images. To ensure uniform input dimensions required by CNNs, all images were resized to 224 × 224 pixels. 2) Mixed-image dataset: This dataset included 1500 synthetic images, each combining spots from two consecutive categories. It was used to train the second CNN for semantic segmentation. Image labeling was conducted by assigning a color to each pixel based on its association with one of the six categories established by the ISO 945-1:2019[11] standard. Background pixels (white) were assigned a value of 0; red pixels (first category in a combination, for instance, Class I in the I + II combination) were labeled 1, and blue pixels (second category) were labeled 2. This categorization implies that the semantic segmentation comprised five image groups (corresponding to adjacent classes I + II, II + III, III + IV, IV + V, and V + VI, respectively), including seven categories: six categories corresponded to the six distinct types of spots (refer to Figure 1), plus the background pixels (corresponding to Class 0). A total of 1500 mixed images of resolution 128 × 128 pixels were produced, with each of the five categories contributing 300 images. The images were allocated as follows: 1000 for training, 250 for validation, and the remaining 250 for testing purposes. When a new image is input into the trained model, it generates a colored mask, matching the resolution of the original image (128 × 128), that reflects the predicted assignment for each pixel (an example can be seen in Figure 2).

The two CNN models developed for this study were designed for different tasks, namely, pure image classification and semantic segmentation.

MobileNet for Pure Image Classification: This CNN was based on the pretrained deep neural network MobileNet,[26] an open-source computer vision model from Google, typically utilized for training classifiers. MobileNet uses depthwise convolutions to substantially decrease the parameter count compared to alternative networks, thereby forming a compact deep neural network with decreased computation time. The training of MobileNet involved the utilization of the COCO dataset,[27] encompassing a total of 2.5 million labeled instances across 328 k images. These images capture complex everyday scenes, consisting of 91 common object types within their natural context. All the layers of this model were frozen to prevent the weights from updating during training. Two additional trainable layers were added, namely, a max pooling layer and a fully connected layer with six neurons and softmax activation to function as a classifier. This way, only ≈6 k out of ≈3 M parameters were trainable. The number of epochs was set to 7 and the optimizer was Adam with a learning rate set to 4. All layers of MobileNet were frozen, and additional layers were added, including a max-pooling layer and a fully connected layer with six neurons and softmax activation for classification. The network was trained on 1800 images, validated on 300, and tested on 1200 images. The Adam optimizer with a learning rate of 0.0001 was used, and training was conducted over seven epochs.

U-Net for Semantic Segmentation: Pretraining was also applied in the context of semantic segmentation. U-Net,[28] a form of CNN initially devised for biomedical image segmentation at the University of Freiburg's, has demonstrated significant success in image segmentation tasks across various image types. U-Net was engineered to operate with a smaller number of training samples while delivering more accurate segmentations. The U-Net weights were pretrained using the ImageNet dataset,[29] comprising over 14 million hand-annotated images spanning over 20 000 categories, each indicating the objects depicted within. The final model utilized the pretrained U-Net CNN tailored for RGB image classification with a resolution of 128 × 128 and seven categories, including background pixels and the six spot classes. All layers employed ReLU as the activation function, with the exception of the final layer, which employed softmax. The output vector from softmax returns seven probabilities, and the class with the highest probability was designated as the classification, following a "winner-takes-all" approach. The training was conducted over 100 epochs with a batch size of eight. Adam was selected as the optimizer, operating at a learning rate of 0.0001, and a fivefold crossvalidation was enforced.

Semantic segmentation might potentially be impacted by class imbalance in this analysis. This scenario occurs when there's an unequal distribution of classes within the training dataset, potentially causing the trained model to exhibit bias toward the class with higher representation. This well-known artifact could be particularly relevant for the synthetically generated images in this research, as more than 90% of the pixels are associated with the background. In such cases, a model may reach high accuracy merely by predicting the majority class (in this instance, predicting all pixels as background). However, such an approach completely fails to capture the minority class, which is often the key purpose of developing the model. Although background pixels are irrelevant, semantic segmentation demands all the pictures in the images to be properly labeled. For illustration, **Table 2** presents the percentage distribution of pixel types within the test set (consisting of 250 images), along with the per-class accuracy and *F*-value scores for one of the initial models tested using categorical cross-entropy as the loss function. As anticipated, Class 0 pixels displayed a notably high accuracy of 99%, whereas the accuracy for Classes II–V was equal to or below 70%, dropping further to 47% and 48% for Classes III and IV, respectively. Class VI exhibited an exceptional accuracy of 87%. The *F*-value distribution followed a similar trend, with an exceptionally low value for Class V where $F = 0.16$. To address this issue, Shruti[30] recommends employing a focal loss function instead of the standard categorical cross-entropy. Focal loss integrates a modulating term into the cross-entropy loss to focus learning on instances that are difficult to classify. It essentially provides a dynamic scaling of the cross-entropy loss, wherein the scaling factor decreases to zero as the confidence in the correct class intensifies. Conceptually, this scaling factor can automatically lessen the contribution of easily classified examples during training and quickly focus the model on hard examples.[31] The significant impact of this technique on both accuracy and the *F*-score is demonstrated in Table 2. Categories II–V display a considerable increase in both accuracy and F-score, with the improvement being particularly notable for Class V, where the *F*-score rose from 0.16 to 0.86.

The implementation of these two CNN models—MobileNet for pure image classification and U-Net for semantic segmentation—demonstrates a robust approach to classifying graphite morphologies in cast iron. By leveraging data augmentation, pretraining, and advanced loss functions, the models achieved high accuracy and generalization capabilities, providing a reliable automated method for graphite classification according to ISO 945-1:2019.[11]

3. Results

In Results, we explore the findings derived from our study to classify graphite microstructures in cast alloys through CNNs. The first part, Section 3.1, presents the results obtained through a pretrained CNN for the task of classifying pure images identifying distinct graphite microstructures in a set of cast alloy synthetic images. The second part, Section 3.2, summarizes the results achieved by means of the method of semantic segmentation applied to a synthetically generated dataset of cast iron mixed images. This segmentation process is instrumental in labeling individual pixels within each image, thereby affording a deeper understanding of the microstructural composition.

3.1. Classification of Pure Images through a Pretrained CNN

Figure 5 provides a graphical representation of the progression of the accuracy and loss function across the five validation set folds. Notably, the scores for both metrics are substantial from the initial epochs, which is attributable to the employment of a pretrained CNN that is well-equipped for feature extraction and pattern recognition within the images. From the fourth epoch onward, the accuracy demonstrates a stagnation, with no substantial enhancements observed, and the rate of loss function reduction becomes significantly slower.

Figure 6 displays the comparative progress of the average accuracy between the training and test sets. During the initial epochs, the training set exhibits marginally superior accuracy as compared to the test set; however, this disparity diminishes substantially from the third epoch onward. This pattern proves the lack of overfitting within the model.

The overall accuracy of the test set, which comprises 300 pure synthetic images (50 per category) unseen by the algorithm during training, is a remarkable $98.9 \pm 0.4\%$. The first row of **Table 3** displays the accuracy achieved by the algorithms for each of the six classes within the test dataset. The k-fold approach produces five partial results, since the model has been trained five times, using a different validation fold in each iteration. The final result is derived by computing the arithmetic mean of these five results, and the uncertainty is the standard deviation of the set. This approach provides an improved assessment of the average performance and stability of the model.

The results across all classes are excellent, with the accuracy consistently exceeding 97%. Although the accuracy for classes IV and V, corresponding to irregular nodular and indistinct nodular

Table 2. The table shows the percentage of pixels belonging to each class (including background pixels) in the images belonging to the test dataset and a comparison of the accuracy and *F*-score obtained using the cross-entropy categorical loss function and the focal function, respectively.

	Class	0	I	II	III	IV	V	VI
	% Pixels	93.1%	0.3%	1.7%	0.9%	1.5%	1.6%	0.8%
Categorical Cross-entropy	Accuracy	99%	70%	66%	47%	48%	63%	87%
	F-Score	0.99	0.76	0.81	0.81	0.81	0.16	0.88
Focal Loss	Accuracy	100%	84%	93%	89%	91%	86%	90%
	F-Value	1.00	0.84	0.93	0.89	0.90	0.86	0.90

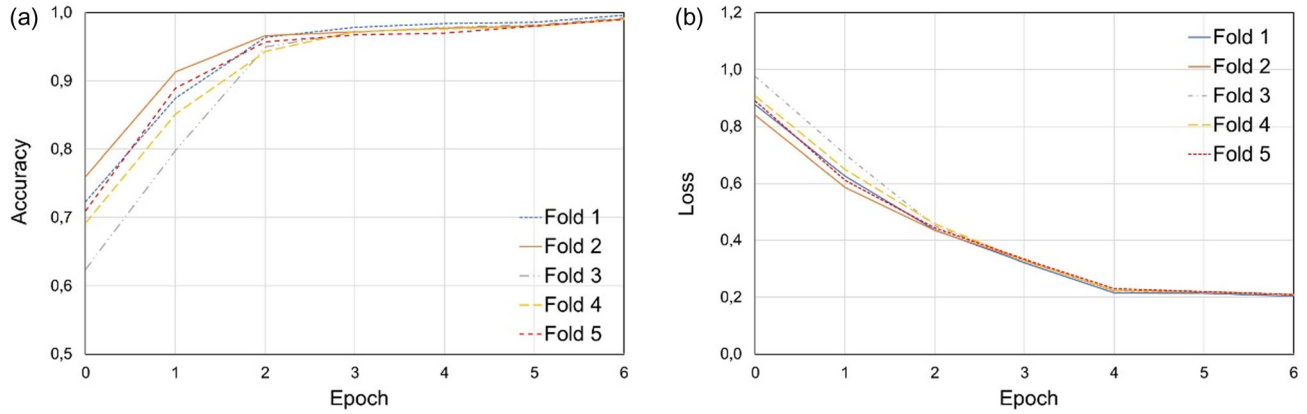

Figure 5. a) Evolution of the accuracy and b) loss function with respect to the number of epochs when evaluating the model on the validation set for each of the five folds.

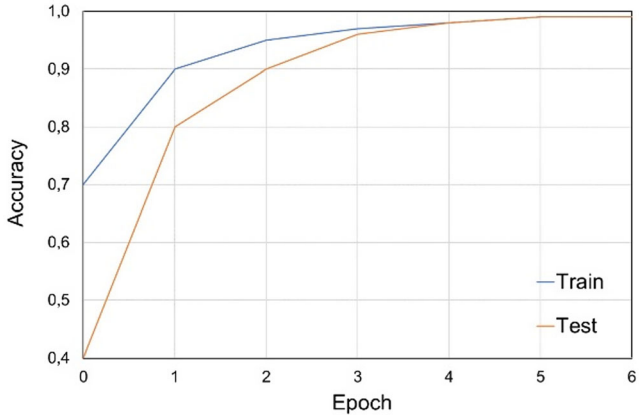

Figure 6. Evolution of mean accuracy across the five folds with respect to the number of epochs for the training and validation sets.

graphite spots respectively, is slightly less than the others, this performance is considered as more than satisfactory. The second row of Table 3 presents the average probabilities assigned by the classifier to the 50 instances of each class (remember that the last layer of the CNN consists of a fully connected layer with six neurons and softmax activation providing the probability attributed by the model to each of the six classes) This measure reflects the algorithm's confidence in classifying each category. Classes I and VI are identified with the highest confidence, whereas the algorithm is slightly less certain when classifying classes IV and V. This aligns well with the marginally lower accuracy recorded for these classes.

Table 4 complements 3 by providing a confusion matrix from the test set, which helps elucidate the sources of misclassifications. Classes IV and V demonstrate the lowest accuracy, which can be attributed to the geometric similarity between the shapes of these categories' spots, as depicted in Figure 1. Therefore, 1.2% of Class IV spots are mistakenly identified as Class V by the model, and reciprocally, 1.9% of Class V spots are misclassified as Class IV. Despite these inaccuracies, such a degree of misclassification is still considered entirely acceptable for practical applications.

3.2. Classification of Mixed Images through Semantic Segmentation

Figure 7 describes the progression of the mean accuracy (left) and mean focal loss function (right), averaged across the five folds during validation, over the span of the training epochs for both the training and validation datasets. Certain patterns of significance can be discerned from these figures. First, the close alignment in the development of these metrics for the training and validation sets underscores the robustness of the model, indicating no need for implementing additional regularization methods. Furthermore, the high accuracy values achieved from the early epochs can be attributed to the application of a pretrained deep neural network, already equipped with the ability of extracting features from the images; this is also aided by the large presence of background pixels in the dataset which the CNN quickly recognizes. The accuracy exhibits an upward trend, and correspondingly, the focal loss function diminishes until around epoch 15; following this, it seemingly plateaus and maintains a stable state. Yet, despite the graphic's limited

Table 3. Accuracy and average probability obtained using the final model to predict the type for 50 images of each category. We show the results obtained for five folds and the error (standard deviation). The second row indicates the average probability with which the algorithm predicts each class. Each average accuracy and probability is accompanied by the standard deviation across the five folds.

Class	I	II	III	IV	V	VI
Accuracy	100 ± 0	99.0 ± 0.4	99.4 ± 0.4	97.9 ± 0.4	97.0 ± 1.0	99.7 ± 0.2
Average Probability	92.8 ± 0.7	87.6 ± 0.9	83.6 ± 0.7	76.0 ± 2.0	79.0 ± 2.0	95.0 ± 0.6

Table 4. Confusion matrix of the six-category classification of pure images. Each cell within the matrix represents the percentage of images from a given category (rows) that have been classified under another category (columns). The data summarized in this matrix are derived from the testing set, which includes 50 synthetic pure images from each category. The errors in this data are indicated by standard deviations. Gray colour highlights the percentages different from zero. Blue colour indicates the highest value per column.

True values	Predicted values						
	Class	I	II	III	IV	V	VI
	I	100 ± 0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	II	0.0 ± 0.0	99.0 ± 0.4	0.7 ± 0.2	0.3 ± 0.4	0.0 ± 0.0	0.0 ± 0.0
	III	0.0 ± 0.0	0.1 ± 0.3	99.4 ± 0.4	0.5 ± 0.4	0.0 ± 0.0	0.0 ± 0.0
	IV	0.0 ± 0.0	0.6 ± 0.4	0.3 ± 0.4	97.9 ± 0.4	1.2 ± 0.2	0.0 ± 0.0
	V	0.0 ± 0.0	0.0 ± 0.0	0.1 ± 0.2	1.9 ± 0.9	97.0 ± 1.0	0.7 ± 0.2
	VI	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.3 ± 0.2	99.7 ± 0.2

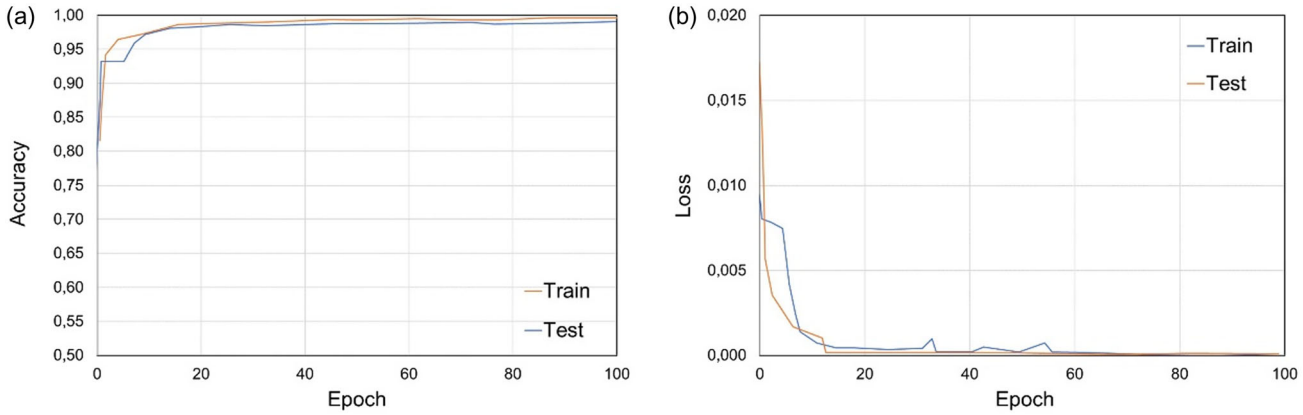

Figure 7. a) Progression of the accuracy and b) the focal loss function relative to the number of training epochs for the training and validation datasets. In each case, the depicted values represent the average across the five folds for each epoch.

resolution, the accuracy continues to marginally ascend up until epoch 100, paired with a minor decrease in the focal loss function. While the late-stage accuracy increase might initially seem irrelevant, this additional training phase is instrumental in enabling the model to identify finer details and thus accurately distinguish between spots of varying categories.

Table 5 presents the confusion matrix derived from the 250 mixed images in the test set (50 images in each category).

Table 5. Confusion matrix of the seven-category classification of mixed images (background pixels are included). Each cell within the matrix represents the percentage of images from a given category (rows) that have been classified under another category (columns). The data summarized in this matrix are derived from the testing set, which includes 50 synthetic mixed images from each category. The errors in this data are indicated by standard deviations. Gray colour highlights the percentages different from zero. Blue colour indicates the highest value per column.

True values	Predicted values							
	Class	0	I	II	III	IV	V	VI
	0	99.8 ± 0.4	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	I	4.2 ± 0.8	84.0 ± 2.0	11.0 ± 2.0	0.5 ± 0.4	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	II	3.4 ± 0.8	0.8 ± 0.4	93.0 ± 2.0	2.3 ± 0.5	0.1 ± 0.1	0.0 ± 0.0	0.0 ± 0.0
	III	4.0 ± 1.0	0.0 ± 0.0	4.0 ± 1.0	89.0 ± 2.0	2.6 ± 0.5	0.0 ± 0.0	0.0 ± 0.0
	IV	2.6 ± 0.6	0.0 ± 0.0	0.1 ± 0.1	1.0 ± 0.2	91.0 ± 2.0	6.0 ± 2.0	0.0 ± 0.0
	V	3.0 ± 0.7	0.0 ± 0.0	0.0 ± 0.5	0.0 ± 0.0	9.0 ± 2.0	86.0 ± 2.0	1.7 ± 0.2
	VI	2.0 ± 1.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.5 ± 0.3	7.0 ± 5.0	90.0 ± 2.0

Table 6. Accuracy obtained for each set of consecutive categories and a predefined percentage of spots for each category. The data summarized in this matrix are derived from the testing set, which includes 250 synthetic mixed images, 50 of them from each category. The errors in this data are indicated by standard deviations.

	20–80%	40–60%	60–40%	80–20%
I–II	99.24 ± 0.05	99.29 ± 0.05	99.30 ± 0.03	99.34 ± 0.06
II–III	99.16 ± 0.08	99.10 ± 0.05	99.12 ± 0.05	99.14 ± 0.04
III–IV	99.1 ± 0.2	99.37 ± 0.03	99.35 ± 0.05	99.3 ± 0.1
IV–V	98.7 ± 0.3	99.0 ± 0.1	98.7 ± 0.1	98.4 ± 0.2
V–VI	99 ± 1	98.9 ± 0.7	99.0 ± 0.2	98.6 ± 0.1

The final accuracy, including all pixels, reaches a significant $99.2 \pm 0.1\%$. When the background pixels are excluded, the accuracy still stands at $91.0\% \pm 1.0\%$. Excluding background pixels, see Table 5, the individual class accuracy fluctuates between $84.0 \pm 2.0\%$ and $93.0 \pm 2.0\%$, demonstrating remarkable results. The classes I and V report the least favorable results. Around 4% of the class I pixels are misclassified as background pixels and ≈11% as class II pixels. For class V, the primary sources of misclassifications are the background (3%), class IV (9%), and class VI (2%), respectively. It's also worth noting that for all categories, a minor percentage of pixels are erroneously classified as class 0 (background). This likely originates from a boundary effect, where the algorithm struggles to accurately identify pixels on the edge that separates the spot from the background. However, given the overall accuracy of the CNN in accurately classifying the majority of pixels, this error is relatively insignificant in practical terms.

Finally, images with spots belonging to consecutive categories (I and II, II and III, III and IV, IV and V, V and VI) have been generated, with the percentages of spots defined as follows: 20–80%, 40–60%, 60–40%, 80–20%. These percentages apply to the number of spots of each type, not to the number of pixels in the image. This analysis has been conducted to understand how the prediction capacity of the model varies depending on the percentage of spots for certain categories. As shown in **Table 6**, all categories yield good results in general, with the IV–V and V–VI categories slightly below the average.

4. Summary and Discussion

The classification of cast iron is determined by the shapes of its graphite particles since they have a direct influence on the final properties of the material. Specifically, the graphite particles' shape significantly impacts properties like fracture toughness and ductility. The shape of graphite is identified by comparing it to the reference images given in ISO 945-1:2019.[11] This standard classifies cast irons into six categories, labeled with Roman numerals from I to VI. Although this assessment must be conducted by an expert in this specialized metallographic technique, the method's intrinsic subjectivity presents substantial challenges to the repeatability and accuracy of the process. Hence, there is a significant need for a reliable, automated method for classifying the various graphite shapes. This study is focused on developing and validating the use of CNNs for the morphological categorization of graphite spots in cast irons, adhering to the guidelines of the ISO 945-1:2019[11] standard.

The first limitation of this study arises from the restricted amount of information available for training, validating, and testing the CNNs. The standard ISO 945-1:2019[11] only includes six reference images, each one of them corresponding to a distinct category in the classification and including a variable number of spots. Consequently, the initial challenge was to generate a sufficient number of reliable synthetic samples. The first step in this process involved isolating individual spots from each reference image of the standard. To prevent dataset redundancies and minimize the risk of overfitting, these isolated spots were subjected to data augmentation via various geometric manipulations, avoiding any unnatural distortions that could modify the physical geometric characteristics of the spots. This approach yielded a broad array of individual synthetic spots, which were later randomly combined to create synthetic reference images to be fed into the CNNs. By means of this methodology, two distinct datasets were generated. The first one contained 2400 "pure images" (224×224 pixels), meaning images composed each of them of synthetic spots extracted from the same category according to ISO 945-1:2019.[11] The second dataset consisted of 1500 "mixed images" (128×128 pixels) where each of them incorporates synthetic spots derived from two adjacent categories (I and II, II and III, III and IV, V and V, and V and VI, respectively).

Each of these datasets was subsequently utilized to feed a pretrained CNN. Specifically, the pure images dataset was provided to the pretrained deep neural network MobileNet. All layers of this network were frozen, and a max pooling layer along with a fully connected layer consisting of six neurons and softmax activation for six categories was added. This approach significantly reduced the number of trainable parameters. The train, validation, and test sets comprised 1800, 300, and 300 samples, respectively. The mixed images dataset was supplied to the pretrained U-Net CNN for a semantic segmentation analysis, with pixels falling into seven categories (the six types of spots as per ISO 945-1:2019,[11] along with the white background pixels in the images). Therefore, the final layer utilizes the softmax activation function for seven classes. The train, validation, and test sets in this case consisted of 1000, 250, and 250 samples, respectively.

The results derived using the deep learning models demonstrate the efficacy of the proposed methodology. In particular, the CNN trained with pure images yields an overall test set accuracy of $98.9 \pm 0.4\%$, surpassing a 97% benchmark across all six classes. Classes IV and V, representing two forms of nodular cast irons (irregular and indistinct, respectively), achieve slightly lower accuracy values. It is worth noting that, in practical scenarios, even experienced operators can encounter challenges distinguishing between these two categories. Moreover, the CNN assigns an average probability, which is indicative of its confidence level, exceeding 76% for all classes and surpassing 90% for classes I and VI. Similarly, the results obtained from the semantic segmentation analysis are equally creditable. The task here involved designing a deep learning classifier capable of isolating spots of adjacent classes within the same image—a more challenging goal. The potential for class imbalance was significant in this problem given that over 90% of the pixels belong to the background. In order to counteract this potential issue,

a focal loss function was employed instead of the conventional categorical cross-entropy loss function. Yet, the achieved accuracy for the different classes (excluding the background pixels) spans from 84% to 93%. In conclusion, the deep learning strategy implemented in this study presents a highly reliable technique for classifying cast irons based on graphite morphology.

In the domain of materials engineering, the metallographic classification of cast irons represents a significant area of research. A notable and very recent contribution to this field was made by Sarrionandia et al.,[25] who explored a topic akin to our specific investigation into the classification of pure images. Their study employed a combination of CNNs, including two pretrained models (VGG16 and VGG19) and a custom model designed for alphanumeric classification, to categorize metallographic images of graphite cast irons. This classification was conducted in accordance with ISO 945-1:2019 standards. To address class imbalance, data augmentation techniques were applied, resulting in an achieved overall classification accuracy of 95%. However, their analysis revealed a higher classification error for class V. Furthermore, the authors developed algorithms to elucidate the features recognized by their DL classifier, enhancing understanding of the CNNs' operational mechanisms. Our research posits that the observed improvement in classification accuracy, as detailed in our study, can be substantially attributed to the application of transfer learning. As discussed in Section 2.2.4 of our article, transfer learning offers several benefits over traditional CNN approaches, including accelerated convergence, reduced data requirements, enhanced model performance, improved generalization capabilities, and decreased computational demand. We conjecture that these advantages have contributed to the superior accuracy rates reported in our findings.

Microstructure modeling with commercial software packages has gained significant prominence in recent years. Conventionally, the automatic classification of graphite's morphology in cast iron has been carried out through traditional statistical methods or conventional ML. This involves a considerable amount of time spent on manual feature selection and engineering, in which the developer relies on domain-specific knowledge to create features that enhance the performance of ML algorithms. Subsequently, these manually curated features are introduced to a classifier, a method followed, for example, by Gomes and Paciornik,[3] among others. These authors correctly identified the challenge in cast iron classification as selecting a set of parameters capable of grouping particles within the same class, while accounting for inherent variability and ensuring optimal discrimination among the six classes defined by the ISO 945-1:2019 standard.[11] However, in DL the need for manual feature extraction from images is eliminated. Deep neural networks autonomously extract features and determine their impact on the output by assigning weights to their connections. In this process, raw images are input into the network and, as they progress through the network layers, the network identifies patterns within the images, from which features are created. As such, deep neural networks can be seen as feature extractors in conjunction with classifiers that are trainable end to end, in contrast with traditional ML models that rely on manually curated features. CNNs demonstrate exceptional proficiency in image classification due to their inherent ability to automatically discern spatial feature hierarchies, such as edges, textures, and shapes, all necessary for object recognition within images. The accumulation of evidence over the past 20 years validates the superior performance of CNNs in contrast to traditional ML for the task of image classification. Specifically, in the context of complex, real-world image data, such as the graphite morphology in cast irons, CNNs significantly outperform alternative methodologies.

5. Conclusion

The following list of conclusions encapsulates the main achievements of this article, providing a concise summary of its contributions to the field of materials science and engineering and the broader domain of AI and ML applications: 1) Innovative application of CNNs: The study introduces a pioneering approach that applies CNNs—MobileNet for image classification and U-Net for semantic segmentation—to automate the classification of graphite cast iron alloys. This marks a significant shift from traditional manual methods to a more objective, reliable, and efficient automated system; 2) High classification accuracy: The research achieves remarkable classification precision, providing an overall accuracy of $98.9 \pm 0.4\%$ for pure image classification across all six classes and accuracy ranging between 84% and 93% for semantic segmentation of mixed images. This demonstrates the model's ability to consistently identify and categorize graphite morphology with precision. 3) Creation of a comprehensive synthetic dataset: To overcome the challenge of limited real-world datasets, a significant contribution of this study is the generation of a synthetic dataset, including 2400 pure and 1500 mixed images based on the ISO 945-1:2019 standard. This synthetic dataset ensures a robust training process, enhancing the model's generalization capability across various graphite morphologies. 4) Effective addressing of class imbalance: The research innovatively addresses the potential issue of class imbalance—a common challenge in ML and particularly relevant due to the high percentage of background pixels in images—using a focal loss function during semantic segmentation. This significantly improves model performance on minority classes. 5) Contributions to materials science and engineering: By automating the classification of cast iron alloys based on graphite morphology, the study directly contributes to materials science and engineering. It aids in the selection process for various industrial applications by providing a more accurate, speedy, and objective method to classify cast iron alloys, which is crucial due to its impact on mechanical properties. 6) Demonstration of transfer learning benefits: Utilizing pretrained models (MobileNet and U-Net) through transfer learning, the study highlights the efficiency and effectiveness of leveraging existing neural networks for new classification tasks, thereby reducing computational resources and time required for training from scratch.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available at DIGITAL.CSIC.[32]

Keywords

cast iron, convolutional neural networks, deep learning, image classifications, semantic segmentations

Received: February 7, 2024
Revised: July 16, 2024
Published online: August 28, 2024

[1] W. D. Callister, *Materials Science and Engineering*, John Wiley & Sons Ltd., Hoboken, NJ **2010**.
[2] D. M. Stefanescu, ASM International Handbook Committee, *ASM Handbook. Volume 1A, Cast Iron Science and Technology*, ASM International, Materials Park, OH **2017**, https://worldcat.org/title/1052568383.
[3] O. d. F. M. Gomes, S. Paciornik, *Microsc. Microanal.* **2005**, *11*, 363.
[4] M. F. Ashby, D. R. H. Jones, *Engineering Materials 1. An Introduction to Properties, Applications, and Design*, Elsevier Ltd., Oxford UK **2012**.
[5] D. A. Porter, K. E. Easterling, M. Y. Sherif, *Phase Transformations in Metals and Alloys*, 4th ed., CRC Press, Boca Raton, FL **2021**.
[6] H. K. D. H. Bhadeshia, R. Honeycombe, *Steels : Microstructure and properties*, Elsevier, Butterworth-Heinemann, Amsterdam **2006**.
[7] H. T. Angus, *Cast Iron: Physical and Engineering Properties*, 2nd ed., Butterworths, London, UK **1978**.
[8] C. R. Loper, P. C. Rosenthal, R. W. Heine, *Principles of Metal Casting*, TMH, New Delhi, India **2013**.
[9] W. Smith, J. Hashemi, *Foundations of Materials Science and Engineering*, McGraw Hill, New Delhi, India **2022**.
[10] M. König, *Int. J. Cast Met. Res.* **2010**, *23*, 185.
[11] International Organization for Standardization, *Microstructure of cast irons - Part 1: Graphite classification by visual analysis* (ISO Standard No 945-1:2019) **2019**, p. 32
[12] M. Warmuzek, M. Żelawski, T. Jałocha, *Comput. Mater. Sci.* **2021**, *199*, 110722.
[13] J. Komenda, *Mater. Charact.* **2001**, *46*, 87.
[14] I. J. Turias, J. M. Gutierrez, P. L. Galindo, *Sci. Eng. Compos. Mater.* **2002**, *10*, 91.
[15] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, *Sci. Rep.* **2018**, *8*, 1.
[16] B. L. DeCost, B. Lei, T. Francis, E. A. Holm, *Microsc. Microanal.* **2019**, *25*, 21.
[17] D. M. Dimiduk, E. A. Holm, S. R. Niezgoda, *Integr. Mater. Manuf. Innov.* **2018**, *7*, 157.
[18] R. Elbana, R. Mostafa, A. Elkeran, *Int. J. Mech. Mechatron. Eng.* **2020**, *20*, 18.
[19] F. Iacoviello, D. Iacoviello, V. Di Cocco, A. De Santis, L. D'Agostino, *Procedia Struct. Integrity* **2017**, *3*, 283.
[20] *ASTM Book of Standards Volume: 01.02* **2016**, p. 13, https://doi.org/10.1520/A0247-16A.
[21] F. Chollet, *Deep Learning with Python*, Manning Publications, Shelter Island, NY **2017**.
[22] M. Elgendy, *Deep Learning for Vision Systems*, Manning Publications **2020**.
[23] L. Che, Z. He, K. Zheng, T. Si, M. Ge, H. Cheng, L. Zeng, *Mater. Today Commun.* **2023**, *37*, 107531.
[24] M. Szatkowski, D. Wilk-Kołodziejczyk, K. Jaśkowiec, M. Małysza, A. Bitka, M. Głowacki, *Materials* **2023**, *16*, 6837.
[25] X. Sarrionandia, J. Nieves, B. Bravo, I. Pastor-López, P. G. Bringas, *J. Manuf. Mater. Process.* **2023**, *7*, 17.
[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam (Preprint), arXiv:1704.04861, v1, Submitted: Apr. **2017**.
[27] COCO - Common Objects in Context, https://cocodataset.org/#home (accessed: July 2023).
[28] O. Ronneberger, P. Fischer, T. Brox (Preprint), arXiv:1505.04597, v1, Submitted: May **2015**, https://doi.org/10.48550/arXiv.1505.04597.
[29] ImageNet, https://www.image-net.org/ (accessed: July 2023).
[30] S. Jadon (Preprint), arXiv:2006.14822, v1, Submitted: Jun. **2020**, https://doi.org/10.48550/arXiv.2006.14822.
[31] T. Y. Lin (Preprint), arXiv:1708.02002, v1, Submitted: Aug. **2017**, https://doi.org/10.48550/arXiv.1708.02002.
[32] M. Bárcena Rodríguez, L. Lloret Iglesias, D. Ferreño, I. Carrascal Vaquero, *Optical microscopy images of cast iron alloys defects [Dataset]*, DIGITAL.CSIC **2024**, https://doi.org/10.20350/digitalCSIC/16484