





Article

Computer Aided Classifier of Colorectal Cancer on Histopathological Whole Slide Images Analyzing Deep Learning Architecture Parameters

Elena Martínez-Fernandez ¹, Ignacio Rojas-Valenzuela ¹, Olga Valenzuela ^{2,*} and Ignacio Rojas ¹¹ School of Technology and Telecommunications Engineering, University of Granada, 18071 Granada, Spain² Department of Applied Mathematics, University of Granada, 18071 Granada, Spain

* Correspondence: olgavc@ugr.es

Abstract: The diagnosis of different pathologies and stages of cancer using whole histopathology slide images (WSI) is the gold standard for determining the degree of tissue metastasis. The use of deep learning systems in the field of medical images, especially histopathology images, is becoming increasingly important. The training and optimization of deep neural network models involve fine-tuning parameters and hyperparameters such as learning rate, batch size (BS), and boost to improve the performance of the model in task-specific applications. Tuning hyperparameters is a major challenge in designing deep neural network models, having a large impact on the performance. This paper analyzes how the parameters and hyperparameters of a deep learning architecture affect the classification of colorectal cancer (CRC) histopathology images using the well-known VGG19 model. This paper also discusses the pre-processing of these images, such as the use of color normalization and stretching transformations on the data set. Among these hyperparameters, the most important neural network hyperparameter is the learning rate (LR). In this paper, different strategies for the optimization of LR are analyzed (both static and dynamic) and a new experiment based on the variation of LR is proposed (the relevance of dynamic strategies over fixed LR is highlighted), after each layer of the neural network together with decreasing variations according to the epochs. The results obtained are very remarkable, obtaining in the simulation an accurate system that achieves 96.4% accuracy on test images (for nine different tissue classes) using the triangular-cyclic learning rate.

Keywords: deep learning; convolutional neural network; WSI; cancer; hyperparameters; histopathology images; discriminative fine tuning



Citation: Martínez-Fernandez, E.; Rojas-Valenzuela, I.; Valenzuela, O.; Rojas, I. Computer Aided Classifier of Colorectal Cancer on Histopathological Whole Slide Images Analyzing Deep Learning Architecture Parameters. *Appl. Sci.* **2023**, *13*, 4594. <https://doi.org/10.3390/app13074594>

Academic Editors: Jan Egger and Syoji Kobashi

Received: 17 February 2023

Revised: 17 March 2023

Accepted: 29 March 2023

Published: 5 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Colorectal carcinoma is one of the most common cancers, and its heterogeneous composition changes greatly as the cancer grows [1]. For this reason, it is of utmost importance to know the different tissues that coexist with tumor cells during pathological colonoscopy examination. This study was developed by applying multiresolution techniques based on deep learning on WSI (Whole-Slide Images) on CRC tissue. Histological images (WSI) are images of the microscopic structure of tissues. The pathologist usually uses a microscope to view the stained sample on a slide. To better visualize the parts of the tissue that are of interest to us, a technique known as hematoxylin–eosin staining is performed. After the tissue is digitized, a WSI is created.

In recent years, machine learning algorithms for image analysis have evolved rapidly with computing power and new image processing techniques. Thanks to the advancement of histological slide scanners, the development and use of digital histopathology in cancer diagnosis is becoming very relevant [2]. An overview of treatment options in classifying histopathological images with machine vision learning models was recently provided by Li et al. [3]. In recent years, the applications of deep learning in the diagnosis and

treatment of histopathological classification mainly include studies based on classification and recognition research. The use of deep learning systems in colorectal cancer has led to a significant increase in the number of scientific publications. Figure 1 shows the evolution of the number of papers indexed on the Web of Science platform (previously known as Web of Knowledge), a platform that provides access to several databases of scientific journals and conference proceedings in the field of colorectal cancer and deep learning models. Figure 2 shows an analysis of publications (2014–2022) by research area (a paper may be attributed to more than one research area). However, as can be seen from [3], there are far fewer studies that address the impact of changing hyperparameters in Deep Learning, although adjusting them can drastically change the results obtained.

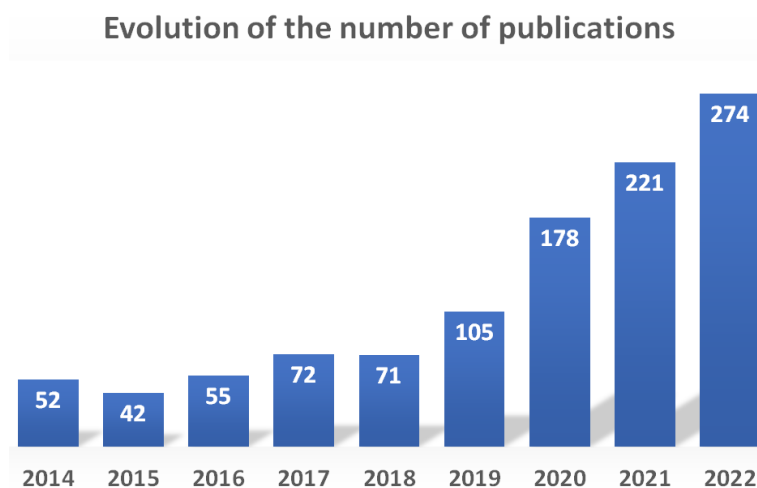


Figure 1. Evolution of the number of contributions (indexed in the ISI Web of Science for year 2014 to 2022) in the field of deep learning techniques applied to colorectal cancer.

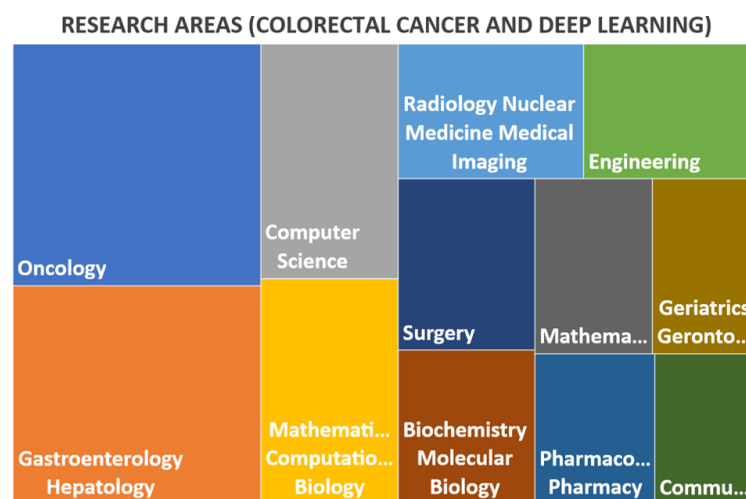


Figure 2. Analysis of publications by research area, in the field of deep learning techniques applied to colorectal cancer.

The learning rate is one of the most important hyperparameters in deep learning training. It controls how much the model changes in response to the estimated error each time the model weights are updated [4]. Choosing the learning rate is difficult because too small a value can lead to a long training process that can become stuck, while too high a value can lead to a suboptimal set of weights that are learned too quickly. There are numerous contributions to the analysis and automation/optimization of the adaptive learning rate and hyperparameter optimization, but this remains an open research problem that depends heavily on the nature of the data and the problem to be solved [4–6].

In most studies conducted in the field of classification systems for histopathological images of colorectal cancer [7,8], the use of deep-learning systems usually involves the use of standard parameters without a detailed analysis of the influence of the parameters, hyperparameters, and preprocessing stages of histopathological images on the behavior of the system. In this paper, we present a novel and comprehensive study on the influence on the classification performance of a classification system (Deep Learning-VGG19) for classifying histopathological images of colorectal cancer and show how an appropriate choice of these parameters can have an important impact on the accuracy of the classifier.

In this paper, we compare the use of different methods to train a neural network by varying the learning rate to perform classification of histopathological CRC images with different experiments. Recently, more and more deep learning methods have been proposed for WSI analysis. However, the study on the influence of LR variation is not so well-known. Some studies compared the influence of not using a fixed LR in each epoch. Anil et al. [9] proposed the use of a dynamic learning rate. Smith et al. [10] proposed cyclic learning rates, a method that lets the learning rate vary cyclically between the appropriate thresholds. Purnendu et al. [7] introduced another technique inspired by cyclic learning rates and stochastic gradient descent with warm restarts. In this paper, we compare these models and propose a new experiment based on the variation of LR after each layer of the neural network, along with decreasing variations according to epochs.

2. Related Work

Several studies have investigated methods for CRC detection, classification, and tissue segmentation by analysis of WSI. Kather et al. [11] presented a new dataset of 5000 histological images of human colorectal cancer that included eight different tissue types. Ten anonymized H&E-stained CRC tissue slides were obtained from the pathology archive at the University Medical Center Mannheim (Heidelberg University, Mannheim, Germany). Contiguous tissue areas were manually labeled and tessellated, resulting in 625 non-overlapping tissue tiles of size 150×150 pixels. The following eight tissue types were selected for analysis: tumor epithelium, simple stroma, complex stroma, immune cells, debris, normal mucosal glands, adipose tissue, and background (the resulting $625 \times 8 = 5000$ images together formed the training and testing set for the classification problem). The authors used four classification strategies (1-nearest neighbor, linear SVM, radial basis function SVM and decision trees) and found that the radial basis function (rbf) support vector machine (SVM) performed the best (87.4% accuracy for multiclass tissue separation).

Ciampi et al. [12] proposed a CRC tissue classification system based on convolutional nets (ConvNets). They used data from two different sources, namely a cohort of images of whole slides of rectal cancer samples (a set of 74 histological slides from 74 patients), and a dataset of colorectal cancer images and patches (from 10 patients, using 5000 patches of 150×150 pixel, the dataset of [11]). They investigated the importance of staining normalization (applying staining normalization to training and test data removes most sources of variability due to staining from the equation) in classifying CRC tissues in H&E stained images and achieved 79.7% accuracy.

Bianconi et al. [13] used several data sets for experimental analysis of a novel method called IOCLBP, which is based on a simple-to-implement yet highly discriminative local descriptor for color images. The authors have demonstrated the superiority of IOCLBP alone and/or in combination with LCC over related methods (LBP variants) for the classification of binary and multiclass problems. One such problem is the so-called Epistroma: histological images of colorectal cancer from 643 patients admitted to Helsinki University Central Hospital, Finland, between 1989 and 1998. The tissue samples were stained with diaminobenzidine and hematoxylin and divided into two classes: Epithelium (825 samples) and Stroma (551 samples). The size of the images varied from 172×172 pixels to 2372×2372 pixels. In this binary classification problem of colorectal cancer, the accuracy achieved was 93.4%.

Alinsaif et al. [14] used different deep learning models (SqueezeNet, MobileNet, ResNet, and DenseNet) to present two different approaches: (1) generating features from pre-trained models (i.e., without fine-tuning); and (2) fine-tuning the CNN from pre-trained models. The second approach was effective and provided better classification results. When training a SVM on deep features, the authors applied ILFS to obtain a reduced subspace of features while achieving high accuracy. The authors used different problems or datasets to analyze the results, including the data from Kather et al. [11] and that from Epistroma (previously discussed in Bianconi et al. [13]). Based on SVM classification using the best-scoring deep features from different pre-trained models, the best results obtained for the Kather problem, with 1000 features, were an accuracy of 95.4% and an AUC of 0.906. For the Epistroma problem (simpler since it is biclass), an accuracy of 99.06% and an AUC of 0.997 were obtained with 250 features, using DenseNet in both datasets.

However, neither study worked with a data set of nine different CRC tissue classes (presented in [15] and used in this paper). Figure 3 shows a typical classification model pipeline, presenting common techniques such as data acquisition/split, image preprocessing, whole slide image (WSI) tiling, and evaluation with a test set for the multiclass problem presented in this paper.

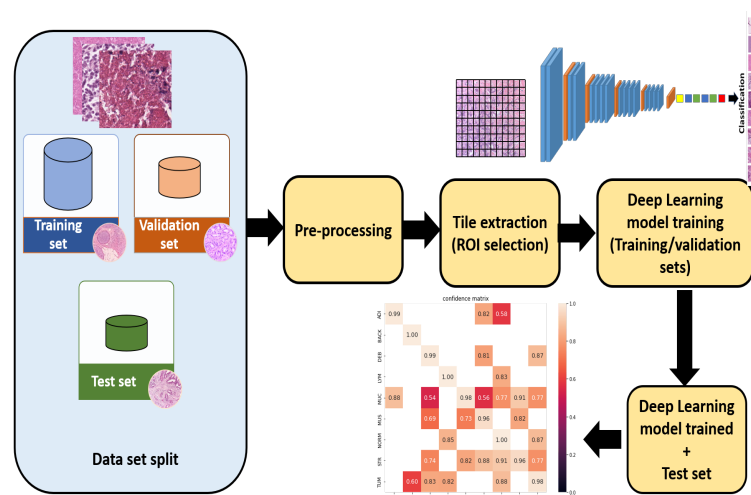


Figure 3. Block diagram for pathology WSI classification with VGG19 deep learning model, in a multi-class problem.

3. Materials and Dataset

The proposed model with hyperparameter investigation was developed and trained using a publicly available dataset of hematoxylin and eosin (HE)-stained histological images of digital WSI of human colorectal cancer (CRC) and normal tissue. These images were manually extracted from 86 H&E-stained human cancer tissue slides. It is a set of 100,000 training and 7180 test images that do not overlap and are 224×224 pixels (px) at 0.5 micrometer per pixel (MPP). There are nine different tissue classes: Adipose tissue (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal intestinal mucosa (NORM), cancer-associated stroma (STR), and colorectal adenocarcinoma epithelium (TUM). The tissue samples included CRC slides from primary tumors and tumor tissue from CRC liver metastases; the normal tissue classes were supplemented with non-tumorous regions from gastrectomy specimens to increase variability. The dataset is available and was constructed by J.K Kather et al. [15]. For training the deep learning model, code was generated to balance all classes (mainly based on data augmentation of the class with the lower number of patches, more detail in Section 5, Experiments and Results), because as shown in Figure 4, the number of patches is different for each tissue.

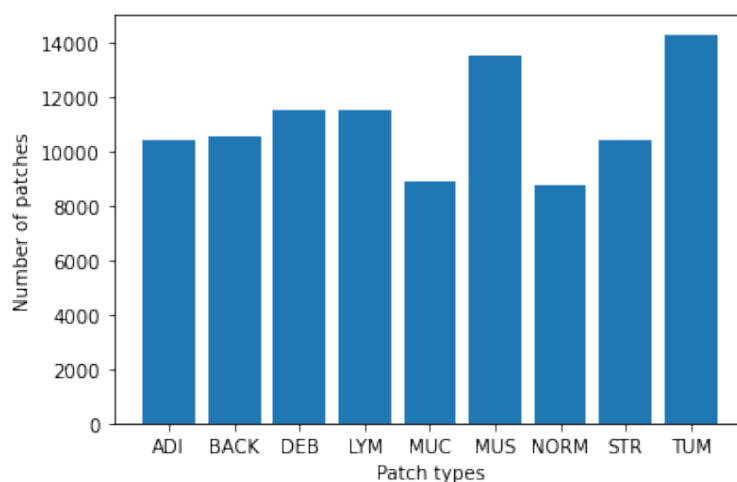


Figure 4. Number of patches in each tissue class.

Automatic recognition of different tissue types in histologic images is an essential component of digital pathology diagnosis. Although histological images often contain multiple tissue types, few studies have addressed the problem of different classes. As shown in our study, homogenization of the different tissues is crucial in classifying them into multiple classes. In this dataset, the images with and without color normalization are used. Figure 5 shows an example of the dataset used.

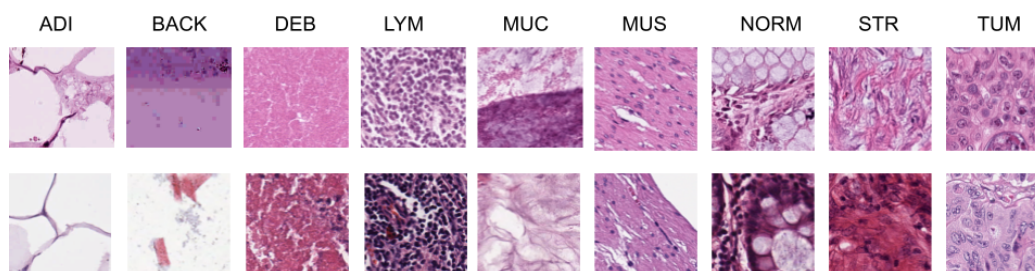


Figure 5. Example of 9 tissue classes in our dataset. The top row shows the images with Macenko filter. In the bottom row, no color normalization was applied to these images.

Neural network preprocessing, training and deployment was performed in Python (version 3.9.0) on one workstation with CPU 11th Gen Intel(R) i7-11700 K, 3.6 GHz, with 128 GB RAM and Nvidia RTX30-60 GPU.

4. Method

In this paper, a new methodology is proposed to determine and analyze the behavior of deep learning systems by optimizing the hyperparameters that determine it, focusing on the problem of classifying histopathological images (9 different tissues, including colon cancer). In this section, we first present our baseline network, an improved VGG19. Then, we describe our proposed framework, which consists of the training scheme and the study of the parameters and hyperparameters of our model. In addition, this section presents the metrics used in the analysis of the classification results for each method.

4.1. VGG19: Parameters and Hyperparameters

The use of VGG19 has been very successfully applied in the literature for the classification of histopathological images [16,17] and also specifically for the analysis of colorectal cancer with digital pathology images [18].

VGG19 is a deep convolutional neural network (CNN), a supervised learning artificial neural network that processes its layers by mimicking the visual cortex of the human eye to recognize various features of the inputs. The CNN contains specialized hidden layers

with a hierarchy in which the first layers can recognize lines and curves and specialize in deeper layers that recognize structures as complex as body tissue. The VGG19 model consists of 19 layers [19]: Sixteen convolutional layers to detect features in an image, three fully connected layers to process data in a neural network, five MaxPool layers to correct distorted images, and one SoftMax layer. We need to add several final layers for our dataset, such as flattening to convert the image to an appropriate representation, and a dropout layer to prevent overfitting. In addition, we add regularization techniques to avoid overfitting and feature selection (Figure 6). In our model, we use ridge regression (L2), which adds the squared magnitude of the coefficient as a penalty value to the loss function. We also add the activation function ReLu to smooth the picture and make the boundaries clear. This function is very fast in terms of training, so it is common to use the activation function for the hidden layers of the model and use the Softmax function on the last layer, since it is more complex and therefore slower.

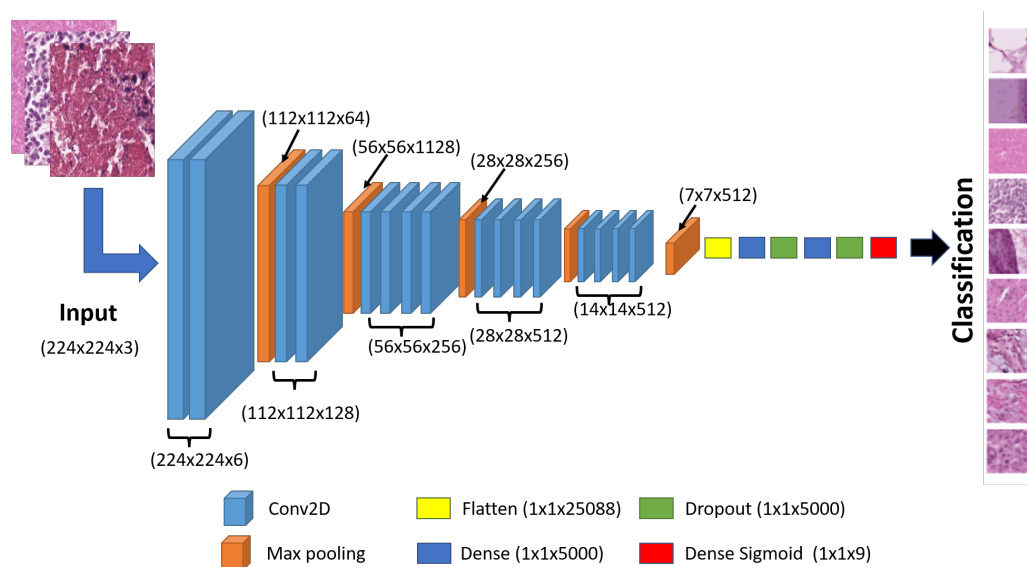


Figure 6. Architecture of the VGG19. Figure showing the convolutional layers in blue, the Maxpool layers in orange, and the final 6 designed layers added to the typical VGG19 neural network. The flatten layers are shown in yellow, the dense layers in dark blue, the green color represents dropout and finally the red dense sigmoid layer to classify the 9 different tissues.

In CNNs, it is important to distinguish between parameters and hyperparameters. Parameters refer to the weights of the neural network, which are automatically adjusted by the training algorithm. Hyperparameters, on the other hand, refer to the configuration of the architecture to be used. Moreover, they are not updated by the model according to the optimization strategy, so manual configuration is always required. Some of them are, for example: the number of layers, the number of neurons per layer, the batch size, the momentum, and the weight decay, to name a few. Optimizing the hyperparameters in deep neural networks is a crucial task for the final performance of a network. Some of the hyperparameters that we have considered in our model are as follows:

- The batch size defines the number of samples used in an epoch to train the model.
- An optimizer is a function that modifies neural network attributes, such as weights and learning rate, to reduce the overall loss and improve accuracy. A typical optimization method is gradient descent, where three types of gradient descent can be distinguished in terms of batch size: In batch gradient descent, all samples of the training set are used in each epoch. In stochastic gradient descent (SGD), a random sample from the training set is selected in each epoch. Finally, in mini-batch gradient descent, a specified number of samples from the training set are given in an epoch. In our training, we will use a (SGD) [20] with momentum that descends directly by optimizing the expected risk, since the samples are drawn randomly from the ground truth distribution.

- The number of steps or training iterations is the number of forward or backward steps, where each step uses a number of stack-size images.
- The number of epochs determines how often the network trains the entire dataset. This parameter should be adjusted according to your learning curve. In our project with Transfer Learning, we will see how our model learns in just a few epochs.
- The weights that connect the layers are the parameter of a neural network that transforms the input data into the hidden layers of the network. The concept of weights is of paramount importance because they are the variables to be found when training the network, and they gradually adjust the network to try to obtain the correct output for all inputs.

In the case of CNNs, an important hyperparameter is the learning rate (LR). The LR may be the most important hyperparameter in the configuration of our model. The learning rate is the rate at which an algorithm converges to a solution when updating weights during training.

4.2. Evaluation Metrics

In this part, we present some evaluation metrics commonly used in histopathological classification. The analysis of these metrics is sufficient, although we have a multiclass problem [21].

We can define True Positive (TP) if the prediction is correct and refers to the predicted label. True Negative (TN) indicates the number of times the model correctly classified a negative sample as negative. True is defined by the correct match between the predicted label and the actual label. False represents a mismatch between the predicted and actual labels. False Positive (FP) means that the model incorrectly classifies a negative sample as positive, and False Negative (FN) means that a positive sample is classified as negative. In a multi-class classification, *Positive* and *Negative* refer to the individual label classes.

- The Confusion Matrix shows the classification performance of the model in validating the data sets.
- Accuracy represents the proportion of the true number of classified samples among all samples.

$$Accuracy = \frac{TP + TN}{Total} \quad (1)$$

where *Total* represents the sum of TP, TN, FP and FN.

- Precision measures how high the proportion of samples classified as positive is that are actually positive.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall indicates how many positive samples are classified as positive.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1-Score is the harmonic average of precision and recall.

$$F1Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (4)$$

- Receiver Operating Characteristic Curve (ROC curve) is a commonly used evaluation metric to assess the quality of a classifier. The classifier can also be evaluated by the area under the ROC curve, called Area Under Curve (AUC). It represents the rate of true positives versus the rate of false positives. The higher the value, the greater the discriminatory capacity of the model. AUC is defined by the rate of true

positives (TPR) and the rate of false positives (FPR). The TPR and FPR are given by $TPR = TP/(TP + FN)$ and $FPR = FP/(FP + TN)$. Thus, AUC can be computed by

$$AUC = \int_{x=0}^1 TPR(FPR^{-1}(x))dx = P(X_1 > X_0) \quad (5)$$

where X_1 is the score for a positive instance and X_0 is the score for a negative instance.

- Loss Function defines a method for evaluating how well the learning algorithm models the data set in terms of prediction. Categorical cross-entropy is a loss function used in classification tasks with multiple classes since it can consider a sample as belonging to one category with probability 1 and to other categories with probability 0. The loss for N classes, of which 9 classes are used in our work, is calculated as follows:

$$Loss = - \sum_{i=1}^N l_i \log(W_i) \quad (6)$$

The loss is calculated as a categorical cross-entropy between the ground true label l_i and W_i labels (Softmax is then applied to these labels, producing the values of the W_i predictions).

5. Experiments and Results

In this section, we present extensive experiments of the different methods of variation of the learning rate to which we add cutting-edge methods. The results (error indexes and computational time) of each training session is presented.

We have trained a CNN for the classification of nine different histological tissues. During training, some parameters should be considered to obtain the best performance of the proposed network with respect to the problem. If we don't have enough data, convolutional neural networks may overfit. Increasing the amount of data improves the generalizability of these networks by transforming images to make the network robust. Data augmentation is commonly used [22,23] to improve performance and avoid the problem of bias and overfitting. The effectiveness of data augmentation using simple techniques such as cropping, rotating, and flipping input images has been demonstrated in the literature [24,25], and has been used in this paper.

Figure 7 shows a representative sample of the different tissues from our dataset with random augmentation.

Normalization improves convergence speed and performance. We randomly flipped the images horizontally and vertically, and also applied a random rotation. In addition, the data were normalized by dividing each value by 255. The values of the images are between 0 and 255 and we want them to be between 0 and 1 for classification. In addition, for overcoming many of the known inconsistencies in the staining process in the preparation of histology slides, all images are color-normalized using Macenko's method [26]. After the pre-processing, different experiments have been carried out.

1. Finding the right LR significantly influences the CNN's ability to learn patterns. Given the importance of finding a suitable CNN architecture and the LR tuning, the influence of the LR range for fast convergence is analyzed.
2. Once the optimal learning rate range has been found, the influence of a constant learning rate in different training modes of the network has been studied.
3. An exhaustive and detailed study has been carried out with different non-constant learning rates.

5.1. Optimal Learning Rate

To obtain a good LR, the first step is to find the range that best fits our network, as this will lead the parameters of our model to optimal solutions at a reasonable pace. If the model is trained with very high or very low learning rate values, the model responds with very low accuracy and a huge loss [27]. To this end, the SGD batch is trained with

increasing LR and investigates when the model deviates. Plotting LR against loss, one can find the region of LR where the loss decreases most rapidly, as this is the steepest part of the graph. The training of the entire network consists of a range of learning rates 10^{-5} and 10^{-2} in 4 epochs. According to the slope of Figure 8.

We see that the optimal values are between 10^{-4} and 10^{-3} .

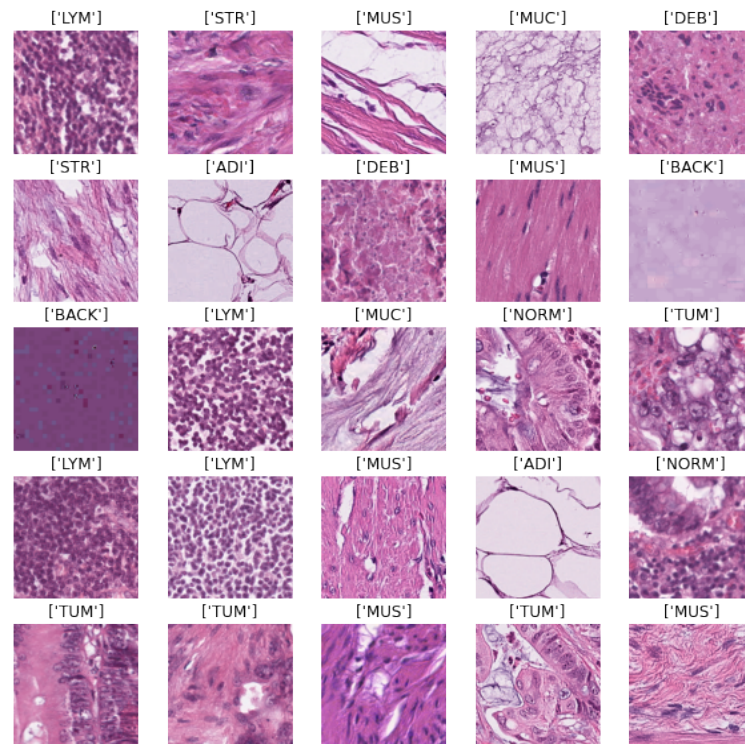


Figure 7. Dataset examples of the different tissues with random augmentation.

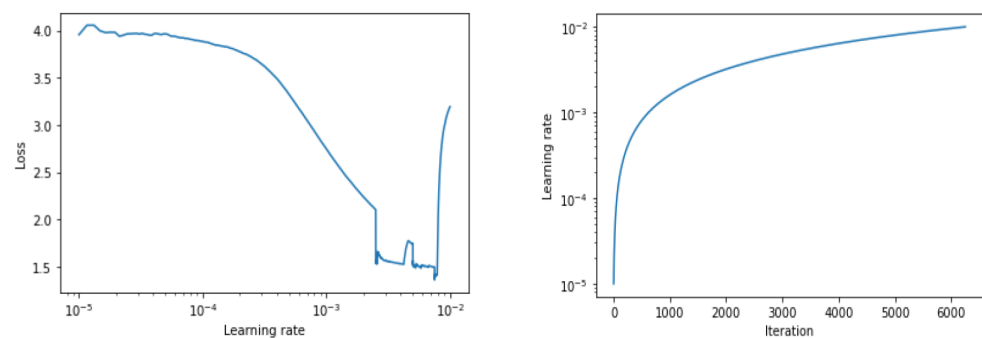


Figure 8. On the left is a graph of learning rate vs. loss to find the range of optimal values of LR; on the right, the iterations vs. learning rate.

5.2. Training from Scratch, Transfer Learning and Frozen Layers

Deep learning can be approached by building an architecture from scratch (by setting up different types of layers and connections), by using an existing network (Transfer Learning), and by Transfer Learning with frozen layers. In this section, we show the results obtained by training our model in these three ways using the LR found in the previous section. In this section, the simulations are performed with a $LR = 5 \cdot 10^{-4}$.

Learning from scratch means creating a completely new model and starting with random values in the weights of the neural model. In this first experiment, we train our model, with the parameters of Table 1, imposing the condition of non-pretrained weights with ImageNet, where we obtain an accuracy of 88%.

In transfer learning, there are a large number of convolutional networks that are trained by default with a set of images and focus on machine vision. In this process, information learned from one problem is used and applied to a related problem. This process usually involves training the final layers of the network or adding new ones, resulting in a reduction in the time required to train the network. We trained our VGG19 model using ImageNet pre-trained weights and re-trained all the layers. It is observed that the accuracy and training speed improve compared to training from scratch, obtaining an accuracy of 94%.

In the process of frozen layers, the aim is to slightly adjust the weights of some of the intermediate layers of the network in order to fine-tune the new problem to which Transfer Learning is applied. When the new model is initialized, the first layers are "frozen" so as not to change their weights, as these are more general feature extraction layers, and the remaining layers are trained with the new database [28]. In this part of the experiment, we freeze 15%, 25% and 50% of the 27 layers that complete our model, obtaining an accuracy of 87%, 83% and 49%, respectively. With this method of freezing layers, the model needs to be trained with fewer parameters and therefore takes less time to train, but the results are worse than with the transfer learning process without freezing layers. Table 2 shows the results obtained in this section, the Precision, Recall, F1 score for each tissue class, Accuracy, Training time and the parameters that can be trained.

Table 1. List of parameters and hyperparameters for our framework.

Parameters and Hyperparameters	Value
Input shape	224×224
Bach size	64
Number of layers	27
Momentum	0.9
Learning rate	10^{-5} – 10^{-1}
Epochs	10
Loss function	Categorical cross entropy
Optimizer	SGD
Weights	Scratch- ImageNet
Dropout	0.5
L2	10^{-3}
Activation	ReLU/Softmax
Dense	5000

Since the best results are obtained with the Transfer Learning training, all subsequent training with our model will follow this strategy.

5.3. Cyclical Learning Rate

This method is based on the fundamental idea that the learning rate varies within a range of values, rather than assuming a fixed value. The learning rate varies cyclically between fixed limits [10]. A short run of only a few epochs in which the learning rate increases linearly is sufficient to estimate the boundary learning rates for the cyclic learning rate (CLR). In this section, we test several adapted synchronization cycles and training iterations (the results are in Table 3). We fit the strategies to our dataset and model. To this end, we experimented with different cyclic functions to test the performance of our model for classifying histopathological images. The features discussed in this work are the following, illustrated in Figure 9. For these experiments a range of LR has been taken between the base value 10^{-4} and a maximum LR value of 10^{-3} .

1. Triangular: The LR adopts a triangular window linearly increasing and then linearly decreasing on a regular basis. Obtaining an accuracy of 96%.
2. Triangular drop: The difference with the previous method is that the upper and lower limits are halved after each cycle without affecting the predefined learning rates. In this case, the metrics are almost unchanged from the previous method.
3. Exponential: The learning rate varies between the minimum and maximum limits and each value of the limit decrease by an exponential factor gamma. This means the learning rate difference drops after each cycle. In this case, this factor is equal to unity and for Exponential 2, gamma is 0.9. This method achieves an accuracy of 95% and 94%, respectively.
4. Sine: The strategy based on a sine function decay is based on a cyclic triangular decay with a sinusoidal amplitude, where the learning rate of the iterations decreases by a fixed amount. With this strategy, the accuracy decreases by up to 92%.

Table 2. Results obtained in the training and validation process from scratch, with transfer learning and frozen different percentages of the model layers.

Model: VGG19						
Metrics	Class	Scratch	Transfer Learning	Transfer Learning + Frozen Layers		
				15% Frozen	25% Frozen	50% Frozen
Precision	ADI	0.98	0.90	0.97	0.99	0.93
	BACK	0.95	0.99	0.96	0.97	0.87
	DEB	0.53	0.94	0.57	0.24	0.04
	LYM	0.94	0.99	0.93	0.90	0.16
	MUC	0.96	1.00	0.98	0.95	0.31
	MUS	0.69	0.84	0.64	0.61	0.26
	NORM	0.74	0.89	0.80	0.75	0.46
	STR	0.72	0.88	0.67	0.40	0.00
	TUM	0.95	0.99	0.88	0.96	0.36
Recall	ADI	0.96	1.00	0.98	0.97	0.89
	BACK	1.00	1.00	1.00	1.00	1.00
	DEB	0.62	1.00	0.48	0.25	0.03
	LYM	0.99	1.00	0.94	0.98	0.23
	MUC	0.83	0.85	0.91	0.95	0.15
	MUS	0.85	0.92	0.83	0.48	0.06
	NORM	0.94	0.99	0.83	0.89	0.02
	STR	0.30	0.70	0.46	0.55	0.00
	TUM	0.92	0.93	0.88	0.78	0.90
F1- Score	ADI	0.97	0.95	0.98	0.98	0.91
	BACK	0.98	1.00	0.98	0.99	0.93
	DEB	0.57	0.97	0.52	0.24	0.03
	LYM	0.97	0.99	0.94	0.94	0.19
	MUC	0.89	0.92	0.95	0.95	0.20
	MUS	0.76	0.88	0.72	0.54	0.10
	NORM	0.83	0.94	0.81	0.81	0.05
	STR	0.43	0.78	0.54	0.46	0.00
	TUM	0.94	0.96	0.88	0.86	0.51
Accuracy	Train	0.91	0.99	0.83	0.82	0.39
	Test	0.88	0.94	0.87	0.83	0.49
Training time		157 m 6 s	155 m 47 s	128 m 26 s	109 m 37 s	72 m 21 s
Trainable params		170,519,393	170,519,393	170,480,673	170,259,233	164,653,857

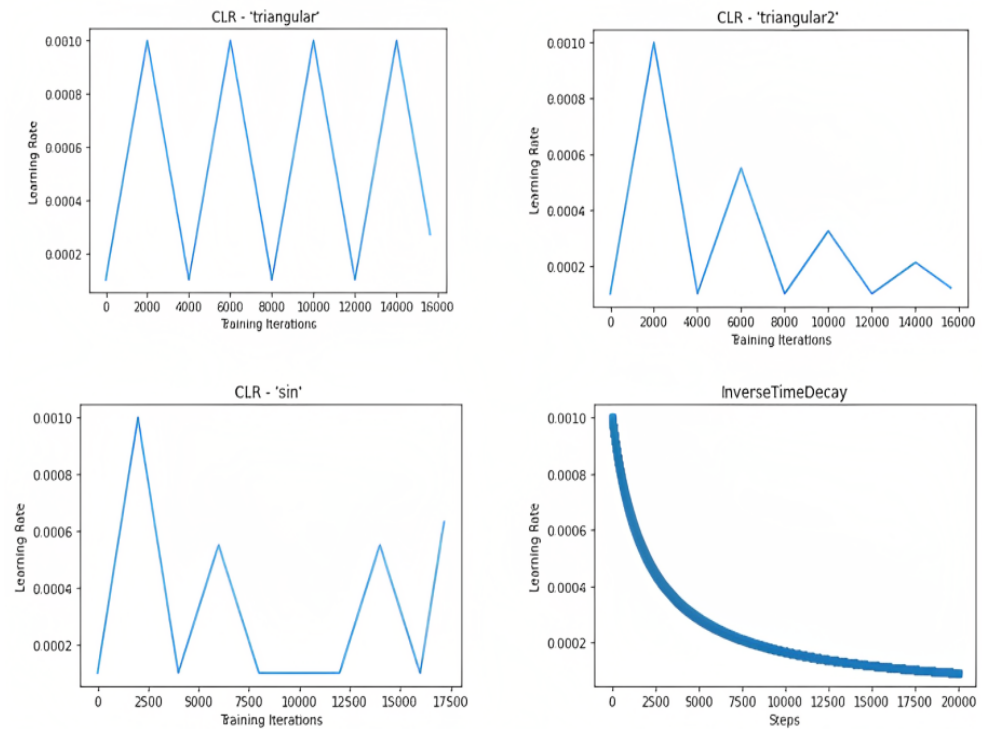


Figure 9. Different learning rate strategies versus training iterations. In the upper left corner, the Triangular LR strategy is depicted, and in the upper right corner, the Triangular 2 adaptation. In the lower-left corner, the LR strategy with the sinusoidal function and in the lower right corner is the Inverse Time Decay strategy.

5.4. Scheduler Learning Rate

In this section, we present a predefined framework, where different LR decay strategies are used without specifying the cyclic values. The results are in Table 4.

1. Decay 1: In this case, we examine how the learning rate changes after the entire epoch rather than for individual steps, using functions that take as input the number of epochs and the current learning rate to provide a new learning rate. This experiment achieves 93% accuracy, where in the first epoch the learning rate is $LR_{initial} = 0.001$, so in each epoch the current LR is divided by 3. In Decay 2, LR is divided by a value of 1.5 to observe a smoother decay, in which case the accuracy is improved to nearly 95%.
2. Step decay: We train our network using SGD and a scheduler with inverse time decay. The following formula $LR_{step} = LR_{initial} / (1 + \frac{DR \cdot Step}{DS})$ is used to calculate the learning rate at each step. Here, DR is the decay rate with a value of 0.5, $LR_{initial} = 0.001$ and DS is the decay step and defines the number of steps after which the learning rate decays. In our case, it is 1000, which gives an accuracy almost equal to the previous one.
3. Polynomial decay [7]: In this part, our network is trained using SGD with polynomial decay. With an initial training and final learning rate between the computed bounds in the optimal convergence region for training. This method improves the accuracy a little by achieving an accuracy of 95%.
4. Piecewise Constant Decay: In this last experiment of this section, we train our network using SGD with a piecewise constant decay scheme. In this process, LR decreases by a few steps. We give LR values for each step, with a constant learning rate every 1000 steps. An accuracy of 95% is achieved.

5.5. Discriminative Fine Tuning

So far, a comparison has been made with different ways of working the LR while keeping it constant for all layers of the network. In this section, a novel technique is proposed. Howard et al. proposes a novel method for the classification of texts. Discriminative fine-tuning [29] allows us to tune each layer with different learning rates. Different layers capture different types of information [30], so a tuning according to the capacity of each layer gives good results.

This method is based on the fact that the SGD update with discriminative fine-tuning is then the following:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \nabla_{\theta^l} J(\theta) \quad (7)$$

where η is the learning rate and $\nabla_{\theta} J(\theta)$ is the gradient with regard to the model's objective function. For discriminative fine-tuning, θ_t is replaced by $\{\theta^1, \dots, \theta^L\}$, where θ^l contains the parameters of the model at the l-th layer. Similarly η is replaced by $\{\eta^1, \dots, \eta^L\}$, where η^l is the learning rate of the l-th layer and L is the number of layers of the model.

Table 3. The impact of using different methods of cyclical variation of the LR with respect to epochs on the training and validation of our model.

VGG19						
Metrics	Class	Cyclical Learning Rate				
		Triangular	Triangular2	Exponential	Exponential2	Sine
Precision	ADI	0.99	0.99	0.95	0.99	0.99
	BACK	1.00	1.00	1.00	1.00	1.00
	DEB	0.98	0.95	0.95	0.91	0.98
	LYM	0.99	0.99	0.99	0.99	0.99
	MUC	1.00	0.98	0.99	0.98	1.00
	MUS	0.84	0.87	0.87	0.83	0.59
	NORM	0.94	0.92	0.96	0.93	0.90
	TUM	0.78	0.81	0.75	0.69	0.90
Recall	ADI	0.97	0.97	0.95	0.97	0.99
	BACK	0.98	0.99	0.99	0.97	0.82
	DEB	1.00	1.00	1.00	1.00	1.00
	LYM	1.00	1.00	1.00	0.96	1.00
	MUC	0.98	0.98	0.92	0.98	0.97
	MUS	0.85	0.84	0.82	0.72	0.95
	NORM	0.97	0.98	0.96	0.96	0.99
	TUM	0.77	0.76	0.77	0.82	0.65
F1-Score	ADI	0.97	0.96	0.98	0.96	0.94
	BACK	0.98	0.99	0.97	0.98	0.90
	DEB	1.00	1.00	1.00	1.00	1.00
	LYM	0.99	0.97	0.97	0.95	0.99
	MUC	0.99	0.98	0.96	0.98	0.99
	MUS	0.84	0.86	0.84	0.77	0.73
	NORM	0.96	0.95	0.96	0.95	0.94
	TUM	0.78	0.78	0.76	0.75	0.75
Accuracy	Train	0.99	0.99	0.99	0.99	0.99
	Test	0.96	0.96	0.95	0.94	0.92
Training time		158 m 9 s	155 m 7 s	157 m 40 s	156 m 59 s	173 m 52 s

We decrease the learning rate with the increasing depth of the layers and are thus in the optimal range for convergence. The LR factors for the different layers assigned a name to

specify this value were chosen per layer bundle, i.e.: The first ten layers have a factor of 0.01, the next thirteen layers have a factor of 0.001, the last four layers reduce their factor to 0.0001.

In multiclass problems, it is important to obtain the so-called confusion table to analyze the behavior of the system. The Figure 10 shows the behavior of the model when using discriminative fine-tuning with a decreasing LR in epochs. Both the confusion matrix and the confidence matrix are shown. The confidence matrix is similar to a confusion matrix, but we are instead measuring the average probability of each decision. AUC-ROC is shown as the precision measure. The evolution of the accuracy of the system and the loss function with the number of epochs (indicating the evolution of learning) is also shown.

Table 4. The impact of using different methods of scheduler variation of the LR with respect to epochs on the training and validation of our model

VGG19						
Metrics	Class	Learning Rate Scheduler				
		Decay 1	Decay 2	Step Decay	Polinomial Decay	Piece Cte Decay
Precision	ADI	0.99	0.99	0.99	0.99	0.99
	BACK	0.99	1.00	1.00	1.00	1.00
	DEB	0.96	0.97	0.99	0.97	0.98
	LYM	0.98	0.99	0.99	0.99	0.99
	MUC	0.93	0.96	0.96	0.99	0.98
	MUS	0.82	0.85	0.85	0.83	0.87
	NORM	0.94	0.93	0.91	0.93	0.91
	STR	0.68	0.69	0.73	0.75	0.72
	TUM	0.94	0.96	0.96	0.98	0.97
Recall	ADI	0.98	0.98	0.98	0.97	0.99
	BACK	1.00	1.00	1.00	1.00	1.00
	DEB	0.96	0.99	0.99	1.00	1.00
	LYM	0.99	0.99	0.99	1.00	1.00
	MUC	0.98	0.98	0.99	0.98	0.98
	MUS	0.78	0.77	0.80	0.80	0.77
	NORM	0.90	0.95	0.94	0.97	0.95
	STR	0.69	0.73	0.69	0.80	0.78
	TUM	0.94	0.96	0.95	0.96	0.95
F1-Score	ADI	0.98	0.99	0.99	0.98	0.99
	BACK	1.00	1.00	1.00	1.00	1.00
	DEB	0.96	0.98	0.99	0.99	0.99
	LYM	0.99	0.99	0.99	0.99	0.99
	MUC	0.96	0.97	0.97	0.98	0.98
	MUS	0.80	0.81	0.83	0.82	0.82
	NORM	0.92	0.94	0.92	0.95	0.93
	STR	0.69	0.71	0.71	0.77	0.75
	TUM	0.94	0.96	0.95	0.97	0.96
Accuracy	Train	0.94	0.98	0.98	0.9895	0.9830
	Tests	0.93	0.95	0.95	0.9532	0.9494
Training time		155 m 22 s	153 m 5 s	156 m 7 s	153 m 29 s	153 m 23 s

5.6. Comparison with Other Methodologies

Table 5 summarizes the relevant methods that appear in the bibliography, which use deep learning models for multiclass classification. In Table 5, there are strategies for dynamic modification of the learning rate, such as cyclical learning rates (Smith et al. [10]), polynomial learning rates (Purnendu et al. [7]) or dynamic learning rates (Anil et al. [9]). There are also static learning rate methods, such as that of Anil et al. [8], and methodologies based on transfer learning (such as that of Alinsaif et al. [14]). The proposed method achieves high accuracy in test images by performing an in-depth study of the optimal strategy for the learning rate. We

believe that the proposed methodology provides excellent accuracy results through a detailed and comprehensive study of the optimal strategy for the learning rate.

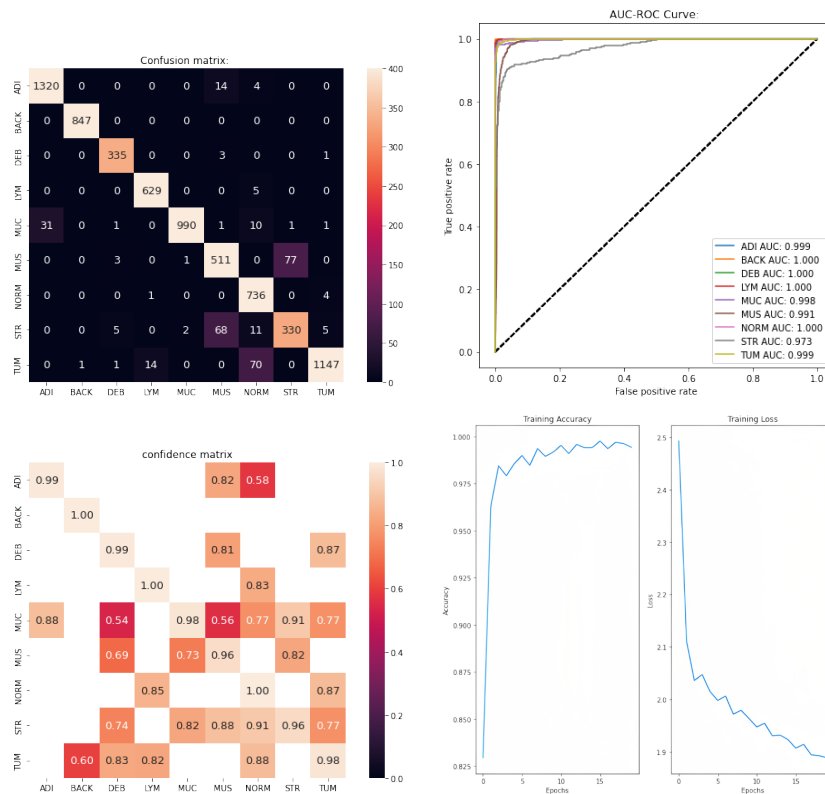


Figure 10. Results obtained by applying discriminative fine tuning with a decreasing LR in epochs. The confusion matrix, the Receiver Operating Characteristic Curve (ROC curve) and the Area Under the Curve (AUC), the confidence matrix and, in the lower right corner, the accuracy and the training loss function versus epochs are shown. In this example, a larger number of epochs has been considered to appreciate the graph.

Table 5. Comparative analysis of related work available in the literature on the study of the learning rate and multi-class data-set classification.

Year	Paper	Methodology	ACC (%)	AUC
2016	Kather et al. [11]	Lower-order and higher-order histogram features Local binary patterns Gray-level co-occurrence matrix	87.4 %	0.968
2017	Ciampi et al. [12]	Stain normalization and ConvNet	79.66%	-
2017	Wang et al. [31]	Convolutional neural networks bilinear (BCNN)	92.6%	0.985
2017	Bianconi et al. [13]	Variation of LBP with point-to-average thresholding	93.4 %	-
2017	Smith et al. [10]	Cyclical learning rates	82.2%	-
2018	Cascianelli et al. [32]	Reduction strategy based on the cross-correlation	84.8%	-
2019	Purnendu et al. [7]	Polynomial learning rate	94.54%	-
2020	Alinsaif et al. [14]	Reduced Deep Features Fine-Tuning	95.40% 95.02%	0.9961 0.9976
2020	Anil et al. [8]	Fixed learning rate	92.8%	0.97
2021	Anil et al. [9]	Dynamic learning rates	91.84%	0.97
	Proposed method	Transfer Learning	Test: 94.4%	0.98
	Proposed method	Schedule Learning Rate (Polynomial decay)	Test: 95.3%	0.98
	Proposed method	Cyclical Learning Rate (Triangular)	Test: 96.4%	0.99

6. Conclusions

This paper systematically analyzes various parameters, hyperparameters and methods for training and optimizing deep learning systems for multiclass classification. The dataset used for training and testing includes various healthy tissues and colorectal cancer. It is important to note that gradient descent is widely used in large-scale optimization problems in machine learning; in particular, it plays an important role in computing and tuning the connection weights of deep learning models. Gradient-based optimization methods have hyperparameters that require infinite possibilities for configuration. Determining the values and optimization methodology of the hyperparameters of a deep-learning system is currently a challenge that is important for the behavior and precision of the system. Moreover, these hyperparameters also affect the computation time and cost. In this paper, the performance of a deep learning model based on the well-known VGG19 structure was evaluated, using three different methods for its training: learning from scratch (i.e., all parameters composing the different levels of the neural network, including the CNN levels, are tuned thanks to the learning phase), transfer learning (using a VGG19 system previously optimized in other classification problems, all parameters are optimized/tuned with the images of the new problem), and transfer learning associated with frozen layers (only a subset of the parameters belonging to the last layers is optimized).

It was analyzed (obtaining different error metrics) how these different strategies have a different behavior in the time necessary for the training of the neural system and to the accuracy of the system. The system that requires the most time is learning from scratch. The system that learns the fastest (less than half the time for learning from scratch) is transfer learning + frozen layers (with a total of 50% of the original layers frozen or not modified). In terms of precision, transfer learning produced the best results. Therefore, this strategy was used for the following analysis in this article: the effect of the learning rate. The learning rate is one of the most important hyperparameters in a neural network and, of course, in deep learning models.

In this article, various strategies for LR optimization, both static and dynamic, have been analyzed in depth. Different dynamic cyclic functions (triangular, triangular drop, exponential, sinusoidal) were used to test the performance of our model for classification of histopathological images. Four different frameworks for LR decay were also used: Decay 1 (decay after a complete epoch), step decay, polynomial decay and piecewise constant decay. There is no significant impact of using different methods of scheduler variation of LR in terms of computation time required, but from the point of view of accuracy, the best method is polynomial decay.

Finally, discriminative fine-tuning is also analyzed as a novel technique proposed in this paper in conjunction with dynamic LR strategies. Discriminative fine-tuning allows tuning layers of the deep learning model with different learning rates.

The results obtained are very remarkable since in the simulation an accurate system that achieves an accuracy of 96.4% and a value close to 1 for the AUC in test images is obtained (for nine different tissue classes), using the triangular-cyclic learning rate.

As future work, the LR strategy and deep learning model can be trained and tested on other cancer data-sets for classification, but taking into account the possible adaptation and restrictions of the new problem.

Author Contributions: Conceptualization, E.M.-F., I.R.-V., O.V. and I.R.; methodology, E.M.-F., I.R.-V., O.V. and I.R.; writing—original draft preparation, E.M.-F., I.R.-V. and I.R.; writing—review and editing, E.M.-F., I.R.-V. and O.V.; visualization, E.M.-F.; supervision, O.V. and I.R.; project administration, I.R.; funding acquisition, I.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Spanish Ministry of Sciences, Innovation, and Universities under Project PID2021-128317OB-I00 in collaboration with the Government of Andalusia under Project P20-00163.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available online. Details in Section 3. The dataset is available at <https://zenodo.org/record/1214456#.YqcMJqhBxaY>, accessed on 10 January 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Egeblad, M.; Nakasone, E.S.; Werb, Z. Tumors as Organs: Complex Tissues that Interface with the Entire Organism. *Dev. Cell* **2010**, *18*, 884–901. [[CrossRef](#)]
2. Pantanowitz, L.; Farahani, N.; Parwani, A. Whole slide imaging in pathology: Advantages, limitations, and emerging perspectives. *Pathol. Lab. Med. Int.* **2015**, *7*, 23–33.
3. Li, X.; Chen, H.; Li, C.; Rahaman, M.M.; Li, X.; Wu, J.; Li, X.; Sun, H.; Grzegorzec, M. What Can Machine Vision Do for Lymphatic Histopathology Image Analysis: A Comprehensive Review. *arXiv* **2022**, arXiv:2201.08550.
4. Zhang, Y.D.; Nayak, D.R.; Zhang, X.; Wang, S.H. Diagnosis of secondary pulmonary tuberculosis by an eight-layer improved convolutional neural network with stochastic pooling and hyperparameter optimization. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *1*–18. [[CrossRef](#)]
5. Nayak, S.R.; Nayak, D.R.; Sinha, U.; Arora, V.; Pachori, R.B. An Efficient Deep Learning Method for Detection of COVID-19 Infection Using Chest X-ray Images. *Diagnostics* **2022**, *13*, 131. [[CrossRef](#)] [[PubMed](#)]
6. Zhang, X.; Wang, T.; Yang, Y. Classification of Small-Sized Sample Hyperspectral Images Based on Multi-Scale Residual Network. *Laser Optoelectron. Prog.* **2020**, *57*, 162801. [[CrossRef](#)]
7. Mishra, P.; Sarawadekar, K. Polynomial Learning Rate Policy with Warm Restart for Deep Neural Network. In Proceedings of the TENCN 2019—2019 IEEE Region 10 Conference (TENCN), Kerala, India, 17–20 October 2019. [[CrossRef](#)]
8. Johnny, A.; KN, D.M.; Nallikuzhy, D.T.J. Optimization of CNN Model With Hyper Parameter Tuning for Enhancing Sturdiness in Classification of Histopathological Images. *SSRN Electron. J.* **2020**. [[CrossRef](#)]
9. Johnny, A.; Madhusoodanan, K.N. Dynamic Learning Rate in Deep CNN Model for Metastasis Detection and Classification of Histopathology Images. *Comput. Math. Methods Med.* **2021**, *2021*, 5557168. [[CrossRef](#)]
10. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 May 2017. [[CrossRef](#)]
11. Kather, J.N.; Weis, C.A.; Bianconi, F.; Melchers, S.M.; Schad, L.R.; Gaiser, T.; Marx, A.; Zöllner, F.G. Multi-class texture analysis in colorectal cancer histology. *Sci. Rep.* **2016**, *6*, 27988. [[CrossRef](#)] [[PubMed](#)]
12. Ciompi, F.; Geessink, O.; Bejnordi, B.E.; de Souza, G.S.; Baidoshvili, A.; Litjens, G.; van Ginneken, B.; Nagtegaal, I.; van der Laak, J. The importance of stain normalization in colorectal tissue classification with convolutional networks. In Proceedings of the 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, Australia, 18–21 April 2017. [[CrossRef](#)]
13. Bianconi, F.; Bello-Cerezo, R.; Napoletano, P. Improved opponent color local binary patterns: an effective local image descriptor for color texture classification. *J. Electron. Imaging* **2017**, *27*, 1. [[CrossRef](#)]
14. Alinsaif, S.; Lang, J. Histological Image Classification using Deep Features and Transfer Learning. In Proceedings of the 2020 17th Conference on Computer and Robot Vision (CRV), Los Alamitos, CA, USA, 13–15 May 2020. [[CrossRef](#)]
15. Kather, J.N.; Halama, N.; Marx, A. 100,000 Histological Images of Human Colorectal Cancer and Healthy Tissue, 2018. Available online: <https://zenodo.org/record/1214456#.ZCz0zvZByUl> (accessed on 10 January 2023). [[CrossRef](#)]
16. Hameed, Z.; Zahia, S.; Garcia-Zapirain, B.; Aguirre, J.J.; Vanegas, A.M. Breast Cancer Histopathology Image Classification Using an Ensemble of Deep Learning Models. *Sensors* **2020**, *20*, 4373. [[CrossRef](#)]
17. Sharma, S.; Mehra, R. Conventional Machine Learning and Deep Learning Approach for Multi-Classification of Breast Cancer Histopathology Images—A Comparative Insight. *J. Digit. Imaging* **2020**, *33*, 632–654. [[CrossRef](#)] [[PubMed](#)]
18. Broad, A.; Wright, A.I.; de Kamps, M.; Treanor, D. Attention-guided sampling for colorectal cancer analysis with digital pathology. *J. Pathol. Inform.* **2022**, *13*, 100110. [[CrossRef](#)]
19. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
20. Bottou, L. Stochastic Gradient Descent Tricks. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436. [[CrossRef](#)]
21. Grandini, M.; Bagli, E.; Visani, G. Metrics for Multi-Class Classification: An Overview. *arXiv* **2020**, arXiv:2008.05756.
22. Senaras, C.; Niazi, M.K.K.; Arole, V.; Chen, W.; Sahiner, B.; Shana'ah, A.; Louissaint, A.; Hasserjian, R.P.; Lozanski, G.; Gurcan, M.N. Segmentation of follicles from CD8-stained slides of follicular lymphoma using deep learning. In Proceedings of the Medical Imaging 2019: Digital Pathology, San Diego, CA, USA, 20–21 February 2019; Tomaszewski, J.E.; Ward, A.D., Eds.; SPIE: Bellingham, DC, USA, 2019. [[CrossRef](#)]
23. Kandel, I.; Castelli, M. A Novel Architecture to Classify Histopathology Images Using Convolutional Neural Networks. *Appl. Sci.* **2020**, *10*, 2929. [[CrossRef](#)]
24. Rajput, S.; Feng, Z.; Charles, Z.; Loh, P.L.; Papailiopoulos, D. Does Data Augmentation Lead to Positive Margin? *arXiv* **2019**, arXiv:1905.03177.
25. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [[CrossRef](#)]

26. Macenko, M.; Niethammer, M.; Marron, J.S.; Borland, D.; Woosley, J.T.; Guan, X.; Schmitt, C.; Thomas, N.E. A method for normalizing histology slides for quantitative analysis. In Proceedings of the 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Boston, MA, USA, 28 June–1 July 2009. [\[CrossRef\]](#)
27. Rathore, P.S.; Dadich, N.; Jha, A.; Pradhan, D. Effect of learning rate on neural network and convolutional neural network. *Int. J. Eng. Res. Technol.* **2018**, *6*, 1–8.
28. Vrbancic, G.; Podgorelec, V. Transfer Learning With Adaptive Fine-Tuning. *IEEE Access* **2020**, *8*, 196197–196211. [\[CrossRef\]](#)
29. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. *arXiv* **2018**, arXiv:1801.06146.
30. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *arXiv* **2014**, arXiv:1411.1792.
31. Wang, C.; Shi, J.; Zhang, Q.; Ying, S. Histopathological image classification with bilinear convolutional neural networks. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju Island, Republic of Korea, 11–15 July 2017. [\[CrossRef\]](#)
32. Cascianelli, S.; Bello-Cerezo, R.; Bianconi, F.; Fravolini, M.L.; Belal, M.; Palumbo, B.; Kather, J.N. Dimensionality Reduction Strategies for CNN-Based Classification of Histopathological Images. In *Intelligent Interactive Multimedia Systems and Services 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 21–30. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.