The Journal of Supercomputing (2021) 77:9494–9519
https://doi.org/10.1007/s11227-020-03575-6

# A novel transfer learning approach for the classification of histological images of colorectal cancer

Elene Firmeza Ohata[1] · João Victor Souza das Chagas[1] · Gabriel Maia Bezerra[1] · Mohammad Mehedi Hassan[2] · Victor Hugo Costa de Albuquerque[1,3,4] · Pedro Pedrosa Rebouças Filho[1,3,4]

Accepted: 14 December 2020 / Published online: 10 February 2021

## Abstract

Colorectal cancer (CRC) is the second most diagnosed cancer in the United States. It is identified by histopathological evaluations of microscopic images of the cancerous region, relying on a subjective interpretation. The Colorectal Histology dataset used in this study contains 5000 images, made available by the University Medical Center Mannheim. This approach proposes the automatic identification of eight types of tissues found in CRC histopathological evaluation. We apply Transfer Learning from architectures of Convolutional Neural Networks (CNNs). We modify the structures of CNNs to extract features from the images and input them to well-known machine learning methods: Naive Bayes, Multilayer Perceptron, k-Nearest Neighbors, Random Forest, and Support Vector Machine (SVM). We evaluated 108 extractor–classifier combinations. The one that achieved the best results is DenseNet169 with SVM (RBF), reaching an Accuracy of 92.083% and F1-Score of 92.117%. Therefore, our approach is capable of distinguishing tissues found in CRC histopathological evaluation.

**Keywords** Transfer learning · Colorectal cancer · Histological images · Convolutional neural networks

## 1 Introduction

Between 2008 and 2017, the incidence rates of colorectal cancer (CRC) increased; the growth rates reached 2% per year. Consequently, the mortality rates increased by 1.3% per year, characteristic of fragility in young adults under the age of 50 years [45]. CRC corresponds to the second most diagnosed type of cancer in the United States of America [46]. This type of cancer is commonly specified according to its

✉ Victor Hugo Costa de Albuquerque
victor.albuquerque@ieee.org

Extended author information available on the last page of the article

initial location, such as colon cancer or rectal cancer. Its incidence and mortality rates are 30–40% higher in men than in women [1, 45]. Globally, it is estimated that by 2030 the incidence of CRC will increase by 60%, reaching values above 2.2 million, which will cause about 1.1 million deaths [4, 46].

Countries with developed economies present the highest CRC incidence and mortality rates [4]. On the other hand, it is noteworthy that the most significant increases in these rates are found in developing countries, highlighting the need to search for efficient and accessible diagnostics to meet this global need. This socioeconomic diversity of the countries promotes variations of up to 10 times in the CRC incidence and mortality rates [4].

The diagnosis of CRC is based on microscopic images of the cancerous area. These images comprise histological exams defined by procedures that enable quality in the microscopic evaluation of the tissue. In the proceedings, the tissue is allocated with 10% neutral formalin, moistened in paraffin, and divided manually by a microtome to produce sections with a thickness of 4–5 μm [48]. Specialists perform the delimitation and classification of the regions found visually and, consequently, subjectively. The subjectivity in diagnosis is due to the similarity of some types of tissues found in the exams [3]. Due to this high similarity, tissues' visual classification can be compromised by tiredness, stress, experience, and human capacity, making automatic systems for classification necessary. Such a classification tool can be applied in medical imaging laboratories to cooperate with the work of technicians and doctors. For instance, if the specialist's classification differs from that of the automatic system, the investigated image will be sent to a revision step, where a specialist will investigate it with greater attention.

The CRC histopathological evaluation is commonly known as histopathological evaluation of Hematoxylin and Eosin (H&E) because of the application of H&E in dewaxed sections [3, 48]. This evaluation of stained tissue sections is complex since the examined structure has different tissues, including tumor tissues. Based on this, the Colorectal Cancer Histology dataset was proposed in [25]. This dataset represents a collection of microscopic images containing the different types of tissues found. The challenge consists of classifying the eight types of tissues presented and highlighting which structures are most similar [25].

Due to the global need for CRC diagnosis, the tools employed in Deep feature extraction and Transfer Learning can be applied and contribute to an effective method of diagnosing cancer. Some other approaches have obtained powerful methods for difficult diagnoses such as aid in the identification and classification of traumas found in computed tomography strokes [24, 41, 42], the malignancy classification of lung nodules [11, 23], the diagnostic of patients with common treatable blinding retinal diseases [8, 28], and the continuous monitoring, based on tests and using the diagnosis of cardiovascular diseases [31, 40].

In the literature, several approaches have already proposed alternatives to automate the classification of histopathological exams, for example, in the classification of cell nuclei found in histopathological images of breast cancer [2, 13, 55, 56], colorectal cancer [5, 25, 27, 43, 57], and lung cancer [10, 14, 58, 59]. In summary, the approaches above are divided into two types, classification based on Convolutional Neural Networks (CNNs) and supervised machine learning approaches with feature extraction methods. In these approaches, the resource extraction stage is conducive to limitations due to the segmentation step and the reduction step of resources based on a chosen characteristic, possibly restricting the proposed system's generalization capacity.

From this, we developed a method for multi-class classification structured in Deep feature extraction, Transfer Learning, and extensive feature classification using Colorectal Cancer Histology dataset. The proposed system's main advantage is to combine efficient aspects of the methods prevalent in the literature. More precisely, creating combinations with the CNNs' feature extraction and the classic machine learning algorithms for classification. Finally, the extractor–classifier combination with the best performance can be applicable in laboratories to analyze histopathological tissues. The approach produces 108 experiments of supervised image classification from the combination of 18 feature extractors with six traditional classifiers. The results present that the combination DenseNet169 with SVM RBF achieved 92.083% and 92.117% for Accuracy and F1-Score, respectively. The proposed method offers the following contributions:

- Use of Deep feature extraction through transfer learning and extensive feature classification for the identification of tissues found in CRC histopathological evaluation;
- Generation of different combinations of extractors and classifiers in the identification of tissues by histopathological exams.
- A fast and precise method to classify CRC histopathological images that assists specialists in giving more accurate diagnoses.

The present paper is divided into sections and organized as follows. Section 2 discusses the related work, Sect. 3 introduces the transfer learning concept, and in Sect. 4, we explain the classifiers which compose the method. Section 5 explains the methodology of the proposed approach, Sect. 6 presents the experimental results, and in Sect. 7, we summarize the conclusions of the experiments.

## 2 Related works

In the literature, several approaches have proposed accelerating and automating the diagnosis of histological exams. The analysis of tissues found in histological images is widespread in several diseases such as breast cancer, colorectal cancer, and lung cancer. The various uses of histopathological evaluation (H&E) demonstrate the vast scope for improvement in the use of histological exams and the importance of contrasting these approaches.

In breast cancer, Xu et al. [56] propose a nuclei detection based on Stacked Sparse Autoencoder (SSAE) instantiated via deep learning but with an unbalanced amount of nuclear and non-nuclear samples. Wang et al. [55] qualitatively characterize the images from the segmentation of the cores, which later, with the extraction of shape and texture characteristics, are classified only in binary form. Gecer et al. [13] present the classification into five diagnostic categories based on the concatenation of four fully convolutional networks. These CNNs provide merging of maps to label each histopathology slides introduced and make the proposed classification, in summary, with high computational costs.

Also, the approach by Coudray et al. [10] performs the classification of the most prevalent subtypes of lung cancer, which are adenocarcinoma (LUAD) and squamous cell carcinoma (LUSC), distinguishing them from healthy tissues. Yu et al. [59] also perform a tissue classification between LUAD and LUSC, more precisely, the classification of transcriptomic subtypes in non-small cell lung cancer (NSCLC). Another classification example by Yu et al. [58] is based on regularized machine learning methods and quantitative image features to extract resources to determine short- or long-term survivors with LUAD and, or LUSC.

In particular, for colorectal cancer, Xu et al. [57] proposed segmentation and multi-classification based on the use of transfer learning and the study of the activation in the last hidden layer. The authors apply a CNN trained on the ImageNet database as a feature extractor and individually observe the sensitivity of each neuron that makes up the hidden layer.

Additionally, Bayramoglu et al. [5] proposed the CRC cell nuclei classification into three classes: epithelial nuclei, fibroblasts, and Inflammatory nuclei. The paper by Bayramoglu et al. covers the problematization and use of deep learning, transfer learning, and fine-tuning to assist in diagnosing CRC cell nuclei. The CNNs used were AlexNet, GoogLeNet, VGG-16, and GenderNet. The authors use all the layers of the CNNs except the last fully connected layer, which is adapted for the classification of the three classes of cell nuclei proposed.

The classification of tissues found in CRC histopathological evaluation is addressed by Kather et al. [27] through training of a CNN via transfer learning. The transfer learning step has a training dataset of 100 thousand histopathological images, divided into nine types of tissue. Another example of multi-classification proposed by Kather et al. [25] corresponds to the analysis of tissue textures found in CRC histopathological images. More precisely, this paper is responsible for presenting the dataset used in our method. In Kather's approach [25], the texture extraction and classification strategy were based on the empirical choice of hyperparameters, demonstrating a weakness in selecting the best combination achieved. In contrast, our approach uses the grid search or random search methods for the hyperparameter selection.

Therefore, the approaches above are mainly divided into two types: classification based on CNNs and supervised machine learning approaches with feature extraction methods. Methods based on cell morphology, defined by segmentation, individual classification of the target regions, and final classification of the patch, are part of the approaches with CNNs.

The methods structured on biomarkers above perform the feature extraction focused on a prominent property of the image, such as the texture, thus reducing the number of total features extracted and classified. In summary, the extraction steps mentioned above are duly dependent on the region's initial segmentation and the correct parameter choice for feature reduction. The addition of these steps can limit the proposed system performance.

Moreover, we highlight some approaches to diagnosing diseases with segmentation and classification methods inserted into Internet of Things (IoT) systems. For example, Tseng et al. [51] achieved excellent cardiac segmentation results with the U-Net network, which was adapted and included in an IoT system for liver cancer segmentation [54]. The classification systems are also highlighted in medical diagnosis as in [16, 32, 33], and in integration with IoT platforms [7, 17, 34].

## 3 Transfer learning

According to LeCun et al. [30], when the pixels of an image are used as input to a classification task, conventional neural networks require an enormous number of parameters and ignore the topology of the input. In order to overcome these problems, LeCun et al. [30] proposed a new network topology called Convolutional Neural Network. This network is divided into two main parts. The first one is composed of convolutional and pooling layers, which are responsible for extracting features from the input image. The second part comprises the fully connected layers, which take the extracted features as input and classify the image into one of the labels presented during the CNN training.

In several image classification or detection challenges, like ImageNet [38], approaches that use CNNs frequently achieve excellent performance and outperform other methods. To achieve good results, CNNs usually need a large dataset for training, such as ImageNet. However, large-scale annotated datasets in the medical imaging domain are scarce [44]. Then, to be able to use the extraction capacity of CNNs without having a large number of annotated images, the Transfer Learning method can be applied.

Transfer Learning consists of transmitting knowledge acquired in a specific task to a different one. This technique is frequently used when there are not many images available for training, thus avoiding overfitting [15].

### 3.1 Feature extractors

A CNN can be turned into a feature extractor by applying the Transfer Learning approach. However, different CNN configurations can yield the most diverse results. Thus, different CNN architectures should be tested to find an adequate feature extractor for a specific task. In Table 1, we present the architectures used in this work and their respective highlights. We choose those CNNs due to the diversity in their structures and their outstanding results in the ImageNet Large Scale Visual Recognition Challenge.

## 4 Classifiers

In this section, we discuss the main points of each classifier used. In the proposed combinations, the following well-known classifiers are used: Naive Bayes, Multilayer Perceptron (MLP), k-Nearest Neighbor (KNN), Random Forest, and Support Vector Machines (SVM).

**Table 1** CNN architectures used as feature extractors and their respective highlights

| Architectures | Highlights |
|---|---|
| VGG [47] | Greater discriminative power due to increased CNN depth, made possible by the use of smaller filters |
| Inception [49] | Execution of multiple parallel convolutions in blocks called "inception modules", improving the network performance at a slight increase in computational cost |
| ResNet [19] | Use of "residual blocks", which learn residual functions from layers inputs, allowing higher accuracy as the network depth is increased |
| NASNet [60] | Construction of architectures that are learned directly from a dataset of interest |
| Xception [9] | Replacement of "inception modules" for "depthwise separable convolutions" |
| MobileNet [21] | Introduction of two hyperparameters that allow adaptation of the model size according to the restrictions of the problem |
| DenseNet [22] | Connection of each layer to every other layer in a feed-forward mode, strengthening feature propagation |

Naive Bayes Classifier is a supervised classifier based on the Bayes Decision Theory. This classification algorithm has class autonomy among the resources presented and thus performs the sample prediction based on the later probability acquired for each class. The highest probability achieved in class calculations corresponds to the prediction made by this classifier [50]. The posterior density calculation is carried out based on three factors: conditional density, a priori probabilities, and the function of the probability density. The most used probability density function in the literature corresponds to the Gaussian distribution, also known as the normal distribution. The highlight of this probability density function is the typical low computational cost [24].

MLP is the enhancement of the Perceptron classifier. Admittedly, the Perceptron classifier achieves satisfactory results only for linearly separable problems. The non-parametric MLP supervised classification algorithm corresponds to a solution to this issue. MLP consists of the structure formed by multiple layers, called artificial neural networks [18]. The operation of this classifier starts with the input of features to the initial layer. From this, the perceptron generates pulses of activation and propagate them through the posterior layers. The perceptron–perceptron connection gives the layer-by-layer transmission via a stipulated weight. Finally, the activation of the perceptrons in the network's output layer is determined by the activation function, which enables the classification of the sample.

KNN is a supervised and instantiated classification machine learning method. Its instantiated characteristic originates from the variable $k$, which corresponds to the number of closest neighbors to be counted in determining the class of a sample. The method is based on electing a new sample label based on the spatial distribution of the data set features. This distribution makes it possible to admit the $k$ closest neighbors to the new input. The similarity between the admitted samples is determined by calculating the distance between them [12].

Random Forest is the classification method most similar to the human way of classifying. Random Forest is considered an unsupervised machine learning algorithm. This method is based on a set of arbitrated conditions interrelated with

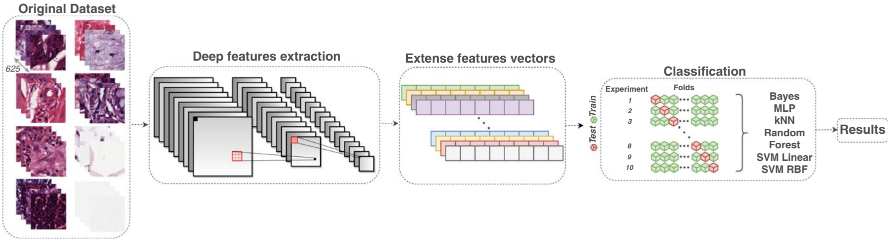


**Fig. 1** Flowchart of the Methodology implemented

attributes from the resources inserted in the classifier [6]. The adjustment of this classifier is performed by an ensemble approach called random Bagging. This algorithm is responsible for modeling the estimators, such as Decision Trees (DTs), in several random subsets of the training database, which ultimately produce results that, when combined, make up the final result of the Random Forest classifier.

SVM are classification methodologies based on the statistical learning of training data. SVM's main characteristic is the search for changes in the data visualization space to facilitate their respective classification. The way to change the data display space represents the set of calculations that define the kernel used in the SVM classifier [52]. For example, the linear and RBF kernels are sets of mathematical calculations used to divide the data with a simple hyperplane, based only on one line, and a set of mathematical calculations for generating new data from a distance between all the kernels are sets to a specific point-center, respectively.

## 5 Methodology

This section presents the proposed methodology, consisting of three steps: feature extraction, classification, and evaluation of the results. In the first step, we extract features from the original dataset through deep feature extraction, creating sub-datasets. Then, we use these sub-datasets as input for the classifiers. Finally, we evaluate the results and choose the best combination of feature extractor–classifier to classify colorectal cancer histology. An overview of the proposed approach is shown in Fig. 1.

### 5.1 Dataset

The dataset used in this approach corresponds to a collection of images from histopathological exams. The challenge is commonly called Colorectal Histology, and it provides a collection of textures in colorectal cancer histology, made publicly available on May 26, 2016 [26]. The images represent a collection of textures found in histological images of human colorectal cancer. The dataset contains 5000 histological images divided into eight different types of classes [25], which are listed below:

(A) Tumor epithelium

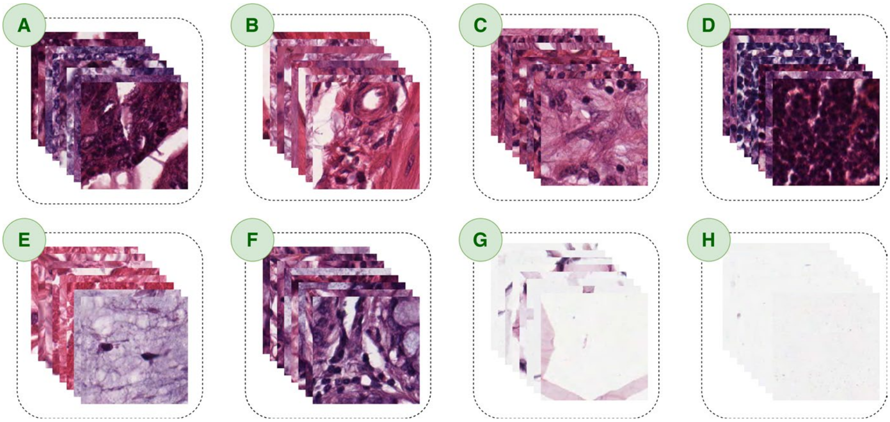


**Fig. 2** Samples of each class of the Colorectal Histology dataset. From left to right, sets of eight random images for the classes **a** tumor epithelium, **b** simple stroma, **c** complex stroma, **d** immune cells, **e** debris and mucus, **f** mucosal gland, **g** adipose tissue, and **h** background

(B) *Simple stroma* homogeneous composition of tumour stroma, extra-tumoural and smooth muscle
(C) *Complex stroma* composition of single tumour cells and immune cells
(D) *Immune cells* covers immune-cell conglomerates and sub-mucosal lymphoid follicles
(E) *Debris* covers regions with necrosis, mucus, and hemorrhage
(F) Normal mucosal glands
(G) Adipose tissue
(H) *Background* no tissue.

All images are evenly distributed across the classes, corresponding to 625 images of 150 × 150 pixels (74 × 74 μm) for each class. The images were obtained by an Aperio ScanScope, which has a capacity of 20× magnification. The images were made available by the Institute of Pathology at the University Medical Center Mannheim (Heidelberg University, Mannheim, Germany), which guarantees and protects the integrity of the exam/patient origin. All samples were collected and approved according to medical ethics board II of University Medical Center Mannheim (Heidelberg University, Germany) and by the Declaration of Helsinki [25].

In Fig. 2, we can notice the variance found in the same type of tissue and the similarity between samples of different classes. This specific characteristic justifies the complexity of the classification, which makes it susceptible to subjectivity and may result in disagreement among specialists.

### 5.2 Feature extraction

In this paper, we propose the use of CNNs to extract features of histological images of human colorectal cancer. We use a standard transfer learning approach to turn a CNN into a deep feature extractor [35, 36]. This approach is achieved by firstly training a CNN architecture in a large dataset, where it learns to extract critical features from the images. In our method, we use the pre-trained weights on the ImageNet dataset. Then, we remove the fully connected layers of the CNN, responsible for classifying the features, and we preserve the convolutional and pooling layers. As a result, the output of this adapted CNN becomes a set of features extracted from the input image.

As a pre-processing step, we perform a resize of the images to the input size of each CNN configuration. We perform this resize using the nearest-neighbor interpolation available in the OpenCV library, which does not have a significant computational cost. Then, we use the knowledge acquired by the adapted CNN to extract features from the pre-processed images of the desired dataset and build new sub-datasets. Since this CNN no longer has its fully connected layers, we can choose a machine learning method to classify the features.

Therefore, we preserve the main structure and parameters of the networks from the original papers. Maintaining these networks' characteristics allows the use of a robust feature extractor, initially used for a more complex challenge, to solve a minor task.

We use the following CNN configurations: VGG16 [47], VGG19 [47], InceptionV3 [49], InceptionResNetV2 [49], ResNet50 [19], ResNet50V2 [20], ResNet101 [19], ResNet101V2 [20], ResNet152 [19], ResNet152V2 [20], NASNetLarge [60], NASNetMobile [60], Xception [9], MobileNet [21], MobileNetV2 [39], DenseNet121, DenseNet169 and DenseNet201 [22]. Each CNN configuration generates a new sub-dataset.

Table 2 presents the different configurations, their base architectures, and the resulting number of features extracted. For instance, the configuration InceptionResNetV2 derives from both Inception and ResNet architectures.

### 5.3 Classification

In order to assign each image to one of the eight classes, we use five classifiers: Bayes, MLP, KNN, Random Forest, and SVM. In this step, each sub-dataset is composed of the features extracted by the CNNs.

We perform a hyperparameter search on the classifiers. We define a "search" range and utilize the grid search and random search methods since they are semi-automatic. We execute a grid search on the number of neighbors of the KNN classifier. While in MLP, Random Forest, and SVM, we perform a random search through 20-iterations and 5-fold cross-validation. Table 3 presents the values that we consider for the hyperparameters search. The number of estimators is the number of decision trees. Instead of the most commonly used stochastic gradient procedure, we use an optimization algorithm called Adam [29]. In the SVM classifier, the $C$ parameter adds a cost for each data point wrongfully classified; and the $\gamma$ parameter is used to control the sphere of influence of a single training point.

**Table 2** CNNs architectures, their evaluated configurations, and the number of features extracted from a single image by each configuration

| Architectures | Configurations | Number of features extracted |
|---|---|---|
| VGG [47] | VGG16 | 512 |
| | VGG19 | 512 |
| Inception [49] | InceptionV3 | 2048 |
| ResNet [19] | InceptionResNetV2 | 1536 |
| | ResNet50 | 2048 |
| | ResNet50V2 [20] | 2048 |
| | ResNet101 | 2048 |
| | ResNet101V2 [20] | 2048 |
| | ResNet152 | 2048 |
| | ResNet152V2 [20] | 2048 |
| NASNet [60] | NASNetLarge | 4032 |
| | NASNetMobile | 1056 |
| Xception [9] | Xception | 2048 |
| MobileNet [21] | MobileNet | 1024 |
| | MobileNetV2 [39] | 1280 |
| DenseNet [22] | DenseNet121 | 1024 |
| | DenseNet169 | 1664 |
| | DenseNet201 | 1920 |

**Table 3** Setup to search for hyperparameters of the classifiers

| Classifier | Search type | Parameter | Setup |
|---|---|---|---|
| Bayes | – | – | Gaussian probability Density Function |
| Random Forest | Random | Number of estimators | 50 to 3000 in steps of 50 |
| MLP | Random | Neurons in hidden layer | 2 to 1000 |
| | | Algorithm | Adam method [29] |
| kNN | Grid | Number of neighbors | 3, 5, 7, 9, 11, 13, 15 |
| SVM (linear kernel) | Random | C | $2^{-5}$ to $2^{15}$ |
| SVM (RBF kernel) | Random | C | $2^{-5}$ to $2^{15}$ |
| | | $\gamma$ | $2^{-15}$ to $2^{3}$ |

## 5.4 Evaluation metrics

The output of each combination, produced by the proposed approach, is evaluated using the confusion matrix tool and the metrics Accuracy and F1-Score. The confusion matrix corresponds to a tool for evaluating classification algorithms. The format of the confusion matrix allows quick and direct visualization of the results obtained for each class.

The confusion matrix is composed of values obtained from the database prediction. The values are divided among the classes of the dataset. These values are percentages of the number of samples correctly and incorrectly predicted for each class of the problem. The use of this tool makes it possible to relate the weaknesses of the classifier once the percentage that is wrongly predicted between each class is evident.

The Accuracy demonstrates the number of samples correctly predicted by the model in relation to the total number of samples. The F1-Score corresponds to the quantity that represents the harmonic average between Precision and Recall. The Precision and Sensitivity are calculated in the generation of the results. Since some works use the Error Rate as an evaluation metric, this metric is calculated only for comparison with other approaches in Sect. 6.1.

In Sect. 6, we mainly focus on Accuracy and F1-Score. A high Accuracy indicates satisfactory performance, and F1-score can show a balance between Recall and Precision of the combinations. Thus, a method with a high F1-Score tends to classify the regions of the tissues more evenly, meaning that the combination does not result in a significant number of false positives and false negatives.

Based on true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values, we calculate the metrics below by expressing the values of the confusion matrix. TP and FN are the samples from the desired class that are correctly and incorrectly classified, respectively. TN and FP are the samples that do not belong to the desired class and are correctly and incorrectly classified, respectively. Equations 1, 2, 3, 4, and 5 represent the Accuracy, Precision, Sensitivity, F1-Score, and Error Rate calculations.

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$\mathrm{Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$\mathrm{F1{-}score} = 2 \times \frac{\mathrm{Sensitivity} \times \mathrm{Precision}}{\mathrm{Sensitivity} + \mathrm{Precision}} \tag{4}$$

$$\mathrm{Error\,Rate} = 1 - \mathrm{Acc} \tag{5}$$

Besides the metrics mentioned above, we also analyze the training and testing times. The length of time that elapses from the beginning of the classifier training to the moment it is ready to classify is defined as training time. The testing time represents the time necessary to predict a class after the classifier receives the feature vector. Hence, training time is of paramount importance during the model configuration. After the configuration is completed, the testing time is more critical.

### 5.5 System configuration

The implementation setup and working environment are composed of an Ubuntu 16.04, with an Intel Xeon processor, 32GB of RAM, with no Graphical Processing Unit (GPU). Furthermore, we used the Python 3.6 programming language, *Keras* v2.3.1, *Scikit-Learn* library v0.21.0, and OpenCV v4.1.0.

## 6 Results

In this section, we present the results of the $18 \times 6 = 108$ experiments of supervised image classification, these being the combination of the 18 feature extractors and the six[1] classifiers evaluated. The results present a mean of a 10-fold-cross-validation. In each fold, we use 90% of the images for training and the rest of the images for testing.

Table 4 presents the Accuracy and F1-Score that were obtained through the extraction using the transfer learning technique and then classification. Table 4 also shows that combinations of DenseNet with SVM (RBF) have the two highest accuracies, in which DenseNet201 achieved an accuracy of 91.996%, and DenseNet169 reached 92.083%; these combinations are highlighted in blue.

Table 4 also exhibits that the proposed methodology achieved Accuracy and F1-Score of at least 70% when we disregard the Bayes classifier. In addition, we should highlight the extractors: DenseNet201, MobileNet, and VGG16. These feature extractors reached an accuracy and F1-score of at least 86%, regardless of the classifier. The SVM (RBF) can be highlighted among the classifiers since it achieved metrics of at least 79%. It was the classifier with the most outstanding metrics, independently of the feature extractor.

Another point that can be noted in Table 4 is that versions 2 (V2) of ResNet50, Resnet101, Resnet152 improved in the classification of the images of colorectal cancer histology. However, the improvement made on MobileNet did not affect the features extracted using the transfer learning technique.

Table 5 shows the training and testing times for the 108 experiments. We can observe that the Bayes classifier achieved the fastest training and testing times when combined with any feature extractors. This is expected since the Bayes classifier only demands calculating the probability of a feature set to be a specific class based on prior knowledge. The MLP classifier obtained the highest training times due to its number of neurons in the hidden layers. In Table 5, we can also observe that KNN achieved the slowest testing times for all feature extractors because of the high number of features extracted by each method.

DenseNet169, combined with SVM (RBF), reached a training time of 179 s and a testing time of 0.765 ms. Furthermore, DenseNet201, combined with SVM (RBF), achieved a training time of 265 s and a testing time of 0.948 ms.

[1] We considered the SVM classifier with different kernels as different classifiers.

**Table 4** Accuracy and F1-Score obtained by each combination of feature extractor and classifier

| Feature extractor | Classifier | Accuracy | F1-Score |
|---|---|---|---|
| ResNet101V2 | Bayes | 72.423 ± 0.893 | 72.683±0.817 |
| | MLP | 87.506 ± 0.989 | 87.548 ± 0.964 |
| | KNN | 80.244 ± 1.333 | 80.360 ± 1.354 |
| | Random Forest | 84.545 ± 1.741 | 84.620 ± 1.710 |
| | SVM Linear | 87.424 ± 0.993 | 87.467 ± 0.968 |
| | SVM RBF | 88.544 ± 1.230 | 88.558 ± 1.199 |
| DenseNet121 | Bayes | 79.244 ± 1.315 | 79.294 ± 1.305 |
| | MLP | 88.320 ± 1.149 | 88.323 ± 1.170 |
| | KNN | 83.952 ± 1.924 | 84.359 ± 1.866 |
| | Random Forest | 87.998 ± 0.929 | 87.995 ± 0.890 |
| | SVM Linear | 88.701 ± 0.847 | 88.744 ± 0.843 |
| | SVM RBF | 90.139 ± 1.325 | 90.171 ± 1.322 |
| **DenseNet201** | Bayes | 80.297 ± 0.998 | 80.339 ± 1.074 |
| | MLP | 89.718 ± 1.143 | 89.735 ± 1.140 |
| | KNN | 86.564 ± 1.492 | 86.809 ± 1.453 |
| | Random Forest | 89.462 ± 1.053 | 89.481 ± 1.061 |
| | SVM Linear | 90.659 ± 1.081 | 90.715 ± 1.061 |
| | **SVM RBF** | **91.996** ± **0.774** | **92.009** ± **0.776** |
| MobileNetV2 | Bayes | 75.858 ± 1.543 | 75.908 ± 1.590 |
| | MLP | 89.279 ± 1.647 | 89.287 ± 1.617 |
| | KNN | 86.124 ± 0.913 | 86.260 ± 0.889 |
| | Random Forest | 84.523 ± 1.541 | 84.529 ± 1.538 |
| | SVM Linear | 88.200 ± 1.768 | 88.219 ± 1.736 |
| | SVM RBF | 90.601 ± 1.589 | 90.612 ± 1.577 |
| ResNet50V2 | Bayes | 74.277 ± 1.070 | 74.602 ± 1.121 |
| | MLP | 88.622 ± 1.401 | 88.618 ± 1.406 |
| | KNN | 82.440 ± 1.076 | 82.454 ± 1.154 |
| | Random Forest | 84.659 ± 1.068 | 84.654 ± 1.120 |
| | SVM Linear | 87.836 ± 1.410 | 87.846 ± 1.435 |
| | SVM RBF | 89.416 ± 1.143 | 89.424 ± 1.147 |
| NASNetLarge | Bayes | 74.799 ± 0.892 | 75.137 ± 0.914 |
| | MLP | 88.504 ± 2.040 | 88.513 ± 2.045 |
| | KNN | 81.786 ± 1.500 | 81.939 ± 1.501 |
| | Random Forest | 84.986 ± 1.870 | 85.047 ± 1.922 |
| | SVM Linear | 87.763 ± 1.701 | 87.819 ± 1.705 |
| | SVM RBF | 89.044 ± 1.651 | 89.081 ± 1.664 |
| InceptionResNetV2 | Bayes | 76.478 ± 1.970 | 76.779 ± 1.839 |
| | MLP | 88.659 ± 1.207 | 88.698 ± 1.179 |
| | KNN | 82.119 ± 1.747 | 82.317 ± 1.723 |
| | Random Forest | 83.361 ± 1.450 | 83.373 ± 1.395 |
| | SVM Linear | 88.656 ± 1.557 | 88.750 ± 1.526 |
| | SVM RBF | 88.657 ± 1.648 | 88.745 ± 1.605 |

**Table 4** (continued)

| Feature extractor | Classifier | Accuracy | F1-Score |
|---|---|---|---|
| **DenseNet169** | Bayes | 80.427±1.931 | 80.583 ± 1.966 |
| | MLP | 90.947 ± 1.365 | 90.978 ± 1.354 |
| | KNN | 84.983 ± 0.631 | 85.348 ± 0.585 |
| | Random Forest | 88.842 ± 1.492 | 88.877 ± 1.469 |
| | SVM Linear | 90.644 ± 0.863 | 90.680 ± 0.849 |
| | **SVM RBF** | **92.083 ± 1.029** | **92.117 ± 1.013** |
| MobileNet | Bayes | 80.548 ± 2.392 | 80.621±2.308 |
| | MLP | 90.525 ± 1.904 | 90.540 ± 1.912 |
| | KNN | 87.545 ± 1.617 | 87.741 ± 1.607 |
| | Random Forest | 87.104 ± 2.205 | 87.178 ± 2.192 |
| | SVM Linear | 90.388 ± 1.852 | 90.400 ± 1.845 |
| | SVM RBF | 91.725 ± 1.695 | 91.756 ± 1.716 |
| VGG16 | Bayes | 77.919 ± 1.296 | 78.048 ± 1.222 |
| | MLP | 88.297 ± 1.325 | 88.297 ± 1.337 |
| | KNN | 86.719 ± 1.216 | 86.700 ± 1.234 |
| | Random Forest | 87.316 ± 1.485 | 87.300 ± 1.485 |
| | SVM Linear | 88.397 ± 1.150 | 88.434 ± 1.154 |
| | SVM RBF | 90.796 ± 1.135 | 90.820 ± 1.145 |
| VGG19 | Bayes | 75.623 ± 1.521 | 75.744 ± 1.540 |
| | MLP | 87.107 ± 1.198 | 87.131 ± 1.195 |
| | KNN | 85.483 ± 1.601 | 85.441 ± 1.598 |
| | Random Forest | 87.001 ± 0.833 | 87.033 ± 0.807 |
| | SVM Linear | 88.200 ± 1.065 | 88.260 ± 1.074 |
| | SVM RBF | 89.303 ± 1.049 | 89.328 ± 1.056 |
| ResNet50 | Bayes | 58.950 ± 2.973 | 57.864 ± 3.153 |
| | MLP | 78.138 ± 1.847 | 78.131 ± 1.913 |
| | KNN | 70.076 ± 2.084 | 70.031 ± 2.084 |
| | Random Forest | 73.671 ± 2.528 | 73.242 ± 2.518 |
| | SVM Linear | 78.898 ± 1.736 | 78.949 ± 1.714 |
| | SVM RBF | 79.580 ± 2.172 | 79.560 ± 2.258 |
| InceptionV3 | Bayes | 77.100 ± 1.133 | 77.353 ± 1.097 |
| | MLP | 88.243 ± 1.751 | 88.244 ± 1.755 |
| | KNN | 83.022 ± 1.657 | 83.275 ± 1.609 |
| | Random Forest | 84.161 ± 1.486 | 84.173 ± 1.523 |
| | SVM Linear | 88.042 ± 1.577 | 88.077 ± 1.570 |
| | SVM RBF | 88.462 ± 1.494 | 88.485 ± 1.517 |

**Table 4** (continued)

| Feature extractor | Classifier | Accuracy | F1-Score |
|---|---|---|---|
| ResNet152 | Bayes | 58.916 ± 1.856 | 57.803 ± 1.974 |
| | MLP | 77.161 ± 1.592 | 77.123 ± 1.582 |
| | KNN | 70.219 ± 1.506 | 70.322 ± 1.408 |
| | Random Forest | 74.818 ± 1.308 | 74.368 ± 1.222 |
| | SVM Linear | 78.040 ± 1.376 | 77.991 ± 1.414 |
| | SVM RBF | 80.004 ± 1.307 | 79.960 ± 1.318 |
| ResNet101 | Bayes | 60.496 ± 1.128 | 60.380 ± 1.180 |
| | MLP | 77.453 ± 1.436 | 77.444 ± 1.372 |
| | KNN | 70.516 ± 1.445 | 70.465 ± 1.427 |
| | Random Forest | 73.833 ± 1.505 | 73.572 ± 1.460 |
| | SVM Linear | 78.373 ± 1.800 | 78.408 ± 1.784 |
| | SVM RBF | 79.275 ± 1.728 | 79.297 ± 1.696 |
| ResNet152V2 | Bayes | 77.463 ± 1.438 | 77.852±1.525 |
| | MLP | 88.961 ± 1.673 | 88.966 ± 1.695 |
| | KNN | 79.556 ± 1.539 | 79.653 ± 1.563 |
| | Random Forest | 85.322 ± 1.633 | 85.341 ± 1.657 |
| | SVM Linear | 88.664 ± 1.526 | 88.703 ± 1.511 |
| | SVM RBF | 88.981 ± 1.527 | 88.994 ± 1.511 |
| Xception | Bayes | 78.482 ± 1.526 | 78.503±1.564 |
| | MLP | 91.082 ± 1.035 | 91.080 ± 1.032 |
| | KNN | 85.782 ± 1.782 | 86.071 ± 1.680 |
| | Random Forest | 87.941 ± 1.227 | 87.943 ± 1.214 |
| | SVM Linear | 90.420 ± 1.283 | 90.441 ± 1.258 |
| | SVM RBF | 91.441 ± 1.199 | 91.462 ± 1.182 |
| NASNetMobile | Bayes | 78.063 ± 2.349 | 78.187 ± 2.276 |
| | MLP | 87.965 ± 1.738 | 87.950 ± 1.742 |
| | KNN | 82.046 ± 1.809 | 82.264 ± 1.731 |
| | Random Forest | 85.500 ± 1.910 | 85.505 ± 1.874 |
| | SVM Linear | 88.264 ± 1.926 | 88.315 ± 1.882 |
| | SVM RBF | 89.263 ± 1.704 | 89.293 ± 1.675 |

Important results are shown in bold

Therefore, the testing time of both feature extractors combined with the classifier is suitable for real-time applications.

Figure 3 presents a summary of the best combinations of feature extractors and classifiers, which are DenseNet201 combined with SVM (RBF) and DenseNet169 combined with SVM (RBF). It is noticeable that the combination with DenseNet169 is slightly better in Accuracy and F1-Score, and its testing

**Table 5** Training and Testing times obtained by each combination of feature extractor and classifier

| Feature extractor | Classifier | Training time (s) | Testing time (ms) |
|---|---|---|---|
| ResNet101V2 | Bayes | 1.256104 ± 0.667072 | 0.005079 ± 0.002390 |
| | MLP | 4471.702107 ± 238.808700 | 0.108807 ± 0.069536 |
| | KNN | 2.757085 ± 1.173433 | 2.037132 ± 0.045359 |
| | Random Forest | 737.463577 ± 37.933863 | 0.849688 ± 0.015398 |
| | SVM Linear | 153.309223 ± 47.390752 | 0.943807 ± 0.278556 |
| | SVM RBF | 376.686515 ± 88.095863 | 1.548771 ± 0.324230 |
| DenseNet121 | Bayes | 0.492448 ± 0.290841 | 0.004071 ± 0.001845 |
| | MLP | 1887.720808 ± 61.145177 | 0.013205 ± 0.006193 |
| | KNN | 1.866221 ± 1.475511 | 1.724742 ± 0.061296 |
| | Random Forest | 474.442361 ± 11.914351 | 0.450649 ± 0.009261 |
| | SVM Linear | 125.490908 ± 19.927303 | 0.772196 ± 0.134570 |
| | SVM RBF | 208.406669 ± 42.927784 | 0.949253 ± 0.177107 |
| DenseNet201 | Bayes | 2.904449 ± 1.722331 | 0.006207 ± 0.003063 |
| | MLP | 557.256617 ± 88.912056 | 0.004126 ± 0.002062 |
| | KNN | 0.958944 ± 0.415036 | 1.788717 ± 0.098374 |
| | Random Forest | 408.918562 ± 13.394929 | 0.231247 ± 0.009132 |
| | SVM Linear | 129.101491 ± 33.952520 | 0.717847±0.182264 |
| | SVM RBF | 265.562540 ± 23.886190 | 0.948397±0.083268 |
| MobileNetV2 | Bayes | 0.500899 ± 0.116833 | 0.003779 ± 0.001563 |
| | MLP | 1187.066163 ± 109.525459 | 0.045051 ± 0.022066 |
| | KNN | 1.391544 ± 0.944361 | 1.546576 ± 0.047736 |
| | Random Forest | 602.477331 ± 24.422236 | 0.328559 ± 0.020104 |
| | SVM Linear | 98.248825 ± 6.716835 | 0.546791 ± 0.023185 |
| | SVM RBF | 239.725331 ± 40.713824 | 0.780648 ± 0.066941 |
| ResNet50V2 | Bayes | 0.651892 ± 0.329824 | 0.004223 ± 0.002002 |
| | MLP | 1581.108403 ± 128.437121 | 0.024782 ± 0.010695 |
| | KNN | 1.408597 ± 0.260703 | 1.638870 ± 0.042978 |
| | Random Forest | 247.085370 ± 6.520427 | 0.179283 ± 0.006390 |
| | SVM Linear | 132.334184 ± 29.790369 | 0.817595 ± 0.090858 |
| | SVM RBF | 259.309113 ± 22.694628 | 1.103358 ± 0.226573 |
| NASNetLarge | Bayes | 1.394583 ± 0.524192 | 0.006003 ± 0.002115 |
| | MLP | 2323.706979 ± 282.362831 | 0.048789 ± 0.018099 |
| | KNN | 2.277660 ± 0.429664 | 3.430507 ± 0.181021 |
| | Random Forest | 806.373208 ± 27.908470 | 0.385989 ± 0.030136 |
| | SVM Linear | 284.737513 ± 31.351743 | 1.583620 ± 0.162267 |
| | SVM RBF | 412.473736 ± 6.816471 | 2.417776 ± 0.577686 |
| InceptionResNetV2 | Bayes | 1.273082 ± 0.467151 | 0.005708 ± 0.002597 |
| | MLP | 676.657151 ± 97.079157 | 0.004195 ± 0.002378 |
| | KNN | 1.511736 ± 1.447230 | 1.393645 ± 0.044317 |
| | Random Forest | 748.135841 ± 30.579288 | 0.269957 ± 0.011832 |
| | SVM Linear | 106.486707 ± 22.140289 | 0.593405 ± 0.051671 |
| | SVM RBF | 119.673637 ± 18.855987 | 0.635505 ± 0.054499 |

**Table 5** (continued)

| Feature extractor | Classifier | Training time (s) | Testing time (ms) |
|---|---|---|---|
| DenseNet169 | Bayes | 0.858342 ± 0.541357 | 0.006008 ± 0.001624 |
| | MLP | 1680.285246 ± 109.458078 | 0.014161 ± 0.006900 |
| | KNN | 2.201647 ± 2.210836 | 1.937445 ± 0.083650 |
| | Random Forest | 1432.010970 ± 61.574034 | 0.788451 ± 0.022768 |
| | SVM Linear | 213.596518 ± 27.513107 | 1.027755 ± 0.014714 |
| | SVM RBF | 179.115986 ± 40.839740 | 0.764738 ± 0.106099 |
| MobileNet | Bayes | 0.546731 ± 0.500385 | 0.002953 ± 0.001838 |
| | MLP | 546.496072 ± 67.986574 | 0.015247 ± 0.003865 |
| | KNN | 2.142582 ± 1.968038 | 1.023620 ± 0.139765 |
| | Random Forest | 438.926362 ± 12.416899 | 0.449182 ± 0.023301 |
| | SVM Linear | 70.660073 ± 15.417306 | 0.464747 ± 0.118563 |
| | SVM RBF | 130.809587 ± 12.778148 | 0.614404 ± 0.183073 |
| VGG16 | Bayes | 0.144882 ± 0.052616 | 0.004564 ± 0.001830 |
| | MLP | 1729.524620 ± 107.380049 | 0.009009 ± 0.004564 |
| | KNN | 0.350058 ± 0.155364 | 0.529276 ± 0.027895 |
| | Random Forest | 877.775962 ± 35.005706 | 0.794564 ± 0.017710 |
| | SVM Linear | 47.368195 ± 15.644190 | 0.236372 ± 0.060112 |
| | SVM RBF | 64.628976 ± 2.776642 | 0.379487 ± 0.029866 |
| VGG19 | Bayes | 0.212815 ± 0.075386 | 0.002374 ± 0.001683 |
| | MLP | 1200.641323 ± 78.660175 | 0.009521 ± 0.005240 |
| | KNN | 0.269711 ± 0.148897 | 0.498642 ± 0.091794 |
| | Random Forest | 110.925332 ± 3.459166 | 0.393141 ± 0.040268 |
| | SVM Linear | 26.258812 ± 4.735973 | 0.187476 ± 0.036375 |
| | SVM RBF | 41.094585 ± 14.761574 | 0.221090±0.056550 |
| ResNet50 | Bayes | 1.366463 ± 0.906024 | 0.007493 ± 0.002562 |
| | MLP | 2548.602765 ± 235.377757 | 0.017675 ± 0.009857 |
| | KNN | 2.255636 ± 0.577269 | 1.614571 ± 0.131339 |
| | Random Forest | 597.865272 ± 28.408460 | 0.551806 ± 0.018161 |
| | SVM Linear | 230.778445 ± 69.345011 | 1.246999 ± 0.371225 |
| | SVM RBF | 316.183878 ± 104.676711 | 1.576447 ± 0.442080 |
| InceptionV3 | Bayes | 0.924846 ± 0.322379 | 0.004046 ± 0.001171 |
| | MLP | 3056.391673 ± 294.808432 | 0.046383 ± 0.026891 |
| | KNN | 2.667241 ± 2.347622 | 2.052711 ± 0.056952 |
| | Random Forest | 1687.597282 ± 69.116237 | 0.487001 ± 0.045884 |
| | SVM Linear | 176.472038 ± 36.993404 | 1.127923 ± 0.094545 |
| | SVM RBF | 353.725177 ± 57.100580 | 1.387665 ± 0.213145 |

**Table 5** (continued)

| Feature extractor | Classifier | Training time (s) | Testing time (ms) |
|---|---|---|---|
| ResNet152 | Bayes | 0.636714 ± 0.286117 | 0.004254 ± 0.001691 |
| | MLP | 6981.808846 ± 226.768505 | 0.040989 ± 0.039438 |
| | KNN | 1.946031 ± 0.143307 | 1.869638 ± 0.040302 |
| | Random Forest | 355.110181 ± 15.749700 | 0.355637 ± 0.010677 |
| | SVM Linear | 265.155542 ± 59.374448 | 1.236719 ± 0.234938 |
| | SVM RBF | 280.955045 ± 28.556761 | 1.147347 ± 0.102684 |
| ResNet101 | Bayes | 1.146787 ± 0.419369 | 0.005764 ± 0.001700 |
| | MLP | 4074.093161 ± 155.083086 | 0.023899 ± 0.012961 |
| | KNN | 2.597779 ± 0.108729 | 2.208008 ± 0.049697 |
| | Random Forest | 96.973143 ± 5.019693 | 0.323434 ± 0.007636 |
| | SVM Linear | 257.035909 ± 64.325011 | 1.342147 ± 0.215593 |
| | SVM RBF | 358.611006 ± 7.339549 | 1.104626 ± 0.021773 |
| ResNet152V2 | Bayes | 0.729931 ± 0.278260 | 0.003357 ± 0.000932 |
| | MLP | 1931.446265 ± 251.967927 | 0.042560 ± 0.020959 |
| | KNN | 3.373562 ± 0.973163 | 2.484569 ± 0.247448 |
| | Random Forest | 1122.204345 ± 49.762525 | 0.815479 ± 0.024736 |
| | SVM Linear | 209.961904 ± 47.978272 | 1.218995 ± 0.253129 |
| | SVM RBF | 297.001313 ± 85.921826 | 1.983845 ± 0.070923 |
| Xception | Bayes | 1.118301 ± 0.717466 | 0.005578 ± 0.003523 |
| | MLP | 3357.911573 ± 259.828742 | 0.106830 ± 0.043945 |
| | KNN | 2.582450 ± 1.055500 | 2.001393±0.083567 |
| | Random Forest | 1127.323146 ± 31.820432 | 0.609433 ± 0.022870 |
| | SVM Linear | 138.670975 ± 2.668562 | 0.853885 ± 0.084326 |
| | SVM RBF | 167.276784 ± 33.990876 | 1.164898 ± 0.356653 |
| NASNetMobile | Bayes | 0.512117 ± 0.345216 | 0.004085 ± 0.002113 |
| | MLP | 1199.904453±158.429208 | 0.010047 ± 0.005943 |
| | KNN | 0.358513 ± 0.020915 | 1.058878 ± 0.034788 |
| | Random Forest | 143.013124 ± 4.656400 | 0.332898 ± 0.018342 |
| | SVM Linear | 69.613175 ± 4.740637 | 0.390025 ± 0.031622 |
| | SVM RBF | 210.921591 ± 37.106153 | 0.576849 ± 0.042696 |

time is also lower than the one of DenseNet201. The standard deviations appear to be large only due to the limits of the *y*-axis.

Analyzing the confusion matrices on Fig. 5 for DenseNet169 - SVM (RBF) and Fig. 4 for DenseNet201 - SVM (RBF), we notice that these combinations clearly distinguish classes 0, 3, 4, 5, 6, and 7, which represent tumor epithelium, immune cells, debris, mucosal gland, adipose tissue, and background, respectively.The confusion mostly occurs in classes 1 and 2, which are the stroma and complex stroma classes.

This confusion occurs since the simple stroma (class 1) and the complex stroma (class 2) can overlap. Besides, complex stroma presents a tendency to cluster

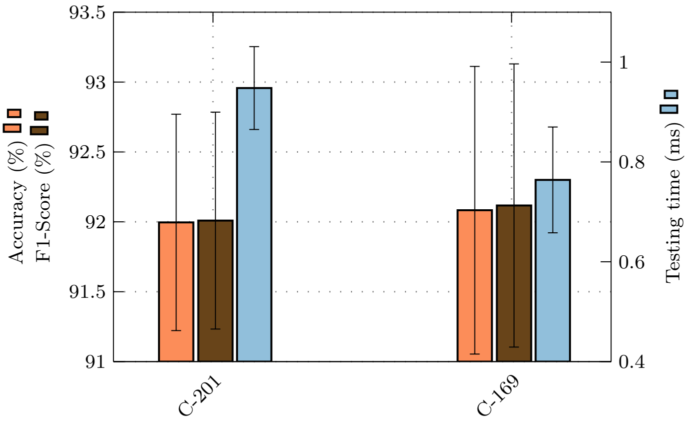
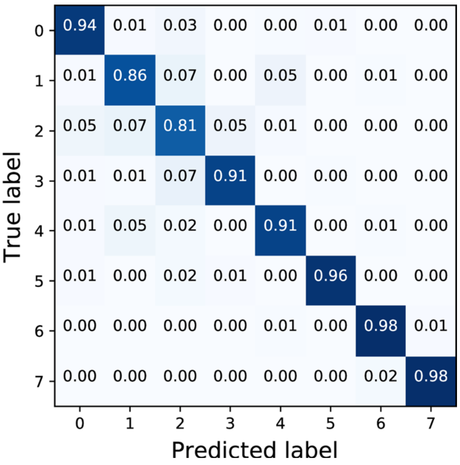


**Fig. 3** Accuracy, F1-Score and testing time for the best combinations of feature extractor with classifier. (C-201: DenseNet201 + SVM(RBF), C-169: DenseNet169 + SVM(RBF)

**Fig. 4** Confusion Matrix of DenseNet201 combined with SVM RBF. Class 0—tumor epithelium, Class 1—simple stroma, Class 2—complex stroma, Class 3—immune cells, Class 4—debris and mucus, Class 5—mucosal gland, Class 6—adipose tissue, and Class 7—background

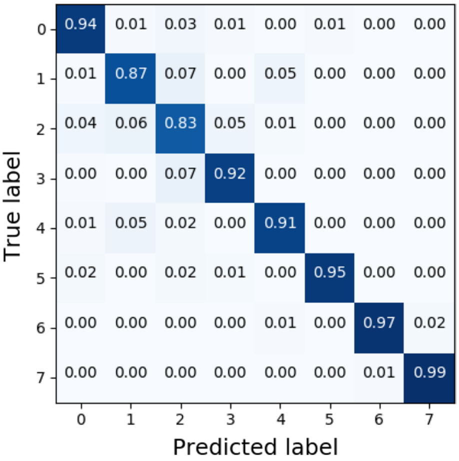


| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.94 | 0.01 | 0.03 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 |
| 1 | 0.01 | 0.86 | 0.07 | 0.00 | 0.05 | 0.00 | 0.01 | 0.00 |
| 2 | 0.05 | 0.07 | 0.81 | 0.05 | 0.01 | 0.00 | 0.00 | 0.00 |
| 3 | 0.01 | 0.01 | 0.07 | 0.91 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.05 | 0.02 | 0.00 | 0.91 | 0.00 | 0.01 | 0.00 |
| 5 | 0.01 | 0.00 | 0.02 | 0.01 | 0.00 | 0.96 | 0.00 | 0.00 |
| 6 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.98 | 0.01 |
| 7 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.98 |

around the tumor epithelium (class 3). Hence, the Accuracy and F1-score for the DenseNet201 and SVM (RBF) combination are slightly inferior to these same metrics for the DenseNet169 and SVM (RBF) combination. The DenseNet169 and SVM (RBF) combination stands out due to its ability to better classify the classes simple stroma (class 1) and complex stroma (class 2).

### 6.1 Comparison between methods in the literature

This section compares the best combination achieved by the proposed approach with methods published in the literature. Table 6 presents a comparison of our approach with studies that use the same dataset as presented in Sect. 5.1.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.94 | 0.01 | 0.03 | 0.01 | 0.00 | 0.01 | 0.00 | 0.00 |
| 1 | 0.01 | 0.87 | 0.07 | 0.00 | 0.05 | 0.00 | 0.00 | 0.00 |
| 2 | 0.04 | 0.06 | 0.83 | 0.05 | 0.01 | 0.00 | 0.00 | 0.00 |
| 3 | 0.00 | 0.00 | 0.07 | 0.92 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.05 | 0.02 | 0.00 | 0.91 | 0.00 | 0.00 | 0.00 |
| 5 | 0.02 | 0.00 | 0.02 | 0.01 | 0.00 | 0.95 | 0.00 | 0.00 |
| 6 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.97 | 0.02 |
| 7 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.99 |

**Fig. 5** Confusion Matrix of DenseNet 169 combined with SVM RBF. Class 0—tumor epithelium, Class 1—simple stroma, Class 2—complex stroma, Class 3—immune cells, Class 4—debris and mucus, Class 5—mucosal gland, Class 6—adipose tissue, and Class 7—background

Table 6 shows that the approach proposed by Wang et al. [53] reached an accuracy 0.52% higher than the approach proposed by this paper. However, they only selected 1000 non-overlapping images. This selection might have influenced the result since images that are difficult to classify may have been removed; for instance, the authors may have removed the most difficult images from the simple stroma and complex stroma classes, which may overlap.

Rachapudi and Devi [37] did not present the overall accuracy; they performed an analysis by class and presented an error rate (ER) of 22.7%. Our approach reached an ER of 7.9%, demonstrating that it had a better performance than the CNN proposed by Rachapudi and Devi [37].

The method proposed by Kather et al. [25] is the most comparable method to the proposed approach. Both approaches performed extensive feature extraction, used different machine learning techniques, and validated the results through a 10-fold-cross-validation. However, Kather et al. [25] applied feature extractors based on texture and made combinations of the features sets. In Table 6, we can observe that the proposed approach achieved an Accuracy of 4.68% higher than the approach proposed by Kather et al. [25]. Therefore, even though Kather et al. [25] combined the feature sets, CNNs can extract more relevant features from the images.

## 7 Conclusions

This paper presents a method to classify tissue images of colorectal cancer histology in eight classes. The approach is divided into two steps: (i) feature extraction using CNNs through the transfer learning technique, and (ii) classification using machine learning methods.

We performed an extensive combination of feature extractors and classifiers, totaling 108 experiments, using the transfer learning method. The results show that CNNs, combined with the transfer learning technique and machine learning methods, can be used as feature extractors for this problem. The feature extractor

**Table 6** Comparison of the results obtained by the proposed method and other approaches

| Author | Number of images | Approach | Method | Results |
|---|---|---|---|---|
| Our approach | 5000 | Feature extraction using CNNs through transfer learning, and then, classification with machine learning methods | DenseNet169 + SVM (RBF) | Acc: 92.08%<br>F1-Score: 92.12%<br>Error rate: 7.9% |
| Wang et al. [53] | 1000 | Decomposed the H&E components, and then, applied a BCNN-based to classify the decomposed images | Decomposition + BCNN | Acc: 92.6% |
| Rachapudi and Devi [37] | 5000 | A new CNN architecture | CNN with five blocks of layers | Error rate: 22.7% |
| Kather et al. [25] | 5000 | Feature extraction combining traditional feature extraction methods, and then, classification with machine learning methods | Combination of feature extraction techniques + SVM (RBF) | Acc: 87.4%<br>Error Rate: 12.6% |

DenseNet169 combined with SVM (RBF) produced the best performance, obtaining 92.08% in Accuracy, 92.117% in F1-Score, and 0.76 ms in the testing time, meaning low computational costs.

Therefore, our method has shown efficiency in classifying CRC histopathological images with high accuracy and speed. Thus, it can be a beneficial tool to help doctors and technicians to identify the correct type of tissue in this exam. For this reason, cancerous samples are less likely to pass unnoticed, and patients will get proper and early treatments more often.

For future works, we intend to expand the application of automatic tissue classification in histopathological exams. Initially, we intend to test the proposed method in different histology datasets. Then, we plan to combine different CNNs and even traditional methods for feature extraction, aiming to improve the metrics. Furthermore, we aim to create an Internet of Things system for this approach to be accessible in developing countries, where we find the highest mortality rates. Finally, we intend to analyze the feasibility of the system with the PACS system.

**Acknowledgements** The authors are grateful to the Deanship of Scientific Research at King Saud University for funding this work through Vice Deanship of Scientific Research Chairs: Chair of Pervasive and Mobile Computing. This study was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Also, Pedro Pedrosa Rebouças Filho acknowledges the sponsorship from the Brazilian National Council for Research and Development (CNPq) via Grants Nos. 431709/2018-1 and 311973/2018-3.

## References

1. Abir F, Alva S, Longo WE, Audiso R, Virgo KS, Johnson FE (2006) The postoperative surveillance of patients with colon cancer and rectal cancer. Am J Surg 192(1):100–108. https://doi.org/10.1016/j.amjsurg.2006.01.053
2. Alhindi TJ, Kalra S, Ng KH, Afrin A, Tizhoosh HR (2018) Comparing LBP, HOG and deep features for classification of histopathology images. In: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–7, https://doi.org/10.1109/IJCNN.2018.8489329
3. Araújo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C, Polónia A, Campilho A (2017) Classification of breast cancer histology images using convolutional neural networks. PLoS ONE 12(6):e0177544. https://doi.org/10.1371/journal.pone.0177544
4. Arnold M, Sierra MS, Laversanne M, Soerjomataram I, Jemal A, Bray F (2017) Global patterns and trends in colorectal cancer incidence and mortality. Gut 66(4):683–691. https://doi.org/10.1136/gutjnl-2015-310912
5. Bayramoglu N, Heikkilä J (2016) Transfer learning for cell nuclei classification in histopathology images. In: European Conference on Computer Vision, Springer, pp 532–539
6. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
7. Chatterjee R, Maitra T, Islam SH, Hassan MM, Alamri A, Fortino G (2019) A novel machine learning based feature selection for motor imagery EEG signal classification in internet of medical things environment. Future Gener Comput Syst 98:419–434. https://doi.org/10.1016/j.future.2019.01.048
8. Choi JY, Yoo TK, Seo JG, Kwak J, Um TT, Rim TH (2017) Multi-categorical deep learning neural network to classify retinal images: a pilot study employing small database. PLoS ONE. https://doi.org/10.1371/journal.pone.0187336
9. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1251–1258. https://doi.org/10.1109/CVPR.2017.195
10. Coudray N, Ocampo PS, Sakellaropoulos T, Narula N, Snuderl M, Fenyö D, Moreira AL, Razavian N, Tsirigos A (2018) Classification and mutation prediction from non-small cell lung cancer

histopathology images using deep learning. Nat Med 24(10):1559–1567. https://doi.org/10.1038/s41591-018-0177-5

11. da Nóbrega RVM, Filho PPR, Rodrigues MB, da Silva SPP, Júnior CMJMD, de Albuquerque VHC (2018) Lung nodule malignancy classification in chest computed tomography images using transfer learning and convolutional neural networks. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3895-1
12. Fukunaga K, Narendra PM (1975) A branch and bound algorithm for computing k-nearest neighbors. IEEE Trans Comput 24(7):750–753. https://doi.org/10.1109/T-C.1975.224297
13. Gecer B, Aksoy S, Mercan E, Shapiro LG, Weaver DL, Elmore JG (2018) Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks. Pattern Recogn 84:345–356. https://doi.org/10.1016/j.patcog.2018.07.022
14. Girard L, Rodriguez-Canales J, Behrens C, Thompson DM, Botros IW, Tang H, Xie Y, Rekhtman N, Travis WD, Wistuba II et al (2016) An expression signature as an aid to the histologic classification of non-small cell lung cancer. Clin Cancer Res 22(19):4880–4889. https://doi.org/10.1158/1078-0432
15. Han D, Liu Q, Fan W (2018) A new image classification method using CNN transfer learning and web data augmentation. Expert Syst Appl 95:43–56. https://doi.org/10.1016/j.eswa.2017.11.028
16. Han T, Nunes VX, De Freitas Souza LF, Marques AG, Silva ICL, Junior MAAF, Sun J, Filho PPR (2020) Internet of medical things-based on deep learning techniques for segmentation of lung and stroke regions in CT scans. IEEE Access 8:71117–71135. https://doi.org/10.1109/ACCESS.2020.2987932
17. Hassan MM, Ullah S, Hossain MS, Alelaiwi A (2020) An end-to-end deep learning model for human activity recognition from highly sparse body sensor data in internet of medical things environment. J Supercomput. https://doi.org/10.1007/s11227-020-03361-4
18. Haykin S (2009) Neural networks and learning machines, vol 3. Pearson Upper Saddle River, NJ, USA
19. He K, Zhang X, Ren S, Sun J (2016a) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
20. He K, Zhang X, Ren S, Sun J (2016b) Identity mappings in deep residual networks. In: European Conference on Computer Vision, Springer, pp 630–645
21. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861
22. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4700–4708. https://doi.org/10.1109/CVPR.2017.243
23. Hussein S, Cao K, Song Q, Bagci U (2017) Risk stratification of lung nodules using 3d CNN-based multi-task learning. Lecture Notes in Computer Science. Springer, Cham, pp 249–260
24. Jr Dourado CM, da Silva SP, da Nóbrega RV, Barros AC, Reboucas Filho PP, de Albuquerque VHC (2019) Deep learning IoT system for online stroke detection in skull computed tomography images. Comput Netw 152:25–39. https://doi.org/10.1016/j.comnet.2019.01.019
25. Kather J, Weis CA, Bianconi F, Melchers S, Schad L, Gaiser T, Marx A, Zöllner F (2016a) Multi-class texture analysis in colorectal cancer histology. Sci Rep 6:27988. https://doi.org/10.1038/srep27988
26. Kather JN, Zöllner FG, Bianconi F, Melchers SM, Schad LR, Gaiser T, Marx A, Weis CA (2016b) Collection of textures in colorectal cancer histology. https://doi.org/10.5281/ZENODO.53169
27. Kather JN, Krisam J, Charoentong P, Luedde T, Herpel E, Weis CA, Gaiser T, Marx A, Valous NA, Ferber D et al (2019) Predicting survival from colorectal cancer histology slides using deep learning: a retrospective multicenter study. PLoS Med. https://doi.org/10.1371/journal.pmed.1002730
28. Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan F, Dong J, Prasadha MK, Pei J, Ting MY, Zhu J, Li C, Hewett S, Dong J, Ziyar I, Shi A, Zhang R, Zheng L, Hou R, Shi W, Fu X, Duan Y, Huu VA, Wen C, Zhang ED, Zhang CL, Li O, Wang X, Singer MA, Sun X, Xu J, Tafreshi A, Lewis MA, Xia H, Zhang K (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122–1131.e9. https://doi.org/10.1016/j.cell.2018.02.010
29. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:14126980

30. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
31. Loh BCS, Then PHH (2017) Deep learning for cardiac computer-aided diagnosis: benefits, issues & solutions. mHealth 3:45. https://doi.org/10.21037/mhealth.2017.09.01
32. Marques JAL, Cortez PC, Madeiro JPDV, Fong SJ, Schlindwein FS, De Albuquerque VHC (2019) Automatic cardiotocography diagnostic system based on hilbert transform and adaptive threshold technique. IEEE Access 7:73085–73094
33. Marques JAL, Cortez PC, Madeiro JP, de Albuquerque VHC, Fong SJ, Schlindwein FS (2020) Nonlinear characterization and complexity analysis of cardiotocographic examinations using entropy measures. J Supercomput 76(2):1305–1320
34. Olivares R, Munoz R, Soto R, Crawford B, Cárdenas D, Ponce A, Taramasco C (2020) An optimized brain-based algorithm for classifying parkinson's disease. Appl Sci 10(5):1827. https://doi.org/10.3390/app10051827
35. Orenstein EC, Beijbom O (2017) Transfer learning and deep feature extraction for planktonic image data sets. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp 1082–1088. https://doi.org/10.1109/WACV.2017.125
36. Paul R, Hawkins SH, Balagurunathan Y, Schabath MB, Gillies RJ, Hall LO, Goldgof DB (2016) Deep feature transfer learning in combination with traditional features predicts survival among patients with lung adenocarcinoma. Tomography 2(4):388. https://doi.org/10.18383/j.tom.2016.00211
37. Rachapudi V, Devi GL (2020) Improved convolutional neural network based histopathological image classification. Evolut Intell 284:1–7
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
39. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4510–4520
40. Santos MA, Munoz R, Olivares R, Rebouças Filho PP, Del Ser J, de Albuquerque VHC (2020) Online heart monitoring systems on the internet of health things environments: a survey, a reference model and an outlook. Inf Fusion 53:222–239. https://doi.org/10.1016/j.inffus.2019.06.004
41. Sarmento RM, Vasconcelos FFX, Rebouças Filho PP, Wu W, De Albuquerque VHC (2019) Automatic neuroimage processing and analysis in stroke—a systematic review. IEEE Rev Biomed Eng 13:130–155
42. Sarmento RM, Vasconcelos FF, Filho PPR, de Albuquerque VHC (2020) An IoT platform for the analysis of brain CT images based on parzen analysis. Future Gener Comput Syst 105:135–147. https://doi.org/10.1016/j.future.2019.11.033
43. Shaban M, Awan R, Fraz MM, Azam A, Tsang YW, Snead D, Rajpoot NM (2020) Context-aware convolutional neural network for grading of colorectal cancer histology images. IEEE Trans Med Imaging. https://doi.org/10.1109/tmi.2020.2971006
44. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 35(5):1285–1298. https://doi.org/10.1109/TMI.2016.2528162
45. Siegel RL, Miller KD, Fedewa SA, Ahnen DJ, Meester RG, Barzi A, Jemal A (2017) Colorectal cancer statistics, 2017. CA Cancer J Clin 67(3):177–193. https://doi.org/10.3322/caac.21395
46. Siegel RL, Miller KD, Sauer AG, Fedewa SA, Butterly LF, Anderson JC, Cercek A, Smith RA, Jemal A (2020) Colorectal cancer statistics, 2020. Cancer J Clin CA. https://doi.org/10.3322/caac.21601
47. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556
48. Slaoui M, Fiette L (2010) Histopathology procedures: from tissue sampling to histopathological evaluation. Methods in molecular biology. Humana Press, Totowa, pp 69–82
49. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
50. Theodoridis S, Koutroumbas K (2008) Pattern recognition, 4th edn. Academic Press, USA

51. Tseng KK, Zhang R, Chen CM, Hassan MM (2020) DNetUnet: a semi-supervised CNN of medical image segmentation for super-computing AI service. J Supercomput. https://doi.org/10.1007/s11227-020-03407-7
52. Vapnik VN (1998) Statistical learning theory. Wiley-Interscience, Hoboken
53. Wang C, Shi J, Zhang Q, Ying S (2017) Histopathological image classification with bilinear convolutional neural networks. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp 4050–4053
54. Wang EK, Chen CM, Hassan MM, Almogren A (2020) A deep learning based medical image segmentation technique in internet-of-medical-things domain. Future Gener Comput Syst 108:135–144. https://doi.org/10.1016/j.future.2020.02.054
55. Wang P, Hu X, Li Y, Liu Q, Zhu X (2016) Automatic cell nuclei segmentation and classification of breast cancer histopathology images. Signal Process 122:1–13. https://doi.org/10.1016/j.sigpro.2015.11.011
56. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, Madabhushi A (2015) Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE Trans Med Imaging 35(1):119–130. https://doi.org/10.1109/TMI.2015.2458702
57. Xu Y, Jia Z, Wang LB, Ai Y, Zhang F, Lai M, Eric I, Chang C (2017) Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. BMC Bioinform 18(1):281. https://doi.org/10.1186/s12859-017-1685-x
58. Yu KH, Zhang C, Berry GJ, Altman RB, Ré C, Rubin DL, Snyder M (2016) Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. Nat Commun 7:12474. https://doi.org/10.1038/ncomms12474
59. Yu KH, Wang F, Berry GJ, Ré C, Altman RB, Snyder M, Kohane IS (2020) Classifying non-small cell lung cancer types and transcriptomic subtypes using convolutional neural networks. J Am Med Inform Assoc 27(5):757–769. https://doi.org/10.1093/jamia/ocz230
60. Zoph B, Le QV (2016) Neural architecture search with reinforcement learning. arXiv preprint arXiv:161101578

## Authors and Affiliations

**Elene Firmeza Ohata[1] · João Victor Souza das Chagas[1] · Gabriel Maia Bezerra[1] · Mohammad Mehedi Hassan[2] · Victor Hugo Costa de Albuquerque[1,3,4] · Pedro Pedrosa Rebouças Filho[1,3,4]**

Elene Firmeza Ohata
elene.ohata@lapisco.ifce.edu.br

João Victor Souza das Chagas
victor.souza@lapisco.ifce.edu.br

Gabriel Maia Bezerra
gabrielmaia@lapisco.ifce.edu.br

Mohammad Mehedi Hassan
mmhassan@ksu.edu.sa

Pedro Pedrosa Rebouças Filho
pedrosarf@ifce.edu.br

[1] Laboratório de Processamento de Imagens, Sinais e Computação Aplicada, Instituto Federal do Ceará, Fortaleza, CE, Brazil

[2] College of Computer and Information Sciences and Research Chair of Pervasive and Mobile Computing, King Saud University, Riyadh 11543, Saudi Arabia

[3] Department of Computer Science, Federal Institute of Education, Science and Technology of Ceará, Fortaleza, CE 60040-215, Brazil

[4] Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, CE 60020-181, Brazil