Intelligence-Based Medicine 5 (2021) 100040

Contents lists available at ScienceDirect

Intelligence-Based Medicine

journal homepage: www.sciencedirect.com/journal/intelligence-based-medicine

Medical image editing in the latent space of Generative Adversarial Networks

Rubén Fernández [a], Pilar Rosado [b], Esteban Vegas [a], Ferran Reverter [a,*]

[a] Department of Genetics, Microbiology and Statistics, Section of Statistics, Faculty of Biology, University of Barcelona, Diagonal, 643, 08028, Barcelona, Spain
[b] Department of Arts, Conservation and Restoration, Faculty of Fine Arts, University of Barcelona, Pau Gargallo, 4, 08028, Barcelona, Spain

ARTICLE INFO

Keywords:
Deep learning
Generative adversarial networks
Medical image editing
Histopathological images

ABSTRACT

We consider a set of arithmetic operations in the latent space of Generative Adversarial Networks (GANs) to edit histopathological images. We analyze thousands of image patches from whole-slide images of breast cancer metastases in histological lymph node sections. Image files were downloaded from the pathology contests CAMELYON 16 and 17. We show that widely known architectures, such as: Deep Convolutional Generative Adversarial Networks (DCGAN) and Conditional Deep Convolutional Generative Adversarial Networks (cDCGAN), allow image editing using semantic concepts that represent underlying visual patterns in histopathological images, expanding GAN's well-known capabilities in medical image editing. We computed the Grad-cam heatmap of real positive images and of generated positive images, validating that the highlighted features both in the real and synthetic images match. We also show that GANs can be used to generate quality images, making GANs a valuable resource for augmenting small medical imaging datasets.

1. Introduction

Medical imaging applications based on Deep Learning models have been growing in the last decade and it seems that the trend will continue in the coming years. Segmentation, pattern detection or classification are some of the most common tasks performed by these type of models, applied in a wide variety of application areas. For instance, a Deep Learning implementation that improves the average accuracy of a cohort of 29 pathologists when establishing the Gleason score for prostate cancer [1]. Other studies show results that exceed in performance those of humans while executing similar tasks [2], in skin cancer diagnosis or [3] for diabetic retinopathy detection.

These results are not limited to tasks where pathologists accuracy is improved but also to tasks where human work can be complemented by Deep Learning implementations, providing more objectivity and efficiency. For example, detecting mitosis in histological images of breast cancer [4], a biomarker used to determine the histological grade that reflects the severity of a tumor. Another application describes a system that allows discarding between 30% and 40% of the Whole Slide Images (WSI) a pathologist usually analyzes, allowing to reduce expert intervention [5]. Automatization of these tasks as well as standardized procedures that reduce human bias, allow pathologists to greatly decrease the workload to which they are subjected.

Deep Learning has opened an avenue of possibilities in the field of medical imaging analysis, however, it has a major drawback, the need of large amounts of tagged data. This fact has an even greater impact in this specific field, as in contrast with more general areas, labeling must be carried out by experts. This leads to a more important shortage on labeled images.

The importance of unsupervised methods such as some types of GANs, that can operate without any prior labeling is, for the aforementioned reasons, justified.

The original Generative Adversarial Networks or "vanilla GANs" as initially devised by Ian Goodfellow [6], are composed of fully connected layers. In 2015, Radford et al. [7] demonstrated that a new architecture based on a combination of convolutional neural networks (CNN) [8] and GANs showed, in addition to an easier training process, interesting properties that enable the semantic editing of images. This type of editing allows the modification of high-level features of the images by performing vector arithmetic in the low-dimensional latent space learned by the GAN.

This article focuses on computational analysis based on Deep Learning of histopathological images. The main goal is to show that semantic image editing can also be successfully applied in medical

* Corresponding author.
E-mail address: freverter@ub.edu (F. Reverter).

https://doi.org/10.1016/j.ibmed.2021.100040
Received 29 November 2020; Received in revised form 15 June 2021; Accepted 21 July 2021
Available online 28 July 2021

images, generating quality samples with visual attributes that are coherent with the operations performed on the original images.

A powerful implication of these results is the use of GANs to enrich annotated biomedical databases that can be later used to improve the learning process of supervised algorithms. We prove that an improved performance is achieved in different classification tasks with GAN augmented datasets versus a baseline dataset of histological images.

2. Data

WSI images in OpenSlide format [9] from 22 patients from the pathology contest CAMELYON [10], organized by the Radboud University Medical Center, were used. The goal of this challenge is to evaluate algorithms for automated detection and classification of breast cancer metastases in WSIs of histological lymph node sections. Seven of the images were part of the 2016 challenge and fifteen images belonged to the 2017 challenge.

Images are accompanied by a coordinate file indicating the metastasized areas. In order to select slides that contain larger areas of metastasized tissue and be able to get more positive samples, we select those WSIs whose coordinate files have a larger number of annotations. We also visually verify, using an histopathology image viewer (ASAP) [11], that the images effectively contain large areas of cancerous tissue.

Patches of size 256 × 256 were created in different zoom levels (0,1) by using WSI-analysis [12] and custom scripts [13] that read the coordinate files and determine if a patch covers a metastasized area or not, allowing us to label every patch as positive or negative (see Fig. 1). After this process we obtained thousands of image samples, this number varying depending on the zoom level (see Table 1).

In order to obtain a balanced scenario in terms of number of samples per class, two datasets were created by randomly subsampling as many positive images as negative patches we had for a certain zoom level. We obtained the two datasets, a zoom level 1 dataset with 22000 patches per class and a zoom level 0 dataset with 62062 patches per class. We want

Table 1
Image patches.

zoom level	negative patches	positive patches	patches per class in balanced scenario
1	116278	22122	22000
0	337165	62062	62062

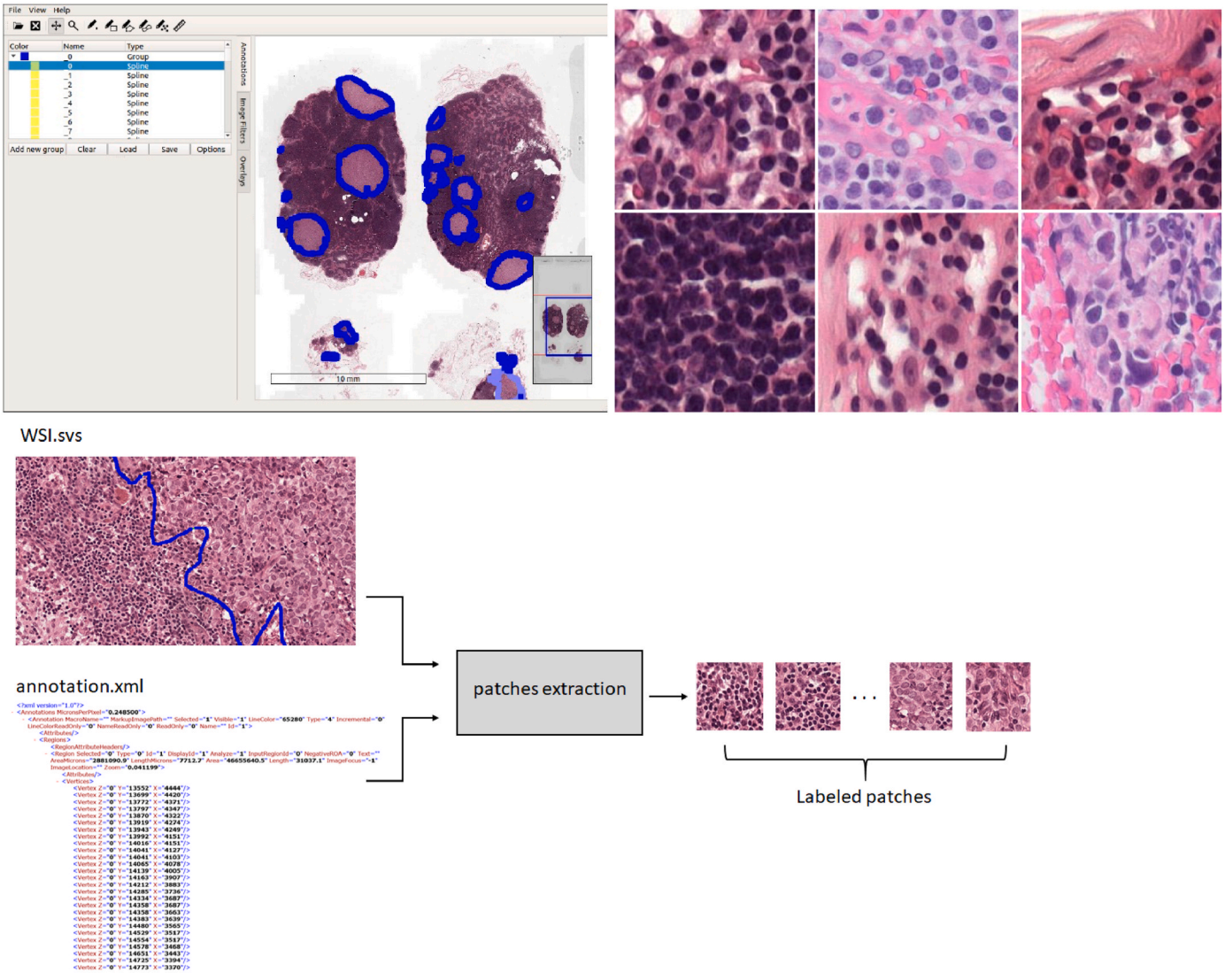

Fig. 1. Viewing a slide in ASAP. In blue, labeled by a coordinate file, the cancerous areas (Top, left). Patches from different patients with different stains (Top, right). Diagram of the patches' extraction process from a WSI file and a coordinate file. A patch is labeled positive when it comes from an annotated region. Negative patches come from unannotated regions (Bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

to highlight two features of the datasets.

First, although most implementations, including the original from the Radford et al. paper [7] work with 64 × 64 or at most 128 × 128 image sizes, we found some examples that showed that larger image sizes, in this case 256 × 256, could be successfully used. At the cost of much longer training times we decided to use a larger image size, as it was essential that we were able to identify certain visual features in order to tell if the vector arithmetic were working in a coherent way. These features would have been difficult to recognize in smaller histological images.

Second, in general, some of the patches show very different stains, brightness, contrast or picture quality. Specifically in our dataset, most of the samples were high quality images but they differed a lot both in the color of the stain and in the brightness and contrast, some of them showing a clear difference between features like the nucleus and the rest of the cell or the cell and the interstitial fluid, and some not due to low contrast or high brightness. Although this variability was not annotated, in practice the GAN has to learn to generate images explaining the underlying variability which adds difficulty to the process.

3. Methods

3.1. GAN architectures

Generative Adversarial Networks (GANs), introduced in 2014 by Goodfellow et al. [6] belong to the so-called generative deep learning models. In contrast to discriminative models, generative models are not only able to tell if an instance belongs to a given distribution but to generate new samples that fit that distribution. In computer vision, GAN models can learn the statistical nature of the latent space of images and they can sample from this space, producing new images with similar features to those the model has analyzed in its training phase. Many variants of GANs have been developed until now [14] which respond mainly to the different architectures and loss functions that they implement. In this work, we used two broadly known types: Deep Convolutional Generative Adversarial Networks (DCGAN) (see Fig. 2) and Conditional Deep Convolutional Generative Adversarial Networks (cDCGAN) (see Fig. 3).

GANs are machine learning models where two artificial neural networks compete with each other to become more accurate in their own tasks. The two neural networks that constitute a GAN are referred to as the generator and the discriminator. The goal of the generator is to produce outputs that could easily be mistaken for real data and the goal of the discriminator is to identify which of the inputs it receives have been artificially created by the generator and which are real data.

The training process of the GAN consists of alternating stochastic gradient descent steps. Two batches are sampled on each step: a batch of real data, x, from the dataset and a batch of synthetic data z generated from values drawn from the model's prior over latent space. Then two gradient steps are made simultaneously: one updating the discriminator weights, θ_D, to reduce the discriminator cost function, J^D, and one updating the generator weights, θ_G, to reduce the generator cost function, J^G.

Consider having n samples from each class, the discriminator cost can be expressed as

$$J^D = -\frac{1}{2n}\sum_{i=1}^{n}\log(D(x_i)) - \frac{1}{2n}\sum_{i=1}^{n}\log(1 - D(G(z_i))) \tag{1}$$

This is the cross-entropy cost that is minimized when training a standard binary classifier with sigmoid output. The only difference is that the classifier is trained on two batches of data; one coming from the dataset, where the label is 1 for all samples, and one coming from the generator, where the label is 0 for all samples. This formulation pushes the discriminator to output a probability close to 1 when samples come from the real distribution and a probability close to 0 when samples are outputs of the generator.

A complete specification requires that we specify a cost function also for the generator; the easiest formulation would be

$$J^G = -J^D \tag{2}$$

This means the generator is trained to maximize the discriminator's loss function, i. e to make the discriminator's job as difficult as possible. This has the nice property of being a zero-sum game, $J^G + J^D = 0$. By defining a value function as $V(\theta_D, \theta_G) = -J^D(\theta_D, \theta_G)$ the entire problem can be formulated as the minimax game:

$$(\theta_G^*, \theta_D^*) = \arg\min_{\theta_G}\max_{\theta_D} V(\theta_D, \theta_G) \tag{3}$$

Zero-sum games are also called minimax games because their solution involves minimization in an outer loop and maximization in an inner loop.

The loss function defined in (2) has some nice theoretical properties but is usually not the one used when training the generator. The cross-entropy error function will still be used but instead of flipping the sign of the discriminator loss when defining the generator loss, the target variables will be flipped. The generated samples get target one and the real ones get target zero. We can then see that the cost function of the generator is of the form

$$J^G = \frac{1}{n}\sum_{i=1}^{n}\log(D(G(z_i))) \tag{4}$$

This is no longer a zero-sum formulation and is heuristics-driven. With this formulation we prevent that both networks have weak gradients when falling behind in the competition.

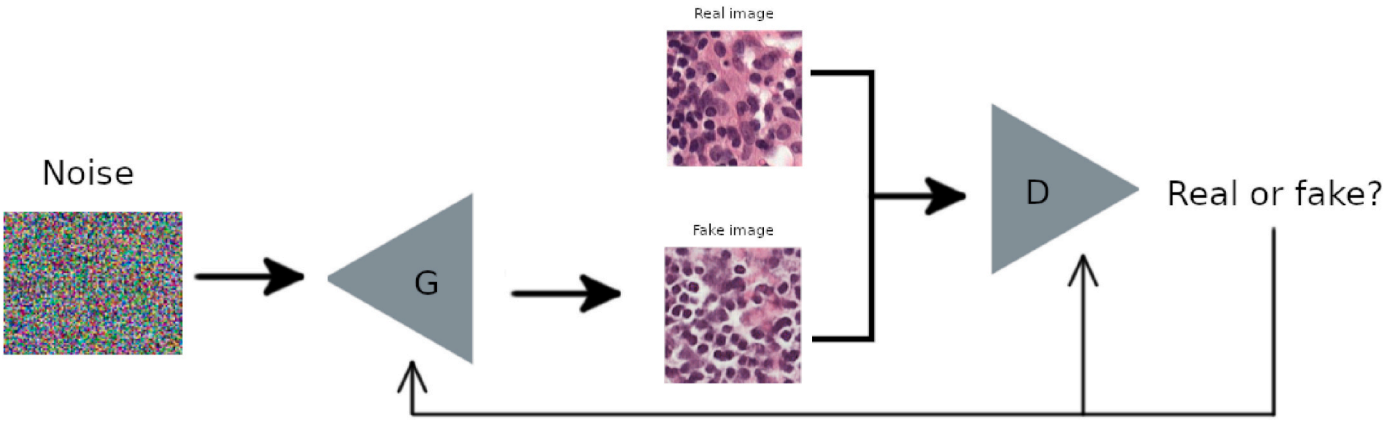

Fig. 2. DCGAN scheme. The basic components of a DCGAN are two convolutional neural networks – a generator that synthesizes new images from scratch, and a discriminator that takes images from both the training data and the generator's output and predicts if they are "real" or "fake".

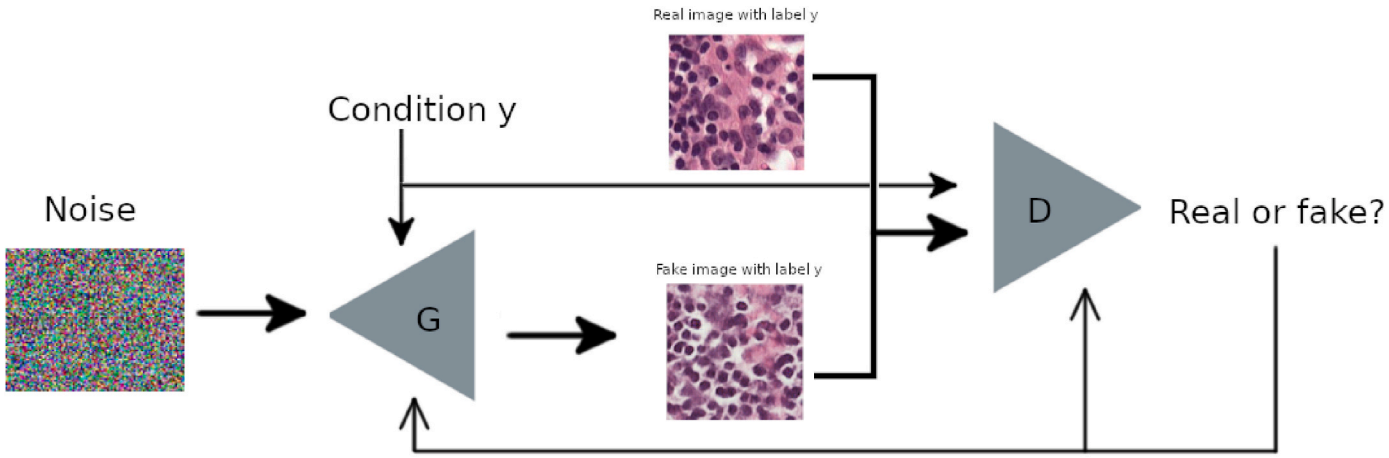

Fig. 3. cDCGAN scheme. DCGAN can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information *y*. *y* could be any type of auxiliary information, such as class labels or complementary data from other sources.

3.1.1. Deep convolutional Generative Adversarial Networks (DCGAN)

On the basis of the original "vanilla GANs", different versions, better adapted to image analysis, were devised. Radford et al. [7] proposed a new type of neural network that also combines antagonistic training with convolutional networks [8] that improved any of the already existing systems. These improvements consists of a more stable training process and the ability to generate quality images while being simpler in its structure.

The implemented DCGAN architecture (see Fig. 4) is very similar to what Radford et al. described in its paper [7] with the exception of a different number of filters per layer.

The latent space $\mathcal{Z}$ was taken as $\mathbb{R}^{100}$. $\mathcal{Z}$ during training it will be sampled at random, for this purpose multivariate uniform distribution in the range [-1,1] was used. Once such a latent space is developed you can pick points from it and by mapping them to image space generate new images.

The generator is made up of 4 transposed convolutional layers, with 5×5 kernels and stride 2. Each convolutional layer is followed by a batch normalization layer and ReLU activation function, except in the last layer where there is no batch normalization and the activation function is the hyperbolic tangent tanh.

The discriminator is composed of 4 convolutional layers with 5×5 and kernels and stride 2. Similarly, each convolutional layer is followed by another batch normalization layer except the first one. The activation function is LeakyRelu for all layers, with slope 0.2. With respect to the original version, half of the filters are used in each layer. This reduction respond to the fact that it does not lead to a drop in terms of quality of the generated samples while having a great impact in the training performance of the network.

Other differences with respect to Ref. [7] are:

1. Batch size of 64 instead of 128 due to hardware limitations (GPU memory size to store bigger batches).
2. Noise is added along with the inputs to the discriminator, in order to weaken it against the generator.
3. Label smoothing is used when calculating the discriminator error. In this way we obtain a similar effect as in the previous point, we avoid overfitting the discriminator, making it less accurate when classifying an image.
4. The generator trains twice for each time the discriminator does it. Except for the batch size, the searched effect of this variations is to let the generator to keep up with the discriminator. If the discriminator

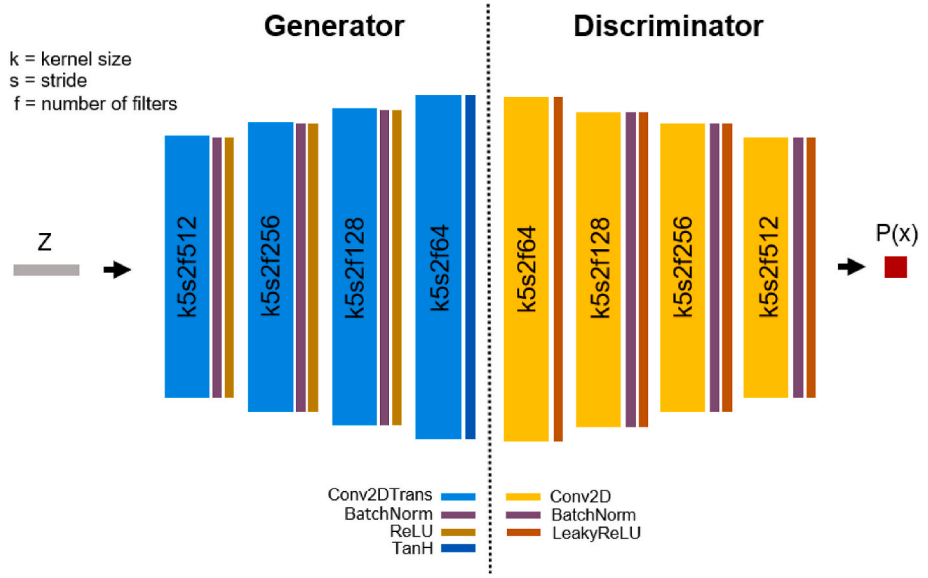

Fig. 4. DCGAN architecture. Four stacked convolutional layers in both the generator and discriminator. The input *z* denotes a vector in the latent space, the output *P* denotes the probability that the image *x* is real or generated.

gets too good at classifying the generator would not be able to improve anymore.

3.1.2. Conditional deep convolutional Generative Adversarial Networks (cDCGAN)

Introduced in Ref. [15], cDCGAN are a supervised version of the DCGAN where one or several attributes of the image are used during training and generation so that they provide a level of additional control on the generated image. During cDCGAN training, the generator learns to produce realistic examples for each annotated class in the training dataset, and the discriminator learns to distinguish fake example-label pairs from real example-label pairs. The discriminator in a cDCGAN does not learn to identify which class is which.

cDCGAN architecture is quite similar to DCGAN except for the input layers. In the input layer the feature vector is concatenated (usually as a one hot encoded vector) to the real image in the case of the discriminator, or to the input vector z in the case of the generator. When it comes to the discriminator the concatenation is done on the first convolutional layer and not directly on the input layer.

What this means is that the GAN can be conditionally trained so the output images contain one or various features from the feature vector. In our case, the cDCGAN was only class conditioned so the generated image would belong either to the positive or the negative group (tumorous or non-tumorous).

3.1.3. Work environment

Experiments were run on a computer with an Intel Core i7 6700 K (QuadCore, 4 GHz, 8 Mb), using 32 GB RAM and two AMD RX580 8 GB GDDR5 GPUs, running Ubuntu 18.04.

3.2. Image editing in the latent space

Once the GAN is trained, the generator encodes the large space of images it has been trained with. We can say that the generator maps each point of latent space $\mathcal{Z}$ to a new image that contain a different combination of visual patterns and characteristics learned from the collection of images that were used during the training phase.

The identification between points of latent space and images allows us to transfer the arithmetic between vectors to an arithmetic between images, and consequently to the possibility of editing images. We analyze four types of operations in latent space to assess their usefulness in editing medical images. We present next the following operations: latent vector reverse search, class inversion, basic arithmetic and interpolation between classes of images.

3.2.1. Latent vector reverse search

Given an image $\mathcal{I}$, we want to determine the vector $z \in \mathcal{Z}$ that encodes the image $\mathcal{I}$. That is, if we pass z through the generator we will get a very close approximation of the original image $\mathcal{I}$. The search in the space $\mathcal{Z}$ has been implemented through minimization of the root mean square error between the recovered and original images by gradient descent. Though latent vector reverse search is not itself an editing operation it is useful to apply the rest of operations when we do not have beforehand the vector that will produce a given image.

A useful application of reverse search is found when a set of histopathological images is given and we want to edit such an image set; the first step would be to determine the vectors in latent space that generate those images.

3.2.2. Basic arithmetic

To perform additional manipulations with the latent vectors, several operations have been defined, such as addition, subtraction and average. For instance, given three points $z_1, z_2, z_3 \in \mathcal{Z}$, the addition $z = z_1 + z_2$, the subtraction $z = z_1 - z_2$ and both $z = z_1 + z_2 - z_3$, determine new points $z \in \mathcal{Z}$. Then, if we pass z through the generator we can observe the visual effect of the operations performed.

We can also operate with averages of several latent vectors. That is, given two sets of vectors $z_1, \ldots, z_n \in \mathcal{Z}$ and $w_1, \ldots, w_m \in \mathcal{Z}$ each representing two different sets of visual characteristics of interest, we can compute the averages $\overline{z}$ and $\overline{w}$ and then apply basic arithmetic operations on that average vectors in order to obtain more reliable results.

Through these operations we can edit images with well-defined characteristics. We can combine features applying the add operation or remove them using the subtract operation. To illustrate the basic arithmetic with an example, let us suppose a study with histopathological images, showing three distinct visual patterns: P1, P2 and P3 (e.g. tubule formation, nuclear pleomorphism and high mitotic count). Then a new image that shows the features of patterns P1 and P2 can be edited by adding the latent vectors of real images having patterns P1 and P2. Analogously, a new image that combines the three patterns can be generated by adding latent vectors of real images showing patterns P1, P2 and P3. On the other hand, for a given histopathological image having both patterns P1 and P2, we can edit a new image only showing pattern P1 by subtracting the latent vector of the image with pattern P2 from the latent vector of the image with both patterns, P1 and P2.

3.2.3. Class inversion

Conditional DCGAN allow us to implement class inversion. Let ℓ_1 and ℓ_2 be the labels to indicate two-class images. Given an image $\mathcal{I}$ labeled as ℓ_1 and the latent vector $z \in \mathcal{Z}$ that encode $\mathcal{I}$ (either by saving it beforehand or by using reverse search). Then, if we keep the vector z fixed but we change the label vector ℓ_1 to ℓ_2 and pass the pair (z, ℓ_2) through the generator we can obtain an image that exhibits opposite features with respect to the original image $\mathcal{I}$.

Applying this operation, we can generate a new image that maintains the features of the original image and exhibits the essential characteristics of the other class. In our case, having only two classes, we can only alternate between positive and negative classes. Class inversion may be a useful strategy to expand an scarce image class.

3.2.4. Interpolation

Given two points $z_1, z_2 \in \mathcal{Z}$ we determine a point $z \in \mathcal{Z}$ by means:

$$z = \lambda z_1 + (1-\lambda) z_2, \qquad \lambda \in [0,1] \tag{5}$$

Having a well trained GAN, we will obtain a continuous latent space that we can travel through producing a smooth and coherent interpolation.

Given two images, one of them the initial and the other the final, this operation allows us to visualize how the initial image becomes the final image through a sequence of intermediate images. We can think of it as the way to generate the frames in a movie. An application of this operation can be illustrated by the process of generating a sequence of images from a normal cell to a tumor cell.

4. Results

Before discussing the results of the GAN in image editing, we will discuss the performance of the GAN in the generation of medical images of sufficient quality. This is a basic requirement for image editing to be useful.

4.1. Measuring the GAN performance

The most common way to measure the performance of GANs is indirectly, using the samples generated by the model in a surrogate task. For instance, we could have an external classifier, for which we know already its performance measuring real images, and let it classify the generated images. Evaluating the obtained results when performing this task [16] will give us a measure of the performance of the GAN.

More specifically, given a classifier whose task would be to correctly label real medical images into its different classes, we could use GAN generated images that share the same visual patterns of real images to

improve the accuracy of this classifier. It is known that having a larger sample size, if the extra instances are not redundant, usually leads to an improvement of the classifier's performance. We can then use the amount of improvement as an objective measure of the capabilities of the GAN to generate realistic images.

We can also train a classifier with a combined dataset of real and generated samples and compare it with the same classifier trained with only real samples. This gives us an idea of both the quality and the diversity of the samples the generative model is able to produce.

The implemented benchmarking method makes use of an Inception V3 classifier [17] trained on our datasets through transfer learning [18]. Classifier's task was to discriminate between positive and negatives samples from two datasets. The first dataset is composed of real histological images. Once the training is complete, results are evaluated on a set of test images generated by the GAN. The second dataset, this time of GAN generated images, is used to train the classifier again but its performance is measured by classifying real images. What we intended to prove is that if the generated images were realistic, that is, they consistently contain features of the real images, then the classifier will get high precision as it still can learn using those features and use the learned representation to accurately classify real images. On the other hand, if the version trained with real images obtains high performance evaluating generated images it would mean that those accurately reproduce patterns from the real images. Evaluations were performed with different amount of data and different training times at two different zoom levels, to give us a better picture and an idea of the impact other variables have on classification accuracy (Table 2 and Table 3).

We observe that the accuracy attained when classifying real images with the model trained with generated images is only slightly lower than the accuracy attained with the model trained with real images. Thus, it follows that the performance of the generator model is acceptable and useful for producing realistic images.

When training with generated images, increasing the number of samples results in an increase of accuracy when classifying real images. We then conclude that the generator network is not generating redundant samples. Images in a lesser zoom level (level 1) are easier to be realistically generated than images at closer zoom level (level 0).

4.1.1. Feature consistence

Neural networks are often depicted as black boxes as they do not provide the user with a clearer understanding of how the model reached a certain decision. Explainability is a desired property in a model because it can provide us with valuable insights. Understanding how the model solves a problem can helps us solve similar problems, think about improvements or learn from the system. It is also a useful tool to diagnose why a model fails when it does.

In order to better understand what is going on under the hood when we use this type of neural network to classify histological images we implemented Grad-cam, a method used commonly when dealing with convolutional neural networks. Grad-cam or Gradient-weighted Class Activation Mapping is a method to highlight in images, important regions that were at the origin of given prediction [19].

In order to obtain the heatmap of important features we flow the weights of a given target prediction into the final convolutional layer. In our case, we modified for this purpose the inception v3 classifier we use in previous sections. The idea is to visually check that the inception classifier works in a similar way when it classifies real images and fake ones, indicating that the GAN is generating faithful representations of the real images.

In order to facilitate the task we chose images where the features that determine that a sample is positive are clearly visible and visually separable from the rest of the image (i.e. in a corner, in one side of the image) as both the GAN and Grad-cam could have problems showing and highlighting fine details with clarity or small or intertwined features.

The feature we chose is pleomorphism of the nuclear envelope (see Fig. 5) where image irregularities in the nuclear size and shape are clear indicators of tumoral tissue. This image was classified as positive and Grad-cam was applied, highlighting the corner were most of this irregularities can be seen. Said otherwise, most of the reasons the model found to classify this sample as positive, are located in the part of the image under the red area.

In addition, to show if the features highlighted in the real and synthetic datasets match we calculated the Grad-cam of real positive images and generated positive images. As can be seen in Fig. 6 the class-features highlighted by the Grad-cam are consistent with real (top) and synthetic images (bottom). Columns, from left to right, correspond to the class activation heatmaps, sample images and transparent overlay with two different colormaps. Therefore, we have further evidence of the quality of the images generated by the GAN since the activations they induce in the final convolutional layer of the inception classifier are similar to those activated by real images.

Related to the benchmark results, an obvious utility for the generator would be its use in data augmentation.

4.1.2. Data augmentation

To further evaluate the performance of the GAN we implemented a data augmentation experiment in which we demonstrated that augmenting a small imbalanced dataset with generated images resulted in a improved accuracy in classification tasks (see Table 4). This improved performance is could already be inferred from our previous results if we consider the fact that training only on generated images was proved useful to classify real images (with only 10% drop in accuracy for level 1 images). The first experiment could be seen as the particular case in which the dataset we wanted to augment has size 0.

In this second experiment we used a dataset composed of 568 patches, 68 of them positive. We trained an Inception V3 classifier with this dataset obtaining 0.524 accuracy over a test dataset of 1000 unseen images. In a second stage we added 25 k positive patches and 25 k negative patches, all of them generated by our cDCGAN, and retrained the classifier with the same parameters. The classifying accuracy jumped to 0.75 over the same test set of images.

We concluded that these results prove that GAN generated images can be useful for data augmentation. An interesting use case for this would be a situation in which we have a small dataset that we cannot augment with real images because those additional images cannot be shared or made public, either because of ethical reasons or because

Table 2
Accuracy comparison of Inception V3 classifier using patches at zoom level 0.

Num	Num	trained with real images		trained with generated images	
samples	Iterations	tested with real	tested with generated	tested with generated	tested with real
5000	500	0.85	0.76	0.93	0.79
	5000	0.86	0.77	0.94	0.80
15000	500	0.88	0.78	0.96	0.82
	5000	0.87	0.79	0.94	0.82

Table 3
Accuracy comparison of Inception V3 classifier using patches at zoom level 1.

Num	Num	trained with real images		trained with generated images	
samples	Iterations	tested with real	tested with generated	tested with generated	tested with real
5000	500	0.90	0.86	0.94	0.85
	5000	0.91	0.84	0.94	0.85
15000	500	0.90	0.80	0.94	0.87
	5000	0.90	0.82	0.95	0.86

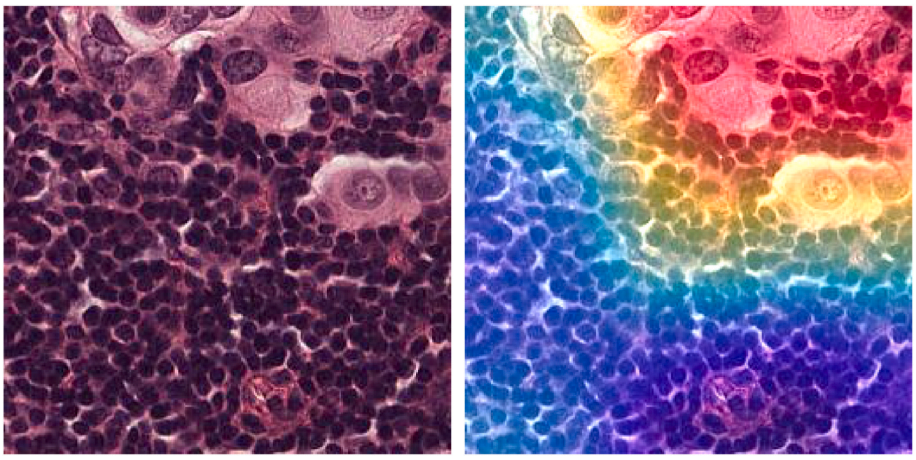

Fig. 5. Superimposing positive class activation heatmap on a real image.

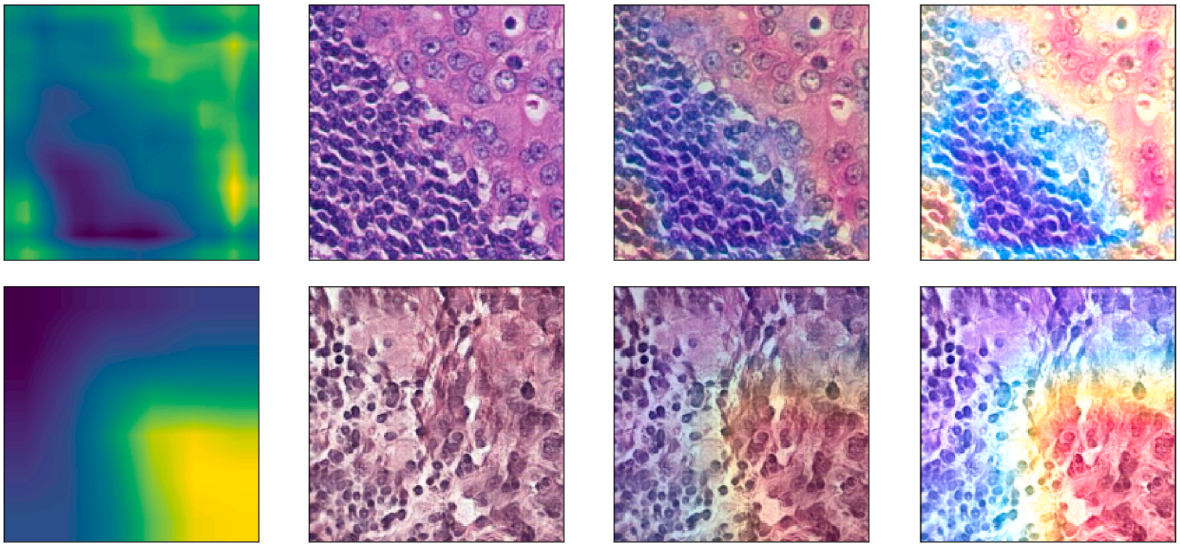

Fig. 6. Consistence of the class-features highlighted by the Grad-cam in real images (top) and synthetic images (bottom).

Table 4
Data augmentation test. Inception V3 classifier trained with transfer learning.

	small dataset	small imbalanced dataset	augmented dataset
generated samples	0	0	50000
negative samples	500	500	25500
positive samples	500	68	25068
training epochs	15	15	8
training accuracy	0.934	0.963	0.901
same unseen test dataset			
negative samples	500	500	500
positive samples	500	500	500
test accuracy	0.597	0.524	0.725

another team or research group cannot share the images at that time. In this case, even if the real images cannot be shared, maybe a GAN model trained of those images could, which would not have those privacy issues but still would allow us to augment our dataset.

An extra training with a small dataset, in this case balanced, was executed as reference to show that the improvement in accuracy was not only due to class balancing but also a product of an increased number of total samples after the generated images were added.

Once we have objectively evaluated the ability of the DCGAN and CDCGAN models to generate realistic samples of the two classes, we carried out a set of visual inspections to assess the usefulness of the vector arithmetic for medical image editing.

4.2. Image editing

We start by showing examples of latent vector reverse search (see Fig. 7). In the case of generated images (top-left), the reverse mapping is almost perfect (top-right). We cannot say the same when we apply reverse search of a real image (bottom-left) where the recovered image (bottom-right) does not get enough similarity. However, we can improve that result both by improving the GAN training process and/or by increasing search time in the reverse searching process.

We will discuss results regarding the class inversion. The top three images in Fig. 8 correspond to three different samples of healthy tissue (negative class) and at the bottom we see the images generated when inverting the class (inverted negative class). In all examples we can see that both images, negative and inverted negative, share visual features except those that are essential to determine that an image is negative. Analogously, we can see the inversion of a positive class, top images in Fig. 9 correspond to samples of cancerous tissue (positive class) and at the bottom we see the images resulting from inverting the class (inverted

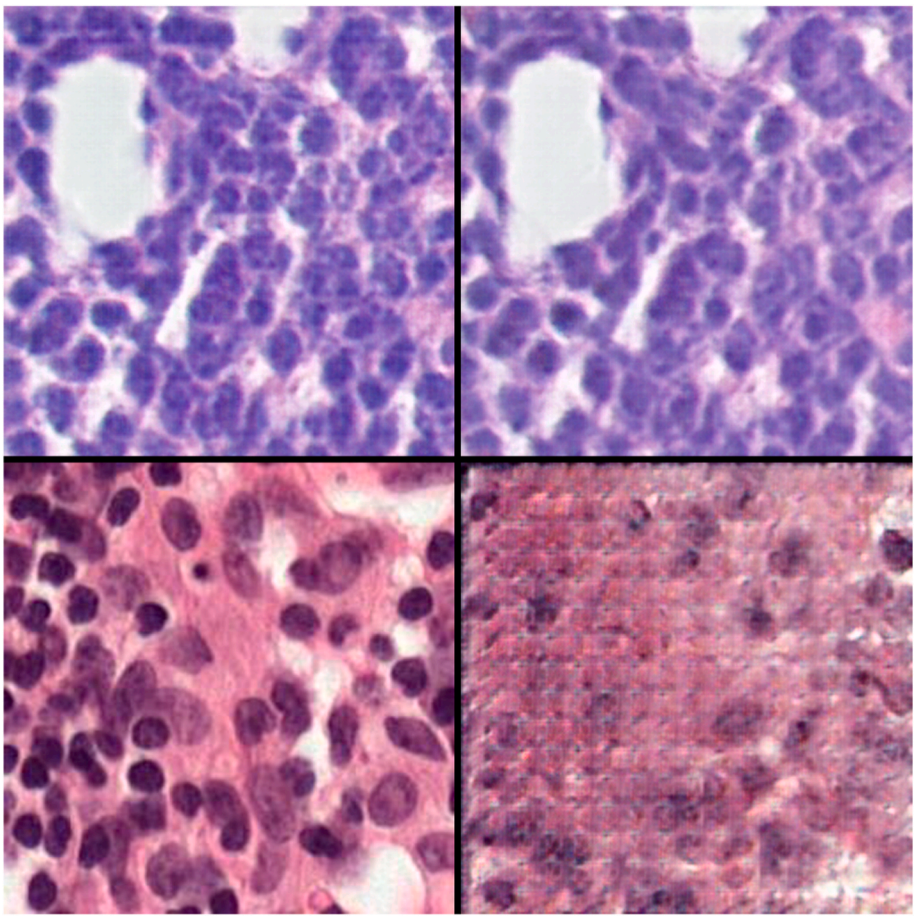

Fig. 7. Vector reverse search. On the left, the target images and, on the right, recovered images. Reverse search operation is easier for generated images (top) compared with real images (bottom).

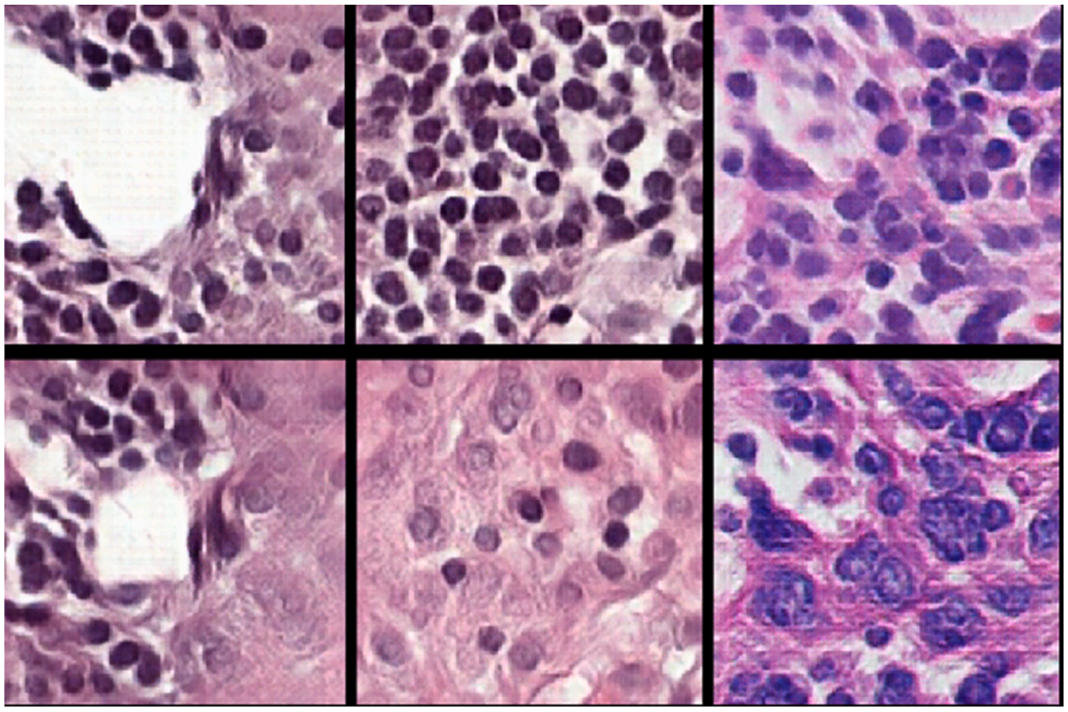

Fig. 8. Inverted negative class. On top three samples of healthy tissue. Bottom the images obtained by inverting the class (negative to positive). Many visual features of the original images remain, but cancerous tissue patterns appear.

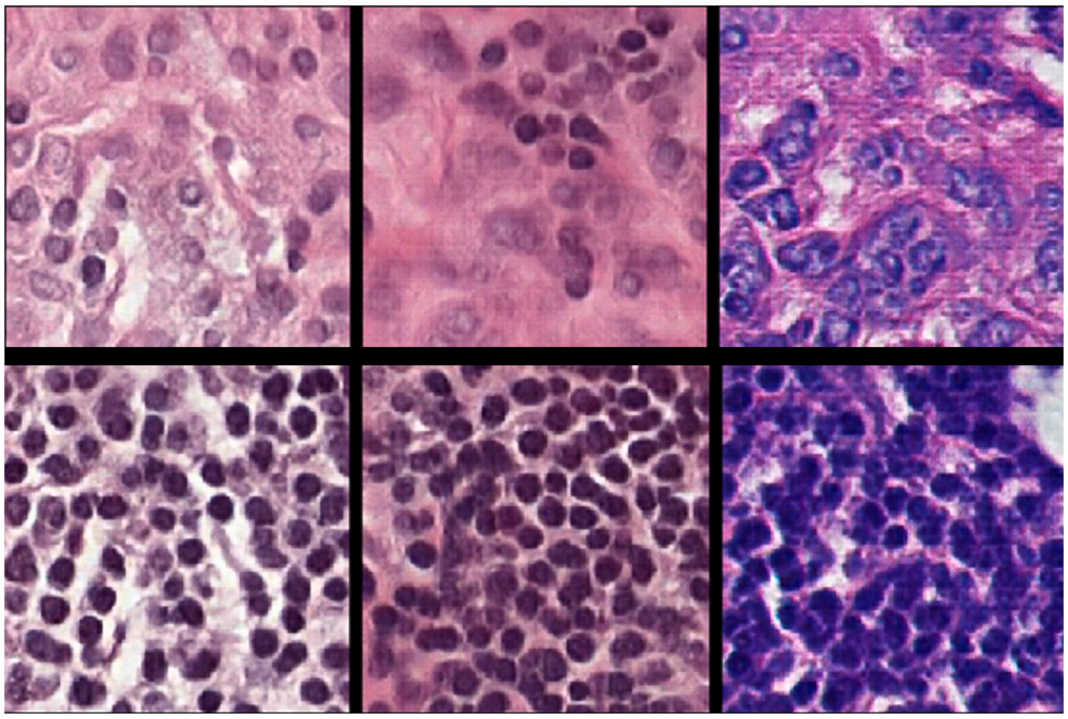

Fig. 9. Inverted positive class. On top three samples of cancerous tissue. Bottom the images obtained by inverting the class (positive to negative). Many visual features of the original images remain, but healthy tissue patterns appear.

positive class). Also, we can see the generated images preserve most of the features from the original image but not those that determine that an image is positive. Changes may seem too abrupt because a single label makes the difference between healthy and cancerous samples, which causes a single parameter to control several features that determine the class and these are block-wise activated or deactivated. With more annotation tags, finer and smoother transitions between image classes could be generated.

Next we can see examples of vector arithmetic showing the addition or subtraction of vectors representing concepts underlaying the images such as presence or absence of lumen, cancerous or healthy tissue. In the first row of the panel in Fig. 10 we can see cancerous samples with visible lumen. In the second row we can observe cancerous samples without lumen and in the third row we can see normal samples without lumen. Then the first operation subtracted the cancer pattern while maintaining the lumen pattern and the second operation adds the healthy pattern to the lumen. Thus, in the last row, we generated a sample representing healthy tissue with lumen. The fourth column shows the same operations but applied to a vector resulting from averaging the latent vectors that generate the images in preceding columns. The result is noticeably better when operating on the averages, where it can observed that lumen was added.

Interpolation results depend on the type of GAN employed; DCGAN or cDCGAN. With DCGAN, no matter what two points z_1 and z_2 we choose in the latent space we will have a continuous latent space that we can go through and the interpolation will always be smooth as long as the GAN was well enough trained. As we can see in Fig. 11 the image is coherent in any of the six directions in which we advance, seeing different attributes appear or disappear so that the image always makes sense.

With cDCGAN, having only two classes in our data set, an abrupt change can be observed when toggling the vector that encodes the class (see Fig. 12), therefore smooth transitions between any sample cannot be guaranteed, only in the case that the two samples belong to the same class. As with the class inversion, this problem is mitigated if we have more annotations in the labels vector, since we can do finer interpolations by changing specific components of the class vector.

4.2.1. *Availability of demo material*

A demo showing the image editing with DCGAN/cDCGAN is available in a notebook.[1]

5. Discussion

DCGANs and cDCGANs define a consistent latent space that can be continuously explored to generate sequences of medical images that reflect the transition from one stage to another in a gradual manner. We have seen that we can move from positive samples, which show tubular formations and nuclear pleomorphism to negative samples lacking these disease phenotypes.

It would be interesting to extend the analysis to other types of images other than histopathological ones, which perhaps due to their microscopic nature, show patterns that are more difficult to represent by arithmetic operations. An interesting case at a more macroscopic level could be its use with magnetic resonance or computed tomography scan images of different organs or with bone images with which to try to generate images that represent growths, fractures or other types of transitions between different states. When large databases of histological images of patients with different treatments are available, we will have visual information about how treatments affect cell and tissue structures, as treatments induce characteristic visual phenotypes.

The ability to interpolate between images associated with different treatments, and produce quality synthetic images will be a tool to visualize the mechanism of action of the way treatments modify tissue formations. It will help us to better understand diseases and even guide us in the development of new treatments. An application that seems interesting could be its use to generate medical images as didactic material for medical doctors training in the interpretation of images.

[1] https://colab.research.google.com/drive/1tO6mqE2rYEmerjJ5vAjNOPFytBAyuw2N?usp=sharing.

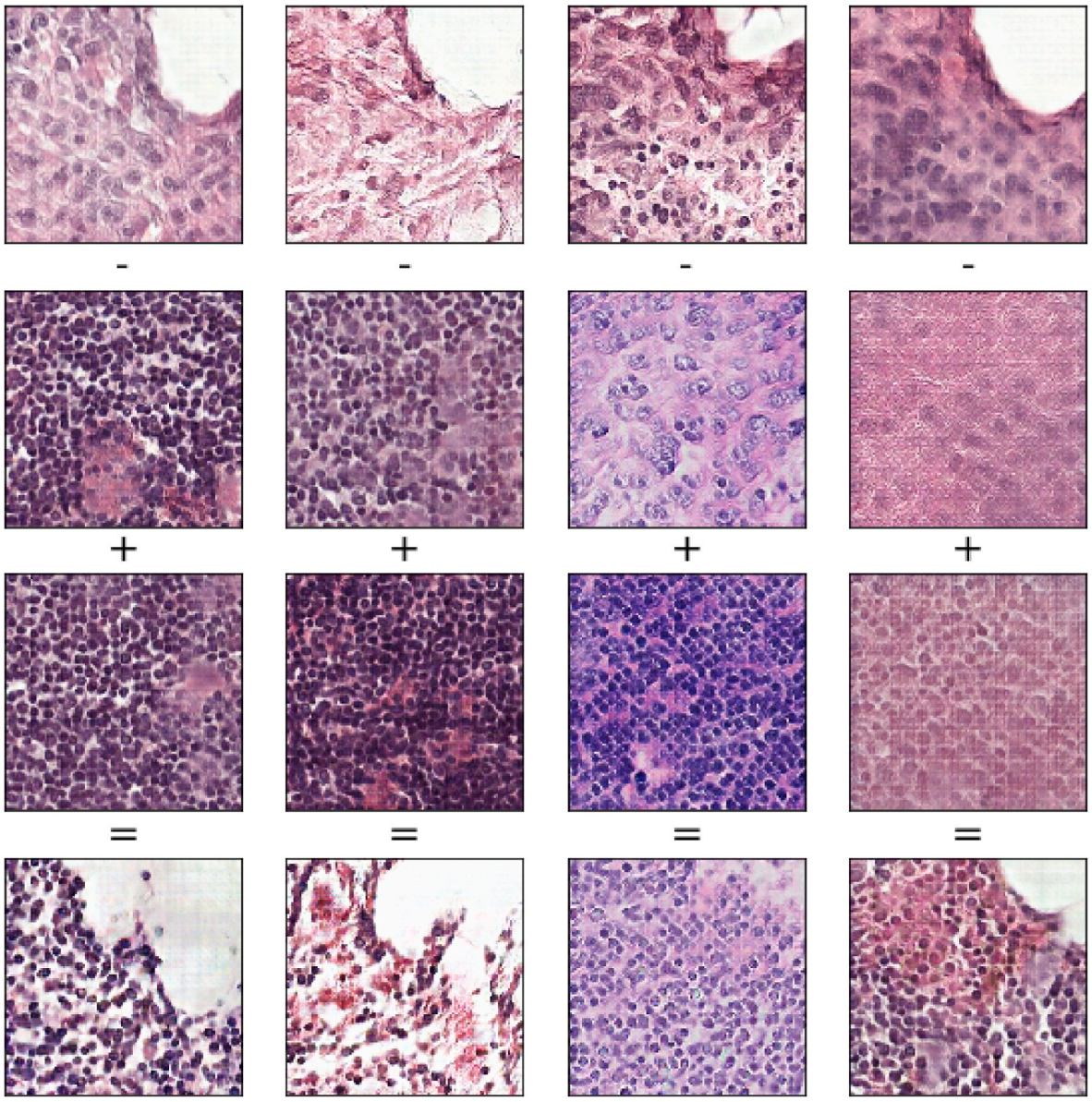

Fig. 10. Examples of vector arithmetic showing the addition or subtraction of vectors representing concepts underlaying the images such as presence or absence of lumen, cancerous or healthy tissue. More details can be found in the text.

Additionally, we show the usefulness of GANs in data augmentation. Synthetic images share visual features similar to real images as we saw in section 4.1.1, which facilitates the use of GANs to augment databases that are often not large in the medical field, and thus reinforce the application of other algorithms in tasks that require supervised learning. Nevertheless, the use of GANs in data augmentation needs to be explored more rigorously through the use of evaluation methods appropriate to the particularities of each type of image.

6. Conclusions

Deep learning is a growing field in medical image applications but until recently most of these applications relied on heavy labeling of samples. Unsupervised methods, which do not require labeling by a human, such as most of GANs implementations, which are the base for this paper, will be increasingly important as the number of applications and the volume of data increases exponentially in next decades.

In this study, we show that histopathological images can be edited using vector arithmetic in the latent space of GANs with promising results. We observed that underlying visual patterns in medical images can be added, subtracted, averaged or even interpolated, by applying arithmetical operations between latent vectors. One of the most obvious implications of these results is its use in data augmentation. We proved its usefulness augmenting datasets through a set of classification tasks, comparing performance of small datasets versus augmented datasets. Grad-cam was computed to asses consistency of the features in the generated samples, as the highlighted characteristics were shared between real and generated images.

As a final conclusion, even though the results are visually spectacular, training and using a GAN is not a simple task and needs to be explored experimentally. Applications based on this technology will likely need heavy tuning for results to be highly relevant. However, solutions around this technology are in continuous development and it is something teams working in medical imaging should consider as a new helpful tool over the following years.

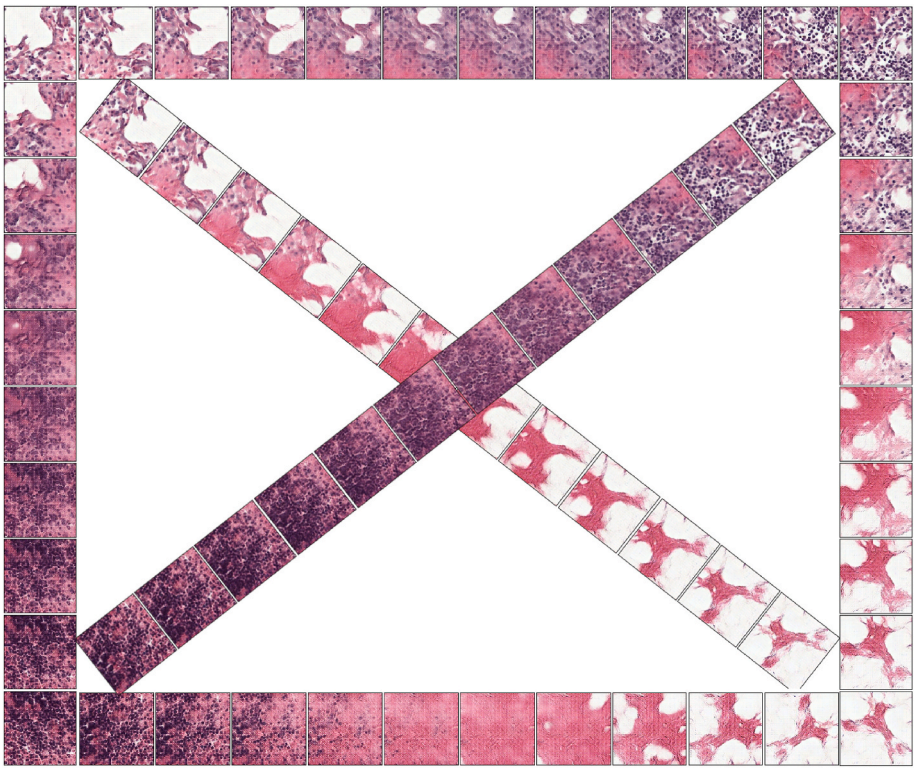

Fig. 11. DCGAN Interpolation. Images are coherent regardless of the path we interpolate. Transitions are smooth and show meaningful attributes.

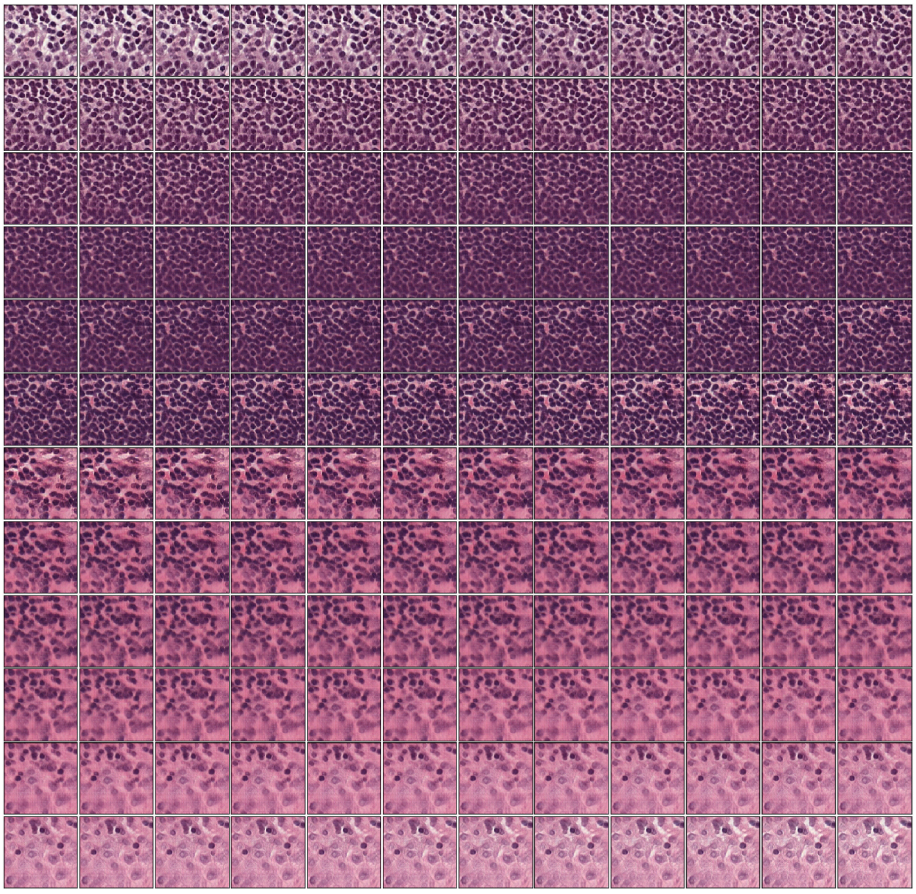

Fig. 12. cDCGAN Interpolation. Transitions are smooth only in the case that the two images that define the interpolation are of the same class.

Declaration of competing interest

None.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported in part by a grant from the 2017 SGR 622 (Generalitat de Catalunya) and in part from the PID2019-104830RB-I00 (Ministerio de Ciencia e Innovación y Ministerio de Universidades), and the support from Consorci de Serveis Universitaris de Catalunya (CSUC). We are grateful to everyone at the CAMELYON challenges.

References

[1] Nagpal Kunal, Foote Davis, Liu Yun, Chen Po-Hsuan, Wulczyn Ellery, Tan Fraser, Olson Niels, Smith Jenny L, Mohtashamian Arash, Wren James H, Corrado Greg S, MacDonald Robert, Peng Lily H, Amin Mahul B, Evans Andrew J, Sangoi Ankur R, Mermel Craig H, Hipp Jason D, Stumpe Martin C. Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer. 2018.

[2] Esteva Andre, Kuprel Brett, Roberto A Novoa, Ko Justin, Swetter Susan M, Blau Helen M, Thrun Sebastian. Dermatologist-level classification of skin cancer with deep neural networks. Nature 2017;542(7639):115–8.

[3] Gulshan Varun, Peng Lily, Coram Marc, Stumpe Martin C, Wu Derek, Narayanaswamy Arunachalam, Venugopalan Subhashini, Widner Kasumi, Madams Tom, Cuadros Jorge, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama 2016;316(22):2402–10.

[4] Cireşan Dan C, Giusti Alessandro, Gambardella Luca M, Schmidhuber Jürgen. Mitosis detection in breast cancer histology images with deep neural networks. In: International conference on medical image computing and computer-assisted intervention. Springer; 2013. p. 411–8.

[5] Litjens Geert, Sánchez Clara I, Timofeeva Nadya, Hermsen Meyke, Nagtegaal Iris, Kovacs Iringo, Hulsbergen-Van De Kaa Christina, Bult Peter, Van Ginneken Bram, Laak Jeroen Van Der. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Sci Rep 2016;6:26286.

[6] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, Bengio Yoshua. Generative adversarial nets. In Advances in neural information processing systems. 2014. p. 2672–80.

[7] Radford Alec, Metz Luke, Chintala Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. arXiv preprint arXiv:1511.06434.

[8] Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep learning. MIT Press; 2016.

[9] Goode Adam, Gilbert Benjamin, Harkes Jan, Jukic Drazen, Satyanarayanan Mahadev. Openslide: a vendor-neutral software foundation for digital pathology. J Pathol Inf 2013;4.

[10] Litjens Geert, Bandi Peter, Ehteshami Bejnordi Babak, Geessink Oscar, Balkenhol Maschenka, Bult Peter, et al. 1399 he-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. GigaScience 2018;7(6): giy065.

[11] Computational Pathology Group Radboud University Medical. Automated slide analysis platform, asap. https://computationalpathologygroup.github.io/ASAP/. [Accessed 8 September 2020].

[12] Wsi-analysis. https://github.com/tcxxxx/WSI-analysis. [Accessed 8 September 2020].

[13] Fernández Rubén. Deep learning para la generación de imágenes histopatológicas realistas mediante aritmética de vectores conceptuales. Master's thesis. Universitat Oberta de Catalunya-Universitat de Barcelona (UOC-UB); 2019.

[14] Wang Zhengwei, She Qi, Ward Tomas E. Generative adversarial networks in computer vision: a survey and taxonomy. 2019. arXiv preprint arXiv:1906.01529.

[15] Mirza Mehdi, Osindero Simon. Conditional generative adversarial nets. 2014.

[16] Manisha P, Gujar Sujit. Generative adversarial networks (gans): what it can generate and what it cannot?. 2018. ArXiv, abs/1804.00140.

[17] Ioffe S, Shlens J, Szegedy C, Vanhoucke V, Wojna Z. Rethinking the inception architecture for computer vision. 2015.

[18] Google. How to retrain an image classifier for new categories — tensorflow hub — tensorflow. https://www.tensorflow.org.

[19] Ramprasaath R Selvaraju, Cogswell Michael, Das Abhishek, Vedantam Ramakrishna, Parikh Devi, Batra Dhruv. Grad-cam: visual explanations from deep networks via gradient-based localization. In: In Proceedings of the IEEE international conference on computer vision; 2017. p. 618–26.