Neural Image Compression for Gigapixel Histopathology Image Analysis

David Tellez [iD], Geert Litjens [iD], Jeroen van der Laak [iD], and Francesco Ciompi

Abstract—We propose Neural Image Compression (NIC), a two-step method to build convolutional neural networks for gigapixel image analysis solely using weak image-level labels. First, gigapixel images are compressed using a neural network trained in an unsupervised fashion, retaining high-level information while suppressing pixel-level noise. Second, a convolutional neural network (CNN) is trained on these compressed image representations to predict image-level labels, avoiding the need for fine-grained manual annotations. We compared several encoding strategies, namely reconstruction error minimization, contrastive training and adversarial feature learning, and evaluated NIC on a synthetic task and two public histopathology datasets. We found that NIC can exploit visual cues associated with image-level labels successfully, integrating both global and local visual information. Furthermore, we visualized the regions of the input gigapixel images where the CNN attended to, and confirmed that they overlapped with annotations from human experts.

Index Terms—Gigapixel image analysis, computational pathology, convolutional neural networks, representation learning

✦

1 INTRODUCTION

GIGAPIXEL images are three-dimensional arrays composed of more than 1 billion pixels; these are common in fields like Computational Pathology [1] and Remote Sensing [2], and are often associated with labels at image level. The fundamental challenge of gigapixel image analysis with weak image-level labels resides in the low signal-to-noise ratio present in these images. Typically, the signal consists of a subtle combination of high- and low-level patterns that are related to the image-level label, while most of the pixels behave as distracting noise. Furthermore, the nature and spatial distribution of the signal are both unknown, often referred to as the *what* and the *where* problems, respectively.

1.1 The *what* and the *where* Problems

Researchers have addressed the challenge of gigapixel image analysis by making different assumptions about the signal, simplifying either the *what* or the *where* problem.

The most widespread simplification assumes that the signal is fully recognizable at a low level of abstraction, i.e., the image-level label has a patch-level representation. This simplification addresses the *what* problem by decomposing the gigapixel image into a set of patches that can be independently annotated. Typically, these patches are manually annotated to perform automatic detection or segmentation using a neural network, relegating the task of performing image-level prediction to a rule-based decision model about the patch-level predictions [1], [3], [4], [5]. This assumption is not valid for image-level labels that do not have a known patch-level representation. Furthermore, patch-level annotation in gigapixel images is a tedious, time consuming and error-prone process, and limits what machine learning models can learn to the knowledge of human annotators.

Other researchers have assumed that the signal can exist at a low level of abstraction, but it is then not fully recognizable, i.e., the image-level label has a patch-level representation that is unknown to human annotators. Furthermore, the mere presence of these patches is enough evidence to make a prediction at the image level, ignoring the spatial arrangement between patches, thus solving the *where* problem. Making this assumption falls into the multiple-instance learning (MIL) framework, which reduces the gigapixel image analysis problem into detecting patches that contain the true signal while suppressing the noisy ones [6], [7], [8], [9], [10]. However, these methods can only take into account patterns present within individual patches, neglecting the potential relationships among them. More generally, MIL techniques cannot exploit patterns present in higher levels of abstraction since they ignore the spatial distribution among patches. This is also true for methods that aggregate patch-level information by means of spatial pooling [6], [11].

In this work, we do not make any assumptions about the nature or spatial distribution of the visual cues associated with image-level labels. We argue that convolutional neural networks (CNN) are designed to solve the *what* and the *where* problems simultaneously [12], and propose a method to use them for gigapixel image analysis. However, feeding CNNs directly with gigapixel images is computationally unfeasible. Instead, we propose Neural Image Compression (NIC), a technique that maps images from a low-level pixel space to a higher-level latent space using neural networks.

• *The authors are with the Diagnostic Image Analysis Group, Department of Pathology, Radboud University Medical Center, Nijmegen 6500HB, The Netherlands. E-mail: david.tellezm@gmail.com, {Geert.Litjens, Jeroen. vanderlaak, francesco.ciompi}@radboudumc.nl.*
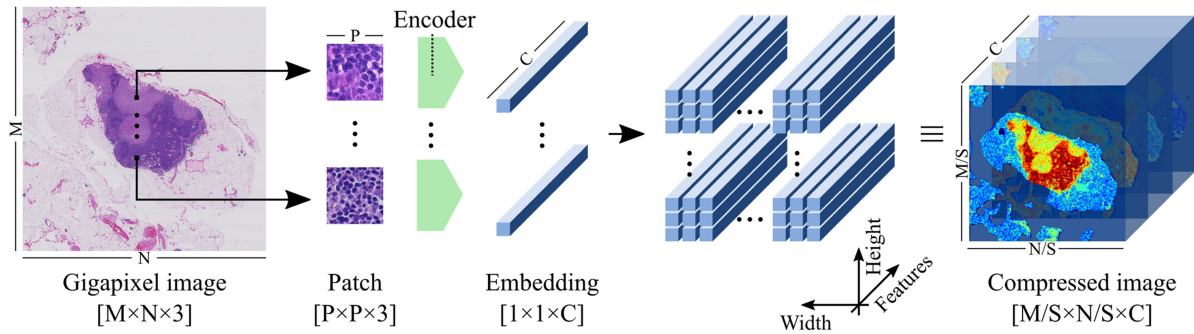
Fig. 1. Gigapixel neural image compression. Left: a gigapixel histopathology whole-slide image is divided into a set of patches mapped to a set of low-dimensional embedding vectors using a neural network (the encoder). Center: these embeddings are stored keeping the spatial arrangement of the original patches. Right: the resulting array is a compressed representation of the gigapixel image. M and N: size of the gigapixel image; P: size of the square patches; C: size of the embedding vectors; and S: stride used to sample the patches. Typically: $M = N = 50,000$ and $P = S = C = 128$.

In this way, gigapixel images are compressed into a highly compact representation, which can be used to train a CNN using a single GPU for predicting any kind of image-level label.

1.2 Neural Image Compression

Gigapixel NIC was designed to reduce the size of a gigapixel image while retaining semantic information by shrinking its spatial dimensions and growing along the feature direction (see Fig. 1). The method works by, first, dividing the gigapixel image into a set of high-resolution patches. Second, each high-resolution patch is compressed with a neural network (the *encoder*) that maps every image into a low-dimensional embedding vector. Finally, each embedding is placed into an array that keeps the original spatial arrangement intact so that neighbor embeddings in the array represent neighbor patches in the original image.

NIC was inspired by cognitive mechanisms. Human observers can describe complex visual patterns using only a few words without needing to describe each individual pixel. Similarly, the *encoder* can describe patches with low-dimensional embedding vectors, ignoring superfluous details. It is a powerful method that competes with classical approaches in terms of compression rate [13]. Moreover, previous works on representation learning and transfer learning have demonstrated that neural networks excel at extracting features that can be exploited by other networks to solve a variety of downstream tasks [14], [15], [16], [17]. This makes NIC an ideal candidate for reducing the size of gigapixel images before feeding a CNN.

The *encoder* network can be trained using a wide variety of techniques. In this work, we selected and compared representative methods from three well-known families of unsupervised representation learning algorithms: reconstruction error minimization, contrastive training, and adversarial feature learning. First, autoencoders (AE) have been proposed as a straightforward method to learn a compact representation of a given data manifold [12]. AEs are neural networks that follow a particular encoder-bottleneck-decoder architecture. They aim to reconstruct input images by minimizing a reconstruction loss, e.g., the mean squared error (MSE). In particular, we considered the case of the variational autoencoder (VAE), a powerful modification of the original AE that relies on a probabilistic approach [18]. Second, we investigated a discriminative model based on contrastive

training [16], [19], [20], [21]. This model senses the world via an encoding network that maps images to embedding vectors. By training this model to distinguish between pairs of images with *same* or *different* semantic information, the encoder is enforced to learn a compact representation of the input data. Third, we investigated adversarial feature learning [14], [15], a training framework based on Generative Adversarial Networks (GAN) [22]. GANs emerged as powerful generative models that map low-dimensional latent distributions into complex data. There is evidence that these latent spaces capture some of the high-level semantic information present in the data [23]. However, standard GAN models do not support the reverse operation, i.e., mapping data to the latent space. The Bidirectional GAN model (BiGAN [14]) learns this mapping using an explicit encoding network in the training loop. Intuitively, the encoder benefits from all the high-level features which were fully automatically discovered by the generator.

1.3 Gigapixel Image Analysis

Without any loss of generality, we applied our method to two of the largest publicly available histopathology datasets to demonstrate its effectiveness in real-world applications: the *Camelyon16* Challenge [4] and the *TUPAC16* Challenge [3]. These datasets consist of gigapixel images of human tissue acquired with brightfield microscopy at very high magnification, also known as whole-slide images (WSI). These WSIs were stained with hematoxylin and eosin (H&E), the most widely used stain in routine histopathology diagnostics, that highlights general tissue morphology such as cell nuclei and cytoplasm. Each WSI is associated with a single image-level label: the presence of tumor metastasis for *Camelyon16*, and the tumor proliferation speed based on gene-expression profiling for *TUPAC16*.

A benefit of using a CNN for gigapixel image analysis is that, once trained, the CNN's areas of interest in the input image can be visualized using gradient-weighted class-activation maps (Grad-CAM) [24]. These saliency maps provide an answer to the *where* problem by locating visual cues related to the image-level labels. Identifying visual evidence for CNN predictions is of utmost importance in the medical domain regarding algorithm interpretation and knowledge discovery. For the first time, we performed this saliency analysis on gigapixel images and compared the resulting maps with the patch-level annotations of an expert observer.

1.4 Contributions

This work is an extension of our conference paper [25]. A number of additions have been made: three new datasets, an additional encoding method, the Grad-CAM analysis, a new experiment at the patch level, a new experiment at the image level, a more thorough evaluation using cross-validation, and an independent test evaluation performed by a third-party.

Our contributions can be summarized as follows:

- We propose Neural Image Compression as a method to reduce gigapixel images to highly-compact representations, suitable for training a CNN end-to-end to predict image-level labels using a single GPU and standard deep learning techniques.
- We compared several encoding methods that map high-resolution image patches to low-dimensional embedding vectors based on different unsupervised learning techniques: reconstruction error minimization, contrastive training, and adversarial feature learning.
- We evaluated NIC in three publicly available datasets: a synthetic set designed to evaluate the method; and two histopathological breast cancer sets of whole-slide images used to train the system to predict the presence of tumor metastasis and the tumor proliferation speed.
- We generated saliency maps representing the CNN's areas of interest in the image in order to discover and localize visual cues associated to the image-level labels.

The paper is organized as follows: Sections 2 and 3 describe the methods in depth; Materials and experimental results are described in Section 4; the discussions and conclusions are stated in Sections 5 and 6, respectively.

2 NEURAL IMAGE COMPRESSION

Let us define $\omega \in \mathbb{R}^{M \times N \times 3}$ as the gigapixel image (e.g., a WSI) to be compressed, with M rows, N columns, and three color channels (RGB). In order to compress ω into a more compact representation ω', two steps were taken. First, ω was divided into a set of high-resolution patches $X = \{x_{ij}\}$ with $x_{ij} \in \mathbb{R}^{P \times P \times 3}$, sampled from the ith row and jth column of an uniform grid of square patches of size P using a stride of S throughout ω. Second, each x_{ij} was compressed independently from each other, generating a set of low-dimensional embedding vectors of length C at each spatial location on the grid: $Y = \{e_{ij}\}$ with $e_{ij} \in \mathbb{R}^C$.

We formulated the task of mapping high-entropy X into low-entropy Y as an instance of an unsupervised representation learning problem, and parameterized this mapping function with a neural network E so that $X \xrightarrow{E} Y$. By sliding E throughout all ij spatial locations, ω was compressed into ω' with a total volume reduction of $F = 3 \frac{S^2}{C}$. More formally

$$\omega \in \mathbb{R}^{M \times N \times 3} \xrightarrow{E} \omega' \in \mathbb{R}^{\frac{M}{S} \times \frac{N}{S} \times C}. \qquad (1)$$

We investigated several unsupervised encoding strategies for learning E. Three of the most well-known and accessible methods in unsupervised image representation learning were selected. In all cases, neural networks were trained to
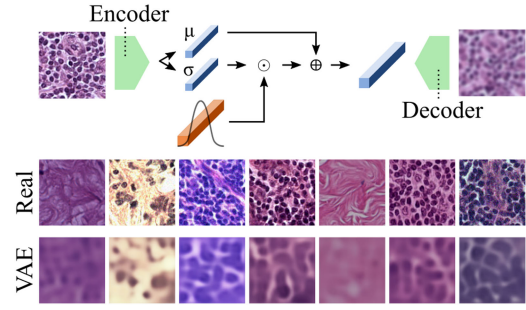

Fig. 2. Variational Autoencoder. Top: the encoder maps a patch to an embedding vector depending on a noise vector while the decoder reconstructs the original patch from the embedding vector. Bottom: pairs of real and reconstructed patch samples using $C = 128$.

solve an auxiliary task and learn E as a by-product of the training process. Note that none of the studied methods required the use of manual annotations. Network architectures and training protocols are detailed in the Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2936841, accompanying this paper.

2.1 Variational Autoencoder

Two networks are trained simultaneously, the encoder E and the decoder D. The task of E is to map an input patch x into a compact embedded representation e, and the task of D is to reconstruct x from e, producing x'. In this work, we used a more sophisticated version of AE, the variational autoencoder [18]. The encoder in the VAE model learns to describe x with an entire probability distribution instead of a single vector (Fig. 2). More formally, E outputs $\mu \in \mathbb{R}^C$ and $\sigma \in \mathbb{R}^C$, two embeddings representing the mean and standard deviation of a normal distribution such that

$$e = \mu + \sigma \odot n, \qquad (2)$$

with $n \sim \mathcal{N}(0,1)$ and $\odot$ denoting element-wise multiplication.

We trained the VAE model by optimizing the following objective:

$$\mathcal{V}_{\text{VAE}}(x, n, \theta_E, \theta_D) =$$
$$= \min_{E,D} \Big[\underbrace{(x - D(E(x,n)))^2}_{\text{Reconstruction error}} + \underbrace{\gamma(1 + \log \sigma^2 - \mu^2 - \sigma^2)}_{\text{KL divergence}} \Big],$$
$$(3)$$

with γ as a scaling factor, and θ_E and θ_D as the parameters of E and D, respectively. Note that we optimized θ_E and θ_D to minimize both the reconstruction error between the input and output data distributions, and the KL divergence between the embedding distribution and the normal $\mathcal{N}(0,1)$ distribution.

This procedure results in a continuous latent space where changes in the embedding vectors are proportional to changes in the input data and vice-versa, effectively retaining semantic knowledge present in the input space.

2.2 Contrastive Training

We assembled a training dataset composed of pairs of patches $x = \{x^{(1)}, x^{(2)}\}$ where each pair x was associated
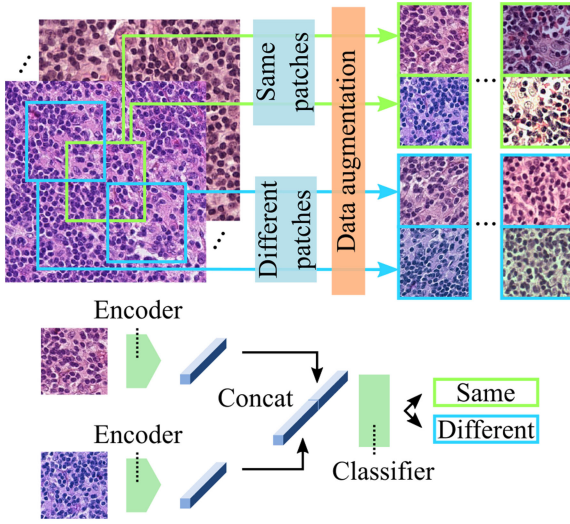
Fig. 3. Contrastive training. Top: pairs of patches are extracted from gigapixel images. Pairs labeled as *same* originate from the same spatial location whereas *different* are extracted from either adjacent locations or different images. Bottom: scheme of a Siamese network trained for binary classification using the previous pairs.

with a binary label y. Each label described whether the patches had been extracted from the *same* or a *different* location in a given gigapixel image, with $y = 1$ and $y = 0$, respectively. We trained a two-branch Siamese network [20] to solve this classification problem (Fig. 3).

We applied heavy data augmentation on all patches as indicated in [26], i.e., rotation, color augmentation, brightness, contrast, zooming, elastic deformation, and added Gaussian noise. Due to the strong augmentation, patches from the *same* location looked substantially different in a highly non-linear fashion while keeping a similar overall structure (semantic), see examples in Fig. 3. Patches from the *different* class were extracted from two distributions: 75 percent of them corresponded to non-overlapping adjacent locations (i.e., neighboring patches) where most of the visual features were shared, and the remaining 25 percent were sampled from different WSIs. Note that we included more of the neighboring *different* pairs to increase the difficulty of the classification task, forcing the network to extract higher-level features. The same data augmentation was applied to other encoders as well to ensure a fair comparison.

2.3 Bidirectional Generative Adversarial Network

The BiGAN setup consists of three networks: a generator G, a discriminator D, and an encoder E (Fig. 4). G maps a latent variable $z \sim \mathcal{N}(0, 1)$ to generated images x'

$$z \sim \mathcal{N}(0,1) \in \mathbb{R}^C \xrightarrow{G} x' \in \mathbb{R}^{P \times P \times 3}, \tag{4}$$

whereas E maps images x sampled from the true data distribution $\mathcal{X}$ to embeddings e

$$x \sim \mathcal{X} \in \mathbb{R}^{P \times P \times 3} \xrightarrow{E} e \in \mathbb{R}^C. \tag{5}$$

During training, the three networks play a minimax game where the discriminator D tries to distinguish between *actual* or *generated* image-embedding pairs, i.e., $\{x, e\}$ and $\{x', z\}$ respectively, while G and E try to fool D by producing
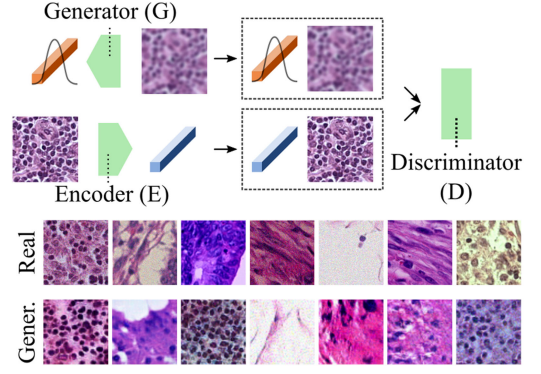

Fig. 4. Adversarial Feature Learning. Top: three networks play a minimax game where the discriminator distinguishes between *actual* or *generated* image-embedding pairs, while the generator and the encoder fool the discriminator by producing increasingly more realistic images and embeddings. Bottom: real and generated patch samples using $C = 128$.

increasingly more realistic images x' and embeddings e closer to $\mathcal{N}(0, 1)$. More formally, we optimized the following objective function:

$$\mathcal{V}_{\mathrm{BiGAN}}(x, z, \theta_G, \theta_E, \theta_D) =$$
$$= \min_{G,E} \max_D \left[\log \left[D(x, \underbrace{E(x)}_{e}) \right] + \log \left[1 - D(\underbrace{G(z)}_{x'}, z) \right] \right], \tag{6}$$

with θ_G, θ_E, and θ_D representing the parameters of G, E, and D, respectively.

The authors of BiGAN theoretically and experimentally demonstrate that G and E learn an approximate inverse mapping function from each other, producing an encoding network E that learns a powerful low-dimensional representation of the image world inherited from G, suitable for downstream tasks such as supervised classification [14].

3 GIGAPIXEL IMAGE ANALYSIS

In this section, we describe a method to train a CNN to predict image-level labels directly from compressed gigapixel images. Furthermore, we analyzed the location of visual cues associated with the image-level labels.

3.1 Feeding a CNN with Compressed Gigapixel Images

We consider a dataset of gigapixel images $\Omega = \{\omega_i\}_{i=1}^Q$ that were compressed into $\Omega' = \{\omega_i'\}_{i=1}^Q$ with $\omega_i' \in \mathbb{R}^{\frac{M}{S} \times \frac{N}{S} \times C}$ using Eq. (1). In order to train a standard CNN on a dataset like Ω', we set the depth of the convolutional filters of the input layer to be equal to the code size C used to compress the images.

We hypothesized that such a CNN can learn to detect highly discriminative features by exploiting two complementary sources of information from Ω': (1) the *global* context encoded within the spatial arrangement of embedding vectors, and (2) the *local* high-resolution information encoded within the features of each embedding vector.

3.2 Preventing Overfitting

Note that in this setup, despite its gigapixel nature, each compressed image ω_i' constitutes a single training data

point. Most public datasets with gigapixel images and their respective image-level labels consist only of a few hundred data points [3], [4], increasing the risk of overfitting. The steps taken to prevent this effect are enumerated below.

First, we extended the training dataset Ω' by taking spatial crops of size $R \times R \times C$ from ω_i', drastically increasing the total number and variability of the samples presented to the CNN [27]. During training, we randomly sampled the location of the center pixel of these crops. During testing, we selected T crops uniformly distributed along the spatial dimensions of ω_i' and averaged the predictions of the CNN across them [27]. Without any loss of generality, we applied this method to histopathology WSIs. As WSIs often contain large empty areas with no tissue, we detected the tissue regions [28] and sampled crops proportionally to the distance to background to accelerate the training, so that areas with higher tissue density were sampled more often. Similarly, test crops were sampled from locations where tissue was present.

The second measure taken to prevent overfitting was a simple augmentation at image level (i.e., 90-degree rotation and mirroring), encoding each image 8 times. This augmentation was carried out during testing as well, averaging the predictions of the CNN across them.

Finally, we designed a CNN architecture aimed at reducing the number of parameters present in the model. In particular, all convolutional layers were set to use depthwise separable convolutions, a type of convolution that reduces the number of parameters while maintaining a similar level of performance [29].

3.3 Visualizing Visual Cues Related to Image-Level Labels

The problem of feature localization is of utmost relevance for gigapixel image analysis: visual cues related to the image-level labels are often sparse and positioned in arbitrary locations within the image. For the purpose of identifying the location of these visual cues, we applied the Gradient-weighted Class-Activation Map (Grad-CAM) algorithm [24] to our trained CNN.

Given a compressed gigapixel image ω', its associated image-level label y, and a trained CNN, Grad-CAM performs a forward pass over ω' to produce a set of J intermediate three-dimensional feature volumes $f_j^{(k)}$, with j and k indicating the jth and kth convolutional layer and feature map, respectively. Subsequently, it computes the gradients of $f_j^{(k)}$ with respect to y for a fixed convolutional layer. It averages the gradients across the spatial dimensions and obtains a set of gradient coefficients $\gamma_j^{(k)}$, indicating how relevant each feature map is for the desired output y. Finally, it performs a weighted sum of the feature maps $f_j^{(k)}$ using the gradient coefficients $\gamma_j^{(k)}$:

$$h^{(k)} = \sum_{j=1}^{J} f_j^{(k)} \gamma_j^{(k)}. \tag{7}$$

We applied the visualization method to the first convolutional layer $(k = 1)$ in order to maximize the heatmap resolution.

4 EXPERIMENTS AND RESULTS

We conducted a series of experiments to evaluate the performance of gigapixel NIC. First, we evaluated NIC in synthetic
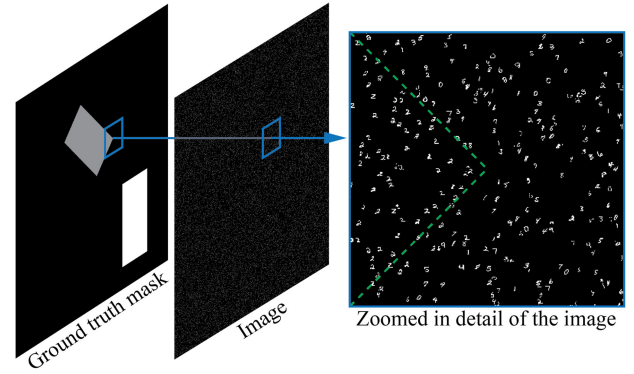

Fig. 5. Example of an image from the synthetic dataset. Left: ground truth mask depicting the *tilted* and *non-tilted* rectangles that simulate lesions in grey and white, respectively. Center: image containing instances of MNIST digits; classes are defined by the rectangles or selected randomly. Right: all digits within the *tilted* rectangle boundary (in green) belong to the same class (number two), which corresponds to the image label as well.

data to gain an understanding of the method and how its hyper-parameters affect performance. Second, we applied the method to several public histopathological datasets.

4.1 Materials

In this work, a synthetic dataset and three histopathology cohorts from different sources were used for supervised and unsupervised training at patch and image level; patients and WSIs were unique across all cohorts.

4.1.1 Synthetic Dataset

We developed and tested NIC with a synthetic dataset that mimicked the task of end-to-end WSI analysis before deploying it with real WSIs. As a substitute for WSIs, a set of images $T = \{t_i\}$ with $t_i \in \mathbb{R}^{A \times B}$ were used, each one associated with a dense pixel-level ground truth mask $M = \{m_i\}$, where $m_i \in \mathbb{R}^{A \times B}$, and an image-level scalar label $Y = \{y_i\}$.

To emulate global patterns in the images (e.g., tumor lesions), we defined two rectangles within each mask placed at random locations and characterized by their own orientation: one was either vertically or horizontally oriented (*non-tilted*); the other was tilted either 45 or 135 degrees (*tilted*). Each rectangle was associated to a randomly selected MNIST [30] digit class. To emulate local patterns (e.g., cells), instances of MNIST digits were placed throughout the images at random locations. The class of these instances was determined by their spatial position, i.e., belonging to a certain rectangle class if placed within the boundaries of a rectangle or otherwise randomly selected. The label of each image was defined by the class of the *tilted* rectangle, with the *non-tilted* rectangle acting as a distraction. See Fig. 5 for an example image.

Note that, in order to solve this classification task, NIC had to detect the *tilted* rectangle and its class without access to the ground-truth masks. Moreover, the method must combine local and global information, i.e., exploiting the local features that identify digit instances' classes while recognizing their global spatial arrangement to detect the orientation of the rectangle.

We downsampled MNIST digits to 9×9 pixels, defining a patch size $P = 9$ and stride $S = 9$ pixels. WSIs are typically 50000×50000 pixels in size, with patch sizes of 128×128 pixels covering structures composed of a few cells. We mimicked this image-patch ratio by using an image size of $A = B = 3600$ pixels, and inserted 25,920 digit instances per image (0.2 percent of the total possible locations). Rectangle size randomly ranged from 1,800 pixels to 36 pixels (long side). This reduced image size enabled us to run more thorough experiments than what we could do with histopathological data.

A total of 50,000 images with balanced labels were created across the 10 digit classes: 2,500 to generate patches to train the encoders, 22,500 to train the NIC CNN (with 75 and 25 percent for training and validation), and 25,000 as an independent test set for the NIC CNN.

4.1.2 Camelyon16 Histopathology Dataset

The *Camelyon16* [4] dataset is a publicly available multicenter cohort that consists of 400 sentinel lymph node H&E WSIs from breast cancer patients. Reference standard exists in two forms: fine-grained annotations of metastatic lesions and image-level labels indicating the presence of tumor metastasis in each slide. Sixty WSIs from the original training set were set aside to train encoders at patch level. The remaining WSIs were combined with the original test set (n=340) to train and evaluate a classification model using image-level labels only.

4.1.3 TUPAC16 Histopathology Dataset

The *TUPAC16* [3] dataset was used, consisting of 492 H&E WSIs from invasive breast cancer patients. It is a publicly available cohort with WSIs from The Cancer Genome Atlas [31] where each WSI is associated with a tumor proliferation speed score, an objective measurement that takes into account the RNA expression of 11 proliferation-associated genes [32]. We set aside 40 WSIs from this set to train encoders at patch level. The remaining WSIs (n=452) were used to train and evaluate a regression model using image-level labels only. Additionally, 321 test WSIs with no public ground truth available were used to perform an independent evaluation.

4.1.4 Rectum Histopathology Dataset

The *Rectum* dataset is a publicly available set of 74 H&E WSIs from rectal carcinoma patients [33]. Manual annotations of 9 tissue classes were made by an expert: blood cells, fatty tissue, epithelium, lymphocytes, mucus, muscle, necrosis, stroma, and tumor. The slides were randomized and organized into ten equal partitions at patient level, five of which were used for training, one for validation, and four for testing. This dataset was used to train and evaluate encoders at patch level only. We extracted a balanced distribution of 15K, 852, and 4K patches per class from the training, validation, and test slides, respectively.

4.1.5 Data Preparation

Regarding the synthetic dataset, one million pairs of patches were extracted to train the encoders, augmented with scaling and elastic deformation. To avoid creating a dataset of empty patches, the probability of sampling a patch containing a white pixel was twice of that of an empty patch.

All WSIs in this study were preprocessed with a tissue-background segmentation algorithm [28] in order to exclude areas not containing tissue from the analysis. Furthermore, all images were analyzed at 0.5μ m/pixel resolution.

A set of patch datasets were assembled to train and evaluate each of the encoding networks described in Section 2 using the set of images that we set aside from each cohort: 60 WSIs from *Camelyon16*, 40 from *TUPAC16*, and all from *Rectum*. Each of these subcohorts were divided into training, validation, and test partitions.

The *contrastive dataset* was created by extracting an equal amount of patches from each source (i.e., *Camelyon16*, *TUPAC16*, and *Rectum*) and merged into 50,000 and 25,000 patch pairs for training and validation, respectively. The *non-contrastive dataset* was then created by randomizing all individual patches within the *contrastive dataset*.

The *supervised-tumor dataset* was created by extracting 50,000, 10,000, and 50,000 patches from the set of 60 *Camelyon16* WSIs for training, validation, and testing, respectively. Finally, the *supervised-tissue dataset* consisted of the *Rectum* training, validation, and test sets containing 131,000, 8,000, and 35,000 patches, respectively. Note that the patches in the *supervised-tumor dataset* and *supervised-tissue dataset* test sets did not undergo any augmentation. The fine-grained tumor annotations were used to sample a balanced distribution of tumor and non-tumor patches in the *supervised-tumor dataset* and 9-class patches in the *supervised-tissue dataset*.

4.2 Experimental Results on Synthetic Data

The *contrastive* encoder was trained using the pairs of patches described in Section 4.1.5. The *VAE* and *BiGAN* encoders were subsequently trained using these same patches, concatenating and shuffling them along the pair dimension. Finally, a *supervised* encoder was trained with MNIST digits to serve as an oracle feature extractor. Once the encoders were trained, all images were encoded to produce a different embedded representation for each encoding configuration. Network architectures and training protocols are detailed in the Supplementary Material, available online, accompanying this paper.

We explored different values for the method hyper-parameters (e.g., code size and stride) using the synthetic data, and evaluated the accuracy of each resulting CNN in the independent test set. We analyzed how this performance was affected by the size of the simulated lesion, i.e., the size of the *tilted* rectangle. Results are summarized in Fig. 6. Overall, the *contrastive* encoder achieved the best performance among the unsupervised techniques, very close to that of the oracle, followed by the *VAE* and *BiGAN* encoders. This trend was maintained when analyzing the impact of the lesion size. We found out that the method's performance degraded quickly when the size of the target lesion was smaller than 10 percent of the image size (see Fig. 6a).

Additionally, the performance impact of the code size used to compress the images was assessed (Fig. 6b). It was observed that larger code sizes generally improved performance, a result that was more evident for less accurate
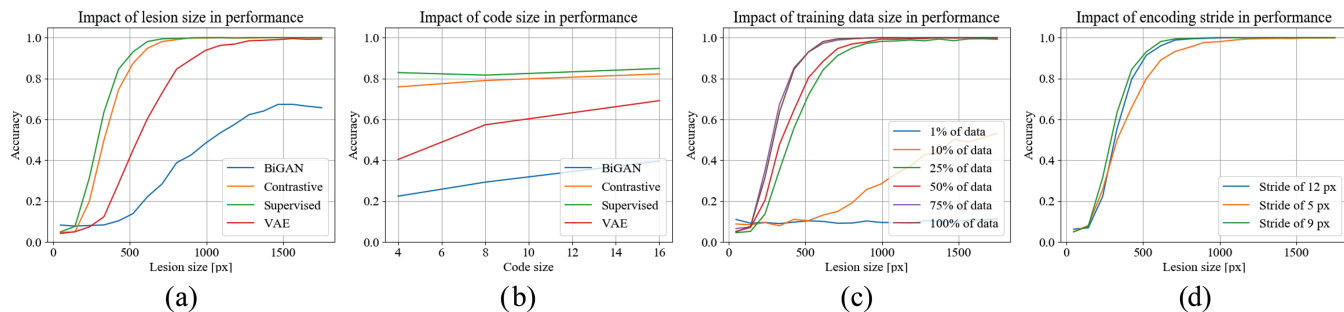
Fig. 6. Experimental results with synthetic data and image-level labels. Default hyper-parameter choice unless specified otherwise is: *supervised* encoder, code size 16, stride 9 pixels, and usage of 100 percent of training data.

encoding methods like *VAE* and *BiGAN*. Subsequently, different stride values were tested using the oracle encoder and a code size of 16: it was found that a smaller stride, producing embedded images with larger spatial resolution, resulted in hampered performance (Fig. 6d). Finally, the impact of training data size in performance was analyzed using the oracle encoder with code size 16 and stride 9 (Fig. 6c). These results indicate that NIC required in the order of thousands of images to perform well, a requisite that is rarely met in real histopathological datasets.

In our last experiment, we applied Grad-CAM to visualize the regions of the input images that were responsible for the CNN prediction (see Fig. 7). Remarkably, the network seemed to be able to discern between background noise and the rectangular patterns. Upon visual inspection, the CNN generally focused on the *tilted* rectangle, the one responsible for the image-level label. We applied a simple general-purpose post-processing routine to denoise the heatmaps and reject spurious activity. We measured the Jaccard similarity coefficient per image between the post-processed heatmap and the ground truth maps, and obtained 0.612 on average across test images.

4.3 Training of Encoders

Due to the computationally expensive nature of experimenting with gigapixel WSIs, we only tested a subset of the hyper-parameters that we explored with synthetic data. We

selected their values using the following heuristics. We used $P = 128$, a common patch size used in the Computational Pathology literature [28], with a stride of the same size $S = 128$ to perform non-overlapping patch sampling. We selected $R = 400$ to obtain crops corresponding to typical sizes of gigapixel WSIs ($50,000 \times 50,000$ pixels) and $T = 10$ as done in the literature [27]. Finally, we selected $C = 128$ to perform our experiments using a single GPU. Network architectures and training protocols are detailed in the Supplementary Material, available online.

We trained the *contrastive* encoder with the *contrastive dataset*, and the *VAE* and *BiGAN* models with the *non-contrastive dataset*. Note that these datasets contained the exact same image patches, ensuring a fair comparison among encoders. No manual annotations were required in this process. We trained a supervised baseline encoder for breast tumor classification using the *supervised-tumor dataset*, and a supervised baseline encoder for rectum tissue classification using the *supervised-tissue dataset*.

It is widely recognized that color-based features can be very informative in histopathology image analysis [34], [35], [36]. Therefore, we included an additional encoding function to capture color information from the raw input by averaging the pixel intensity across spatial dimensions from input RGB patches. It provided a simple yet effective baseline to compare with more sophisticated encoding mechanisms.

This entire training process resulted in 6 encoding networks used in subsequent experiments: the *mean-RGB* baseline, *VAE* encoder, *contrastive* encoder, *BiGAN* encoder, *supervised-tumor* baseline, and *supervised-tissue* baseline.

4.4 Comparing Encoding Performance

Due to the lack of a common evaluation methodology for unsupervised representation learning, we compared the performance of these 6 encoders when used as fixed feature extractors for related supervised classification tasks. We defined two tasks: (1) discerning between tumor and non-tumor patches on the *supervised-tumor dataset* (*Task-1*), and (2) performing 9-class tissue classification on the *supervised-tissue dataset* (*Task-2*). For each task, we trained an MLP on top of each encoder with frozen weights and reported the accuracy metric for each test set.

Results in Table 1 highlight several observations. First, *VAE*, *contrastive*, and *BiGAN* performed better than the lower baseline for both *Task 1* and *Task 2*, stressing their ability to describe complex patterns beyond simple features related to color intensity. Second, the *VAE* encoder obtained a higher performance than the *contrastive* one, particularly
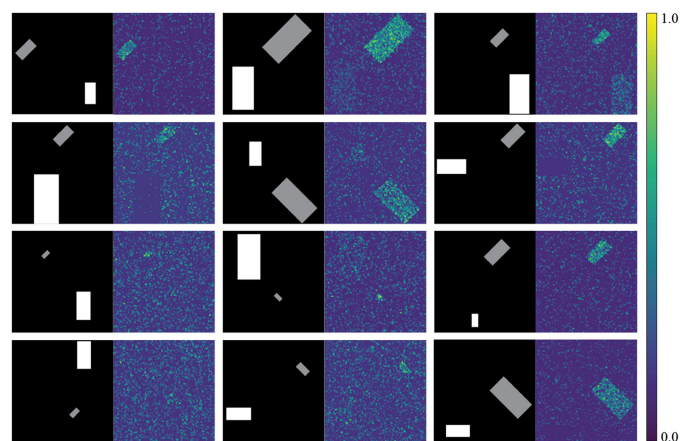

Fig. 7. Grad-CAM visualization applied to randomly selected synthetic test images. Left images within the pairs correspond to the ground truth masks (unseen by the model), and right ones to the saliency heatmaps. Note that areas corresponding to the grey *tilted* rectangles (responsible for the image-level labels) are highly salient with respect to the rest of the image.

TABLE 1
Patch-Level Classification Performance (Accuracy)

Encoder	Camelyon Tumor	Rectum										
		Blood	Fat	Epith	Lymph	Mucus	Muscle	Necro	Strom	Tumor	Global	
VAE	0.799(0.004)	0.602(0.034)	0.735(0.154)	0.556(0.006)	0.811(0.018)	0.623(0.125)	**0.823(0.014)**	0.170(0.018)	**0.768(0.008)**	0.667(0.000)	0.639(0.010)	
Contrastive	0.789(0.004)	0.304(0.018)	**0.966(0.003)**	0.502(0.005)	0.850(0.014)	0.240(0.011)	0.609(0.006)	0.140(0.010)	0.595(0.005)	0.476(0.014)	0.520(0.002)	
BiGAN	**0.806(0.022)**	**0.738(0.034)**	0.879(0.059)	**0.627(0.000)**	**0.899(0.008)**	**0.802(0.055)**	0.796(0.002)	**0.769(0.021)**	0.601(0.066)	**0.770(0.010)**	**0.765(0.013)**	
Mean-RGB	0.772(0.001)	0.736(0.000)	0.635(0.355)	0.202(0.049)	0.385(0.068)	0.720(0.270)	0.904(0.008)	0.030(0.028)	0.668(0.039)	0.252(0.000)	0.504(0.022)	
Sup.-tumor	0.855(0.001)	0.578(0.090)	0.896(0.005)	0.400(0.007)	0.981(0.004)	0.868(0.021)	0.507(0.061)	0.494(0.049)	0.467(0.027)	0.618(0.019)	0.646(0.008)	
Sup.-tissue	0.800(0.006)	0.835(0.003)	0.958(0.008)	0.832(0.029)	0.935(0.010)	0.937(0.026)	0.940(0.002)	0.906(0.005)	0.863(0.009)	0.934(0.002)	0.904(0.000)	

Task-1 *and* Task-2 *in the text refer to columns* Camelyon-Tumor *and* Rectum-Global. *Reporting mean and standard deviation using two random weight initializations.*

for *Task 2*. Third, the *BiGAN* encoder achieved the best performance among all the unsupervised methods, with a relatively large margin for the more complex *Task 2* with respect to the runner-up *VAE* model. Furthermore, the *BiGAN* encoder obtained the best result for 5 out of 9 classes in *Task 2*, and it achieved the first or second best result for 8 out of 9 classes among the unsupervised models. Remarkably, *BiGAN* succeeded at classifying patches from challenging tissue classes such as blood cells and necrotic tissue.

4.5 Predicting the Presence of Metastasis at Image Level

In this experiment, we trained a CNN to perform binary classification on compressed gigapixel WSIs from the *Camelyon16* cohort, identifying the presence of tumor metastasis using image-level labels only. Due to the limited amount of images in this cohort (340 WSIs), we divided the dataset into four equal-sized partitions and performed four rounds of cross-validation using two partitions for training, one for validation and one for testing, rotating them in each round. We trained a different CNN classifier for each encoder, i.e., *mean-RGB*, *VAE*, *contrastive*, *BiGAN*, and the upper baseline *supervised-tumor*. We reported the area under the receiver operating characteristic (AUC) on three evaluation sets.

The first evaluation set (*All*) concatenated all samples in each of the hold-out partitions. Note that each hold-out partition was evaluated by a different CNN that had never seen the data. The second evaluation set (*Test*) was a subset of *All* that matched the official test set of the *Camelyon16* Challenge, used for comparison with the public leaderboard. The third evaluation set (*Macro*) used the same WSIs as in *Test* but considering only those that presented a macro metastasis as positive labels, i.e., a tumor lesion larger than

2 mm. The macro labels were only available for the *Camelyon16* test set. The *Macro* set was relevant to evaluate how the method performed with lesions visible at low resolution.

Results in Table 2 demonstrate that the method presented in this work is an effective technique for gigapixel image analysis using image-level labels only. Regarding the *All* evaluation set, *BiGAN* achieved a remarkable performance of 0.716 AUC, with a relative difference from the *supervised* baseline of only 6 percent. The *contrastive* and *VAE* models also surpassed the lower baseline, but obtained substantially lower performance scores compared to *BiGAN*. Regarding the *Test* set, the *BiGAN* encoder obtained a lower performance of 0.674 AUC. In the *Macro* set, the performance gap between the *supervised* baseline and the *BiGAN* encoder increased substantially from 0.095 to 0.184. The state-of-the-art in *Camelyon16* obtained 0.9935 AUC in the *Test* set using accurate pixel-level annotations to train their model.

Additionally, we analyzed the performance of our method as a function of the lesion size in the *All* test set. The lesion size is a measurement determined by pathologists taking the distribution of tumor cell clusters within a WSI into account. Since this annotation was not available for all WSIs, we approximated it by computing the radius of an hypothetical circle with an area composed of all pixels annotated as tumor in each WSI. Results in Fig. 8 indicated
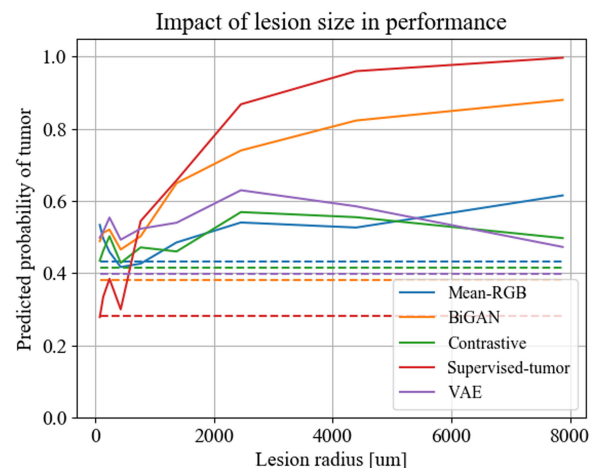
TABLE 2
Predicting the Presence of Metastasis at WSI Level (AUC)

Encoder	All	Test	Macro
VAE	0.661(0.007)	0.671(0.008)	0.634(0.003)
Contrastive	0.608(0.001)	0.651(0.016)	0.606(0.012)
BiGAN	**0.725(0.009)**	**0.704(0.030)**	**0.720(0.010)**
Mean-RGB	0.582(0.006)	0.578(0.016)	0.585(0.014)
Supervised-tumor	0.760(0.002)	0.771(0.002)	0.914(0.000)

Reporting mean and standard deviation using two random weight initializations.

Fig. 8. Experimental results with respect to lesion size in *Camelyon16 all* test set using multiple encoders. Solid lines: average probability of samples with positive labels; dashed lines: average probability of samples with negative labels (no lesion).
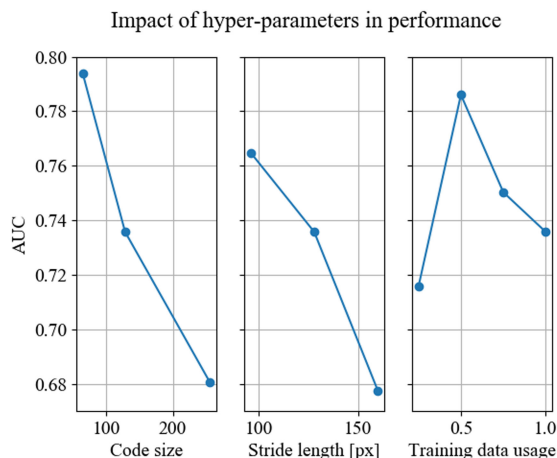
Fig. 9. Hyper-parameter value analysis performed in *Camelyon16* data using the *supervised* encoder. Evaluated on unseen images from the first data partition out of the 4-fold cross-validation sets. Left: varying code size using a fix stride of 128 pixels; center: varying stride while using a fix code size of 128 elements; and right: varying the number of WSIs used during training.

TABLE 3
Predicting Tumor Proliferation Speed at WSI Level
(Spearman Corr.)

Encoder	All	Test
VAE	0.419(0.004)	-
Contrastive	0.390(0.006)	-
BiGAN	**0.522(0.001)**	0.558(0.001)
Mean-RGB	0.238(0.020)	-
Supervised-tumor	0.427(0.014)	-

Reporting mean and standard deviation using two random weight initializations.

that our method's performance degraded with small tumor lesions across most encoders, in line with the results obtained with synthetic data. Furthermore, we experimented with different hyper-parameters such as code size, stride, and training data size using the *supervised* encoder (Fig. 9). We found that performance improvements might be gained from careful hyper-parameter tuning of the code size and stride parameters. Moreover, there seemed to be a weak but positive correlation between model performance and training data size.

4.6 Predicting Tumor Proliferation Speed at Image Level

In this experiment, we trained a CNN to perform a regression task on compressed gigapixel WSIs from the *TUPAC16* cohort, predicting the tumor proliferation speed based on gene-expression profiling. We performed 4-fold cross-validation as in the previous experiment, and reported the Spearman correlation between the predicted and the true scores of two evaluation sets.

The first evaluation set (*All*) concatenated all samples in each of the hold-out partitions. The second evaluation set

(*Test*) matched the test set used in the *TUPAC16* Challenge, whose ground truth is not public. Using the encoder that obtained the highest performance, we evaluated each WSI in *Test* four times using each of the CNNs trained during cross-validation and submitted the average score per slide. Our predictions were independently evaluated by the challenge organizers, ensuring a fair and independent comparison with the state of the art.

The results in Table 3 showed that *BiGAN* achieved the highest performance with a 0.521 Spearman correlation. Remarkably, this score was superior to that of any other unsupervised or supervised encoder. In addition, we obtained a score of 0.557 on the *TUPAC16* Challenge test set, superior to the state-of-the-art for image-level regression with a score of 0.516. Note that the first entry of the leaderboard used an additional set of manual annotations of mitotic figures, thus it cannot be compared with our setup.

4.7 Visualizing *where* the Information Is Located

We conducted a qualitative analysis on the trained CNNs to locate the spatial position of visual cues relevant in predicting the image-level labels. We applied the Grad-CAM algorithm to the CNNs trained for both tasks at image level. For the tumor metastasis prediction task, we compared the saliency maps with fine-grained manual annotations. Figs. 10 and 11 include the results for a few samples; the results for the remaining WSIs can be found in the Supplementary Material, available online. Note that each WSI was evaluated by a CNN that had not yet seen the image (hold-out partition).

Fig. 10 shows that the *mean-RGB* baseline model lacked the ability to focus on specific tissue regions, suggesting
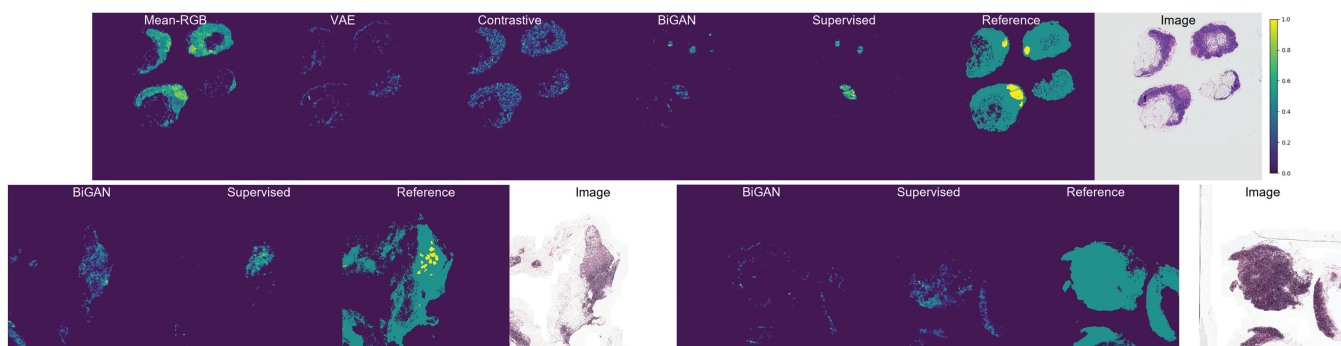

Fig. 10. Grad-CAM visualization applied to several WSIs from *Camelyon16*. Top: the first five images represent the saliency maps for CNNs trained with 5 different encoders, respectively. The sixth and seventh images are the reference standard (manual annotations) and RGB thumbnail of the WSI, respectively. Dark blue represents low saliency, whereas yellow indicates high saliency. Bottom-left: failure case where the *BiGAN* model failed to recognize the tumor area. Bottom-right: failure case where the *BiGAN* model attended to a region with no tumor cells.
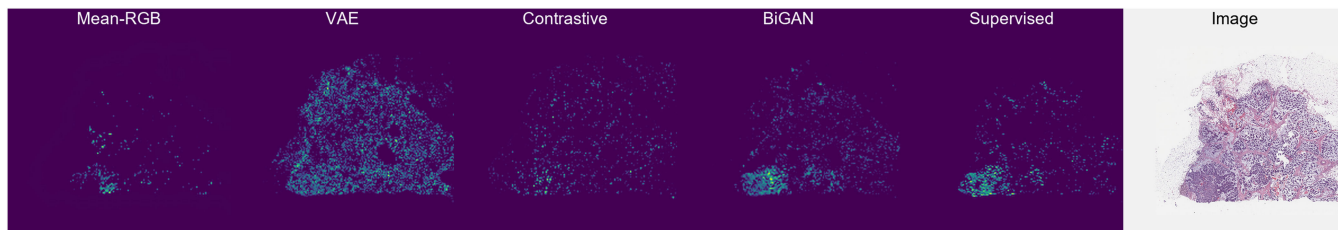
Fig. 11. Grad-CAM visualization applied to a sample case from *TUPAC16*. The first five images represent the saliency maps for CNNs trained with five different encoders, respectively. The last image is an RGB thumbnail of the WSI. Dark blue represents low saliency, whereas yellow indicates high saliency.

that it was unable to learn discriminative features from image-level labels. The *VAE* and *contrastive* models exhibited a suboptimal behavior, scattering attention all over the image. Remarkably, the *BiGAN* model seemed to focus on tumor regions only, discarding empty areas, fatty tissue, and healthy dense tissue. It showed a strong discriminative power to discern between tumor and non-tumor regions, even though the CNN had access to image-level labels only. For completeness, we also included the *supervised-tumor* baseline that also exhibited a focus on tumor regions. Nevertheless, these heatmaps are often difficult to interpret and cannot be used for a more quantitative analysis. Failure cases can be seen in the bottom part of Fig. 10, where the CNN highlighted non-tumorous regions.

Regarding Fig. 11, a similar trend to the one found in the previous task was observed for all encoders: the *BiGAN* model focused on very specific regions of the WSIs that seemed compatible with active tumor regions. The *supervised-tumor* baseline focused on irrelevant areas, in line with its poor performance for this task.

5 DISCUSSION

Our experimental results support the hypothesis that visual cues associated with weak image-level labels can be exploited by our method, integrating information from global structure and local high-resolution visual cues. Furthermore, we have shown that this methodology is flexible and completely label-agnostic, delivering relevant results for both classification and regression tasks in synthetic as well as histopathological data. It emerges as a promising strategy to tackle the analysis of more challenging image-level labels that are closely related to patient outcome, e.g., overall survival and recurrence-free survival. Gigapixel NIC paves the way for leveraging existing computer vision algorithms that could not be applied in the gigapixel domain until now, such as image captioning (useful to generate written clinical reports), visual question answering, image retrieval (to find similar pathologies), anomaly detection, and generative modeling [37], [38], [39], [40], [41].

A key assumption in our method was that high-resolution image patches could be represented by low-dimensional highly compressed embedding vectors. We analyzed several unsupervised strategies to achieve such a compression and found that the *BiGAN* encoder, trained using adversarial feature learning, was superior to all other methods across all experiments with histopathological data. We believe that this relative improvement with respect to the *VAE* and *contrastive* methods is explained by intrinsic algorithmic differences among the methods. In particular, the *VAE* model relies on

minimizing the MSE objective, which is a unimodal function that fails to capture high-level semantics; it focuses on reconstructing low-level pixel information instead, wasting embedding capacity. On the other hand, the *contrastive* encoder uses the embedding capacity more efficiently, but its performance is driven by the design of the hand-engineered contrastive task. Remarkably, the *BiGAN* model learns an encoder that fully automatically inverts a complex mapping between the latent space and the image space. By doing so, the encoder benefits from all the high-level features and semantics already discovered by the generator, producing very effective discriminative embedding vectors. Furthermore, *BiGAN* achieved the best classification accuracy on the challenging blood, mucus, and necrotic tissue classes that rarely appear in the *Camelyon16* and *TUPAC16* WSIs. We hypothesize that the adversarial method can model these rare data modes more effectively than the *contrastive* or *VAE* approaches. Nevertheless, we believe that the choice of encoder may be data-dependent, since the *contrastive* encoder outperformed the other approaches in the synthetic dataset.

We trained a CNN to predict the breast tumor proliferation speed based on gene-expression profiling, a label associated with unknown visual cues. Our method succeeded in finding and exploiting these patterns in order to predict expected tumor proliferation speed, surpassing the current state-of-the-art for image-level based methods. This shows that our method constitutes an effective solution to deal with gigapixel image-level labels with unknown associated visual cues. Moreover, our method could be used in future works to effectively mine datasets with thousands of gigapixel images [11]; other automatically generated labels from immunohistochemistry, genomics, or proteomics can be targeted, and visual patterns beyond the knowledge of human pathologists may be discovered.

For the first time, the regions of a gigapixel image that a trained CNN attends to when predicting image-level labels were visualized, and the effect of different encoding methods was compared. We discovered that only the CNNs trained with images compressed with the *BiGAN* encoder and the *supervised-tumor* baseline were able to attend to regions of the image where tumor cells were present. The fact that the *BiGAN* model simultaneously learned to delimit metastatic lesions and identify tumor features within the patch embeddings validates our hypothesis that CNNs are an effective method for analyzing gigapixel images, i.e., since they can exploit both global and local context.

We targeted the presence of tumor metastasis in breast lymph nodes and showed that the *BiGAN* setup performed similarly to the *supervised* baseline. However, our best-

performing algorithm was still inferior to that of the *Camelyon16* leadingboard (0.9935 AUC using accurate pixel-level annotations). This performance gap is likely due to two factors. First, the majority of the images marked as positive contain tumor lesions comprised of only a few tumor cells (i.e., micro-metastasis), becoming almost undetectable with the compression setup tested in this work (see Fig. 8). Second, the lack of training data (only a few hundred training images) may lead the CNN into the overfitting regime.

We acknowledge several limitations of our method. For one, it requires a substantial amount of I/O throughput and storage due to the need to write compressed WSI representations to disk before training, and repetitively read them to assemble mini-batches during training. This computational burden prevented us from performing a wide hyper-parameter value search, which may have resulted in a suboptimal parameter selection. Second, it was also observed that the method's performance was proportional to lesion size. In particular, it struggled to detect micro-metastasis in *Camelyon16* data, i.e., tumor lesions smaller than 2 mm, limiting the applicability of NIC to tasks with large lesions.

This method can be extended in multiple ways. More sophisticated encoders may improve the low-dimensional representation of the image patches [16], [42]. Incorporating attention mechanisms may make it easier for the CNN to attend to relevant regions for the image-level labels [43], improving the detection of small lesions. Finally, gradient checkpointing [44] could be used to backpropagate the training signal from the image-level labels towards the encoder weights.

6 CONCLUSION

Our method for gigapixel neural image compression was able to distill relevant information into compact image representations. The fact that a CNN could be trained using these alternative learned representations opens opportunities to use other methods: gigapixel images are no longer considered as low-level pixel arrays, but operate in a higher level of abstraction. In this work, we showed examples of classification, regression, and visualization performed in a latent space learned by a neural network. These positive results enable performing more advanced gigapixel applications in the latent space, such as data augmentation, generative modeling, content retrieval, anomaly detection, and image captioning.

REFERENCES

[1] G. Litjens, et al., "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

[2] L. Zhang, et al., "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[3] M. Veta, et al., "Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge," *Med. Image Anal.*, vol. 54, pp. 111–121, 2019.

[4] B. Ehteshami Bejnordi, et al., "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *J. Amer. Med. Assoc.*, vol. 318, no. 22, pp. 2199–2210, 2017.

[5] K. Sirinukunwattana, et al., "Gland segmentation in colon histology images: The glas challenge contest," *Med. Image Anal.*, vol. 35, pp. 489–502, 2017.

[6] X. Wang, et al., "Weakly supervised learning for whole slide lung cancer image classification," *Med. Imag. Deep Learn.*, 2018.

[7] M. Ilse, et al., "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2132–2141.

[8] M. Combalia, et al., "Monte-Carlo sampling applied to multiple instance learning for whole slide image classification," *Med. Imag. Deep Learn.*, 2018.

[9] J. M. Tomczak, et al., "Histopathological classification of precursor lesions of esophageal adenocarcinoma: A deep multiple instance learning approach," *Med. Imag. Deep Learn.*, 2018.

[10] L. Hou, et al., "Patch-based convolutional neural network for whole slide tissue image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2424–2433.

[11] N. Coudray, et al., "Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning," *Nature Med.*, vol. 24, no. 10, 2018, Art. no. 1559.

[12] I. Goodfellow, et al., "Convolutional networks," in *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, ch. 9.

[13] L. Theis, et al., "Lossy image compression with compressive autoencoders," in *Proc. Int. Conf. Learn. Representations*, 2017.

[14] J. Donahue, et al., "Adversarial feature learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[15] V. Dumoulin, et al., "Adversarially learned inference," in *Proc. Int. Conf. Learn. Representations*, 2017.

[16] A. van den Oord, et al., "Representation learning with contrastive predictive coding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018.

[17] A. van den Oord, et al., "Neural discrete representation learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6306–6315.

[18] D. P. Kingma, et al., "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, 2013.

[19] G. Koch, et al., "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, 2015, vol. 2.

[20] I. Melekhov, et al., "Siamese network features for image matching," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 378–383.

[21] A. Hyvarinen, et al., "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3765–3773.

[22] I. Goodfellow, et al., "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[23] X. Chen, et al., "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2180–2188.

[24] R. R. Selvaraju, et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[25] D. Tellez, et al., "Gigapixel whole-slide image classification using unsupervised image compression and contrastive training," *Med. Imag. Deep Learn.*, 2018.

[26] D. Tellez, et al., "Whole-slide mitosis detection in H&E breast histology using PHH3 as a reference to train distilled stain-invariant convolutional networks," *IEEE Trans. Med. Imag.*, vol. 37, no. 9, pp. 2126–2136, Sep. 2018.

[27] A. Krizhevsky, et al., "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] P. Bándi, et al., "Comparison of different methods for tissue segmentation in histopathological whole-slide images," in *Proc. IEEE 14th Int. Symp. Biomed. Imag.*, 2017, pp. 591–595.

[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[30] Y. LeCun, et al., "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[31] J. N. Weinstein, et al., "The cancer genome atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, no. 10, 2013, Art. no. 1113.

[32] T. O. Nielsen, et al., "A comparison of PAM50 intrinsic subtyping with immunohistochemistry and clinical prognostic factors in tamoxifen-treated estrogen receptor–positive breast cancer," *Clinical Cancer Res.*, vol. 16, pp. 5222–5232, 2010.

[33] F. Ciompi, et al., "The importance of stain normalization in colorectal tissue classification with convolutional networks," in *Proc. IEEE 14th Int. Symp. Biomed. Imag.*, 2017, pp. 160–163.

[34] O. Sertel, et al., "Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development," *Pattern Recognit.*, vol. 42, no. 6, pp. 1093–1103, 2009.

[35] A. Tabesh, et al., "Multifeature prostate cancer diagnosis and gleason grading of histological images," *IEEE Trans. Med. Imag.*, vol. 26, no. 10, pp. 1366–1378, Oct. 2007.

[36] J. Kong, et al., "Image analysis for automated assessment of grade of neuroblastic differentiation," in *Proc. 4th IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2007, pp. 61–64.

[37] H.-C. Shin, et al., "Learning to read chest X-rays: Recurrent neural cascade model for automated image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2497–2506.

[38] S. A. Hasan, et al., "Overview of the Image CLEF 2018 medical domain visual question answering task," in *Proc. Conf. Labs Eval. Forum (Working Notes)*, 2018.

[39] Y. Anavi, et al., "A comparative study for chest radiograph image retrieval using binary texture and deep learning classification," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 2940–2943.

[40] T. Schlegl, et al., "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[41] Q. Yang, et al., "Low dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.

[42] M. Caron, et al., "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[43] S. Jetley, et al., "Learn to pay attention," in *Proc. Int. Conf. Learn. Representations*, 2018.

[44] T. Chen, et al., "Training deep nets with sublinear memory cost," *arXiv:1604.06174*, 2016.

[45] F. Chollet, et al., "Keras," 2015. [Online]. Available: https://keras.io

David Tellez received the BSc and MSc degrees in telecommunication engineering from the University of Seville, Spain, in 2014. He is working toward the PhD degree in the Computational Pathology Group of Jeroen van der Laak, Radboud University Medical Center, The Netherlands. His research interests include computer vision and medical imaging.

Geert Litjens received the PhD thesis on "Computerized detection of prostate cancer in multi-parametric MRI" at the Radboud University Medical Center. He is currently assistant professor in computation pathology with the same institution. His research focuses on applications of machine learning to improve oncology diagnostics.

Jeroen van der Laak received the MSc degree in computer science and the PhD degree from Radboud University Medical Center, in Nijmegen, The Netherlands, where he currently is associate professor of computational pathology. His research focuses on deep learning algorithm development and validation for improved pathology diagnostics.

Francesco Ciompi received the MSc degree in computer vision and artificial intelligence from the Autonomous University of Barcelona, in 2008, and the PhD degree Cum Laude from the University of Barcelona, in July 2012. Since 2013, he is part of the Diagnostic Image Analysis Group of Radboud University Medical Center. He is assistant professor of computational pathology, working on deep learning for automatic analysis of digital pathology whole-slide images.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.