Scaling mega-pixel images classification via attention sampling: lymphoma subtype case study

Alberto Sinigaglia - 2044712
alberto.sinigaglia@studenti.unipd.it

***Abstract*—Lymphoma subtype classification is a very important task as it strongly determines the treatment that the patient should get; however, it requires the analysis of high resolution images by experts to be very accurate. Advances in Machine and Deep learning created new tools suited for tasks like this. Unfortunately, the high resolution images required for the classification are still prohibitive for the current state-of-the-art hardware and architectures, and current solutions do not take the hardware scaling factor into account. In this paper we propose to use an attention-sampling mechanism that enables the use of much smaller architectures, as well as orders of magnitude smaller usage of memory and FLOPS.**

The proposed technique will have comparable performances to state-of-the-art models, but will use more than 10x less parameters and thus much less memory and floating point operations.

Thanks to the structure of the model, we also gain explainability over the classification as a by-product. The model also creates a general framework for other datasets, as the proposed attention will also work in cases where the information is local and very focused on the image, thus rescaling and random cropping would not be effective.

***Index Terms*—Deep learning, Reinforcement learning, Convolutional Neural Networks, Attention Sampling**

I. Introduction

Recent advances in Artificial Intelligence, in particular within the field of Deep Learning, allowed the application of the field's techniques to a more broad range of problems, one of which is automated classification for medical treatment and cancer discovery.

Recently, governments, in order to prevent cancer development in their residents, have started campaigns of screening, where people have periodic checks and the images are analysed to catch early development of some types of cancers.

However, this requires experts to analyze thousands of images from people, which does not scale as soon as these prevention techniques are applied to an increasing number of diseases or cancer types.

Deep learning techniques are known to achieve human-like, if not better, performances in image classification and other tasks, given enough labeled examples.

The downside of those new techniques is the price, which initially might be less than the cost of experts on a small scale, but as the scale increases, they will require powerful hardware which might be cost-prohibitive, such as for state-wide screening. It's thus necessary to find ways to scale those models for those magnitudes, and current state-of-the-art models do not consider this.

While the final score is impressive and outperforms humans, the cost causes a problem in the application of this research, since the calculations are still too expensive for current hardware.

Lymphoma is a type of cancer that accounts for 4% of cancer diagnoses and has affected half a million people worldwide, making it one of the most common types of cancer.

This work aims to develop a technique to identify three of the most famous types of lymphoma: Chronic Lymphocytic Leukemia (CLL), Follicular Lymphoma (FL), and Mantle Cell Lymphoma (MCL), by using mega pixel images from the patients check-ups. In order to allow large scale deployment and usage without requiring large scale infrastructures, this work requires minimal usage of resources both on the memory and computational side.

This paper is structured as follows: In Section II, we describe the state-of-the-art; the system and data models are respectively presented in Section III, in Section IV, there is the derivation for the gradient for the attention network, followed by Section V, which explains how to cope with the exploration/exploitation dilemma. The detailed explanation of the full model is carried out in Section VI followed by Section VII with the training details. In Sections' VIII, IX and X, we will report the results and the analysis of the results, and finally in Sections' XI and XII, there are the conclusions about the reported work.

II. Related Work

In the following section, we will list some of the most relevant advances in this topic present in the literature.

A. Machine Learning related

Initial attempts on this field exploited machine learning models to classify the images, for example used by Orlov et al. [1] in *"Automatic Classification of Lymphoma Images With Transform-Based Global Features"* and by Ferjaoui et al [2] in *"Lymphoma Lesions Detection from Whole Body Diffusion-Weighted Magnetic Resonance Images"*, exploiting models such as Weighted Neighbor Distance (WND), Naive Bayes Networks (NBN) and Radial Bias Function (RBF) classifiers for automated classification. Those techniques however had some problems as they require a two-stage processing in order to extract relevant features from the images, making them very focused on the specific problem and not generalizing

to broader types of cancers detection. Thanks to this specialization they nonetheless achieved $\sim$90% accuracy.

B. Deep Learning related

The following summarizes the advances in image classification via deep learning techniques exploiting neural networks.
Rucha et al [3] in *"Towards Designing an Automated Classification of Lymphoma subtypes using Deep Neural Networks"* exploits an architecture similar to the AlexNet and Inception model to perform the classification, by using Convolutional layers to reduce the size of the models and increase its generalization capabilities.
However, those techniques use random patching in order to reduce the dimensionality of the input, as the dataset is composed by 1300x1000 px images, which is a prohibitive size for those types of models.
The final best model has $\sim$95% accuracy.

III. Processing Pipeline

A. Dataset structure

The dataset used for this work is composed of 374 RGB images, 113 images for CLL, 139 images for FL and 122 for MCL of size 1388x1040px.
The dataset is small for current deep learning projects, thus at each epoch, each image received some transformations between:

- random flip: images are randomly flipped horizontally and vertically
- random contrast: even though this might interfere with the classification, as the color might be one of the determining factors: in the case it interferes, it can be seen as a label-smoothing technique, which is known to avoid overfitting and overconfident predictions.
- random zoom: image are randomly zoomed in a random location, equivalent to a random crop followed by a bilinear interpolation to restore the original size.

After the augmentation, each pixel is clipped in the range $[0, 255]$, then normalized in $[0, 1]$. Furthermore, the structure of the model will itself be a regularization technique, which will be further explained in the following sections.
However, different from all the previous models, the considered images are the entire high-resolution images; this is required to avoid any prior over the distribution of the informations in the images. In fact, considering a random crop as done in previous works would imply that any part of the image has enough information to classify the disease/cancer. This might be true for the specific chosen dataset, but it's not true in general: for example, given a full body x-ray, or some brain scan, a tumor is very local to a specific area of the fed image.
An alternative might be rescaling the image to a low-res version, which however might disrupt very important information, thus it's necessary to use the original full resolution image.

B. The proposed model

The proposed model is composed of 3 main components: an attention network, a feature network, and a classification network.
The intuition behind the 3 components are the following:

1) Attention network: it sees a low-resolution version of the image, which still holds local information, allowing the network to understand "where it is worth to look"
2) Feature network: given a patch of the image, it will compute some features that should encode the information regarding that patch
3) Classification network: given a list of features of an image, and the corresponding attention, it will compute the final classification.

The following is a detailed explanation of each of those modules.

1) Attention network a: The first component is the attention network. It is responsible for understanding where the information is likely to be in the image. In order to make this scalable, the network will receive a downscaled version of the image, in order to avoid the overhead due to the high resolution, which should still hold enough information to understand how much each "patch" is likely to be used.
Thus, given an image of size $N \times M$, the network will receive its version of size $(N/c) \times (M/c)$ where $c \in \mathbb{Z}^+$, and outputs a map of size $(N/c) \times (M/c)$, which is the attention distribution. Thus, given a image $I^{w \times h}$, its output $O = a(I)$ and $\sum O_i = 1$. In order to do that, a softmax activation is applied on both axes.
We then sample some patches from the high-resolution image according to the resulting attention distribution.

2) Feature network f: Once P patches have been sampled from the original image, each of them is processed independently by this feature network, which maps each patch to a vector.
Thus, given P patches of size $w_p \times h_p$, each of them is mapped to a corresponding vector $v_i \in \mathbb{R}^C$, where C is the number of classes in the classification task.
Those feature vectors will then be normalized such that $||v_i||_2 = 1$

3) Classification network g: Given the P normalized feature vectors obtained through f from the P sampled patches, the classification network will calculate the expected feature, considering the attention of each of the patches.
Thus, for each image, given P features, it will calculate the classification as follows:

$$o_j = g\left(\frac{\sum_{i=0}^{P} a(x;\theta)_{\text{patch}_i} \cdot f(\text{patch}_i;\phi)}{\sum_{i=0}^{P} a(x;\theta)_{\text{patch}_i}}\right) \tag{1}$$

In other words, each patch is processed by f to calculate its feature vector, which is then normalized in order to make each of them equally relevant, then they are weighted accordingly to the calculated attention, and finally they are transformed in a categorical distribution via g, which is a function for

the desiderd task, in this case study it will be a $softmax$ activation function.
The denominator is instead needed as the predicted probabilities of the attention map tends to be very small, thus it acts as a regularization/temperature for the softmax activation function; this happens due to the size of the categorical distribution that the attention network will output, since the size of it is as big as the scaled input image.

IV. Gradient estimation

Since the attention sampling is not differentiable, the attention network cannot be trained through back-propagation.
The gradient should be:

$$\nabla_{\theta_a} L = \frac{\delta L}{\delta o}\frac{\delta o}{\delta f}\frac{\delta f}{\delta a}\frac{\delta a}{\delta \theta_a}$$

However, the second to last term $\frac{\delta f}{\delta a}$ cannot be calculated.
In order to fix this, we will resort to the field of reinforcement learning, which is indeed based on sampling procedures in order to maximize a reward function.
The classification network is non-parametrized, thus it's not necessary to train it. The feature network instead is fully trainable with back-propagation. The attention network is the only piece that is needed to be trained in a reinforcement learning fashion.
As done for the derivation of the REINFORCE algorithm, the following can be shown:

$$\begin{aligned}\frac{\delta}{\delta\theta}\sum f(x;\phi) &\approx \mathbb{E}_{p\sim a(x;\theta)}[\frac{\frac{\delta}{\delta\theta}a(x;\theta)f(p;\phi)}{a(x;\theta)}] \\ &= \mathbb{E}_{p\sim a(x;\theta)}[\frac{\delta}{\delta\theta}\log a(x;\theta)f(p;\phi)]\end{aligned}$$

Then, using the chain rule, we can extend this to the classification network and the loss function:

$$\frac{\delta L}{\delta\theta} \approx \mathbb{E}_{p\sim a(x;\theta)}[\frac{\delta}{\delta\theta}\log a(x;\theta)L(g(f(p;\phi));y)]$$

Finally for the attention network, the objective function is:

$$max\, \mathbb{E}_{p\sim a(x;\theta)}[-L(g(f(p));y)] \tag{2}$$

Specifically, we could use any reward function, the only requirement is that the function is proportional to the loss of the classification. Since the considered task is a classification task, the loss considered is the categorical cross-entropy, defined as follows:

$$L(o;y) = -\sum_{i=0}^{\text{n. of classes}} y_i \cdot \log(o_i)$$

In particular, if we directly use the loss function as reward, we would have an unbounded reward, which causes high variance in the Monte Carlo estimation of the gradient for a. To avoid this, we can consider a single patch p and the categorical cross-entropy loss with respect to that specific image, so that the reward function is the following:

$$\begin{aligned}r(p;y) &= exp(-L(g(f(p));y)) \\ &= g(f(p))_{y_{\text{correct}}}\end{aligned}$$

This function is intuitive, as the reward will be nothing more than the classification probability of the correct classification. To further decrease the variance of the gradient, the rewards will be standardized, which is known in the RL community as the policy gradient with baseline:

$$\mathbb{E}_{p\sim a(x;\theta)}[\frac{\delta}{\delta\theta}\log a(x;\theta)(r - avg(r))]$$

V. Exploration/Exploitation dilemma

The exploration/exploitation dilemma is well-known in reinforcement learning. In summary, the dilemma states that a model should both time exploit what it has learnt to have a good result, but also explore, in order not to get stuck in a spurious good solution, and to find new possible better solutions.
It thus also holds for the attention network, as it's asked to find good patches, so that the feature network can learn their features. However, the attention network is also asked to not overly exploit what it has learnt, as it might just be a bad initialization, or a spurious good patch.
In order to cope with this problem, the attention network is required not to become too greedy/overly sure of the attention. To induce this, an entropy regularizer is also introduced during training for the attention network:

$$L'_a(x;y) = L_a(x;y) + \lambda\mathcal{H}(a(x))$$

With entropy defined as:

$$\mathcal{H}(p) = \sum_x p(x)\log(p(x))$$

This will thus penalize the attention network gradient if the attention map predicted is too greedy/deterministic, causing the sampling to be too predictable.

VI. Model

The model is based on 4 stages:

1) Attention
2) Sampling
3) Feature
4) Classification

A. Attention

The attention network is a deep convolutional neural network, which exploits multiple Convolutional layers alternated with Batch Normalization layers, layers very well known in the RL community to be fundamental to stabilize the learning. The network then takes the shape of the discriminator of the PatchGAN [4], which also exploits a very peculiar property of this architecture: since convolution has very narrow local fields, stacking multiple of them will increase that local field, but stacking just a few of them will allow the network to classify just a patch.

This is exactly the property needed in this case, as the attention network will output an attention score for each position, only depending on the local field around it. Furthermore, thanks to the fact that neither convolutional-layer nor batch normalization-layers' number of parameters depend on the size of the input image, this architecture will maintain the same number of parameters even for larger images.
The model used for the result is composed by the following layers:

1) Average pooling layer: used to downscale the input high resolution image to a low resolution version
2) Convolution + Batch norm x3: the convolutional layer uses 8 filters with 3x3 kernels and leaky relu activation functions
3) Linear Convolutional: convolutional layer that for each image, outputs logits
4) Softmax layer: softmax activation applied on both height and width of the resulting attention map

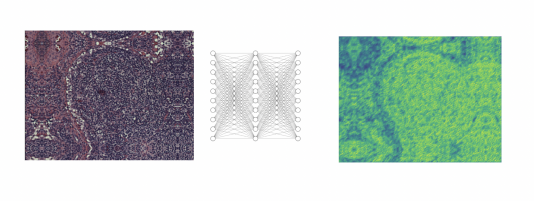

Fig. 1: Input/output of the attention network

The final model has only 1.513 trainable parameters.

B. Sampling

Given the attention map, we now proceed to sample patches according to that map. In order to do so, we will need to fix a-priori both the number of patches, and the patch size, which is only required to be greater than the scaling factor of the average pooling layer at the beginning of the attention network (which is usually no more than 5), to guarantee coverage.

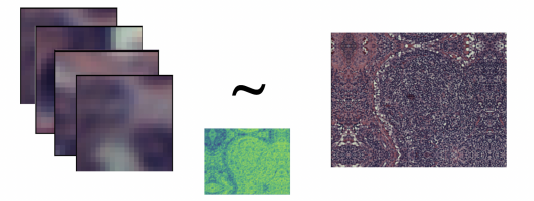

Fig. 2: Example of sampling patches

C. Feature

Once P patches have been sampled from the original image, the feature network will process each of them independently to extract a normalized logit vector.
To do so, a small neural network is used, with the following architecture:

1) Convolution x4: composed by 16 filters with 3x3 kernels and leaky relu activation
2) Global Max Pooling: average over the width and height channels
3) Affine transformation: linear dense layer to map from n_{channels} to n_{classes}
4) Normalization: layers that guarantees that each output vector is normalized such that $||v||_2 = 1$

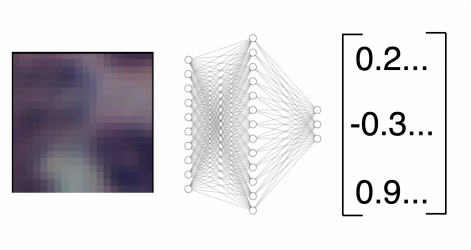

Fig. 3: Input/output of the feature network

Thanks to its structure, the final network used for this experiment uses only 7459 trainable parameters.

D. Classification

Finally, the classification is performed considering each normalized feature vector, weighted with respect to the corresponding attention, and the resulting expected feature vector is fed to a softmax layer that calculates the probability.

$$\text{softmax}\Big(\sum a(x)_i \cdot \begin{bmatrix} 0.2... \\ -0.3... \\ 0.9... \end{bmatrix}\Big)$$

Fig. 4: Classification framework

VII. Training details

Reinforcement learning-based learning models are known to be difficult/long to train, due to the fact that the learning is based on trial and error. Therefore, the initial phase is very random, and the reward is also noisy since the classifier is still at the beginning of the training.
While almost all the computational burden falls on the shoulders of the training, the network will become very light and fast later on in the evaluation/deploy, thanks to the only 9.000 parameters and the small input size for the feature network.
Furthermore, the training depends on 2 main hyperparameters: number of patches and patch size.
For the training, both networks have been trained for 300 epochs with the Adam optimizer, with stepsize $1e-3$ and almost no decay for both momentum and stepsize.
The batchsize used for all the models is 8, however each batch is replicated 4 times, as the sampled patches will be different, thus the same image can act as multiple training samples.

VIII. Results

Due to the dataset size, the evaluation of the models are very sensitive to the initialization. In order to overcome this, each of the results reported are the average of 5 random trials, approximating a 5-fold-cross-validation, which cannot be performed due to the training time for the networks.

P. size	N. of P.	Training acc.	Test acc.	Time
50	1	82% ± 2%	81% ± 2%	0:55h
50	5	90% ± 1%	89% ± 2%	1:45h
50	20	95% ± 1%	94% ± 1%	2:50h
20	5	88% ± 2%	87% ± 2%	1:25h

The best model uses 20 patches of 50x50 px thus the total area analyzed is composed by 50.000 pixels, compared to the 1.300.000 of the original image, thus only 3% of it; yet, it has comparable results with the state-of-the-art model presented at the beginning of the paper.

The following are the training loss and the validation loss, and the corresponding training accuracy and validation accuracy, of one run of training of the best model reported:

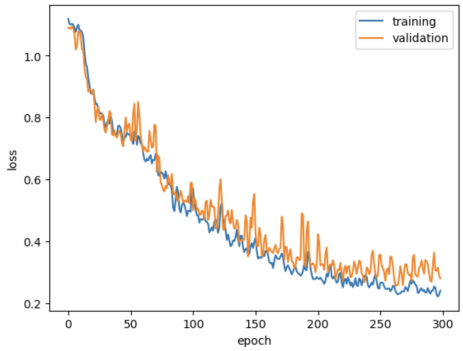

Fig. 5: Training and validation loss of the best model

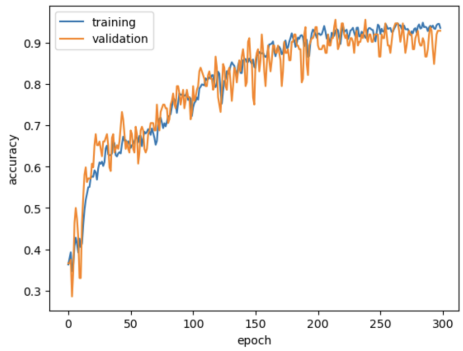

Fig. 6: Training and validation accuracy of the best model

It can be seen that even though the training dataset is composed of only 261 samples, and even though the model is trained with full resolution images, the attention network serves as regularization for the classifier, as it's fed with different patches each time. Instead, the attention network, since it's composed by very few parameters, is difficult to overfit by itself. Nonetheless, the training loss after 300 epochs shows little to no overfitting, yet it's the model trained with the most patches.

IX. Analysis of attention map

The following are 3 samples of attention of the best model, which uses patches of 50x50 px:

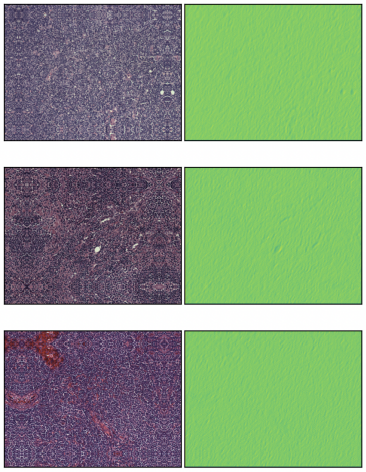

Fig. 7: Learned attention of the best model

This clearly explains why even random patching, used in [2] works fine, as the information necessary to classify the image is almost uniformly distributed across the whole image.
However, the same reasoning would not work for other types of cancers/diseases, where the effects of them are very localized. Instead the proposed method would work just fine, as the attention network would learn where in the image the classifier should pay attention.

An interesting insight instead can be drawn comparing the attention map with different patch sizes. The following is the comparison of the attention map of the model with 20x20 px patch sizes and 50x50 px patch sizes:

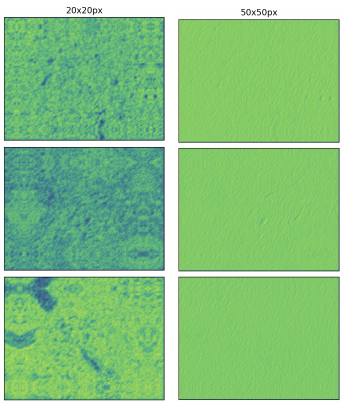

Fig. 8: Comparison of learned attention map between models with different patch size

Clearly, the model that has been forced to get only patches of size 20x20 px, has to be much more specific with which patches to get, thus the attention map is much less uniform, implying that the features used for classification might have a size about that. Instead, the model with 50x50px patches, was fine with any patch, since any patch of that size contained enough informations.

X. Explainability

A very important point that we would like to highlight is that this model is not only an order of magnitude smaller (around 9Kb total), but also much more explainable compared to the state-of-the-art.
The original model is limited in its explainability and unreliable given that it relies not only the output of a black-box classifier, but also on random patches from the original image. Instead, thanks to the 2-stage architecture of the proposed model, it's very simple to double check the results by experts, providing them just the analyzed patches.
This furthermore allows an expert to check if the network is giving attention to the right elements, thus giving more trust to the deployed model.
Finally, analyzing the training attention map by an expert, could allow an evaluation of the network accuracy, as the expert can confirm if the features that the attention network pays attention to aligns with the expert's experience.

XI. Possible improvements

A. Improvement regarding the classifier

In addition to the explainability, this model has another advantage: the attention network can be thought as a posterior over random cropping, and it's completely independent from the classifier.
Therefore, once the attention network has been trained with a simple classifier, we can use the network with more advanced models, such as state-of-the-art classifiers. We can provide the advanced model with patches from the attention distribution and weight the classification with the attention score of each patch, or use any other decision fusion mechanism.

B. Improvement regarding the attention/sampling

A better sampling procedure can be implemented. For this paper, in particular, the attention uses independent sampling to get patches from the original image. However, this is suboptimal. Let's take for example an image where the object that we want to understand is if an image is family friendly or not, and say it contains 2 objects of size 100x100 px, but the patch size selected is 50x50 px. If we sample 2 patches, it's possible that one patch falls in one mode of the distribution and the second one in the other mode of the distribution. Due to this, we have 2 patches of size 50x50 px that are spatially far apart and not very recognizable, as the non-family-friendly objects are 100x100 px.
Instead, to overcome this limitation, it would be enough to use conditional sampling, which is widely adopted in NLP. To do so, we would require a RNN to take the initial image, output a distribution $q(p_1|\text{image})$, from which we sample p_1 and at the next step, feed the network with the sampled element, and output the conditional distribution $q(p_2|p_1, \text{image})$, and so on. Thanks to this, once a patch is sampled from one mode, the network might change the distribution to a local one near the first patch, in order to increase the probability that both patches comes from the same mode of the distribution.

XII. Concluding Remarks

With this work we have shown how an attention network can be useful when considering large images, where only a subset of it is necessary for the classification. This allowed us to use small networks for both the attention and the classification stage, and the size of the two components could be chosen independently depending on the available hardware.
The structure of this architecture allows it to be used in a wider range of classification problems, where the information necessary for the classification is very local, in which random cropping would likely fail.
Furthermore, thanks to the attention module, experts can easily double check the results from the model, making it a potential first-frontier for mass-screening campaigns, followed by a second stage with experts that can futher investigate the results for the most severe cases, considering only the relevant patches.

Student analysis

Thanks to this work, we learned concrete application of RL outside the control sector, which allowed us to integrate non-differentiable pieces into the architectures. However, this could only have been possible thanks to another course in the engineering department, as this subject is not covered elsewhere.
Additionally, no library was found for image sampling, and similar works implements these steps using a C++ wrapper, while for the implementation of this architecture, we implemented that piece using Numpy. Even though the final code for this part is aroud 20 lines long, it look great effort to almost completely write it in Numpy to be highly performant. Its naive python implementation was two order of magnitudes slower, which would have been prohibitive for the training of the final model.
Also, the dataset is composed by high quality images, which have the drawback of being heavy, thus cloud computing like Colab was impossible, as it would have required uploading the whole dataset multiple times.
Furthermore, even though evaluation of the model is very cheap, the training requires a lot of RAM, which would have been higher than the maximum allowed by Colab.

References

[1] N. V. Orlov, "Automatic classification of lymphoma images with transform-based global features," *IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 4, pp. 1003-1013, July 2010. doi: 10.1109/TITB.2010.2050695.*

[2] R. Ferjaoui, "Lymphoma Lesions Detection from Whole Body Diffusion-Weighted Magnetic Resonance Images," *5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, 2018, pp. 364-369. doi: 10.1109/CoDIT.2018.8394840*.

[3] T. Rucha, "Towards Designing an Automated Classification of Lymphoma subtypes using Deep Neural Networks," *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.