- ORIGINAL ARTICLE -

Intermediate Task Fine-Tuning in Cancer Classification

Clasificación de Cáncer mediante Transferencia de Conocimiento con Tarea Intermedia

Mario Alejandro García[1], Martín Nicolás Gramática[1], and Juan Pablo Ricapito[1]

[1] *Universidad Tecnológica Nacional Facultad Regional Córdoba, Argentina*
{mgarcia, mgramatica}@frc.utn.edu.ar

Abstract

Reducing the amount of annotated data required to train predictive models is one of the main challenges in applying artificial intelligence to histopathology. In this paper, we propose a method to enhance the performance of deep learning models trained with limited data in the field of digital pathology. The method relies on a two-stage transfer learning process, where an intermediate model serves as a bridge between a pretrained model on ImageNet and the final cancer classification model. The intermediate model is fine-tuned with a dataset of over 4,000,000 images weakly labeled with clinical data extracted from TCGA program. The model obtained through the proposed method significantly outperforms a model trained with a traditional transfer learning process.

Keywords: deep learning, digital pathology, histopathology, intermediate task fine-tuning, transfer learning

Resumen

Reducir la cantidad de datos etiquetados necesarios para entrenar modelos predictivos es uno de los principales desafíos para la aplicación de la inteligencia artificial en patología digital. En este trabajo se propone un método para mejorar la capacidad de predicción de redes neuronales profundas entrenadas con cantidades limitadas de imágenes de patología digital. El método es un proceso de *transfer learning* de dos etapas, donde se utiliza un modelo intermedio como puente entre un modelo preentrenado con ImageNet y un modelo final de clasificación de cáncer. El modelo intermedio es ajustado con un dataset de más de 4.000.000 de imágenes débilmente etiquetadas con datos clínicos extraídos del programa TCGA. El modelo obtenido a través del método propuesto mejora significativamente los resultados de un modelo ajustado con el proceso tradicional de *transfer learning*.

Palabras claves: ajuste fino con tarea intermedia, aprendizaje profundo, histopatología, patología digital, transferencia de conocimiento.

1 Introduction

Artificial intelligence (AI), mainly through deep learning (DL), has made great advances in medicine in recent years. These advances, carried out in research laboratories, have had very little impact on clinical practice. The challenges that still need to be overcome for AI to achieve clinical value have been widely discussed [1, 2, 3, 4, 5, 6]. One of the main barriers is the difficulty (or cost) of obtaining large amounts of expertly annotated multicentre data.

AI in medicine has recently attracted a lot of interest through automated image analysis in the area of histopathology. One example is PathLAKE [1], one of five state-established histopathology and AI centers in the UK. In digital histopathology, images are managed using a special technology called virtual microscopy or, more commonly, whole-slide imaging (WSI). In this context, a slide is a gigapixel image of tissue stored in a hierarchical structure. A broader overview of the topic can be found in [5, 7].

Obtaining labelled data in histopathology is challenging due to the size of the slides. Deep neural networks, the state of the art in image pattern recognition, are models with many internal parameters and therefore require a large amount of data to train without overfitting. In this context, the main challenge for AI in histopathology is to reduce the amount of data required for training or to automatically/semi-automatically label data.

Due to the promising results of the first AI applications, the size of the datasets has increased. However, even though collecting large numbers of slides is a manageable task for pathology laboratories and medical centers, labeling remains an obstacle [3]. Labeling can mean both manual annotation of image regions (such as identifying tissue regions or the location of specific cell types) and clinical annotation (such as assessing molecular subtypes, treatment response, and survival). Collecting manual image annotations is a tedious task that requires domain expertise. Clinical annotations, on the other hand, require access to pathology reports and electronic patient records, either from a hospital (to retrieve information on grades, molecu-

[1] `https://www.pathlake.org/`

lar subtypes, or treatment response) or from a regional or national registry (to retrieve survival information), and can only be provided by clinical researchers or authorized data managers. Clinical labeling of slides is usually easier to accomplish than manual annotation. This has resulted in large clinical datasets. Nevertheless, it is expected that building AI models using only clinical annotations will not be possible or efficient for all medical imaging diagnostic applications. Therefore, manual labeling will continue to be necessary, and techniques will need to be developed to efficiently utilize and produce these annotations.

The aim of this paper is to analyze the behavior of DL models trained by a two-stage transfer learning process. In the first stage, weakly labeled data from the intermediate domain is used to train an intermediate model, and in the second stage, the labeled data for the target tasks is used to train the final models.

2 Background

We use the notation and definition of Pan and Yang [8] regarding Transfer Learning. Firstly, we define "domain" and "task".

A domain $\mathscr{D}$ is composed of two parts: a feature space $\mathscr{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathscr{X}$. In the task of image classification, $\mathscr{X}$ is the space of all images in the domain, x_i is the i^{th} image, and X is a particular training dataset. In general, if two domains are different, they may have different feature spaces or different marginal probability distributions.

Given a specific domain $\mathscr{D} = \{\mathscr{X}, P(X)\}$, a task $\mathscr{T} = \{\mathscr{Y}, f(\cdot)\}$ is composed of two components: a label space $\mathscr{Y}$ and a predictive function $f(\cdot)$, which is not known but can be learned from training data, consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathscr{Y}$. The function $f(\cdot)$ can be used to predict the label of a new input instance x.

To define the concept of transfer learning, a source domain $\mathscr{D}_S$ and a target domain $\mathscr{D}_T$ are considered. More specifically, we denote the source domain data as $D_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_{n_S}}, y_{S_{n_S}})\}$, where $x_{S_i} \in X_S$ is the i^{th} input instance and y_{S_i} is its corresponding class label. In the case of image classification, x_{S_i} is an image that belongs to class y_{S_i}. Similarly, we denote target domain data as $D_T = \{(x_{T_1}, y_{T_1}), \ldots, (x_{T_{n_T}}, y_{T_{n_T}})\}$, where $x_{T_i} \in \mathscr{X}_T$ and $y_{T_i} \in \mathscr{Y}_T$. In most cases, $0 < n_T \ll n_S$.

Given a source domain $\mathscr{D}_S$ and task $\mathscr{T}_S$, a target domain $\mathscr{D}_T$, and task $\mathscr{T}_T$, *transfer learning* aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathscr{D}_T$ using knowledge from $\mathscr{D}_S$ and $\mathscr{T}_S$, where $\mathscr{D}_S \neq \mathscr{D}_T$ or $\mathscr{T}_S \neq \mathscr{T}_T$.

The condition $\mathscr{T}_S \neq \mathscr{T}_T$ implies that $\mathscr{Y}_S \neq \mathscr{Y}_T$ or that $P(Y_S|X_S) \neq P(Y_T|X_T)$.

In the present work, the kind of transfer learning used is called *inductive*. This is the case when the target task is different from the source task. In this case, some labeled data from the source domain are used to *induce* a predictive model $f_T(\cdot)$.

The most commonly used approach in DL is knowledge transfer through parameters. Parameter transfer approaches assume that individual models for related tasks should share some parameters or hyperparameter distributions. Specifically, in *network-based deep transfer learning*, a pre-trained network in a source domain $f_S(\cdot)$, including its structure and parameters, is partially reused and transferred to a new neural network $f_T(\cdot)$ used in the target domain [9].

3 Objectives

In this paper, we intend to reduce the amount of data needed for cancer classification in histopathology images by using clinically labeled data and a double transfer learning process, where the model trained to recognize clinical data acts as a bridge between the original model and the target model.

To achieve this, a deep neural network $f_T(\cdot)$ is trained on the $\mathscr{T}_T$ task, recognizing patterns on images of the D_T dataset. The training process has two stages: (1) The parameters of the pre-trained model $f_S(\cdot)$ on the dataset D_S to perform the task $\mathscr{T}_S$ are taken, and then fitted to perform the intermediate task $\mathscr{T}_I$ on the dataset D_I obtaining the $f_I(\cdot)$ model; (2) Finally, the parameters of $f_I(\cdot)$ are transferred to $f_T(\cdot)$ and fitted to perform the task $f_T(\cdot)$ on the dataset D_T.

To investigate the behavior of the process under different conditions, two experiments are conducted with distinct target tasks ($\mathscr{T}_{T_H}$ and $\mathscr{T}_{T_D}$) and datasets (D_{T_H} and D_{T_D}). In both experiments, D_S corresponds ImageNet dataset, while D_I is a dataset extracted from The Cancer Genome Atlas (TCGA)[2]. The classes y_{I_i} represent tissue types in WSI tiles, and thus can be considered as clinical data rather than annotations. It is worth noting that D_{T_H} was also sourced from TCGA, thereby suggesting a high proximity between the domains $\mathscr{D}_I$ and $\mathscr{D}_{T_H}$. D_{T_D} is composed of WSI tiles obtained from a different source and featuring distinct size, therefore $\mathscr{D}_{T_D}$ is further from $\mathscr{D}_I$.

On the other hand, D_{T_D} comprises WSI image tiles obtained from a different source and features distinct dimensions compared to D_I, leading to a greater dissimilarity between domains, denoted as $\mathscr{D}_{T_D}$ and $\mathscr{D}_I$ respectively.

The performances of $f_T(\cdot)$ and $f'_T(\cdot)$ are compared. $f'_T(\cdot)$ is obtained by a direct transfer learning process, i.e., without going through the intermediate model. The performance is expected to improve because $n_S \gg n_I \gg n_T$ and the input data x_{I_i} are closer to x_{T_i} than x_{S_i} data.

[2] https://www.cancer.gov/ccg/research/genome-sequencing/tcga

4 Related Work

Phang *et al.*, working in the field of natural language processing, proposed an approach similar to ours and called it Supplementary Training on Intermediate Labeled-data Tasks (STILTs) [10]. The procedure consists of three steps: (1) train a language model on unsupervised data; (2) next, train the model on an intermediate task for which sufficient labeled data is available; (3) finally, fine-tune and evaluate the model for the target task.

In the field of medical imaging, Niu *et al.* [11] diagnose COVID-19 on computed tomography (CT) images of the lungs using a similar approach which they call Distant Domain Transfer Learning (DDTL) and which is inspired by the Transitive Transfer Learning (TTL) proposed by Tan *et al.* [12]. They use data from four non-medical source domains, chest X-rays as an intermediate domain, and CT images of the lungs as the target domain. Intuitively, the target domain is closer to the intermediate domain than to the source domain, similar to our work.

5 Materials and Methods

5.1 Data

5.1.1 Target dataset D_{T_H}: HIUTR

The target dataset D_{T_H} is Histology Images from Uniform Tumor Regions in TCGA Whole Slide Images (HIUTR)[3] [13] created by Komura *et al.* [14]. It consists of 1,608,060 images of 32 types of cancer in hematoxylin-eosin stained tissues. The images were taken from 7,951 patients, and there may be more than one slide per patient. The images are available at six magnification levels: 0: $0.5 : \mu m/pixel$, 1: $0.6 : \mu m/pixel$, 2: $0.7 : \mu m/pixel$, 3: $0.8 : \mu m/pixel$, 4: $0.9 : \mu m/pixel$, and 5: $1.0 : \mu m/pixel$. The cancer types and the number of samples for each one are shown in Table 1. The original names are kept to facilitate comparison with other works.

Two pathologists analyzed the downloaded images and removed those that did not meet certain quality criteria, such as out-of-focus images or staining issues. They then labeled the images by marking uniform tumor regions on each slide. The images in the dataset are the tiles extracted from the labeled regions.

As an example, one of the images from HIUTR is shown in Figure 1.

In Table 1, it is clear that the classes are not balanced. The type of cancer with the lowest occurrence has 360 instances, while the most common one has 14,070. Regarding magnifications, the number of images is uniformly distributed.

[3] https://zenodo.org/record/5889558#.ZFeOIHbMK00

Table 1: HIUTR dataset classes.

#	Class	Samples
1	Adrenocortical_carcinoma	2880
2	Bladder_Urothelial_Carcinoma	5900
3	Brain_Lower_Grade_Glioma	13760
4	Breast_invasive_carcinoma	14070
05	Cervical_squamous_cell_carcinoma_and...	3600
6	Cholangiocarcinoma	540
7	Colon_adenocarcinoma	4920
8	Esophageal_carcinoma	1850
9	Glioblastoma_multiforme	13840
10	Head_and_Neck_squamous_cell_carcinoma	6690
11	Kidney_Chromophobe	1360
12	Kidney_renal_clear_cell_carcinoma	6780
13	Kidney_renal_papillary_cell_carcinoma	3930
14	Liver_hepatocellular_carcinoma	4860
15	Lung_adenocarcinoma	9600
16	Lung_squamous_cell_carcinoma	9340
17	Lymphoid_Neoplasm_Diffuse_Large_B...	360
18	Mesothelioma	1260
19	Ovarian_serous_cystadenocarcinoma	1420
20	Pancreatic_adenocarcinoma	2210
21	Pheochromocytoma_and_Paraganglioma	720
22	Prostate_adenocarcinoma	5440
23	Rectum_adenocarcinoma	970
24	Sarcoma	8070
25	Skin_Cutaneous_Melanoma	5750
26	Stomach_adenocarcinoma	5660
27	Testicular_Germ_Cell_Tumors	3500
28	Thymoma	2100
29	Thyroid_carcinoma	6540
30	Uterine_Carcinosarcoma	1240
31	Uterine_Corpus_Endometrial_Carcinoma	7380
32	Uveal_Melanoma	890

5.1.2 Target dataset D_{T_D}: DeepHisto

The target dataset D_{T_D} is DeepHisto[4][15]. It consists of 40,777 images of 5 classes, 3 glioma subtypes, necrosis and normal brain tissue. The images are tiles of hematoxylin-eosin stained WSI collected at the National Center of Pathology (NCP), Luxembourg National Health Laboratory. WSIs were acquired with an average slide resolution of $0.25 \mu m/pixel$.

Region annotation of WSIs was done by a pathologist, and the regions of interest are further divided into square 512×512 tiles, each of them associated with a particular class denoting the respective tumor entity, normal brain tissue or necrosis (Table 2).

Tiles are further divided into training and test subsets patient-wise.

As an example, an image from DeepHisto is shown in Figure 2.

[4] https://zenodo.org/record/7941080

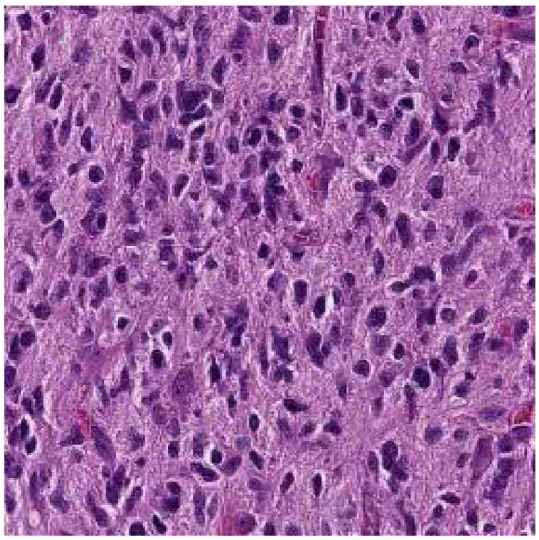

Figure 1: Image of an case of the Glioblastoma_multiforme class at magnification level 3, from the HIUTR dataset.

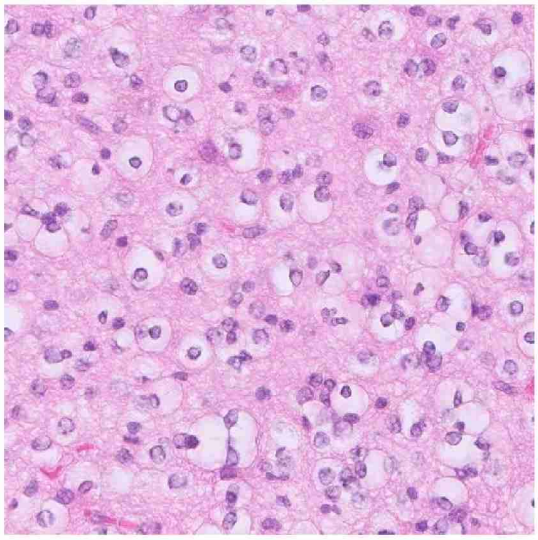

Figure 2: Image of an case of the Oligodendroglioma class from the DeepHisto dataset.

Table 2: DeepHisto dataset classes.

Class	Samples
Astrocytoma	4220
Glioblastoma	1874
Oligodendroglioma	1874
Necrosis	479
Normal brain tissue	32330

5.1.3 Intermediate dataset D_I: PathoNet

The intermediate dataset, PathoNet[5] [16], was created for this work. It consists of 4,462,126 images divided into 12 classes (tissues). These images were extracted from the TCGA program, like HIUTR, but no annotations were made, only the tissue type was taken from the program metadata.

For each tissue, 400,000 256×256 pixel images were randomly selected and downloaded from 400 WSIs. An automated cleaning process was then performed to eliminate cases with excessive white content and blurred images.

Table 3 shows the final number of images for each class.

Table 3: PathoNet dataset classes.

Class	Samples
Bladder	386770
Brain	393168
Breast	380050
Bronchus and lung	385738
Colon	296685
Corpus uteri	391476
Kidney	388139
Liver and intrahepatic bile ducts	393195
Prostate gland	369125
Skin	383206
Stomach	368673
Thyroid gland	325931

As an example, one of the images from PathoNet is shown in Figure 3.

5.2 Performance metrics

The most common performance measure for this kind of problem is accuracy. While accuracy would be a good choice for a case with balanced classes, we decided to use F-score as the primary metric because it is sensitive to biases caused by differences in data sets. To apply F-score in multiclass classification, we performed a one-vs-all (OVA) calculation, where the metric value is calculated for each class by simulating

[5] https://zenodo.org/record/8116751

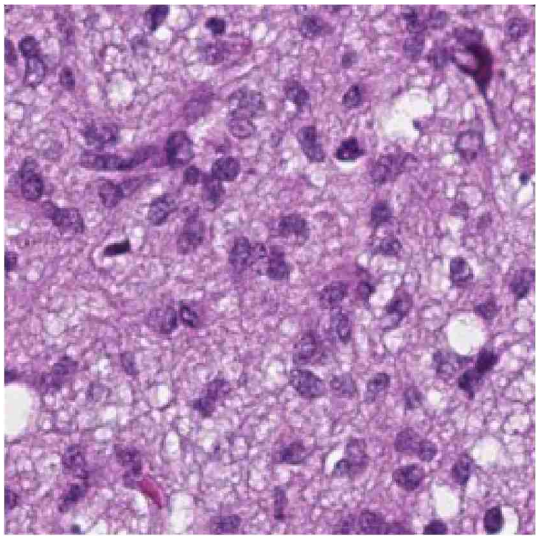

Figure 3: Image of an case of Brain class from the PathoNet dataset.

a binary classification of the class of interest against the rest. After calculating the value for each class, the mean across all classes is calculated.

In addition to the F1 OVA, we calculated the recall OVA and the accuracy OVA to study their behavior. Differences between the metrics could indicate that some models are more prone to bias, for example due to class imbalance. We also calculated absolute accuracy to allow comparison with other studies and to monitor during training.

6 Experiments

6.1 Neural network

Throughout the experiments, the behavior of a ResNet-18 model was analyzed. Five distinct configurations of the neural network were employed: (C1) all trainable weights; (C2) trainable weights from the second stage to the end; (C3) trainable weights from the third stage to the end; (C4) trainable weights from the fourth stage to the end; (C5) only the trainable output layer. Figure 4 shows the five configurations graphically.

6.2 Model f_I training

The tissue prediction model for the PathoNet dataset was obtained by fine-tuning the weights of a neural network (f_S) pre-trained with ImageNet. Two configurations, C1 and C4, were used. Furthermore, to compare the performance, an adjustment from random weights (without taking the parameters of f_S) was carried out.

Table 4 presents the accuracy results for the three configurations.

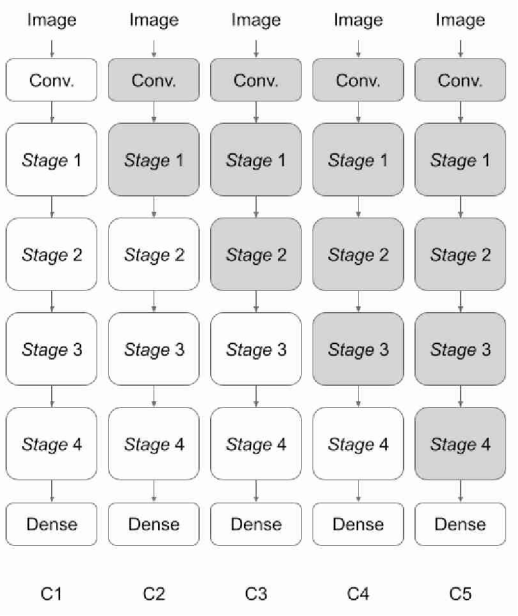

Figure 4: Model configuration diagrams. White layers fit their weights and gray layers do not.

Table 4: Accuracy of f_I model in tissue prediction on PathoNet.

Training configuration	Accuracy
ImageNet-C1	0.8089
ImageNet-C4	0.6877
Random-C1	0.7572

6.3 Model f_{T_H} training

The model f_{T_H} predicts the type of cancer on the HIUTR dataset.

In order to analyze the behavior under different conditions, the five configurations (C1-C5) were used to fine-tune f_{T_H}.

Each model was trained in two ways: (A) with one stage of transfer learning, transferring parameters from a pre-trained model using ImageNet to the final model; (B) with two stages of transfer learning, using the intermediate model as a bridge between the source and target models.

The loss function was weighted according to the number of cases for each HIUTR class in order to reduce the effects of class imbalance.

20% of the HIUTR dataset was set aside for testing. As the source of both the HIUTR and PathoNet datasets is the same, a process was defined to ensure that individuals with images in the HIUTR test dataset did not have images in either the HIUTR training dataset or the PathoNet dataset.

6.4 Model f_{T_D} training

f_{T_D} predicts the glioma subtype on the DeepHisto dataset. The training process is similar to that conducted with f_{T_H}, with the following differences:

In the initial trials, it was noticed that the model f_I with C1 configuration (ImageNet and random) did not serve as a suitable starting point for f_{T_D}. Therefore, the experiments for this model were exclusively conducted using the ImageNet-C4 configuration of f_I.

Data splitting (test and training data) is defined in the dataset documentation.

6.5 Code availability

The source code is available at `https://github.com/PatologiaDigitalUTN/itft`

7 Results

7.1 Model f_{T_H}

Table 5 contains the mean metric values for three training epochs of each case and table 6 contains the deviations of these metrics. We took epochs 1-3 for training methods B and 3-5 for methods A because these represent the best iterations from each group before overfitting. We selected three epochs instead of one in order to show the variability in the network's output. It can be seen that cases with training method B consistently outperform cases with training method A for the same configuration. For method B, the weights of f_I fitted with the ImageNet-C1 configuration were employed as a starting point. Starting from Random-C1 and ImageNet-C4 configurations did not surpass the performance achieved by method A.

Table 5: Mean metrics values for three training epochs of each case of f_{T_H}.

Case	F-score	Acc OVA	Recall	Accuracy
A-C1	0.540	0.760	0.533	0.615
B-C1	0.614	0.802	0.614	0.686
A-C2	0.496	0.741	0.496	0.568
B-C2	0.613	0.800	0.610	0.687
A-C3	0.487	0.739	0.491	0.574
B-C3	0.601	0.795	0.601	0.677
A-C4	0.438	0.715	0.446	0.526
B-C4	0.604	0.791	0.593	0.677
A-C5	0.290	0.659	0.338	0.359
B-C5	0.484	0.757	0.529	0.554

Table 7 shows the average improvement of method B over method A in all metrics for each configuration.

The source data from column F-Score in table 7 is shown in Figure 5.

In all cases, three epochs were used to compute the means. In model trainings B-C1, B-C2, B-C3 and BC-4, the best result is obtained in the first epoch (then overfit) and the first three epochs were used to calculate the metrics. In cases A-C1, A-C2, A-C3, and AC-4, the best results are obtained near the fifth epoch, and the third, fourth, and fifth epochs were used to compute the metrics. In cases A-C5 and B-C5, where only the output dense layer is trained, the models are more stable and take longer to overfit, therefore the data were taken from the three epochs before overfitting.

Table 6: Average absolute deviation of each metric for three training epochs of each case of f_{T_H}.

Case	F-score	Acc OVA	Recall	Accuracy
A-C1	0.0108	0.0040	0.0077	0.0089
B-C1	0.0017	0.0009	0.0018	0.0013
A-C2	0.0219	0.0123	0.0240	0.0313
B-C2	0.0078	0.0033	0.0064	0.0044
A-C3	0.0030	0.0023	0.0047	0.0026
B-C3	0.0013	0.0020	0.0040	0.0014
A-C4	0.0011	0.0009	0.0018	0.0034
B-C4	0.0034	0.0005	0.0010	0.0022
A-C5	0.0015	0.0027	0.0054	0.0022
B-C5	0.0013	0.0003	0.0007	0.0013

Table 7: Average improvement of training method B over training method A for each configuration.

Conf.	F-score	Acc OVA	Recall	Accuracy
C1	13.82%	5.47%	15.18%	11.53%
C2	23.61%	7.94%	22.94%	20.83%
C3	23.47%	7.68%	22.43%	17.89%
C4	37.94%	10.63%	33.03%	28.66%
C5	66.78%	14.95%	56.37%	54.14%

Table 12 shows the confusion matrix of f_{T_H} B-C1, epoch 3.

In Figure 6, the training evolution for cases A-C1 and B-C1 is shown. Notice that the lowest error for B-C1 is reached at the end of the first epoch. The same occurs for B-C2, B-C3, and B-C4.

7.2 Model f_{T_D}

As mentioned earlier, results surpassing the performance of process A (one-stage transfer learning) were not achieved when starting from f_I with the ImageNet-C1 configuration. In this case, f_{T_D} tends to overfit. Additionally, unsatisfactory outcomes were obtained for the Random-C1 configuration, f_{T_D} underfits. The results presented in the remaining section pertain to the ImageNet-C4 configuration of f_I. Figure 7 shows a sample of training for the three configurations of f_I and the C5 configuration of f_{T_D}.

Configurations C1 to C4 of f_{T_D} tend to overfit, and the training process becomes unstable as more parame-

Table 8: Confusion matrix of f_{T_H} B-C1 epoch 3. Codes (#) correspond to those in Table 1. Columns correspond to predicted classes and rows to true classes.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	**4894**	91	163	38	16	0	0	16	11	3	11	0	83	502	1	58	0	18	0	3	1	17	0	85	269	17	0	1	0	1	0	1
2	109	**5004**	218	667	595	4	190	423	80	582	2	11	146	244	299	1034	33	218	17	13	0	298	98	442	304	185	312	13	31	65	129	94
3	200	131	**24458**	80	33	0	2	6	1139	35	17	143	78	54	38	86	24	6	0	1	67	114	0	549	88	5	81	132	18	1	4	0
4	31	903	54	**20010**	464	167	226	216	31	224	14	47	64	338	501	687	47	581	200	72	73	459	6	282	531	257	146	134	307	225	989	4
5	14	279	58	198	**3512**	15	75	227	8	251	0	12	200	48	167	723	16	84	9	11	53	30	2	58	92	294	0	91	55	76	810	2
6	0	85	0	163	0	**266**	0	20	0	12	4	0	15	209	2	26	0	12	0	45	0	97	0	0	0	122	0	0	2	0	0	0
7	0	6	4	57	48	0	**5615**	163	0	62	0	11	38	44	567	268	8	57	18	184	0	152	1733	2	0	156	11	1	11	0	74	0
8	3	570	0	177	114	12	22	**1569**	20	588	8	0	1	83	20	254	0	32	151	10	7	14	24	21	92	223	7	34	14	0	55	5
9	184	242	3549	34	81	1	29	107	**20798**	181	1	16	216	31	61	56	5	18	1	7	5	163	1	1023	237	75	137	6	9	252	3	11
10	13	659	47	43	688	5	241	610	68	**8321**	5	16	11	82	356	1599	6	68	4	19	47	54	4	99	202	340	22	28	19	30	43	1
11	20	2	15	14	0	0	0	0	0	52	**1860**	243	223	153	2	2	0	24	14	0	8	67	0	0	1	0	56	0	2	0	12	0
12	6	4	26	24	15	0	52	0	226	460	308	**11771**	376	191	131	46	0	0	0	8	0	146	0	171	0	11	3	0	9	6	0	0
13	8	164	30	41	79	0	1	5	58	53	173	846	**4715**	289	447	330	0	1	34	81	27	275	0	31	12	25	5	7	176	4	29	4
14	54	200	32	178	67	9	3	126	9	181	60	108	167	**8132**	75	224	0	11	0	0	24	210	0	31	40	85	1	3	64	0	103	3
15	0	91	49	277	126	19	461	101	32	297	3	236	377	33	**12073**	3263	34	71	32	175	124	195	38	86	148	564	69	8	65	11	419	3
16	1	363	55	239	682	15	163	451	77	697	29	186	198	246	2300	**12742**	35	74	22	66	3	146	57	161	244	222	10	98	24	103	244	37
17	0	82	185	35	2	0	9	1	0	0	0	0	6	0	2	6	**483**	65	0	0	0	14	17	48	63	50	4	1	0	0	1	6
18	26	106	33	307	20	22	0	0	84	46	0	1	202	342	150	204	0	**547**	1	75	2	109	1	75	54	27	0	0	10	0	14	2
19	31	117	0	42	92	6	2	6	0	11	0	0	0	94	17	84	0	16	**2248**	3	2	33	0	0	75	5	27	1	17	50	41	0
20	4	90	1	2	77	30	47	204	1	82	0	14	221	94	277	89	0	55	14	**2889**	2	109	16	50	23	169	25	0	10	20	5	0
21	273	31	42	0	0	0	0	0	1	1	0	1	3	65	14	1	0	0	0	2	**1075**	11	0	75	1	0	0	3	11	0	0	0
22	0	3	27	99	10	35	80	1	3	2	0	54	146	36	281	119	0	1	41	100	30	**9553**	4	0	0	66	9	1	8	2	9	0
23	0	0	0	48	15	0	1498	5	0	8	0	0	1	0	25	5	0	25	25	0	0	23	**308**	0	0	43	0	0	117	0	94	0
24	88	614	206	175	242	7	3	61	89	264	0	234	77	143	55	429	102	135	2	94	111	110	0	**12132**	399	28	100	26	3	4	287	0
25	92	428	142	225	535	0	104	329	133	342	1	0	164	216	208	254	23	27	2	1	81	36	5	161	**6918**	10	174	130	34	64	203	528
26	1	108	8	374	357	16	476	302	4	345	4	207	55	196	688	609	10	11	7	152	0	222	114	87	152	**6063**	98	14	11	9	187	3
27	6	70	2	118	32	11	14	24	0	56	0	0	28	12	75	3	2	4	0	1	1	18	0	126	18	475	**5925**	2	0	6	11	0
28	0	154	10	57	206	0	0	13	1	99	0	0	10	19	3	132	167	0	2	1	0	54	0	180	20	10	97	**2875**	1	2	207	0
29	14	19	9	245	28	19	9	106	7	140	12	36	223	49	183	49	0	14	1	9	1	283	0	69	74	93	0	73	**11460**	2	303	0
30	80	31	150	199	6	0	16	12	30	23	0	0	7	0	132	55	26	0	84	19	1	0	0	24	190	43	74	2	1	**1110**	345	0
31	91	326	36	747	556	1	506	259	8	27	8	1	283	99	809	524	18	13	290	62	24	298	52	39	304	267	37	98	255	235	**8584**	3
32	0	106	28	6	4	0	0	31	0	29	0	0	37	8	1	16	0	1	1	0	80	0	0	1	190	0	1	0	0	1	0	**1439**

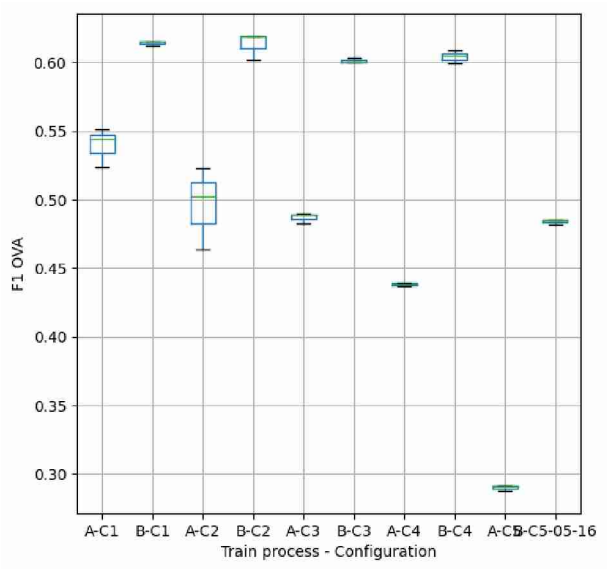

Figure 5: F-score OVA on test data by training mode and configuration.

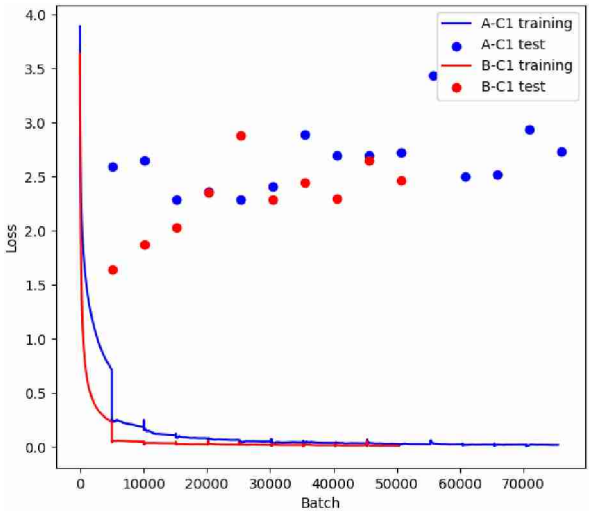

Figure 6: Training evolution of cases A-C1 and B-C1 of f_{T_H}. The solid line shows the training loss calculated in each batch. The dots indicate the test loss at the end of each epoch.

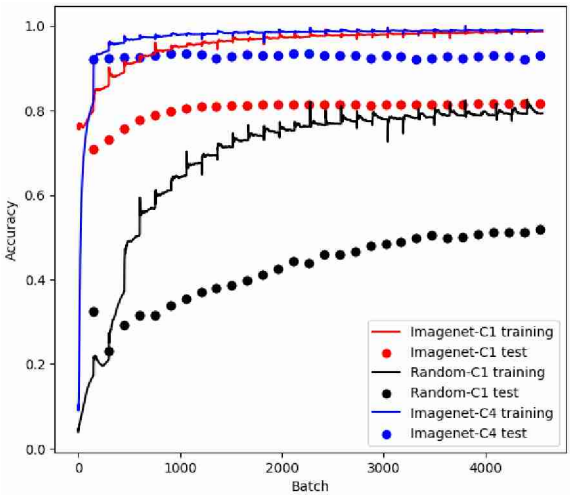

Figure 7: Training evolution of f_{T_D} B-C5 with ImageNet-C1, Random-C1 and ImageNet-C4 configurations for f_I. The solid line shows the training accuracy in each batch. The dots indicate the test accuracy at the end of each epoch (151 batches).

ters are released, likely due to the relatively small size of the dataset. An illustrative example with process A on configurations C1, C4, and C5 is presented in Figure 8. Configuration C5 exhibits the highest stability and best performance, albeit it is dependent on the epoch of fine-tuning of f_I from which the parameters are taken. The subsequent results are based on the C5 configuration of f_{T_D}.

Table 9 displays the mean metric values for 30 (last 10 epochs of 3 trainings) training epochs per case. It is evident that the performance is enhanced and surpasses process A when the weights from the initial epochs of the f_I training are utilized. Table 10 contains the deviations of metrics in Table 9.

Table 9: Mean metrics values for thirty epochs of training of f_{T_D}. The last number in *Case* column is the training epoch of f_I.

Case	F-score	Acc OVA	Recall	Acc
BC-5-1	**0.859**	**0.921**	**0.856**	**0.938**
BC-5-2	0.795	**0.876**	**0.777**	0.917
BC-5-3	**0.859**	**0.916**	**0.847**	**0.941**
BC-5-4	0.784	**0.883**	**0.787**	0.911
BC-5-5	0.705	0.851	0.727	0.894
BC-5-6	0.744	0.865	0.753	0.897
BC-5-7	0.728	0.850	0.726	0.897
AC-5	0.796	0.874	0.773	0.926

Table 11 shows the average improvement of method B over method A in all metrics for C5, Figure 9 depicts the data from the F-score column in Table 9 and Table 12 shows an example of confusion matrix.

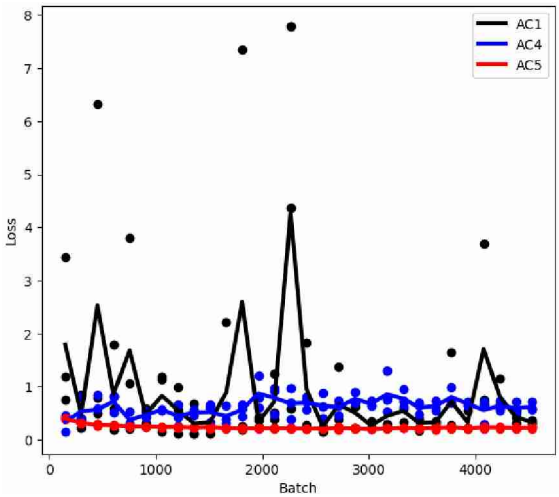

Figure 8: Training evolution of f_{T_D} A-C1, A-C4 and A-C5 on three executions by configuration. The solid line shows the test mean loss by epoch (151 batches). The dots indicate individual loss by epoch.

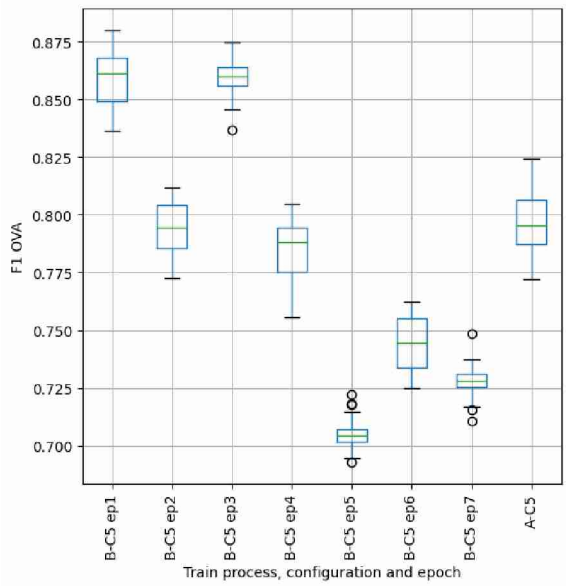

Figure 9: F-Score OVA on test data by training mode, configuration and epoch of f_I.

Table 10: Average absolute deviation of metrics in Table 9.

Case	F-score	Acc OVA	Recall	Acc
BC-5-1	0.0102	0.0057	0.0105	0.0034
BC-5-2	0.0090	0.0050	0.0097	0.0018
BC-5-3	0.0059	0.0038	0.0071	0.0021
BC-5-4	0.0116	0.0056	0.0107	0.0032
BC-5-5	0.0051	0.0028	0.0050	0.0024
BC-5-6	0.0099	0.0042	0.0079	0.0023
BC-5-7	0.0048	0.0034	0.0060	0.0022
AC-5	0.0111	0.0075	0.0139	0.0037

Table 11: Average improvement of method B over method A in all metrics for C5 configuration in training of f_{T_D}. The last number in *Case* column is the training epoch of f_I.

Case	F-score	Acc OVA	Recall	Acc
BC-5-1	**7.85%**	**5.35%**	**10.81%**	**1.27%**
BC-5-2	-0.22%	**0.22%**	**0.63%**	-1.01%
BC-5-3	**7.88%**	**4.79%**	**9.64%**	**1.60%**
BC-5-4	-1.57%	**1.04%**	**1.87%**	-1.69%
BC-5-5	-11.43%	-2.60%	-5.96%	-3.45%
BC-5-6	-6.54%	-1.03%	-2.51%	-3.18%
BC-5-7	-8.59%	-2.74%	-6.04%	-3.16%

8 Conclusions

The two-stage transfer learning process using PathoNet dataset to fit the coefficients of the intermediate model significantly improved the classification of images extracted from TCGA program (HIUTR). For DeepHisto the improvement is also significant, although it was not achieved as directly as in the first case.

It is concluded that the proposed methodology is promising in cases where there are few annotated medical images, but a sufficient volume of images with general clinical labels can be accessed.

9 Discusion and future work

While the proposed method achieved improvements in both tasks, the outcomes varied. $\mathscr{T}_{T_H}$ demonstrated enhancement across all five configurations studied, utilizing the C1 configuration for fine-tuning the intermediate task. On the other hand, $\mathscr{T}_{T_D}$ could only be explored under the C5 configuration and achieved improvements when utilizing the initial epochs of fine-tuning the intermediate task under C4 configuration. The datasets D_{T_H} and D_{T_D} differ in two dimensions: their proximity to domain $\mathscr{D}_I$ and the amount of data available. Further experiments with alternative datasets are necessary to investigate the individual effects of these factors.

Table 12: Confusion matrix of f_{T_D} B-C5 epoch 30 from f_I epoch 2. Columns correspond to predicted classes and rows to true classes.

Astro.	465	0	0	0	0
Gliob.	15	148	0	9	69
Necrosis	0	2	80	7	1
Normal	18	2	5	2916	6
Oligo.	107	5	0	1	318

In the results over HIUTR dataset, it is also observed that cases A-Cx reduce the performance as the coefficients of the model are frozen, while cases B-Cx (except B-C5) maintain the same accuracy. This behavior could indicate that the encoding achieved in all three initial stages of the model is common to the two tasks adjusted on TCGA program data ($\mathscr{T}_I$ and $\mathscr{T}_{T_H}$). $\mathscr{T}_I$ aims to detect tissues and $\mathscr{T}_{T_H}$ detects cancer types.

Competing interests

No competing interests exist.

Funding

This work was supported by the Universidad Tecnológica Nacional, PID UTN8436.

Authors' contribution

All authors contributed equally to this work. The final manuscript was read and approved by all authors.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation through the NVIDIA GPU Grant Program.

References

[1] R. Colling, H. Pitman, K. Oien, N. Rajpoot, P. Macklin, C.-P. A. in Histopathology Working Group, V. Bachtiar, R. Booth, A. Bryant, J. Bull, *et al.*, "Artificial intelligence in digital pathology: a roadmap to routine use in clinical practice," *The Journal of pathology*, vol. 249, no. 2, pp. 143–150, 2019.

[2] B. Acs, M. Rantalainen, and J. Hartman, "Artificial intelligence as the next step towards precision pathology," *Journal of internal medicine*, vol. 288, no. 1, pp. 62–81, 2020.

[3] J. Van der Laak, G. Litjens, and F. Ciompi, "Deep learning in histopathology: the path to the clinic," *Nature medicine*, vol. 27, no. 5, pp. 775–784, 2021.

[4] H. Yoshida and T. Kiyuna, "Requirements for implementation of artificial intelligence in the practice of gastrointestinal pathology," *World journal of gastroenterology*, vol. 27, no. 21, p. 2818, 2021.

[5] S. Kobayashi, J. H. Saltz, and V. W. Yang, "State of machine and deep learning in histopathological applications in digestive diseases," *World Journal of Gastroenterology*, vol. 27, no. 20, p. 2545, 2021.

[6] A. Reinke, M. D. Tizabi, C. H. Sudre, M. Eisenmann, T. Rädsch, M. Baumgartner, L. Acion, M. Antonelli, T. Arbel, S. Bakas, *et al.*, "Common limitations of image processing metrics: A picture story," *arXiv preprint arXiv:2104.05642*, 2021.

[7] L. Pantanowitz, A. Sharma, A. B. Carter, T. Kurc, A. Sussman, and J. Saltz, "Twenty years of digital pathology: an overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives," *Journal of pathology informatics*, vol. 9, no. 1, p. 40, 2018.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pp. 270–279, Springer, 2018.

[10] J. Phang, T. Févry, and S. R. Bowman, "Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks," *arXiv preprint arXiv:1811.01088*, 2018.

[11] S. Niu, M. Liu, Y. Liu, J. Wang, and H. Song, "Distant domain transfer learning for medical imaging," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3784–3793, 2021.

[12] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1155–1164, 2015.

[13] D. Komura and S. Ishikawa, "Histology images from uniform tumor regions in TCGA Whole Slide Images." Available at: `https://doi.org/10.5281/zenodo.5889558`, Jan. 2021.

[14] D. Komura, A. Kawabe, K. Fukuta, K. Sano, T. Umezaki, H. Koda, R. Suzuki, K. Tominaga, M. Ochi, H. Konishi, *et al.*, "Universal encoding of pan-cancer histology by deep texture representations," *Cell Reports*, vol. 38, no. 9, p. 110424, 2022.

[15] M. Mittelbronn, A.-C. Hau, S.-Y. Kim, P. V. Nazarov, V. Despotovic, A. Kakoichankava, F. B. K. Borgmann, and G. G. Klamminger, "DeepHisto: Dataset for glioma subtype classification from Whole Slide Images." Available at: `https://doi.org/10.5281/zenodo.7941080`, May 2023.

[16] M. A. Garcia, M. N. Gramatica, J. P. Ricapito, T. S. Fiezzi, M. Ángel Gignone, and L. Rostagno, "Pathonet." Available at: `https://doi.org/10.5281/zenodo.8116751`, July 2023.

Citation: M.A. García, M.N. Gramática and J.P. Ricapito. *Intermediate Task Fine-Tuning in Cancer Classification*. Journal of Computer Science & Technology, vol. 23, no. 2, pp. 135–144, 2023.
DOI: 10.24215/16666038.23.e12.
Received: April 17, 2023 **Accepted:** October 2, 2023.