Efficient Deep Learning for Image Classification: Lighter Preprocessing and Fewer Parameters

Samuel Felipe dos Santos*§, Nicu Sebe†, and Jurandy Almeida*

*Federal University of São Carlos - UFSCar, 18052-780, São Carlos, SP - Brazil
Emails: samuel.felipe@ufscar.br, jurandy.almeida@ufscar.br
†University of Trento - UniTN, 38123, Trento, Trentino - Italy
Email: niculae.sebe@unitn.it

***Abstract*—Convolutional neural networks have achieved state-of-the-art performance in several computer vision tasks recently, learning high-level representations directly from RGB images. However, using deeper architectures has led to high computational costs, hindering deployment on devices with limited resources. Additionally, models are usually specialized in a single domain/task while an increasing amount of real-world applications need to deal with multiple domains simultaneously. The computational cost of storing and running multiple instances of those costly models can limit their utilization even more. This Ph.D. thesis aims to reduce the computational burden of deep learning, focusing on two main aspects: reducing data preprocessing cost and sharing parameters across multiple domains/tasks. These contributions have led to the creation of efficient models with high classification performance and reduced costs, allowing them to be deployed in a wider array of devices.**

I. Introduction

Convolutional neural networks (CNNs) are an specialization of Artificial Neural Networks (ANNs). They are designed to process data with multiple layers of convolution operations in a hierarchical manner, allowing them to learn features directly from the raw input data [1], [2]. Due to this motive, CNNs have achieved state-of-the-art performance on a wide array of computer vision tasks, such as, classification, segmentation, object detection, image super-resolution, denoising, medical images, autonomous driving, road surveillance, among others [3], [4].

However, to achieve this performance, increasingly deeper and more complicated architectures have been used, and at the same time, acquisition devices are capable of obtaining images with higher resolutions [1]. This makes computational cost one of the main problems faced by deep learning, since inference and training are very expensive [5]. The need for high processing power and abundant memory capacity makes it difficult to apply such models to devices whose available computing resources are limited, like mobile phones and other edge devices. Therefore, specialized optimizations at both software and hardware levels are an imperative need for developing efficient and effective deep learning-based solutions [6].

Another limitation is that deep learning research is usually focused at increasing performance on a single domain [7], which requires learning and storing a whole new computational expensive model for each domain [8]. The heavy computational cost to run and the great amount of memory to store these models can hinder their deployment in environments with limit computation resources, like embedded devices [9].

For storage and transmission purposes, most image data available are often stored in a compressed format, like JPEG, PNG and GIF [10]. From these formats, JPEG has remained the most popular despite the advances in video compression and is considered a simple solution to store and transmit visual data [11]. To use this data with a typical CNN, it would be required to decode it to obtain the RGB images used as input, a task demanding high memory and computational cost [3]. In edge devices with low computational power, like embedded systems, this decoding step can also became a bottleneck [12].

A possible alternative to alleviate this problem is to design CNNs capable of learning with DCT (Discrete Cosine Transform) coefficients rather than RGB pixels [3], [5], [10]–[19]. The DCT is a representation of the data in the frequency domain easily extracted by partial decoding, saving computational cost. Frequency domain image processing can yield advantages like computational efficiency and spatial redundancy removal, being used in many computer vision tasks, like image compression, image coding, face recognition, signature verification, gender classification, human gait recognition, lane detection, brain tumor classification, among others [20]–[22].

The key idea exploited by existing works is adapting traditional CNN architectures to facilitate the learning with DCT coefficients rather than RGB pixels. However, the changes in the network generally lead to a significant increase in its computational complexity.

In addition to making preprocessing faster, it is also important to use the same model for multiple domains/tasks, since many applications demand it and storing multiple entire deep models can be difficult for devices with limited memory. A possible solution is Multi-Domain Learning (MDL), an approach based on the observation that, although the domains can be very different, they still can share a significant amount of low and mid-level visual patterns [7]. Therefore, to tackle this problem, a common goal is to learn a single compact model that performs well in several domains while sharing the majority of the parameters among them, with only a few

§This work presents the main contributions of the Ph.D. thesis defended by Samuel Felipe dos Santos.

domain-specific ones. This reduces the cost of having to store and learn a whole new model for each new domain.

Berriel et al. [8] point out that one limitation of those methods is that, when handling multiple domains, their computational complexity is at best equal to the backbone model for a single domain. Therefore, they are not capable of adapting their amount of parameters to custom hardware constraints or user-defined budgets. To address this issue, they proposed the modules named Budget-Aware Adapters (BA^2) that were designed to be added to a pre-trained model to allow them to handle new domains and to limit the network complexity according to a user-defined budget. They act as switches, selecting the convolutional channels that will be used in each domain. Even though using this method reduces the number of parameters required for each domain, the entire model is still required at test time if it aims to handle all the domains. The main reason is that they share few parameters among the domains, which forces loading all potentially needed parameters for all the domains of interest.

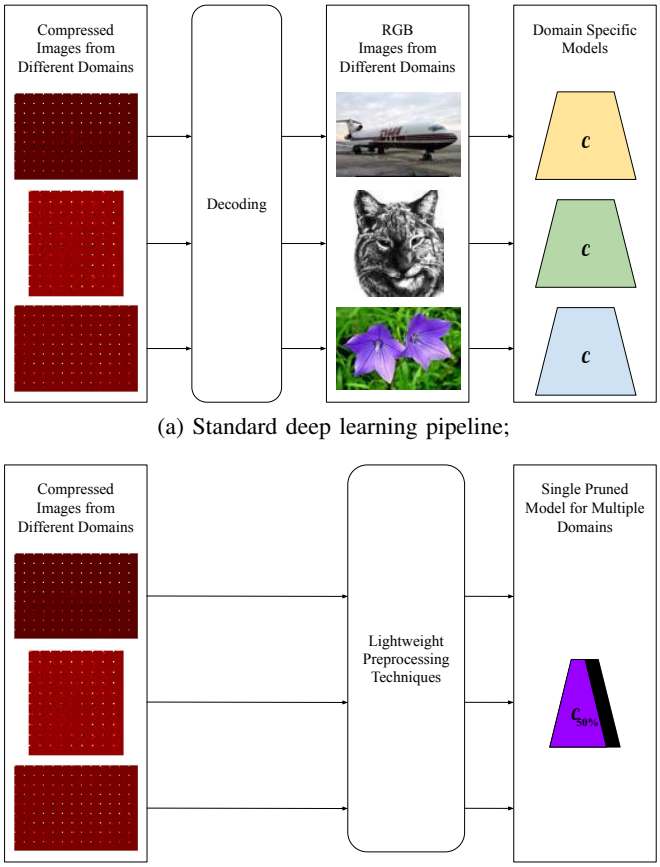
A possible strategy to reduce the amount of parameters are model compression techniques, like network pruning, where the least import weights are removed, reducing the amount of parameters. In unstructured pruning, single connections are removed from the model, while in structured pruning, entire filters, channels and even layers are removed.

Figure 1 compares a standard deep learning pipeline and a more efficient deep learning that uses the strategies proposed in this Ph.D. thesis. In standard deep learning, images are usually stored in compressed formats, like JPEG, and need to be decoded before being fed to the network, moreover for each different domain, a new model is required.

The main objective of this Ph.D. thesis was to make deep learning more efficient. In order to achieve this goals, two different strategies were explored. First, methods to design deep models directly for compressed images were explored, using lightweight preprocessing techniques in order to avoid the cost of decoding and thus reduce the computational complexity. Then, multi-domain learning jointly with model compression techniques were used to obtain a single compact model for multiple domains, avoiding the need of learning and storing a whole new model for each domain of images while once again reducing computational cost.

The main contributions generated by this work were [23]:

- A new approach for better preprocessing data from the compressed domain, allowing for models designed directly for compressed images to have equal or lower computational complexity to their RGB counterpart. This way, the applications using it can take full advantage of the speed up obtained by avoiding the decoding of the images [18], [24], [25];
- Taking advantage of the high amount of channels from the DCT representation available at compressed images like JPEG, this work showed that it is possible to skip more stages of the model while keeping good accuracy, making the models even faster [18], [24], [25];



(a) Standard deep learning pipeline;

(b) Efficient deep learning using strategies proposed in this work.

Fig. 1. Comparison of (a) standard deep learning pipeline and (b) efficient deep learning using approaches proposed in this work.

- Measurements of inference speed from preprocesing images and passing them through the network are presented, allowing for the analysis of the gains obtained by using models designed for compressed images in a more clear manner [18], [24], [25];
- Strategies for sharing parameters while imposing budget constraints in multi-domain learning are proposed, obtaining models that can handle multiple domains, while fitting the computational resources available to users [26];
- The approach for multi-domain learning proposed in this work, as far as we know, is the first capable of handling multiple domains while having a lower amount of parameters than the original model for a single domain [26];
- The S_E efficiency-effectiveness score metric was presented for image classification in a single domain/task and for multi-domain learning, combining classification accuracy, computational complexity, and amount of parameters in a single metric, allowing for a better analysis of the trade-off between these factors [23].

II. Reducing Data Preprocessing Costs

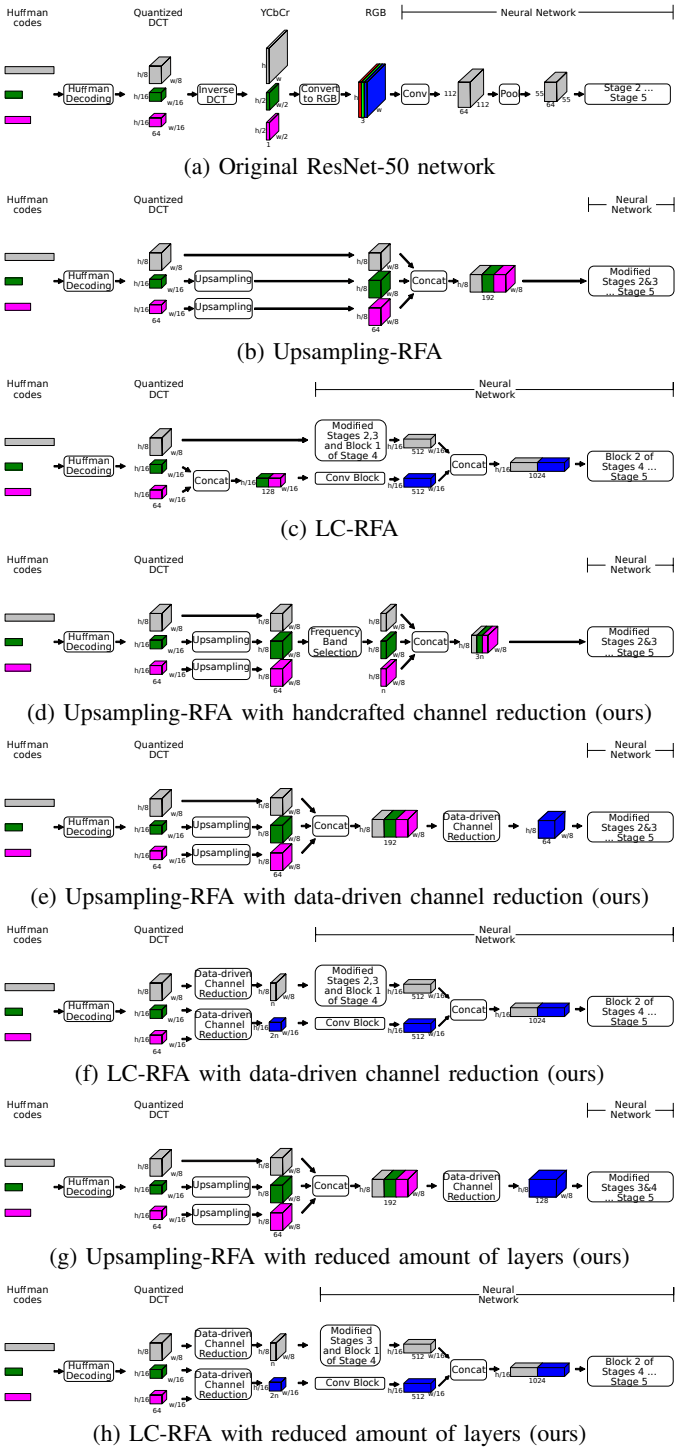
High computational cost and memory consumption are some of the main problems faced by state-of-the-art deep

learning models, since they hinder their applicability in environments with limited computational resources, like embedded devices. For this motive, several methods have been proposed in the literature to accelerate CNNs. However, only a few of these works explore reducing the cost of preprocessing the images fed to the network, like avoiding the cost of decoding images by designing models that work directly with information available on the compressed representation of the image. In the contributions of this work published in [18], [24], [25], the potential of CNNs designed for the compressed domain is analyzed, evaluating if it is worth avoiding the cost of decompression, and the consequences of doing so. The experiments considered not only the classification accuracy but also the computational efficiency of the models.

Figure 2 show an overview of the proposed methods. The ResNet-50 [27] is a widely-used deep learning model fed with RGB images. Gueguen et al. [13] proposed modified versions of the ResNet-50 that are fed directly with DCT coefficients, a representation of the image that is available at the compressed image, avoiding the cost of fully decoding the image. Despite the speed-up obtained by partially decoding compressed data, the architectural changes made to the model led to a significant decrease in computational efficiency. This Ph.D. thesis extended Gueguen et al. [13] work, alleviating the computational complexity and number of parameters of the model with handcrafted [24] and data-driven [25] preprocessing techniques for reducing the amount of input channels that are fed to the network. Finally, a strategy to skip stages of the network in order to reduce computational cost even further was also proposed.

To alleviate the computational complexity and number of parameters of the Upsampling-RFA and LC-RFA [13], the Frequency Band Selection (FBS) technique was proposed to manually select the most relevant DCT coefficients before feeding them to the network. Since higher frequency information has little visual effect on the image, the n lowest frequency coefficients are retained. For that, the second stage of the ResNet-50 network was changed to accommodate the amount of retained coefficients, i.e., $3n$ input channels. For more details, refer to [23], [24].

The results obtained by the proposed FBS technique indicated that reducing the number of channels in the early stages of the network can be effective in reducing the computational costs of the network. For this reason, smarter strategies capable of learning how to make this reduction were explored. Different strategies were used to reduce the number of input channels to the amount expected by the original ResNet-50 at that stage. The decreased strides at early blocks, as proposed by Gueguen et al. [13], were kept. To accommodate this amount of input channels, the number of output channels of the second and third stages were changed to be the same as the original ResNet-50. Unlike the FBS, where the DCT inputs are discarded by hand, potentially losing image information, these techniques take advantage of all DCT inputs and learn how to combine them in a data-driven fashion. For this, three different approaches were evaluated: (1) a linear projection (LP) of the DCT inputs, (2) a local attention (LA) mechanism, and (3) a cross-channel parametric pooling (CCPP). For a more detailed explanation of each o these methods, refer to [23], [25].



(a) Original ResNet-50 network

(b) Upsampling-RFA

(c) LC-RFA

(d) Upsampling-RFA with handcrafted channel reduction (ours)

(e) Upsampling-RFA with data-driven channel reduction (ours)

(f) LC-RFA with data-driven channel reduction (ours)

(g) Upsampling-RFA with reduced amount of layers (ours)

(h) LC-RFA with reduced amount of layers (ours)

Fig. 2. Overview of (a) the ResNet-50 for RGB images, (b,c) models of Gueguen et al. [13] for DCT coefficients, (d,e) methods proposed in this work for decreasing the preprocessing cost by reducing the amount of input channels and (f,g) reducing the overall computational cost by skipping stages of the model.

Both the Upsampling-RFA and LC-RFA [13] skips the first stage of the original ResNet-50, feeding the DCT coefficients to the second stage, which is modified to accommodate the amount of input channels. To reduce the complexity and amount of parameters from the network even further, the effects of also skipping the second, third, and fourth stages were analyzed, but the stride reduction proposed by Gueguen et al. [13] at the early blocks of the initial stage in which the DCT coefficients are provided as input were maintained.

To take into consideration the accuracy, amount of parameters, and computational complexity in a single metric, being able to evaluate the trade-off between classification performance and computational cost, the efficiency-effectiveness S_E score was proposed [23].

Table I shows the main results obtained. The Upsampling-RFA and LC-RFA from Gueguen et al. [13] have the lowest S_E score since they increase the amount of parameters and computational complexity. The handcrafted FBS technique was effective, but relevant information from the DCT inputs was discarded, leading to a similar S_E score to the baseline ResNet-50. Learning how to combine all DCT inputs in a data-driven fashion reduced the computational complexity and number of parameters while keeping a similar accuracy, reaching higher S_E scores than the baseline ResNet-50. Finally, skipping stages 1 and 2 of the network was beneficial, leading to models with a good balance between efficiency and effectiveness, being the strategies with the best S_E score.

TABLE I
COMPARISON OF COMPUTATIONAL COMPLEXITY (GFLOPS), NUMBER OF PARAMETERS, FRAMES PER SECOND ON INFERENCE AND CLASSIFICATION ACCURACY ON THE IMAGENET DATASET FOR THE ORIGINAL RESNET-50 WITH RGB, NETWORKS DESIGNED FOR DCT, AND THIS PH.D. THESIS STRATEGIES FOR REDUCING THE NUMBER OF INPUT CHANNELS AND LAYERS OF DCT MODELS.

Approach	GFLOPs	Params	FPS	Accuracy	S_E
ResNet-50 (3x1)	3.86	25.6M	588	73.46	1.00
Upsampling-RFA (3x64)	5.40	28.4M	494	72.33	0.62
LC-RFA (3x64)	5.11	27.4M	510	72.75	0.69
Upsampling-RFA + FBS (3x32)	3.68	26.2M	616	70.22	0.94
Upsampling-RFA + FBS (3x16)	3.18	25.6M	645	67.03	1.01
Upsampling-RFA + CCPP (1x64)	3.20	25.6M	639	69.73	1.09
LC-RFA + CCPP (1x32|1x64)	3.14	24.7M	616	71.04	1.19
LC-RFA + CCPP (1x16|1x32)	3.13	24.7M	624	69.84	1.15
Upsampling-RFA + CCPP + skipping 1^{st} and 2^{nd} stages (1x128)	2.86	25.1M	771	70.49	1.27
LC-RFA + CCPP + skipping 1^{st} and 2^{nd} stages (1x32|1x64)	2.91	24.4M	735	70.04	1.26
LC-RFA + CCPP + skipping 1^{st} and 2^{nd} stages (1x16|1x32)	2.90	24.4M	727	69.14	1.24

The results obtained in this work indicate that designing a model to work directly with data from the compressed domain is promising, being able to obtain similar accuracy while greatly improving the computational cost of these models.

III. SHARING PARAMETERS ACROSS MULTIPLE DOMAINS/TASKS

A relevant limitation of deep models is that they are usually specialized into a single domain or task, this way, when the model needs to be used in a new domain, it is necessary to learn and store a new set of parameters for it. As an alternative to alleviate this problem, Multi-Domain Learning (MDL) strategies can be used. These approaches attempt to have a single compact model capable of dealing with images from multiple domains.

Berriel et al. [8] proposed the BA^2 modules. They work as binary masks that select which convolutional channels are used for each of the domains, having a user-defined budget capable of reducing computational complexity by decreasing the amount of channels used by each domain. The main limitation of BA^2 is that it is not capable of using a lower amount of parameters than the backbone model for a single domain, since the masks from each domain select different sets of parameters.

The contribution of this work published in [26] was built upon BA^2 [8] by encouraging multiple domains to share convolutional filters, enabling the pruning of the weights not used by any of the domains at test time. Therefore, it made it possible to create a single model with lower computational complexity and fewer parameters than the baseline model for a single domain. Such a model is capable of better fitting the desired budget from users with limited access to computational resources.

Figure 3 shows an overview of the problem addressed by the proposed method, comparing it to previous MDL solutions and emphasizing their limitations. As it can be seen, standard adapters use the entire model, while BA^2 [8] reduces the number of parameters used in each domain, but requires a different set of parameters per domain. Therefore, the entire model is needed for handling all the domains together and nothing can be effectively pruned. On the other hand, the proposed approach increases the probability of using a similar set of parameters for all the domains. In this way, parameters that are not used for any of the domains can be pruned at test time. These compact models have fewer parameters and computational complexity than the original backbone model, which facilitates their use in resource-limited environments.

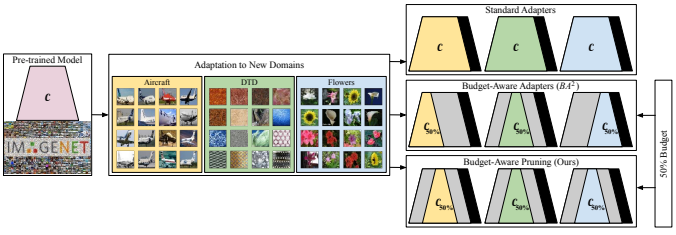


Fig. 3. The multi-domain learning (MDL) problem, where a pre-trained model is adapted to handle new domains. In standard adapters, the amount of parameters from the domain-specific models (indicated in colored C) is equal to or greater than the backbone model (due to the mask represented in black). Budget-Aware Adapters can reduce the number of parameters required for each domain (unused parameters are denoted in gray). However, the whole model is needed at test time if handling distinct domains (colored areas share few parameters). The model proposed in this Ph.D. thesis encourages different domains to use the same parameters (colored areas share most parameters). Thus, when handling multi-domains at test time, the unused parameters can be pruned without affecting the domains.

The proposed method has a user-defined budget that allows the model to fit the available processing and storage capabili-

TABLE II
COMPUTATIONAL COMPLEXITY, NUMBER OF PARAMETERS, ACCURACY PER DOMAIN, S, S_O, S_P, AND S_E SCORES ON THE VISUAL DOMAIN DECATHLON.

Method	FLOP	Params	ImNet	Airc.	C100	DPed	DTD	GTSR	Flwr.	Oglt.	SVHN	UCF	S-score	S_O	S_P	S_E
Baselines [7]:																
Feature	1.000	1.00	59.7	23.3	63.1	80.3	45.4	68.2	73.7	58.8	43.5	26.8	544	544	544	1
Finetune	1.000	10.0	59.9	**60.3**	82.1	92.8	55.5	97.5	81.4	87.7	96.6	51.2	2500	2500	250	2.11
TAPS [28]:																
$\lambda = 0.25$	1.004	5.68	**63.5**	50.3	**83.7**	94.1	60.1	8.7	86.6	88.5	96.6	50.2	3113	3101	548	5.74
$\lambda = 0.50$	1.003	5.10	**63.5**	50.8	83.5	94.9	59.8	97.2	87.1	88.1	96.6	50.5	2821	2813	553	5.26
$\lambda = 0.75$	1.003	4.44	**63.5**	48.2	83.4	94.1	**61.5**	98.2	**87.2**	87.7	96.3	**51.9**	2919	2910	657	6.46
$\lambda = 1.00$	1.002	3.78	**63.5**	49.9	83.3	93.8	59.1	98.3	86.3	87.7	96.2	50.5	2862	2856	757	7.30
BA^2 [8]:																
$\beta = 1.00$	0.646	1.03	56.9	49.9	78.1	95.5	55.1	99.4	86.1	88.7	**96.9**	50.2	**3199**	4952	3106	51.97
$\beta = 0.75$	0.612	1.03	56.9	47.0	78.4	95.3	55.0	99.2	85.6	88.8	96.8	48.7	3063	5005	2974	50.30
$\beta = 0.50$	0.543	1.03	56.9	45.7	76.6	95.0	55.2	99.4	83.3	**88.9**	**96.9**	46.8	2999	5523	2912	54.35
$\beta = 0.25$	0.325	1.03	56.9	42.2	71.0	93.4	52.4	99.1	82.0	88.5	**96.9**	43.9	2538	7809	2464	65.02
Ours:																
$\beta = 1.00$	0.581	1.03	56.9	48.7	77.9	95.5	55.1	99.2	85.1	88.5	96.7	47.6	3036	5226	2948	52.06
$\beta = 0.75$	0.461	0.84	56.9	43.4	76.8	**95.7**	54.0	**99.5**	78.3	88.0	96.6	44.3	2821	6119	3358	69.43
$\beta = 0.50$	0.403	0.73	56.9	43.1	76.2	95.4	52.1	99.0	76.7	88.2	96.4	44.1	2546	6318	3488	74.47
$\beta = 0.25$	**0.212**	**0.41**	56.9	35.2	69.8	95.4	48.5	98.8	71.4	87.3	96.3	41.9	2138	**10085**	**5215**	**177.72**

ties. To achieve these goals, an extra loss function was added to BA^2 to encourage parameter sharing among domains and prune the weights that are not used by any domain. It was also necessary to train simultaneously in all the domains to be able to handle them all together at test time. Figure 4 show an overview of the method.

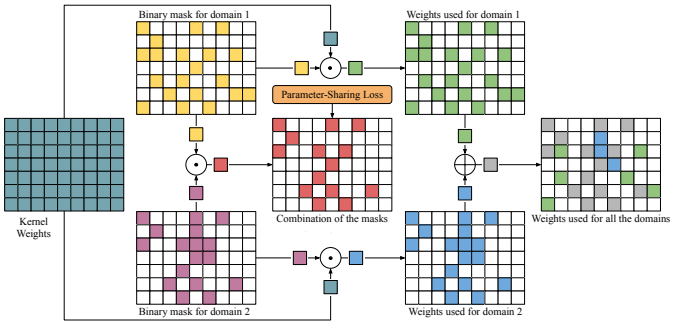


Fig. 4. Overview of the proposed strategy for sharing parameters among domains. The parameter-sharing loss function is computed over a combination of the masks from all the domains and is used to encourage the sharing of parameters between them. Parameters that are not used by any domain can be pruned, reducing the number of parameters and computational cost of the model. Colors represent data (i.e., weights, masks, etc), therefore, the colored squares denote the input data for each operation as well as its resulting output.

Three novel loss functions that encourage the sharing of convolutional features among distinct domains were proposed. The different parameter-sharing loss functions were developed based on different relationships between the masks that select the parameters of each domain. The $L_{PS}^{Int.}$ loss takes into consideration the intersection between the masks, the L_{PS}^{Union} used the union of the masks, and the $L_{PS}^{Jaccard}$ used the Jaccard similarity coefficient. Experiments were run defining the importance of this loss manually and learning it as a parameter of the model.

The S-score metric [7] is used to measure classification performance over all domains for MDL. The S_O and S_P [8] are the ratios between S-score and computational complexity and amount of parameters, respectively. This Ph.D. thesis proposed adapting the efficiency-effectiveness S_E score to the MDL problem, combining both S_O and S_P, being a metric that takes into consideration S-score, computational complexity, and amount of parameters (for more details, refer to [23]).

Among the methods proposed in this work, the best overall results were obtained by L_{PS}^{Union} when its importance is learned and can be seen in Table II, where the proposed method is compared with other baseline and state-of-the-art methods in the Visual Decathlon Challenge [7], a well-known benchmark for MDL comprised of 10 different image domains (for the complete results of all experiments, refer to [23]).

The proposed method was capable of vastly outperforming the baseline methods of only training the classifier (feature) and finetuning a pre-trained model for the current domain. Compared to the state-of-the-art methods, i.e., TAPS [28] and BA^2 [13], the proposed method has a lower S-score in most cases, but its S_O, S_P, and S_E are all considerably higher. This is due to the main focus of TAPS being to obtain the best possible performance while adding few parameters, always having slightly higher computational complexity and using considerably more parameters than the baseline for a single domain, and BA^2 can reduce the computational complexity of the base model, but it always uses all the parameters.

The main contributions of this work were proposing a method for multi-domain learning competitive with other state-of-the-art methods, while offering good trade-offs between classification performance and computational cost according to user needs, and proposing a metric that takes into consideration both computational complexity and number of parameters for MDL. As far as we know, our approach is the first capable of reducing both the amount of parameters and computational complexity while tackling multiple domains at once.

IV. CONCLUSION

The main objective of this Ph.D. thesis was to make deep learning methods more efficient, allowing them to be

deployed in a wider array of devices, including those with limited memory and computational capabilities. To achieve these goals, new methods to reduce the cost of two crucial aspects of deep learning in computer vision tasks were studied and proposed: data preprocessing and parameter optimization.

To reduce data preprocessing costs, models trained in information easily available in compressed data were used, avoiding the cost of decoding every image before feeding it to the network. Previous work in the literature made modifications to the network that increased its computational complexity, going in the opposite direction of the desired goal of reducing the overall computational cost. To deal with this problem, handcrafted and data-driven preprocessing techniques were proposed to reduce the amount of input channels and computational complexity of the model, allowing to take full advantage of the speed-up obtained by not decoding the images.

The proposed approaches were effective in reducing the computational complexity and number of parameters of the network, while retaining similar accuracy, making the models more efficient. Learning how to combine all input channels in a data-driven fashion performed better than manually selecting them. Skipping some stages of the network was beneficial. This work directly resulted in three papers, published at CIARP'21 [25], ICIP'20 [24], and SIBGRAPI'20 [18].

For parameter optimization, multi-domain learning was explored, obtaining a single compact model for multiple domains. Methods in literature were capable of reducing the computational complexity of the model, but were not able to decrease the amount of parameters, since the domains share few parameters, this way, the entire model needs to be used for tackling all domains. With that limitation in mind, the encouragement of sharing of parameters among different domains was proposed, allowing the pruning of parameters unused by any domain. To encourage parameter-sharing, three loss functions taking into consideration combinations of the masks for each domain were proposed.

Results indicate that the proposed strategies were crucial for encouraging parameter-sharing, being one of the only models capable of tackling multi-domain learning while also having a lower amount of parameters than the baseline model for a single domain. The accuracy of models was competitive with other state-of-the-art methods while offering good trade-offs between classification performance and computational cost according to user needs. This research was developed during a doctoral internship with a CAPES PSDE scholarship at the University of Trento, Italy, and resulted in the publication of a paper at ICIAP'23 [26].

Metrics that take into consideration the classification performance, computational complexity, and amount of parameters were also proposed for a single domain and MDL, allowing for analysis of the effectiveness-efficiency trade-off of the models.

By using the strategies proposed in this Ph.D. thesis to reduce the cost of preprocessing and share parameters across multiple domains/tasks, the goals of generating more efficient deep models, and easing their deployment in computationally limited devices were successfully achieved.

ACKNOWLEDGMENT

This work was supported by São Paulo Research Foundation - FAPESP (grants 2023/17577-0 and 2024/04500-2), FAPESP-Microsoft Research Institute (grant 2017/25908-6), Brazilian National Council for Scientific and Technological Development - CNPq (grants 310330/2020-3, 314868/2020-8, 315220/2023-6, and 420442/2023-5), Coordination for the Improvement of Higher Education Personnel - CAPES (grant 88881.624512/2021-01), LNCC via resources of the SDumont supercomputer of the IDeepS project, MUR PNRR project FAIR (PE00000013) funded by the NextGenerationEU and EU H2020 project AI4Media (No. 951911).

AWARDS AND PUBLICATIONS

Part of this Ph.D. thesis was developed during a doctoral internship with a CAPES PSDE scholarship (88881.624512/2021-01) under the supervision of Professor Nicu Sebe, leader of the Multimedia and Human Understanding Group (MHUG) at the University of Trento, Italy, resulting in the publication of a paper at ICIAP'23 [26].

The contributions from this Ph.D. thesis resulted in the following publications:

- S. F. Santos, R. Berriel, T. O. Santos, N. Sebe, and J. Almeida, "Budget-aware pruning for multi-domain learning," in International Conference on Image Analysis and Processing (ICIAP'23), 2023 [26].
- S. F. Santos and J. Almeida, "Less is more: Accelerating faster neural networks straight from JPEG," in Iberoamerican Congress on Pattern Recognition (CIARP'21), 2021 [25].
- S. F. Santos and J. Almeida, "Faster and accurate compressed video action recognition straight from the frequency domain," in Conference on Graphics, Patterns and Images (SIBGRAPI'20), 2020, pp. 1–7 [18].
- S. F. Santos, N. Sebe, and J. Almeida, "The good, the bad, and the ugly: Neural networks straight from JPEG," in IEEE International Conference on Image Processing (ICIP'20), 2020, pp. 1896–1900 [24].
- J. G. C. Presotto, S. F. Santos, L. P. Valem, F. A. Faria, J. P. Papa, J. Almeida, and D. C. G. Pedronette, "Weakly supervised learning based on hypergraph manifold ranking," in Journal of Visual Communication and Image Representation, vol. 89, p. 103666, 2022 [29].
- M. D. S. Miranda, L. F. A. e Silva, S. F. Santos, V. A. de Santiago Junior, T. S. Körting, and J. Almeida, "A high-spatial resolution dataset and few-shot deep learning benchmark for image classification," in Conference on Graphics, Patterns and Images (SIBGRAPI'22), 2022, pp. 19–24 [30].

Extended versions of the ICIP'20 and ICIAP'23 papers have been submitted to the Neurocomputing and Pattern Recognition journals, respectively, and are currently under review. Preliminary versions have been deposited on ArXiv [31], [32].

REFERENCES

[1] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2019.

[2] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4244–4268, 2022.

[3] B. Deguerre, C. Chatelain, and G. Gasso, "Fast object detection in compressed JPEG images," in *IEEE Intelligent Transportation Systems Conference (ITSC'19)*, 2019, pp. 333–338.

[4] Y. Li, S. Gu, L. V. Gool, and R. Timofte, "Learning filter basis for convolutional neural network compression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5623–5632.

[5] M. Ehrlich and L. S. Davis, "Deep residual learning in the JPEG transform domain," in *IEEE International Conference on Computer Vision (ICCV'19)*, 2019, pp. 3484–3493.

[6] A. Marchisio, M. A. Hanif, F. Khalid, G. Plastiras, C. Kyrkou, T. Theocharides, and M. Shafique, "Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges," in *IEEE Computer Society Annual Symposium on VLSI (ISVLS'19)*, 2019, pp. 553–559.

[7] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems*, 2017, pp. 506–516.

[8] R. Berriel, S. Lathuillere, M. Nabi, T. Klein, T. Oliveira-Santos, N. Sebe, and E. Ricci, "Budget-aware adapters for multi-domain learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 382–391.

[9] Y. Du, Z. Chen, C. Jia, X. Li, and Y.-G. Jiang, "Bag of tricks for building an accurate and slim object detector for embedded applications," in *International Conference on Multimedia Retrieval (ICMR'21)*, 2021, pp. 519–525.

[10] B. Deguerre, C. Chatelain, and G. Gasso, "Object detection in the DCT domain: is luminance the solution?" in *IEEE Int. Conf. on Pattern Recog. (ICPR'20)*, 2021, pp. 2627–2634.

[11] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Analyzing and mitigating jpeg compression defects in deep learning," in *IEEE/CVF Int. Conf. on Comput. Vis. Workshops (ICCVW'21)*, 2021, pp. 2357–2367.

[12] X. Wang, Z. Zhou, Z. Yuan, J. Zhu, G. Sun, Y. Cao, Y. Zhang, and K. Sun, "Fd-cnn: A frequency-domain fpga acceleration scheme for cnn-based image processing applications," *ACM Trans. on Embedded Comput. Syst. (TECS)*, 2022.

[13] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Annual Conference on Neural Information Processing Systems (NIPS'18)*, 2018, pp. 3937–3948.

[14] S.-Y. Lo and H.-M. Hang, "Exploring semantic segmentation on the DCT representation," in *Proceedings of the ACM Multimedia Asia*, 2019, pp. 1–6.

[15] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the Frequency Domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1740–1749. [Online]. Available: https://github.com/PSCLab-ASU/Learning-in-the-Frequency-Domain

[16] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Quantization Guided JPEG Artifact Correction," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020.

[17] S. F. Santos, N. Sebe, and J. Almeida, "CV-C3D: action recognition on compressed videos with convolutional 3d networks," in *SIBGRAPI – Conference on Graphics, Patterns and Images (SIBGRAPI'19)*, 2019, pp. 24–30.

[18] S. F. Santos and J. Almeida, "Faster and accurate compressed video action recognition straight from the frequency domain," in *Conference on Graphics, Patterns and Images (SIBGRAPI'20)*, 2020, pp. 1–7.

[19] B. Rajesh, M. Javed, S. Srivastava *et al.*, "DCT-CompCNN: A novel image classification network using JPEG compressed DCT coefficients," in *IEEE Conf. on Information and Communication Technol. (CICT'19)*, 2019, pp. 1–6.

[20] Y. Tang, X. Zhang, X. Hu, S. Wang, and H. Wang, "Facial expression recognition using frequency neural network," *IEEE Trans. on Image Process. (IEEE TIP)*, vol. 30, pp. 444–457, 2020.

[21] Y. He, W. Chen, Z. Liang, D. Chen, Y. Tan, X. Luo, C. Li, and Y. Guo, "Fast and accurate lane detection via frequency domain learning," in *ACM Int. Conf. on Multimedia (ACM-MM'21)*, 2021, pp. 890–898.

[22] A. Deshpande, V. V. Estrela, and P. Patavardhan, "The dct-cnn-resnet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the resnet50," *Neuroscience Informatics*, vol. 1, no. 4, p. 100013, 2021.

[23] S. F. Santos, "Aprendizado profundo eficiente para classificação de imagens: Reduzindo o custo de pré-processamento e otimizando parâmetros," Ph.D. dissertation, Universidade Federal de São Paulo (UNIFESP). Instituto de Ciência e Tecnologia, 2023.

[24] S. F. Santos, N. Sebe, and J. Almeida, "The good, the bad, and the ugly: Neural networks straight from jpeg," in *IEEE International Conference on Image Processing (ICIP'20)*, 2020, pp. 1896–1900.

[25] S. F. Santos and J. Almeida, "Less is more: Accelerating faster neural networks straight from jpeg," *Iberoamerican Congress on Pattern Recognition (CIARP'21)*, 2021.

[26] S. F. Santos, R. Berriel, T. O. Santos, N. Sebe, and J. Almeida, "Budget-aware pruning for multi-domain learning," in *International Conference on Image Analysis and Processing (ICIAP'23)*, 2023.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 770–778.

[28] M. Wallingford, H. Li, A. Achille, A. Ravichandran, C. Fowlkes, R. Bhotika, and S. Soatto, "Task adaptive parameter sharing for multi-task learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 7561–7570.

[29] J. G. C. Presotto, S. F. Santos, L. P. Valem, F. A. Faria, J. P. Papa, J. Almeida, and D. C. G. Pedronette, "Weakly supervised learning based on hypergraph manifold ranking," *Journal of Visual Communication and Image Representation*, vol. 89, p. 103666, 2022.

[30] M. D. S. Miranda, L. F. A. e Silva, S. F. Santos, V. A. de Santiago Júnior, T. S. Körting, and J. Almeida, "A high-spatial resolution dataset and few-shot deep learning benchmark for image classification," in *Conference on Graphics, Patterns and Images (SIBGRAPI'22)*, 2022, pp. 19–24.

[31] S. F. Santos, N. Sebe, and J. Almeida, "Cnns for jpegs: A study in computational cost," *arXiv preprint arXiv:2012.14426*, 2023.

[32] S. F. Santos, R. Berriel, T. Oliveira-Santos, N. Sebe, and J. Almeida, "Budget-aware pruning: Handling multiple domains with less parameters," *arXiv preprint arXiv:2309.11464*, 2023.