Pattern Recognition Letters 116 (2018) 29–35

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Classifying low-resolution images by integrating privileged information in deep CNNs

Marion Chevalier[b,a,*], Nicolas Thome[a], Gilles Hénaff[b], Matthieu Cord[a]

[a] UPMC Sorbonne Universités, 4 place Jussieu, 75005 Paris, France
[b] Thales Optronique S.A.S., 2 avenue Gay-Lussac, 78990 Elancourt, France

ARTICLE INFO

Article history:
Received 22 December 2017
Available online 5 September 2018

Keywords:
Image classification
Deep convolutional neural networks
Learning using privileged information.

ABSTRACT

As introduced by [1], the privileged information is a complementary datum related to a training example that is unavailable for the test examples. In this paper, we consider the problem of recognizing low-resolution images (targeted task), while leveraging their high-resolution version as privileged information. In this context, we propose a novel framework for integrating privileged information in the learning phase of a deep neural network. We present a natural multi-class formulation of the addressed problem, while providing an end-to-end training framework of the internal deep representations. Based on a detailed analysis of the state-of-the-art approaches, we propose a novel loss function, combining two different ways of computing indicators of an example's difficulty, based on its privileged information. We experimentally validate our approach in various contexts, proving the interest of our model for different tasks such as fine-grained image classification or image recognition from a dataset containing annotation noise.

1. Introduction

During the last decades, image classification has occupied a prominent place in the work of the Computer Vision and Machine Learning communities. In this paper, we tackle the problem of image classification using privileged information. In a military context, visual sensors may be embedded in airborne systems in order to recognize vehicles at a long range. In this context, during the acquisition of training examples, the images may be acquired at different ranges depending on the situation of the airborne system. However, during the test phase, only images aquired at a long range (*i.e.* low resolution images) are available. Learning Using Privileged Information (LUPI) is a particularly appropriate framework for integrating complementary data. For instance, [2] explore an image classification problem where the training images are associated to a textual description. In our context, we focus on classifying low-resolution images while having access to the high-resolution versions of the training images. In a Machine Learning context, [1] define the privileged information (PI) as a supplementary datum related to an example, that is not necessarily of the same nature, and is unavailable for test examples. This complementary element brings additional information to enhance the learning of the desired decision function. To tackle this issue, several methods have already been proposed in various contexts. PI has been integrated in various classification approaches, such as metric learning [3], ranking approaches [2], SVM classifiers [1,4–6], or structural SVM for object localization [7]. These methods are however largely based on shallow SVM methods, and do not provide a framework for learning a deep model using this complementary information.

During the past few years, deep convolutional neural networks (CNNs) have successfully tackled a vast majority of computer vision tasks, such as object detection [8,9], action recognition [10], or semantic segmentation [11]. In particular, since the breakthrough of the AlexNet of [12], numerous innovations have been proposed to successfully improve the image classification performances [13–16]. Furthermore, these architectures have successfully been used in a transfer fashion in various contexts [13,17,18], which enables using deep pre-trained CNNs on medium-scale datasets. CNNs have also proven to be good candidates for low-resolution image classification [19].

In this paper, we propose DeepLUPI, a novel end-to-end training model for integrating PI into a deep CNN. Our DeepLUPI structures the addressed problem of having several image resolutions during the training phase in a framework for learning a deep CNN using privileged information. Our model combines two major LUPI approaches, using the PI both as an absolute and a relative difficulty

* Corresponding author at: UPMC Sorbonne Universités, 4 place Jussieu, 75005 Paris, France.
E-mail address: chevalier.marion.cm@gmail.com (M. Chevalier).

https://doi.org/10.1016/j.patrec.2018.09.007
0167-8655/

indicator between the targeted and privileged spaces. Besides, our model directly tackles the multi-class formulation of the classification tasks. We experimentally validate our approaches on various contexts, including a protocol similar to that of [1], as well as on a large web-crawled dataset and in fine-grained contexts, hence proving the efficiency and scalability of our deep multi-class approach. We additionally provide a thorough study of different aspects of our model.

2. Previous work

2.1. Learning Using Privileged Information (LUPI)

As introduced by [1], the privileged information, a complementary training datum, may be of various types, like attributes [2,20], textual description [1,21], depth information [22].

In [6], the authors propose Margin Transfer (MT), an algorithm that first optimizes a classifier in the privileged space, then uses the classification score as an indicator of the difficulty for each example. This coefficient is then introduced in the optimization criterion of a binary SVM learned in the targeted space, enforcing a large classification margin for the easiest examples and reducing the enforced margin for the most difficult ones. Their approach relies on the hypothesis that an example which is difficult to recognize in the privileged space is all the more difficult to classify in the targeted space, and may even be an outlier. This method uses an *absolute* difficulty level, only determined by the PI, and transfered unchanged to the targeted space. However, this formulation is based on the SVM approach, and is not adapted to other classification methods such as deep neural networks. Besides, their formulation is based on a binary SVM approach, which is a strong limitation when dealing with a large number of labels.

Based on a complementary approach, [1] - which first introduced a framework for integrating PI during a system's training phase - use the PI as a *relative* difficulty indicator, comparing scores in both spaces. In their model, the PI is used as a proxy to the slack variables in the targeted space, and no classification constraint is enforced in the privileged space: when training the SVM classifier in the targeted space, the slack variables ξ_i are replaced by the score $\langle w^*, x_i^* \rangle + b^*$, with (w^*, b^*) resp. the weight vector and bias term in the privileged space. The weight vectors in both the privileged and the targeted spaces are learned in a joint manner, contrarily to Margin Transfer, which requires sequentially learning two SVM classifiers. Moreover, while Margin Transfer [6] enforces that the examples are correctly classified in the privileged space, SVM+ [1] does not formulate any constraint on the classification of the PI. Accordingly, while MT relies on an absolute difficulty definition, SVM+ relies on a comparison between the scores in both spaces, resulting in a relative difficulty level. It is worth noticing that, like for Margin Transfer, the SVM+ formulation is an enhancement of the binary SVM model. As for MT, the learned model is shallow, and the multi-class recognition tasks are not to be directly addressed by this model.

Complementary approaches have been proposed, enforcing a form of resemblance constraint between the privileged and targeted spaces. For instance, the Loss Inequality Regularization model (LIR) of [20] relies on the intuition that the privileged score represents the maximal reachable score for the targeted classifier: if the targeted score ever gets better than the privileged score for a given example, then this indicates that the targeted space overfits over that example. The system is then penalized whenever the targeted loss is lower than the privileged loss. This approach has however only been tested on shallow SVM models. Moreover, it only considers the PI as a regularization element, which may be a somehow limited usage of these complementary data. Complementary works have also focused on more theoretical properties of LUPI formulations [23,24].

In this paper, we propose DeepLUPI for integrating the PI in deep neural networks, naturally resulting in a multi-class formulation, and enabling learning all the internal deep representations in an end-to-end training fashion.

2.2. Convolutional neural networks

Convolutional Neural Networks (CNN) have been a leading method on a large majority of image recognition tasks, such as image classification [12,15,16,25], action recognition [10], object localization [8,9,26,27].

To process small- or medium-scale datasets, pre-trained CNNs can be used in a transfer fashion [13,17,18]. The weights are pre-trained on a large external dataset (*e.g.* ImageNet [28]), then the first layers of the CNN are considered as a feature extractor on the target dataset, and a classifier is learned using these deep features.

In this paper, we tackle the problem of low-resolution image classification with access to the high-resolution version of the training images. In this context, we propose DeepLUPI, an end-to-end method for classifying low-resolution images by integrating PI in a deep CNN. Our model combines both MT absolute approach of the difficulty level of each example, and the relative difficulty definition of SVM+ comparing the scores in both spaces. Moreover, through our model, we provide a deep multi-class approach, particularly adapted to fine-grained classification.

3. DeepLUPI model

Our model consists in a two-fold loss formulation. The first part of our loss, mainly based on the same approach as [6], computes an weighting coefficient based exclusively on the classification scores in the privileged space, giving an important weight to the examples that are the most correctly recognized in the privileged space. On the other hand, the second part of our loss function adds a criterion that controls the difference between the classification scores in the privileged and the targeted spaces, thus increasing the influence of the examples that are not enough well classified yet in the targeted space. Our model relies on the difficulty transfer from the privileged space to the targeted space - containing the low-resolution images. This transfer seems relevant in our context, since the privileged information is the high-resolution version of each training image, while the targeted information is the low-resolution image. As stated in [19], the resolution loss occludes the discriminant details, whereas the high-resolution images still contain substantially more relevant information. In this context, the hypothesis that the difficulty level of an example can be directly transfered from the privileged space to the targeted space is particularly well verified. Moreover, given the nature of the PI, the outliers should be extremely difficult to recognize in the privileged space, whereas the easiest examples should be the most representative of their class. Thus, an example is said to be easy if it is well classified with a sufficiently high score in the privileged space, while the most difficult examples are the ones with a too low score or even not recognized by the classifier learned in the privileged space.

3.1. DeepLUPI architecture

We present on Fig. 1 the global framework of our DeepLUPI algorithm. Our approach contains two phases: first, a multi-class recognition model is learnt in the privileged space (green part in the figure). This system enables computing the privileged score for each example i for their ground-truth class c: $\tilde{z}_c^*(x_i^*)$. For each example, we also compute a ρ_i coefficient interpreted as the diffi-

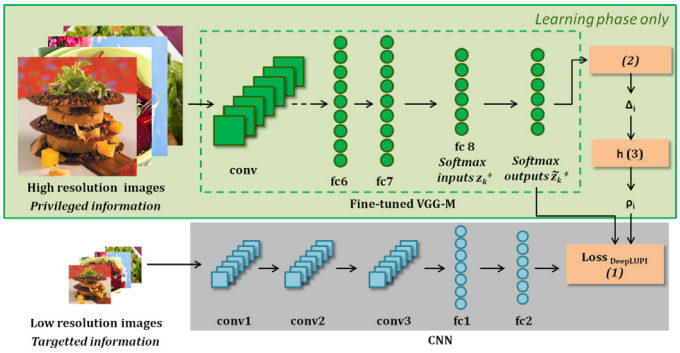

Fig. 1. Architecture of our DeepLUPI model. For each example, the CNN learned on the privileged data (green) enables computing a ρ_i coefficient, characterizing the difficulty of the example, as well as its ground-truth privileged score $\tilde{z}_c(x_i^*)$. Both these quantities are integrated in the loss function $loss_{DeepLUPI}$ of a CNN learned on the targeted data (grey). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

culty of recognizing this example. More precisely, ρ_i is high for an easy example, and low for a difficult one. This coefficient may be qualified as an *absolute* vision of the difficulty level, for it only depends on the privileged space.

A second multi-class recognition system is then learned in the targeted space (grey part). This system benefits from the PI through a new loss function $loss_{DeepLUPI}$ that integrates both these quantities in the targeted CNN learning. More specifically, the error for each example is weighted by the ρ_i coefficient. The easiest (resp. most difficult) examples have thus a higher (resp. lower) influence on the targeted model learning. A second term adds to this approach, measuring the proximity between the ground-truth scores in both spaces. The aim of this *relative* term is to penalize the examples whose targeted score is too low with respect to their privileged score. The intuition behind this complementary penalization is that the PI is richer than the targeted information, which means it should enable learning a more efficient model. Encouraging the targeted model to copy the privileged model's answers can thus be beneficial. More specifically, we penalize the examples whose targeted ground-truth score $\tilde{z}_c(x_i)$ is too low compared to their privileged ground-truth score $\tilde{z}_c^*(x_i^*)$.

In the following, we present the different steps of our DeepLUPI model, detailing the loss function $loss_{DeepLUPI}$, which integrates the absolute and relative difficulty levels of each example in the learning phase of the targeted CNN. The optimization process is summarized in Algorithm 1. We also specify the ρ_i computation, expressing the absolute difficulty level of each example. We also discuss the relation of our model with other state-of-the-art LUPI methods in Section 3.3.

3.2. DeepLUPI computational blocks

3.2.1. Loss function

Our DeepLUPI model uses the PI to compute two difficulty terms taking part in the targeted CNN optimization criterion. The ρ_i coefficient measures an absolute difficulty level given by the privileged space. This coefficient enables differentiating the example difficulties according to their associated privileged representation. The intuition of our model is to assign an important weight to the examples that are the most representative of their class - *i.e.* the easiest according to their privileged representation -, and decrease the impact of the examples with the most uncommon features - *i.e.* the most difficult ones according to the privileged model -, which are hence the most likely to be outliers.

A second term measuring the relative difficulty between both spaces further penalizes an example as long as it is too badly classified. This complementary penalization must however vanish whenever the example gets well enough classified, *i.e.* when its targeted ground-truth score is high enough. More specifically, for each example, we want to bring closer the targeted score from the privileged score. This new term can then be considered as a relative difficulty term between both spaces. As such, the loss function writes as follows:

$$loss_{DeepLUPI} = \sum_{i=1}^{N} -\Big(\rho_i + \gamma[\tilde{z}_c^*(x_i^*) - \tilde{z}_c(x_i)]_+\Big) \ln\left(\tilde{z}_c(x_i)\right) \quad (1)$$

where $\tilde{z}_c^*(x_i^*)$ (resp. $\tilde{z}_c(x_i)$) is the privileged (resp. targeted) softmax output for the ground-truth class c, and $[.]_+$ denotes the function $\max(., 0)$. γ is a trade-off hyper-parameter between both terms. A high value of γ gives a high importance to the relative cost term.

The term $[\tilde{z}_c^*(x_i^*) - \tilde{z}_c(x_i)]_+$ contributes in the loss function only if $\tilde{z}_c(x_i) \leq \tilde{z}_c^*(x_i^*)$, *i.e.* when example i is worse recognized in the targeted space than in the privileged space. This term is proportional to the difference between both scores. For a given example i, the lower the targeted score with respect to the privileged space, the more influence this example has on the learning of the network. We thus meet the intuition of SVM+, which strongly penalizes the examples not recognized with a sufficient margin, given by the PI.

3.2.2. Deep multi-class ρ_i coefficients

As illustrated in Fig. 1, our model relies on multi-class ρ_i coefficients computed in the privileged space. These coefficients are then integrated into the loss function of the CNN learned in the targeted space, weighting the individual error of each example. These coefficients measure the difficulty level of each example: a low value of ρ_i is associated to a difficult example, while the highest ρ_i values are associated to the examples that are best recognized in the privileged space. In this paper, we propose to learn a CNN in the privileged space, then to compute the coefficients as follows:

$$\Delta_i = \tilde{z}_c(x_i^*) - \max_{k \neq c} \tilde{z}_k(x_i^*) \quad (2)$$

where $\tilde{z}_k(x_i^*)$ is the softmax output for class k for image x_i^* of ground-truth class c in the privileged space. The scores $\tilde{z}_k(x_i^*)$ are computed after a softmax, which means they necessarily lie between 0 and 1, hence the quantity Δ_i lies between -1 and 1. The easiest examples thus have an associated $\Delta_i \simeq 1$, and the most difficult ones may have a $\Delta_i < 0$. In order to ensure that all the weigths are in $[\tau_{\min}, \tau_{\max}]$ with $\tau_{\min} > 0$, we remap the Δ_i through a function h:

$$\rho_i = h(\Delta_i). \quad (3)$$

3.3. Discussion

Several LUPI approaches are based on this intuition of similitude between both spaces.

Our DeepLUPI model can be seen as an extension of both SVM+ [1] and Margin Transfer [6] models, by jointly incorporating an absolute difficulty constraint and a relative difficulty constraint during training.

Our model also echoes the intuition of the LIR model of [20], in the sense that we constrain targeted scores and privileged scores to be related. Our intuition is however in contradiction with that of LIR, which uses the privileged space as an indicator of overfitting. This approach considers that if the targeted model recognizes some of the examples *too well*, then this model might be overfitting on these examples. To prevent this situation, the classification scores in the targeted space are forced to remain inferior to the privileged scores. On the contrary, in our deepLUPI model, the second term of the loss function is meant to use the privileged space to force the examples to have a *sufficiently good* classification score

in the targeted space. In other words, this term forces the targeted model to take into account the examples that are not well enough classified yet.

4. Experimental evaluation

We focus on image classification tasks. To show the interest of our modeling, we compare our model with several state-of-the-art LUPI methods on various image classification tasks.

Datasets. To evaluate our model, we consider two fine-grained oriented datasets: FGVC-Aircraft [29] and UPMC-Food-101 [30]. FGVC-Aircraft is composed of 10,000 images equally distributed both between 100 aircraft variants, and train, validation and test sets. For all our experiments we report the results of the methods learned on the train and validation sets, and tested on the test set. UPMC-Food-101 is a large dataset containing about 100,000 images of 101 different dishes. As in [30], we randomly choose 600 images per class for training, and the remaining for test. This dataset has the particularity of containing annotation noise for it is a web-crawled image dataset. This enables us to measure our model's robustness to this phenomenon. On both fine-grained oriented bases, we aim at recognizing low resolution 32×32 images - which are the targeted data. During the training phase, we also have the high resolution 224×224 versions of these images.

In order to further validate our model, we also explore the image recognition framework on MNIST [31] proposed by [1] and explored in several other works [21,23,32]. This framework aims at recognizing the 10×10 reduced MNIST images, while having the 28×28 images as the privileged information. On this dataset, we use a standard LeNet, which is why the 10×10 images are further magnified to 28×28 to match the input size of the network. In their experiments, the authors tackle the binary problem of discriminating the two classes of digits "5" and "8". However, our approach naturally calls to treating the complete 10-class MNIST problem. We use the train / test repartition proposed by [31].

Experimental protocol. We compare three types of approaches: the targeted features classified by a linear multi-class SVM - *i.e.* without PI -, the state of the art LUPI methods, and our DeepLUPI model.

In this paper, we compare our model with the major state of the art LUPI methods, both based on SVM classifiers: SVM+ [1] (using online code in [5][1]), and Margin Transfer [6] (using LibLinear [33] to implement both SVMs). In order to provide results on very recent state-of-the-art methods, we have also implemented and tested both LIR [20] and Generalized Distillation [23] methods, yet since the performances of these models were not satisfying, we only report their performances on MNIST.

For the shallow methods (targeted features, SVM+ [1] and MT [6]), we use features extracted from a LeNet learned from scratch on the targeted images on MNIST, and features extracted from the first fully-connected layer of a VGG-M pre-trained on ImageNet for FGVC-Aircraft and UPMC-Food-101. As privileged representations, we use the features extracted from a LeNet learned from scratch on the privileged images on MNIST, and we fine-tune a VGG-M pre-trained on ImageNet for the 224×224 images of both fine-grained datasets.

For our DeepLUPI model, on each dataset, we use LeNet for MNIST, and the LR-CNN of [19] on both fine-grained oriented datasets. As privileged features, we use a LeNet learned on the privileged images from MNIST. On UPMC-Food-101, we reinitialize the weights of the last fully-connected layer of a VGG-M in order to get the correct number of outputs - *i.e.* the number of classes - then we fine-tune the weights of all the layers. We use a similar protocol on FGVC-Aircraft. However, since this dataset only contains a restricted number of training images, the gradient of the error is only backpropagated through the last fc layer, to prevent overfitting.

[1] http://www.cs.technion.ac.il/~pechyony/

Table 1
Comparison of our DeepLUPI method with state-of-the-art LUPI methods. On both fine-grained datasets we report the multi-class accuracy, while on MNIST we report the number of mistakes. *: results obtained with the online code provided by the authors.

Method	MNIST (# err)	UPMC-Food-101 (acc)	FGVC-Aircraft (acc)
Target feature w/o PI	130	28.9%	32.7%
SVM+ [1]*	122	-	31.2%
MT [6]*	119	30.6%	33.5%
LIR [20]	120	-	-
Generalized Distillation [23]	110	-	-
DeepLUPI (ours)	**95**	**31.9%**	**39.8%**

Algorithm 1 DeepLUPI learning.

Require: $(x_i^*, x_i, y_i)_{i=1.N}$
1: Learn a CNN on the privileged data $(x_i^*, y_i)_{i=1.N}$
2: Compute the ρ_i (2,3)
3: Initialize the targeted weights **w**
4: **for** epoch $= 1.\text{epoch}_{\max}$ **do**
5: On targeted data $(x_i, y_i)_{i=1.N}$, compute $loss_{DeepLUPI}$ (1)
6: Update weights **w** by backpropagating the gradient of $loss_{DeepLUPI}$
7: **end for**

4.1. Comparison with state-of-the-art methods

We report in Table 1 the results for all the above described methods. On MNIST, our DeepLUPI model makes 24 fewer mistakes than MT, and 27 fewer than SVM+. We show that our approach for taking into account the example difficulty is particularly more efficient than the tested state-of-the-art LUPI methods in this context. Moreover, on this dataset, the LIR method of [20] makes 120 errors, *i.e.* 25 more than our model. This method relies on a similar formulation to our DeepLUPI model for it compares the costs in both spaces, yet it is based on the opposite intuition to our model: the example must be better recognized in the privileged space than in the targeted space. The more recent Generalized Distillation approach of [23] leads to 110 mistakes in this context, which means 15 more mistakes than our model. Their method enforces a mimic constraint between the outputs from the targeted and the privileged models. This experiment shows that our approach is more adapted to the problem in this context.

On UPMC-Food-101, our method improves of 1.3% the performances of Margin Transfer[2] One should notice that UPMC-Food-101 is a fine-grained oriented multi-class dataset, thus containing a large number of classes to discriminate. The state-of-the-art binary LUPI methods Margin Transfer and SVM+ are not particularly adapted to this kind of task, whereas our method enables directly treating the multi-class problems. Moreover, this dataset contains a lot of training images, which is easily dealt with by our method, whereas the state-of-the-art methods such as SVM+ are not adapted at all to this kind of situation. Finally, this dataset

[2] We do not report any result for SVM+, since the online available code uses an optimization in the dual space, which is not suited to the important number of training images on UPMC-Food-101.

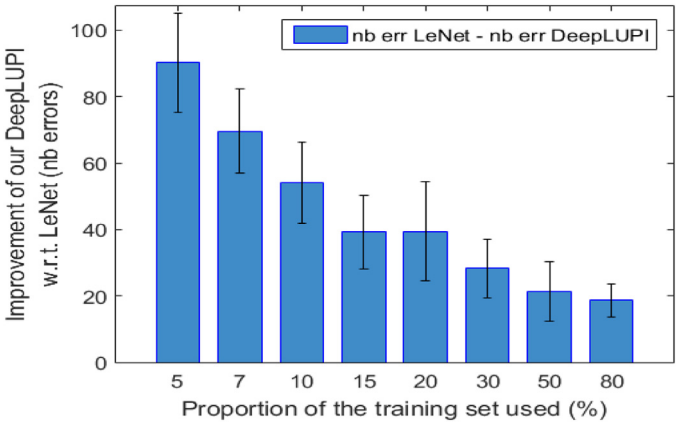

Fig. 2. Improvement of our DeepLUPI model w.r.t. the proportion of the training set on MNIST.

contains a certain level of annotation noise, which our model obviously achieves better dealing with than Margin Transfer.

Finally, on FGVC-Aircraft, our method also outperforms both shallow concurrent models, improving by 6.3% the performances of Margin Transfer and by 8.6% that of SVM+. A part of this performance gain is due to the use of a deep model in the targeted space, while the concurrent methods are based on a shallow SVM classifier. The possibility of using a deep network enables our model to improve these models' performances.

Our DeepLUPI model outperforms all the tested state-of-the-art LUPI methods on the three datasets. We prove that our end-to-end learning method actually takes advantage of the privileged information to improve the internal representations in the targeted network, contrarily to shallow SVM-based LUPI methods such as Margin Transfer or SVM+. Our model also directly takes into account the multi-class aspect of the addressed problems, while the major state-of-the-art LUPI methods require an *ad hoc* binary reformulation.

4.2. Impact of the size of the training set

We also show the interest of our model in the case where a few training examples are available. To this end, we compare deep and DeepLUPI performances when reducing the number of training images. More specifically, we run eight independent experiments, each of them using a similar protocol to that of [1]: for each experiment, p percent of the training images are randomly chosen in each class, then both CNNs are learned on this training subset. Both networks are based on the same structure and learning parameters as previously. They are then tested on the whole test set (10,000 images). For each of the eight experiments, this process is repeated on 12 randomly chosen training subsets. On Fig. 2, we report for each proportion p the mean and standard deviation values for $nb\_errors\_deep - nb\_errors\_DeepLUPI$. This value is then high when LeNet makes much more mistakes than our model, *i.e.* when our DeepLUPI model performs better than the LeNet without PI.

On this figure, we show that our DeepLUPI model enables improving the performances of the CNN without PI whatever the proportion of the training images used. For instance, with 20% of the training set, our DeepLUPI makes in average 39.4 fewer mistakes than a LeNet. Moreover, the standard deviation values reported indicate that these results are significant.

This figure also shows that the performance gap between DeepLUPI and LeNet grows when the number of training images decreases. Indeed, with 50% of the images, DeepLUPI improves by 21.3 mistakes the performances of the LeNet; with only 7% of the training images, DeepLUPI improves by 69.6 mistakes the performances of LeNet. This observation meets the intuitions and conclusions of [1] stating that the privileged information proves even more informative when the training set is small. These results, showing that our method largely improves the performances of the standard CNN without privileged information, offer an interesting perspective in the current context of deep CNN learning requiring a lot of training data.

Fig. 3. Relevance of the ρ_i values computed on UPMC-Food-101 images.

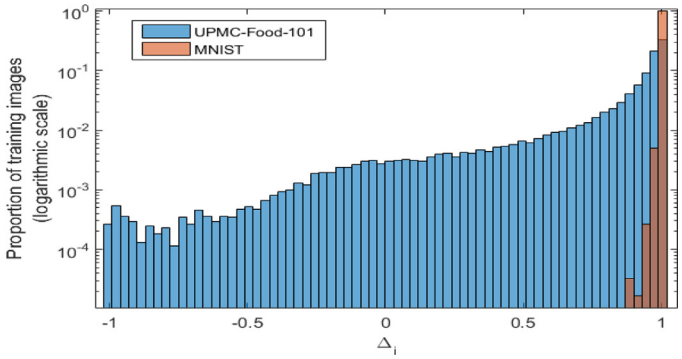

Fig. 4. Δ_i distribution on UPMC-Food-101 (blue) and MNIST (orange). Note: for more readability, the distributions are presented on a logarithmic scale. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. DeepLUPI further analysis

5.1. ρ_i analysis

We focus here on the relevance of the ρ_i coefficients. This study aims at highlighting the capacity of our model to discriminate the examples most representative of their class from the most difficult ones, possibly outliers, that a human eye would struggle to recognize properly. For four random classes of UPMC-Food-101, we report on Fig. 3 the examples associated with the lowest (resp. highest) Δ_i value on the bottom (resp. top) line. We show that the images associated to the highest ρ_i values are quite easily recognizable, whatever the class. On the contrary, the images associated with the lowest ρ_i values are more difficult to categorize, since they may be unusual forms of the dish (*e.g.* the greek salad resembles a bruschetta dish), or even difficult outliers (*e.g.* a pizza image categorized as nachos).

We are now interested in showing the capacity of our model to reveal the difficulty level of a given dataset, by studying the distribution of the Δ_i on different datasets. We present on Fig. 4 the repartition of the Δ_i values on UPMC-Food-101 (blue) and MNIST (orange). On this figure, we show that on a quite easy dataset such as MNIST, all the Δ_i have a high value, *i.e.* all the images are identified as easily recognized. Indeed, on this dataset, all the Δ_i are superior to 0.85. On UPMC-Food-101 however, we show that the Δ_i are much more distributed. This dataset is constituted of

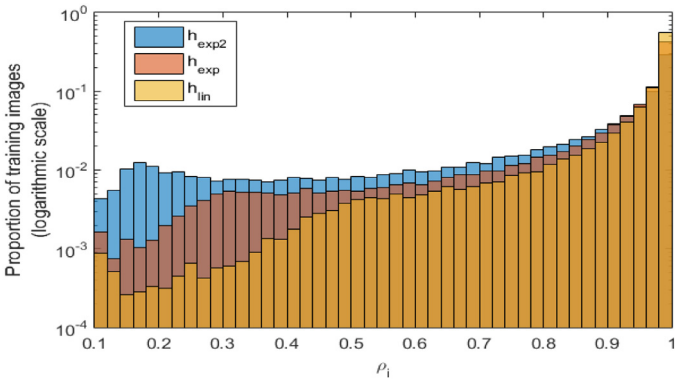

Fig. 5. Influence of the different remapping types on the ρ_i distribution on UPMC-Food-101. Note: for more readability, the distributions are presented on a logarithmic scale.

Table 2
Analysis of the improvement brought by our DeepLUPI model.

Method	MNIST	UPMC-Food-101
Deep	130 errors	30.7%
DeepLUPI, $\gamma = 0$	107 errors	31.6%
DeepLUPI, $\rho_i = 0$	135 errors	31.1%
DeepLUPI	**95 errors**	**31.9%**

web-crawled images, which is the reason why it contains a certain amount of annotation noise, *i.e.* not easily recognized images and outliers. Here, the Δ_i are quite spread between both bounds; more than 2000 images have a negative Δ_i value.

We present on Fig. 5 the ρ_i distribution after the different remapping functions h_{lin} (orange), $h_{\exp}$ (red) and h_{exp2} (blue). h_{lin} denotes a linear remapping, $h_{\exp}$ denotes an exponential remapping, and h_{exp2} stands for a remapping where two exponential functions are successively applied to the data. For each remapping, the final ρ_i are in [0.1; 1]. On this figure, we show that the non linear remapping functions enable better distributing the ρ_i between both bounds.

5.2. DeepLUPI ablations

In order to highlight the improvement of each part of our DeepLUPI model, we focus on the improvement brought by only one of both difficulty terms. On this purpose, we consider our DeepLUPI model with all the ρ_i values at 0 - only the relative cost is taken into account -, then with $\gamma = 0$ - only the absolute cost is taken into account. The baseline consists in a standard CNN learnt on the low resolution images without PI.

All the results are reported in Table 2. On MNIST, when only taking into account the ρ_i term (*i.e.* when $\gamma = 0$), our DeepLUPI model makes 13 fewer errors than the CNN without PI. Adding the relative cost (*i.e.* with our complete DeepLUPI model) further improves by 12 errors this result. On UPMC-Food-101, the absolute difficulty alone ($\gamma = 0$) also improves de performances w.r.t. a standard CNN, by 0.9%. Adding the relative cost further improves this result by 0.3%. This lesser gap may be explained by the consistence of the ρ_i absolute difficulty coefficients, which already bring a more significant improvement.

When taking $\rho_i = 0$, this approach enables improving the performances w.r.t. a standard CNN on UPMC-Food-101. Since this dataset contains an annotation noise, mislabeled examples may disturb the learned targeted CNN without PI. This protocol thus enables relaxing the constraint on the outliers, and thus could explain the performance gain on a noisy dataset such as UPMC-Food-101. On MNIST, the standard CNN achieves somewhat better performances than the relative cost alone. This result tends to prove that this relative cost alone does not improve the performances on a dataset without outlier. However, the results of our DeepLUPI model show that the information carried by this relative cost is complementary to that of the absolute difficulty term, since the combination of both improve the performances on all the datasets.

Finally, when using the complete DeepLUPI formulation, the performances are improved on both MNIST and UPMC-Food-101 datasets compared with a standard deep CNN approach. Indeed, DeepLUPI misclassifies 35 fewer images than a standard deep CNN (without privileged information) on MNIST, and improves by 1.2% the classification performances on UPMC-Food-101. This result tends to show that the introduction of a deep structure in the LUPI framework instead of shallow models is not the only element contributing to the improvement achieved by our DeepLUPI model.

To conclude, we show that our DeepLUPI model enables improving the performances over the standard CNN in most cases. Moreover, the absolute cost - incarnated by the ρ_i term - always achieves improving the performances in all cases. The relative cost alone of our DeepLUPI leads to performances consistent with the nature of the different datasets.

6. Conclusion

In this paper, we present DeepLUPI, a novel framework for integrating privileged information for training a deep neural network, providing a natural end-to-end training framework of the internal representations as well as a multi-class formulation for the addressed problem. Our model leverages a novel loss formulation, optimizing a combination of both absolute and relative approaches of the difficulty level of each example. We experimentally validate our approach on several datasets, proving its interest in various challenging contexts such as fine-grained oriented image classification or image recognition from a dataset containing noisy labels. Furthermore, we propose an extensive experimental analysis of the different aspects of our model, showing the consistency of our approach as well as its robustness to various context changes, especially when the number of training images decreases. These results may be particularly interesting in a deep learning context, where most methods require a large amount of training data. Further works include developing a joint optimization process between both spaces.

Declarations of interest: none.

References

[1] V. Vapnik, A. Vashist, A new learning paradigm: Learning using privileged information, Neural Networks, 2009.
[2] V. Sharmanska, N. Quadrianto, C.H. Lampert, Learning to rank using privileged information, ICCV, 2013.
[3] S. Fouad, P. Tino, S. Raychaudhury, P. Schneider, Incorporating privileged information through metric learning, IEEE Transactions on Neural Networks and Learning Systems (TNNLS), 2013.
[4] H. Li, Y. Li, F. Porikli, Deeptrack: learning discriminative feature representations by convolutional neural networks for visual tracking, in: British Machine Vision Conference (BMVC), 2014.
[5] D. Pechyony, V. Vapnik, Fast optimization algorithms for solving svm+, Chapter in Statistical Learning and Data Science, 2011.
[6] V. Sharmanska, N. Quadrianto, C.H. Lampert, Learning to transfer privileged information, arXiv:1410.0389, 2014.
[7] J. Feyereisl, S. Kwak, J. Son, B. Han, Object localization based on structural svm using privileged information, NIPS, 2014.
[8] R. Girshick, Fast r-cnn, ICCV, 2015.
[9] S. Bell, C.L. Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, CVPR, 2016.
[10] G. Gkioxari, R. Girshick, J. Malik, Contextual action recognition with r*cnn, ICCV, 2015.
[11] X. Chen, C.L. Zitnick, Minds eye: A recurrent visual representation for image caption generation, CVPR, 2015.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS, 2012.
[13] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, in: British Machine Vision Conference (BMVC), 2014.
[14] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale visual recognition, in: International Conference on Learning Representations (ICLR), 2015.
[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CVPR, 2015.
[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[17] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in: CVPR DeepVision Workshop, 2014, pp. 512–519.
[18] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, CVPR, 2014.
[19] M. Chevalier, N. Thome, M. Cord, J. Fournier, G. Hénaff, E. Dusch, Lr-cnn for fine-grained classification with varying resolution, ICIP, 2015.
[20] Z. Wang, Q. Ji, Classifier learning with hidden information, CVPR, 2015.
[21] W. Li, D. Dai, M. Tan, D. Xu, L.V. Gool, Fast algorithms for linear and kernel svm+, CVPR, 2016.
[22] J. Hoffman, S. Gupta, T. Darrell, Learning with side information through modality hallucination, CVPR, 2016.
[23] D. Lopez-Paz, L. Bottou, B. Schölkopf, V. Vapnik, Unifying distillation and privileged information, in: International Conference on Learning Representations (ICLR), 2016.
[24] D. Pechyony, V. Vapnik, On the theory of learning using privileged information, NIPS, 2010.
[25] N. van Noord, E.O. Postma, Learning scale-variant and scale-invariant features for deep image classification, Pattern Recognit. 61 (2017) 583–592.
[26] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, NIPS, 2015.
[27] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, NIPS, 2016.
[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, CVPR, 2009.
[29] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-Grained Visual Classification of Aircraft, Technical Report, 2013.
[30] X. Wang, D. Kumar, N. Thome, M. Cord, F. Precioso, Recipe recognition with large multimodal food dataset, in: IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2015.
[31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.
[32] M. Lapin, M. Hein, B. Schiele, Learning using privileged information: SVM+ and weighted SVM, in: Neural Networks, 53, 2014, pp. 95–108.
[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification 9 (2008) 1871–1874. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear/.