

# A neural network strategy for supervised classification via the Learning Under Privileged Information paradigm

Ludovica Sacco<sup>1</sup>, Dino Ienco<sup>2,4</sup> and Roberto Interdonato<sup>3,4</sup>

<sup>1</sup>DIMES - University of Calabria, P. Bucci 41C, 87036 Rende (CS), Italy

<sup>2</sup>INRAE, UMR TETIS, Univ. Montpellier, Montpellier, France

<sup>3</sup>CIRAD, UMR TETIS, Montpellier

<sup>4</sup>TETIS, Univ. of Montpellier, APT, Cirad, CNRS, INRAE, Montpellier, France

## Abstract

Devising new methodologies to handle and analyse Big Data has become a fundamental task in our increasingly service-oriented and interconnected society. One of the problems arising while handling such data is that, for a given set of entities, not all the entities may be described at the same level of detail, i.e., the number of features describing each entity may vary. In general cases, in order to apply classic data science methods, it is necessary to have a common features set over a data set. This will then correspond to the maximum number of common features among the entities, resulting in a loss of information for the entities for which additional information may be available. In order to exploit such additional information, the Learning Using Privileged Information (LUPI) paradigm has been proposed, based on the use of the teacher role in the learning process. In this schema the teacher acquires a strategic position, by exploiting at the training stage some additional privileged information about the entities, which will not be available at the test stage. In this work, we apply this paradigm in the context of neural networks, by proposing a LUPI based deep learning architecture able to exploit a larger set of attributes at training time, with the aim to improve classification performances on a set of entities associated to a reduced attribute set. Experimental results show how the proposed approach improves upon the ones applying the same schema to classic machine learning methods (e.g., SVM).

## Keywords

Big Data, LUPI paradigm, Neural Networks

## 1. Introduction


The developments achieved in recent years in machine learning, due both to theoretical studies and to the constant increase of computational power of the processors, have led to several advancements on the traditional learning techniques. Moreover, these years have also seen a sudden increase in the quantity of available data. Massive quantities of information are produced, collected and digitally stored everyday, in contexts that touch different domains


---

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

✉ l.sacco@dimes.unical.it (L. Sacco); dino.ienco@inrae.fr (D. Ienco); roberto.interdonato@cirad.fr (R. Interdonato)

ORCID 0000-0003-0235-6273 (L. Sacco); 0000-0002-8736-3132 (D. Ienco); 0000-0002-0536-6277 (R. Interdonato)

 © 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

and aspects of human life (human health and behavior, agriculture, biology, and so on). The availability of these so called *big data*, together with that of machines that can handle higher volumes of data than ever before, have also led to an increase in the use of supervised learning techniques, i.e., techniques that rely on labeled data in order to train a machine learning model. However, in practical contexts, these data are generally noisy, and may include a diversity of features that are not always easy to exploit for computational analyses. A typical scenario is that in which entities of the same type are associated to a non homogeneous set of attributes that describe them. When using traditional machine learning methods, this would require the need to reduce the set of the attributes exploited to learn the model to just the common ones, thus resulting in a significant loss of information.

One of the solutions that have been proposed in order to overcome this problem, and allow the use of such diverse set of attributes, is the Learning Using Privileged Information paradigm (LUPI) paradigm [1]. The idea behind LUPI is to introduce a teacher-student model in the learning process, that allows to train a model on a larger set of attributes than that available at testing time. More specifically, the idea is to select a subset of entities for which a larger set of attributes is available (i.e., the *privileged information*), and then to train a model that can use such information to discern among entities that are associated to a reduced set of attributes (i.e., entities for which the privileged information is not available, but just a limited set of *regular* attributes). The original implementations of the LUPI paradigm were introduced on classic machine learning techniques, e.g., Support Vector Machines (SVM) [1]. However, this paradigm can also be integrated in more advanced frameworks, such as the ones based on deep learning architectures [2], that are nowadays the state of the art in several domains.

In this paper, we present a LUPI based deep learning approach based on a three-branch architecture where a first model (M1) is trained on the full set of attributes (regular and privileged information), a second one (M2) exploits the representation learnt by M1 in order to stretch the representation learnt from regular attributes towards the one obtained from privileged ones, and a third model (M3) is optimized on the regular attributes only. The final feature set is then obtained by combining the output of M2 and M3. While the proposed architecture is implemented by using a classic Multi-Layer Perceptron to instantiate the three models, it is general enough to be extended to different neural network models (e.g., Convolutional Neural Networks and Recurrent Neural Networks), and to be effectively applied to data coming from different domains. Experimental evaluation on six real world datasets show how the proposed methodology improves upon a competitor that applies the LUPI paradigm to an SVM approach (SVM+). The paper is organized as follows: Section 2 discusses the related work, section 3 introduces the proposed architecture, Section 4 discusses experimental results, while Section 5 concludes the work.

## 2. Related Work

The first algorithm that implements the LUPI paradigm is an extension of the classic SVM algorithm, known as SVM+ [1, 3]. SVM+ proposes a modified formulation of SVM in order to evaluate the misclassification of the training example through a function that learns from the privileged information as slack variables. Successively, different improvements and alternatives

have found place into supervised learning problems. In [4] it has been remarked that the LUPI paradigm has always been implemented with the L-2 support vector machine, with the result of the number of the tuning parameters doubled and a consequent increase of the computational cost. So the decision to employ the L-1 norm, widely used for feature and variable selection [5], brought to the creation of L-1 SVM and its use in LUPI. Some amelioration to the optimization problem of the SVM+ is provided with the presentation of two new SMO-style algorithms [6]: aSMO and gSMO. When large training sets are used, both show to be more efficient in terms of generalization error/running treade-off than the generic optimizer LOQO, based on the interior point optimizers. Further similar algorithms can be found in [7], with some comparison between aSMO and caSMO. Considerations on differentiating easy examples from the difficult ones are discussed and formalized in [8], where a new approach in choosing weights associated with every training example is presented. More recent works [9, 10] discuss the importance of the knowledge transfer concept into SVM+ and neural networks to improve their convergence properties and therefore accelerating the speed of Student's learning.

Privileged information plays an important role in domains like computer vision, where several studies have successfully introduced the LUPI paradigm [11, 12]. Other applications are based on the use of bounding-boxes, image captions and descriptions as additional information and combine it with the Rank Transfer techniques to overcome the limitations of the SVM+ [13]. In recent years, other studied use the privileged information often associated to neural networks and exploited to solve various tasks. Classifying fine-grained images or recognizing images from a data set containing annotation noise is possible thanks to the DeepLUPI framework [14]. In [15] a two-stream fully convolutional network, named MIML-FCN+, is proposed to solve the multi-instance multi-label problem by using privileged bags of labels. A LUPI framework for CNNs and RNNs is offered in [16], where a heteroscedastic dropout is used and the privileged information is represented by the variance of the dropout. In this work, while we exploit a simple Multi-Layer Perceptron to instantiate the models at the base of our architecture, we propose a deep learning framework that is general enough to be extended to different neural network models (e.g., Convolutional Neural Networks and Recurrent Neural Networks), and to be effectively applied to data coming from different domains.

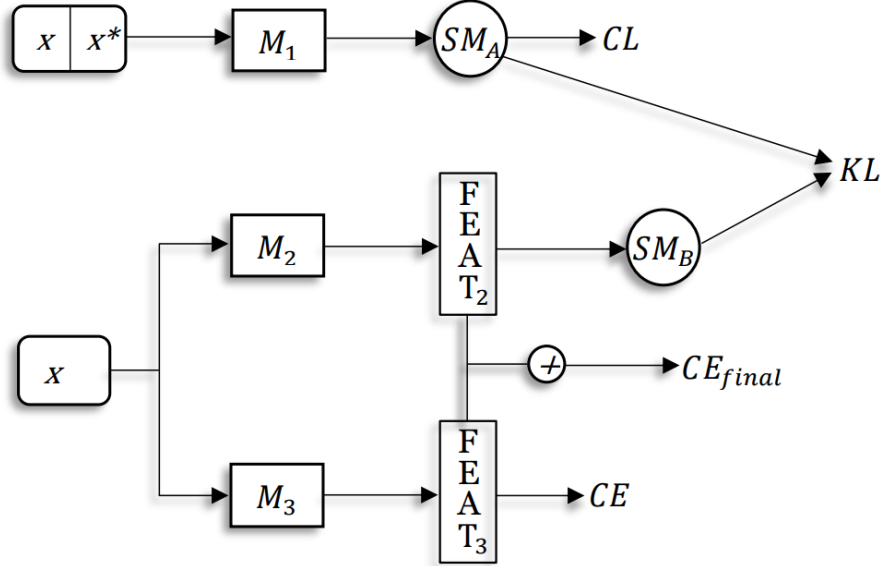
### 3. The LUPI Architecture

#### 3.1. Learning Using Privileged Information Paradigm

In the classic machine learning paradigm, the role of a teacher in the human learning process has been underestimated. The supervised learning task considers a simple strategy. Given a set of pairs  $(x_1, y_1), \dots, (x_l, y_l), x_i \in X, y_i \in \{-1, 1\}$ , where the vector  $x_i \in X$  is the description of the example and  $y_i$  its classification, finding the function  $y = f(x, \alpha^*)$  that minimizes the probability of incorrect classifications. In the LUPI paradigm the goal is exactly the same, but it contemplates the knowledge provided by the teacher at the training stage, the so called Privileged Information. So triplets  $(x_1, x_1^*, y_1), \dots, (x_l, x_l^*, y_l), x_i \in X, x_i^* \in X^*, y_i \in \{-1, 1\}$  are now considered instead of pairs and each one is independently generated by some underlying unknown distribution  $P(x, x^*, y)$ .

### 3.2. Implementation

The proposed LUPI based deep learning architecture (Figure 1) is an end-to-end framework consisting of three neural networks (e.g. multi-layer perceptron) that process the input data to produce the final classification.



**Figure 1:** The general overview of the proposed LUPI based architecture. The privileged information is identified by  $X^*$  and the regular training example by  $X$ . The architecture has three different streams, two of which ( $M_2$  and  $M_3$ ) run in parallel, and their outputs are combined to get the final classification.

The first stream analyzes the training examples  $X$  (i.e., the entity with the associated regular information) with the addition of the privileged information  $X^*$ , while the other two streams work only on the training example  $X$ . This choice allows to exploit the knowledge acquired when the privileged information is available. In particular, the distribution of the labels got as an output from the execution of the first model ( $M_1$ ) is channeled into the second neural network. From this configuration, the second model ( $M_2$ ) uses as input a set of pairs  $(x_i, y_p)$  where  $x_i \in X$  and  $y_p \in \{-1, 1\}$  and  $y_p$  corresponds to the output obtained from  $M_1$ . Moreover, to train the second neural network we choose the Kullback Leibler divergence [17] to let  $M_2$  behaves as similar as possible to  $M_1$ . Finally, the third stream manages the training examples without any additional information to help during the classification. It should be noted that the  $M_1$  model is executed separately from the other two, and that  $M_2$  and  $M_3$  are run in parallel, as part of a single multi-output neural network.

Regarding the loss function the Categorical Cross Entropy has been used, except for  $M_2$ , where the Kullback Leibler divergence is employed with the aim to mimic the behavior of the  $M_1$  model trained with the privileged information.

## 4. Experiments

This section reports the experimental setup, the results achieved and the data used to evaluate our architecture. To evaluate the proposed LUPI architecture, we carried out the experiments on six dataset, from different domains. We describe how we preprocessed them for the utilization in the experimental phase. We introduce setup details for both our implementation and competitors, specifying the metrics adopted for the comparison. Finally we discuss the results, aiming to validate our architecture.

### 4.1. Datasets

Six well-known dataset were used in the experiments :

- COIL20 [18]: image dataset including 20 different objects. For each of them 72 images are included in the size 128x128. The background has been discarded.
- HAR [19]: sensors data, built through a smartphone, from the recording of 30 subjects, performing various activities of daily living and containing six different activities: walking, walking upstairs, walking downstairs, sitting, standing and laying.
- Isolet [20]: 150 subjects spoke the name of each letter of the alphabet twice. Hence, there are 52 training examples from each speaker.
- landsat [21]: dataset of agricultural land images in Australia to classify seven different soil classes constituting dissimilar soil types.
- USPS [22]: 9298 images belonging to 10 different classes with size 16x16.
- waveform [23]: generator generating 3 classes of waves. Each class is generated from a combination of 2 to 3 base waves.

Structural characteristics of these datasets are summarized in Table 1.

**Table 1**

Datasets used to perform the experiments.

	# Features	# Objects	# Classes	Type
COIL20	1024	1440	20	Image
HAR	561	10299	6	Sensor
Isolet	617	1560	26	Speech
landsat	36	2859	7	Image
USPS	256	9298	10	Image
waveform	40	5000	3	Numbers

### 4.2. Setup

In order to evaluate the significance of the proposed approach, we provide an experimental evaluation including tests on different percentages of privileged information and a comparison with the performance of the SVM+ algorithm [1, 3]. For training and evaluation purposes, the datasets were divided into three parts: training, validation and test set, respectively representing

the 50%, the 20% and the 30% of the instances (i.e., *rows* of the dataset). In line with recent deep learning literature, the model obtaining the best performance on the validation set has been used for the evaluation on the test set. The privileged information is provided by selecting a portion of the available features for each dataset, i.e., a certain number of *columns*. In detail, indices are randomly shuffled and then 9 different percentages of attributes are taken into account: 5%, 10%, 25%, 30%, 50%, 60%, 70%, 80%, 90%. The indices are selected in an incremental fashion, so that the higher percentages contain the indices included in the lower ones. Bear in mind that the percentage of privileged and “regular” information are complementary, so that for a certain percentage  $x$  of privileged information, the number of attributes available at testing time will be the  $(100 - x)\%$  of total attributes. In our setting, this means that higher percentages of privileged information will correspond to harder test stages, because lower quantities of information will be available at testing time. Obviously this assumption is only related to our need to “artificially” partition the available attributes in privileged and regular, and does not hold in real world cases, where higher quantities of privileged information should correspond to better performances.

In order to avoid the bias induced by this selection process, these operations are executed 10 times, in order to get 10 different configurations of privileged indices for each considered percentage of privileged information. In relation to the deep learning parameters, the number of epochs is set to 100 and the learning rate of the Adam optimizer is equal to  $1 \times 10^{-4}$  [24]. While the metrics identified to evaluate our architecture are *Accuracy* and *F-measure*.

Regarding the internal structure of the LUPi architecture, the three models, M1, M2 and M3 are built using a multi-layer perceptron neural network (MLP). Each MLP has an input, three internal layers and an output layer, with respectively 256, 192, 128, 64 neural units and the number of classes of each dataset as the units for the output layer. As the activation function, the Rectified Linear Unit (ReLU) is employed for each layer, a part from the output one which utilized the SoftMax function to normalize the output vector values so that their sum is equal to 1. The experiments are run 30 times for each configuration of the different privileged indices, then mean and standard deviation are taken into account to evaluate the results. The reported experiments are carried out on the following platform: Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz 2.71 GHz with 8,00 GB of RAM. The LUPi based deep learning architecture is implemented in Python Tensorflow 2.0 library. The source code for SVM+ is written in C and it is available online <sup>1</sup>.

### 4.3. Experimental Results

Table 2 and table 3 report the average result in term of Accuracy and F-Measure for the 6 different datasets. It can be noted that the proposed approach has better performances for percentages of privileged information lower or equal to 30%. This result is actually not surprising since, as explained in Section 4.2, higher percentages of privileged information correspond to harder test stages (i.e., lower quantities of information are available at testing time). For higher percentages of privileged information, performances are slightly lower, but still comparable for most datasets, showing the robustness of the proposed approach even in the cases when only 10% of the total attributes are available at testing time (i.e., 90% of privileged information).

---

<sup>1</sup><https://github.com/cypw/svm-plus>

**Table 2**

Average and standard deviation accuracy performance of the proposed LUPI based architecture as the percentage of privileged information in the dataset (columns in the table) increases.

	5%	10%	25%	30%	50%	60%	70%	80%	90%
COIL20	98.17 $\pm$ 0.44	98.19 $\pm$ 0.44	98.18 $\pm$ 0.50	<b>98.21</b> $\pm$ 0.43	98.19 $\pm$ 0.49	98.14 $\pm$ 0.45	98.05 $\pm$ 0.48	98.06 $\pm$ 0.49	97.31 $\pm$ 0.46
HAR	<b>94.15</b> $\pm$ 0.87	94.08 $\pm$ 0.86	93.74 $\pm$ 0.89	93.65 $\pm$ 0.70	93.23 $\pm$ 0.71	92.66 $\pm$ 0.83	92.21 $\pm$ 0.79	91.05 $\pm$ 0.79	86.44 $\pm$ 0.87
Isolet	<b>91.60</b> $\pm$ 0.86	91.51 $\pm$ 0.80	91.08 $\pm$ 0.75	91.02 $\pm$ 0.82	90.11 $\pm$ 0.84	89.43 $\pm$ 0.87	88.30 $\pm$ 0.92	86.81 $\pm$ 0.93	81.01 $\pm$ 1.21
landsat	<b>88.82</b> $\pm$ 0.94	88.60 $\pm$ 0.93	88.14 $\pm$ 0.90	88.02 $\pm$ 0.95	86.97 $\pm$ 0.97	86.41 $\pm$ 0.96	85.57 $\pm$ 1.01	84.78 $\pm$ 0.96	79.34 $\pm$ 0.85
USPS	96.49 $\pm$ 0.26	<b>96.52</b> $\pm$ 0.23	96.44 $\pm$ 0.20	96.42 $\pm$ 0.23	96.34 $\pm$ 0.19	96.21 $\pm$ 0.23	95.87 $\pm$ 0.21	94.79 $\pm$ 0.26	91.72 $\pm$ 0.34
waveform	<b>85.66</b> $\pm$ 0.65	85.36 $\pm$ 0.63	83.54 $\pm$ 0.63	82.95 $\pm$ 0.64	79.85 $\pm$ 0.63	76.95 $\pm$ 0.55	73.94 $\pm$ 0.57	68.91 $\pm$ 0.53	54.50 $\pm$ 0.65

**Table 3**

Average and standard deviation F-Measure performance of the proposed LUPI based architecture as the percentage of privileged information in the dataset (columns in the table) increases.

	5%	10%	25%	30%	50%	60%	70%	80%	90%
COIL20	98.22 $\pm$ 0.49	98.26 $\pm$ 0.49	98.22 $\pm$ 0.55	<b>98.27</b> $\pm$ 0.50	98.25 $\pm$ 0.53	98.19 $\pm$ 0.50	98.14 $\pm$ 0.52	98.13 $\pm$ 0.53	97.39 $\pm$ 0.53
HAR	<b>94.02</b> $\pm$ 0.95	93.97 $\pm$ 0.92	93.65 $\pm$ 0.95	93.54 $\pm$ 0.77	93.08 $\pm$ 0.78	92.57 $\pm$ 0.88	92.09 $\pm$ 0.85	90.95 $\pm$ 0.89	86.37 $\pm$ 0.95
Isolet	<b>91.90</b> $\pm$ 0.92	91.83 $\pm$ 0.82	91.40 $\pm$ 0.81	91.35 $\pm$ 0.85	90.45 $\pm$ 0.94	89.82 $\pm$ 0.92	88.67 $\pm$ 1.00	87.10 $\pm$ 0.96	81.34 $\pm$ 1.27
landsat	<b>86.51</b> $\pm$ 1.48	86.15 $\pm$ 1.52	85.56 $\pm$ 1.45	85.37 $\pm$ 1.58	83.79 $\pm$ 1.63	82.99 $\pm$ 1.65	81.63 $\pm$ 1.88	80.29 $\pm$ 1.91	73.70 $\pm$ 1.56
USPS	96.04 $\pm$ 0.31	<b>96.08</b> $\pm$ 0.31	96.02 $\pm$ 0.18	96.03 $\pm$ 0.22	95.92 $\pm$ 0.22	95.88 $\pm$ 0.24	95.54 $\pm$ 0.36	94.32 $\pm$ 0.33	90.89 $\pm$ 0.44
waveform	<b>85.63</b> $\pm$ 0.70	85.34 $\pm$ 0.68	83.48 $\pm$ 0.67	82.91 $\pm$ 0.70	79.77 $\pm$ 0.65	76.83 $\pm$ 0.60	73.64 $\pm$ 0.70	67.63 $\pm$ 0.87	51.07 $\pm$ 1.61

Actually, the only dataset showing a significance degradation in performances for higher percentages of privileged information is *waveform*, which is also the one corresponding to the lowest performances for all configuration. A reason for this behavior may be found in its synthetic nature. On the contrary, the dataset who produces the best results is *COIL20*, which is the dataset including the highest number of features, also corresponding to a relatively low number of objects. This probably allows to obtain a better classification, thanks to the large amount of available information at the training time over a relatively small numbers of tuples. As a general observation, all the other datasets show good Accuracy and F-Measure results, confirming the effectiveness of the proposed approach on datasets coming from different application domains.

Table 4 shows the average Accuracy results for the SVM+ algorithm. It can be noted how in most cases the performances begin to degrade already for a *PI* around the 10%, making SVM+

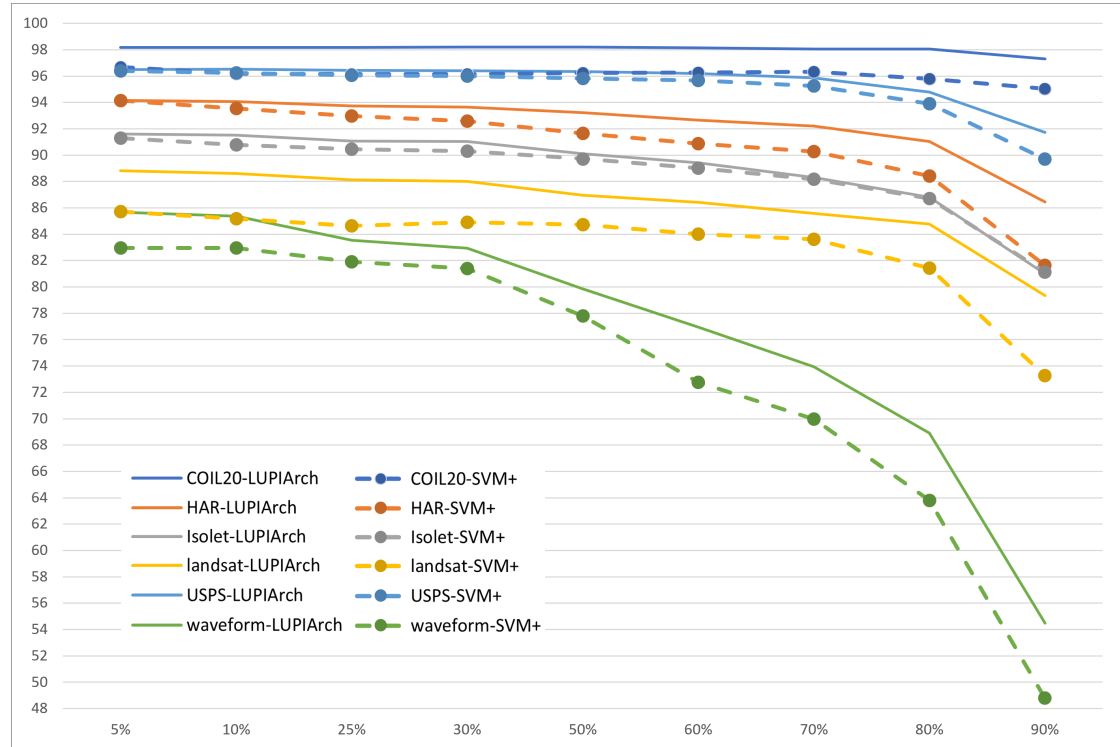


**Table 4**

Average accuracy performance of the SVM+ as the percentage of privileged information in the dataset (columns in the table) increases.

	5%	10%	25%	30%	50%	60%	70%	80%	90%
COIL20	<b>96.70</b>	96.17	96.16	96.14	96.23	96.28	96.33	95.80	95.03
HAR	<b>94.15</b>	93.55	92.97	92.58	91.65	90.86	90.27	88.43	81.63
Isolet	<b>91.30</b>	90.79	90.45	90.30	89.72	89.02	88.18	86.71	81.13
landsat	<b>85.73</b>	85.19	84.65	84.92	84.74	84.02	83.61	81.42	73.27
USPS	<b>96.38</b>	96.25	96.07	96.01	95.81	95.67	95.26	93.90	89.71
waveform	<b>82.96</b>	82.95	81.97	81.40	77.81	72.77	69.97	63.79	48.81

significantly less robust to our approach. To ease the comparison between the two approaches, in Figure 2 we present a visual comparison between our architecture and SVM+ in terms of Accuracy. It is evident how the LUPI architecture dominates the scene, clearly outperforming SVM+ in most datasets. The only two datasets in which the performances are not clearly superior (remain still comparable) are USPS and Isolet.

**Figure 2:** Accuracy comparison between our LUPI architecture and the SVM+ algorithm

The scenario described so far reflects the reality, regarding the Teacher-Student metaphor on which the LUPI paradigm is based. Too high percentages of *PI* would not allow the student



to develop such skills to adequately face the testing phase. While percentages of up to 30% represent the right measure to develop one's own method and acquire such knowledge as to be able to build precise classification models. Our architecture proves to be able to fully perform this task, providing a valid structure for an optimal learning up to 30% of *PI*, also allowing to improve the performance in terms of Accuracy compared to the SVM+ algorithm.

## 5. Conclusions

In this work, we introduced a new deep learning architecture based on the Learning Using Privileged Information (LUPI) paradigm. The LUPI paradigm allows to exploit extra information that may be available only for a subset of the total objects in a dataset, i.e., the so called *privileged* information. More specifically, a model is trained by exploiting such privileged information, which is then able to classify test examples associated only to regular one. Experimental performances on six well-known datasets show the significance of our approach, proving its robustness to different available quantities of privileged and regular information. Experimental results also showed how the proposed deep learning architecture outperforms a competitor integrating the LUPI paradigm on a classic machine learning method (SVM+).

While in this work, in order to test the significance of our approach, we resorted to an artificial partitioning of the available information between privileged and regular one, in the future we plan to test the architecture on real-world case studies, corresponding to specific application domains. A first example can be that of remote sensing, when the availability of different sensors on specific areas may be exploited as privileged information (e.g., exploiting drone images associated to a limited geographical area, associated to satellite images available on larger portions of the scene). From a methodological point of view, we also plan to extend our approach by instantiating the three models included in the architectures with different neural networks (e.g., Recurrent and Convolutional ones).

## 6. Acknowledgments

This work was supported by the Regione Calabria, as part of the Ph.D scholarship of Ludovica Sacco.

## References

- [1] V. Vapnik, A. Vashist, A new learning paradigm: Learning using privileged information, *Neural Networks* 22 (2009) 544–557. doi:10.1016/j.neunet.2009.06.042.
- [2] Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, *Nat.* 521 (2015) 436–444. URL: <https://doi.org/10.1038/nature14539>. doi:10.1038/nature14539.
- [3] V. Vapnik, A. Vashist, N. Pavlovitch, Learning using hidden information (learning with teacher), in: *Proceedings of the 2009 International Joint Conference on Neural Networks, IJCNN'09*, IEEE Press, 2009, p. 1252–1259.

- [4] L. Niu, Y. Shi, J. Wu, Learning using privileged information with l-1 support vector machine, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, IEEE, 2012. doi:10.1109/wi-iat.2012.52.
- [5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer New York, 2009. doi:10.1007/978-0-387-84858-7.
- [6] D. Pechyony, R. Izmailov, A. Vashist, V. Vapnik, Smo-style algorithms for learning using privileged information., 2010, pp. 235–241.
- [7] D. Pechyony, V. Vapnik, Fast Optimization Algorithms for Solving SVM+, 2011, pp. 27–42. doi:10.1201/b11429-5.
- [8] M. Lapin, M. Hein, B. Schiele, Learning using privileged information: SVM and weighted SVM, Neural Networks 53 (2014) 95–108. doi:10.1016/j.neunet.2014.02.002.
- [9] V. Vapnik, R. Izmailov, Learning using privileged information: Similarity control and knowledge transfer, Journal of Machine Learning Research 16 (2015) 2023–2049. URL: <http://jmlr.org/papers/v16/vapnik15b.html>.
- [10] V. Vapnik, R. Izmailov, Knowledge transfer in SVM and neural networks, Annals of Mathematics and Artificial Intelligence 81 (2017) 3–19. doi:10.1007/s10472-017-9538-x.
- [11] J. Feyereisl, S. Kwak, J. Son, B. Han, Object localization based on structural svm using privileged information, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 27, Curran Associates, Inc., 2014. URL: <https://proceedings.neurips.cc/paper/2014/file/6c4b761a28b734fe93831e3fb400ce87-Paper.pdf>.
- [12] W. Li, L. Niu, D. Xu, Exploiting privileged information from web data for image categorization, in: Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 437–452. doi:10.1007/978-3-319-10602-1\_29.
- [13] V. Sharmanska, N. Quadrianto, C. H. Lampert, Learning to rank using privileged information, in: 2013 IEEE International Conference on Computer Vision, IEEE, 2013. doi:10.1109/iccv.2013.107.
- [14] M. Chevalier, N. Thome, G. Hénaff, M. Cord, Classifying low-resolution images by integrating privileged information in deep CNNs, Pattern Recognition Letters 116 (2018) 29–35. doi:10.1016/j.patrec.2018.09.007.
- [15] H. Yang, J. Zhou, J. Cai, Y. Ong, Mlml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information, 2017, pp. 5996–6004. doi:10.1109/CVPR.2017.635.
- [16] J. Lambert, O. Sener, S. Savarese, Deep learning under privileged information using heteroscedastic dropout (2018). arXiv:1805.11614.
- [17] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1951) 79–86. doi:10.1214/aoms/1177729694.
- [18] S. A. Nene, S. K. Nayar, H. Murase, Columbia university image library (coil-20) (1996). URL: <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>.
- [19] J. L. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, X. Parra, A public domain dataset for human activity recognition using smartphones, 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2013).
- [20] R. Cole, M. Fanty, Spoken letter recognition, in Advances in Neural Information Processing Systems (1991) 220–226.

- [21] Statlog landsat satellite data set: Machine learning repository, Univ. California, Irvine, Irvine, CA, USA (1993).
- [22] J. Hull, A database for handwritten text recognition research, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (1994) 550–554. doi:10.1109/34.291440.
- [23] L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and regression trees* (1984) 49–55.
- [24] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.