Towards Automated Deep Learning: Analysis of the AutoDL challenge series 2019

Zhengying Liu, Zhen Xu, Shangeth Rajaa, Meysam Madadi, Julio C S Jacques, Sergio Escalera, Adrien Pavao, Sebastien Treguer, Wei-Wei Tu, Isabelle Guyon

Proceedings of Machine Learning Research 123:242–252, 2020 NeurIPS 2019 Competition and Demonstration Track

Towards Automated Deep Learning: Analysis of the AutoDL challenge series 2019

Zhengying Liu ZHENGYING.LIU@INRIA.FR
TAU, LRI-CNRS–INRIA, Université Paris-Saclay, France

Zhen Xu XUZHEN@4PARADIGM.COM
4Paradigm, Beijing, China

Shangeth Rajaa F20160442@GOA.BITS-PILANI.AC.IN
BITS Pilani, India

Meysam Madadi MEYSAM.MADADI@GMAIL.COM
Computer Vision Center and Universitat de Barcelona, Spain

Julio C. S. Jacques Junior JSILVEIRA@UOC.EDU
Universitat Oberta de Catalunya and Computer Vision Center

Sergio Escalera SERGIO@MAIA.UB.ES
Universitat de Barcelona and Computer Vision Center, Spain

Adrien Pavao ADRIEN.PAVAO@GMAIL.COM
TAU, LRI-CNRS–INRIA, Université Paris-Saclay, France

Sebastien Treguer STREGUER@GMAIL.COM
La Paillasse, Paris, France

Wei-Wei Tu TUWWCN@GMAIL.COM
4Paradigm, Beijing, China

Isabelle Guyon GUYON@CHALEARN.ORG
TAU, LRI-CNRS–INRIA, Université Paris-Saclay, France

Editors: Hugo Jair Escalante and Raia Hadsell

Abstract

We present the design and results of recent competitions in Automated Deep Learning (AutoDL). In the AutoDL challenge series 2019, we organized 5 machine learning challenges: AutoCV, AutoCV2, AutoNLP, AutoSpeech and AutoDL. The first 4 challenges concern each a specific application domain, such as computer vision, natural language processing and speech recognition. At the time of March 2020, the last challenge AutoDL is still on-going and we only present its design.[1] Some highlights of this work include: (1) a benchmark suite of baseline AutoML solutions, with emphasis on domains for which Deep Learning methods have had prior success (image, video, text, speech, etc); (2) a novel "anytime learning" framework, which opens doors for further theoretical consideration; (3) a repository of around 100 datasets (from all above domains) over half of which are released as public datasets to enable research on meta-learning; (4) analyses revealing that winning solutions generalize to new unseen datasets, validating progress towards universal AutoML

1. Its results will be presented in future work together with detailed introduction of winning solutions of each challenge.

solution; (5) open-sourcing of the challenge platform, the starting kit, the dataset formatting toolkit, and all winning solutions (All information available at autodl.chalearn.org).
Keywords: Hyper-parameter optimization, AutoML, Deep Learning, AutoDL, Neural architecture search, benchmarks

1. Introduction

Machine Learning (ML) keeps delivering impressive novel applications in our daily lives. But it is still facing enormous challenges, preventing its more universal deployment by users having direct needs but no time or resources to hire ML experts. In fact, even for ML experts, effectively tuning hyper-parameters is still a daunting task, particularly for Deep Learning models, let alone addressing higher level aspects of model design, including problem definition, experimental design, data collection, preprocessing, design of metrics, computation of error bars, detection of bias in data, etc. Certainly, automating the entire modeling pipeline is still a far reaching goal, but the challenges we present in this paper allowed us to make great strides. We present here the results of the Automated Deep Learning (AutoDL) challenge series, addressing tasks in Computer Vision (Liu et al., 2019), Natural Language Processing (NLP), Speech Recognition, etc. With the solutions provided (and open-sourced) by the winners, users must only preprocess data to horseshoe-fit them in a generic tensor format to have automated algorithms train and test Deep Learning neural networks for them. The problems addressed are multi-label classification, in an amazingly broad range of application domains (medical imaging, object or gesture classification, satellite imaging, to name a few). Besides saving human effort, the benefit of such automated solutions include reproducibility and accountability, freeing us potentially from the variability of human solutions and possibly increasing reliability.

The AutoDL challenge series is part of a larger effort on Automated Machine learning (AutoML) with **code submission** in which the solutions of participants are blind tested on the CodaLab challenge platform. In all of our AutoML challenges we seek to enforce learning within a fixed time budget and limited computational resources. One particularity of AutoDL challenges, compared to previous AutoML challenges, is that we seek to enforce **any-time learning**, which encourages solutions performing reasonably good early on in the whole learning process. This is achieved by using the Area under the Learning Curve (ALC) metric, as explained in Section 3. To help participants develop their code, we provide a starting kit in Python with TensorFlow/PyTorch interfaces, sample "public" datasets and sample code submissions. Some basic facts about the challenge series are summarized in Table 1.

While most of our challenges are run in two phases (a **feedback phase** with immediate feedback on a leaderboard on $N = 5$ practice datasets and a **final phase** with a single evaluation on $N = 5$ final test datasets), in AutoCV, we evaluated the participants on the results of the feedback phase, to make it slightly easier. However, we ran privately a final test phase of which we report here the results. Since the 5 AutoCV final phase datasets were not disclosed, we re-used some in subsequent phases. AutoCV2 was run regularly in 2 phases. Even practice datasets during the feedback phase were not revealed to the participants (they were solely visible to their "autonomous agent").

Table 1: **Basic facts on AutoDL challenges**.

Challenge	Begin date 2019	End date 2019-20	#Teams	#Submissions	#Phases
AutoCV1	May 1	Jun 29	102	938	1
AutoCV2	Jul 2	Aug 20	34	336	2
AutoNLP	Aug 2	Aug 31	66	420	2
AutoSpeech	Sep 16	Oct 16	33	234	2
AutoDL	Dec 14	Mar 14	28	80	2

2. Data

In AutoDL challenges, **raw data** (images, videos, audio, text, etc) are provided to participants formatted in a uniform tensor manner (namely TFRecords, a standard generic data format used by TensorFlow). For images with native compression formats (e.g. JPEG, BMP, GIF), we directly use the bytes. Our data reader decodes them on-the-fly to obtain a 4D tensor. Video files in mp4/avi format (without the audio track) are used in a similar manner. For text datasets, each example (i.e. a document) is a sequence of integer indices. Each index corresponds to a word (for English) or character (for Chinese) in a vocabulary given in the metadata. For speech datasets, each example is represented by a sequence of floating numbers specifying the amplitude at each timestamp, similar to uncompressed WAV format. Lastly, tabular datasets' feature vector representation can be naturally considered as a special case of our 4D tensor representation.

For practical reasons, each dataset was kept under 4GB, which required sometimes reducing image resolution, cropping, and/or downsampling videos. We made sure to include application domains in which the scales varied a lot. We formatted around 100 datasets in total and used 61 of them for AutoDL challenges: 16 image, 9 video, 15 text, 15 speech and 6 tabular. The distribution of domain and size is visualized in Figure 1. More basic information/meta-features are presented in Table 3 in Appendix A. All datasets marked "public" can be downloaded on corresponding challenge websites, for example on the Get Data page of AutoDL challenge. All tasks are supervised multi-label classification problems, *i.e.* data samples are provided in pairs $\{X, Y\}$, X being an input 4D tensor of shape (time, row, col, chnl) and Y a target binary vector (withheld from in test data).

3. Evaluation Metrics

AutoDL challenges encourage **any-time learning** by scoring participants with the Area under the Learning Curve (ALC) (see example curves in Figure 2). The participants can train in increments of a chosen duration (not necessarily fixed) to progressively improve *performance*, until the time limit is attained. Performance is measured by the NAUC or *Normalized* Area Under ROC Curve (AUC) $NAUC = 2 \times AUC - 1$ averaged over all classes. Multi-class classification metrics are not being considered, *i.e.* each class is scored independently. Since several predictions can be made during the learning process, this allows us to plot learning curves, *i.e.* "performance" (on test set) as a function of time. Then for each dataset, we compute the Area under Learning Curve (ALC). The time axis

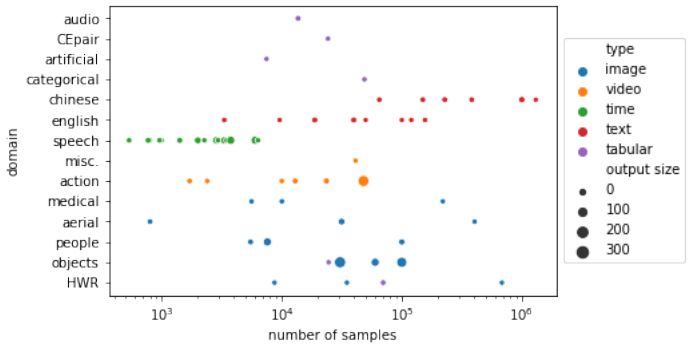


Figure 1: **Distribution of type, size and domain for all 61 AutoDL datasets.**

is log scaled (see the `xlabel`'s in Figure 2 for the formula used for time transformation) to put more emphasis on the beginning of the curve. In this way, we encouraged participants to develop techniques that improve performance rapidly at the beginning of the training process. This should be important to treat large redundant and/or imbalanced datasets and small datasets alike, *e.g.* by treating effectively redundancy in large training datasets or using learning machines pre-trained on other data if training samples are scarce. Finally, in each phase, an overall rank for the participants is obtained by averaging their ALC ranks obtained on each individual dataset. The average rank in the final phase is used to determine the winners.

4. Baselines

As each challenge (except for AutoDL) involves a specific domain, different baselines are provided for different challenges.

4.1. Baselines for AutoCV & AutoCV2

For AutoCV and AutoCV2, we introduced 3 baseline methods with varied complexity and computer resource requirements. **Baseline 0** makes one single all-zero prediction and always gets 0 NAUC score (hence 0 ALC as well). **Baseline 1** is a linear classifier. It uses a cross entropy loss and an Adam optimizer (Kingma and Ba, 2014). If the input shape is variable, resize all images to a fixed shape 112×112. When the number of frames (time axis) is variable, sample 10 consecutive frames at random, both for training and testing. The scheduling strategy is to double the number of training steps at each iteration. Stop training and predicting when time budget is not enough for next iteration. **Baseline 2** uses a neural network architecture determined by the tensor shape of the input examples. More concretely, 3D convolutional layer is repeatedly applied followed by a 3D max-pooling layer, until the number of neurons of the hidden layer is less than a pre-defined number (e.g. 1000), then apply a fully connected layer for classification. More details on these baselines can be found in Liu et al. (2019). Lastly, we also prepared two private baselines with fixed backbone architecture ResNet-50 He et al. (2015) and Inception-V3 Szegedy et al. (2016) but these baselines are only used by the organizers for testing and comparison purpose.

4.2. Baselines for AutoNLP

For AutoNLP, **Baseline 0** uses a Support Vector Machine (SVM) as classifier. The input text is preprocessed by keeping only the alphanumeric characters and been vectorized with the Term Frequency Inverse Document Frequency (TF-IDF) vectorizer with a maximum vocabulary size of 20000. **Baseline 1** follows Kim (2014) and uses a convolutional neural network (CNN). In this method, the text input data is preprocessed like in the previous baseline. The vocabulary is indexed and the integer sequence is then padded with a maximum sequence length. The model architecture consists of an Embedding layer with a dimension of 200, two 1D convolutional layers, a Maxpooling layer, two 1D convolutional layers, a Global Average pooling layer and a fully connected layer. Compared to Baseline 1, **Baseline 2** uses pretrained word embedding weights in addition. This method uses the same preprocessing, vectorization and model architecture from the previous baseline with an embedding layer of dimension 300. The embedding layer weights are loaded with a pretrained embedding from FastText (Bojanowski et al., 2016).

4.3. Baseline for AutoSpeech

The baseline method for AutoSpeech is relatively straightforward. Features are extracted on each dataset using Mel-Frequency Cepstral Coefficients (MFCC) (Mermelstein, 1976), with shape padding. We then apply a CNN backbone model on the extracted preprocessed features, automatically adapting the number of layers according to the number of features. We train only one iteration or perform early stopping at convergence. For prediction, the same MFCC feature preprocessing is applied and use the trained model for inference.

4.4. Baseline for AutoDL

For the final (and still on-going, as of 8 March 2020) AutoDL challenge, we provide a baseline referred to as **Baseline 3**, which is a combination of the winner solutions of AutoCV (*kakaobrain*), AutoNLP (*upwind_flys*) and AutoSpeech (*PASA_NJU*), using domain inference which depends only on the input shape of the 4D tensor. On tabular datasets (which are never used in above challenges), the model chosen is simply a fully connected neural network with 2 hidden layers of 256 neurons.

5. Challenge results

In this section, we present the results and analysis of AutoCV, AutoCV2, AutoNLP and AutoSpeech. Because of space constraints, we cannot provide details on winning solutions. However, the codes have been released and more details will be provided in an extended paper in preparation. We only consider the **top-10 participants in the final phase** of each challenge for all analyses. The names of the top-3 teams can be found in Table 2.

5.1. Learning curves obtained in each challenge

For a given task, we plot all learning curves of top-10 participants in the same figure, for a clear comparison. In Figure 2, four of such figures are shown, each from a different

Table 2: **Top-3 winners** and **Pearson correlation coefficient** between average ranking vectors in feedback phase and final phase, with corresponding p-value. For all challenges except AutoCV2, the Pearson correlation is close to 1 with significant p-value, which means that the feedback phase results and final phase results are consistent, suggesting the generalization ability of these AutoDL methods. For AutoCV2, top-10 participants used very similar approaches (all similar to the solution of *kakaobrain*, in AutoCV), which makes the performances of different teams very close.

Challenge	Top-3 teams	Pearson's r	p-value
AutoCV	*kakaobrain*, *DKKimHCLee*, *base_1*	0.8321	2.836×10^{-3}
AutoCV2	*kakaobrain*, *tanglang*, *kvr*	0.3555	3.133×10^{-1}
AutoNLP	*DeepBlueAI*, *upwind_flys*, *txta*	0.8718	1.010×10^{-3}
AutoSpeech	*PASA_NJU*, *DeepWisdom*, *Kon*	0.8761	8.844×10^{-4}

challenge. From these curves, one can spot very different learning curve patterns, suggesting very different learning and predicting strategies.

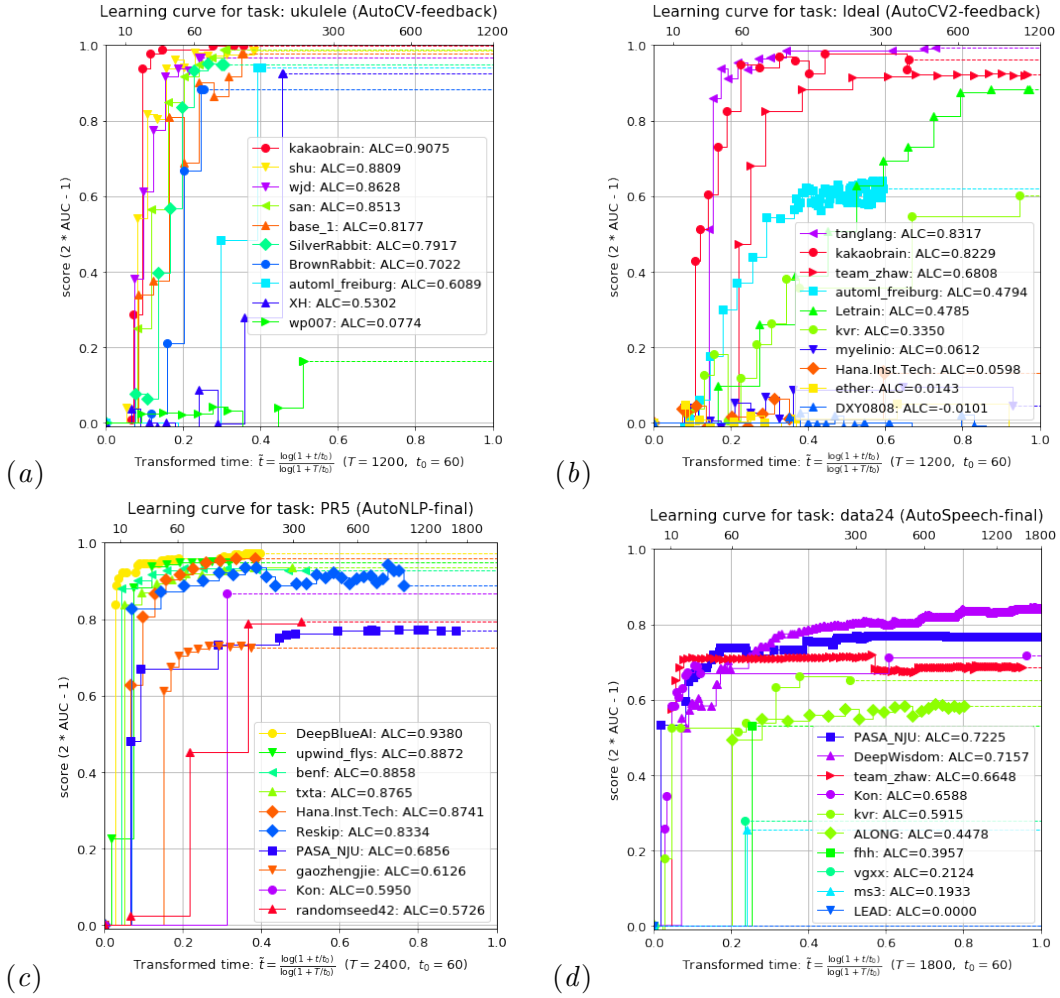
5.2. Generalization ability of AutoML methods

To evaluate the generalization ability of AutoML methods on unseen datasets, we compute the average rank over 5 datasets in both phases and consider their correlation (recall that datasets are DIFFERENT in each phase). This gives an average rank pair $(r_1, r_2) \in \mathbb{R}^2$ for each participant. When r_1 is close to r_2, the method is considered to super-generalize, *i.e.* generalize in the AutoML sense to NEW datasets, not just to a different test set from the same datasets as in common ML challenges. We plot all these pairs (r_1, r_2) in Figure 3. We observe that most participants' rank pairs are close to the diagonal, suggesting generalization ability for most methods. Some outliers such as *LEAD* are due to technical failure of code execution, e.g. with an out of memory (OOM) error. And to evaluate the soundness of the choice of datasets for evaluating generalization, we also compute the Pearson correlation coefficient for all r_1, r_2 in Table 2 using $\rho_{X,Y} = \mathrm{cov}(X, Y)/(\sigma_X \sigma_Y)$.

5.3. Modeling difficulty of datasets

To benchmark the modeling difficulty of each task/dataset, Figure 4 shows the best-vs-worst performances among top-10 participants. We see that many datasets from AutoNLP such as *PR1* to *PR5* are found on the top-right, meaning that the performance variance is small. This could be due to the pre-trained word embedding weights from FastText (Bojanowski et al., 2016) or BERT (Devlin et al., 2018) we provide in a Docker image shared by all participants. With these pre-trained word embedding, even weak classifiers could obtain relatively good performance. On the other hand, datasets from AutoSpeech such as *data11* and *data25* have large modeling difficulty, which might be related to the fact that raw speech datasets often require careful pre-processing steps in order to train a successful classifier.

Figure 2: **Learning curves** for one specific task in each challenge. From these curves, we can already see that the strategies used by participants vary a lot. The number of predictions (i.e. number of points on a learning curve) ranges from 1 (e.g. in (c), *Kon* on *PR5* in AutoNLP final phase) to 789 of *DeepWisdom* on *data24* in AutoSpeech final phase, in (d). And from whether the curve decreases dramatically at some point (e.g. in (a), *base_1* and *XH* on *ukulele*), we can infer whether the submitted method uses a validation set to determine if a prediction should be made.



5.4. Addressing the any-time learning problem

The Figure 5 informs on participant's effectiveness to address the *any-time learning* problem. We first factored out dataset difficulty by re-scaling ALC and NAUC scores (resulting scores on each dataset having mean 0 and variance 1). Then we plotted, for each participant, their fraction of submissions in which ALC is larger than NAUC (FRAC for short) *vs.* $correlation(ALC, NAUC)$ (CORR for short). The participants in the bottom half of the figure did not address well the *any-time learning* problem because their FRAC is lower than 50%. Those participants did not perform well in the challenge either (small symbols).

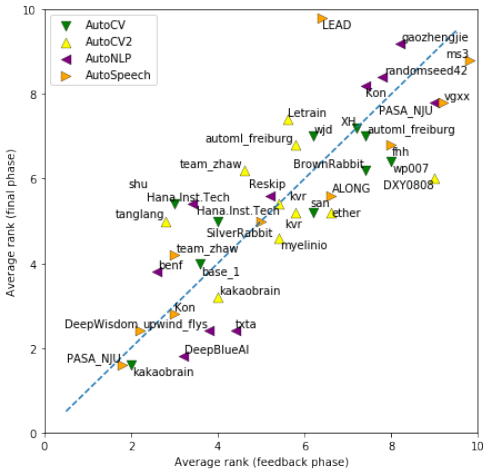


Figure 3: **Generalization ability** of AutoML methods. For each participant in a challenge, the average rank (over 5 datasets) in both phases is computed as x-axis and y-axis. When the scattered point is close to the diagonal, the feedback phase (with leaderboard feedback) result and final phase (with unseen datasets) result are consistent.

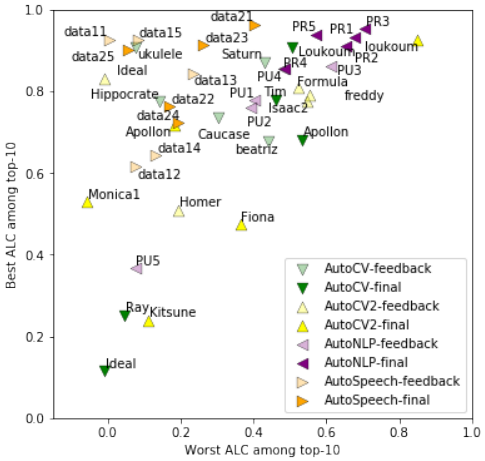


Figure 4: **Modeling difficulty** of each task/dataset. The x-axis (resp. y-axis) is the minimum (resp. maximum) ALC among top-10 participants in each challenge-phase. Tasks on top-left have larger modeling difficulty, while those close to the diagonal have small performance variance and model difficulty.

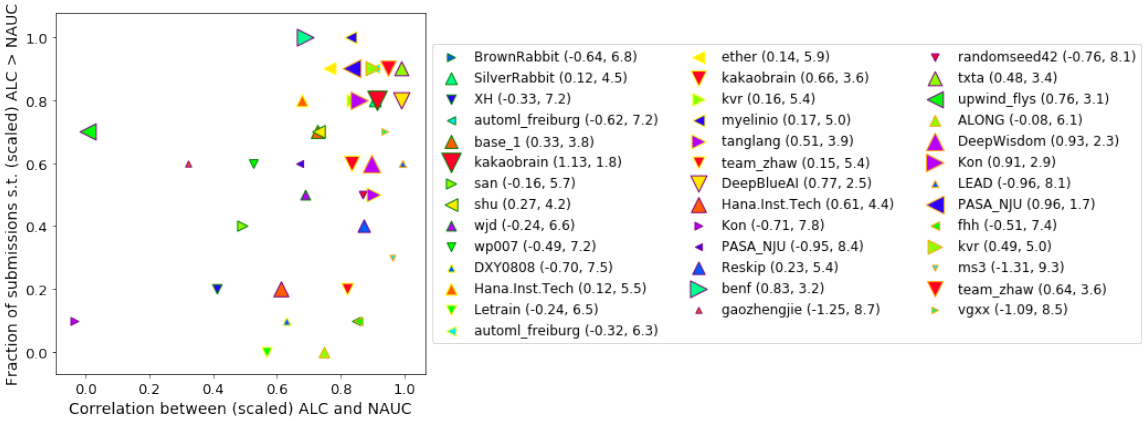


Figure 5: CORR vs FRAC (%(ALC > NAUC) *vs* correlation(ALC, NAUC)). ALC and NAUC were "scaled" (see text). The numbers in the legend are average scaled ALC and average rank of each participant. The marker size increases monotonically with average scaled ALC. 34 out of 40 participants have a CORR greater than 0.5 and 30 out of 40 participants have a FRAC above 0.5.

The participants that did well in the challenge (large symbols) are all in the upper right quadrant, with both FRAC larger than 50% and CORR larger than 0.7.

6. Discussion & Conclusion

The challenges AutoCV, AutoCV2, AutoNLP and AutoSpeech have allowed us to demonstrate that fully automated solution for one **specific domain** such as computer vision, natural language processing or speech recognition, is not as far away as many thought it was. The winning solutions have proven to be generalizable to unseen datasets, at least for each given domain. We have now publicly available software capable of handling any image, video, text and speech classification tasks without any human intervention whatsoever. However, whether these domain-specific approaches can be improved by cross-domain meta-learning methods is yet to be examined after the results of the final AutoDL challenge. For the any-time learning aspect, results show that methods having good any-time performance also have good final performance, but not always. This justifies our choice metric and suggests that it is harder than usual fixed-time performance metrics. The fact that there are learning curves crossing each other indicates that there is space for improvement in that respect. As for the repository of datasets that begins to take shape (already 100 datasets formatted), we benchmarked the datasets used in AutoDL challenges by participants' submissions and we see various modeling difficulty across different domains. The datasets we formatted together with our analysis will help the community push forward the study of meta-learning.

Acknowledgments

This work was sponsored with a grant from Google Research (Zürich) and additional funding from 4Paradigm, Amazon and Microsoft. It has been partially supported by the Spanish projects TIN2015-66951-C2-2-R, RTI2018-095232-B-C22, TIN2016-74946-P (MINECO/FEDER, UE) and CERCA Programme / Generalitat de Catalunya and ICREA under the ICREA Academia programme. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research. It received in kind support from the institutions of the co-authors. We are very indebted to Olivier Bousquet and André Elisseeff at Google for their help with the design of the challenge and the countless hours that André spent engineering the data format. The special version of the CodaLab platform we used was implemented by Tyler Thomas, with the help of Eric Carmichael, CK Collab, LLC, USA. Many people contributed time to help formatting datasets, prepare baseline results, and facilitate the logistics. The full list can be found at https://autodl.chalearn.org.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. December 2014.

Zhengying Liu, Zhen Xu, Sergio Escalera, Isabelle Guyon, Julio Jacques Junior, Meysam Madadi, Adrien Pavao, Sebastien Treguer, and Wei-Wei Tu. Towards automated computer vision: Analysis of the autocv challenges 2019. 2019. URL https://hal.archives-ouvertes.fr/hal-02386805/document.

Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. 1976.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, pages 2818–2826, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.308.

Appendix A. AutoDL datasets

We summarize in Table 3 some basic information of the datasets we formatted and used in AutoDL challenges.

Table 3: **Datasets used in AutoDL challenges.** "HWR" means handwriting recognition, "chnl" channel, and "var" variable size.

#	Dataset	Challenge(s)	Phase	Domain	Type	Class num.	Sample number train	Sample number test	Tensor dimension time	row	col	chnl
1	Munster	AutoCV1	public	HWR	image	10	60000	10000	1	28	28	1
2	Chucky	AutoCV1	public	objects	image	100	48061	11939	1	32	32	3
3	Pedro	AutoCV1	public	people	image	26	80095	19905	1	var	var	3
4	Decal	AutoCV1	public	aerial	image	11	634	166	1	var	var	3
5	Hammer	AutoCV1	public	medical	image	7	8050	1965	1	600	450	3
6	Ukulele	AutoCV1	feedback	HWR	image	3	6979	1719	1	var	var	3
7	Caucase	AutoCV1	feedback	objects	image	257	24518	6089	1	var	var	3
8	Beatriz	AutoCV1	feedback	people	image	15	4406	1094	1	350	350	3
9	Saturn	AutoCV1	feedback	aerial	image	3	324000	81000	1	28	28	4
10	Hippocrate	AutoCV1	feedback	medical	image	2	175917	44108	1	96	96	3
11	Loukoum	AutoCV1 AutoCV2	final final	HWR	image	3	27938	6939	1	var	var	3
12	Tim	AutoCV1	final	objects	image	200	80000	20000	1	32	32	3
13	Apollon	AutoCV1 AutoCV2 **AutoDL**	final final feedback	people	image	100	6077	1514	1	var	var	3
14	Ideal	AutoCV1 AutoCV2	final feedback	aerial	image	45	25231	6269	1	256	256	3
15	Ray	AutoCV1 **AutoDL**	final final	medical	image	7	4492	1114	1	976	976	3
16	Kraut	AutoCV2	public	action	video	4	1528	863	var	120	160	1
17	Katze	AutoCV2	public	action	video	6	1528	863	var	120	160	1
18	Kreatur	AutoCV2	public	action	video	4	1528	863	var	60	80	1
19	Freddy	AutoCV2	feedback	HWR	image	2	546055	136371	1	var	var	3
20	Homer	AutoCV2	feedback	action	video	12	1354	353	var	var	var	3
21	Isaac2	AutoCV2	feedback	action	video	249	38372	9561	var	102	78	1
22	Formula	AutoCV2	feedback	misc.	video	4	32994	8203	var	80	80	3
23	Fiona	AutoCV2 **AutoDL**	final final	action	video	6	8038	1962	var	var	var	3
24	Monica1	AutoCV2 **AutoDL**	final feedback	action	video	20	10380	2565	var	168	168	3
25	Kitsune	AutoCV2	final	action	video	25	18602	4963	var	46	82	3
26	data01	AutoSpeech	public	speech	time	100	3000	3000	var	1	1	1
27	data02	AutoSpeech	public	speech	time	7	428	107	var	1	1	1
28	data03	AutoSpeech	public	speech	time	3	796	200	var	1	1	1
29	data04	AutoSpeech	public	speech	time	20	939	474	var	1	1	1
30	data05	AutoSpeech	public	speech	time	10	199	597	var	1	1	1
31	data11	AutoSpeech	feedback	speech	time	55	1300	2000	var	1	1	1
32	data12	AutoSpeech	feedback	speech	time	5	3120	346	var	1	1	1
33	data13	AutoSpeech	feedback	speech	time	3	5000	1330	var	1	1	1
34	data14	AutoSpeech	feedback	speech	time	8	767	191	var	1	1	1
35	data15	AutoSpeech	feedback	speech	time	76	2286	571	var	1	1	1
36	data21	AutoSpeech	final	speech	time	50	800	1200	var	1	1	1
37	data22	AutoSpeech	final	speech	time	4	2649	294	var	1	1	1
38	data23 Oreal	AutoSpeech **AutoDL**	final final	speech	time	3	2000	264	var	1	1	1
39	data24	AutoSpeech	final	speech	time	16	384	386	var	1	1	1
40	data25 Sahak	AutoSpeech **AutoDL**	final feedback	speech	time	100	3008	752	var	1	1	1
41	O1	AutoNLP	public	english	text	2	7792	1821	var	1	1	1
42	O2	AutoNLP	public	english	text	20	11314	7532	var	1	1	1
43	O3	AutoNLP	public	english	text	2	60000	40000	var	1	1	1
44	O4	AutoNLP	public	chinese	text	10	55000	10000	var	1	1	1
45	O5	AutoNLP	public	chinese	text	18	156000	72000	var	1	1	1

46	PU1	AutoNLP	feedback	english	text	9	2822	499	var	1	1	1
47	PU2	AutoNLP	feedback	english	text	5	132651	23409	var	1	1	1
48	PU3	AutoNLP	feedback	chinese	text	2	1110203	195919	var	1	1	1
49	PU4	AutoNLP	feedback	chinese	text	11	100000	50000	var	1	1	1
50	PU5	AutoNLP	feedback	chinese	text	31	600000	400000	var	1	1	1
51	PR1	AutoNLP	final	english	text	20	33807	5967	var	1	1	1
52	PR2 Tanak	AutoNLP **AutoDL**	final feedback	english	text	2	42500	7501	var	1	1	1
53	PR3	AutoNLP	final	english	text	4	90000	30000	var	1	1	1
54	PR4	AutoNLP	final	chinese	text	11	100000	50000	var	1	1	1
55	PR5 Tal	AutoNLP **AutoDL**	final final	chinese	text	15	250000	132688	var	1	1	1
56	Adult	**AutoDL**	public	categorical	tabular	5	39074	9768	1	1	24	1
57	Dilbert	**AutoDL**	public	objects	tabular	5	14860	9720	1	1	2000	1
58	Digits	**AutoDL**	public	HWR	tabular	10	35000	35000	1	1	1568	1
59	Madeline	**AutoDL**	public	artificial	tabular	2	4220	3240	1	1	259	1
60	Barak	**AutoDL**	feedback	CE pair	tabular	4	21869	2430	1	1	270	1
61	Bilal	**AutoDL**	final	audio	tabular	20	10931	2733	1	1	400	1