7th ICML Workshop on Automated Machine Learning (2020)

How far are we from true AutoML: reflection from winning solutions and results of AutoDL challenge

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

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

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

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

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

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

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

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

Abstract

Following the completion of the AutoDL challenge (the final challenge in the ChaLearn AutoDL challenge series 2019), we investigate winning solutions and challenge results to answer an important motivational question: how far are we from achieving true AutoML? On one hand, the winning solutions achieve good (accurate *and* fast) classification performance on *unseen* datasets. On the other hand, all winning solutions still contain a considerable amount of hard-coded knowledge on the *domain* (or modality) such as image, video, text, speech and tabular. This form of ad-hoc meta-learning could be replaced by more automated forms of meta-learning in the future. Organizing a meta-learning challenge could help forging AutoML solutions that generalize to new unseen domains (e.g. new types of sensor data) as well as gaining insights on the AutoML problem from a more fundamental point of view. The datasets of the AutoDL challenge are a resource that can be used for further benchmarks and the code of the winners has been outsourced, which is a big step towards "democratizing" Deep Learning.

1. Introduction

Following the completion of the ChaLearn AutoDL challenges 2019 (Liu et al., 2020), we are interested in how an important motivational question has been addressed: how far are we from achieving true AutoML (Hutter et al., 2018)? Here the AutoML problem asks whether one could have one single algorithm (an *AutoML algorithm*) that can perform learning on a large spectrum of data and always has consistently good performance, removing the need for human expertise (which is exactly the opposite of No Free Lunch theorems (Wolpert and Macready, 1997; Wolpert, 1996, 2001)). And by "good" performance, we actually mean "accurate" and "fast", which corresponds to the **any-time learning** setting emphasized by the AutoDL challenge.

On the negative side, disappointingly, there was no novel theoretical insight that transpired from the contributions made in this challenge. Also, despite our effort to format all datasets uniformly to encourage generic solutions, the participants adopted specific workflows for each domain/modality. And, although some solutions improved over Baseline 3 (the strongest baseline we provide to participants), it strongly influenced many. Deep Learning solutions dominated, but Neural Architecture Search was impractical within the time budget imposed. Most solutions relied on fixed-architecture pre-trained networks, with fine-tuning.

However, on the positive side, several interesting and important results were obtained, including that the top two winners passed all final tests without failure, a significant step towards true automation since **their code was blind-tested for training and testing on datasets never seen before**, albeit from the same domains. Their solutions were open-sourced, see `http://autodl.chalearn.org`. Also, **any-time learning was addressed successfully, without sacrificing final performance**. In the rest of the paper, review these results in more details and suggest future directions, including the organization of a meta-learning challenge, which would push AutoML one step further, toward generalizing to new domains.

2. Challenge design

In AutoDL challenge, **raw data** (images, videos, audio, text, tabular, etc) are provided to participants formatted in a uniform tensor manner (namely TFRecords, a standard generic data format used by TensorFlow). We formatted around 100 datasets in total and used 66 of them for all AutoDL challenges: 17 image, 10 video, 16 text, 16 speech and 7 tabular. 15 datasets are used in AutoDL challenge (Table 1). Information on some meta-features of all AutoDL datasets can be found on the "Benchmark" page[1] of our website.

An important feature of the AutoDL challenge is that the code of the participants is **blind tested**, without any human intervention, in uniform conditions imposing restrictions on training and test time and memory resources, to push the state-of-the-art in automated machine learning. The challenge had 2 phases: a **feedback phase** during which methods were trained and tested on the platform on 5 practice datasets. During the feedback phase, the participants could make several submissions per day and get immediate feedback on a leaderboard. In the **final phase**, 10 fresh datasets are used. Only one final code submission was allowed in that phase. We ran the challenge on the CodaLab platform (`http://competitions.codalab.org`), with support from Google Cloud virtual machines equipped with NVIDIA Tesla P100 GPUs.

The AutoDL challenge encourages **any-time learning** by scoring participants with the Area under the Learning Curve (ALC) (see definition in (Liu et al., 2019a), and examples of learning curves can in Figure 1). 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

[1] `https://autodl.chalearn.org/benchmark`

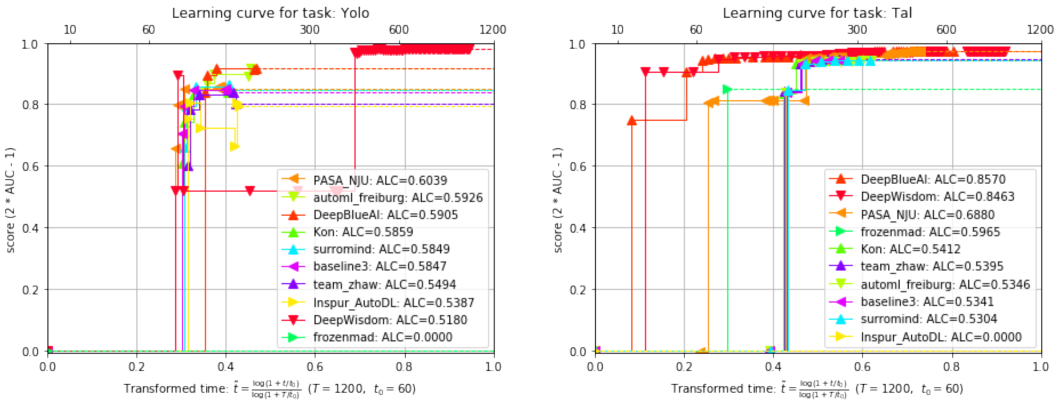


Figure 1: **Learning curves of top-9 teams** (together with one baseline) on the datasets *Yolo*(video) and *Tal*(text) from the AutoDL challenge final phase. We observe different patterns of learning curves, revealing various strategies adopted by participating teams. For *Tal*, the curve of *DeepBlueAI* goes up quickly at the beginning but stabilizes at an inferior final performance (and also inferior any-time performance) than *DeepWisdom*. The fact that these two curves cross each other suggests that one might be able to combine these 2 methods to improve the exploration-exploitation trade-off. Finally, although different patterns are found, some teams such as *team_zhaw*, *surromind* and *automl_freiburg* show very similar patterns on *Tal*. This is because all teams adopted a domain-dependent approach and some teams simply used the code of Baseline 3 for certain domains (text in this case).

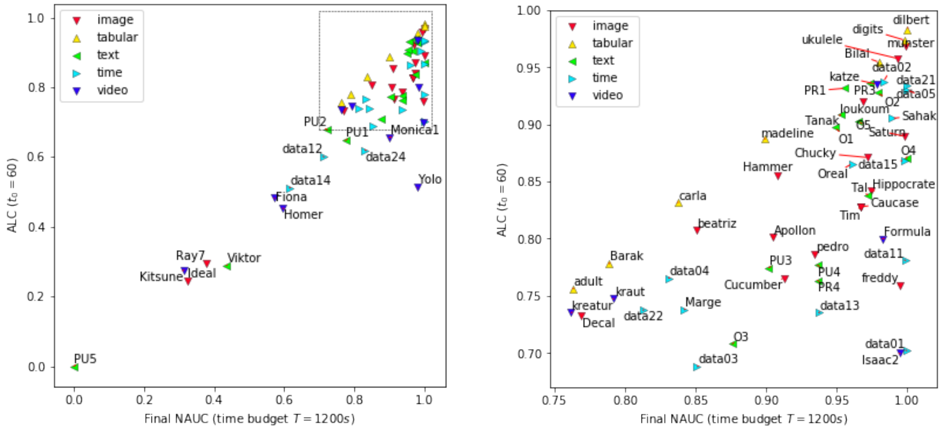
(AUC) $NAUC = 2 \times AUC - 1$ averaged over all classes. Since several predictions can be made during the learning process, this allows us to plot learning curves, i. e. "performance" (on test set) as a function of time. Then for each dataset, we compute the Area under Learning Curve (ALC). The time axis is log scaled (with time transformation defined in (Liu et al., 2019a)) to put more emphasis on the beginning of the curve. 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.

As in previous challenges (e.g. AutoCV, AutoCV2, AutoNLP and AutoSpeech), we provide 3 **baselines** (Baseline 0, 1 and 2) for different levels of use: Baseline 0 is just constant predictions for debug purposes, Baseline 1 a linear model, and Baseline 2 a CNN (see (Liu et al., 2019b) for details). In the AutoDL challenge, we provide additionally a Baseline 3 Liu et al. (2020) which combines the winning solutions of previous challenges[2].

3. AutoDL challenge results

The AutoDL challenge (the last challenge in the AutoDL challenges series 2019) lasted from 14 Dec 2019 (launched during NeurIPS 2019) to 3 Apr 2020. It has had a participation of 54 teams with 247 submissions in total and 2614 dataset-wise submissions. Among

[2] The code of Baseline 3 can be found at `https://autodl.chalearn.org/benchmark`.



(a) All results included

(b) Rectangular area in 2a zoomed

Figure 2: **ALC and final NAUC performances of *DeepWisdom*** on all 66 AutoDL datasets. Different domains are shown with different markers. In 2a, the dataset name is shown beside each point except the top-right area, which is shown in Figure 2b. Note that among all 66 AutoDL datasets, *DeepWisdom* only fails on *PU5* (due to a time limit exceeded error), showing the robustness of the winning method.

these teams, 19 of them managed to get a better performance (i.e. average rank over the 5 feedback phase datasets) than that of Baseline 3 in feedback phase and entered the final phase of blind test. According to our challenge rules, only teams that provided a description of their approach (by filling out some fact sheets we sent out) were eligible for getting a ranking in the final phase. We received 8 copies of these fact sheets and thus only these 8 teams were ranked. These teams are (alphabetical order): *DeepBlueAI*, *DeepWisdom*, *frozenmad*, *Inspur_AutoDL*, *Kon*, *PASA_NJU*, *surromind*, *team_zhaw*. One team (*automl_freiburg*) made a late submission and isn't eligible for prizes but will be included in the post-analysis for scientific purpose.

The final ranking is computed from the performances on the 10 unseen datasets in the final phase. To reduce the variance from diverse factors such as randomness in the submission code and randomness of the execution environment (which makes the exact ALC scores very hard to reproduce since the wall-time is hard to control exactly), we re-run every submission several times and average the ALC scores. The average ALC scores obtained by each team is shown in Figure 3 (the teams are ordered by their final ranking). The large error bars account for code failures.

4. Winning approaches

A summary of the winning approaches on each domain can be found in Table 2. Another summary using a categorization by machine learning techniques can be found in Table 3. We see in Table 2 that almost all approaches used 5 different methods from 5 domains. And for each domain, the winning teams' approaches are much inspired by Baseline 3. This means that we haven't achieved true AutoML since for each new domain we still need to

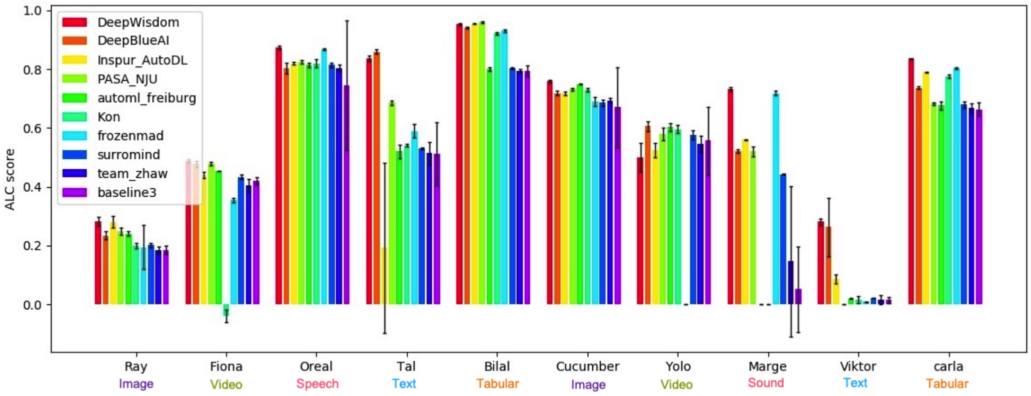


Figure 3: **ALC scores of top 9 teams in AutoDL final phase** averaged over repeated evaluations (and Baseline 3, for comparison). The entry of top 6 teams are re-run 9 times and 3 times for other teams. Error bars are shown with (half) length corresponding to the standard deviation from these runs. Some rare entries are excluded for computing these statistics due to failures caused by the challenge platform backend. The team ordering follows that of their average rank in the final phase. More information on the task can be found in Table 1.

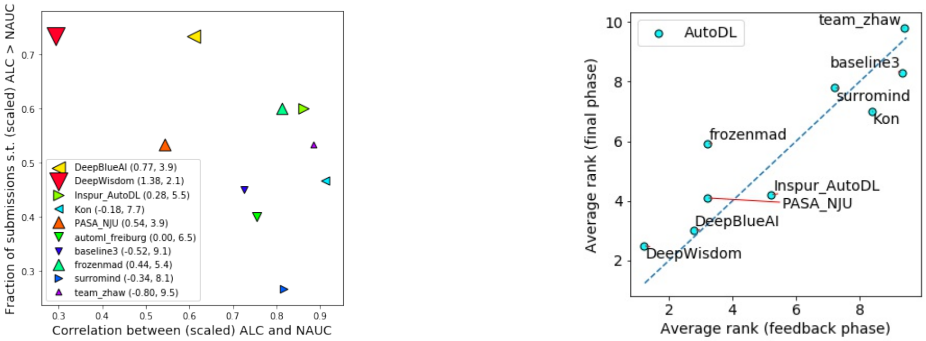
hard-code a new approach. In Table 3, we see that almost all different machine learning techniques are actively present and frequently used in all domains (exception some rare cases for example transfer learning on tabular data).

4.1 AutoML generalization ability of winning approaches

One crucial question for all AutoML methods is whether the method can have good performances on unseen datasets. We propose to compare the average rank of all top-8 methods in both feedback phase and final phase, then compute the Pearson correlation (Pearson's ρ) of the 2 rank vectors (thus similar to Spearman's rank correlation (Wikipedia, 2020)). The average ranks of top methods are shown in Figure 4b, with a Pearson correlation $\rho_{X,Y} = 0.91$ and p-value $p = 5.8 \times 10^{-4}$. This means that the correlation is statistically significant and no leaderboard overfitting is observed. Thus the winning solutions can indeed generalize to *unseen* datasets, showing AutoML generalization ability. To show this even further, we ran *DeepWisdom*'s solution on all 66 AutoDL datasets and the results are shown in Figure 2. We see that the winning approach *DeepWisdom* only fails on 1 out of the 66 tasks, showing the AutoML generalization ability of the winning approach.

4.2 Dealing with any-time learning

Figure 4a 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 *vs.* $correlation(ALC, NAUC)$. From the figure, we see that any-time performance (ALC) and final performance (NAUC) are often quite correlated, but only those who favor ALC can win the challenge. This



(a) $\%(ALC > NAUC)$ *vs* $\text{corr}(ALC, NAUC)$. ALC and NAUC were “scaled” within each task. The numbers in the legend are average scaled ALC and average rank of each participant. The marker size increases monotonically with average scaled ALC. We see that the top-2 teams *DeepWisdom* and *DeepBlueAI* indeed have higher fraction of $(ALC > NAUC)$, meaning that they put much effort to improve any-time learning performance (ALC score). However, *DeepWisdom* shows lower correlation between ALC and NAUC, which means that their final performance (NAUC) may not always be the best.

(b) **Task over-modeling:** We compare performance in the feedback and final phase, in an effort to detect possible habituation to the feedback datasets due to multiple submissions. The average rank of the top-8 teams is shown. The figure suggests no strong over-modeling (over-fitting at the meta-learning level): A team having a significantly better rank in the feedback phase than in the final phase would be over-modeling (far above the diagonal). The Pearson correlation is $\rho_{X,Y} = 0.91$ and p-value $p = 5.8 \times 10^{-4}$.

suggests that the any-time learning problem could be strictly harder than the usual final performance problem.

5. Conclusion and further work

We reviewed the design and results of the final challenge in AutoDL series 2019: the AutoDL challenge. Deep learning is still dominant and more importantly, fixed domain-dependent pre-trained neural architectures are heavily used. Diverse human knowledge (especially that of deep learning) is hard-coded in these architectures and deployed to different domains such as image, video, text, speech and tabular. Neural architecture search (NAS) (see e.g. Hutter et al., 2018, for a review) hasn't been employed due to its huge computational cost which doesn't fit well in our any-time learning setting, with a relatively small maximum time budget. Nevertheless, the AutoDL challenge helped pushing the state of the art in AutoDL. Among other things, the challenge revealed that Automated Deep Learning methods are ripe for all these domains and show good performance on *unseen* datasets, which is one of the most important goals of AutoML. Also, meta-learning seems one of the most promising avenues to future explore. While our AutoDL challenge series continues (with currently the AutoGraph challenge, see `http://autodl.chalearn.org`), we are currently investigating several possible meta-learning challenge protocols for a future cross-modal NAS challenge. This could encourage researchers to automate meta-learning, leading perhaps to a universal workflow, universal coding, cross-modal feature representations, universal neural architectures or meta-architectures, and/or universal hyper-parameter search trainable agents.

References

James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc., 2011. URL `http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf`.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Twenty-first international conference on Machine learning - ICML '04*, page 18, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015432. URL `http://portal.acm.org/citation.cfm?doid=1015330.1015432`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. page 10.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/finn17a.html`.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges.* Springer, 2018. In press, available at http://automl.org/book.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf`.

Marius Lindauer, Holger H. Hoos, Frank Hutter, and Torsten Schaub. AutoFolio: an automatically configured algorithm selector. *Journal of Artificial Intelligence Research*, 53(1):745–778, May 2015. ISSN 1076-9757.

Zhengying Liu, Isabelle Guyon, Julio Jacques Junior, Meysam Madadi, Sergio Escalera, Adrien Pavao, Hugo Jair Escalante, Wei-Wei Tu, Zhen Xu, and Sebastien Treguer. AutoCV Challenge Design and Baseline Results. In *CAp 2019 - Conférence sur l'Apprentissage Automatique*, Toulouse, France, July 2019a. URL https://hal.archives-ouvertes.fr/hal-02265053.

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, November 2019b. URL https://hal.archives-ouvertes.fr/hal-02386805.

Zhengying Liu, Adrien Pavao, Zhen Xu, Sergio Escalera, Fabio Ferreira, Isabelle Guyon, Sirui Hong, Frank Hutter, Rongrong Ji, Thomas Nierhoff, Kangning Niu, Chunguang Pan, Danny Stoll, Sebastien Treguer, Jin Wang, Peng Wang, Chenglin Wu, and Youcheng Xiong. Winning solutions and post-challenge analyses of the ChaLearn AutoDL challenge 2019. *under review for IEEE TPAMI*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*, January 2015. URL http://arxiv.org/abs/1409.0575. arXiv: 1409.0575.

Wikipedia. Spearman's rank correlation coefficient, April 2020. URL https://en.wikipedia.org/w/index.php?title=Spearman%27s_rank_correlation_coefficient&oldid=953109044. Page Version ID: 953109044.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997. ISSN 1089-778X. doi: 10.1109/4235.585893. URL https://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf.

David Wolpert. The Supervised Learning No-Free-Lunch Theorems. In *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*, January 2001. doi: 10.1007/978-1-4471-0123-9_3.

David H. Wolpert. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390, October 1996. ISSN 0899-7667. doi: 10.1162/neco.1996.8.7.1341. URL https://doi.org/10.1162/neco.1996.8.7.1341.

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 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 is found on our website `https://autodl.chalearn.org/`.

Appendix A. Datasets used in AutoDL challenge

Table 1: **Datasets of the AutoDL challenge,** for both phases. The final phase datasets (**meta-test** datasets) vary a lot in terms of number of classes, number of training examples, and tensor dimension, compared to those in the feedback phase. This was one of the difficulties of the AutoDL challenge. “chnl” codes for channel, “var” for variable size, “CE pair” for “cause-effect pair”. More information on all 66 datasets used in AutoDL challenges can be found at `https://autodl.chalearn.org/benchmark`.

					Class	Sample number		Tensor dimension			
#	Dataset	Phase	Topic	Domain	num.	train	test	time	row	col	chnl
1	Apollon	feedback	people	image	100	6077	1514	1	var	var	3
2	Monica1	feedback	action	video	20	10380	2565	var	168	168	3
3	Sahak	feedback	speech	time	100	3008	752	var	1	1	1
4	Tanak	feedback	english	text	2	42500	7501	var	1	1	1
5	Barak	feedback	CE pair	tabular	4	21869	2430	1	1	270	1
6	Ray	final	medical	image	7	4492	1114	1	976	976	3
7	Fiona	final	action	video	6	8038	1962	var	var	var	3
8	Oreal	final	speech	time	3	2000	264	var	1	1	1
9	Tal	final	chinese	text	15	250000	132688	var	1	1	1
10	Bilal	final	audio	tabular	20	10931	2733	1	1	400	1
11	Cucumber	final	people	image	100	18366	4635	1	var	var	3
12	Yolo	final	action	video	1600	836	764	var	var	var	3
13	Marge	final	music	time	88	9301	4859	var	1	1	1
14	Viktor	final	english	text	4	2605324	289803	var	1	1	1
15	Carla	final	neural	tabular	20	10931	2733	1	1	535	1

Appendix B. Implementation details of winning solutions

Table 2: **Summary of the five top ranking solutions** and their average rank in the final phase. The participant's average rank (over all tasks) in the final phase is shown in parenthesis (automl_freibug and Baseline 3 were not ranked in the challenge). Each entry concerns the algorithm used for each domain and is of the form "[pre-processing / data augmentation]-[transfer learning/meta-learning]-[model/architecture]-[optimizer]" (when applicable).

Team	image	video	speech	text	tabular
1.DeepWisdom (1.8)	[ResNet-18 and ResNet-9 models] [pre-trained on ImageNet]	[MC3 model] [pre-trained on Kinetics]	[fewshot learning] [LR, Thin ResNet34 models] [pre-trained on VoxCeleb2]	[fewshot learning] [task difficulty and similarity evaluation for model selection] [SVM, TextCNN,[fewshot learning] RCNN, GRU, GRU with Attention]	[LightGBM, Xgboost, Catboost, DNN models] [no pre-trained]
2.DeepBlueAI (3.5)	[data augmentation with Fast AutoAugment] [ResNet-18 model]	[subsampling keeping 1/6 frames] [Fusion of 2 best models]	[iterative data loader (7, 28, 66, 90%)] [MFCC and Mel Spectrogram preprocessing] [LR, CNN, CNN+GRU models]	[Samples truncation and meaningless words filtering] [Fasttext, TextCNN, BiGRU models] [Ensemble with restrictive linear model]	[3 lightGBM models] [Ensemble with Bagging]
3.Inspur_AutoDL (4)	Tuned version of Baseline 3				[Incremental data loading and training][HyperOpt][LightGBM]
4.PASA_NJU (4.1)	[shape standardization and image flip (data augmentation)][ResNet-18 and SeResnext50]	[shape standardization and image flip (data augmentation)][ResNet-18 and SeResnext50]	[data truncation(2.5s to 22.5s)][LSTM, VggVox ResNet with pre-trained weights of DeepWisdom(AutoSpeech2019) Thin-ResNet34]	[data truncation(300 to 1600 words)][TF-IDF and word embedding]	[iterative data loading] [Non Neural Nets models] [models complexity increasing over time] [Bayesian Optimization of hyperparameters]
5.frozenmad (5)	[images resized under 128x128] [progressive data loading increasing over time and epochs] [ResNet-18 model] [pre-trained on ImageNet]	[Successive frames difference as input of the model] [pre-trained ResNet-18 with RNN models]	[progressive data loading in 3 steps 0.01, 0.4, 0.7] [time length adjustment with repeating and clipping] [STFT and Mel Spectrogram preprocessing] [LR, LightGBM, VggVox models]	[TF-IDF and BERT tokenizers] [SVM, RandomForest , CNN, tinyBERT]	[progressive data loading] [no preprocessing] [Vanilla Decision Tree, RandomForest, Gradient Boosting models applied sequentially over time]
automl_freiburg	Architecture and hyperparameters learned offline on meta-training tasks with BOHB. Transfer-learning on unseen meta-test tasks with AutoFolio. Models: EfficientNet [pre-trained on ImageNet with AdvProp], ResNet-18 [KakaoBrain weights], SVM, Random Forest, Logistic Regression		Baseline 3		
Baseline 3	[Data augmentation with Fast AutoAugment, adaptive input size][Pretrained on ImageNet][ResNet-18(selected offline)]	[Data augmentation with Fast AutoAugment, adaptive input size, sample first few frames, apply stem CNN to reduce to 3 channels][Pretrained on ImageNet][ResNet-18(selected offline)]	[MFCC/STFT feature][LR, LightGBM, Thin-ResNet-34, VggVox, LSTM]	[resampling training examples][LinearSVC, LSTM, BERT]	[interpolate missing value][MLP of four hidden layers]

Table 3: **Machine learning techniques** applied to each of the 5 domains considered in AutoDL challenge.

ML technique	image	video	speech	text	tabular
Meta-learning	Offline meta-training transferred with AutoFolio (Lindauer et al., 2015) based on meta-features (*automl_freiburg*, for image and video) Offline meta-training generating solution agents, searching for optimal sub-operators in predefined sub-spaces, based on dataset meta-data. (*DeepWisdom*) MAML-like method (Finn et al., 2017) (*team_zhaw*)				
Preprocessing	image cropping and data augmentation (*PASANJU*), fast autoaugment (*DeepBlueAI*)	Sub-sampling keeping 1/6 frames and adaptive image size (*DeepBlueAI*) Adaptive image size	MFCC, Mel Spectrogram, STFT	root features extractions with stemmer, meaningless words filtering (*DeepBlueAI*)	Numerical and Categorical data detection and encoding
Hyperparameter Optimization	Offline with BOHB (Falkner et al.) (Bayesian Optimization and Multi-armed Bandit) (*automl_freiburg*) Sequential Model-Based Optimization for General Algorithm Configuration (SMAC) (Hutter et al., 2011) (*automl_freiburg*)		Online model complexity adaptation (*PASA_NJU*)	Online model selection and early stopping using validation set (*Baseline 3*(_flys))	Bayesian Optimization (*PASANJU*) HyperOpt (Bergstra et al., 2011) (*Inspur_AutoDL*)
Transfer learning	Pre-trained on ImageNet (Russakovsky et al., 2015) (all teams except *Kon*)	Pre-trained on ImageNet (Russakovsky et al., 2015) (all top-8 teams except *Kon*) MC3 model pre-trained on Kinetics (*DeepWisdom*)	ThinResnet34 pre-trained on VoxCeleb2 (*DeepWisdom*)	BERT-like (Devlin et al., 2019) models pre-trained on FastText	(not applicable)
Ensemble learning	Adaptive Ensemble Learning (ensemble latest 2 to 5 predictions) (*DeepBlueAI*)	Ensemble Selection (Caruana et al., 2004) (top 5 validation predictions are fused) (*DeepBlueAI*); Ensemble models sampling 3, 10, 12 frames (*DeepBlueA*)	last best predictions ensemble strategy (*DeepWisdom*) averaging 5 best overall and best of each model: LR, CNN, CNN+GRU (*DeepBlueA*)	Weighted Ensemble over 20 best models (Caruana et al., 2004) (*DeepWisdom*)	LightGBM ensemble with bagging method (Ke et al., 2017) (*DeepBlueAI*), Stacking and blending (*DeepWisdom*)