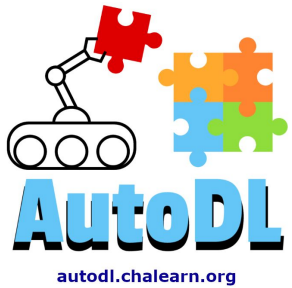
Automated Deep Learning Self-Service

Michael VACCARO (michavaccaro@gmail.com)
23/01/2021



université PARIS-SACLAY

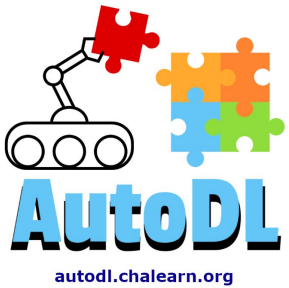
Plan

I. About the AutoDL challenge series
II. The AutoDL Self-Service
III. How does Deep Wisdom (winner) work ?
IV. Benchmarking the AutoDL solutions

I. About the AutoDL challenge series

AutoDL, a series of challenges

- Objective: provide “universal learning machines”, which can learn and predict without any human intervention.
- Classification tasks: from image, video, speech, text or tabular domains, formatted in an uniform way.
- Series of challenges in 2019-2020 : AutoCV, AutoCV2, AutoNLP, AutoSpeech, AutoWSL and the final **AutoDL [1]** (combining all types of data).



CodaLab

CHALEARN

Google

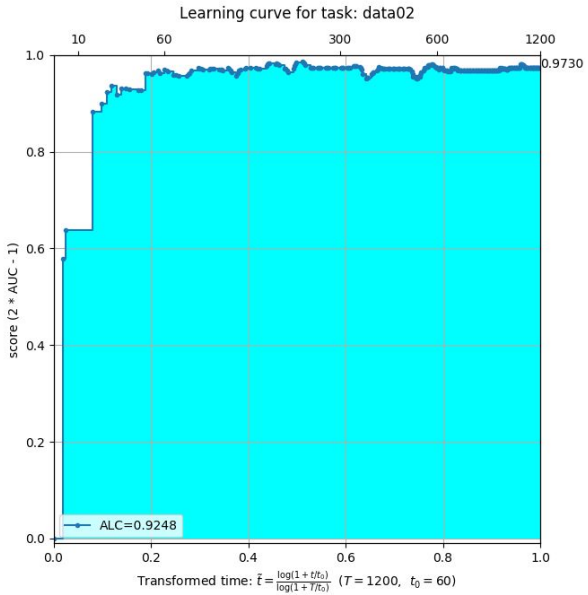
4Paradigm

AutoDL challenge design

- Five datasets from all domains in the final phase
- Time limit: 20min to initialize the model and 20min to run.
- Limited resources
- Baselines provided: Baseline0 (constant model), Baseline1 (Linear model), Baseline2 (CNN-based solution) and **Baseline3** (combining winning solutions from previous domain-specific challenges).

AutoDL challenge design

- Metric : Area under the Learning Curve (ALC).



Each prediction is scored with the Normalized ROC AUC metric (NAUC).

NAUC = 2 AUC - 1

II. The AutoDL Self-Service

A Codalab* platform

The link: https://competitions.codalab.org/competitions/27082

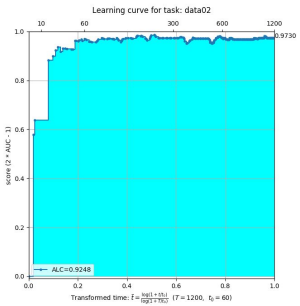
Through this “Inverted” competition:

- Upload YOUR dataset
- Find automatically and train a deep learning model on your task
- Rate it on your test set
- Predict on unseen data afterwards

*competitions.codalab.org/

What is behind?

- **Model used:** *DeepWisdom**, winner of AutoDL** challenge
- Submissions run for 20 minutes on GPUs
- Metric used to “rate” submissions: ALC (based on predictions curve, scored by ROC NAUC)



Each prediction is scored with the Normalized ROC AUC metric (NAUC).

NAUC = 2 AUC - 1

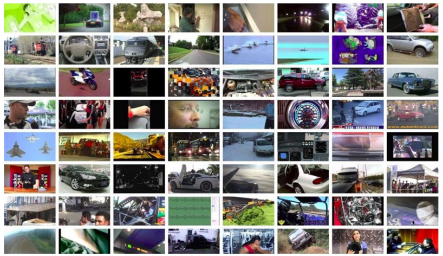
*github.com/DeepWisdom/AutoDL **autodl.chalearn.org

Datasets

- Datasets can be **any classification task** formatted as 4D tensors (time, row, col, channels) in a generic TFRecords format.

6	60	48	1	59	73289	2	1	0	14
5	50	30	1	67	212490	1	1	0	1
5	58	28	21	23	289293	1	1	0	11
2	45	51	1	59	154950	1	1	0	6
12	40	18	1	43	93449	1	3	0	4
6	37	59	1	33	182074	1	1	0	1
6	50	40	1	26	164299	5	1	0	1
9	40	31	1	17	386120	1	2	0	4
9	40	21	1	75	211013	1	1	0	2
5	25	23	1	32	178649	1	1	0	1
8	40	36	1	35	154641	1	1	0	4
1	40	29	1	38	159449	1	NaN	0	4
13	50	46	1	50	192485	5	1	1887	6

Tabular → (1, 1, 10, 1)



Video → (-1, 200, 300, 3)

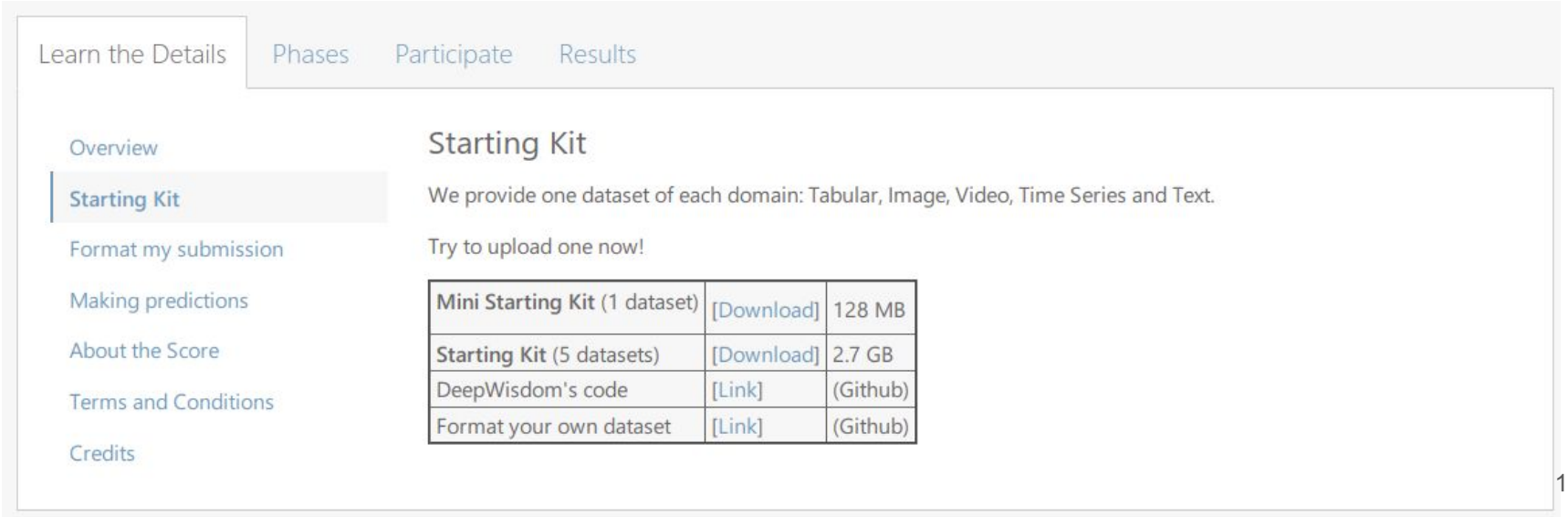
Variable

Image → (1, 28, 28, 1)

- [Github: provided to convert your raw data into the right format.]

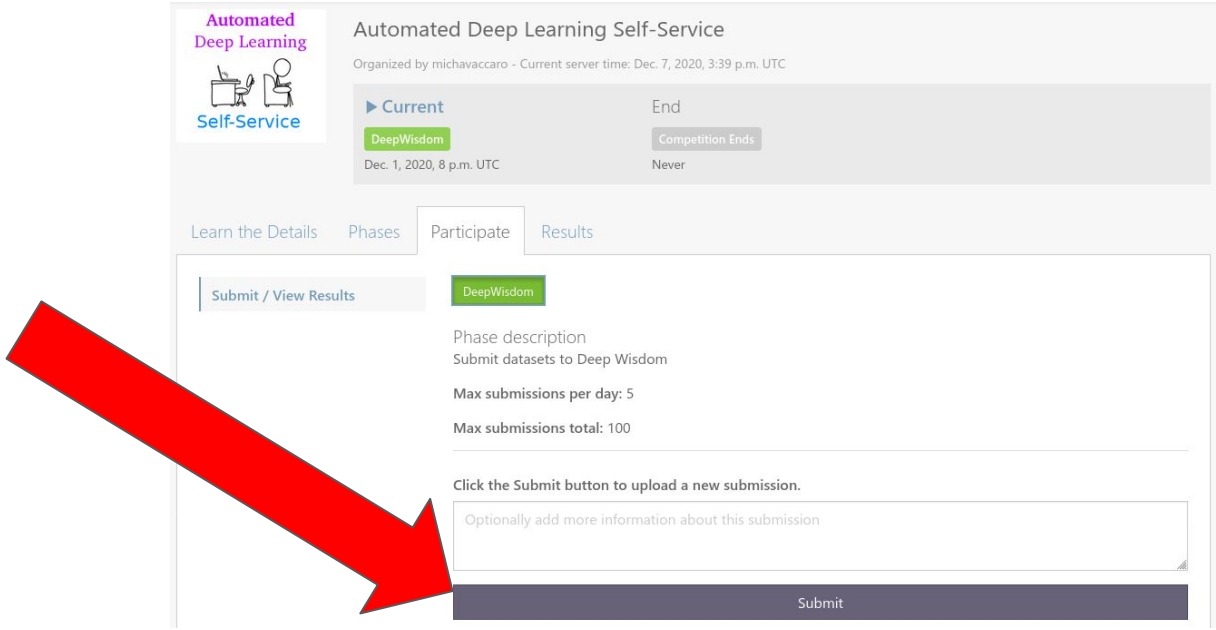
How to try?

- Log-in to Codalab and register to the “competition”
- Read the overview
- Try the interface with the starting kit:

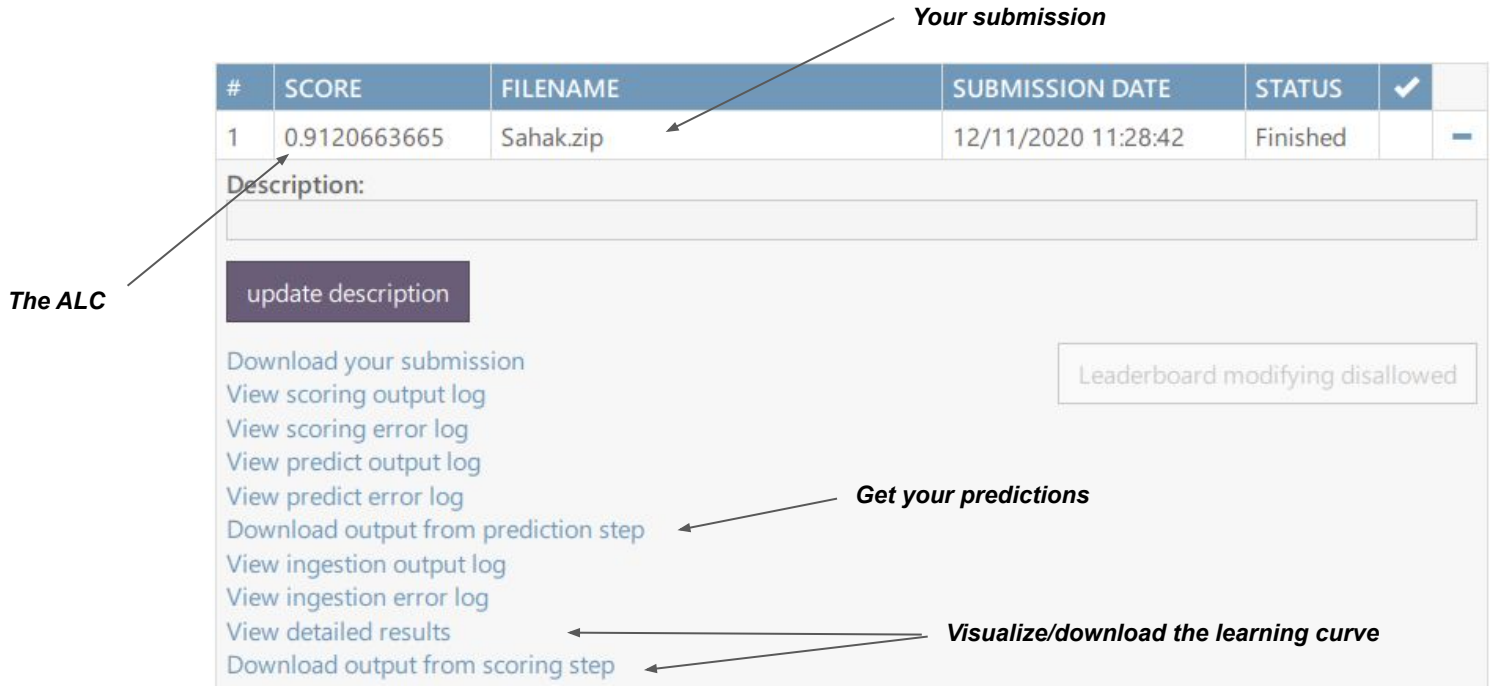


How to try?

- Format your own data in the AutoDL format
- Submit!

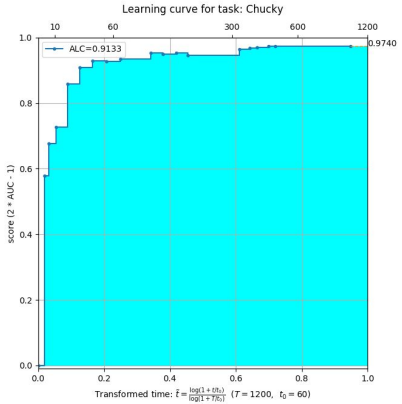


Get the results



Summary

Training and testing



Predictions on unseen data

0.00120394 0.000341728 0.000270689 0.000168973 0.00030238 0.00028764 0.000226916 0.993954 0.00031066? 0.0029327
0.000828241 0.00198071 0.0110102 0.941986 0.0031404 0.0191072 0.0134318 0.00202414 0.00511608 0.00139547
0.000760687 0.990328 0.000716812 0.000431503 0.002623 0.000891106 0.00200823 0.000575951 0.000814777 0.000850009
0.979625 0.00136646 0.000978387 0.00121633 0.00119111 0.00240173 0.00616321 0.000804491 0.0022879 0.00396534
0.00105605 0.00196499 0.000855562 0.000478205 0.988694 0.000239895 0.00306998 0.000760649 0.000367557 0.00251343
0.000904679 0.984798 0.000960247 0.000620535 0.00615297 0.000983322 0.00187504 0.00103625 0.00105802 0.00161122

URL: https://competitions.codalab.org/competitions/27082

Next

- Advantage : self-service (no ML knowledge needed, no computational resources needed)

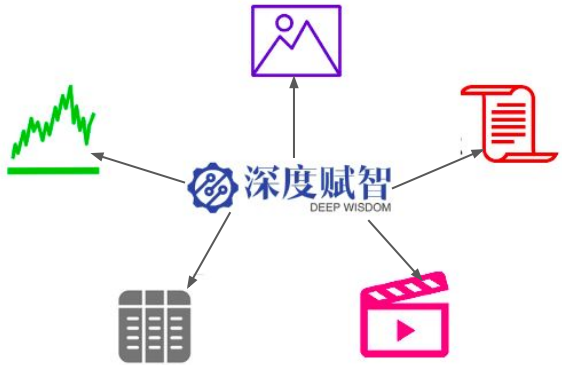
Next:

- Getting users
- Adding other models
- Benchmarking?

III. How does Deep Wisdom work ?

Key features

A multi-modal solution adapted to every classification tasks (single-label or multilabel)



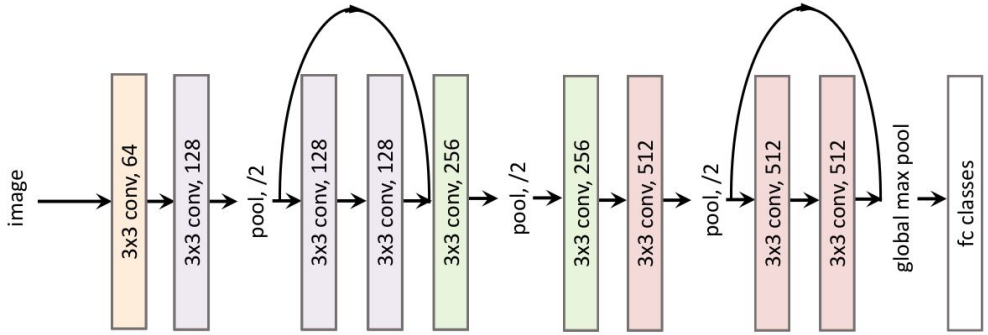
Meta-learning framework to learn domain-specific workflows

Use offline datasets (public datasets…)

Image domain

Transfer learning:

- In the early learning phases (t<0.2): Resnet-18 (architecture already used in **Baseline3**)
- Resnet-9 then.

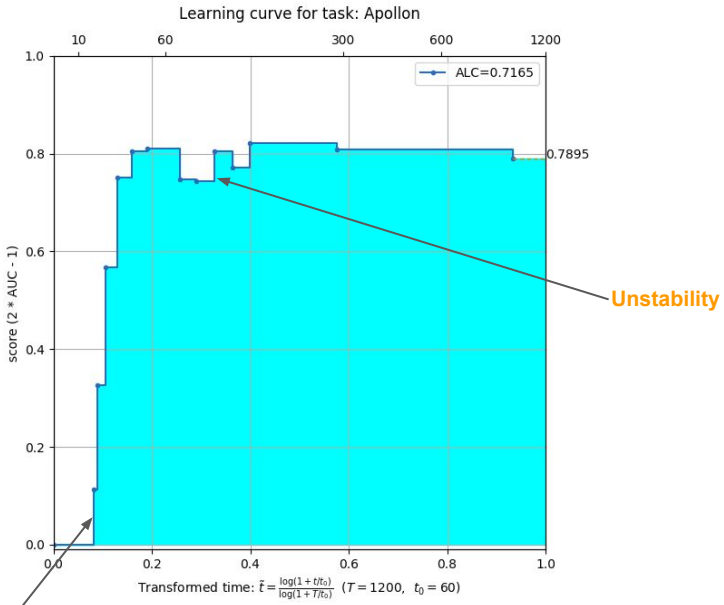


Resnet-9 architecture

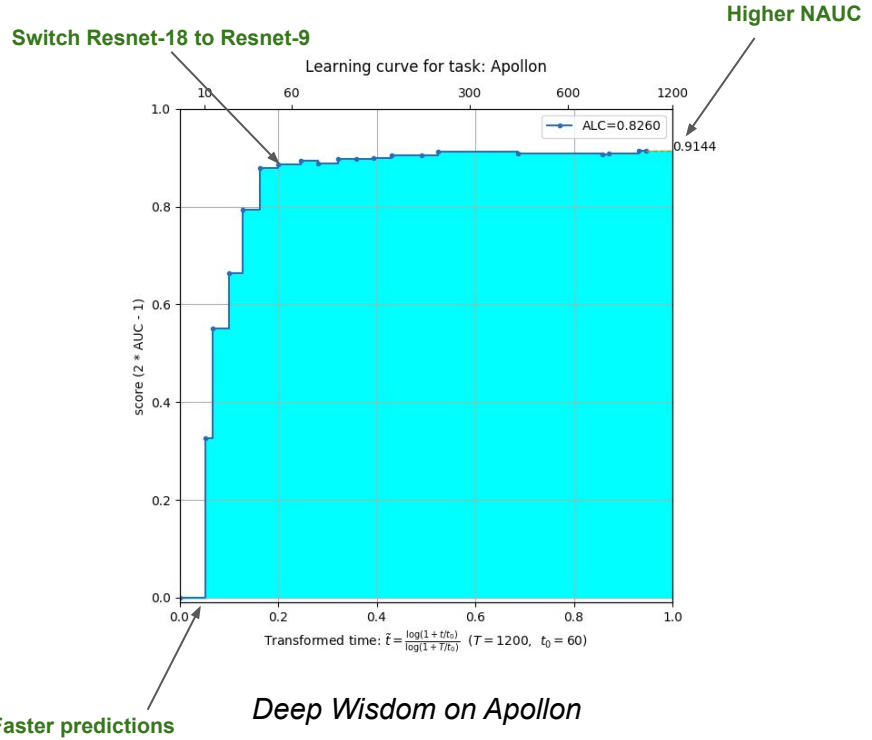
The resnets are pre-trained on **ImageNet [2] dataset**. Batch Normalization and Bias parameters are reinitialized.

Fast autoaugmentation [3] is applied on later training phases to improve last AUCs.

Image domain



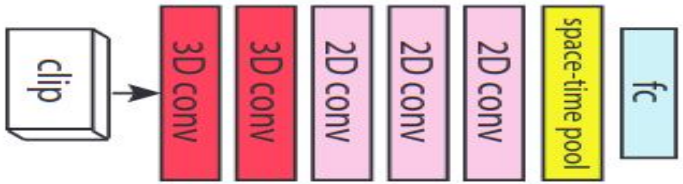
Baseline3 (all winners) on Apollon



Deep Wisdom on Apollon

Video domain

Transfer learning:
MC3 [4] (mixed convolutions) instead of Resnet-18 (used in **Baseline3**) pretrained on Kinetics.

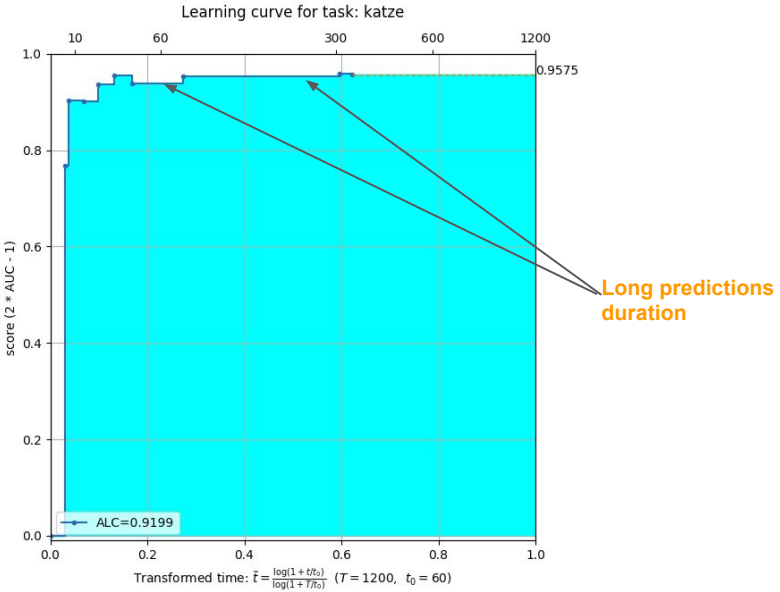


First two layers are freezed. Bias and linear weights are reinitialized.

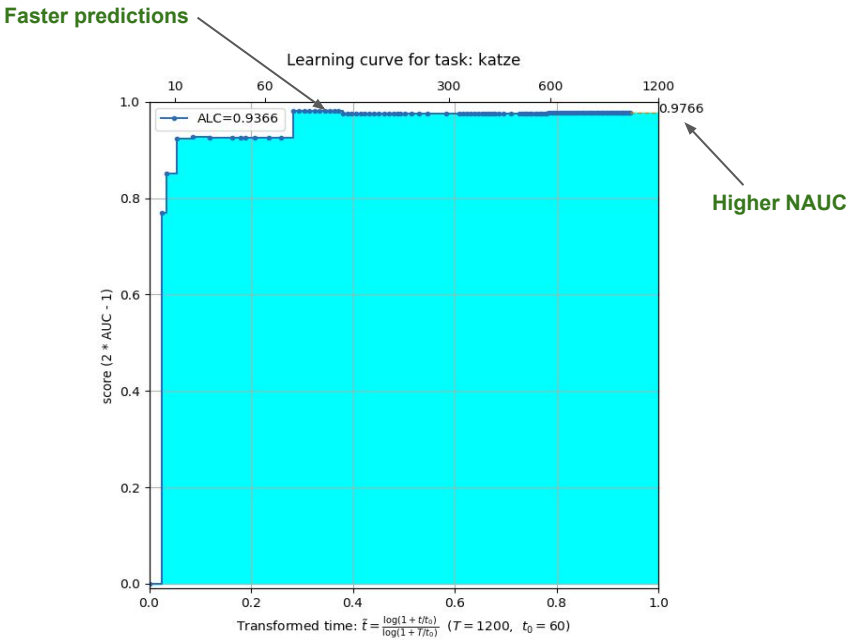
Frame extraction: extracting few frames to accelerate predictions (*3D Conv being slower*).

Weighting ensemble strategy: predict on 3-,10-,12-frames extracted data with MC3 and combine them.

Video domain



Baseline3 on Katze



Deep Wisdom on Katze

Speech domain

Resulting of Meta-training on offline datasets.
Model selection: Logistic Regression and ThinResNet34 [5] pretrained on VoxCeleb2 [6].

First layers are freezed.

Workflow optimization:

- Multilabel/unilabel specificity
- Softmax → Sigmoid for last activation, different losses
- **Skip validation** in the beginning (no impact on ALC)

Best-last-predictions **ensemble strategy**

Module	Input Spectrogram ($257 \times T \times 1$)	Output Size
Thin ResNet	conv2d, 7×7, 64	$257 \times T \times 64$
	max pool, 2×2, stride $(2, 2)$	$128 \times T/2 \times 64$
	$\begin{bmatrix} \text{conv}, 1 \times 1, 48 \\ \text{conv}, 3 \times 3, 48 \\ \text{conv}, 1 \times 1, 96 \end{bmatrix} \times 2$	$128 \times T/2 \times 96$
	$\begin{bmatrix} \text{conv}, 1 \times 1, 96 \\ \text{conv}, 3 \times 3, 96 \\ \text{conv}, 1 \times 1, 128 \end{bmatrix} \times 3$	$64 \times T/4 \times 128$
	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 256 \end{bmatrix} \times 3$	$32 \times T/8 \times 256$
	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 512 \end{bmatrix} \times 3$	$16 \times T/16 \times 512$
	max pool, 3×1, stride $(2, 2)$	$7 \times T/32 \times 512$
	conv2d, 7×1, 512	$1 \times T/32 \times 512$

Text domain

Various preprocessing methods: word frequency filtering, word segmentation, sentences truncating.

Features engineering :

- word / char level features, tokenization with metadata as guideline.
- word embedding : Fast text embedding [7] (pre-trained or reinitialized)

Models :

- include TextCNN, RCNN, GRU, GRU with attention and SVM.
- using datasets metadata to find training parameters

Weighted ensembling : returns weighted top 20 models (for NAUC).

Tabular domain

Use optimized **boosting frameworks** such as:
LightGBM, XGBoost, CatBoost and DNN.

A **weighted ensemble strategy** is applied:

The train set $D = (x_1, y_1, ..., x_n, y_n)$ is divided in **K**-folds, which are themselves divided in 3 sets (one for training, one for hyperparameter search and one for weights computing) :

$$\{(D_{train}^{(1)}, D_{valid1}^{(1)}, D_{valid2}^{(1)}), ..., (D_{train}^{(k)}, D_{valid1}^{(k)}, D_{valid2}^{(k)})\}$$

Ensemble learning strategy

Ensemble learning : a stacking/blending strategy

Algorithm 2: AutoEnsemble

Require: F: Model Space, D: Data Space, M: metric

Require: k: the number of data folds, m: the number of models,

stratified splits D into k folds $\{(D^{(1)}_{train}, D^{(1)}_{valid1}, D^{(1)}_{valid2}), ..., (D^{(k)}_{train}, D^{(k)}_{valid1}, D^{(k)}_{valid2})\}$

for $i \in [1, k]$ **do**

  for $j \in [1, m]$ **do**

    $f^{j}_{\lambda^*}(\theta^*) = \arg\min_{\theta,\lambda}(L(f^{j}_{\lambda}(\theta), D^{(i)}_{train}, D^{(i)}_{valid1}))$

    $score_{ij} = M(f^{j}_{\lambda^*}(D_{valid2}|\theta^*))$

    $preds_{ij} = f^{j}_{\lambda^*}(D_{test}|\theta^*)$

  end for

end for

$$w_{ij} = \frac{score_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{m}(score_{ij})}$$

$$preds = \sum_{i=1}^{k}\sum_{j=1}^{m}(w_{ij} * preds_{ij})$$

Summary

Modality	Non-neural	Neural	Pretrained	Generalization/ Acceleration
Image	-	ResNet9 ResNet18	ImageNet	re-initializing
Video	-	MC3	Kinetics	re-initializing freezing
Speech	LR	ThinResnet34	VoxCeleb2	freezing
Text	SVM	TextCNN/RCNN GRU/GRUA	-	-
Tabular	LightGBM/Xgboost Catboost	DNN	-	-

IV. Benchmarking the AutoDL solutions

Settings

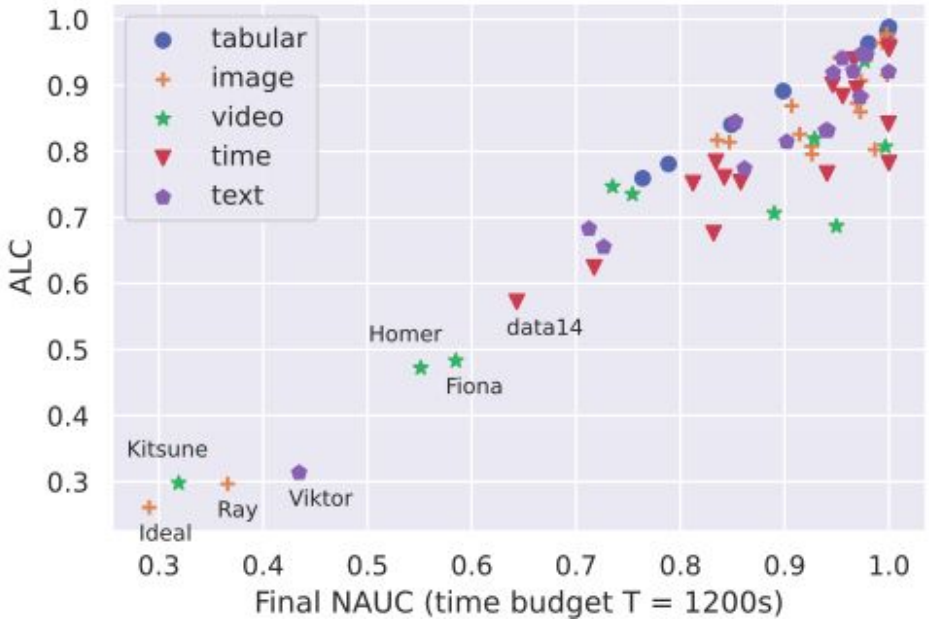
13 models

- 9 participants (DeepWisdom, DeepBlueAI, PasaNJU, AutoMLFreiburg, Inspur_AutoDL, Frozenmad, TeamZhaw, Surromind, Kon)
- 4 baselines (Baseline1: Linear NN, Baseline2: 3DCNN, Baseline3: All winner solution, Pre-trained Inception)

66 datasets

- 17 image datasets
- 10 video datasets
- 16 text datasets
- 16 time series datasets
- 7 tabular datasets

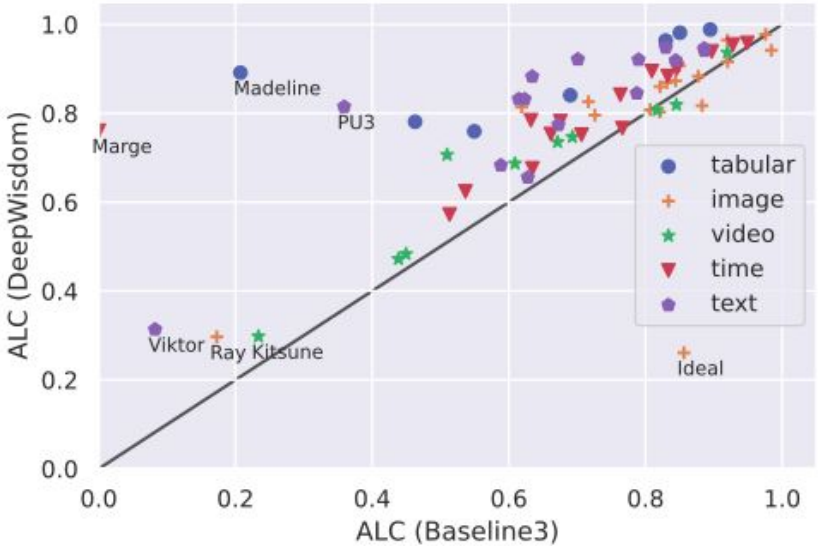
Deep Wisdom results

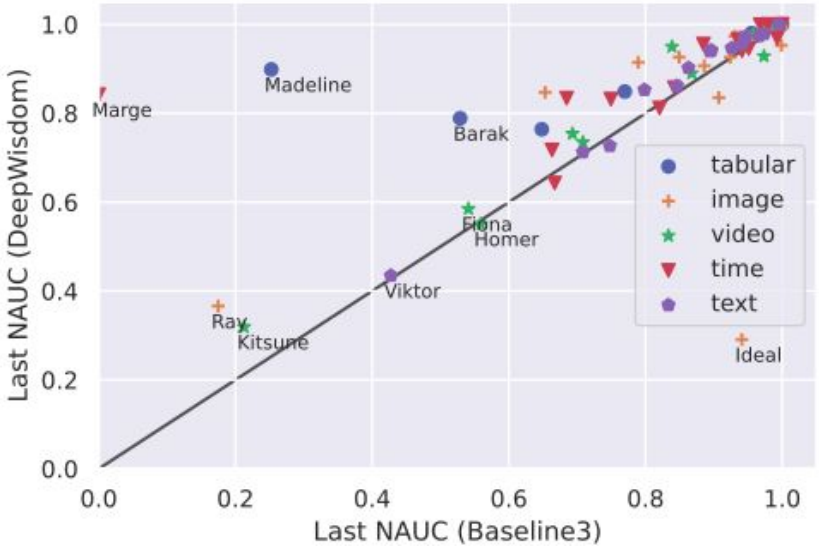


Worst datasets:

- Ideal (avg ALC: 0.7737)
- Kitsune (avg ALC: 0.2093)
- Ray (avg ALC: 0.2555)
- Viktor (avg ALC: 0.2763)

Deep Wisdom v. Baseline3

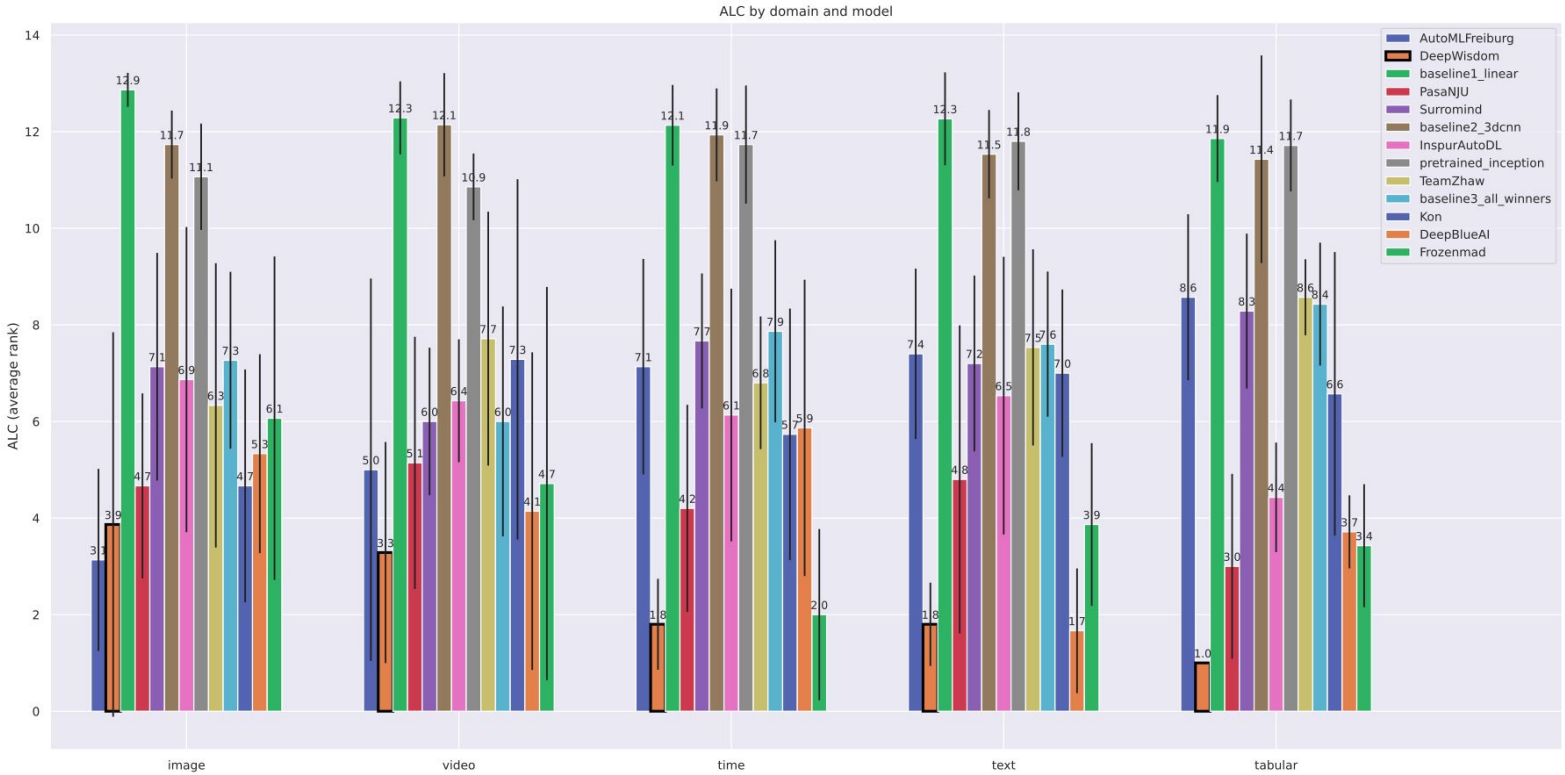




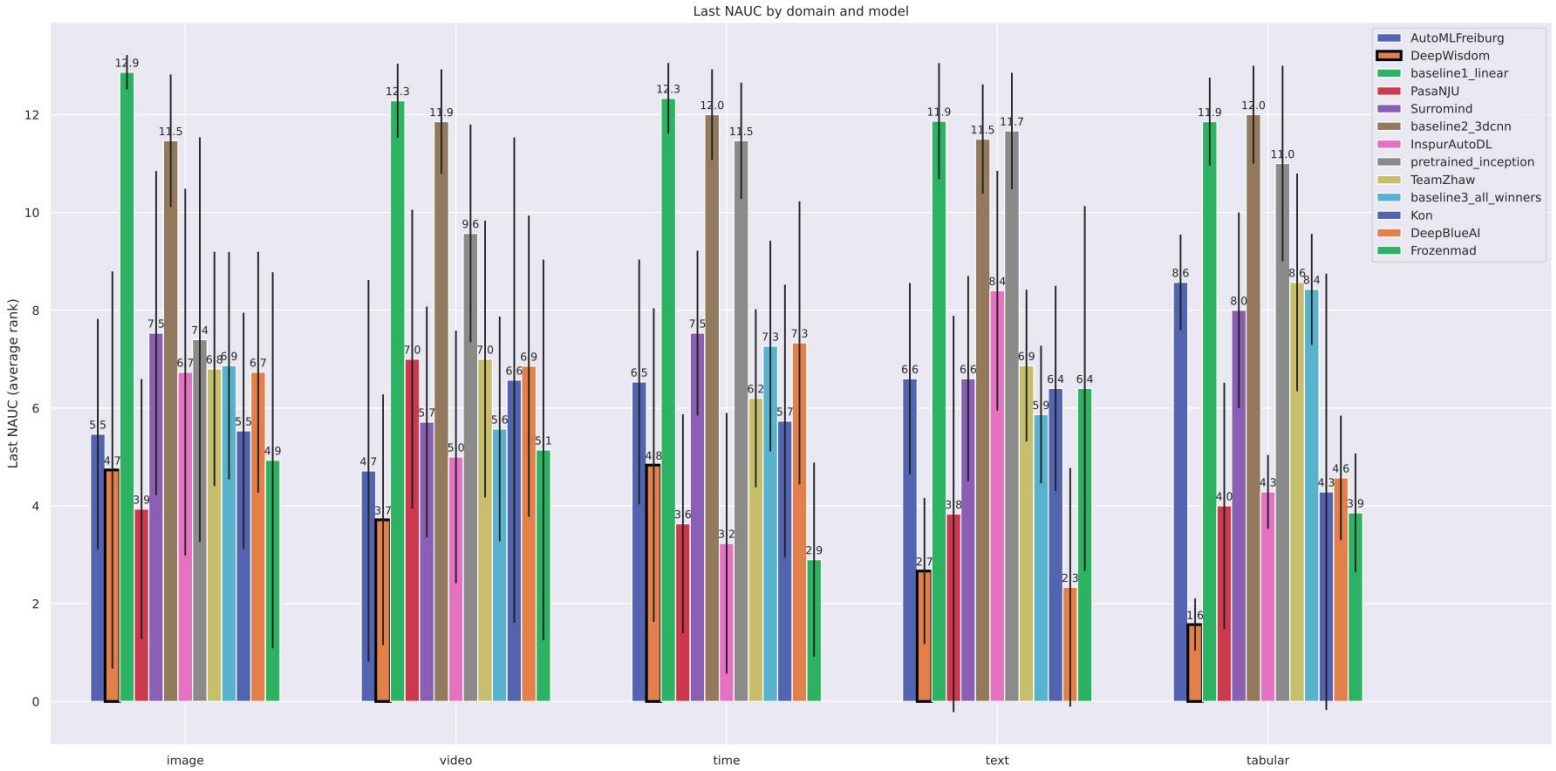
Time comparison with Baseline3

	Image	Video	Time	Text	Tabular	Overall
Average duration Deep Wisdom (s)	571.1211	937.5797	**900.0690**	577.6960	870.2734	742.2048
Average duration Baseline3 (s)	**461.5072**	**488.4777**	1067.8111	**195.9854**	**305.7676**	**536.8543**
Average first prediction duration Deep Wisdom (s)	**10.2858**	**17.2410**	**12.0937**	**9.4392**	**0.8963**	**10.5709**
Average first prediction duration Baseline3 (s)	11.9154	20.2322	28.8647	97.0922	13.6596	37.3415

ALC benchmark on each domain



Last NAUC benchmark on each domain



Conclusion

Try it out!

Deep Wisdom : multimodal, no unified AutoDL framework emerged.

References

[1] 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,* 2019.
[2] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. 2009. p. 248–55.
[3] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim and Sungwoong Kim. Fast AutoAugment. 2019. ArXiv eprint: 1905.00397
[4] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In CVPR, pages 6450–6459, 2018.
[5] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman. Utterance-level aggregation for speaker recognition in the wild. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pages 5791–5795, 2019.
[6] Joon Son Chung, Arsha Nagrani and Andrew Zisserman. VoxCeleb2: Deep Speaker Recognition. In *Interspeech 2018*. ISCA, 2018.
[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606. 2016.