Incorporating Task-Agnostic Information in Task-Based Active Learning Using a Variational Autoencoder

Curtis Godwin[‡][†][*], Meekail Zain[§][†][*], Nathan Safir[‡], Bella Humphrey[§], Shannon P Quinn[§][¶]

Abstract—It is often much easier and less expensive to collect data than to label it. Active learning (AL) ([Set09]) responds to this issue by selecting which unlabeled data are best to label next. Standard approaches utilize task-aware AL, which identifies informative samples based on a trained supervised model. Task-agnostic AL ignores the task model and instead makes selections based on learned properties of the dataset. We seek to combine these approaches and measure the contribution of incorporating task-agnostic information into standard AL, with the suspicion that the extra information in the task-agnostic features may improve the selection process. We test this on various AL methods using a ResNet classifier with and without added unsupervised information from a variational autoencoder (VAE). Although the results do not show a significant improvement, we investigate the effects on the acquisition function and suggest potential approaches for extending the work.

Index Terms—active learning, variational autoencoder, deep learning, pytorch, semi-supervised learning, unsupervised learning

Introduction

In deep learning, the capacity for data gathering often significantly outpaces the labeling. This is easily observed in the field of bioimaging, where ground-truth labeling usually requires the expertise of a clinician. For example, producing a large quantity of CT scans is relatively simple, but having them labeled for COVID-19 by cardiologists takes much more time and money. These constraints ultimately limit the contribution of deep learning to many crucial research problems.

This labeling issue has compelled advancements in the field of active learning (AL) ([Set09]). In a typical AL setting, there is a set of labeled data and a (usually larger) set of unlabeled data. A model is trained on the labeled data, then the model is analyzed to evaluate which unlabeled points should be labeled to best improve the loss objective after further training. AL acknowledges labeling constraints by specifying a budget of points that can be labeled at a time and evaluating against this budget.

In AL, the model for which we select new labels is referred to as the task model. If this model is a classifier neural network, the space in which it maps inputs before classifying them is known as the latent space or representation space. A recent branch of AL ([SS18], [SCN+18], [YK19]), prominent for its applications to deep models, focuses on mapping unlabeled points into the task model's latent space before comparing them.

These methods are limited in their analysis by the labeled data they must train on, failing to make use of potentially useful information embedded in the unlabeled data. We therefore suggest that this family of methods may be improved by extending their representation spaces to include unsupervised features learned over the entire dataset. For this purpose, we opt to use a variational autoencoder (VAE) ([KW13]) , which is a prominent method for unsupervised representation learning. Our main contributions are (a) a new methodology for extending AL methods using VAE features and (b) an experiment comparing AL performance across two recent feature-based AL methods using the new method.

Related Literature

Active learning

Much of the early active learning (AL) literature is based on shallower, less computationally demanding networks since deeper architectures were not well-developed at the time. Settles ([Set09]) provides a review of these early methods. The modern approach uses an acquisition function, which involves ranking all available unlabeled points by some chosen heuristic $\mathscr{H}$ and choosing to label the points of highest ranking.

Algorithm 1: Measuring the performance of a given active learning heuristic

```
Input: training dataset $D$, task model $\mathscr{T}$, budget $\beta$, initial number of labels $\xi$,
       desired number of labels $\eta$, set selection heuristic $\mathscr{H}$
$L \leftarrow \xi$-sized random subset of $D$
$U \leftarrow D \backslash L$
$A \leftarrow \varnothing$
train $\mathscr{T}$ on $L$
while $|L| \leq \eta$ do
    $S \leftarrow \beta$-sized subset of $U$, selected using $\mathscr{H}$
    $L \leftarrow L \cup S$
    retrain or fine-tune $\mathscr{T}$ on $L$
    $a \leftarrow$ (validation accuracy of $\mathscr{T}, |L|$)        // save accuracy tuple
    $A \leftarrow A \cup a$        // record accuracies across the labeling process
create a line graph plotting $a_0$ against $a_1$ for each $a \in A$
```

The popularity of the acquisition approach has led to a widely-used evaluation procedure, which we describe in Algorithm 1.

[†] *These authors contributed equally.*
[*] *Corresponding author: cmgodwin263@gmail.com, meekail.zain@uga.edu*
[‡] *Institute for Artificial Intelligence, University of Georgia, Athens, GA 30602 USA*
[*] *Corresponding author: cmgodwin263@gmail.com, meekail.zain@uga.edu*
[§] *Department of Computer Science, University of Georgia, Athens, GA 30602 USA*
[¶] *Department of Cellular Biology, University of Georgia, Athens, GA 30602 USA*

This procedure trains a task model $\mathscr{T}$ on the initial labeled data, records its test accuracy, then uses $\mathscr{H}$ to label a set of unlabeled points. We then once again train $\mathscr{T}$ on the labeled data and record its accuracy. This is repeated until a desired number of labels is reached, and then the accuracies can be graphed against the number of available labels to demonstrate performance over the course of labeling. We can use this evaluation algorithm to separately evaluate multiple acquisition functions on their resulting accuracy graphs. This is utilized in many AL papers to show the efficacy of their suggested heuristics in comparison to others ([WZL+16], [SS18], [SCN+18], [YK19]).

The prevailing approach to point selection has been to choose unlabeled points for which the model is most uncertain, the assumption being that uncertain points will be the most informative ([BRK21]). A popular early method was to label the unlabeled points of highest Shannon entropy ([Sha48]) under the task model, which is a measure of uncertainty between the classes of the data. This method is now more commonly used in combination with a representativeness measure ([WZL+16]) to avoid selecting condensed clusters of very similar points.

Recent heuristics using deep features

For convolutional neural networks (CNNs) in image classification settings, the task model $\mathscr{T}$ can be decomposed into a feature-generating module

$$\mathscr{T}_f \colon \mathbb{R}^n \to \mathbb{R}^f,$$

which maps the input data vectors to the output of the final fully connected layer before classification, and a classification module

$$\mathscr{T}_c \colon \mathbb{R}^f \to \{0, 1, ..., c\},$$

where c is the number of classes.

Recent deep learning-based AL methods have approached the notion of model uncertainty in terms of the rich features generated by the learned model. Core-set ([SS18]) and MedAL ([SCN+18]) select unlabeled points that are the furthest from the labeled set in terms of L_2 distance between the learned features. For core-set, each point constructing the set S in step 6 of Algorithm 1 is chosen by

$$\mathbf{u}^* = \underset{\mathbf{u}\in U}{\operatorname{argmax}} \min_{\boldsymbol{\ell}\in L} ||(\mathscr{T}_f(\mathbf{u}) - \mathscr{T}_f(\boldsymbol{\ell}))||^2, \tag{1}$$

where U is the unlabeled set and L is the labeled set. The analogous operation for MedAL is

$$\mathbf{u}^* = \underset{\mathbf{u}\in U}{\operatorname{argmax}} \frac{1}{|L|} \sum_{i=1}^{|L|} ||\mathscr{T}_f(\mathbf{u}) - \mathscr{T}_f(\mathbf{L_i})||^2. \tag{2}$$

Note that after a point $\mathbf{u}^*$ is chosen, the selection of the next point assumes the previous $\mathbf{u}^*$ to be in the labeled set. This way we discourage choosing sets that are closely packed together, leading to sets that are more diverse in terms of their features. This effect is more pronounced in the core-set method since it takes the minimum distance whereas MedAL uses the average distance.

Another recent method ([YK19]) trains a regression network to predict the loss of the task model, then takes the heuristic $\mathscr{H}$ in Algorithm 1 to select the unlabeled points of highest predicted loss. To implement this, the loss prediction network $\mathscr{P}$ is attached to a ResNet task model $\mathscr{T}$ and is trained jointly with $\mathscr{T}$. The inputs to $\mathscr{P}$ are the features output by the ResNet's four residual blocks. These features are mapped into the same dimensionality via a fully connected layer and then concatenated to form a representation $\mathbf{c}$. An additional fully connected layer then maps $\mathbf{c}$ into a single value constituting the loss prediction.

When attempting to train a network to directly predict $\mathscr{T}$'s loss during training, the ground truth losses naturally decrease as $\mathscr{T}$ is optimized, resulting in a moving objective. The authors of ([YK19]) find that a more stable ground truth is the inequality between the losses of given pairs of points. In this case, $\mathscr{P}$ is trained on pairs of labeled points, so that $\mathscr{P}$ is penalized for producing predicted loss pairs that exhibit a different inequality than the corresponding true loss pair.

More specifically, for each batch of labeled data $L_{batch} \subset L$ that is propagated through $\mathscr{T}$ during training, the batch of true losses is computed and split randomly into a batch of pairs P_{batch}. The loss prediction network produces a corresponding batch of predicted loss pairs, denoted $\widetilde{P}_{batch}$. The following pair loss is then computed given each $p \in P_{batch}$ and its corresponding $\tilde{p} \in \widetilde{P}_{batch}$:

$$\mathscr{L}_{pair}(p, \tilde{p}) = \max(0, -\mathscr{I}(p) \cdot (\tilde{p}^{(1)} - \tilde{p}^{(2)}) + \xi), \tag{3}$$

where $\mathscr{I}$ is the following indicator function for pair inequality:

$$\mathscr{I}(p) = \begin{cases} 1, & p^{(1)} > p^{(2)} \\ -1, & p^{(1)} \leq p^{(2)} \end{cases}. \tag{4}$$

Variational Autoencoders

Variational autoencoders (VAEs) ([KW13]) are an unsupervised method for modeling data using Bayesian posterior inference. We begin with the Bayesian assumption that the data is well-modeled by some distribution, often a multivariate Gaussian. We also assume that this data distribution can be inferred reasonably well by a lower dimensional random variable, also often modeled by a multivariate Gaussian.

The inference process then consists of an encoding into the lower dimensional latent variable, followed by a decoding back into the data dimension. We parametrize both the encoder and the decoder as neural networks, jointly optimizing their parameters with the following loss function ([KW19]):

$$\mathscr{L}_{\theta,\phi}(\mathbf{x}) = \log p_\theta(\mathbf{x}|\mathbf{z}) + [\log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})], \tag{5}$$

where θ and ϕ are the parameters of the encoder and the decoder, respectively. The first term is the reconstruction error, penalizing the parameters for producing poor reconstructions of the input data. The second term is the regularization error, encouraging the encoding to resemble a pre-selected prior distribution, commonly a unit Gaussian prior.

The encoder of a well-optimized VAE can be used to generate latent encodings with rich features which are sufficient to approximately reconstruct the data. The features also have some geometric consistency, in the sense that the encoder is encouraged to generate encodings in the pattern of a Gaussian distribution.

Methods

We observe that the notions of uncertainty developed in the core-set and MedAL methods rely on distances between feature vectors modeled by the task model $\mathscr{T}$. Additionally, loss prediction relies on a fully connected layer mapping from a feature space to a single value, producing different predictions depending on the values of the relevant feature vector. Thus all of these methods utilize spatial reasoning in a vector space.

Furthermore, in each of these methods, the heuristic $\mathscr{H}$ only has access to information learned by the task model, which is trained only on the labeled points at a given timestep in the labeling procedure. Since variational autoencoder (VAE) encodings are not limited by the contents of the labeled set, we suggest that the aforementioned methods may benefit by expanding the vector spaces they investigate to include VAE features learned across the entire dataset, including the unlabeled data. These additional features will constitute representative and previously inaccessible information regarding the data, which may improve the active learning process.

We implement this by first training a VAE model $\mathscr{V}$ on the given dataset. $\mathscr{V}$ can then be used as a function returning the VAE features for any given datapoint. We append these additional features to the relevant vector spaces using vector concatenation, an operation we denote with the symbol $\frown$. The modified point selection operation in core-set then becomes

$$\mathbf{u}^* = \operatorname*{argmax}_{\mathbf{u}\in U} \min_{\boldsymbol{\ell}\in L} ||([\mathscr{T}_f(\mathbf{u}) \frown \alpha\mathscr{V}(\mathbf{u})] - [\mathscr{T}_f(\boldsymbol{\ell}) \frown \alpha\mathscr{V}(\boldsymbol{\ell})]||^2, \tag{6}$$

where α is a hyperparameter that scales the influence of the VAE features in computing the vector distance. To similarly modify the loss prediction method, we concatenate the VAE features to the final ResNet feature concatenation $\mathbf{c}$ before the loss prediction, so that the extra information is factored into the training of the prediction network $\mathscr{P}$.

Experiments

In order to measure the efficacy of the newly proposed methods, we generate accuracy graphs using Algorithm 1, freezing all settings except the selection heuristic $\mathscr{H}$. We then compare the performance of the core-set and loss prediction heuristics with their VAE-augmented counterparts.

We use ResNet-18 pretrained on ImageNet as the task model, using the SGD optimizer with learning rate 0.001 and momentum 0.9. We train on the MNIST ([Den12]) and ChestMNIST ([YSN21]) datasets. ChestMNIST consists of 112,120 chest X-ray images resized to 28x28 and is one of several benchmark medical image datasets introduced in ([YSN21]).

For both datasets we experiment on randomly selected subsets, using 25000 points for MNIST and 30000 points for ChestMNIST. In both cases we begin with 3000 initial labels and label 3000 points per active learning step. We opt to retrain the task model after each labeling step instead of fine-tuning.

We use a similar training strategy as in ([SCN+18]), training the task model until >99% train accuracy before selecting new points to label. This ensures that the ResNet is similarly well fit to the labeled data at each labeling iteration. This is implemented by training for 10 epochs on the initial training set and increasing the training epochs by 5 after each labeling iteration.

The VAEs used for the experiments are trained for 20 epochs using an Adam optimizer with learning rate 0.001 and weight decay 0.005. The VAE encoder architecture consists of four convolutional downsampling filters and two linear layers to learn the low dimensional mean and log variance. The decoder consists of an upsampling convolution and four size-preserving convolutions to learn the reconstruction.

Experiments were run five times, each with a separate set of randomly chosen initial labels, with the displayed results showing the average validation accuracies across all runs. Figures 1 and 3 show the core-set results, while Figures 2 and 4 show the loss prediction results. In all cases, shared random seeds were used to ensure that the task models being compared were supplied with the same initial set of labels.

With four NVIDIA 2080 GPUs, the total runtime for the MNIST experiments was 5113s for core-set and 4955s for loss prediction; for ChestMNIST, the total runtime was 7085s for core-set and 7209s for loss prediction.

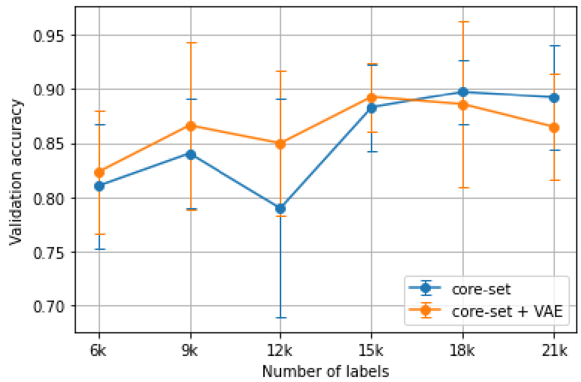

Fig. 1: The average MNIST results using the core-set heuristic versus the VAE-augmented core-set heuristic for Algorithm 1 over 5 runs.

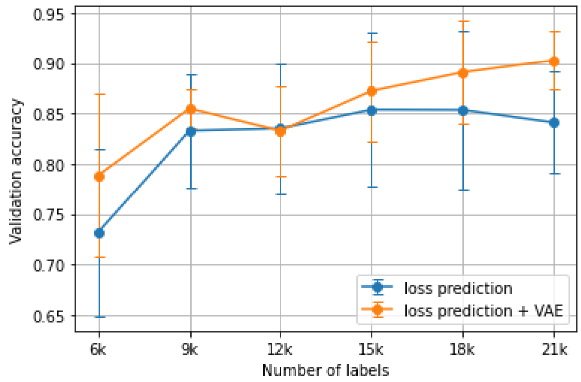

Fig. 2: The average MNIST results using the loss prediction heuristic versus the VAE-augmented loss prediction heuristic for Algorithm 1 over 5 runs.

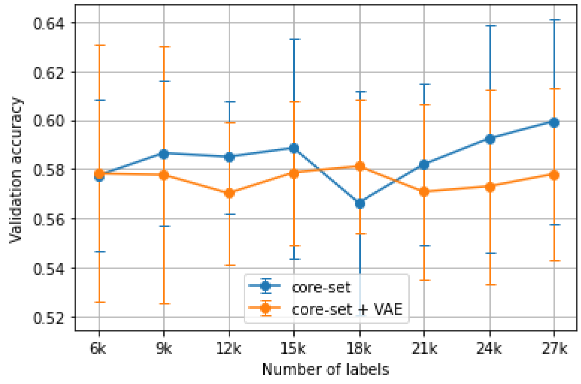

Fig. 3: The average ChestMNIST results using the core-set heuristic versus the VAE-augmented core-set heuristic for Algorithm 1 over 5 runs.

To investigate the qualitative difference between the VAE and non-VAE approaches, we performed an additional experiment

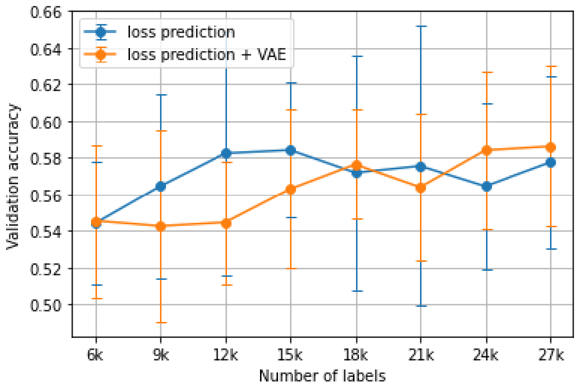

Fig. 4: *The average ChestMNIST results using the loss prediction heuristic versus the VAE-augmented loss prediction heuristic for Algorithm 1 over 5 runs.*

to visualize an example of core-set selection. We first train the ResNet-18 with the same hyperparameter settings on 1000 initial labels from the ChestMNIST dataset, then randomly choose 1556 (5%) of the unlabeled points from which to select 100 points to label. These smaller sizes were chosen to promote visual clarity in the output graphs.

We use t-SNE ([VdMH08]) dimensionality reduction to show the ResNet features of the labeled set, the unlabeled set, and the points chosen to be labeled by core-set.

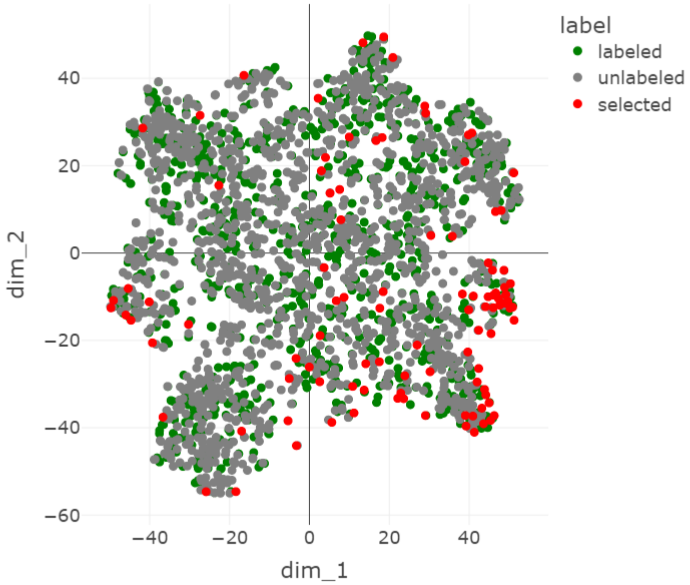

Fig. 5: *A t-SNE visualization of the ChestMNIST points chosen by core-set.*

Discussion

Overall, the VAE-augmented active learning heuristics did not exhibit a significant performance difference when compared with their counterparts. The only case of a significant p-value (<0.05) occurred during loss prediction on the MNIST dataset at 21000 labels.

The t-SNE visualizations in Figures 5 and 6 show some of the influence that the VAE features have on the core-set selection process. In 5, the selected points tend to be more spread out, while in 6 they cluster at one edge. This appears to mirror the transformation of the rest of the data, which is more spread out without the VAE features, but becomes condensed in the center when they are introduced, approaching the shape of a Gaussian distribution.

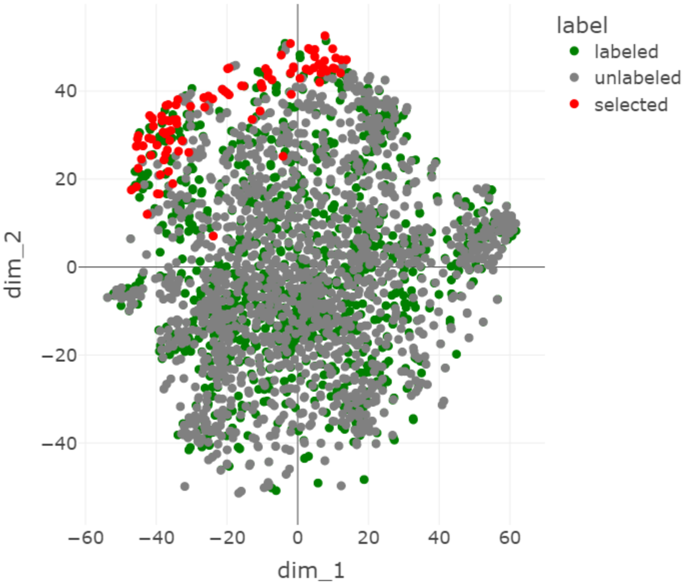

Fig. 6: *A t-SNE visualization of the ChestMNIST points chosen by core-set when the ResNet features are augmented with VAE features.*

It seems that with the added VAE features, the selected points are further out of distribution in the latent space. This makes sense because points tend to be more sparse at the tails of a Guassian distribution and core-set prioritizes points that are well-isolated from other points.

One reason for the lack of performance improvement may be the homogeneous nature of the VAE, where the optimization goal is reconstruction rather than classification. This could be improved by using a multimodal prior in the VAE, which may do a better job of modeling relevant differences between points.

Conclusion

Our original intuition was that additional unsupervised information may improve established active learning methods, especially when using a modern unsupervised representation method such as a VAE. The experimental results did not indicate this hypothesis, but additional investigation of the VAE features showed a notable change in the task model latent space. Though this did not result in superior point selections in our case, it is of interest whether different approaches to latent space augmentation in active learning may fare better.

Future work may explore the use of class-conditional VAEs in a similar application, since a VAE that can utilize the available class labels may produce more effective representations, and it could be retrained along with the task model after each labeling iteration.

References

[BRK21] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021. doi:10.1016/j.media.2021.102062.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi:10.1109/MSP.2012.2211477.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KW19] Diederik P. Kingma and Max Welling. *An Introduction to Variational Autoencoders*. Now Publishers, 2019. URL: https://doi.org/10.1561%2F9781680836233, doi:10.1561/9781680836233.

[SCN+18] Asim Smailagic, Pedro Costa, Hae Young Noh, Devesh Walawalkar, Kartik Khandelwal, Adrian Galdran, Mostafa Mirshekari, Jonathon Fagert, Susu Xu, Pei Zhang, et al. Medal: Accurate and robust deep active learning for medical image analysis. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 481–488. IEEE, 2018. doi:10.1109/icmla.2018.00078.

[Set09] Burr Settles. Active learning literature survey. 2009.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[SS18] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL: https://openreview.net/forum?id=H1aIuk-RW.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[WZL+16] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016. doi:10.1109/tcsvt.2016.2589879.

[YK19] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019. doi:10.1109/CVPR.2019.00018.

[YSN21] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. doi:10.1109/ISBI48211.2021.9434062.