# Estimating Data Requirements for Learning-Enabled Systems using Metadata

Chi Phuong Ngoc Huynh
*Department of Computer Science*
*Vanderbilt University*
Nashville, TN, United States
chi.phuong.ngoc.huynh@vanderbilt.edu

James Weimer
*Department of Computer Science*
*Vanderbilt University*
Nashville, TN, United States
james.weimer@vanderbilt.edu

2024 International Conference on Assured Autonomy (ICAA) | 979-8-3315-2101-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAA64256.2024.00015

***Abstract*—In machine learning, the efficiency and reliability of models are critically dependent on the quantity and quality of data used for training. Despite this, accurately estimating the data requirements necessary to achieve optimal model performance remains a significant challenge – especially in applications where target domain data is unavailable to inform data requirement estimation. The primary aim of this work is to demonstrate the potential for estimating data requirements using metadata prior to data collection – which could help transition data requirement estimation in practice from solely expert-driven to incorporating data-driven elements. Consequently, this paper presents a novel framework designed to assess data requirements, prior to data collection, for machine learning models that aims to enhance both efficiency and assurance. To show the promise of the proposed framework, we introduce a novel metadata dataset, ImageMeta, that records various features of models and existing datasets collected from public repositories. Leveraging ImageMeta, a model of the proposed framework is realized that can estimate data requirements for a host of computer vision tasks. Evaluations demonstrate that even with a simple architecture, models still performed well on evaluating difficulty of tasks and target data. These results establish that data requirements can be estimated from metadata alone and provides valuable guidelines for practitioners/experimentalists seeking to optimize data collection methods and improve model robustness.**

***Index Terms*—data requirements, metadata-based estimation, learning-enabled systems, machine learning**

## I. Introduction

Machine learning algorithms are increasingly being deployed in safety-critical autonomous systems, spanning manufacturing [1], [2], transportation [3], and medicine [4], [5]. Broadly speaking, traditional approaches and best practices in developing these learning-enabled autonomous systems involve three sequential efforts. First, engineered experiments are conducted to collect data for machine learning. Second, machine learning is applied to the collected data to generate a model that is trained, validated, and tested for some combination of liveliness and safety. Lastly, the integrated learning-enabled autonomous system is (extensively) tested for assurance. With appropriate oversight and safe guards, the traditional approach involving first data collection, then machine learning, and lastly real-world testing has proven effective for designing safe autonomous systems.

However, the amount and quality of collected data needed to learn an effective model continues to plague all early-stage safe autonomy development efforts – especially in learning-enabled applications where pilot/initial data collection involves significant costs and resources (*e.g.,* stroke detection [6] or postpartum hemorrhage risk prediction [7]). Complicating the matter, and especially in many medical applications, even collecting a small representative dataset can be problematic and require years of costly engineering experimental platform development and regulatory approvals. For example, our recent work on stroke detection [6] required 3 years to collect pilot data and postpartum hemorrhage risk prediction [7] required 1.5 years to collect pilot data (including experimental platform development) – all before establishing whether the data was even sufficient to achieve the stated goal (*e.g.,* detect stroke or predict postpartum hemorrhage risk). In both scenarios, experiments to collect data were performed and informed relying predominantly on prior experience and domain knowledge, which introduces significant development risks. To reduce this risk in these specialized applications, an approach that can accurately predict experimentally-collected data requirements *and* the corresponding performance of the trained classifier *prior* to experimentally collecting/generating any representative data is needed.

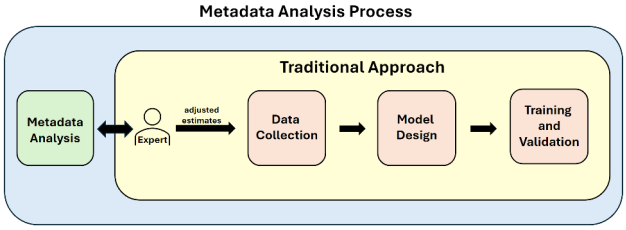


Fig. 1: **Metadata Analysis as a Guide for the Research Process** We propose a metadata analysis step that works as a guideline for the data collection and experimentation process. The output of the metadata analysis model can provide advice to the expert on the amount of data to compile, given some information on task and desired performance.

The problem of estimating data requirements for machine learning/safe autonomy is not fundamentally new – but remains largely unresolved, especially when pilot data is unavailable. Practically speaking, data requirements are often

979-8-3315-2101-1/24/$31.00 ©2024 IEEE
DOI 10.1109/ICAA64256.2024.00015

driven by experience and funding. In real-world learning-enabled component design, expert intuition/experience is heavily relied upon to effectively guess data requirements prior to data collection – often resulting in costly under- and over-approximation errors that drives experimentalists to collect as much data as can be afforded. To address these practical issues, theoretical bounds [8] and difficulty-based estimates [9]–[11] have been introduced, but they produce either overly conservative estimates, require a pre-specification of the model architecture, and/or require a small sample dataset. Individually testing architectures on specific dataset give results that only highlights the drawback of the model rather than the samples. Moreover, existing methods are time-consuming to implement for many samples, and often lack generalizability among tasks and specificity to the data instead of the architecture. Lastly, we note that power analysis techniques used to estimate data requirements (as commonly employed in clinical trial designs) can not be directly applied since access to *a priori* estimates of the performance/effect size is required – and are likely unavailable prior to initial data collection.

Consequently, this work seeks to demonstrate a proof-of-possible solution to one main question:

*Can the number of samples needed for machine learning be practically estimated before collecting training data and choosing a model architecture?*

Previous work have shown that additional steps such as feature selection or other data pre-processing methods work to increase data quality and performance of models [12]–[14]. Similar to how these methods address data quality, our work focus on the issue of data quantity. It could be augmented as a step in the research process before building the model. This step can assess the difficulty of a problem in the context of available models. If a task has low performance from the existing reported models then it is considered more difficult. Given a specific task and desired accuracy, the model outputs the difficulty and size of the target dataset. This process can provide more confidence in the number of sufficient and relevant samples collected. Estimating the data requirements can hence assure machine learning efficiency. Our method further streamlines the process by completing the calculations without any sample data required from the target dataset.

Compared to previous work, our method differs in that it does not rely on particular data samples but on the performance of previous models. This allows us to build a larger database of datasets and their respective difficulty. We assume that users are capable of estimating the difficulty of their target data using expert intuition. This allows the models to be more generalized for a variety of different tasks. Existing methods of analyzing data typically require several samples, delaying the development process for collection of additional data. Requiring no specific data sample means our model can significantly shorten the time and computation power needed to analyze and estimate data requirements.

To demonstrate proof-of-possibility, we compiled ImageMeta, a dataset geared toward the task of estimating potential performance and the number of required samples. To our knowledge, this is the first dataset of this scale that contains information on dataset difficulty and other metadata. This is made possible by our use of our data-agnostic dataset analysis metric, Maximal Performance Index (MPI), which is highly efficient to calculate. While ImageMeta focuses on images due to their wide availability and prevalence in computer vision research, our future work involves expanding ImageMeta to include other dataset types (*e.g.,* time-series). In this work, aimed at establishing the feasibility/proof-of-possiblity of estimating data requirements for learning-enabled systems using metadata, we limit our focus (temporarily) to images. However, we note that our process for data collection and processing can be applied to any subfields to further broaden the scope of this dataset. With this expansion, we believe the resulting dataset could encompass every machine learning task available and reduce development time when applying ML in new learning-enabled applications.

Leveraging ImageMeta, we demonstrate three potential architectures for the prediction of sample size based on the desired performance. The models show promising results, with the MSE being below 0.1 , showing a high ability to adapt to the patterns shown in the dataset. With further refinement, a model based on these findings can reliably make prediction on the necessary samples for a significant effect size for future research projects and practical applications. In comparison to existing methods, using maximal performance as a proxy for difficulty is data-agnostic and much more efficient. Without needing intensive computing, we are able to gather a large dataset of datasets, tasks, their associated difficulty and models' performance.

In summary, the main contributions of our work are:

- A metadata-based approach to estimating difficulty prior to data analysis and collection that does not require prior target-domain samples or specifying a model architecture.
- Developing and open-sourcing a comprehensive dataset, ImageMeta, that catalogs metadata across various computer vision datasets and tasks to study the problem of data requirement estimation using metadata.
- Implementation and evaluation of the metadata-based approach on the ImageMeta dataset establishing it is possible to estimate data difficulty and sample size for a machine learning task prior to collecting target-domain data.

We introduce our work and its motivations in Section I. Section II discusses previous related works on difficulty estimation for data. Section III formally states our problem. Section IV details our methods for generating our dataset and models. In Section V, we describe the data gathering and processing implementation details. Section VI shows the experimental setup and results. Section VII evaluates our dataset and architecture. We end with the conclusion and discussion of future work.

## II. Background and Related Work

Addressing the ambiguity of data sufficiency requires distinctions between different data samples and tasks. Not all machine learning tasks are uniform in its difficulty. Even within the task of classification, there is a big difference in difficulty among datasets. For a harder problem, you would need more samples than for an easier, "solved", problem. To adequately address this factor, we must address the difficulty of existing and target datasets. In this section, we discuss the previous methods used to analyze the difficulty of data, including dynamic methods - running a model to examine their efficacy, and static methods - using algorithms and functions to dissect their properties.

### A. Performance-based method to measure Difficulty

There have been some previous papers in the domain of computer vision that measure the difficulty of a dataset. In this section, we discuss models that require performing the target task to determine the difficulty of a sample. Performance in this context could either be measured via machine learning models or human testers.

In previous work, human participants have been used as a benchmark for machine learning capabilities. They can complete the task, such as classification or enumeration of objects, as a comparison to the performance of machines. Information on how long they spend time looking at a sample can also provide some information on difficulty. In [10], [15], the authors present image samples to human testers with a task and record the viewing time. The association is that the longer a tester takes to view the image, the more difficult it is. This process is too inefficient to be replicated with many datasets. Additionally, with more complicated tasks such as tumor detection, expert tester is required, which is difficult to arrange.

Machine-based dynamic methods test the performance of a single model or architecture on a dataset to extract its difficulty level. Usable information or usable-$\mathcal{V}$, is measured by running a model on a gold-label example and a noise sample to determine the difference in confidence for the correct category [16]. In this work, the certainty of correct classification between the true sample and the noise one shows the amount of feature-related information provided by the sample. Another approach is to train and validate a simplified network with some reduction in time versus fine-tuning a full model [17]. In [18], they used gradients to determine difficulty of samples within their class for classification problems. These models improve on the traditional methods of using human judgment for determining difficulty or fully training models on a dataset to assess its difficulty since they are more efficient and simpler to reproduce. Since they use a model for generating the difficulty values, they can specifically show the fit between that architecture and the dataset. A drawback to this method is they require more time for training and testing the datasets. Individual sampling and calculation is possible if only a few datasets are present. However, for a large number of datasets and samples, it is not realistic. Another drawback is its partiality: while some models might suit a dataset, others might not. Their low performance may not indicate the dataset's general toughness but rather the incompatibility between the model and the data.

### B. Data analysis to measure Difficulty

Static methods of difficulty estimation for samples largely use a framework of functions that evaluate different features of the data like size, resolution, contrast, etc. They do not require training machine learning architectures, which is more efficient.

Some methods rely on quantifiable metrics for analyzing the structure and properties of the data samples. They apply a framework of several statistics to measure different properties of the data samples. These properties can be measured more in detail than training and validating a proxy model. [19] tested several frameworks of measurements to determine various features of an X-ray image such as clutter, view-difficulty, etc. The results shows similarity between the rankings derived from these metrics and human testers. In [20], the authors explored a formal definition for sample difficulty and tested several metrics that determine how hard is a sample to learn. Several of these works display how specific measures can be highly effective in human-in-the-loop applications where they can be used to point out challenging examples. However, they require several samples from the target dataset, which is unavailable before developing the model. An additional problem for applying these methods is that they are not universal for different tasks. They can make comparisons to samples within the same class or among samples that are for the same task. However, they cannot generalize beyond the task boundaries.

While the methods above are not suited for our purpose of building a large archive of datasets and their difficulty, they do show how difficulty is commonly represented.

## III. Problem Formulation

Our work is focused on generating a dataset and model for identifying the necessary data resources for a machine learning problem. The features of interest in this problem is the size of a dataset sample and its difficulty. We consider the best recorded performance for a problem and a dataset as a proxy for the difficulty of that data. For a minimum $\mathcal{A}$ and a maximum $\mathcal{B}$ performance within a task, we calculate the difficulty $\mathcal{D}$ measure for a performance $\mathcal{E}$ as:

$$\mathcal{D} = 1 - \frac{\mathcal{E} - \mathcal{A}}{\mathcal{B} - \mathcal{A}}$$

The values of difficulty is between 0 and 1 with most difficult datasets having a value of 1.

Based on these values, we define our problem in the following. Let $\mathcal{Z} = \{\{z_1, z_2, \dots\} | z_n = (x_n, y_n, z_n, s_n)\},\ x_n \in \mathcal{X},\ y_n \in \mathcal{Y}\ z_n \in \mathcal{Z}\ s_n \in \mathcal{S}$ be the space of labeled datasets where $\mathcal{X}$ is the task space, $\mathcal{Y}$ is the task performance space, $\mathcal{D}$ is the task difficulty space and $\mathcal{S}$ is the dataset size space. We are given a task $X_0 \in \mathcal{X}$, a desired performance $Y_0 \in \mathcal{Y}$, and

a difficulty estimation metric $Z_0 \in \mathcal{Z}$. Find $\min S_{X \sim Z}[Z, Y]$, which is the minimum number of samples required to achieve performance $Y_0$ for task $X_0$ at difficulty $Z_0$.

## IV. An Approach to Estimating Data Requirements using Metadata

In this section, we overview our approach to estimating data requirements for learning-enabled systems using metadata, as illustrated in Figure 2. In the following, we first motivate the approach, then discuss the metadata features and our model design approach.

### A. Motivation

Our work centers on developing an architecture to estimate the sample size needed for any task. While there are individual works that provide estimates for a specific task, the generalizability of our model requires the use of a machine learning framework. Currently, no dataset exists that captures the metadata values we are interested in such as dataset, task, etc. This necessitates the compilation of a new dataset that gathers information on the dataset, the task and the properties of that dataset.

From our observation, more challenging datasets usually have more samples to facilitate better performance from models. Datasets have many of their characteristics reported such as size, content, data type, etc. However, there is no values that directly quantify its level of challenge. In Section II, we discuss several methods that were previously explored in the subject of data complexity analysis.

One of the key issues in existing works we want to improve on is data requirement. Functions and architectures mentioned in Section II require at least a small number of samples from the target dataset to assess the difficulty or model's ability to perform on it. This is a barrier for their usability since it can be hard to procure a sample due to the experiments and approvals needed. Therefore, we abstract the sample analysis with a difficulty metric that summarizes our ability to complete a task on the sample - Maximal Performance Index (MPI).

We formalize the requirements of a candidate metadataset as:

1) task indication: reports on the "goal" or "theme" of the dataset and what question it is posing
2) performance of models: shows the range of potential performance on a dataset given its task
3) dataset properties: represents the descriptive values of the dataset, contextualizes it among other datasets of the same task

### B. MetaData Features

In our approach, we utilize a total of 3 features, listed below:

- performance
- task identifier
- dataset-task associated MPI

Our choice of these three features is heavily influenced by the problem formulation and feasibility.

From analysis, it seems clear that the properties of a dataset such as difficulty or complexity is heavily influenced by the task it poses. After all, data is usually heavily engineered to present a pointed challenge for models. Samples for object locating and image classification are very different and should not be judged by the same scale. Therefore, we supply a task identifier token to differentiate between them.

The performance token shows the values that architectures can achieve with the same dataset and task. Due to the advancement in machine learning, there is a big improvement in how models would perform. The desired performance input by the user would correspond to some values along this range.

Within our dataset, we chose MPI as our dataset property to report. Since it is based on the highest performance available, it is very simple to calculate from the available performance tokens. Additionally, we also believe it represents the general difficulty level of a dataset in comparison to our technical abilities. However, we acknowledge that MPI alone might not encompass every interesting detail on a dataset. The dataset properties token could be extended to multiple other feature based on usage. Currently, this is beyond the scope of our work.

The difficulty of a dataset-task pair is assessed based on the highest performance recorded. This value is then normalized relative to all datasets for that specific task. The resulting data has a task identifier, a performance value related to an anonymous model, and the difficulty associated with the dataset-task pair per sample. From these features, the model can determine the task association, and the expected difficulty range given a performance value.

The raw information collected is the task, dataset, models' name, performance, and other related details. Since the performance of models is measured with different metrics, and on different scales, the value is normalized based on the task group's performance information. Since our model is learning the correlation between dataset difficulties, desired performance, and number of samples, it is important to have a wide range of datasets with varying levels of challenge and size. When expanding this dataset, it would be important to place emphasis on the number of datasets represented per task.

### C. Model Design

Our model structure can be split into 2 stages (Figure 2):

- input preprocessing: takes the input from the user and normalize to within the range of the existing samples in the dataset
- difficulty output: can be reverse map to the size of the datasets in the range

The user is asked to provide their task and the desired performance. In our dataset, the performance values are normalized with a min-max normalization scheme given the task. When the user input the task token, and their desired performance, these values are incorporated into the spectrum of values previously established. The task information is represented as a one-hot encoding before being sent to the model. This normalization process ensures that the inputs are

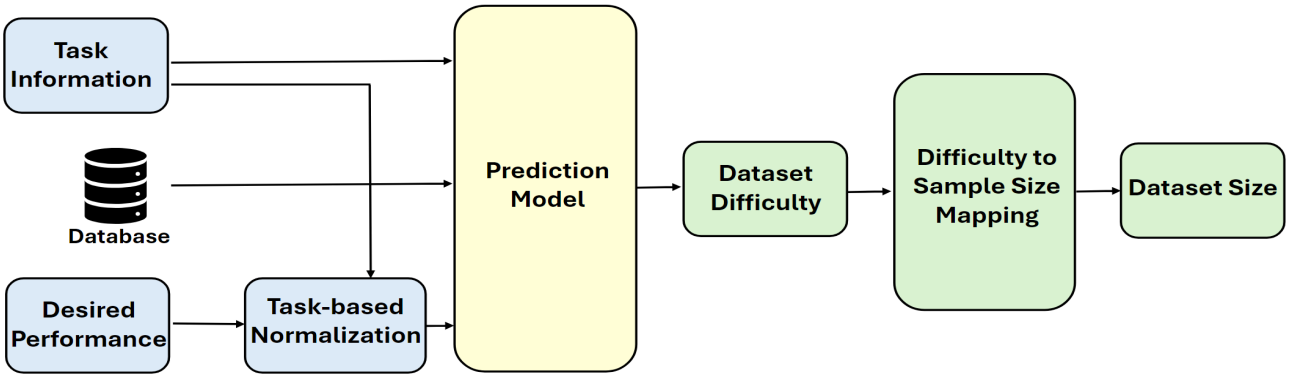


Fig. 2: **Architecture of the Prediction Model.**

standardized, facilitating more accurate predictions. Subsequently, these normalized values, along with the estimated difficulty provided by the user, are fed into the prediction model.

The model's output is the estimated difficulty of the target dataset, which can be reverse mapped to the number of samples. The difficulty mapping component utilizes information from our existing dataset pool to assess both the required dataset size and the quality of the input features. This mapping can be fine-tuned if the user has specific knowledge about the size and complexity of their data, allowing for more tailored predictions.

## V. ImageMeta: A Metadataset for Image Datasets and Tasks

In this section we introduce ImageMeta, as the first open-source compiled dataset containing metadata for image datasets and tasks. While there were previous work that compiled metadatasets, they do not report on properties such as challenge level or difficulty level of the dataset. Other works that focus on assessing the samples for its difficulty are very computationally complex, and subsequently insufficient in number. Our dataset is the first work that both reports on metadata and quality properties of sufficient size. ImageData contains over $1,000$ pairings of task and dataset for around $17,000$ individual samples. These categories encompass most of the computer vision field, including tasks ranging from semantic segmentation to image classification. A complete distribution of tasks and datasets is shown in Figure 3. While ImageMeta focuses on images due to their wide availability and prevalence in computer vision research, our future work involves expanding ImageMeta to include other dataset types (*e.g.,* time-series). In this work, aimed at establishing the feasibility of estimating data requirements for learning-enabled systems using meta data, we limit our focus (temporarily) to images. In the following, we present our methods for compiling meta data and executing data processing in the development of ImageMeta.

### A. Data Compilation

ImageMeta is compiled from an online resource [21] which reports on the dataset, the task and the performance of various machine learning models to that task and dataset pair. The values provided includes tasks, datasets, methods, papers, results, etc. This is a comprehensive source for metadata for models and datasets. It includes the values of work that has been published and those that were not. Within the scope of our experiment, we only included computer vision tasks, but the process can be replicated with any machine learning task. For the purpose of this work, which is to provide proof-of-possibility, we find consideration of computer vision tasks only sufficient. Additionally, selecting tasks from only one field, further implies similarity among the tasks and datasets, which simplifies the correlation the model needs to learn.

Since it is a complete and up to date resource, we believe the performance in this database represent the highest level of our current technical abilities. The maximum performance is important because it is what we rely on to estimate the difficulty of a task or dataset. The assumption is if a dataset and task has near perfect performance, it is easier to accomplish than one with a lower values.

Although the data pulled is abundant, and mostly well-recorded, there are issues of consistency in metrics used. From the collected data, we eliminate entries that are noisy or fail to report performance with the widely accepted metric for the task. Even though we are only considering task within one field, there is a wide variety of measurements that exist in different ranges: 0 to 100, 0 to 1, etc. To provide uniformity and ease the comparison between them, the values are normalized based on a min-max normalization scheme. The difficulty of a dataset-task pair is assessed based on the highest performance recorded. This value is then normalized relative to all datasets for that specific task. The ordering of difficulty of the datasets in each task is then established. The resulting data has a task token, a performance value, and the difficulty associated with the dataset-task pair per sample.

### B. Data pre-processing

One of the biggest issues when gathering this dataset is filtering for values of interest. We generated a short list of tasks that we would like to be represented within our dataset. The complete database is then scraped for only datasets that exist within the tasks we are interested in. The complete task list is shown in Figure 3.

The data scraped contains many fields, most of which were not of direct relation to our features. We selected the features relevant by text analysis. Commonly, among the multiple available performance metrics, we select the first field since it is often the most well populated with results. The remaining values sometimes contain noise or are out of range. We further filter the samples for only valid performance reports. These values are then placed within task and dataset groupings, where they are normalized. Taking the maximum performance within these groupings, that value now represents the least difficult dataset within that task. The opposite is true for the minimum normalized performance. The maximum performance of the task-dataset groupings is the MPI difficulty token generated for that dataset and task pair.

The highest recorded performance, MPI, is how we calculated task difficulty. Our assumption is within one task if a dataset has a higher performance measured, then its difficulty is lower. This is another reason why having many datasets for each task is important, to make more meaningful comparisons.

## VI. Experimentation

### A. Experimental Setup and Comparisons

In our experiments, we evaluated three different models: a linear regression model, a support vector regression model, and a sequential neural network model. The outcomes of these experiments are detailed in Table I.

Each model was tested using the complete dataset, employing task information and the best recorded performance as the difficulty metric. In addition to the data group that employ MPI as its difficulty metric, we also calculated a subset that uses PVI [16]. This is a metric based on how much feature related information is present in a gold label sample versus a noise sample. We used the CLIP architecture for our analysis since it has good results at a variety of data samples and tasks, showcasing good generalizability [22]. The subset contains 10 datasets within the Image Classification task, where PVI directly replaces MPI. The subset for PVI is much smaller than that of MPI is due to data scarcity. Since MPI does not require specific data samples, we can simply gather the performance without needing to source datasets that are often no longer available or only have limited availability. In contrast, data dependent methods require sourcing the data, and setting up a pipeline to process it. This introduces two issues: many datasets are not publicly available or they might be corrupted due to time and lack of maintenance, and setting up these pipelines for analysis requires time and expertise. Both of these problems become much more emphasized when you are trying to gather enough data points to complete tasks in machine learning.

From Figure 4, there is not a high rank correlation between the rankings generated by usable information and by the performance based difficulty. This may reflect that CLIP does not work optimally on all datasets, which is expected since it is not fine-tuned. The usable-$\mathcal{V}$ and PVI can reflect how difficult each sample are for CLIP but not for any model available.

### B. Experimental Results

The main goal of our work is to provide a proof-of-concept to estimating data requirements prior to developing models. In this section, we show three simple architectures that are capable of achieving good results estimating the properties of the target dataset when tested on ImageMeta.

We experimented with a neural network, a linear regression model and an support vector machine machine. Linear models seem most appropriate due to the limited number of features in this dataset. The three architectures have comparable results (Table I), with the linear models being much more efficient to train. When using the alternative difficulty metric compared to MPI, the two linear models performing significantly better than the neural network (Table Ib). The result indicate that linear models are capable of graphing the trend observed in the data more efficiently. When maximum performance was used as the difficulty metric, a strong correlation emerged between difficulty and the reported performance of the models, resulting in satisfactory performance across all three approaches.

In the comparison of PVI versus MPI as the difficulty metric, models have much better results when employed on the MPI system. In Table I, models across the board have much lower errors on the complete data split versus one using only PVI.

While these initial findings are encouraging, we recognize the need to expand this research to encompass a broader range of tasks and sample sizes. However, this shows that the concept and application of MPI can work on a large scale. The models are currently capable of determining the difficulty of the datasets with significant accuracy. With more enhancement, this method could alleviate the strain of repetitive data collection and reduce needed time when transferring machine learning technology to new fields. While a complete discussion of these results is contained in Section VII, these results demonstrate that data requirement estimation using metadata is feasible.

## VII. Discussion

One of the main goals of our work is to introduce the metadata estimation problem as a valuable pre-processing step in machine learning research. This is an initial attempt at solving this problem. In this section, we discuss the results, benefits and drawbacks to our methods, including our data gathering process, and estimation metrics.

Based on the results reported above, it is determined that the sample size needed for a task can be estimated before beginning the research process. By further refining the model design outlined above, along with more comprehensive data compilation methods, data requirements can be practically

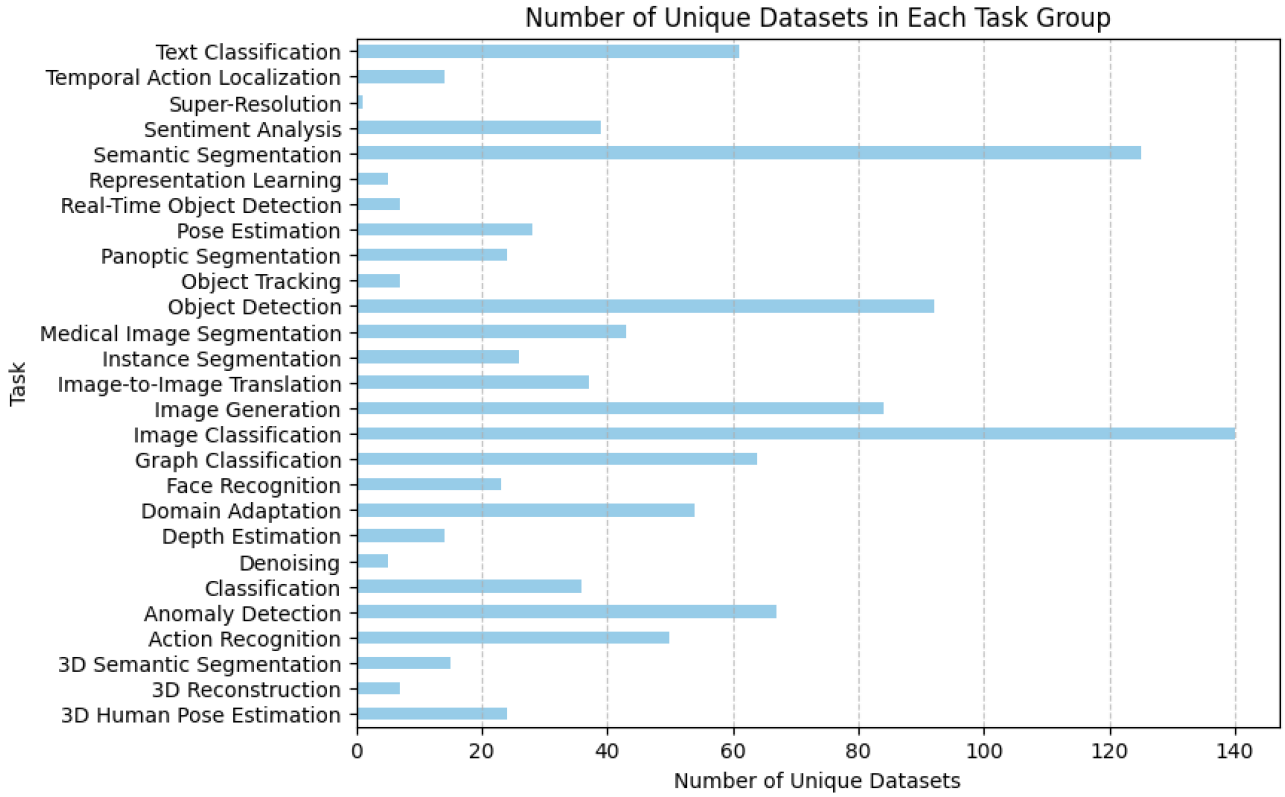


Fig. 3: **Number of Unique Datasets in Each Task Group.** The total number of combinations is 1092 task and dataset pairings. There are a total of over 17000 samples.

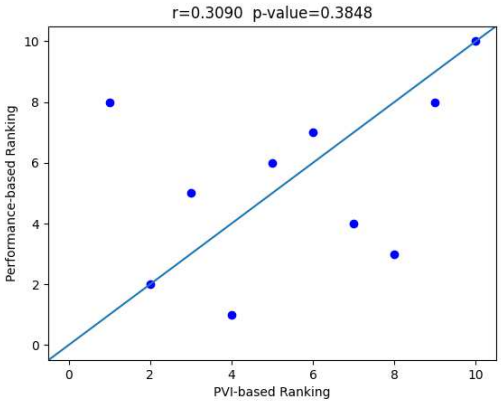


Fig. 4: **Rank Correlation between PVI and Performance Based Difficulty Scoring.** The Spearman's Rho Correlation value and its significance are reported on top of the figure.

estimated for a number of problems, reducing the ambiguity of the process while increasing usability for machine learning methods in new applications.

| Model | MSE | MAE |
|---|---|---|
| Neural Network | 0.0356 | 0.1180 |
| Linear Regression | 0.0365 | 0.1173 |
| SVR | 0.0394 | 0.1218 |

Table 1a: Comparison of Model Performance on the Full Data with Performance-based Difficulty Metric

| Model | MSE | MAE |
|---|---|---|
| Neural Network | 46.5381 | 6.2757 |
| Linear Regression | 8.1796 | 1.5952 |
| SVR | 11.8389 | 2.7748 |

Table 1b: Comparison of Model Performance on the Partial Data with PVI-based Difficulty Metric

TABLE I: Comparison of Models on Two Splits of Data

### *A. Effect of Noise Values on Data Processing and Normalization*

While the performance of a dataset is broadly reported in every paper that utilizes it, there is disparity among the records. These variations come from the metrics used for reporting performance, or datasets being used.

PapersWithCode is a self-reporting repository where we gathered our samples. Most of the values reported is consistent and based in real results. However, it is unavoidable that a small number of values are not true. These values can add

noise to our normalizing range and cause some changes in our data. However, since we employ a min-max normalization scheme, as long as the lowest and highest values are accurate, the scale is not influenced by the noise.

Since there is a big difference in scale and information encoding, the values are normalized to the 0 to 1 scale. This normalization happens twice within the task and dataset level. Within each task, the highest performance for each dataset is compiled to rank the difficulties of the dataset within the task. The performance within each dataset is also normalized with the best being 1.0 and the worst being 0.0. This second normalization layer enables easier comparison once ground on a difficulty level and a task. Currently, the normalization function used is min-max. There is also additional concerns over if the reported minimum or maximum values are not up to date or incorrect. While this could alter the range of values, it does not change the ordering of the datasets by much. In these cases, we can still rely on the overall ranking for the datasets. In the future, we would like to experiment with other normalization methods that could represent the relationship between values more accurately and are more resistant to noise.

There are multiple metrics that can be used to compare performance within one task. This leads to discrepancies in our comparison and difficulty in processing the values to a normalized range. The other source of disparity is the measurement used to qualify performance in different tasks. While in classification tasks, accuracy is often used, mean squared errors or other specialized metrics can be more common for other tasks. The needed specialist knowledge to decipher these values makes it more difficult to gather a lot of samples effectively. For the tasks we selected for our dataset, this was not a significant issue. More work will be needed to expand this dataset to more tasks and more fields aside from computer vision, where the performance metric is more ambiguous.

### *B. Maximal Performance as a Difficulty Metric*

As mentioned above, Maximal Performance Index (MPI) is the value chosen for our difficulty metric. The current best performance across different tasks is represented in our resource [21]. The values are reported as per dataset and task pair. Based on this, we can compare the reported performance within the same task among different datasets. The dataset with the lowest performance is most difficult and vice versa. Given a choice of any architecture, this value represents the best performance you could achieve.

Using MPI also has the additional benefit of being efficient and simple to calculate. While other values require training models or engineering functions to calculate, we can simply scrape and process this information. As a comparison, even with an efficient framework, using a dynamic method for the difficulty score required days to sample several datasets in the same task, whereas MPI takes only a few hours to scrape and process. Due to this efficiency, we were able to build a much larger and more comprehensive dataset.

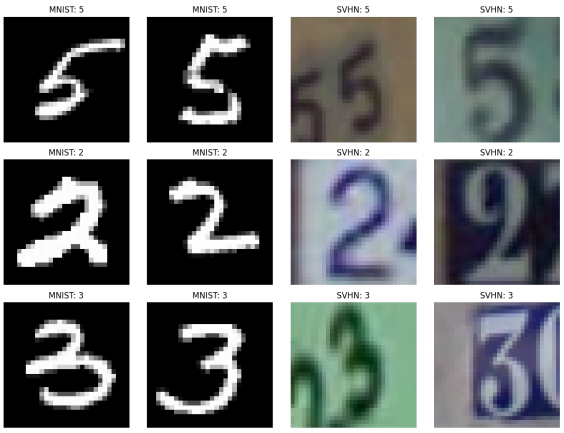


Fig. 5: **Comparison of Difficulty of Samples from MNIST and SVHN** SVHN (right) have a higher perceptual difficulty compared to MNIST (left)

Many existing metrics compare various aspects of diverse types of data. For images, there are FID, CLIP scores, measurements for the quality and noise of the images, etc. For text data, there are Flesch-Kincaid scores, token counts, etc. These are quantifiable ways to measure how complex a sample is based on different details. In the beginning, we experimented with these methods to add more context to the complexities of the data samples within our domain. While insightful, they do not necessarily correlate to the difficulty level of a dataset. They are also inefficient to calculate for each sample.

As we continue to expand the problem statement and assess the usage of the resulting model, there is a need for different tasks and different fields to be represented within our sample. The variety in data makes the use of specific metrics more improbable due to the differing requirements of different data types. Firstly, the functions available can make comparisons within the same domain, but they are not easily transferable across different domains. Adding them to our collection of features introduces inconsistencies that could negatively affect performance and decrease general clarity. On the other hand, we found that asking for samples to make these comparisons and generate the metrics is difficult and unfeasible at inference time due to the number of samples needed. For these reasons, we decided to use the performance already recorded of the models, normalized to within the range of 0 to 1 as our difficulty metric. The main issue with this approach is that it relies on expert knowledge from the user. They must be able to gauge what their potential data would be like and how it compares to the datasets within our pool. We find this method to be efficient and sufficient for comparing the overall difficulty. However, it lacks the analytical power that the other methods have, since it does not directly assess the data samples. We suggest using this process to gather general information for data gathering, in combination with another evaluation method to derive structural information about the data after collection.

We also included a subsection of our data with a sample-based difficulty measurement. For these, we calculate the PVI score for a small subset of the full dataset, which represents the amount of features extracted from the data versus a complete noise sample.

As mentioned in section II, usable-$\mathcal{V}$ generates a value of usable information from each sample by comparing the probability of the accurate prediction versus when using a noise sample. While in [16], they train an architecture on the training set and calculate the values on the test set. Due to time constraints, we opted to use a good one-shot model [22]. The choice of model is based on versatility and ability to generalize to multiple tasks and dataset without needing retraining.

Usable information is a metric previously used in natural language processing [16] which compares how much information is usable within a labeled sample when compared to a noise sample. This method is theoretically sound and offers a broad analysis of the difficulty levels within the data itself. Using CLIP for our predictions reduced the time needed to calculate the usable information value since it did not require retraining on the labeled images. Since CLIP is used on every one of our datasets, it is a grounding factor for the difficulty of the samples. The resulting values can be ranked to determine their ordered difficulty.

Based on the test results in table Ib, using PVI as the difficulty calculation resulted in much lower performance for all three models.

For calculating PVI, downloading and running tests for each dataset is significantly time consuming. Extracting the index and class mappings of the datasets is not a standardized process and requires manual work. Therefore, it is not efficient for a large number of datasets. In the future, if there are more standardized pipelines, we would like to revisit this method for estimating difficulty.

### C. Usage and Improvements for the Estimation Architectures

Following our discussion on the suitability and efficacy of our dataset difficulty metric, MPI, we would like to explore the functionality and usage of our models.

The models that we have discussed above are clear and transparent in their formulation but capable of estimating the difficulty of the target dataset with low MSE. With simplistic architectures, the performance on ImageMeta is not insignificant. These results show promise in the development of future methods that are geared towards solving the data estimation problem. However, we acknowledge that this architecture requires further refinement to increase the robustness and representation of data sizes. Currently, our mapping from performance to difficulty relies heavily on min-max normalization. This method is simple and easy to implement, which allows us to process and generalize the high volume of data gathered. With further work, we can exchange this normalization method for a more robust and expressive method, which will increase the descriptive power of the rankings.

On the other end of the model architecture, the difficulty token is reversed map to gain the sample size. The mapping associates a difficulty value with a dataset size existing within our dataset. Again, the simplicity of this process enables transparency and clarity. However, there definitely could be future work on a continuous mapping between MPI and sample size. This addition would enable more flexible and robust estimations.

## VIII. Conclusion and Future Work

In this paper, we introduced a novel dataset that addresses general dataset difficulty and its correlation to the performance and size of the dataset. To our knowledge, this is the first attempt at generalized difficulty estimation across architecture and task lines. We proposed three simple architectures that predict difficulty from a task and specified accuracy, which shows promising results in estimating data requirements.

Currently, our dataset only includes tasks within the vision domains. This simplifies the problem because there are implied similarities between the tasks. The difficulty then only relies on the quality of the dataset. However, as we add more samples from other domains, that similarity becomes more ambiguous. In the future, we would like to explore combining our difficulty metric for each dataset with one that compares the similarity or relative difficulty of a task. Previous work has been done on this topic [23] that is of particular interest.

Based on the work presented above, there are two main points that we would like to build on: extending the tasks to include a wider variety of machine learning tasks, and incorporating the comparison of relative similarity and difficulty between tasks.

## References


[1] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[2] R. Rai, M. K. Tiwari, D. Ivanov, and A. Dolgui, "Machine learning in manufacturing and industry 4.0 applications," pp. 4773–4778, 2021.

[3] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and iot in smart transportation," *Future Internet*, vol. 11, no. 4, p. 94, 2019.

[4] R. C. Deo, "Machine learning in medicine," *Circulation*, vol. 132, no. 20, pp. 1920–1930, 2015.

[5] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[6] S. R. Messé, S. E. Kasner, B. L. Cucchiara, M. L. McGarvey, S. Cummings, M. A. Acker, N. Desai, P. Atluri, G. J. Wang, B. M. Jackson *et al.*, "Derivation and validation of an algorithm to detect stroke using arm accelerometry data," *Journal of the American Heart Association*, vol. 12, no. 3, p. e028819, 2023.

[7] K. K. Trout, S. Modri, H. M. Sehdev, and J. Weimer, "69 derivation and validation of an intrapartum algorithm to predict postpartum hemorrhage risk using photoplethysmography data," *American Journal of Obstetrics & Gynecology*, vol. 230, no. 1, p. S52, 2024.

[8] T. Elomaa and M. Kääriäinen, "Progressive rademacher sampling," in *AAAI/IAAI*, 2002, pp. 140–145.

[9] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 289–300, 2002.

[10] R. T. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, "How hard can it be? estimating the difficulty of visual search in an image," 2017.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
[12] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.
[13] M. Frye, J. Mohren, and R. H. Schmitt, "Benchmarking of data preprocessing methods for machine learning-applications in production," *Procedia CIRP*, vol. 104, pp. 50–55, 2021.
[14] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.
[15] D. Mayo, J. Cummings, X. Lin, D. Gutfreund, B. Katz, and A. Barbu, "How hard are computer vision datasets? calibrating dataset difficulty to viewing time," *Advances in Neural Information Processing Systems*, vol. 36, pp. 11 008–11 036, 2023.
[16] K. Ethayarajh, Y. Choi, and S. Swayamdipta, "Understanding dataset difficulty with $\mathcal{V}$-usable information," 2022.
[17] F. Scheidegger, R. Istrate, G. Mariani, L. Benini, C. Bekas, and C. Malossi, "Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy," *The Visual Computer*, vol. 37, no. 6, pp. 1593–1610, 2021.
[18] C. Agarwal, D. D'souza, and S. Hooker, "Estimating example difficulty using variance of gradients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 368–10 378.
[19] A. Schwaninger, S. Michel, and A. Bolfing, "A statistical approach for image difficulty estimation in x-ray screening using image measurements," in *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, 2007, pp. 123–130.
[20] W. Zhu, O. Wu, F. Su, and Y. Deng, "Exploring the learning difficulty of data theory and measure," 2022. [Online]. Available: https://arxiv.org/abs/2205.07427
[21] PapersWithCode, "Papers with code datasets," https://github.com/paperswithcode/paperswithcode-data, 2021.
[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020
[23] X. Liu, Y. Bai, Y. Lu, A. Soltoggio, and S. Kolouri, "Wasserstein task embedding for measuring task similarities," 2022.