Towards an Efficient ML System: Unveiling a Trade-off between Task Accuracy and Engineering Efficiency in a Large-scale Car Sharing Platform

Kyung Ho Park*
kp@socar.kr
SOCAR AI Research
Seoul, Republic of Korea

Hyunhee Chung*
esther@socar.kr
SOCAR AI Research
Seoul, Republic of Korea

Soonwoo Kwon
tigger@socar.kr
SOCAR AI Research
Seoul, Republic of Korea

ABSTRACT

Upon the significant performance of the supervised deep neural networks, conventional procedures of developing ML system are task-centric, which aims to maximize the task accuracy. However, we scrutinized this task-centric ML system lacks in engineering efficiency when the ML practitioners solve multiple tasks in their domain. To resolve this problem, we propose an efficiency-centric ML system that concatenates numerous datasets, classifiers, out-of-distribution detectors, and prediction tables existing in the practitioners’ domain into a single ML pipeline. Under various image recognition tasks in the real world car-sharing platform, our study illustrates how we established the proposed system and lessons learned from this journey as follows. First, the proposed ML system accomplishes supreme engineering efficiency while achieving a competitive task accuracy. Moreover, compared to the task-centric paradigm, we discovered that the efficiency-centric ML system yields satisfactory prediction results on multi-labelable samples, which frequently exist in the real world. We analyze these benefits derived from the representation power, which learned broader label spaces from the concatenated dataset. Last but not least, our study elaborated how we deployed this efficiency-centric ML system is deployed in the real world live cloud environment. Based on the proposed analogies, we highly expect that ML practitioners can utilize our study to elevate engineering efficiency in their domain.

CCS CONCEPTS

• Computing methodologies
  → Artificial Intelligence.

KEYWORDS

Task-centric ML system, Efficiency-centric ML system, Representation Learning, Model Calibration, Multi-task Learning

ACM Reference Format:

Kyung Ho Park*, Hyunhee Chung*, and Soonwoo Kwon. 2019. Towards an Efficient ML System: Unveiling a Trade-off between Task Accuracy and Engineering Efficiency in a Large-scale Car Sharing Platform. In Proceedings of DLP-KDD 2022. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent progress of machine learning (ML) and convolutional neural networks (CNN) has empowered a great success in various real-world image recognition tasks [16]. When machine learning practitioners desire to solve a classification task in the real world, as shown in Figure. 1. (a), conventional procedures of establishing an ML system are as follows. First, the practitioners build a labeling scheme and acquire a large amount of finely-labeled dataset. Second, they search for a state-of-the-art model fit to their task, implement the model, and train the classifier with the acquired dataset. Third, the practitioners build an out-of-distribution (OOD) detector if the model’s performance is sufficient to be deployed. As the samples irrelevant to the task are often provided to the deployed classifier, the practitioners utilize the OOD detector to filter them out. Fourth, they deploy the trained classifier and OOD detector at the server. Lastly, they build an ML pipeline that sequentially performs the following steps. A system retrieves task-related data from the database with a query script, morph them as an inference dataset (which are candidate samples to the inference), filter out irrelevant samples with the OOD detector, provide the filtered inference data to the trained classifier, and accumulate the inference results to a prediction result table. When the pipeline is established, end-users (i.e., business operators, data Analysts) use the prediction result tables. When the ML practitioners encounter another task, they repeat the aforementioned procedures to solve a new task. In our study, we denote these procedures as a task-centric approach as it focuses on solving an individual task with precise task performance.

However, the task-centric approach bears several inefficiencies (denoted as engineering inefficiency in our study) when the ML practitioners encounter multiple tasks. In real-world circumstances, ML practitioners, especially in the car sharing platform, usually encounter a particular class that is included in multiple tasks. Suppose the ML practitioners solve two classification tasks of label 0 v. label 1 and label 0 v. label 2. Following the aforementioned task-centric approach on two tasks, we analyzed there exists four primary inefficient points. First, in the perspective of model training, they should train the model twice for each task while the two tasks simultaneously utilize the label 0 class. At this point, we postulate a research question: "What if we concatenate two datasets (label 0 v. label 1 and label 0 v. label 2) as a single one (label 0 v. label 1 v. label 2)? If a model trained with the concatenated dataset does
not experience harsh performance degradation, can’t we substitute two binary classifiers with a single 3-class classifier? Second, there also exists room for improvement at the OOD detector. Establishing an effective OOD detector in the real world is challenging as the OOD detector cannot learn the irrelevant samples a priori [6, 8]. Upon the difficulty of OOD detection, we posit another question: “If the concatenated dataset exists, would the OOD detector based on the concatenated dataset reject irrelevant samples better?” As the concatenated dataset let the model to learn wider knowledge compared to the task-centric approach, we presume the OOD detector based on the concatenated dataset can improve the OOD detection performance. Third, in the perspective of deployment, task-centric approach wastes a computing resource in a particular manner. Referring to two classifiers of label 0 v. label 1 and label 0 v. label 2, the label 0 samples are included in a inference batch twice at each task, although the two trained classifiers would yield the same prediction of label 0. Lastly, there also exists inconvenience for end-users when a sample is multi-labelable. Suppose a sample has attributes of both label 1 and label 2. The label 0 v. label 1 classifier would yield a prediction of label 1, and the label 0 v. label 2 classifier would yield a prediction of label 2. As these results are accumulated in a separated prediction result tables, the end-users have to merge these two tables when they use these results in business operations. At this point, we also postulate the last question: “What if a single classifier trained on the concatenated dataset yields predictions of label 1 and label 2 simultaneously? If so, can’t we just accumulate top-2 predictions at a single prediction result table and let the end-users use it?”

To resolve the proposed inefficiencies of the task-centric approach, MLOps [1, 17] paradigm utilizes various engineering tools to ease the task management with dataset versioning, model versioning, and fast model deployment. In other viewpoint, our study proposes a novel ML system paradigm denoted as efficiency-centric approach, which resolves the inefficiencies of task-centric approach by concatenating every task-centric dataset, model, OOD detector, and prediction result table into a single one. Based on two real-world image recognition tasks at SOCAR, the largest car-sharing platform in the Republic of Korea (it does operations similar to ZipCar in the United States), we introduced a series of analyses illustrating how we substituted a priori task-centric ML system into the efficiency-centric ML system. To examine the effectiveness of efficiency-centric paradigm, we postulate three research questions: 1) Does the concatenated classifier solve each task better? 2) Does the concatenated OOD detector rejects irrelevant samples better? 3) Does the concatenated classifier calibrate well enough to provide useful prediction results on a multi-labelable sample? Throughout the study, we provide answers to these research questions and describe how we deployed the concatenated ML pipeline in the real world. We expect the proposed lessons learned will be a solid benchmark for candidate ML practitioners who desire to reduce inefficiencies at task-centric paradigm. We acknowledge that this work is an applicable study in a large-scale car-sharing platform domain, and it aims to provide practical guidelines to machine learning practitioners who face similar challenges with us.

Throughout the study, key contributions are as follows:

- We discovered the concatenated classifier accomplished a promising image recognition performance compared to the task-centric classifiers. However, as the concatenated classifier does not always outperform the task-centric approach, we recommend that ML practitioners cautiously adapt the efficiency-centric ML system when the task accuracy is not absolutely important in their domain.
- We figured out the concatenated OOD detector significantly outperformed the prior task-centric paradigm. We examined the efficiency-centric paradigm’s key strength is an overwhelming OOD detection performance compared to the task-centric approaches; thus, we highly recommend the ML practitioners utilize our efficiency-centric paradigm when OOD samples frequently exist.
- We experimentally validated the concatenated classifier is calibrated well to provide sufficient prediction results to the multi-labelable samples; therefore, we justified a efficiency-centric paradigm can reduce both the wasted computation resource consumption and the inconvenience of end-users.
- We illustrated how the efficiency-centric paradigm is deployed in real-world live service at the large-scale car sharing platform. We expect the proposed deployment architecture will be a solid benchmark for candidate ML practitioners to apply in their domains.

2 PRELIMINARY

2.1 Background

The core business of the car-sharing platform is enabling users to borrow a car with a smartphone application. When the users want to borrow a car, they can easily make a reservation on the smartphone application. Then, they go to the nearest parking station, open the car with a smartphone application, and start their trip. After using the car, the users park the car at the parking station where they borrowed it, and lock it on the smartphone to finish their use. During the aforementioned journey, SOCAR enforces the users to take pictures of the car and send them to the company through the application on particular events: before and after using the car, accident, car wash, charging washer fluids, and so on. To monitor the car’s state without the human inspector’s physical visit, human inspectors in SOCAR utilize these images to inspect the car’s states. In SOCAR, there exist three primary car states that human inspectors are concerned with: Dirt, Defect, and Normal. The Dirt illustrates a state where the car’s surface is dirty. The human inspectors are concerned with Dirt state as dirty cars severely drop down the user experience and cause harsh claims. When the human inspectors recognize the car is in Dirt state, they wash the car right away. The Defect describes a car state where defects (i.e., scratch, dent) exist on the car’s surface. The human inspectors are also concerned with Defect as the damaged cars might cause the user’s safety problems during the trip; thus, the inspectors send the car with any defects to the auto repair shop when they find it. The Normal state implies that the car does not have any dirt or defects on its surface.
Towards an Efficient ML System: Unveiling a Trade-off between Task Accuracy and Engineering Efficiency in a Large-scale Car Sharing Platform

DLP-KDD 2022, August 2022, Washington, DC, USA

(a) Task-Centric Approach

(b) Efficiency-centric Approach

Figure 1: A comparison between the prior Task-centric ML system and the proposed Efficiency-centric ML system. The task-centric ML system requires the practitioners to establish data-retrieving query scripts, OOD detector, classifier, and prediction result tables on each task. The end-user should refer to several independent tables to scrutinize a multi-labelable sample. On the other hand, our efficiency-centric ML system resolves the aforementioned limits by concatenating the aforementioned components in the ML pipeline. The end-users under the efficiency-centric paradigm can easily inspect multi-labelable samples in a single prediction table.

2.2 Prior Image Recognition System

In this section, we aim to describe how the SOCAR has operated image recognition models before the introduction of efficiency-centric paradigm. Along with the three car states mentioned above, there were two image classification systems: car dirt recognition and car defect recognition. The car dirt recognition model classifies between Normal and Dirt given a single image while the car defect-recognition model identifies between Normal and Defect. Following the task-centric paradigm, we have acquired labeled datasets for each task, trained the models, established the OOD detectors, deployed them, and accumulated the prediction results in separated tables. We would like to highlight that every user-generated car image is accumulated in a single database. As the image recognition system performs a prediction daily, the image recognition system retrieves every image uploaded on a target day and provides them to the OOD detector and the trained classifier in sequential order.

Figure 2: Image Recognition under SOFAR-Image Label space

In this image recognition system, we figured out several improvement avenues as described in section 1. For ease of understanding, we visualized the image recognition tasks in Figure 2. First, the ML practitioners have acquired two datasets for two tasks while they share the Normal label simultaneously, as shown in Figure 2. Second, the prior image recognition system operates two OOD detectors for each task, although both tasks also share the same OOD samples that frequently exist in the database. Third, as both tasks retrieve inference samples from the same database, a single sample is provided into two classification systems while it does not have to be; thus, it causes a waste of computation resources. Lastly, as the prediction results are accumulated in separated tables, the human inspectors (end-users in Figure 1) had to write two query scripts to understand the car’s state. In a nutshell, we figured out the prior task-centric ML systems bear several inefficiencies in the real-world environment. Motivated by this room for improvement, our study proposes a simple but effective approach to the ML system denoted as efficiency-centric paradigm. While multi-task learning shares similar motivation with the proposed paradigm, it differs in the number of classification heads. Our efficiency-centric paradigm utilizes a classifier with a single classification head while multi-task learning embraces multiple heads. Our study presumes that concatenated head is much more advantageous in representation learning than utilizing multiple heads. A unified head under cross-entropy loss lets the model choose the most probable label; thus, it can enforce the model to be more focused on discriminative characteristics among labels.

3 EFFICIENCY-CENTRIC ML SYSTEM

3.1 Description

In this section, we elaborate on how we established efficiency-centric ML system in the car sharing platform domain. The development of efficiency-centric ML system consists of three steps: dataset acquisition, model training, and establishing an OOD detector. The detailed descriptions of each step are illustrated in the following sections.

3.1.1 Dataset Acquisition. First and foremost, we acquired a domain-specific benchmark dataset by concatenating every task-centric dataset. Practically, we investigated every acquired dataset at SOCAR, clarified the labeling scheme, and concatenated the datasets into a single one. For example, given car dirt recognition dataset
We postulated several research questions to examine whether the task-centric ML system accomplishes better OOD detection compared to the task-centric approach.

- RQ 1. Does the efficiency-centric ML system accomplish comparable task accuracy to the task-centric approach?
  - Our study aimed to examine whether the efficiency-centric approach can achieve competitive task accuracy with the task-centric approach.

- RQ 2. Does the efficiency-centric ML system perform OOD detection better rather than task-centric approach?
  - We set experiments to examine whether the proposed efficiency-centric ML system accomplishes better OOD detection compared to the task-centric paradigm.

- RQ 3. Does the Top-n predictions efficiency-centric ML system yield correct labels on a multi-labelable sample? Lastly, RQ 3 justifies the substitutability of our approach on the concatenation of prediction result tables. We presume a single, concatenated prediction result table can substitute multiple task-centric tables if the efficiency-centric classifier is calibrated well enough to predict multi labels on the multi-labelable samples.

### 3.3 Baseline for Comparative Study

#### 3.3.1 Task-Centric ML System

The task-centric ML system is a conventional approach to the image recognition task. Given a labeled dataset regarding the task, (i.e., Normal and Defect at SOFAR-Image to the car defect recognition), the task-centric paradigm trains the classifier under the supervised regime. To establish an OOD detector, we followed the same OOD detection method [7] with the efficiency-centric ML system for a proper comparative study.

#### 3.3.2 Multi-Task Learner

The multi-task learner is an intuitive way to handle multiple tasks and be computationally efficient. We employed this multi-task learner as a baseline of our study as it jointly solves several tasks given a single backbone neural network; thus, it can be a candidate alternative to the task-centric ML system. We design a conventional multi-heads classifier with a CNN backbone and a few heads for each task for comparison with the efficiency-centric ML system. As a classifier, we use four heads; three heads are for binary classification (Defect v. Normal, Dirt v. Normal, Bubble Wash v. Normal), and one head for multi-class classification on the other labels of SOFAR-Image. For training, binary cross-entropy loss with sigmoid function is used for the binary classification heads, and cross-entropy loss with softmax function is used for a multi-class classification head. The training loss is a summation of losses that occurred from each head. In the evaluation phase, we only use the prediction of the target task head. For example, when we desire to investigate defect recognition performance of the multi-task learner, other predictions except the defect task head are ignored.

Unfortunately, the OOD detection method [7] utilized in both task-centric and efficiency-centric ML systems cannot be directly applied to the multi-task learner due to its network architecture. Given an Normal sample to the trained multi-task learner, the output logits of the binary heads would be nearly zero (implying an OOD) even though it distributes within the training set. To resolve this limit, we utilized the Kullback-Leibler (KL) divergence between a uniform distribution and the concatenated logit of the multi-task learner. We presume an OOD sample should not be activated in every head of the multi-task learner. We expect a sample with large KL divergence implies that the model understands particular characteristics of the image; thus, the sample is adjacent to the training set distribution. For the sample with small KL divergence, vice versa. For example, given an OOD sample to the multi-task learner, the distribution of predicted logit shall be similar to the uniform distribution as follows. A logit at each binary head should be nearly 0.5, and the one from the multi-class head shall be 1 / number of classes on the multi-class head.
4 DOES THE EFFICIENCY-CENTRIC ML SYSTEM SOLVE THE TARGET TASK WELL?

4.1 Setup

In pursuit of discovering an answer to the RQ 1, we aim to examine whether the efficiency-centric ML system can accomplish promising task performance compared to the task-centric approach. As there exist more labels in engineering-centric approach, we presumed the proposed ML system is exposed to the higher risk of making a faulty prediction. Suppose that we solve a car defect recognition task (Normal v. Defect). Under the task-centric paradigm, the model would learn a representation power biased to the two labels only. It implies that the model learns discriminative characteristics of Defect label relative to the Normal, and vice versa; thus, the model shall scrutinize only two label spaces at the end of the training. On the other hand, under the efficiency-centric paradigm, the model would acquire the representation power, which can classify more labels (in the SOFAR-Image dataset). This widened label space might contribute to more qualified representation power for the target class (Defect) or disqualify it by adding unnecessary knowledge to the task. To clarify it, we designed experiments to examine how the efficiency-centric ML system accomplishes task performances at two target image recognition tasks: car defect recognition and car dirt recognition.

We trained the proposed efficiency-centric classifier (11-class classifier) and two benchmark classifiers: task-centric classifier and multi-task classifier. We established two evaluation suites throughout the experiments: SOFAR-Image-Test and several external validation datasets. The SOFAR-Image-Test implies a test set at the SOFAR-Image, which shares the same distribution with the training set; thus, we presumed the SOFAR-Image-Test does not bear much dataset shift [5] from the training set. On the other hand, to validate our efficiency-centric ML system’s robustness, we also acquired external validation datasets that have a particular dataset shift from the training set. The SOFAR-Image includes samples of white cars the users took on a sunny day, which implies less noisy patterns from the rain or snow. In contrast, we established external validation datasets consisting of car images taken in various weather conditions. We empirically presume car images affected by various weather conditions bear noisy patterns (i.e., snow or raindrops on the car’s surface); thus, these samples are adequate to evaluate the robustness of our approach. We established four external validation sets for car defect recognition (Ext-Snow, Ext-After Snow, Ext-Rain, Ext-After Rain) and two external validation set for car dirt recognition (Ext-H1, Ext-H2) for car dirt recognition. Note that H1 implies the time horizon from January to June, and the H2 describes the other months. We set these external validation sets following the guide provided by the human inspectors in the car sharing platform. They noted that the defect detection should be robust in various weather conditions while the dirt detection had better be robust in different time horizons. We measured the target Accuracy and F1-score at each task as an evaluation metric. We recorded the mean and the standard deviations of each evaluation metric on three trials. The experiment results are shown in Table 1 and Table 2.

4.2 Experiment Results

Along with the experiment results, we discovered that the proposed efficiency-centric paradigm accomplishing competitive task performance in two image recognition tasks, but there still exists room for improvement in the car dirt recognition task. We analyzed these results occurred because the efficiency-centric classifier acquired more general, knowledgeable representation power rather than other baselines. A prominent characteristic of task-centric classifier is that it can create a representation power particularly biased to the given classes. Referring to car defect recognition task, the task-centric classifier learns the knowledge focused only at Normal and Defect. The multi-task learner also learns knowledge biased to the given classes as the last head performs a binary classification (Defect v. No Defect). Conversely, the proposed efficiency-centric classifier acquires the representation power less-biased to particular classes. As it scrutinizes wider representation space from the SOFAR-Image dataset (including 11 labels), we analyzed our efficiency-centric classifier learned more generalizable representation power compared to the baselines. This comparatively general representation power was more advantageous in solving a car defect recognition but less effective in the car dirt recognition. To paraphrase it, more general representation power can fulfill sufficient task accuracy at each task, but sometimes the biased representation power could be much more advantageous. Note that scrutinizing this relationship between the representation power’s characteristics and task accuracy is an improvement avenue of our study. Nevertheless, we evaluate the proposed efficiency-centric classifier achieved competitive task performances compared to other benchmarks. Practically, human inspectors at SOCAR evaluated our efficiency-centric classifier’s performance on SOFAR-Test as sufficient enough to be applied in the real world; thus, we decided to integrate it.
Table 1: Accuracy on various image recognition performances under several evaluation suites consisting of the SOFAR-Image-Test and series of external validation sets. While our efficiency-centric classifier accomplished superior performance in car defect recognition, it achieved competitive performance rather than baseline approaches.

| Method                      | Image Recognition Tasks | Car Defect Recognition | Car Dirt Recognition |
|-----------------------------|-------------------------|------------------------|---------------------|
|                             | SOFAR-Test              | Ext-Snow               | Ext-Rain             | Ext-H1               | Ext-H2               |
| efficiency-centric (OURS)   | 0.930 ± 0.017           | 0.399 ± 0.014          | 0.614 ± 0.012       | 0.934 ± 0.011       | 0.817 ± 0.016       |
| Multi-Task Learning         | 0.914 ± 0.011           | 0.442 ± 0.007          | 0.548 ± 0.011       | 0.612 ± 0.016       | 0.817 ± 0.016       |
| Task-Centric-PMG            | 0.916 ± 0.023           | 0.474 ± 0.037          | 0.557 ± 0.040       | 0.593 ± 0.040       | 0.640 ± 0.045       |

Table 2: F1 score on various image recognition performances under several evaluation suites consisting of the SOFAR-Image-Test and series of external validation sets. While our efficiency-centric classifier accomplished superior performance in car defect recognition, it achieved competitive performance rather than baseline approaches.

| Method                      | Image Recognition Tasks | Car Defect Recognition | Car Dirt Recognition |
|-----------------------------|-------------------------|------------------------|---------------------|
|                             | SOFAR-Test              | Ext-Snow               | Ext-Rain             | Ext-H1               | Ext-H2               |
| efficiency-centric (OURS)   | 0.530 ± 0.017           | 0.691 ± 0.022          | 0.728 ± 0.016       | 0.739 ± 0.021       | 0.787 ± 0.017       |
| Multi-Task Learning         | 0.918 ± 0.011           | 0.567 ± 0.012          | 0.655 ± 0.015       | 0.712 ± 0.023       | 0.766 ± 0.008       |
| Task-Centric-PMG            | 0.917 ± 0.023           | 0.622 ± 0.072          | 0.671 ± 0.064       | 0.692 ± 0.059       | 0.724 ± 0.058       |

4.3 Recommendations
As a lesson learned from the experiment, we highly recommend candidate ML practitioners to establish efficiency-centric classifiers when their domain does not require perfect task performance. The car sharing platform can comparatively allow the incorrect prediction of the machine learning model as it does not cause excessively fatal damage. Moreover, damage from the faulty predictions is mitigated as the human inspectors check the prediction results before the business operation. Thus, we expect candidate ML practitioners in the domain similar to our situation can utilize the proposed efficiency-centric paradigm as it acquires competitive performance to the prior task-centric paradigm. On the other hand, we recommend ML practitioners in accuracy-prioritized domains (i.e., medical applications) should be cautious in utilizing the proposed paradigm as the degraded task performance might cause serious damage to the end-users.

5 DOES THE EFFICIENCY-CENTRIC CLASSIFIER SOLVE THE OOD DETECTION BETTER?

5.1 Setup
As an answer to RQ 2, our study validated whether the proposed efficiency-centric ML system achieves better OOD detection performance rather than baselines. We expect the efficiency-centric ML system would solve the OOD detection better as its representation is less biased to particular classes; thus, it can understand the characteristics of samples irrelevant to the target task and reject them. To examine the effectiveness of the proposed efficiency-centric ML system, we postulated two OOD detection tasks: One-class recognition and irrelevant sample rejection.

5.1.1 One Class Recognition. First, the one-class recognition aims to validate whether the efficiency-centric ML system effectively identifies the target label among the labels in the SOFAR-Image-Test dataset. Our study designed a one class recognition experiment upon two target labels: Defect and Dirt. Regarding the Defect label, given an inference dataset (SOFAR-Image-Test), we set the model to discriminate Defect samples from the other labels; thus, we let the model solve one v. all recognition task. On the Dirt label, the model shall identify Dirt samples from the other labels. We interpret the higher one-class recognition performance implies the better OOD detection performance as it can effectively discriminate the target label from the other labels. As an evaluation metric for one-class recognition, we use the AUROC along with various OOD detection thresholds. The experiment result is illustrated in Table 3.

5.1.2 Irrelevant Sample Rejection. Furthermore, we further examined whether the efficiency-centric ML system effectively rejects samples irrelevant to the domain. In the real world, there frequently exists samples that are not related to the target domain. While we deal with labels regarding the car’s state based on the SOFAR-Image dataset, the human inspectors frequently discovered images not relevant to the car’s state, as shown in Figure 4. If these samples are not rejected in the OOD detector, the model will yield a faulty prediction result on the irrelevant sample, which deteriorates the end-user’s business operations. Therefore, our study retrieved about 180 samples at two irrelevant labels (Receipts and Documents) from the database and checked whether the proposed efficiency-centric ML system precisely rejects them. We also employed the AUROC for evaluation. The experiment results are shown in Table 4.

![Figure 4: Samples in OOD labels (Document and Receipts).](image-url)
5.2 Experiment Results
Along with the OOD detection performance in Table 3 and Table 4, we resulted in the proposed efficiency-centric ML system is surprisingly outperformed the task-centric approach and multi-task learner. We analyzed that this significant performance derives from the wider representation power of the proposed ML system compared to the other benchmarks. As we noted in section 4.3, one primary characteristic of task-centric paradigm is biased representation power on the given classes. While this characteristic was advantageous in accomplishing higher task accuracy, however, we analyze this biased representation becomes a disadvantage in OOD detection. Referring to numerous studies on OOD detection [7, 20], a key element of an effective OOD detector is a qualified representation power that can understand samples in various labels, even if they exist outside of the domain. As the task-centric ML system aims to establish a biased representation to the target class only, we analyze the representation power cannot understand OOD samples properly. Not only the task-centric approach, but the multi-task learning approach also lacks the quality of representation power as its biasedness exists in multiple heads that solve each task. On the other hand, our efficiency-centric ML system lets the model explore wider representation space based on the concatenated dataset (SOFAR-Image); thus, the model can acquire more qualified representation power which understands more general characteristics of various samples. Consequently, we resulted in the proposed efficiency-centric ML system being more robust to OOD samples in a real world setting. In a practical manner, we analyzed this superficial OOD detection performance is a key advantage of our efficiency-centric ML system.

Table 3: One-Class Recognition Performance

| Method               | AUROC  |
|----------------------|--------|
|                      | Dirt   | Defect |
| Efficiency-centric (OURS) | 0.998  | 0.995  |
| Multi-Task Learning  | 0.938  | 0.873  |
| Task-Centric         | 0.993  | 0.971  |

Table 4: Irrelevant Sample Rejection Performance

| Method               | AUROC  |
|----------------------|--------|
|                      | Receipts | Documents |
| Efficiency-centric (OURS) | 0.958    | 0.971      |
| Multi-Task Learning  | 0.604   | 0.577      |
| Task-Centric (Dirt)   | 0.884   | 0.853      |
| Task-Centric (Defect) | 0.841   | 0.817      |

5.3 Recommendations
Following the experiment results, we recommend the candidate ML practitioners to employ our efficiency-centric ML system when their domain prioritizes an OOD detection performance. Along with our recommendations proposed in Section 4, the task-centric paradigm’s biased representation power is a double-edged sword. While the biased representation power of the task-centric classifier accomplishes higher task accuracy, it also has less robustness toward the OOD samples. On the other hand, the proposed efficiency-centric paradigm achieves competitive performance to the task-centric approach, but it guarantees significant robustness against the OOD samples. We would like to highlight a trade-off between task-centric paradigm and efficiency-centric paradigm to the candidate ML practitioners. If their domain does not include many OOD samples (i.e., the data source is well-controlled not to create OOD samples), they might adapt task-centric ML system to enjoy the precise task accuracy. Conversely, suppose their domain bears many OOD samples similar to our car sharing platform (i.e., the data source is hardly controlled or user-generated samples on a large scale). In that case, we highly recommend the to adapt the proposed efficiency-centric ML system with competitive task accuracy and superficial OOD detection performance.

6 DOES THE EFFICIENCY-CENTRIC CLASSIFIER CALIBRATE WELL?

6.1 Setup
Last but not least, we aimed to validate whether the efficiency-centric ML system can yield sufficient prediction results on a multi-labelable sample. The multi-labelable sample is a sample that has attributes of two labels simultaneously. In a car sharing platform, as shown in Figure 5., a multi-labelable image frequently exists that has both dirt and defect on its surface. Suppose these multi-labelable samples exist in the inference dataset. Under the task-centric ML system, a single ML system can only identify one state (label) of the given multi-labelable sample. Referring to car defect recognition system and car dirt recognition system, the defect recognition system can only identify Dirt while the dirt recognition system can only recognize Defect. As human inspectors (end-users) desire to acquire both prediction results (Defect and Dirt), the prior task-centric paradigm repetitively provide the inference dataset into two ML systems (defect recognition and dirt recognition). Then, prediction results at each task are accumulated in separated prediction result tables, and the end-user has to query every table to utilize prediction results.

Figure 5: Example images of the multi-labelable samples

We figure out several drawbacks upon the aforementioned procedures dealing with multi-labelable samples. First, one sample has to be inferenced twice to provide multi-labels, and it creates a particular amount of resource consumption. Second, the end-user should query multiple tables to retrieve prediction results (Defect
and Dirt) on a single sample, which causes inconvenience. Therefore, we aim to examine whether our efficiency-centric ML system can provide adequate prediction results on a multi-labelable sample only with a single inference. As the proposed classifier learned the characteristics of both Defect and Dirt, we expected the Top-2 prediction results of the efficiency-centric classifier would include Defect and Dirt label. If our expectation is valid, we can perform inference for a single time on the inference dataset and accumulate Top-2 prediction results in a single table; this would reduce resource consumption and provide convenience to the end-users. Therefore, we acquired 100 multi-labelable samples of Defect and Dirt and examined whether the efficiency-centric classifier yield these two labels in Top-2 prediction results. The experiment result is illustrated in Figure 6.

Figure 6: Top-2 predictions of our efficiency-centric classifier

Figure 7: Prediction results of the multi-task learner

6.2 Experiment Results

Based on the experiment result shown in Figure 6 and Figure 7, we resulted in our efficiency-centric classifier can provide satisfactory prediction results on a multi-labelable sample. Given multi-labelable samples of Defect and Dirt, majority of Top-2 prediction results include these two labels. We expect that this result derived from the widened representation power of the proposed efficiency-centric classifier. As the classifier learns each label’s characteristics from the SOFAR-Image, within the top-2 predictions, it properly predicted both labels (Defect and Dirt) if a given image has attributes of two labels. Note that the gap between prediction results of Defect and Dirt is not particularly large on the efficiency-centric classifier, which implies the representation power is calibrated well. On the other hand, we figured out that multi-task learned succeeded in predicting Defect on the multi-labelable samples while it could not effectively recognize Dirt. Moreover, the gap between predictions on Defect and Dirt is comparatively larger than the one in efficiency-centric classifier. We expect this different calibration performance between our efficiency-centric classifier and the multi-task learner derived from the representation power. As we have described in the prior section 5.2, the multi-task learner is biased to particular classes compared to the efficiency-centric classifier. Upon the result shown in Figure 7, we expect the representation power of multi-task learner is much biased to the characteristics of Defect rather than Defect. Thus, it creates a significant understanding of the Defect label, but comparatively worsens the knowledge on the Dirt label. Conversely, our efficiency-centric classifier yields competitive prediction results on both labels as long as its representation power is less biased to the particular label. In a nutshell, we validated that the proposed efficiency-centric classifier can provide adequate multiple prediction results on a multi-label label sample; thus, it can substitute the task-centric ML system to reduce the waste of computation resource and enhance the convenience of end-users.

7 REAL-WORLD DEPLOYMENT

Throughout the following sections, we discovered a trade-off between task accuracy and engineering efficiency regarding prior task-centric ML system and the proposed efficiency-centric ML system. While the task-centric ML system provides a precise task performance, it lacks engineering efficiency in two aspects: 1) The number of ML pipeline increases linear to the number of tasks, 2) As the task-centric ML system is not robust to OOD samples, the data pipeline should be well-controlled. Conversely, our efficiency-centric ML system yields an engineering efficiency. It can reduce the number of ML pipeline into one (at our problem setting), and the ML system effectively identifies OOD samples. Upon the trade-off between task accuracy and engineering efficiency, SOCAR adapted efficiency-centric paradigm due to several aspects. First, the damage from faulty predictions at the image recognition task was not excessively toxic, as the professional human inspectors monitor the model’s prediction results before business actions. Second, we prioritized robustness against OOD samples as we utilized user-generated images, which are hard to control. Lastly, we expect the efficiency-centric ML system would evade the computation resource waste when there exist many tasks in the near future. To provide a guideline to the candidate ML practitioners, this section describes how we deployed our efficiency-centric ML system in real world and how the end-users utilize it.

We deployed the proposed efficiency-centric ML system in a cloud environment, especially under the Kubernetes Engine on the Google Cloud Platform. As shown in Figure 8., the deployed system consists of serial actions of three components: producer pod, Kubernetes-based Event Driven Autoscaling (KEDA), and consumer pods. First, the producer pod retrieves the inference samples from the database, which is operated on the GCP BigQuery. When the cronjob operator triggers the operation, the producer publishes the inference dataset to the Cloud Pub/Sub. After the Cloud Pub/Sub receives the data from the producer pod, the KEDA [9] checks the number message in the Cloud Pub/Sub and decides the number of consumer pods established in the cloud environment. There were two primary drawbacks once we fixed the number of consumer pods
in the prior deployment system. We experienced frequent overloads on consumer pods when many inference samples were retrieved. We also encountered a waste of resources on consumer pods when fewer inference samples existed. Therefore, we set the KEDA in our deployment system to efficiently manage the number of consumer pods based on the size of the inference dataset. Lastly, the consumer pods retrieve inference samples from the Cloud Pub/Sub, feed the sample into the efficiency-centric OOD detector and the classifier to acquire the Top-2 prediction results. These results are inserted into the prediction result table, which is also managed with BigQuery. In a nutshell, we aim to design an efficient ML pipeline that can dynamically manage the number of consumer pods to maximize engineering efficiency.

8 RELATED WORKS

8.1 Multi-task Learning
Multi-task learning [3, 21], one of the traditional machine learning problems, aims to not only solve multiple tasks efficiently but also improve the performance of each task by exploiting the extracted knowledge from other tasks. There are well-known problems of training multiple tasks at once. First, when the scale of loss of each task varies, the model is easy to overfit to partial tasks. [2] proposed to normalize gradient by using the norm of loss from each task to alleviate the problem. Second, knowledge from one task can negatively affect other tasks. [11] tries to prevent the negative transfer by introducing an asymmetric autoencoder term. The conventional architecture of multi-task learning in the vision domain comprises a single backbone network (i.e., CNN) and a few heads (linear layer) for each task.

8.2 Out-of-Distribution Detection
The reliability of deep neural network is one of the most critical problems in real-world ML applications. The deployed model must be able to say ‘I don’t know’ when OOD input is given. Using pre-trained softmax deep neural network is one of the leading research directions. [7] gives a simple baseline, using the maximum value of softmax probability, for OOD detection. [13] shows temperature scaling and input pre-processing are effective for OOD detection. [12] proposes to use a simple Gaussian generative classifier and shows outstanding performance using Mahalanobis distance from test sample to mean of each class Gaussian. Even though many works based on the generative model to approximate training data distribution have emerged[10, 15], logit-based methods show better performance.

9 DISCUSSIONS AND CONCLUSIONS
In this study, we aim to introduce how we transformed prior task-centric ML system into the proposed efficiency-centric paradigm. We propose an efficient ML framework that concatenates multiple task-centric datasets, classifier, OOD detector, and prediction result table into a single pipeline to resolve this problem. We validated several takeaways throughout experiments on the various datasets retrieved from the real world car sharing platform. First, there exists a trade-off between task-centric and efficiency-centric paradigms. As the efficiency-centric ML system explores a larger label space, it acquired more general, less-biased representation power. We figured out this general representation power contributes to the competitive task accuracy with a supreme OOD detection performance. For the task-centric paradigm and other benchmarks, vice versa. Furthermore, we validated the efficiency-centric classifier calibrates better as it yields more qualified prediction results on the multi-labelable sample, while the task-centric classifier yields particularly biased prediction results. Lastly, we also elaborated on how we deployed this efficiency-centric ML system in the real world cloud environment. Still, there exist improvement avenues of our study. We would further excavate an underlying reason behind the trade-off between the task-centric and efficiency-centric paradigms. Moreover, we shall scrutinize how the efficiency-centric representation power differs from the baseline paradigms. Lastly, we would continuously evaluate the proposed paradigm’s effectiveness and robustness under the increased number of tasks in the real world. As a closing remark, we highly expect ML practitioners to utilize the proposed efficiency-centric ML system in their domains for the sake of engineering efficiency in the real world.

10 ACKNOWLEDGEMENT
We sincerely appreciate Sangtae Humphrey Ahn for his valuable guides, architecture designs, and supportings in deployment.
REFERENCES

[1] Sridhar Alla and Suman Kalyan Adari. 2021. What is mlops? In Beginning MLops with MLFlow. Springer, 79–124.

[2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multi-task Networks. arXiv:cs.CV/1711.02257

[3] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. arXiv:2009.09796 (2020).

[4] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhuina, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. 2020. Fine-Grained Visual Classification via Progressive Multi-Granularity Training of Jigsaw Patches. arXiv:cs.CV/2003.03836

[5] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. 2020. Rethinking importance weighting for deep learning under distribution shift. Advances in Neural Information Processing Systems 33 (2020), 11996–12007.

[6] Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016).

[7] Dan Hendrycks and Kevin Gimpel. 2018. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. arXiv:cs.NE/1610.02136

[8] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. arXiv:1812.04606 (2018).

[9] KEDA. [n.d.]. KEDA: Kubernetes Event-driven Autoscaling. https://keda.sh/

[10] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2020. Why Normalizing Flows Fail to Detect Out-of-Distribution Data. arXiv:stat.ML/2006.08545

[11] Hae Beom Lee, Eunho Yang, and Sung Ju Hwang. 2018. Deep Asymmetric Multi-task Feature Learning. arXiv:cs.LG/1708.00260

[12] Kinmin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. arXiv:stat.ML/1807.03888

[13] Shiyu Liang, Tixuan Li, and R. Srikant. 2020. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. arXiv:cs.LG/1706.02690

[14] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:cs.LG/1608.03983

[15] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2019. Do Deep Generative Models Know What They Don’t Know? arXiv:stat.ML/1810.09136

[16] Siddhartha Sankar Nath, Girish nichra, Jainyaseni Kar, Sayan Chakraborty, and Nilanjan Dey. 2014. A survey of image classification methods and techniques. In 2014 International conference on control, instrumentation, communication and computational technologies (ICCICCT). IEEE, 554–557.

[17] Damian A Tamburri. 2020. Sustainable mlops: Trends and challenges. In 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE, 17–23.

[18] Yaming Wang, Vlad I Moraru, and Larry S Davis. 2018. Learning a discriminative filter bank within a cnn for fine-grained recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4148–4157.

[19] Xiushen Wei, Jianxin Wu, and Quan Cui. 2019. Deep learning for fine-grained image analysis: A survey. arXiv preprint arXiv:1907.03089 (2019).

[20] Jingdong Yang, Kaiming Zhou, Tixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334 (2021).

[21] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering (2021).

[22] Bo Zhao, Jiaxu Feng, Xiao Wu, and Shusheng Yan. 2017. A survey on deep learning-based fine-grained object classification and semantic segmentation. International Journal of Automation and Computing 14, 2 (2017), 119–135.