



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Master's Thesis of Engineering

# Data Augmentation and Filtering for Supervised Learning using Splash Data Preprocessor

Splash 데이터 전처리 연산자를 이용한 지도  
학습 데이터 증강과 필터링

August 2021

Graduate School of Electrical and Computer  
Engineering  
Seoul National University

Yehyun Kim

# Data Augmentation and Filtering for Supervised Learning using Splash Data Preprocessor

Seongsoo Hong

Submitting a master's thesis of  
Engineering

August 2021

Graduate School of Electrical and Computer  
Engineering  
Seoul National University

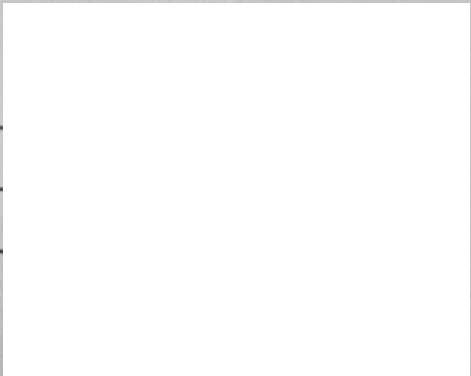
Yehyun Kim

Confirming the master's thesis written by

Yehyun Kim

August 2021

Chair	<u>Taewhan Kim</u>
Vice Chair	<u>Seongsoo Hong</u>
Examiner	<u>Kyuseok Shim</u>



# Abstract

Splash is a graphical user interface programming framework designed to support artificial intelligence application development. Artificial intelligence experts in various fields including data, modeling, control engineers can easily develop artificial intelligence applications without profound programming knowledge through Splash's programming abstraction. To further increase Splash's functionality for supporting artificial intelligence application development, we are adding a language construct in Splash for data preprocessing. This language construct provides an easy-to-use data augementer and data filter, which are the main tasks of data preprocessing for data engineers in supervised learning.

Data augmentation and filtering are particularly important tasks in supervised learning because the training dataset's quality and quantity directly affect the accuracy of the model. Datasets such as MNIST and datasets prepared in person have data with accurate labels yet lack an amount of data and labels, so the datasets need augmentation for an increase in dataset quantity. When using a data label platform such as crowdsourcing or an automated label program to utilize numerous datasets for training, the datasets need filtering because they often include noisy labels. In this thesis, we implement basic data

augmentation and filtering techniques as a Splash language construct, called data preprocessor, to support data engineers.

Data augmentation function in Splash data preprocessor increases dataset quantity by using seven augmentation techniques: horizontal and vertical shift, horizontal and vertical flip, random rotation, random brightness, and random zoom. The data filtering function finds duplicated images with different and same labels, then removes those images to improve the quality of the training dataset. To demonstrate the feasibility of using Splash data preprocessor and to confirm the correctness of the data preprocessor implementation, we trained the CIFAR-10 dataset as an experiment using Splash data preprocessor. This experiment shows that training data filtering and augmentation can be easily performed using the Splash data preprocessor.

**Keyword:** Splash programming framework, training data preprocessing, supervised learning data filtering, data augmentation

**Student Number:** 2019-28252

# Table of Contents

Chapter 1. Introduction .....	1
Chapter 2. Splash programming language .....	4
Chapter 3. Splash data preprocessor .....	9
Chapter 4. Splash data preprocessor experiment.....	14
Chapter 5. Conclusion .....	18
References .....	19
Abstract in Korean .....	21

**Tables**

**[Table 1]..... 8**

**Figures**

**[Figure 1]..... 4**

**[Figure 2]..... 6**

**[Figure 3]..... 9**

**[Figure 4]..... 9**

**[Figure 5]..... 11**

**[Figure 6]..... 13**

**[Figure 7]..... 14**

**[Figure 8]..... 16**

**[Figure 9]..... 17**

# Chapter 1. Introduction

Splash is a graphical user interface programming framework that allows developers to easily handle software structures that are becoming more complex with the advancement of artificial intelligence. The main goals of Splash are to provide programming abstraction, real-time stream processing support for artificial intelligence applications, programming support for real-time control systems, and performance optimization of a software system [1]. Artificial intelligence experts in various fields including data, modeling, control engineers can easily develop artificial intelligence applications without profound programming knowledge through Splash's programming abstraction. To further increase Splash's functionality for supporting artificial intelligence application development, we are adding a language construct in Splash for data engineers.

The data engineer's primary job is data preprocessing, such as data filtering and augmentation. Data preprocessing is becoming more important because the quantity and quality of the dataset required are increasing as the range of industries that demand the use of artificial intelligence applications are broadening [2][3][4]. These artificial intelligence applications that are used by various industries often use supervised learning as a method to train machine learning models. In supervised learning, a dataset refers to labeled data, and labeled data is necessary for training a model [5]. Labeled data can be obtained by using datasets such as MNIST and CIFAR-10, which have been



carefully reviewed for a long time, or by labeling the data in person. MNIST and CIFAR-10 datasets are reliable, but the amount of labeled data are limited. Labeling the data in person requires tremendous time so it is impractical to obtain a large amount of labeled data [6][7].

To acquire a large number of classes and data, one can use a data labeling platform such as crowdsourcing or automatic labeling that uses a vast amount of data and labels on the internet. However, labeled data procured from crowdsourcing platforms or automatic labeling often includes data with incorrect labels, which are named noisy labeled data. Training with noisy labeled data causes performance degradation of a trained model [8].

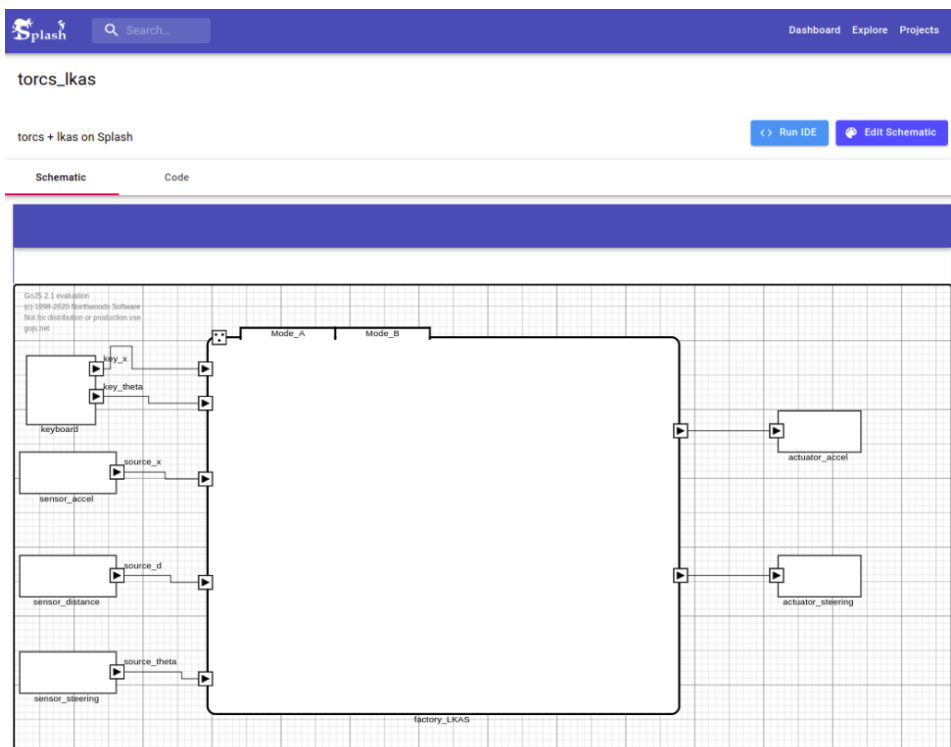
We are implementing a data preprocessor as a language construct of Splash to support data engineers in two ways: data augmentation for an increase in the amount of dataset and data filtering to easily filter noisy labeled data.

Data augmentation function in Splash data preprocessor increases dataset quantity by using seven augmentation techniques: horizontal and vertical shift, horizontal and vertical flip, random rotation, random brightness, and random zoom. The data filter function finds duplicated images with different and same labels. Duplicated images with different labels are removed and duplicated images with the same labels are removed except for one. These removals of duplicated images improve the quality of the training dataset. To demonstrate the feasibility of using Splash data preprocessor and to validate the correctness of

Splash data preprocessor implementation, we trained the CIFAR-10 dataset using Splash data preprocessor as an experiment. From this experiment, we show that training data filtering and augmentation can be easily performed using the Splash data preprocessor.

## Chapter 2. Splash Programming Language

Splash is a graphical user interface programming framework that supports the development of artificial intelligence applications. Figure 1 shows an example of the Splash program. Splash was developed with an emphasis on supporting the fusion and real-time processing of sensor stream data.



**Figure 1. A Splash program example.**

Splash provides developers with programming abstraction to intuitively develop artificial intelligence applications. Developers can use Splash to specify an application's end-to-end timing constraints,

and Splash monitors and handles violations of specified timing constraints at runtime. In addition, Splash provides support for implementing sensor data fusion, exception handling, mode change, and features to help optimize and tune application performance.

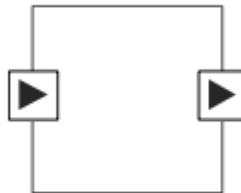
The splash application developer uses a program in the form of a directed data flow graph. The data flow graph is automatically converted into a template code for data transmission/reception by Splash's code generator. This automated code generation of data flow graph makes it convenient for developers to handle complex applications since the developer only needs to focus on the implementation of the internal algorithm.

The language constructs of the Splash program are component, port, channel, and clink. This section explains these language constructs and their functionalities.

## 2.1. Components

A component corresponds to the node of the directed graph and is the basic execution unit of the Splash application. In Splash, there are two kinds of components: composite and atomic components. The composite component is a factory, which is the largest building block in the Splash program. The atomic component consists of a processing component, a source component, a sink component, and a fusion operator.

A processing component, as shown in Figure 2, performs user-defined computation on data received from the stream input port then generates output data to the stream output port. The processing component may have multiple stream input and output ports.



**Figure 2. Processing component.**

A source component receives the signal from sensors outside of the Splash system and produces stream data as an output. The source component does not have a stream input port, since it receives signals outside of the Splash system, and it only has a single stream output port.




A sink component is opposite to a source component. It has a single stream input port and no stream output port since it receives stream data from the Splash system and transfers those data into a system out of Splash.

A fusion operator merges multiple stream input data into a single stream output data. The fusion operator is made for developers to conveniently manage the complicated implementation challenges of sensor fusion algorithms [1].

## 2.2. Ports

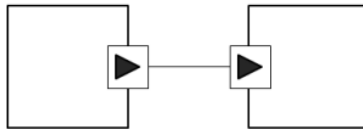
A port receives stream data from a channel and sends stream data out to a channel. There are three types of ports in Splash: stream input and output ports for receiving and sending stream data, event input and output ports for receiving and sending events, and mode change input and output ports for receiving and sending mode change signals. Graphical symbols for different types of ports are shown in Table 1.

**Table 1. Symbol for Ports**

Stream Port	
Event Port	
Mode Change Port	

## 2.3. Channel and Clink

A channel is a path for stream data that connects the stream input port and stream output port. As shown in Figure 3, the channel is symbolized as a solid line. Clink is a path for event and mode change signals. Clink is represented as a dotted line as shown in Figure 4.



**Figure 3. Graphical representation of a channel.**



**Figure 4. Graphical representation of a clink.**



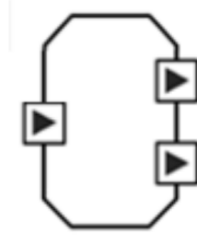
## Chapter 3. Splash Data Preprocessor

This chapter introduces data preprocessor, a new language construct to provide programming abstraction for data preprocessing and the semantics of data preprocessor. Also, this chapter describes the implementation of the Splash data preprocessor.

### 3.1. Data preprocessor and semantics

Figure 5 is the graphical representation of the data preprocessor. The data preprocessor has one stream input port and two stream output ports. Stream input port receives data with labels through a channel. The developer can set a rule to configure several augmentations to add per labeled data by double-clicking the data preprocessor. Also, developers can set a rule on whether to filter the data. The top stream output port produces augmented dataset or clean labeled data if filtering function is used, and the bottom stream output port produces the filtered noisy labeled data.

The Splash data augmentation function supports seven augmentation techniques: horizontal and vertical shift, horizontal and vertical flip, random rotation, random brightness, and random zoom. The Splash data filtering function performs two main tasks. First, it finds duplicate images with the same label and leaves only one image with label. If the duplicate images have different labels, then it excludes all of those datasets.



**Figure 5. Data processor.**

### **3.2. Data augmentation function**

We used ImageDataGenerator available at the Keras deep learning library for data augmentation [9]. Seven data augmentation techniques are used to increase the number of datasets. All of the data augmentation techniques keep the image dimensions the same. Horizontal and vertical shift augmentation technique shifts all pixels of the image horizontally or vertically. Horizontal and vertical flip technique flips the images randomly. The random rotation augmentation technique can rotate the image clockwise from 0 to 360 degrees. The random brightness augmentation technique randomly darkens or brightens images. The random zoom augmentation technique randomly zooms in or out ranging from 50% zoom in to 100% zoom out.

Developers can specify which data augmentation techniques and how many to use to increase the amount of dataset. Also, developers can just specify the number of the dataset to increase, and the Splash data augmentation function will randomly pick from seven techniques.

### 3.3. Data filtering function

Figure 6 is the pseudocode of the Splash data filtering algorithm. The goal of the algorithm is to return filtered labeled data. The first input is a list  $D$  that is created using  $d_i$ , which is an image in the form of an array. The second input is a list of labels  $L$  that corresponds to the images in  $D$ .  $H$  is a hash table that stores image and label information.  $H$  uses an image array  $d_i$  as a key, and uses a list consisting of a label and a logic filter flag as a value.

Lines 2 of the **DUPLICATEFILTER** algorithm looks for key-value  $d_i$  in the hash table  $H$ . If the key-value  $d_i$  is not in  $H$ , the algorithm adds key  $d_i$  and value of a list that consists of  $l_i$  and flag 0 to  $H$  (line 3). If the key-value  $d_i$  is already in  $H$ , then the algorithm checks whether the label  $l_i$  is the same with the label stored in  $H$  (line 5). If the label  $l_i$  is the same as the label stored in  $H$ , then the algorithm removes  $d_i$  and  $l_i$  (line 6). If the label  $l_i$  is different from the label stored in  $H$ , then the algorithm removes  $d_i$  and  $l_i$  and sets the logic flag in  $H$  to 1 (line 8-9). Line 10-12 of the **DUPLICATEFILTER** algorithm iterates through  $H$  to find a logic filter flag with value 1 and removes  $d_i$  and  $l_i$  if the flag value is 1.

---

**ALGORITHM 1. DUPLICATEFILTER**

---

**Input:** A list of data images  $D = [d_1, d_2, \dots, d_m]$   
A list of data labels  $L = [l_1, l_2, \dots, l_m]$

**Variables:** An Empty Dictionary  $H = \{\}$

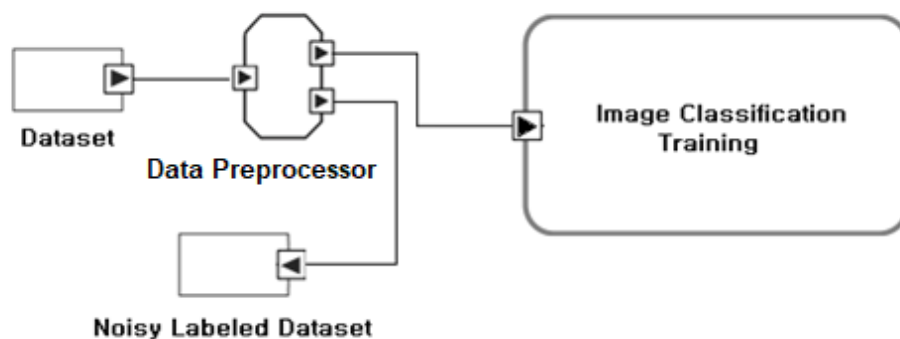
```
DUPLICATEFILTER ( $D, L, S$ )
1:  for  $i = 1$  to  $m$ 
2:      if  $H.get(d_i) == \text{None}$ 
3:           $H[d_i] = [l_i, 0]$ 
4:      else
5:          if  $l_i == H[d_i][0]$ 
6:              MOVETOREDUNDANT( $d_i, l_i$ )
7:          else
8:              MOVETOLOGICERROR( $d_i, l_i$ )
9:               $H[d_i][1] = 1$ 
10:  for  $j = 1$  to  $\text{len}(H)$ 
11:      if  $H[d_j][1] == 1$ 
12:          MOVETOLOGICERROR( $d_j, l_j$ )
13:  return ( $D, L$ )
```

---

**Figure 6. Pseudocode of DUPLICATEFILTER**

## Chapter 4. Splash Data Preprocessor Experiment

This chapter demonstrates an example of using the Splash data preprocessor on training a model for image classification. Two experiments are performed using Splash data preprocessor. One experiment uses the Splash data filtering function and the other uses the Splash data augmentation function.



**Figure 7. Image classification with Splash data preprocessor.**

Figure 7 is an example of a supervised learning using data preprocessor between a dataset and an image classification training model. When a developer adds the Splash data preprocessor to the right of the dataset and sets the rules for filtering or augmentation, Splash automatically filters or augments the dataset. Developers can easily filter and augment data without having to deal with complicated coding works.

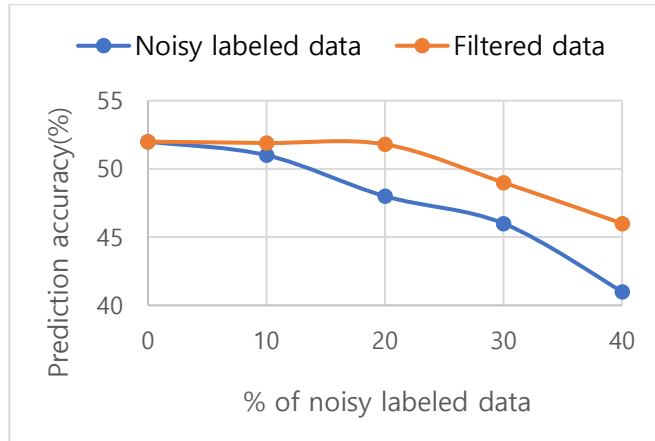
To confirm the correctness of the Splash data filtering and data

augmentation implementation, we trained a simple convolutional neural network with the CIFAR-10 dataset. This convolutional neural network has 3 convolutional layers, flatten layer, 2 hidden layers, and an output layer with a SoftMax activation function.

To check the correctness of Splash data filtering implementation, noisy labeled data was generated by assigning a random label on the clean CIFAR-10 dataset then those noisy labeled data were added to the original CIFAR-10 dataset. We added the noisy labeled data so that the percentage of the noisy labeled data ranges from 10% to 40% of the dataset.

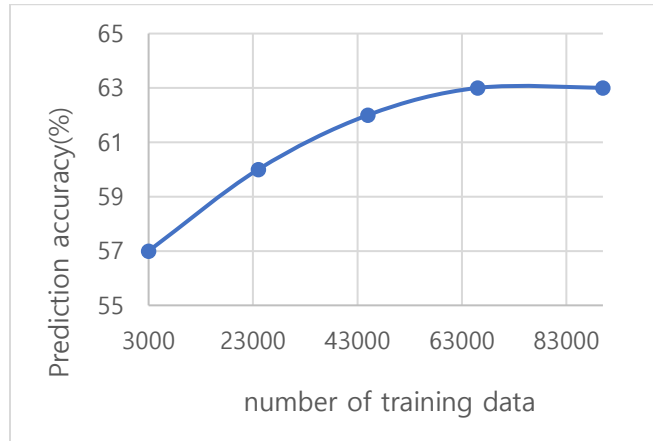
The image classification model was trained with a noisy labeled CIFAR-10 dataset to get the prediction accuracy of the model. Then we added the Splash data preprocessor to filter out the noisy labels and obtained the prediction accuracy of the image classification model.

The comparison between the prediction accuracy of the two models is shown in Figure 8. In Figure 8, the x-axis represents the ratio of noisy labeled data to the entire dataset, and the y-axis represents the prediction accuracy of the model after the training. The prediction accuracy of the noisy labeled dataset that passed through the Splash data filter was 5% higher than the noisy labeled dataset when there were 40% of noisy labeled data.



**Figure 8. Splash data filtering experiment result.**

To check the correctness of Splash data augmentation implementation, 1000 out of 50000 CIFAR-10 datasets were used for image classification. Randomly selected augmentation techniques were used to increase the number of the training dataset. In Figure 9, the x-axis represents the number of labeled data added to the training dataset using the data augmentation function, and the y-axis represents the prediction accuracy of the model after the training. Prediction accuracy of the augmented dataset that was added through the Splash data augmentation function increased up to 6%.



**Figure 9. Splash data augmentation experiment result.**



## Chapter 5. Conclusion

This thesis introduced a graphical user interface programming framework designed to support artificial intelligence application development named Splash, along with the key components and language constructs. The thesis demonstrated the feasibility of using Splash data preprocessor to filter out noisy labeled data and to augment data.

To confirm the correctness of the Splash data filter implemented in this thesis, the noisy labeled CIFAR-10 dataset was used to train an image classification model on Splash runtime. As a result of the experiment, the use of the Splash data filter improved the model's prediction accuracy by 7% in the dataset containing 40% of noisy labeled data.

To check the correctness of the Splash data augmentation implemented in this thesis, a limited number of CIFAR-10 dataset was used to train an image classification model Splash runtime. With the increase in the dataset with Splash data augmentation, the prediction accuracy improved from 57% to 63%.

For future work, Splash can provide more practicability to the developers by adding a language construct that automates all data preprocessing tasks, which is essential for artificial intelligence application development, including data filtering.

## References

- [1] N. Soonhyun and H. Seongsoo, "Splash: A Graphical Programming Framework for an Autonomous Machine," The 16th International Conference on Ubiquitous Robots (UR 2019) , pp. 660-666, Jun 2019.
- [2] Iliou, Theodoros, et al. "A novel machine learning data preprocessing method for enhancing classification algorithms performance." *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*. 2015.
- [3] Cubuk, Ekin D., et al. "Autoaugment: Learning augmentation strategies from data." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
- [4] Soni, Neha, et al. "Impact of artificial intelligence on businesses: from research, innovation, market deployment to future shifts in business models." *arXiv preprint arXiv:1905.02092*. 2019.
- [5] A. Khetan, Z. C. Lipton, A. Anandkumar, "Learning from Noisy Singly-Labeled Data," in arXiv preprint arXiv: 1712.04577, 2017.
- [6] I. Misra, C. L. Zitnick, M. Mitchell, R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016
- [7] T. Xiao, T. Xia, Yi Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[8] J. Li, Y. Wong, Q. Zhao and M. S. Kankanhalli, "Learning to Learn from Noisy Labeled Data," in Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[9] "How to Configure Image Data Augmentation in Keras," [Online]. Available: <https://gist.github.com/mohdsanadzakirizvi>

# Abstract

Splash는 인공 지능 응용 개발을 지원하기 위해 만들어진 GUI 프로그래밍 프레임워크이다. Splash는 프로그래밍 추상화를 통해 데이터, AI 모델링, 제어 엔지니어를 포함한 여러 분야 전문가들이 프로그래밍적 지식 없이도 손쉽게 사용할 수 있도록 만들어졌다. 인공 지능 응용 개발을 지원하는 Splash의 기능을 더욱 향상시키기 위하여 데이터 전처리 기능을 Splash의 언어 구조로 추가하였다. 이 언어 구조는 데이터 엔지니어의 주요 업무인 데이터 전처리 중 데이터 필터링과 증강 기능을 지원한다.

지도 학습(supervised learning)에서 데이터 필터링과 증강은 특히 중요한 작업이다. 지도학습을 위해서는 레이블이 되어있는 데이터가 필요한데, 쉽게 구할 수 있는 MNIST와 같은 학습 데이터셋이나 직접 레이블링 한 데이터셋은 수가 한정적이다. 따라서 데이터의 수를 증가시키기 위하여 데이터 증강 기술이 필요하다. 많은 수의 데이터셋을 활용하기 위해서 클라우드소싱 같은 데이터 레이블 플랫폼이나 자동 레이블 프로그램을 이용하는 경우, 레이블이 잘못되어 있는 경우가 많기 때문에 이를 필터링해야 한다. 본 논문에서는 지도 학습에서 필요한 기본적인 데이터 필터링 기법과 데이터 증강 기법을 Splash에 구현하여 데이터 엔지니어가 손쉽게 이용할 수 있도록 한다. Splash 데이터 전처리 연산자는 이미지의 중복성을 판단하여 필터링하고, 일곱 가지 방법으로 이미지를 증강시킨다. 우리는 Splash 데이터 전처리 연산자를 사용하여 지도 학습 데이터 필터링 및 증강을 쉽게 수행 할 수 있음을 보였다.

**주요어:** Splash 프로그래밍 프레임워크, 학습 데이터 전처리, 지도 학습

데이터 필터링, 데이터 증강

**학 번:** 2019-28252