A Visualization Method for Training Data Comparison

*

Karen Kosaka
Ochanomizu University
Tokyo, Japan
kosaka.karen@is.ocha.ac.jp

Takayuki Itoh
Ochanomizu University
Tokyo, Japan
itot@is.ocha.ac.jp

***Abstract*—With the diversification of machine learning applications, the quality verification and comparison of training data has been an important process. For example, while performing transfer learning, verification the difference in the quality between the source and the target data can prevent the accuracy of the model from deteriorating. However, training datasets for deep learning is getting larger and larger, and analysis of such datasets is not always easy. As a solution to this problem, we are working on the visualization for training data validation. In this study, we apply dimensionality reduction to the training datasets and display them as scatterplots to realize a visual analysis that can easily detect differences in the quality. Our current implementation draws the regions where the points are concentrated as semitransparent polygons for each label in the scatterplot. Also, the implementation provides a slider to set a threshold for the interactive adjustment of polygon generation. This allows us to observe the differences in the distribution of labels among the training data.**

***Index Terms*—visualization, machine learning, training data**

I. Introduction

As the data for machine learning has been more and more diverse, the comparison of training datasets has also been more and more important. For example, in transfer learning, differences in the quality of the source and target data may worsen the accuracy of the trained model. In other cases, for example, when training datasets are generated from multiple datasets in the process of model building, it is worthwhile to analyze the differences in the datasets. Meanwhile, training data sets used in machine learning have been getting larger in recent years. Comparison of such large datasets has been complex accordingly. It is therefore important to compare the training datasets not only quantitatively but also qualitatively; visualization is an effective tool for such qualitative data comparison. In this study, we define the target training dataset as follows.

- A training dataset consists of a large number of samples. The samples include image files, audio files, and document files. We target still images in this study.
- Multidimensional feature vectors are computed for the samples. Our current implementation assumes that each sample has exclusively one label, but we would like to extend this implementation so that one or more labels can be assigned to them.
- Two or more training datasets are visualized on the same screen. We assume the same feature values are calculated for all training datasets. The label assigned to each training dataset does not have to be exactly the same.

The requirements for visualization of such training datasets in this study are as follows.

Requirement 1:
: Represent the differences in distribution among the training datasets on the same screen.

Requirement 2:
: Represent how the similar samples concentrate and how the outlier samples distribute for each label assigned to the training data comprehensively.

Requirement 3:
: Represent how a group of samples with the same label has different distributions depending on the training datasets comprehensively.

In order to satisfy the above requirements, we propose the following visualization method.

- The method applies the same dimensionality reduction method to all the samples in the training datasets and maps all the samples to the same screen space. This satisfies Requirement 1.
- The method generates polygons enclosing groups of samples densely placed on the screen and the same class is assigned. The method also highlights the outliers that are not enclosed by the polygons. These satisfy Requirement 2.
- The method assigns the same hue to a group of samples with the same class belonging to one of the multiple training datasets. This satisfies Requirement 3.

Our goal in this study is to show the users the factors that worsen the accuracy of machine learning models by applying the visualization methods described above.
The next section introduces related work. We describe the visualization method in Section 3, and present the example

Identify applicable funding agency here. If none, delete this.

in Section 4. Section 5 concludes this paper with a discussion on limitations and future work.

II. Related Work

A. Transfer Learning

Transfer learning [1] is a machine learning method that learns by transferring information to different domains or tasks. Various application domains such as computer vision use it to reduce the burden of manual labeling. One of the main problems of transfer learning is the effect of the mismatch of distributions between different domains. Many studies that solve this problem have been proposed.

Meanwhile, while reusing the learning models to another machine learning task [2], we may need to incorporate the learning results to another neural network that has the same structure without destroying the models. Also, representation learning, which uses the lower layers of a Convolutional Neural Network (CNN) to transfer information, has a similar problem. Autoencoder [3] is one of the most popular representation learning methods that form a neural network aiming to obtain smaller feature representations.

Ma et al. [4] used the Office-31 dataset as an example data for transfer learning. Office-31 is a real-world dataset that has been widely used to demonstrate transfer learning algorithms. Images of Amazon product pages (”amazon”, 2817 images in total) are used as source domain data while webcam photos (”webcam”, 795 images in total) are used as target domain data in the study of Ma et al [4]. Figure 1 shows the visualization screen of this study. This example shows that there are classes with the same accuracy in both models, such as bike and calculator, and classes with different accuracy between the two models, such as filecabinet and phone. Figure 2 also shows that the images in the two domains share the same features such as vision and object appearance in the class where the performance is similar between the two models. On the other hand, the patterns are very different between the two domains in the class where the performance of the two models is different.

We aim to visualize such differences in quality between the multiple datasets in this study. In addition, as shown in the previous example, usage of different quality of training datasets may worsen the accuracy of the model. We aim to visualize the possibility that the accuracy of the model is worsened due to the difference of quality so that the users can explore what kind of datasets can be useful to generate models with higher accuracy.

B. Visualization for Machine Learning

There have been studies on visualization specific to datasets used in machine learning models. Swabha et al. [5] proposed a method to analyze and visualize the quality of datasets. This method builds a data map of the dataset and visualizes the dataset with respect to the model. Specifically, we categorized each dataset in terms of its contribution to the improvement of the accuracy for learning different models. This classification is divided into three regions: easy-to-learn, ambiguous, and

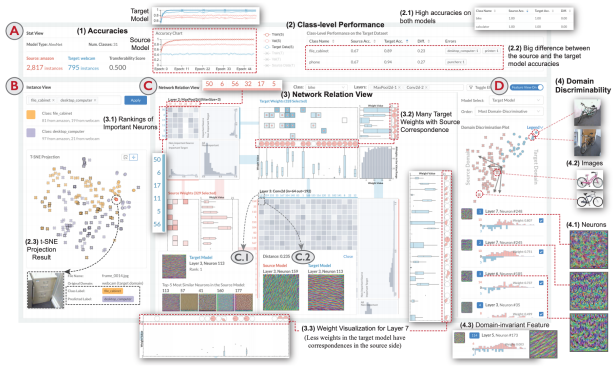

Fig. 1. Visualization of transfer learning by Ma et al. [4] consists of four visualization components.

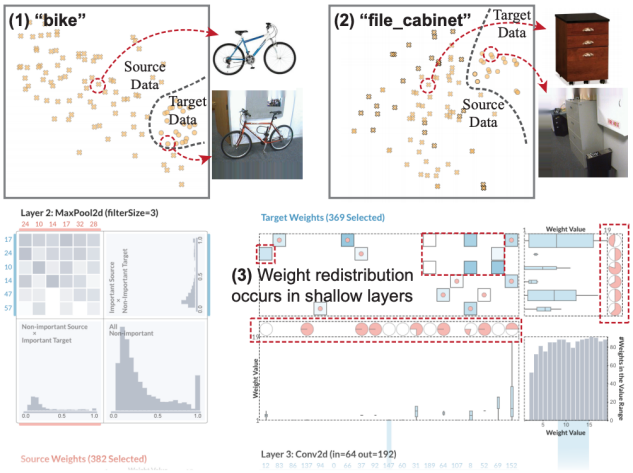

Fig. 2. Visualization results for (1):bike and (2):filecabinet using t-SNE by Ma et al [4].

hard-to-learn, which allows users to know how the data is contributing to the learning of the model. Smilkov et al. [6] also set up three tasks to visualize the dataset by investigating how users would like to use the dimensionally reduced data. The first task searches for local neighborhoods, the second task displays the global geometry and find clusters, and the third task finds meaningful ”directions”. The first task assists in determining if the points in a particular neighborhood were semantically related. The second task aims to find clusters of related data. The third task assists in determining if the embedding space contained meaningful directions.

A visualization method by Ma et al. [4] is specific to transfer learning. This study is based on the assumption that the training data and the unlabeled data constitute the same distribution in many models. However, this assumption does not make sense in many real-world situations. Ma et al. [4] developed a visualization method that illustrates how knowledge learned from an existing model is transferred to a new learning task in the learning process of Deep Neural Network (DNN). On the other hand, under the situation where a training dataset is selected during the process of creating a

model, there have not been many studies on visualization to analyze and compare the differences in the quality of training datasets. In this paper, we propose a visualization method that enables users who are not familiar with machine learning to understand how the quality of multiple training datasets differs. In addition, by visualizing the differences in the quality of the training datasets, we aim to infer the differences in the quality of machine learning in the process before starting to build the model.

C. Visualization of multidimensional data

Since the training datasets targeted in this research form sets of samples that have multidimensional vectors, we can apply multidimensional data visualization methods to represent the datasets. Multidimensional data visualization is an active research topic that has been discussed for a long time. Many visualization methods extract low-dimensional subspaces that are highly significant to visualize, as an approach to visualizing only the important parts of multidimensional data. The visualization method presented by Itoh et al. [7] displays a group of low-dimensional subspaces, which are selected by interactively manipulating a dimensional scatterplot on the right part of the screen, by means of multiple parallel coordinate plots (PCP) on the left part of the screen. This method makes it possible to interactively adjust the number of PCPs. Extending the idea of this method, Nakabayashi et al. [8] proposed a multidimensional data visualization method displaying selective sets of scatterplots instead of low-dimensional PCPs. The processing procedure of this method consists of the following two steps.

- Simple and interactive slider operation is used to select a user-defined number of important scatterplots that assign arbitrary pairs of variables in a multidimensional dataset to the two axes.
- Unique drawing of scatterplots consisting of polygons that represent subregions where points are densely placed and highlighted outlier points.

The visualization method presented in this paper applies dimensionality reduction instead of selecting scatterplots in the method of Nakabayashi et al. [8] while inheriting the representation that draws polygons and outlier points.
The reasons why we chose this visualization design are

- Many machine learning engineers are familiar with scatter plots applying dimensionality reduction schemes that represent the data distribution.
- Colors have a good property to identify multiple datasets and labels on the scatter plot. It is easy to distinguish datasets and labels if we assign brightness and hue to them if the number of datasets is less than 4 and the number of labels is less than 12.
- It is preferable to clearly represent the boundaries of each cluster.

III. Visualization of multiple training datasets

This section presents our visualization method. As discussed in Section 1, our method applies dimensionality reduction to the features of all samples belonging to multiple training datasets and projects them onto a two-dimensional screen space. In our current implementation, we use t-SNE as the dimensionality reduction method. Then, for a set of samples that belong to the same training dataset and have the same label, we display the regions where the points are densely placed on the scatter plot as polygons assigning unique colors to them. Figure 3 shows an example of the visualization using the above drawing method.

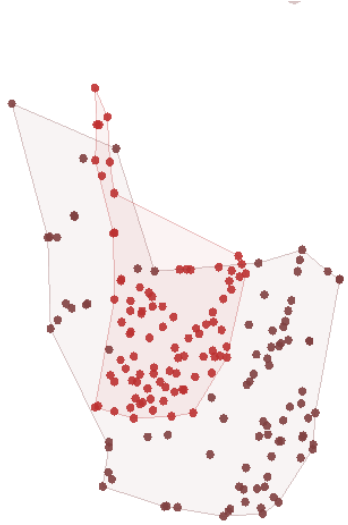

Fig. 3. Example of polygons enclosing densely placed points that belong to the same training dataset and have the same label.

We apply the Delaunay triangulation method to generate polygons that enclose the regions where points are densely placed as implemented by Nakabayashi et al [8]. The Delaunay triangulation method generates a triangulated mesh by connecting a given set of points so that the minimum angle of the triangles constituting the mesh is maximized. The implementation of Nakabayashi et al. [8], firstly generates a large rectangle that encloses all the points in the scatterplot. Then, it adds the points one by one into the triangular mesh and connects as vertices to update the triangular mesh sequentially. After all the points are added, the first large rectangle and the edges connected to the vertices of the rectangle are deleted. This implementation then deletes triangles that have the edges longer than a user-specified threshold t_{len} and modifies the triangular mesh so that it consists of only points with close distances. Then, it forms the outer boundaries of remaining triangles corresponding to the polygons that enclose densely placed points without outlier points. In other words, the points outside the polygon are outliers. As a result of the above process, this method draws the following three types of objects.

- **Object 1:** Outlier points as small circles.
- **Object 2:** Semi-transparent triangles corresponding to regions where points are densely places.

- **Object 3:** Outer bondaries of the groups of semi-transparent triangles as thick edges.

Here, the color of each dataset and each label is specified by the following formula based on the HSB color system.

$$H = 2\pi \frac{i}{N}$$

$$S = B = \alpha \frac{j+1}{M} + (1.0 - \alpha)$$

The above formula specifies the HSB values of the i-th label of the j-th dataset, where N and M $(0 \leq i < N, 0 \leq j < M)$ are the total number of labels and datasets. a is a real parameter satisfying $(0 \leq a \leq 1)$. This formula assigns a unique saturation and lightness to each dataset, and a unique hue to each label.

Our implementation automatically assigns different hues to each of the labels without any operations for manual color specifications. Figure 4 shows a list of hues assigned to each label. The rows correspond to the datasets while the columns correspond to the labels. Same hues are assigned to the points that have the same label, while the same brightness and saturation values are assigned to the points belonging to the same dataset.

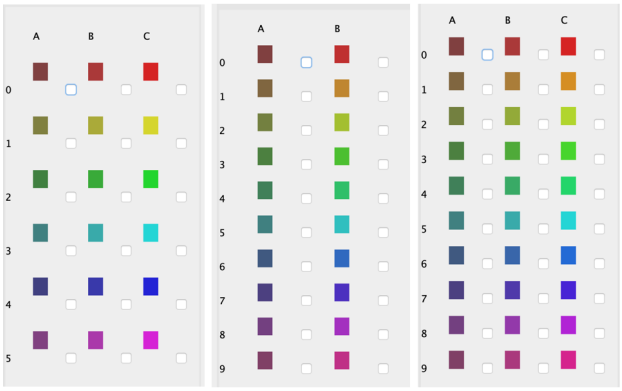

Fig. 4. A widget that shows automatic color assignment to each label, each dataset.

IV. Case Study

This section introduces a visualization example that aims at the comparison of multiple datasets. We applied two types of handwritten numeric image datasets, MNIST [1] and USPS [2], to this experiment. The original dataset of MNIST includes 60,000 training images and 10,000 test images where the number of pixels of each image is 28 × 28. The original dataset of USPS includes 7,291 training images and 2,007 test images where the number of pixels of each image is 16 × 16. Each image in these datasets is labeled with one of the numeric characters 0 to 9, which is represented as 10 different hues in the visualization result. We randomly selected 100 images for each label for each dataset and consequently formed subsets of MNIST and USPS that include 1,000 images for this experiment. The distribution of image features calculated from each image is visualized by applying dimensionality reduction in this experiment.

Figure 5 shows a snapshot of the visualization tool developed in this study. The implementation provides slider operations so that users can adjust several parameters. Figure 5(1) is a slider to adjust the threshold t_{len} which can be manipulated to adjust the size of the polygons. Figure 5(2) and 5(3) are sliders that adjust the variables α_1, α_3, which can be manipulated to adjust the colors of the outlier points and polygons.

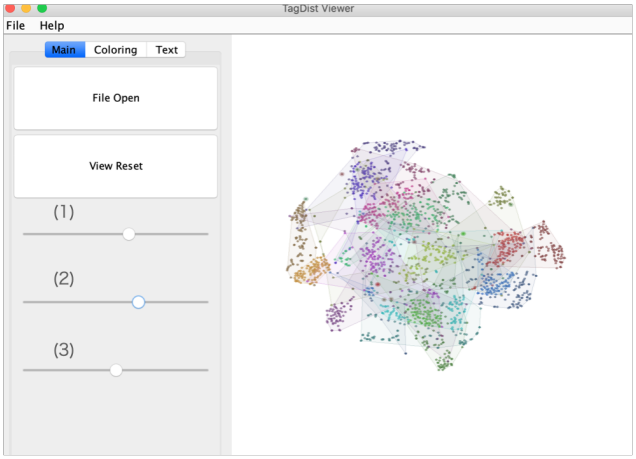

Fig. 5. An example. We use two training datasets, MNIST and USPS. The slider (1) is used to adjust the size of polygons. The sliders (2) and (3) are used to adjust the colors.

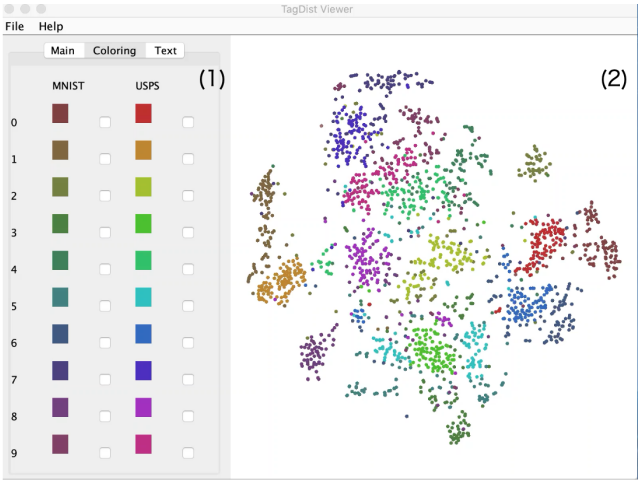

Fig. 6. An example. Each image is labeled with one of the numeric characters 0 to 9, which is represented as 10 different hues. The distribution of image features is visualized by applying dimensionality reduction.

Then, we selectively displayed the samples labeled as 6 or 9 from MNIST and USPS. We chose labels 6 and 9 in this experiment because they may look similar and therefore be confusing. Figure 7 shows the visualization. The points with

[1] http://yann.lecun.com/exdb/mnist/

[2] https://pytorch.org/vision/0.8/datasets.html#usps

the same label in the different datasets are closely placed each other. The groups of points labeled as 6 and groups of points labeled as 9 are located far from each other because their features are much different.

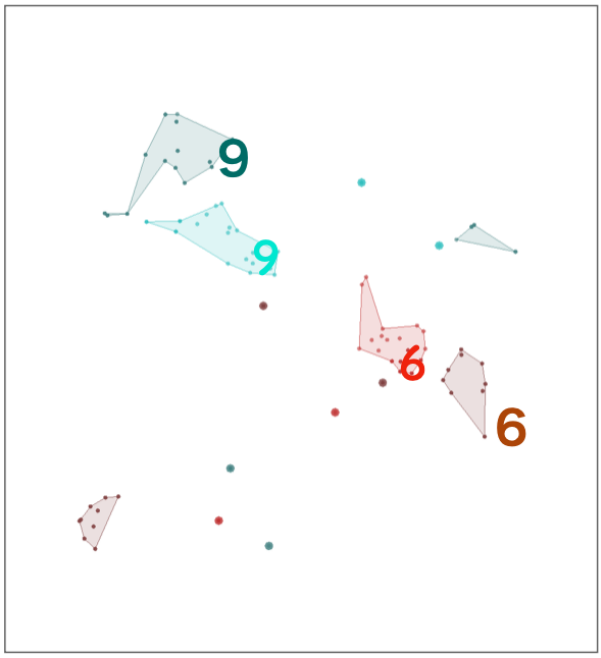

Fig. 7. The points labeled as 6 in MNIST is depicted in brown, labeled as 9 in green. The points labeled as 6 in USPS is depicted in red, labeled as 9 in light blue.

Figure 8 shows a visualization that selects samples labeled as 9 for each dataset, while Figure 9 shows another visualization that selects samples labeled as 6 for each dataset. We can observe that the points that have the same label will be closely placed each other even if they belong to different datasets. We can also observe several clear outlier points, and also, we can recognize which types of images are outliers in Figures 11 and 10.

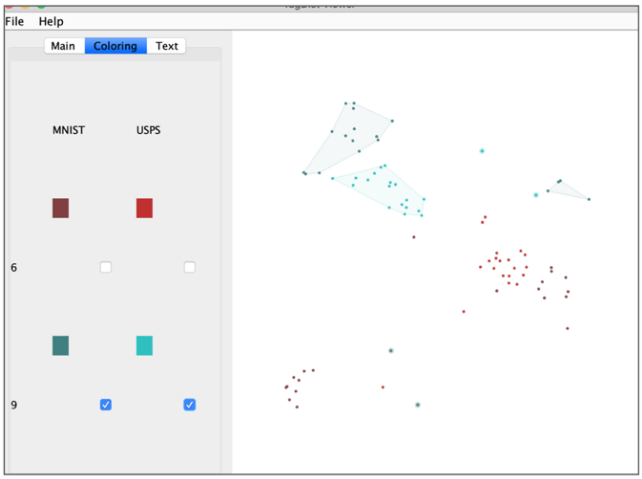

Fig. 8. The points labeled as 9 belonging to each of the datasets are displayed.

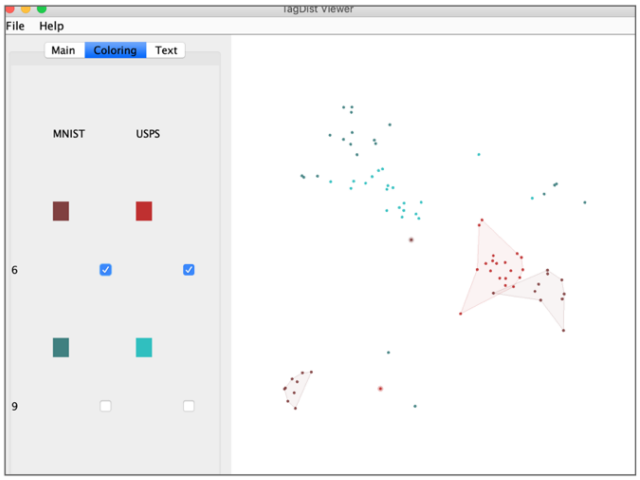

Fig. 9. The points labeled as 6 belonging to each of the datasets are displayed.

Figure 10 shows a visualization displaying only images written as 9 belonging to MINST or USPS. We suppose the points in the larger cluster are regular images, and the points outside the cluster are outliers. Here, we checked that the outliers were not mislabeled as shown in Figure 10. Figure 11 shows a visualization displaying only images written as 6 belonging to MNIST or USPS. This visualization result looks similar to those of the images written as 9. We could not find significant differences in the appearance of the outlier images. We estimate the reason for the generation of two clusters in Figure 11 is just a lack of the number of images written as 6. We would like to extend this experiment as future work by applying larger datasets and calculating features from middle layers of deep neural networks during the training phases.

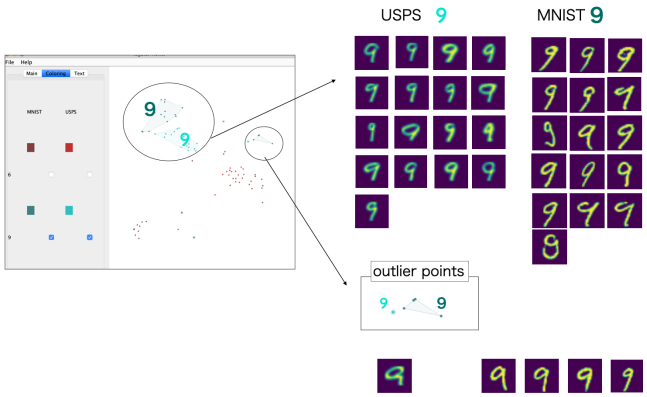

Fig. 10. Images written as 9 belonging to MNIST or USPS.

V. Conclusions and Future Directions

In this paper, we proposed a visualization method for comparing training datasets. We assume multiple training datasets consisting of a set of samples with feature vectors and labels in this study. The method applies the same dimensionality reduction to the datasets and displays them on a single screen. We can discover the factors that worsen the accuracy of the

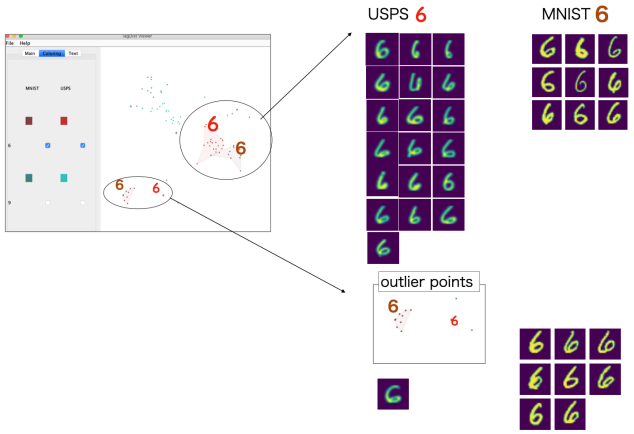

Fig. 11. Images written as 6 belonging to MNIST or USPS.

models hidden in the training datasets by visualizing features such as the middle layers of deep neural network models by using this method.

We would like to address the following issues in the future. First of all, we would like to implement the automatic setting of appropriate threshold values for defining the sizes of polygons enclosing densely placed points independently for each label. In addition, we would like to solve the limitation of the current visual representation that depicts a set of labels and datasets by colors. Also, we would like to extend the visual representation so that we can represent the samples that have two or more labels.

After resolving these issues, we would like to verify the effectiveness of our method with various datasets. Then, we would like to revalidate the effectiveness of our method through user evaluation experiments.

VI. Acknowledgement

This work is partically supported by JSPS KAKENHI grants.

References

[1] S. Jialin Pan,Q. Yang, “A Survey on Transfer Learning, Institute of Electrical and Electronics Engineers,” IEEE Pages 1345-1359, 2010.

[2] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, D. Wierstra, “PathNet: Evolution Channels Gradient Descent in Super Neural Networks,” Neural and Evolutionary Computing, arXiv:1701:08734v1, 2017.

[3] A. Ng, “Sparse autoencoder,” CS294A Lectire notes, 2011.

[4] Y. Ma, A. Fan, J. He, A. Reddy Nelakurthi, R. Maciejewski, “A Visual Analytics Framework for Explaining and Diagnosing Transfer Learning Processes,” IEEE Transactions on Visualization and Computer Graphics, 2020.

[5] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, Noah A. Smith, Y. Choi, “Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics,” Proceedings of EMNLP, 2020.

[6] D. Smilkov, N. Thorat, C, Nicholson, E. Reif, F. B. Viégas, M. Wattenberg, “Embedding Projector: Interactive Visualization and Interpretation of Embeddings,” NIPS 2016 Workshop on Interpretable Machine Learning in Complex Systems, 2016.

[7] T. Itoh, A. Kumar, K. Klein, and J. Kim, “ High-Dimensional Data Visualization by Interactive Construction of Low-Dimensional Parallel Coordinate Plots,” Journal of Visual Languages and Computing, Vol. 43, pp. 1–13, 2017.

[8] A. Nakabayashi, T. Itoh, “ A Technique for Se- lection and Drawing of Scatterplots for Multi-Dimensional Data Visualization,” Proceedings of 23rd International Conference on Information Visualisation (IV2019), pp. 62–67, 2019.