A Visualization System for Performance Analysis of Image Classification Models

Chanhee Park, Ajou University, Suwon, Republic of Korea
Hyojin Kim, Lawrence Livermore National Laboratory, Livermore, CA
Kyungwon Lee, Ajou University, Suwon, Republic of Korea

Abstract

Developing machine learning models for image classification problems involves various tasks such as model selection, layer design, and hyperparameter tuning for improving the model performance. However, regarding deep learning models, insufficient model interpretability renders it infeasible to understand how they make predictions. To facilitate model interpretation, performance analysis at the class and instance levels with model visualization is essential. We herein present an interactive visual analytics system to provide a wide range of performance evaluations of different machine learning models for image classification. The proposed system aims to overcome challenges by providing visual performance analysis at different levels and visualizing misclassification instances. The system which comprises five views - ranking, projection, matrix, and instance list views, enables the comparison and analysis different models through user interaction. Several use cases of the proposed system are described and the application of the system based on MNIST data is explained. Our demo app is available at https://chanhee13p.github.io/VisMlic/.

Introduction

Image classification refers to the problem of predicting classes from images in computer vision. Recent data-driven models for image classification requires machine learning practitioners to perform various tasks such as collecting data, designing a model, tuning hyperparameters, and comparing the model against different classification algorithms. To facilitate these tasks and maximize the model performance, model interpretability is critical. Recently, deep-learning-based approaches have progressed significantly in image classification tasks. Despite their admirable success, however, insufficient model interpretability in these deep-learning-based approaches renders image classification difficult. Thus, suitable performance analysis tools with interactive, visual interpretation are desired.

Designing and finding an optimal machine learning algorithm for image classification requires various tasks to achieve the desired level of performance.[2]. First, the most appropriate classification algorithm must be selected. Several types of algorithms are available for solving image classification problems[3]. Performance metrics such as accuracy, recall, precision, and f1 score are the main criteria for selecting a model. Next, model architecture and hyperparameters are adjusted to improve the model performance. A practitioner determines the number of layers and neurons when designing the model. Furthermore, various hyperparameters are used to develop and learn models, such as epoch, batch size, and activation functions[4]. Subsequently, when the model performs poorly (e.g., overfitting), the practitioner collects or generates additional data samples. Data augmentation is typically used to generate additional data samples to be learned[5]. All of these processes aim to improve the model performance. Several recent studies have analyzed and compared the effectiveness of each process based on the model type[6] and hyperparameter values[7]. In general, evaluation metrics are used to quantify and compare the performances of models. However, performance analysis becomes difficult when certain metrics have similar values in different models. Furthermore, the metrics do not provide the model vulnerability and the insight to fix it[12]. Therefore, visual performance analysis tools are required to overcome the limitations of metrics.

Although several visual performance analysis methods have been proposed[9, 10, 11, 12, 15], comparing performances at the class level and identifying patterns in misclassified instances remain a challenge. Insufficient performance analysis and comparison at the class level renders model design difficult. Furthermore, without knowing the patterns of the misclassified instances, it is difficult for practitioners to understand how a model predicts. Therefore, to understand and fix defects in a model, the performance analysis should provide class-level analysis and show patterns in error instances.

The most typical approach for evaluating classification performance is using metrics such as precision, recall, f-measure, and accuracy. These metrics enables the overall model performance to be verified quickly. However, the abovementioned metrics are not suitable for analyzing the model performance at the class and instance levels. For example, they are not effective for comparing two models of similar overall performance but different class-specific performances. Another method to analyze the model performance is by using a confusion matrix[8]. The latter shows the number of misclassifications of a classifier by the predicted and actual values in each row and column. Furthermore, it provides class-wise model performance. However, the confusion matrix does not provide a pattern of misclassified instances. Furthermore, it is not suitable for evaluating performance differences with different algorithms and/or a different set of hyperparameters.

We herein present a new visual analytics system to evaluate performance at the class and instance levels to effectively compare multiple models as well as to understand individual

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-PROC-791802.

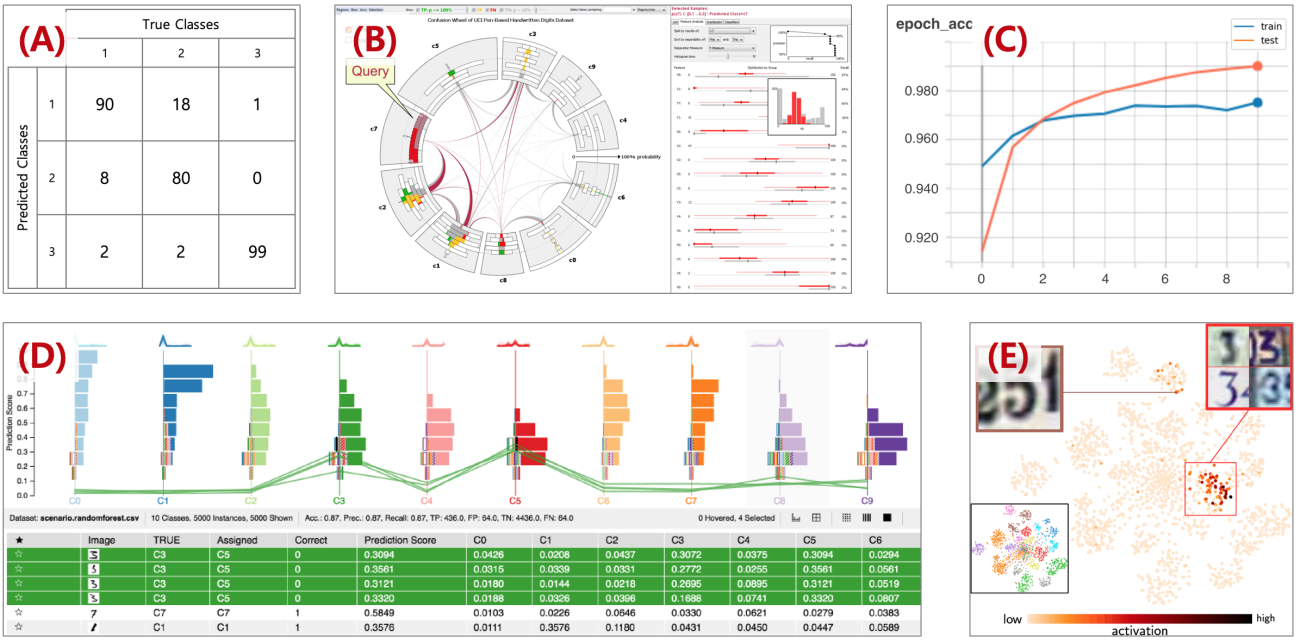

Figure 1. *Techniques for visualizing performance of classification models: (A) a simple confusion matrix, (B) confusion wheel[10], (C) a line graph to indicate the change in accuracy per number of epochs visualized by Tensorboard[14], (D) Squares[11], (E) t-sne dimension reduction for classification results[13].*

model behaviors. The proposed system is an interactive visualization system for image classification models that (1) compares the ranking of several image classification algorithms in the class level, (2) shows the effects of hyperparameters in the class level, and (3) visualizes the patterns of misclassified instances in the confusion level. The proposed system visualizes performance rankings at the class level to evaluate differences in algorithms and hyperparameters. Furthermore, the proposed system explores the misclassification patterns of each classifier through the confusion matrix in addition to displaying representative values. Despite the availability of several performance visualization systems[9, 10, 11, 12, 15], to the best of our knowledge, Our system is the first interactive systems that visually ranks several algorithms and hyperparameter values in the class level. In addition, to provide actionable insights for model improvement, the system provides misclassified instances and performance analysis simultaneously. This system enables one to understand and analyze the strengths and weaknesses of the models.

Related Studies

Performance analysis and visual interpretation are useful for the design, analysis, and improvement of a classification model. Hence, various visual methods have been proposed. The confusion matrix[8] is traditionally used to visualize the performance of a classification model (Fig. 1A). This approach focuses on summarizing the number of misclassification instances by placing actual and predicted classes in the rows and columns of the matrix. Although the confusion matrix is an effective and intuitive tool for demonstrating the performance of a classification model, it presents some limitations: It cannot be used to compare the performances of multiple models and to detect the misclassification pattern. Squares[11] is a visualization tool that effectively identifies class-level performances (Fig. 1D). It uses less space and provides a faster identification performance than the confusion matrix. Because it uses less space, it can place the performances of multiple models on one screen. However, as it is not a visualization tool to compare multiple models, compare three or more models simultaneously using this tool is difficult. Moreover, it does not provide the patterns of misclassified instances. The confusion wheel[10] is a system that visualizes the patterns of misclassified instances (Fig. 1B). This system compares the feature distributions of misclassified and correctly classified instances at the class level. However, it does not provide model interpretability and only provides the feature values of misclassification instances. This tool does not provide a visualization clue to determine which process of the model development, such as layer change or hyperparameter adjustment, contributed to the model's weaknesses. Furthermore, these methods cannot be applied to image classification problems because the image datasets cannot be compared at the feature level (pixel level). TensorBoard[14] is an interactive machine learning system that provides the visualizations required in each process of model development. This system tracks and visualizes changes in metrics such as loss and accuracy according to hyperparameters and model architecture (Fig. 1C). Moreover, it facilitates in understanding and debugging the model by displaying images, text, or audio data. However, it cannot easily compare the performances of two or more models simultaneously. Therefore, a major challenge is the effective design of a visualization system that provides performance comparisons at the class level and patterns of misclassifications at the instance-level.

According to a recent study[11], machine learning practitioners consider model prioritization at the class level as an important but the most difficult task. Therefore, the performance analysis system should assist in prioritization at the class level. When comparing multiple models, it is necessary to provide not only the measures of model performance, but also their ranks. In

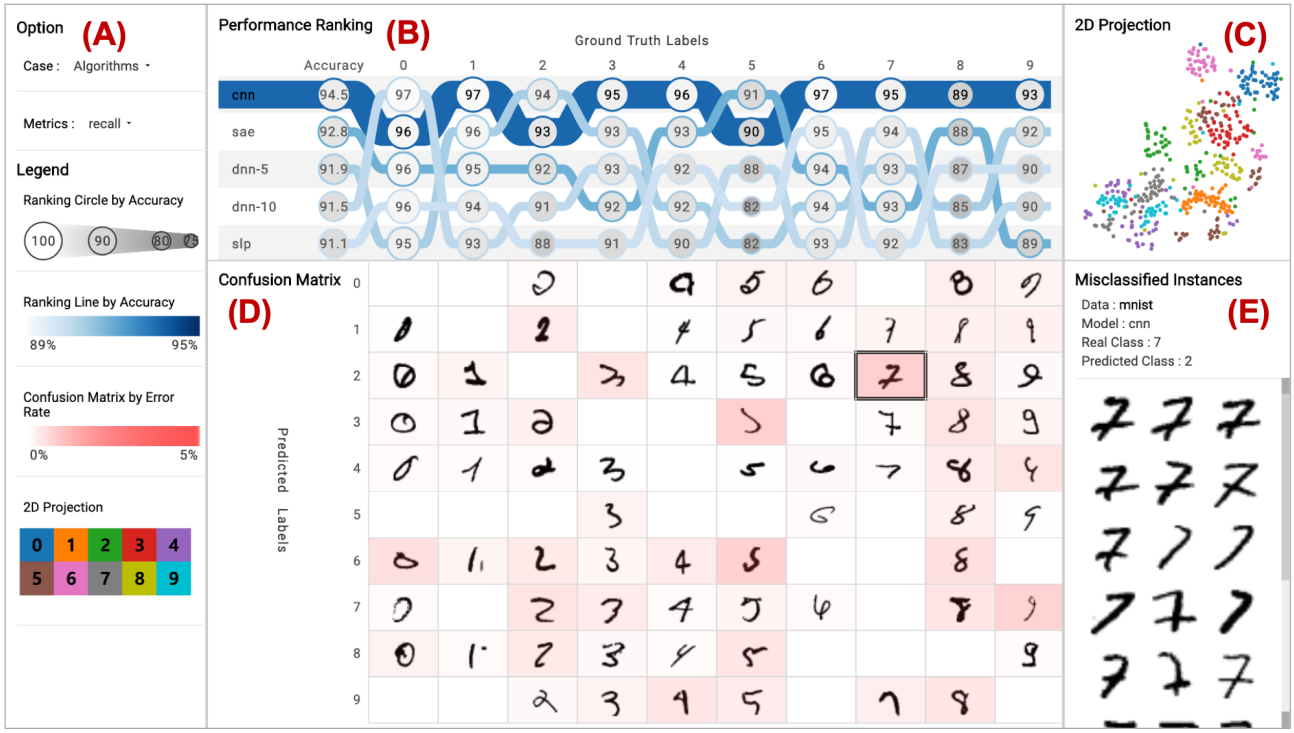

Figure 2. *A proposed visual analysis system for evaluating image classification models. The web-based interactive exploration system provides information regarding performance, weakness, and insights for improvement of image classification models. This system comprises five views: (A) Options and legends view (B) performance ranking view, (C) two-dimensional projection view, (D) confusion matrix view, and (E) instance list view. The integration of the five views enables the performance comparison of different models, investigation of vulnerabilities of each model, and understanding of misclassification patterns of the model.*

this study, we address this challenge by introducing performance ranking visualizations. ActiVis[12], a recent study, reports that practitioners wish to identify model operating patterns based on instances and subsets to improve model performance. Our proposed system investigates the pattern of misclassified instances through class and between-class confusion subsets. Displaying the pattern of misclassified instances, our system allows machine learning practitioners to deduce the misclassification cause. It provides actionable insights to improve models, such as sensing if a model is overfitting and identifying data samples for further training to improve performance.

Visualization Methodology

Design Goal

Shows performance ranking at the class level in each process of model development. Traditional methods of performance analysis vary depending on the development task. When a machine learning algorithm needs to change, a table format summarizing performance metrics of different algorithms is generally used. To evaluate the choice of hyperparameters such as the number of epochs, a curve graph showing the loss at each epoch is widely used. While these methods provide the overall performance of each model, they lack an in-depth comparative analysis of multiple models. To address this issue, we propose a suite of interactive visualization tools to evaluate performance at the class and instance levels to effectively compare multiple models as well as to understand individual model behaviors. Furthermore, we apply the visualization to different processes and scenarios including the change of algorithms, hyperparameters, and model structures.

Investigation of vulnerabilities in a model at the class and confusion levels. Machine learning practitioners select the most appropriate model through performance ranking. However, although the overall performance of the model is sufficient, it exhibits weaknesses in some classes. We need to observe the model behavior with the individual classes or instances. Therefore, the model vulnerabilities should be identified in the performance analysis system. The visual analysis system should display the amount and pattern of the misclassifications in the class and confusion levels. These will provide actionable insights for model improvement such as detecting overfitting at the class level and identifying the characteristics of data that require additional collection. The proposed system takes advantage of multiple views to the data which is meaningful for the different visualization goals to be achieved.

Visualization Preparation

This section describes the input format required for the proposed visualization system. Since the proposed system visualizes performance of multiple models and their class- and instance-level behaviors, both image dataset and model performance data are needed. The image dataset is a set of image files used for testing the models, which includes the image file indices, the filenames and their ground truth labels. The model performance data

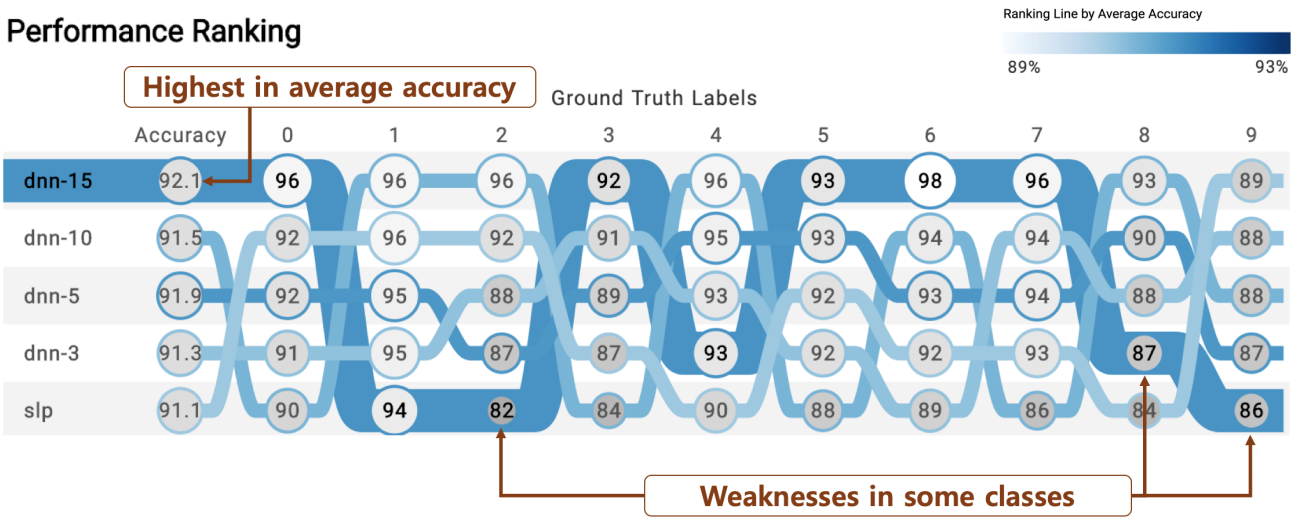

Figure 3. *This ranking visualization shows the change in model performance based on the number of layers. From this visualization, we easily observe that the 15-layer model tends to misclassify instances in some classes while this model yields the highest overall accuracy.*

includes the following information: the model description such as name, layer structure and hyperparameter setting; and the model checkpoint data including the labels that the model predicted. The proposed system provides instance-level performance evaluation by showing misclassified instances in the reverse order of the prediction confidence. To use this function, the test dataset should be provided accordingly. The image file name should follow the format $n(m).jpg$ where n and m are the ground truth label and the index of the image array, respectively. For example, if the image filename is "3(27).jpg", it is the 27th image and the ground truth label is 3. Table 1 shows an example of the input data structure. The input data for the system can be stored as a csv or json format.

Visualization and Interaction

The proposed visualization system comprises five views. 1) Options and legends view for showing options, such as selecting metrics to rank, and legends such as color scales. 2) A ranking view for comparing the performance of multiple classifiers. 3) A projection view of the latent space, a hidden feature representation in each model. 4) A matrix view for diagnosing the weaknesses of each classifier. 5) An instance list view for showing how the model predicts at the instance level. The integration of these five views facilitates machine learning practitioners to effectively compare the performances of models at the class level and investigate weaknesses at the confusion level. The interactive visual diagnostics that the proposed system offers allows us to select the best algorithms for resolving specific classification problems, observing the weaknesses and strengths of each model, and identifying the characteristics of misclassified instances. The proposed system is used in the order of selecting a model from the ranking view, followed by selecting a confusion case from the confusion matrix view.

Table 1: Preparation needed for the provided system

Key	Example Values
Model Name	"CNN", "DNN", "SVM"
Ground Truth Labels	[0,0,0, 1,1,1, 2,2,2]
Predicted Labels	[0,0,0, 1,1,1, 2,2,2]
Testset Filenames	['0(1).jpg', '0(2).jpg', ... 2(2).jpg, '2(3).jpg']

The overall procedure of the proposed system is as follows: (1) the system begins with the performance ranking view [Fig. 2B]. Each model can be selected by a mouse click or a mouseover on ranking line. One model can be selected by a click or a mouseover on the ranking line. (2) When a model is selected, the system shows the two-dimensional projection [Fig. 2C] and confusion matrix [Fig. 2D] of the selected model. A specific case of confusion, a combination of a ground truth class and predicted class, can be selected on the confusion matrix. (3) When a particular confusion case is selected, misclassified instances [Fig. 2E] are then displayed in the bottom right section.

Performance Ranking View

The performance ranking view enables fast but sophisticated performance evaluations of different models. This view allows practitioners to evaluate class-level performances by ranking visualizations. This ranking visualization represents both the relative and absolute performances of each model by showing the performance ranking of different models based on the performance metrics at the class level.

In this visualization, each model has one ranking line. This ranking line displays the performance rankings for each class. Each column and row represent the ground truth class and model, respectively. User interaction rearranges the ranking lines through either class-level precision or recall by classes. If the ranking line of a model is shown as high rank in all classes, it implies that the model is stable and exhibits high performance in all classes. This indicates that the model is superior to all other models in all classes. The relative performance (ranking) by each class is determined by the y-axis position of the ranking line. Furthermore, each ranking line has a different saturation of blue depending on the overall accuracy (absolute comparison) or ranking (relative

comparison). Additionally, user interaction can change this saturation to display the performance ranking, not the average performance. The saturation of the model ranking line and the change in its ranking by class facilitate in identifying low-ranked classes of high-performance models. Therefore, the weaknesses of each model can be represented effectively.

In addition, this ranking visualization shows both the relative (ranking) and absolute performances (metric values). Each model has a circle that shows performance by class. In the center of the circle, the metric value is written. Numbers inside the circle of each ranking line represents performance by class. Furthermore, the size and brightness of the circle are proportional to the performance metrics of a particular class. If the model yields higher performance in the class, a brighter-colored large circle will be shown. If the model yields lower performance in another class, on the other hand, a small circle with a darker color will be shown. The size and brightness of these circles represent the strengths and weaknesses of each model.

This performance ranking visualization is applicable to performance evaluation throughout the development of the machine learning model. This visualization explores how changes in classification algorithms, the number of layers, the loss function and the optimizer affect model performance at the class level. Therefore, this visualization guides us to select suitable algorithms, tune the hyperparameters, and identify the weaknesses of the model.

Two-Dimensional Projection View

The ranking view is effective for comparing the priorities of several models. However, it is ineffective for verifying that each individual model separates each class. Hence, the proposed system visualizes the latent space learned by the model. To obtain the latent space of the model, the hidden representation of the output layer is projected into a two-dimensional space by t-sne[16]. Depending on the features learned by the model, the instances may be located in a two-dimensional latent space. Each instance is represented as a point. To determine if each class is separated well, different colors are assigned to instance points. This allows the model to be verified to ensure that each class is well separated quickly and intuitively. If an instance is evenly distributed by a class in a two-dimensional space, it suggests that the model does not separate classes clearly. However, if each class is clearly separated and the same class is densely packed, it is inferred that the model has learned the features to separate classes well. If instances of two or more classes are located in a similar area, the weaknesses of the model are clearly identified. This is because when instances of different classes are located in similar locations, the model is more likely to misclassify the instances of those classes. As such, machine learning practitioners are investigating the weaknesses of the model and how performance varies with changes, including classification algorithms, structures, and hyperparameters.

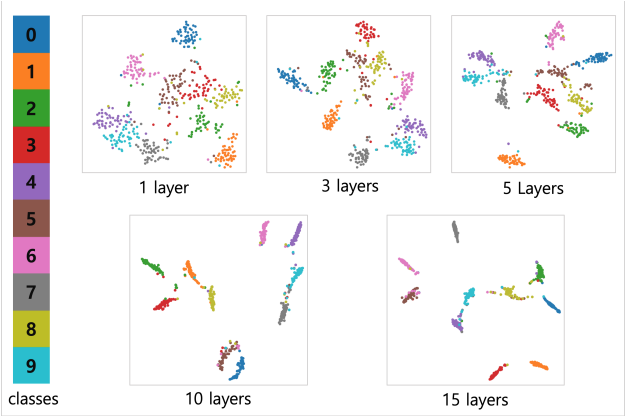

Figure 4. *Two-dimensional projection views by the number of layers in the neural network model. As the number of layers increases, the model tends to learn better discriminative feature representation across the classes. However, when the number of layers is 15, the confusion between classes 5 and 6 increases.*

Confusion Matrix View

The ranking view summarizes the performance of multiple models and identifies classes that are not predicting well by each model. However, analyzing individual models is insufficient. Our confusion matrix view is useful for exploring vulnerabilities in individual models at the class and instance levels of confusion and for identifying misclassified representative instances. This view visualizes the representative instances of the basic confusion matrix. Columns and rows correspond to actual and predicted classes, respectively. Each cell shows misclassifications by the number of instances at the confusion level. The number of misclassified instances is expressed as saturation. If a cell displays a highly saturated red, it indicates that a higher number of misclassified instances occur at that combination of actual and predicted classes. Through this visualization, the weaknesses of model are identified at the class and confusion levels through this visualization, which resembles a heat map. In addition, this view displays the representative instance of each cell. In other words, a representative misclassified instance is placed in each cell of the matrix. Most recent deep learning algorithms such as convolutional neural networks can be used to construct a probabilistic classification model. Therefore, the representative instance of each cell is an instance of high confidence at the predicted class of that cell. If multiple instances with the same predictive confidence value exist, only one is selected randomly. Similarly, if the model does not perform probabilistic classification, an instance corresponding to each cell is randomly selected. Showing these representative instances provides a coarse understanding of the context that results in each misclassification.

Misclassified Instances View.

Only one representative instance shown in the matrix view is insufficient to understand the pattern of misclassified instances. To this end, this view shows all image instances that satisfy the selected condition of a model, actual class and, predicted class. The misclassification condition can be selected with a click in the matrix view. Instances are sorted from top-left to bottom-right, based on the prediction confidence in the misclassification class. This view enables us to easily capture instances that are more likely to be misclassified. Misclassification patterns can be investigation with this view. In addition, by finding the common ground between misclassified instances, the misclassification pattern can be obtained. Obtaining these patterns not only provides an understanding of the model, but also provides actionable insights for improving the model performance. The misclassified instance

view facilitates machine learning practitioners in identifying the cause of misclassification of a model. Moreover, it shows additional instances that need to be collected or augmented.

Usage Scenarios

We used the MNIST database[17] to introduce the usage and effectiveness of our proposed system. The MNIST database, which contains handwritten digits, is widely used to demonstrate image classification problems. The MNIST problem involves training models that recognize handwritten digits to classes between 0 and 9. In this section, we illustrate a usage scenario to demonstrate the effectiveness of the system in designing classification models. In this section, we describe three usage scenarios of the proposed system: First, an appropriate model is selected by comparing the ranking of several image classification algorithms at the class level; next, hyperparameters are tuned by the effects of the hyperparameters at the class level; finally, vulnerabilities in the model are identified through the visualized pattern of the misclassified instances at the confusion level.

Select the most appropriate classifier.

Selecting an appropriate model is a main task in machine learning. Once the problem is defined and data collected, practitioners must decide which model to use. A typical approach to select a model is to develop several models and select the most appropriate among them. When the performance results are significantly different between several models, comparison becomes easy. However, when the performances are similar, comparison and model selection become difficult. Our proposed system can be used for difficult cases. [Fig. 2B] shows a comparison of five models with different algorithms but similar overall accuracy. This ranking line shows that the model "cnn" not only has a higher average performance, but also a more stable prediction for the entire class compared with other models. However, the ranking circle indicates that the model performs poorly in "class 5" than the model predicts instances in other classes. Therefore, performance ranking visualization [Fig. 2B] shows the necessity to select the "cnn" model and to ameliorate its weakness, "class 5." Furthermore, as shown in the matrix view [Fig. 2D], the cells in the column representing the "ground truth label of 5" and the row representing the "predicted labels are 2, 3, and 6" are highly saturated red. It appears that the weaknesses of "cnn" are misclassifying 5 as 2, 5 as 3, and 7 as 6. The misclassification instances are shown in the instance view. It allows machine learning practitioners to identify the characteristics of data to be generated or collected additionally.

In deep learning, the model performance depends on the model structure such as the numbers of layers or neurons. For example, if the numbers of neurons and layers are small, the classifier cannot obtain the appropriate features to classify the image. However, if the numbers of neurons and layers are large, the model may suffer from overfitting or fail to converge. Therefore, to improve the performance of the classifier, using appropriate parameter settings are crucial. The proposed system can be used to determine the appropriate parameters. [Fig. 3] shows the change in model performance depending on the number of layers. A model with 15 layers yields the highest overall accuracy but misclassifies instances of "class 2," "8," and "9." This suggests that the 15-layer model has the highest overall performance but exhibit weaknesses in some classes. Meanwhile, a 5-layer model is stable in all classes, although the difference in overall accuracy is subtle. These findings facilitate in determining the optimum number of layers.

Find proper hyperparameters

Even with the same deep learning algorithm, the model performance depends on the selection of hyperparameters such as number of epochs, batch size, loss function, and optimizer. Therefore, the hyperparameters should be tuned to maximize the performance. Our proposed system aids in optimizing the hyperparameters. [Fig. 5] shows the difference in model performance depending on the number of epochs. In this figure, each color of the ranking lines is proportional to the rank. According to this visualization, the performance of the deep neural network model tends to improve as the number of epochs increases. However, when the number of epochs reaches 30, the performance degrades in "class 6," indicating that the model may suffer from overfitting when the number of layers becomes 30. In addition, when the number of epochs is 20, the overall accuracy is 0.8% less than that of the 30-epochs model but stable in the entire class. This shows that if the performance of the model should be stable for the entire class, it is recommended to adjust the number of epochs to 20 than 30. In other words, the proposed system exhibits the strengths and weaknesses of the model depending on the hyperparameter setting. Through the visualization offered by the system, machine learning practitioners are able to not only adjust the hyperparameters optimally, but also understand the limitation of each model.

Identify patterns of the misclassified instances

A method to reduce the misclassification of the model is to investigate the tendencies of misclassified instances. It is difficult to detect trends from analyzing the statistics of misclassified instances because image data are difficult to analyze at the feature level (pixel level). On the other hand, humans see an image as an object, not as a collection of pixels. Therefore, merely referring to misclassified instances in the order of importance aids in identifying misclassification tendencies. The ranking line shown in [Fig. 2B] indicates that "cnn" is most vulnerable in "class 5." Furthermore, [Fig. 2C] shows that "cnn" tends to frequently misclassify "5" as "6" and "3." To improve the vulnerability of this model, the pattern for cases where class "5" is misclassified as "3" and "6" must be identified. The patterns can be identified by the misclassified instance view. [Fig. 6] shows the instances in those confusion cases. The list on the left in [Fig. 6] show instances of "5" misclassified as "3". These instances tend to exhibit a short horizontal straight line at the top of the number 5. Furthermore, the list on the right in [Fig. 6] shows instances of "5" misclassified as "6." In these instances, curves at the bottom of the number 5 are often closed. These lists allow machine learning practitioners to understand the model behavior (i.e., how the model classifies a particular image instance). In this case, it appears that the "cnn" model views the horizontal line at the top of the image and the open curve at the bottom of the image when it classifies an image as class "5." A machine learning practitioner, who deduced a model's prediction pattern, can determine the data that should be collected or augmented to improve the performance. Therefore, visualizing a list of misclassification instances allows machine learning practitioners to obtain patterns of instances that

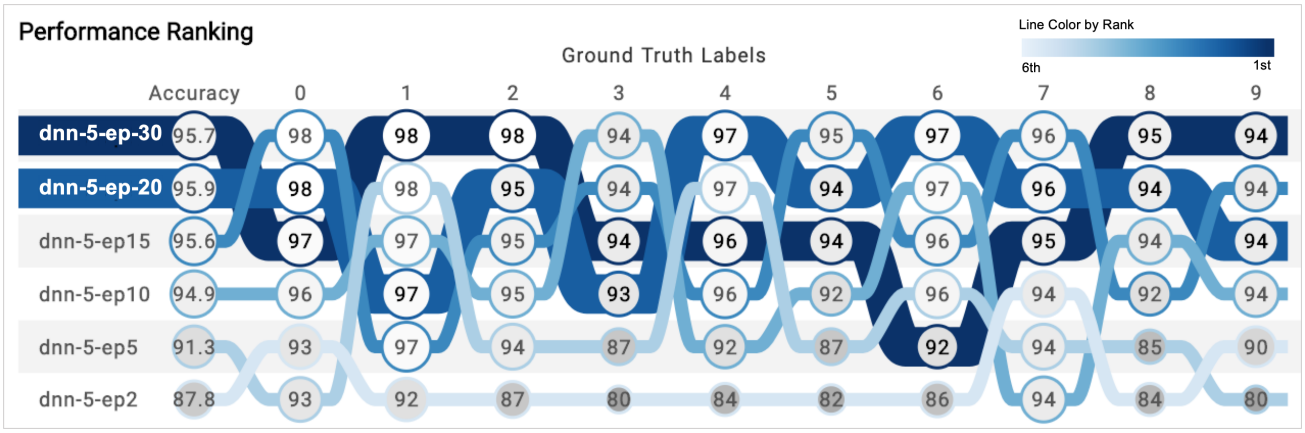

Figure 5. *Ranking visualization to perform comparison with changes in number of epochs. In this visualization, each case exhibits a color proportional to the ranking of the overall accuracy.*

the model misclassifies and understand how the model classifies a particular image of interest.

Conclusions

We presented an interactive performance visualization system for comparing and analyzing image classification models. This visualization compared the performances of multiple classifiers at the class level. Additionally, it displayed misclassification patterns at the confusion level. This system could assist machine learning practitioners to select the appropriate model type, design the model structure, adjust the hyperparameters, and investigate patterns of data that must be further collected. Recently, several studies regarding models that can classify more than 100 classes have emerged in computer vision research. For future work, we will address this problem, having highly many kinds of classes, through class alignment and grouping. Additionally, we will develop a visualization system that is applicable to hundreds of classes of classification problems. Finally, we will further develop the system such that it becomes a visual interactive machine learning tool to modify and learn models in real-time.

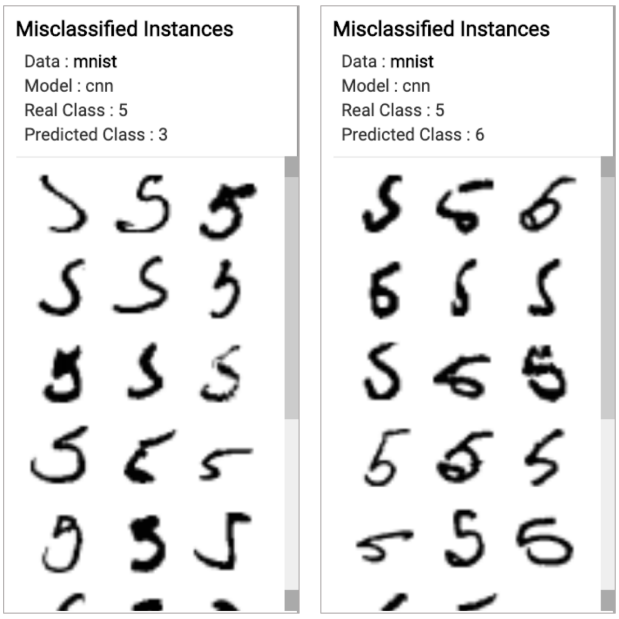

Figure 6. *Two instance list views. These visualizations show that "cnn" model misclassifies class "5" as "3"(left) and "6" (right).*

References

[1] Chanhee Park, Jina Lee, Hyunwoo Han, Kyungwon Lee: ComDia+: An Interactive Visual Analytics System for Comparing, Diagnosing, and Improving Multiclass Classifiers. The 12th IEEE Pacific Visualization Symposium (2019).

[2] Lu, Dengsheng and Qihao Weng: A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing 28(5), pp. 823-870 (2007).

[3] Tanmoy Das: Machine Learning algorithms for Image Classification of hand digits and face recognition dataset. International Research Journal of Engineering and Technology, 4(12), pp. 640-649 (2017).

[4] Dong, X., Shen, J., Wang, W., Liu, Y., Shao, L. and Porikli, F.: Hyperparameter optimization for tracking with continuous deep q-learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 518-527 (2018).

[5] Perez, L., and Wang, J.: The effectiveness of data augmentation in image classification using deep learning. Convolutional Neural Networks Vis. Recognit (2017).

[6] Lin, Y. and Le Kernec, J: Performance analysis of classification algorithms for activity recognition using micro-Doppler feature. In 2017 13th IEEE International Conference on Computational Intelligence and Security (CIS), pp. 480-483 (2017).

[7] Schmidt, M., Safarani, S., Gastinger, J., Jacobs, T., Nicolas, S. and Schülke, A.: On the Performance of Differential Evolution for Hyperparameter Tuning. arXiv preprint arXiv:1904.06960 (2019).

[8] Townsend, James T.: Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics, 9(1), Springer, pp. 40-50 (1971).

[9] Amershi, S., Chickering, M., Drucker, S. M., Lee, B., Simard, P., Suh, J. : Modeltracker: Redesigning performance analysis tools for machine learning. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 337-346 (2015).

[10] Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S., and Rauber,

A.: Visual methods for analyzing probabilistic classification data. IEEE transactions on visualization and computer graphics, 20(12), pp. 1703-1712 (2016).

[11] Ren, D., Amershi, S., Lee, B., Suh, J., and Williams, J. D.: Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE transactions on visualization and computer graphics, 23(1), pp. 61-70 (2016).

[12] Kahng, M., Andrews, P. Y., Kalro, A., and Chau, D. H. P.: ActiVis: Visual exploration of industry-scale deep neural network models. IEEE transactions on visualization and computer graphics, 24(1), pp. 88-97 (2017).

[13] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. and C. Telea.: Visualizing the hidden activity of artificial neural networks. IEEE Transactions on Visualization and Computer Graphics, 23(1), pp. 101–110, (2017).

[14] Wongsuphasawat, K., Smilkov, D., Wexler, J., Wilson, J., Mane, D., Fritz, D., ... and Wattenberg, M.: Visualizing dataflow graphs of deep learning models in tensorflow. IEEE transactions on visualization and computer graphics, 24(1), pp. 1-12 (2017).

[15] Chaudhuri, Abon: A Visual Technique to Analyze Flow of Information in a Machine Learning System. Electronic Imaging (2018).

[16] Maaten, L. V. D., and Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research, pp. 2579-2605 (2008).

[17] Deng, L.: The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6), pp. 141-142 (2012).

Disclaimer

Author Biography

Chanhee Park received his BS in digital media from the Ajou University (2019). He is now in the master's program at the same school. He's been doing research in the field of information visualization and machine learning, and he's interested in the kind of explainable artificial intelligence(XAI).

Hyojin Kim is a computer scientist at the Lawrence Livermore National Laboratory. His research interests are broadly computer vision, machine learning, data analysis and visualization. He earned a Ph.D. in Computer Science at the University of California, Davis in 2012.

Kyungwon Lee received the MFA degree in computer graphics and interactive media from the Pratt Institute in 2002. He is a professor in the Department of Digital Media and the director of Integrated Design Lab at Ajou University. His research interests include information visualization, human-computer interaction, and media art. He is a member of the IEEE.