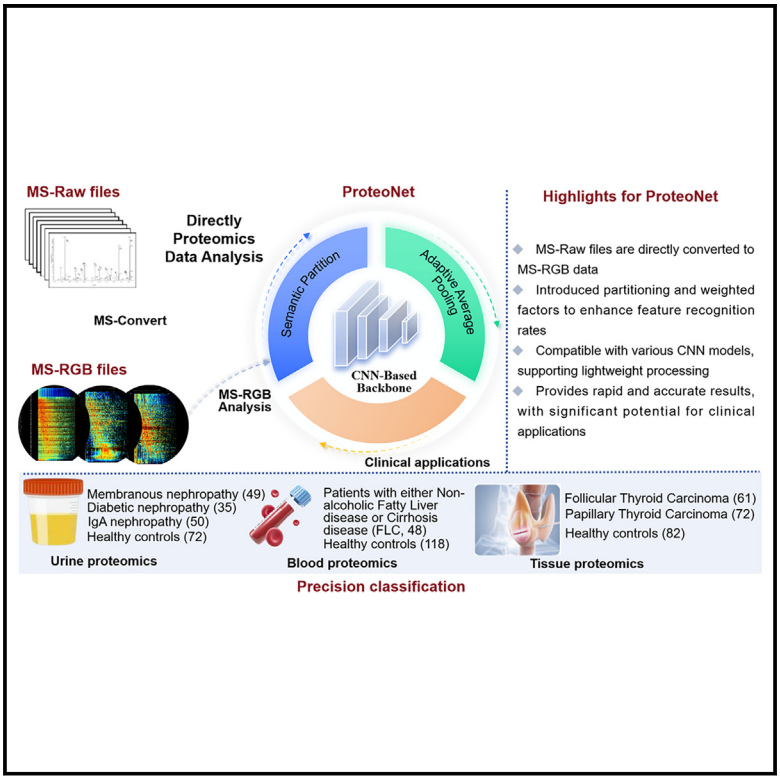
iScience

Article

ProteoNet: A CNN-based framework for analyzing proteomics MS-RGB images

Graphical abstract



Highlights

- A deep learning framework for analyzing MS-RGB data in clinical proteomics
- MS-RGB analysis with semantic partitioning, adaptive pooling, and weighted factors
- ProteoNet excels in speed and accuracy for analyzing proteomics across diseases

Authors

Jinze Huang, Yimin Li, Bo Meng, ..., Dong An, Yang Zhao, Xiang Fang

Correspondence

daixh@nim.ac.cn (X.D.), andong@cau.edu.cn (D.A.), zhaoy@nim.ac.cn (Y.Z.), fangxiang@nim.ac.cn (X.F.)

In brief

Computer-aided diagnosis method; Proteomics; Machine learning

Huang et al., 2024, iScience 27, 111362
December 20, 2024
https://doi.org/10.1016/j.isci.2024.111362

CellPress

iScience

CellPress
OPEN ACCESS

Article

ProteoNet: A CNN-based framework for analyzing proteomics MS-RGB images

Jinze Huang,[1,4] Yimin Li,[2,4] Bo Meng,[1] Yong Zhang,[3] Yaoguang Wei,[2] Xinhua Dai,[1,*] Dong An,[2,*] Yang Zhao,[1,2,5,*] and Xiang Fang[1,*]

[1]Technology Innovation Center of Mass Spectrometry for State Market Regulation, Center for Advanced Measurement Science, National Institute of Metrology, Beijing 100029, China
[2]College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[3]Institutes for Systems Genetics, West China Hospital, Sichuan University, Chengdu 610041, China
[4]These authors contributed equally
[5]Lead contact
*Correspondence: daixh@nim.ac.cn (X.D.), andong@cau.edu.cn (D.A.), zhaoy@nim.ac.cn (Y.Z.), fangxiang@nim.ac.cn (X.F.)
https://doi.org/10.1016/j.isci.2024.111362

SUMMARY

Proteomics is crucial in clinical research, yet the clinical application of proteomic data remains challenging. Transforming proteomic mass spectrometry (MS) data into red, green, and blue color (MS-RGB) image formats and applying deep learning (DL) techniques has shown great potential to enhance analysis efficiency. However, current DL models often fail to extract subtle, crucial features from MS-RGB data. To address this, we developed ProteoNet, a deep learning framework that refines MS-RGB data analysis. ProteoNet incorporates semantic partitioning, adaptive average pooling, and weighted factors into the Convolutional Neural Network (CNN) model, thus enhancing data analysis accuracy. Our experiments with proteomics data from urine, blood, and tissue samples related to liver, kidney, and thyroid diseases demonstrate that ProteoNet outperforms existing models in accuracy. ProteoNet also provides a direct conversion method for MS-RGB data, enabling a seamless workflow. Moreover, its compatibility with various CNN architectures, including lightweight models like MobileNetV2, underscores its scalability and clinical potential.

INTRODUCTION

Mass spectrometry (MS)-based proteomics, which can rapidly profile thousands of proteins from minimal biofluid or tissue samples, has greatly advanced clinical diagnostics and treatment strategies.[1–3] However, the complexity and volume of the resulting proteomic data present substantial analytical challenges. Artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL) models, has emerged as valuable tools to effectively manage and interpret this intricate data.[4–7] For instance, ML models have enabled developments such as the KDClassifier, which achieves over 95% accuracy in distinguishing between various nephropathies 8 and protein-based ML models for classifying thyroid nodules.[9] Nonetheless, applying ML in clinical proteomics faces challenges related to pre- and post-data processing, including transforming MS data, normalizing data, and selecting features. These steps can be labor-intensive, prone to errors, and may lead to biased or biologically irrelevant results.[1] This highlights the necessity of integrating advanced AI techniques into the clinical proteomics for more rapid and accurate diagnosis.

DL is an advanced branch of AI that focuses on constructing deep neural networks with multiple interconnected layers.[10] Unlike ML models, DL frameworks utilize algorithms like gradient descent to automatically adjust internal parameters based on feedback from the loss function.[11] This adaptive approach eliminates the need for manually predefined attributes, thereby simplifying the complexities involved in traditional feature extraction and selection processes.[12] By training the model to autonomously learn and prioritize features from the data, DL not only streamlines the analysis process but also enhances the model's ability to identify and exploit relevant patterns within complex datasets.[13,14] This capability makes DL a particularly promising candidate for clinical proteomics analysis.

Furthermore, recent advancements in DL have shown promising results in clinical proteomics. For example, using pretrained CNN models, researchers have been able to distinguish tumor from normal samples in SWATH-MS (Sequential Window Acquisition of all Theoretical Mass Spectra) data with remarkable accuracy.[15] Additionally, modified ResNet models have successfully identified liver cancer and thyroid nodules from DIA-MS data with high accuracy.[16] These developments highlight DL's potential to learn directly from proteomic MS-RGB images (mass spectrometry data converted to red, green, and blue color images), reducing reliance on complex data processing pipelines. However, proteomic MS-RGB images present the challenges due to their complex data structures and specific noise patterns.[17–19] Tailoring CNN models to effectively interpret these images, while considering their chemical and molecular information, is essential. Although global average pooling layers in CNNs can simplify models, they might overlook subtle variations critical

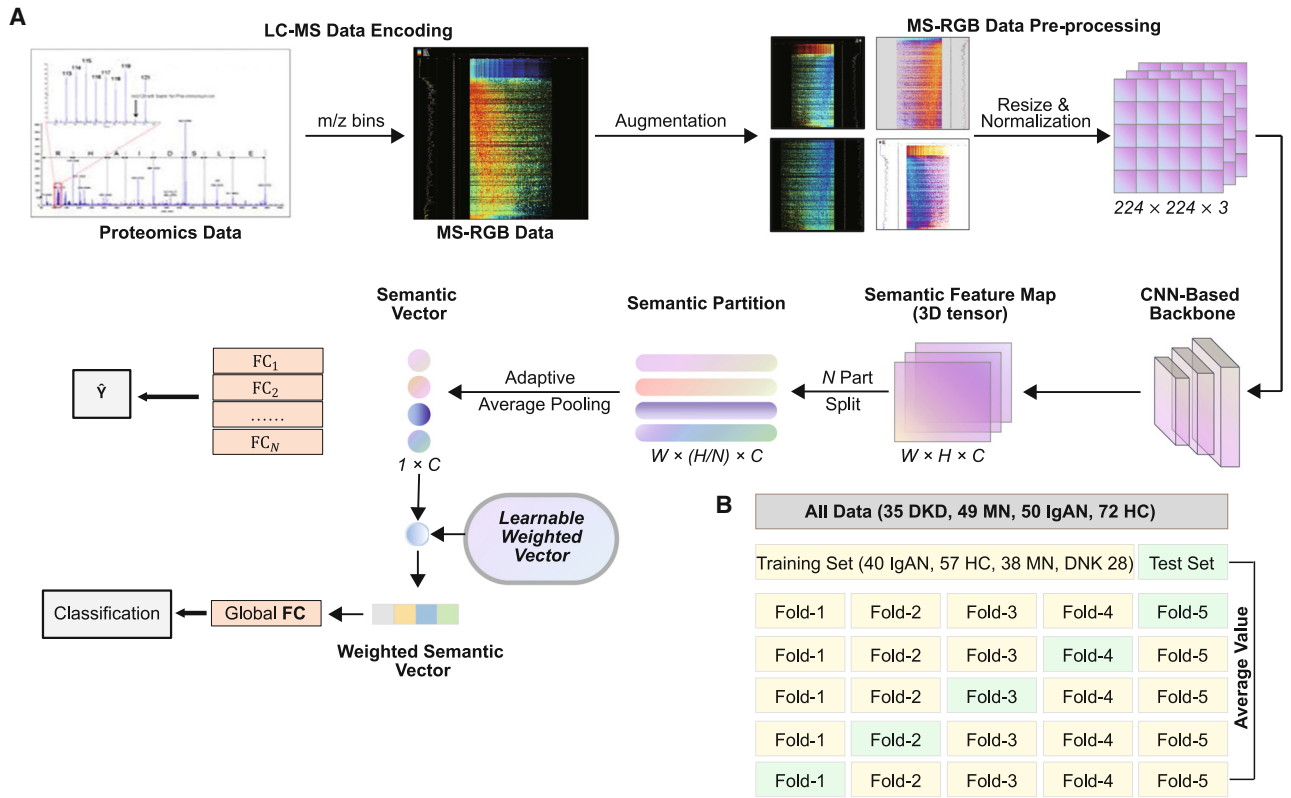


Figure 1. Overview of ProteoNet architecture and experimental setup

(A) The architechure of ProteoNet. In this model, the Convolutional Backbone can be replaced by any CNN-based structure. The output from the Backbone is divided into *N* distinct regions and processed by ***N*** Average pooling for part features. Afterward, part classifiers and weighted-integration are employed for guiding model to learning the feature of proteomic MS-RGB images.

(B) Overview of the experimental procedure used to evaluate the performance of ProteoNet.

in proteomic MS-RGB images.[18] Hence, improving models to capture these details is crucial for accurate analysis.

In response to these challenges, we introduce ProteoNet, a CNN-based framework specifically designed to enhance the analysis of proteomic MS-RGB data. ProteoNet starts with the MSConvert tool,[20] which transforms MS data into MS-RGB images, facilitating better visualization and data augmentation. This framework uses an architecture that operates without a predetermined backbone, focusing instead on extracting semantic feature maps from various data partitions. This design enables the integration of learnable weighted factors throughout the model, optimizing feature extraction and emphasizing discriminative features in highly similar regions of MS-RGB data. ProteoNet's approach not only increases the efficiency of feature extraction but also refines the analysis of high-similarity areas in proteomic MS-RGB images, offering a significant advance in the field of clinical diagnostics.

RESULTS

Effectiveness of ProteoNet framework

The ProteoNet framework comprises three main steps: LC-MS Data Encoding, MS-RGB Data Pre-processing, and Model Training/Inference, as illustrated in Figure 1. The model input consists of augmented MS-RGB data, as shown in Figure 2. Detailed steps can be found in the Star Methods section. To assess the convergence status of a model, it is common to construct graphical representations that track the loss function and validation accuracy throughout training. These visualizations are crucial for monitoring both training and validation phases, providing key insights into the incremental convergence of the model. In our study, we highlight the capability of the ProteoNet model to accurately recognize proteomics MS-RGB images using a singular approach. We employed a 5-fold cross-validation method for a thorough evaluation, paying particular attention to the data from one of these folds to detail the model's effectiveness. This focus enabled a detailed analysis of the model's learning trajectory, evidenced by a decrease in training loss and an increase in validation accuracy in this fold. These trends, as illustrated in Figure 3, indicate that ProteoNet achieved optimal classification efficiency and effectively avoided overfitting or underfitting.

Furthermore, we conducted an extensive evaluation of the ProteoNet framework's diagnostic capabilities across different proteomics datasets, which included urinary proteomics for kidney diseases, blood proteomics for liver diseases, and tissue proteomics for thyroid cancers, corresponding to multi-class classification challenges of four, two, and three classes, respectively. Results, detailed in Table 1, show that on the kidney dataset,

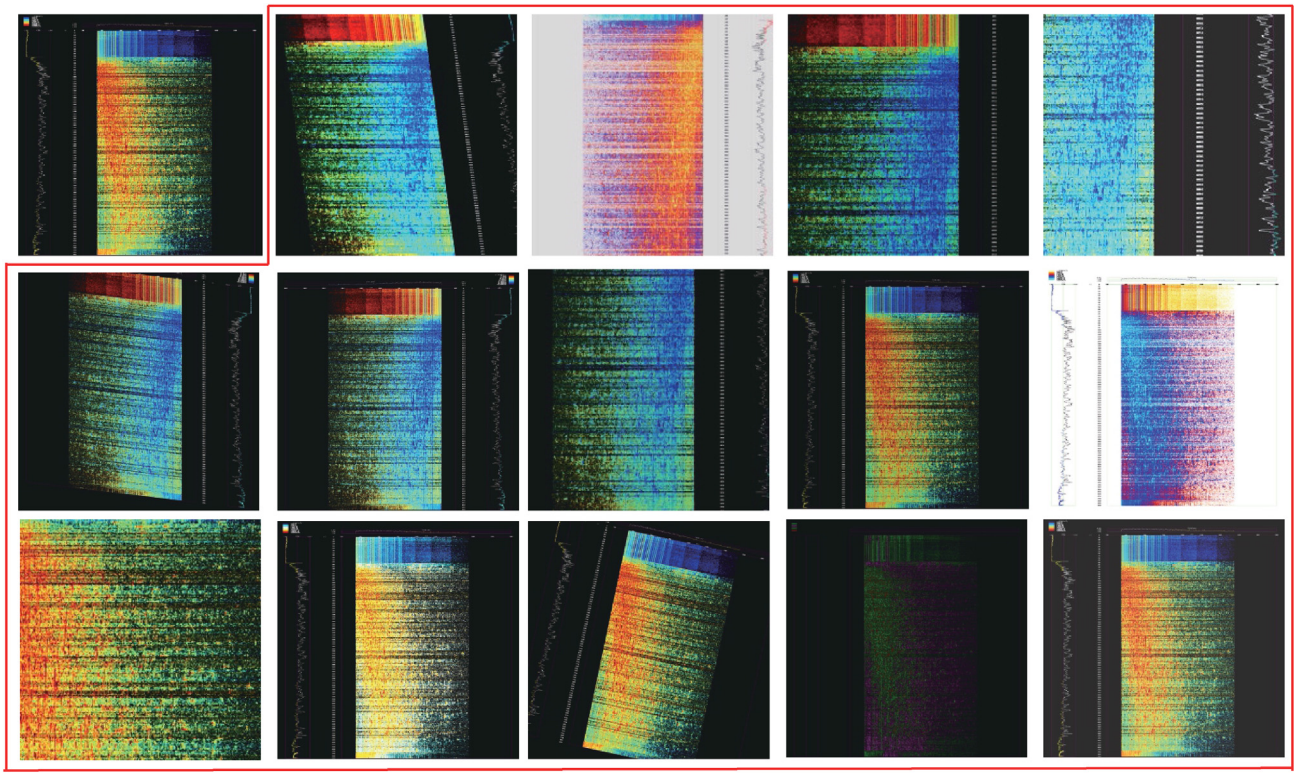


Figure 2. Examples of data augmentation techniques

The original MS-RGB image is shown in the upper-left corner outside the red box, while the images inside the red box illustrate various data augmentation methods applied.

ProteoNet demonstrated high precision in diagnosing healthy control (HC) individuals, diabetic nephropathy (DN) patients, and IgA nephropathy (IgAN) patients, with precision rates exceeding 95%. However, the recall rates for DN and IgAN patients were moderately lower, at 88.1% and 90.0% respectively, pointing to some instances of missed diagnoses. Nonetheless, the F1 scores for all categories were notably high, reaching 91.3%, which underscores ProteoNet's accuracy and reliability in diagnosing a range of kidney-related conditions from MS-RGB data.

Additionally, in the thyroid cancer dataset, ProteoNet achieved more than 90% in precision, recall, and F1 scores for identifying patients with papillary thyroid carcinoma (PTC) (Table 1). However, the recall rate for follicular thyroid carcinoma (FTC) patients was relatively low, suggesting a tendency to misclassify these cases as other disease types. This indicates a potential area for improvement in the ProteoNet model's performance on this specific dataset. In contrast, in the liver disease dataset, ProteoNet performed exceptionally well, with all metrics—precision, recall, and F1 score—surpassing 90% (Table 1). This performance highlights the model's strong capability in binary classification tasks, emphasizing its robustness across different diagnostic settings.

Overall, the results highlight the ProteoNet model's robust diagnostic capabilities across diverse proteomic datasets, showcasing its precision in most cases, yet also pinpointing opportunities for improvement to further optimize its performance.

Comparing with other ML and DL models

In our comparative study using kidney dataset, we evaluated the ProteoNet model against established machine learning models, such as support vector machine (SVM), random forest (RF), K-nearest neighbor (KNN), and XGBoost, as well as deep learning models including ResNet-50 and Vision Transformer (ViT). ProteoNet outperforms these models in accuracy, distinguishing diseases like membranous nephropathy (MN), DN, IgAN, and identifying HC individuals with an accuracy of 95.1% (Table 2). This performance exceeds that of the leading ML model, SVM, which achieved an accuracy of 94.1%, and is significantly higher than ResNet-50 and ViT DL models, which recorded accuracies of 89.3% and 87.8%, respectively.

ProteoNet also demonstrates superior discriminative ability across all four kidney phenotypes, maintaining F1 scores above 90.0%, which significantly outperforms the other models (Table 2). While other models generally performed well in identifying HC individuals, achieving F1 scores up to 100%, their effectiveness declined when distinguishing between DN, MN, and IgAN patients, with F1 scores dropping to around 70.0% in some cases. Moreover, ProteoNet's balanced performance was evident in its precision and recall rates, both exceeding 90.0%, underscoring its efficacy and reliability in diverse diagnostic settings.

The 5-fold cross-validation average accuracies for all models across three datasets are shown in Figure 4, where ProteoNet

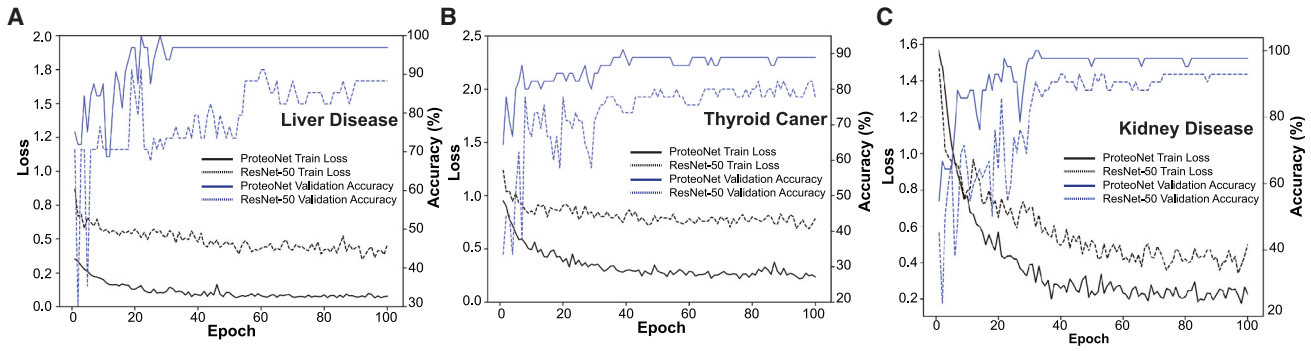


Figure 3. Training loss and validation accuracy trends across liver, thyroid, and kidney datasets

(A) Trends for liver diseases.

(B) Trends for thyroid cancers.

(C) Trends for kidney diseases. A lower training loss (black) indicates better model fitting, while a higher validation accuracy (blue) suggests better generalization. The solid line represents the ProteoNet model, and the dashed line represents the ResNet-50 baseline.

consistently led, with scores of 96.4%, 86.6%, and 95.1% on each dataset, respectively. These results were superior to those achieved by both the traditional ML models and advanced deep learning models. This outstanding performance of ProteoNet can be attributed to its robust feature extraction mechanism, which integrates a CNN-based architecture with techniques like semantic partition, adaptive average pooling, and optimized weighting factors. This configuration enhances its capability to effectively analyze MS-RGB data, contrasting with ML methods that depend heavily on database searching and prior knowledge. ProteoNet, which refines 1D features directly from loss function adjustments and a fully connected layer, is highly effective in clearly defining sample boundaries and accurately classifying disease subtypes. These features make ProteoNet exceptionally capable of managing complex MS-RGB data and effectively addressing challenges related to high regional similarity.

Accuracy evaluation of models

A confusion matrix is a key visualization tool used for assessing the accuracy of the classification models. Our analysis, using the kidney dataset, indicated that all models effectively identified HC individuals from patients with MN, IgAN, and DN diseases (Figure 5). This indicates a strong capability of the models in identifying general kidney disease presence.

However, when it comes to distinguishing between MN, IgAN, and DN patients, ML models such as SVM, RF, KNN, and XGBoost showed limited effectiveness. The accuracy for classifying these specific disease subtypes was around 80% (Figures 5A–5D). The KNN model, in particular, exhibited lower accuracy, about 70%, in differentiating both MN and DN patients (Figure 5D). The ViT model showed around 80% effectiveness in differentiating both DN and MN patient types, slightly outperforming the KNN model but still less effective compared to other models (Figure 5E). In contrast, the adoption of CNN-based DL approaches, such as ResNet-50 and our ProteoNet, has significantly improved predictive accuracy (Figures 5F and 5G). ProteoNet, in particular, exhibited remarkable performance, misclassifying only 4% of IgAN samples as MN patients, while achieving an impressive 100% accuracy in distinguishing other disease subtypes (Figure 5G).

Overall, these results highlight the superior efficacy of CNN-based approaches, particularly in proteomics MS-RGB image analysis, for disease diagnosis compared to traditional ML and transformer methods. Our ProteoNet model stands out with its high precision, especially in differentiating between IgAN and MN, while maintaining excellent overall accuracy for other disease subtypes. This demonstrates the advantages and potential of advanced CNN models like ProteoNet in clinical settings. The enhancements provided by the inclusion of semantic partition, adaptive average pooling, and weighting factors in ProteoNet significantly improve upon the standard ResNet-50 model, showcasing the effectiveness of these modifications.

Table 1. Average accuracy of ProteoNet from 5-fold cross-validation

Dataset	Disease status	Precision	Recall	F1 score
MS-RGB: Kidney disease	HC individuals	98.9%	100%	99.4%
	DN patients	95.8%	88.1%	91.3%
	MN patients	90.9%	98.3%	94.3%
	IgAN patients	95.0%	90.0%	92.3%
MS-RGB: Thyroid cancer	HC individuals	79.5%	85.4%	82.1%
	FTC patients	84.0%	73.7%	75.5%
	PTC patients	93.3%	93.2%	92.9%
MS-RGB: Liver disease	HC individuals	98.4%	96.6%	97.4%
	FLC patients	93.4%	95.6%	93.8%

Visual proof of image feature capture

Deep learning models are often perceived as a "black boxes". To demystify their decision-making processes, we introduced the visualization method of class activation maps (CAM). CAM is a specially designed visual tool that reveals how CNN networks make decisions in classification tasks.[21] It substitutes the fully connected layer at the end of a convolutional neural

Table 2. Average accuracy of all classifier models from 5-fold cross-validation on the kidney dataset

ID	Disease status	Precision	Recall	F1 score	Accuracy
Support vector machine (SVM)	HC individuals	100%	98.6%	99.3%	94.1%
	DN patients	85.8%	82.4%	83.8%	
	MN patients	92.0%	91.3%	91.4%	
	IgAN patients	91.2%	94.0%	92.1%	
Random forest (RF)	HC individuals	100%	94.8%	99.4%	87.4%
	DN patients	77.3%	84.8%	80.2%	
	MN patients	81.6%	71.1%	76.1%	
	IgAN patients	78.6%	87.8%	82.3%	
K-Nearest Neighbor (KNN)	HC individuals	100%	98.2%	99.1%	85.5%
	DN patients	82.6%	65.5%	71.3%	
	MN patients	86.5%	75.0%	78.8%	
	IgAN patients	69.4%	88.7%	78.3%	
XGBoost	HC individuals	100%	100%	100%	88.8%
	DN patients	82.9%	85.3%	82.3%	
	MN patients	82.1%	79.4%	78.5%	
	IgAN patients	87.4%	88.0%	87.0%	
ResNet-50	HC individuals	100%	100%	100%	89.3%
	DN patients	95.0%	80.0%	86.3%	
	MN patients	80.6%	91.8%	85.5%	
	IgAN patients	87.3%	82.0%	84.1%	
Vision Transformer (ViT)	HC individuals	100%	100%	100%	87.8%
	DN patients	88.3%	74.3%	80.1%	
	MN patients	87.8%	79.8%	82.2%	
	IgAN patients	76.3%	88%	80.5%	
ProteoNet	HC individuals	98.9%	100%	99.4%	95.1%
	DN patients	95.8%	88.1%	91.3%	
	MN patients	90.9%	98.3%	94.3%	
	IgAN patients	95.0%	90.0%	92.3%	

network with a fully convolutional layer and projects the weights of the output layer backward onto the convolutional layer features. This enables the effective localization of critical regions within an image that contribute to the classification task. In CAM visualizations, brighter areas represent regions with higher activation levels for a specific class, indicating the important areas the neural network focuses on during classification.

In our analysis, we observed that the baseline ResNet-50 model utilized limited discriminative information when analyzing proteomics MS-RGB data, concentrating primarily on a few key areas (Figure 6A). In contrast, our enhancements to the ResNet-50 backbone in the ProteoNet model significantly increased the number of crucial discriminative regions considered during the analysis (Figure 6B). The ProteoNet model more effectively utilizes information from various parts of the data, leading to a more comprehensive analysis.

Furthermore, we discovered notable differences in the key discriminative regions selected by the ResNet-50 model when processing images of the same disease category, such as HC individuals and IgAN patients (Figure 6A). However, the ProteoNet model demonstrates higher consistency and similarity in the selection of discriminative features for classification. This consistency underlines the model's enhanced ability in distinguishing between different kidney disease subtypes, indicating its effectiveness in MS-RGB data analysis.

The effectiveness of partition number

As described in the network design, the partition parameter "***N***" in ProteoNet is deliberately set as a hyperparameter. In this section, we explore how the model's performance varies with different values of "***N***".

As illustrated in Figure 7, the model achieves its highest accuracy when "***N***" is set to 5, followed by 7, and then 3. This finding highlights that an increase in the number of model partitions does not necessarily lead to improved performance. The decline in performance with a higher number of partitions can be largely attributed to the excessive granularity that results, causing the model to focus too much on details of individual blocks and reducing the effectiveness of weighting. Conversely, with "***N***" set to 3, the reduced number of part divisions means the model is less capable of capturing subtle variations within each part, leading to decreased accuracy. Nonetheless, on the whole, regardless of the number of partitions, there is still a noticeable

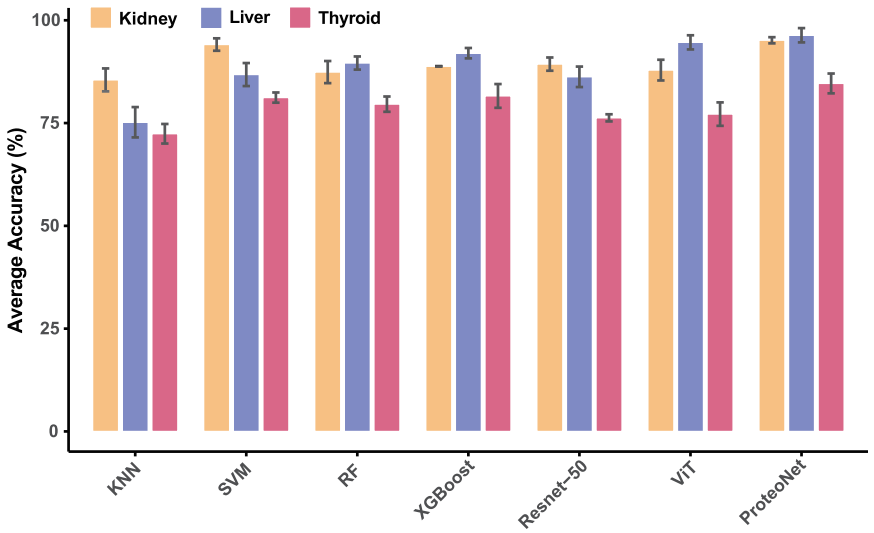


Figure 4. Average accuracy comparison of 5-fold cross-validation across three disease types

The average accuracy is shown for each model across kidney diseases (yellow), liver diseases (blue), and thyroid cancers (red). Liver diseases involve binary classification, thyroid cancers are a three-class problem, and kidney diseases present a four-class classification challenge All data are presented as mean ± standard error (SE) from 5-fold cross-validation.

improvement in accuracy relative to the baseline ResNet-50 model.

The generalization of ProteoNet framework

In our methodology, we highlight the adaptability of ProteoNet with different CNN-based backbone networks. Instead of conducting extensive experiments with various classical models like VGG[22] and AlexNet,[23] which are similar in parameter volume to ResNet-50, we chose to focus on MobileNetV2.[24] Our selection of MobileNetV2 is primarily motivated by its compact and lightweight architecture, making it highly suitable for applications in mobile and embedded devices where computational resources are often limited. This choice underscores our commitment to developing solutions that are not only efficient and accurate but also practical for real-world scenarios where resource constraints are a significant consideration.

As shown in Figure 8, the data clearly demonstrates that ProteoNet, when integrated with the MobileNetV2 backbone, achieves a remarkable improvement in performance metrics, such as accuracy, precision, recall, and F1 scores, all of which are significantly enhanced compared to the baseline MobileNetV2 model. Notably, this enhancement in performance does not come at the cost of increased computational burden. In fact, the switch to the MobileNetV2 backbone has led to a dramatic reduction in both the computational workload, measured in floating point operations per second (FLOPS), and the model size. These reductions are substantial, with computational workload decreasing by (6.24–0.32)/6.24 = 94.8% and

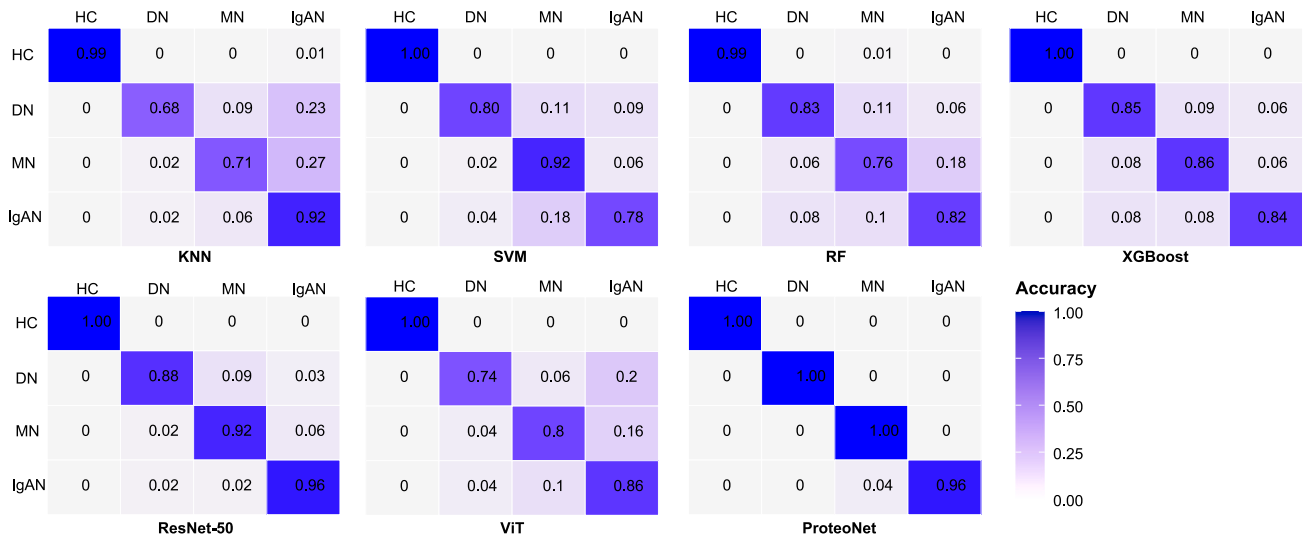


Figure 5. Confusion matrices for machine learning and deep learning models

The figure displays the confusion matrices for KNN, SVM, RF, and XGBoost machine learning models, as well as for Vision Transformer (ViT), ResNet-50, and ProteoNet deep learning models.

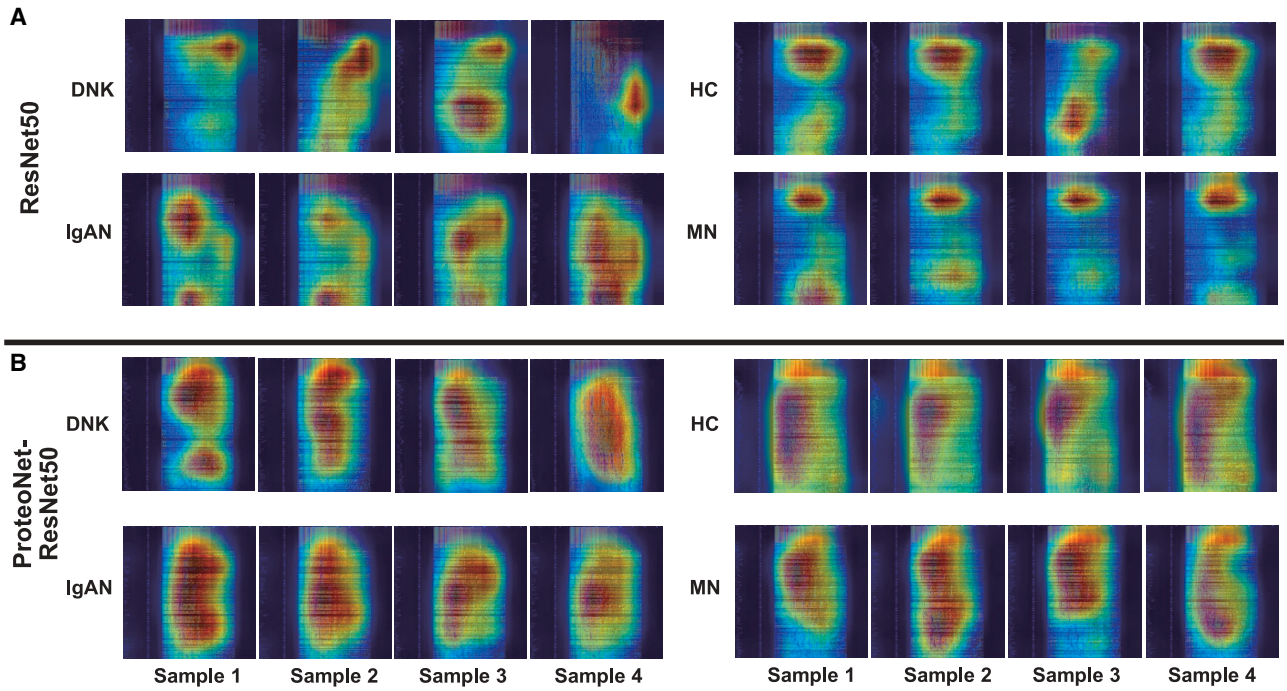


Figure 6. Class activation map comparison between ResNet-50 and ProteoNet for kidney disease subtypes

The red-highlighted regions indicate areas of higher importance identified by the model, suggesting the presence of distinct discriminative information. The kidney disease subtypes include HC (healthy controls), MN (membranous nephropathy), IgAN (IgA nephropathy), and DN (diabetic nephropathy).

model size by (26.15–5.16)/26.15 = 80.26%, while maintaining an impressive accuracy rate consistently above 92.2% (Table 3). In the case of employing the same backbone architecture, a significant improvement in accuracy has been observed with a minimal increase in computational overhead (≈400 K in ResNet-50 and ≈62.72 K in MobileNetV2) (Table 3). This balance of high efficiency and reduced resource demands highlights ProteoNet's suitability for deployment in portable devices. It offers a compelling solution for scenarios where high accuracy is required but computational resources and storage space are limited, reaffirming our focus on practical, real-world applications.

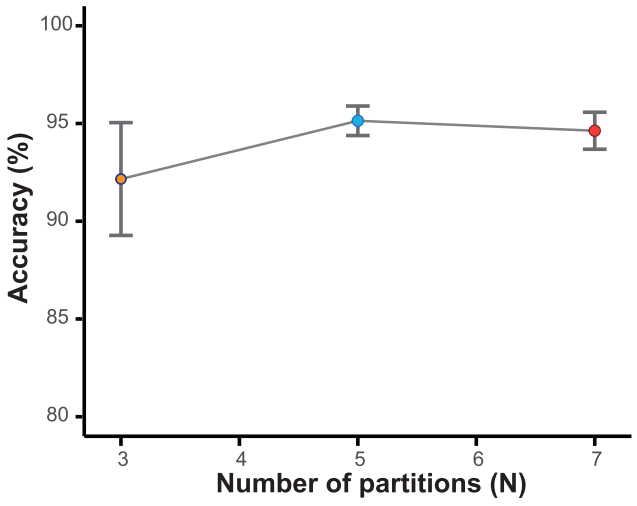


Figure 7. Correlation between number of partitions (*N*) and accuracy in ProteoNet for kidney disease data

This graph illustrates that, for the dataset analyzed in this study, an increase in the number of partitions within ProteoNet does not uniformly lead to enhanced accuracy. Data are presented as mean ± SE from 5-fold cross-validation.

DISCUSSION

Proteomics is essential for the accurate diagnosis of diseases, the discovery of biomarkers, and understanding disease mechanisms.[25,26] Despite its significant potential, the application of proteomics in clinical settings is often hindered by complex and time-intensive data processing.[15] To overcome these challenges, we developed the ProteoNet framework, which rapidly converts MS data into MS-RGB images and utilizes the power of CNNs for image recognition, significantly speeding up the analysis process and enhancing its precision.

Although CNN models like ResNet and MobileNetV2 have achieved notable success in image classification, they often employ global pooling strategies that can miss subtle yet crucial features in MS-RGB images, potentially compromising the accuracy of diagnostics.[24,27] ProteoNet addresses these limitations by incorporating semantic partitioning, adaptive average pooling, and weighted factors into its architecture. This integration enhances the model's focus on critical image regions, significantly improving diagnostic precision by allowing detailed analysis of essential areas within the MS-RGB images.

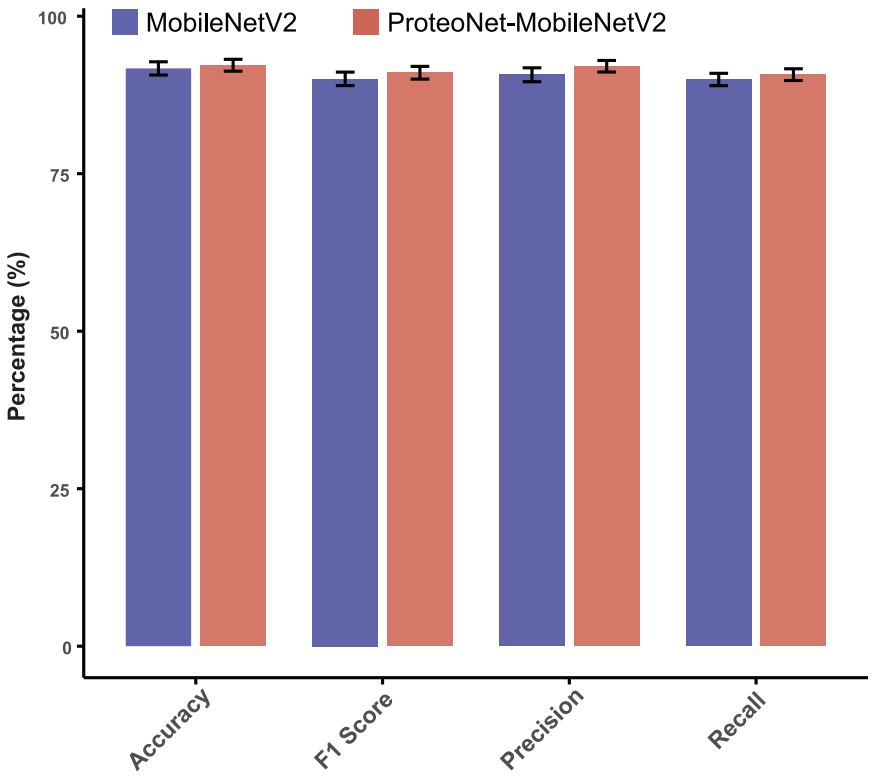


Figure 8. Performance evaluation of ProteoNet with MobileNetV2 on kidney diseases

All values are represented as the mean ± SE from 5-fold cross-validation.

The semantic partitioning approach in ProteoNet effectively segments significant regions for targeted analysis, reducing computational complexity and increasing the efficiency of data processing. However, this technique may introduce decision-making noise, particularly when images display uniform patterns across the segmented areas. To mitigate this, ProteoNet employs a learnable weighted factor that dynamically adjusts the significance of each partitioned segment during classification, thereby enhancing the model's adaptability and improving accuracy throughout the training process.

The effectiveness of ProteoNet has been demonstrated with urinary, blood, and tissue proteomics data from kidney diseases, liver diseases, and thyroid cancers, covering a range of classifications. Compared to deep learning models like ResNet-50 and ViT transformer, and ML models, such as SVM, RF, KNN, and XGBoost, ProteoNet has shown superior classification performance. Additionally, its ability to analyze and optimize high-level semantic feature maps makes it not only compatible with robust models like ResNet-50 but also adaptable to more compact architectures like MobileNetV2. This versatility underlines its potential suitability for deployment in mobile and embedded devices, where computational resources are limited, thus significantly enhancing its practical value for real-world applications.

Limitations of the study

Although our research focused on designing model structures for regional high similarity in MS-RGB data, there is still room for improvement in key feature extraction. Furthermore, validation on more disease phenotypes is necessary. Future research

Table 3. Computational and storage requirements comparison between ProteoNet, ResNet-50, and MobileNetV2

	Model Performance		
Backbone Network	FLOPS (K) ↓	Model Size (M) ↓	Accuracy (%) ↑
ProteoNet-ResNet-50	6244453.38	26.15	95.1
ResNet-50	6244051.97	23.52	89.3
ProteoNet-MobileNetv2	319081.73	5.16	92.2
MobileNetv2	319019.01	3.51	91.1

Note: Floating Point Operations Per Second (FLOPS) quantifies the computational performance of a deep learning model, with a lower FLOPS value indicates reduced computational load. Model size refers to the storage space required by the model, where a smaller value means less storage space is needed.

should aim to refine ProteoNet's features, extend its application across a broader spectrum of diseases, and continually improve its performance. This ongoing development will help establish ProteoNet as a critical tool in clinical diagnostics, enhancing the capability of healthcare professionals to make rapid and accurate diagnoses.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Yang Zhao (Zhaoy@nim.ac.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All data reported in this paper are shared in the relevant literature, with DOIs provided in the key resources table under Deposited Data.
- The source code employed in the current research can be accessed on the GitHub page: https://github.com/whisperH/ProteoNet
- Additional information for reanalyzing the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (2022YFF0608404 and 2022YFF0705001), Special Support Program for National High-level Talents in Science and Technology (WR2202), National Natural Science Foundation of China (nos. 21927812), Research Project of the National Institute of Metrology (AKYZD2111 and APT2101-5).

AUTHOR CONTRIBUTIONS

D.A., Yang Zhao, X.D., and X.F. directed and designed research; J.H., Y.L.,Yang Zhao, Y.W., and D.A. developed the corresponding algorithmic workflow and software; Yang Zhao, B.M., and Yong Zhang collected, organized and analyzed the quantitative data; Yang Zhao, J.H., D.A., and X.F. wrote and reviewed the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
- METHOD DETAILS
  - Overview of ProteoNet
  - Data source and MS-RGB image conversion
  - Data augmentation of MS-RGB data
  - Evaluation of ProteoNet framework
  - Implementation details
  - ML and DL methods for comparison
- QUANTIFICATION AND STATISTICAL ANALYSIS
- ADDITIONAL RESOURCES

Received: January 8, 2024
Revised: June 15, 2024
Accepted: November 7, 2024
Published: November 12, 2024

REFERENCES

1. Aebersold, R., and Mann, M. (2016). Mass-spectrometric exploration of proteome structure and function. Nature *537*, 347–355. https://doi.org/10.1038/nature19949.
2. Ding, Z., Wang, N., Ji, N., and Chen, Z.S. (2022). Proteomics technologies for cancer liquid biopsies. Mol. Cancer *21*, 53. https://doi.org/10.1186/s12943-022-01526-8.
3. Zhao, Y., Xue, Q., Wang, M., Meng, B., Jiang, Y., Zhai, R., Zhang, Y., Dai, X., and Fang, X. (2023). Evolution of Mass Spectrometry Instruments and Techniques for Blood Proteomics. J. Proteome Res. *22*, 1009–1023. https://doi.org/10.1021/acs.jproteome.3c00102.
4. Seyed Tabib, N.S., Madgwick, M., Sudhakar, P., Verstockt, B., Korcsmaros, T., and Vermeire, S. (2020). Big data in IBD: big progress for clinical practice. Gut *69*, 1520–1532. https://doi.org/10.1136/gutjnl-2019-320065.
5. Xiao, Q., Zhang, F., Xu, L., Yue, L., Kon, O.L., Zhu, Y., and Guo, T. (2021). High-throughput proteomics and AI for cancer biomarker discovery. Adv. Drug Deliv. Rev. *176*, 113844. https://doi.org/10.1016/j.addr.2021.113844.
6. Yue, L., Zhang, F., Sun, R., Sun, Y., Yuan, C., Zhu, Y., and Guo, T. (2020). Generating Proteomic Big Data for Precision Medicine. Proteomics *20*, e1900358. https://doi.org/10.1002/pmic.201900358.
7. Huang, J., Zhao, Y., Meng, B., Lu, A., Wei, Y., Dong, L., Fang, X., An, D., and Dai, X. (2024). SEAOP: a statistical ensemble approach for outlier detection in quantitative proteomics data. Brief. Bioinform. *25*, bbae129. https://doi.org/10.1093/bib/bbae129.
8. Zhao, W. (2021). KDClassifier: A urinary proteomic spectra analysis tool based on machine learning for the classification of kidney diseases. Aging Pathobiol. Therapeut. *3*, 63–72. https://doi.org/10.31491/apt.2021.09.064.
9. Sun, Y., Selvarajan, S., Zang, Z., Liu, W., Zhu, Y., Zhang, H., Chen, W., Chen, H., Li, L., Cai, X., et al. (2022). Artificial intelligence defines protein-based classification of thyroid nodules. Cell Discov. *8*, 85. https://doi.org/10.1038/s41421-022-00442-x.
10. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444. https://doi.org/10.1038/nature14539.
11. Strack, R. (2019). Deep learning in imaging. Nat. Methods *16*, 17. https://doi.org/10.1038/s41592-018-0267-9.
12. Liu, Y., De Vijlder, T., Bittremieux, W., Laukens, K., and Heyndrickx, W. (2021). Current and future deep learning algorithms for tandem mass spectrometry (MS/MS)-based small molecule structure elucidation. Rapid Commun. Mass Spectrom. *1*, e9120. https://doi.org/10.1002/rcm.9120.
13. Stahlschmidt, S.R., Ulfenborg, B., and Synnergren, J. (2022). Multimodal deep learning for biomedical data fusion: a review. Brief. Bioinform. *23*, bbab569. https://doi.org/10.1093/bib/bbab569.
14. Garcia-Jacas, C.R., Pinacho-Castellanos, S.A., Garcia-Gonzalez, L.A., and Brizuela, C.A. (2022). Do deep learning models make a difference in the identification of antimicrobial peptides? Brief Bioinform. *23*, bbac094. https://doi.org/10.1093/bib/bbac094.
15. Cadow, J., Manica, M., Mathis, R., Guo, T., Aebersold, R., and Rodríguez Martínez, M. (2021). On the feasibility of deep learning applications using raw mass spectrometry data. Bioinformatics *37*, i245–i253. https://doi.org/10.1093/bioinformatics/btab311.
16. Zhang, F., Yu, S., Wu, L., Zang, Z., Yi, X., Zhu, J., Lu, C., Sun, P., Sun, Y., Selvarajan, S., et al. (2020). Phenotype Classification using Proteome Data in a Data-Independent Acquisition Tensor Format. J. Am. Soc. Mass Spectrom. *31*, 2296–2304. https://doi.org/10.1021/jasms.0c00254.
17. Buchberger, A.R., DeLaney, K., Johnson, J., and Li, L. (2018). Mass Spectrometry Imaging: A Review of Emerging Advancements and Future Insights. Anal. Chem. *90*, 240–265. https://doi.org/10.1021/acs.analchem.7b04733.

18. Meyer, J.G. (2021). Deep learning neural network tools for proteomics. Cell Rep. Methods *1*, 100003. https://doi.org/10.1016/j.crmeth.2021.100003.

19. Neely, B.A., Dorfer, V., Martens, L., Bludau, I., Bouwmeester, R., Degroeve, S., Deutsch, E.W., Gessulat, S., Käll, L., Palczynski, P., et al. (2023). Toward an Integrated Machine Learning Model of a Proteomics Experiment. J. Proteome Res. *22*, 681–696. https://doi.org/10.1021/acs.jproteome.2c00711.

20. Adusumilli, R., and Mallick, P. (2017). Data Conversion with ProteoWizard msConvert. Methods Mol. Biol. *1550*, 339–368. https://doi.org/10.1007/978-1-4939-6747-6_23.

21. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In IEEE International Conference on Computer Vision (ICCV).

22. Simonyan, K., and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In International Conference on Learning Representations (Computational and Biological Learning Society).

23. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Commun. ACM *60*, 84–90. https://doi.org/10.1145/3065386.

24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.

25. Zeng, J., Rong, W., Meng, B., Zheng, L., Peng, T., Zhai, R., Jiang, Y., Xiao, T., Fang, X., Zhang, Y., et al. (2024). Integrated plasma proteomics and N-glycoproteomics reveals alterations in the N-glycosylation in Chinese hepatocellular carcinoma patients. Proteomics. Clin. Appl. *18*, e202300029. https://doi.org/10.1002/prca.202300029.

26. Jiang, Y., Sun, A., Zhao, Y., Ying, W., Sun, H., Yang, X., Xing, B., Sun, W., Ren, L., Hu, B., et al. (2019). Proteomics identifies new therapeutic targets of early-stage hepatocellular carcinoma. Nature *567*, 257–261. https://doi.org/10.1038/s41586-019-0987-8.

27. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

28. Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. *2*, 1–27. https://doi.org/10.1145/1961189.1961199.

29. Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32. https://doi.org/10.1023/a:1010933404324.

30. Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. IEEE Trans. Inf. Theory *13*, 21–27. https://doi.org/10.1109/tit.1967.1053964.

31. Chen, T., and Guestrin, C. (2016). XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.

32. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv. https://doi.org/10.48550/arXiv.2010.11929.

33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

34. Lee, M. (2023). Recent Advancements in Deep Learning Using Whole Slide Imaging for Cancer Prognosis. Bioengineering (Basel) *10*, 897. https://doi.org/10.3390/bioengineering10080897.

35. Yang, Z., Hu, F., Wang, J., Zhang, J., and Li, L. (2017). Learning Image Representation Based on Convolutional Neural Networks. In Neural Information Processing: 24th International Conference, ICONIP 2017 (Springer International Publishing), pp. 642–652.

36. Pang, Y., Sun, M., Jiang, X., and Li, X. (2018). Convolution in Convolution for Network in Network. IEEE Trans. Neural Netw. Learn. Syst. *29*, 1587–1597. https://doi.org/10.1109/TNNLS.2017.2676130.

37. Ma, J., Chen, T., Wu, S., Yang, C., Bai, M., Shu, K., Li, K., Zhang, G., Jin, Z., He, F., et al. (2019). iProX: an integrated proteome resource. Nucleic Acids Res. *47*, D1211–D1217. https://doi.org/10.1093/nar/gky869.

38. Niu, L., Geyer, P.E., Wewer Albrechtsen, N.J., Gluud, L.L., Santos, A., Doll, S., Treit, P.V., Holst, J.J., Knop, F.K., Vilsbøll, T., et al. (2019). Plasma proteome profiling discovers novel proteins associated with non-alcoholic fatty liver disease. Mol. Syst. Biol. *15*, e8793. https://doi.org/10.15252/msb.20188793.

39. Shorten, C., and Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. J. Big Data *6*, 60. https://doi.org/10.1186/s40537-019-0197-0.

40. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A Large-Scale Hierarchical Image Database. In 2009 IEEE conference on computer vision and pattern recognition (IEEE), pp. 248–255.

41. Zhao, Y., Zhang, Y., Meng, B., Luo, M., Li, G., Liu, F., Chang, C., Dai, X., and Fang, X. (2024). A Novel Integrated Pipeline for Site-Specific Quantification of N-glycosylation. Phenomics *4*, 213–226. https://doi.org/10.1007/s43657-023-00150-w.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
proteomic data of kidney diseases	A urinary proteomic data[8]	https://doi.org/10.31491/APT.2021.09.064
proteomic data of thyroid cancers	A thyroid proteomic data[9]	https://doi.org/10.1038/s41421-022-00442-x
proteomic data of liver diseases	A liver proteomic data[27]	https://doi.org/10.15252/msb.20188793
MS-RGB: Kidney diseases	This paper	https://drive.google.com/file/d/1vwe9UpybCOmZKiQNtAoYu90AFGdN_93H/view?usp=drive_link
MS-RGB: Thyroid cancer	This paper	https://drive.google.com/file/d/1APegDokyOHrEa1qgLdphLj3rh3Od43Tx/view?usp=drive_link
MS-RGB: Liver diseases	This paper	https://drive.google.com/file/d/1bj3LSZqbJpLAQeU1tnmgccTZADknn0e2/view?usp=drive_link
Software and algorithms		
MSConvert	Adusumilli et al.[20]	https://proteowizard.sourceforge.io/
SVM classifier	Chang et al.[28]	https://scikit-learn.org/stable/auto_examples/svm/index.html
RF	Breiman et al.[29]	https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
KNN	Cover et al.[30]	https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
XGBoost	Chen et al.[31]	https://xgboost.readthedocs.io/en/stable/
ResNet-50	He et al.[27]	https://huggingface.co/microsoft/resnet-50
Vision Transformer	Dehghani et al.[32]	https://huggingface.co/docs/transformers/model_doc/vit
Scikit-Learn	Fabian et al.[33]	https://scikit-learn.org
Custom computer code	This paper	https://github.com/whisperH/ProteoNet
Pytorch	Version 1.7.1	https://pytorch.org/
Python	Version 3.6.0	https://www.python.org/

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

This study is computational science research and does not utilize experimental models typical of the life sciences. All original data were obtained from published literature, with references listed in the key resources table.

METHOD DETAILS

Overview of ProteoNet

In clinical proteomics, distinguishing between disease subgroups relies on detecting variations in protein expression profiles. These variations are reflected in MS-RGB data as a series of spectral signals showing changes in specific regions, while other areas of the spectrum often display notable similarities. Thus, accurately analyzing these crucial signal regions is essential for effective clinical diagnostics or classification. To address this needed, we introduce ProteoNet, an innovative CNN-based architecture designed to enable detailed analysis of small and specific feature regions within proteomic MS-RGB data. The structure of this model is detailed in the Figure 1.

(1) Backbone network of ProteoNet

In computer vision, CNN models have emerged as key tools for processing visual data due to their profound capability to automatically learn and extract multi-layer features from visual inputs. They have been widely used in various AI applications for image

recognition.[34] A typical CNN architecture includes several key components: 1) **Convolutional Layers**: Often considered the core of a CNN, these layers use small filters to extract local features from an image, aiding in pattern recognition across various sizes and locations. 2) **Pooling Layers**: Positioned after the convolutional layers, they reduce the spatial size of the feature maps, cutting down computational complexity while retaining key information. 3) **Fully Connected Layers**: These layers come into play after the image has been processed by the convolutional and pooling layers, serving critical roles in classification or regression tasks by making decisions based on the features extracted. 4) **Activation Functions**: Functions such as ReLU introduce non-linearity into the network, which is crucial for handling complex data relationships.

Recent advancements in CNN have seen models like ResNet[27] and VGG[22] become the go-to choices for image analysis due to their advanced feature extraction capabilities. Among these, the ResNet-50 architecture stands out for its accuracy and adaptability across diverse datasets, making it an optimal choice for complex tasks. Within our ProteoNet framework, we have integrated the ResNet-50 architecture as the backbone network to harness its efficient processing of complex image features and robust performance with large datasets. This integration significantly boosts ProteoNet's capabilities in image recognition.[35] By processing MS-RGB data through ResNet-50, a function denoted as $h(\cdot, \theta)$ with parameters θ, we convert them into 3D (Width × Height × Channel) Tensor $\boldsymbol{T}$, which contain rich semantic information vital for our sophisticated analytical applications. The formula is as follows:

$$T = h(\cdot, \theta)$$

(2) Semantic partition and weighted factors

In typical CNN architectures, a global average pooling layer is used to generalize feature extraction by calculating the average values of all semantic features, simplifying model complexity.[36] While effective, this method may smooth out subtle, yet crucial, differences within the MS-RGB data. In clinical proteomics, where these differences can represent important biomarkers or distinctions in protein profiles, preserving fine-grained details is essential for accurate clinical diagnosis.[18,37]

To overcome these challenges, ProteoNet incorporates two strategies: firstly, the use of multiple semantic partitions to provide more detailed and discriminative features, and secondly, the application of weighted feature integration to mitigate the impact of noise and disturbances. For multiple semantic partitions, the Tensor $\boldsymbol{T}$ is proposed to divide into $\boldsymbol{N}$ regions, aiming at enhancing the prominent differences in the proteomics MS images by optimizing the semantic features in each specific region. Specifically, $\boldsymbol{T}$ is divided along the width axis into row vectors, and then all these row vectors are averaged into a single stripe, resulting in a part-level column vector. Following this partitioning, multiple convolutional layers equipped with ReLU[22] activation are utilized. This process integrates the channel information and simultaneously reduces computational complexity. Finally, each part-level column vector, particularly the i-th part, is input into a classifier with parameters φ_i, denoting as $g_i(\cdot; \varphi_i)$. Training for θ and φ in the model to guide models to focus on extracting the more significant features. Therefore, the i-th part with backbone network can be summarized as equation:

$$f(\cdot; \theta, \varphi_i) = \sigma(g_i(T; \varphi_i))$$

Where σ is *softmax* function.

The semantic partition architecture, developed to enhance localized signal information, faces a challenge when proteomics image data from diverse parts exhibit uniform band patterns in specific regions. In such case, the partition architecture risks introducing noise and may contribute to model overfitting. To mitigate this, our ProteoNet incorporates a learnable Weighted factor. This factor is crucial in determining the role of each semantic part in classifying proteomics image data: a higher weight indicates a greater relevance for classification, whereas a lower weight suggests a reduced or negligible contribution. Considering the dynamic and adaptable nature of deep learning models, the Weighted factor is designed as an end-to-end learnable parameter, enabling automatic adjustment during training. Consequently, the revised Equation, incorporating the i-th Weighted factor, is reformulated as follows:

$$f(\cdot; \theta, \varphi) = \sum_{i=0}^{N} \sigma(g_i(T; \varphi_i))$$

And finally, the training of the parameters w, θ and ϕ in the network is optimized by a focal loss function, which can be represented as:

$$\mathcal{L}_c = -\sum_{(\mathbf{x},\mathbf{y}) \in D} \alpha(1 - f_c)^{\gamma} \log(f_c)$$

Where f_c is the predicted probability of c class in equation. α serves as a balancing factor to adjust the weights of positive and negative classes and is typically set as learnable parameters in this paper. $(1 - f_c)^{\gamma}$ term referred to as the 'focusing parameter', which helps decrease the weight of easily classified samples while increasing the weight of challenging ones. Typically, γ is set to 2 in this paper. $\mathbf{x}$ is the set of proteomics MS-RGB images, $\mathbf{y}$ indicates the label set of the corresponding MS-RGB image and D is the sample space.

Data source and MS-RGB image conversion

Next, we evaluated the performance of ProteoNet using three clinical proteomic datasets, specifically focusing on kidney diseases,[8] liver diseases[38] and thyroid cancers.[9] The kidney dataset comprises urinary proteomic MS data from 50 patients with IgA

nephropathy (IgAN), 49 with membranous nephropathy (MN), 35 with diabetic nephropathy (DN), and 72 healthy controls (HC). The liver dataset includes serum proteomic MS profiles from 48 patients with either Non-alcoholic Fatty Liver disease or Cirrhosis disease (FLC) and 118 healthy controls (HC). The thyroid dataset contains tissue proteomic MS profiles from 61 patients with Follicular Thyroid Carcinoma (FTC), 72 with Papillary Thyroid Carcinoma (PTC), and 82 healthy controls (HC).

To covert the proteomic MS data into MS-RGB format, we employed ProteoWizard, an open-source software commonly used in proteomics. Specifically, we develop a code snippet that directly utilizes the MSConvert toolkit within the ProteoWizard software, transforming the MS data into high-resolution, 3-channel MS-RGB images, each 2048 x 2048 pixels in size.[20] The conversion process involved setting the mass-to-charge (m/z) display range from 200 to 2000. We used the maximum intensity value from each bin for visualization, applying a blue-red-yellow gradient heatmap to represent different intensity levels.

The proteomic MS files for kidney diseases, liver diseases and thyroid cancers are deposited in the ProteomeXchange Consortium with the IDs PXD018996, PXD011839, PXD036554 and PXD036554, respectively.

Data augmentation of MS-RGB data

Numerous studies have shown that a diverse dataset can significantly improve model training, reduce overfitting, and enhance generalization capabilities. Given the limited size of our evaluation dataset, we conducted data augmentation on our MS-RGB data. This technique is widely employed in machine learning and deep learning to increase the size of the dataset and enhance the diversity of the samples. Specifically, we employed 19 distinct data augmentation techniques, which include Random Resize and Crop, Random Flipping, ColorJitter, AutoContrast, Equalize, Invert, Rotation, Brightness, Sharpness, Shear, Normalization, Cutout, among others, to enhance MS-RGB data diversity.[39] These techniques were applied in pairs throughout the training phase to ensure a diverse dataset. Conversely, during the testing and validation phases, data manipulation was confined to center cropping and normalization only. The effects of these augmentations are illustrated in the Figure 2.

Evaluation of ProteoNet framework

To evaluate our ProteoNet model, we conducted performance assessments on the MS-RGB data for kidney diseases, thyroid cancers, and liver diseases, addressing four-class, three-class, and binary classification problems, respectively. During the testing and inference stages, the final outcome was determined by averaging the results from a 5-fold cross-validation process. This process involves evaluating the model's performance over five separate iterations of cross-validation, and then calculating the average of these evaluations. Such an approach helps to reduce the variability caused by data partitioning process and stochastic elements, thereby providing a more stable and dependable performance estimation.

In our study, the effectiveness of the ProteoNet classification was evaluated using several metrics: accuracy, precision, recall and F1 score. Accuracy here refers to the model's capability to correctly identify various classes, defined as the ratio of correctly identified samples to the overall number of samples. Additionally, precision in this context was used to measure the fraction of correctly predicted instances of a specific class against all predictions made for that class. On the other hand, recall measures the proportion of correctly identified items relative to the total number of items that should have been classified. These performance metrics can be formally defined as equation to:

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \times 100\%$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\%$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$$

Implementation details

We utilize the ResNet-50 pretrained on ImageNet as the feature extractor.[40] The batch size is set to 32, total epoch is set to 100. And the optimizer is Adam, the learning rate is set to 0.0001 initially, with a decay rate of 0.1 and the step decay schedule in 30, 60 and 90. All the input images of experiments and comparisons are resized to 224 × 224× 3. In addition, the number of partitions is set as 5 in this paper. In addition, during the discussion phase, we also addressed the sensitivity of the model's performance to the number of partitions. Afterward, the whole architecture is implemented with PyTorch 1.7.1 and trained on a single NVIDIA RTX GPU. The source code employed in the current research can be accessed on the GitHub page: https://github.com/whisperH/ProteoNet

ML and DL methods for comparison

To assess the efficacy of ProteoNet, we conducted a comparative analysis using a variety of methods. This included traditional machine learning algorithms as well as advanced deep learning models such as ResNet-50 and Vision Transformer. For the machine learning component, proteomic MS data from kidney diseases, liver diseases and thyroid cancers were processed using

MaxQuant software to generate quantitative protein expression data.[41] The result datasets comprised 1812 proteins across 206 samples for kidney diseases, 5628 proteins across 329 samples for liver diseases, and 2333 proteins across 150 samples for thyroid cancers. Before model construction, the data were subjected to quantile normalization and log2 transformation. Additionally, any missing values were filled using the minimum observed values in the dataset. These processed matrices were then used as input for several machine learning classifiers: Support Vector Machine (SVM),[28] Random Forest (RF),[29] K-Nearest Neighbors (KNN),[30] and Extreme Gradient Boosting (XGBoost).[31] For the deep learning models, the MS-RGB data were directly used as input for model analysis.

To ensure accuracy and reliability of our models, we set specific parameters for each classifier. For the SVM classifier, a linear kernel was employed, and the penalty parameter was set at 0.1. In the RF model, the maximum number of features was limited to 20, with a forest comprising 100 trees. The KNN classifier was configured with K set to 5 and used the Euclidean metric for distance calculation. Meanwhile, the XGBoost model was set up with a learning rate of 0.01, a maximum tree depth of 5, and a total of 100 trees. For both the ResNet-50 and Vision Transformer models, the model parameters follow the standard architecture as released by the original authors.[27,32] And two models conducted for 100 epochs and a batch size of 32. The ResNet-50 utilized the Adam optimizer with a learning rate set at 0.001, and the Vision Transformer was optimized by AdamW with a learning rate of 0.000125. The performance of these models was evaluated using 5-fold cross-validation, averaging the results for a comprehensive metric. This analysis was conducted using Python 3.7 and the Scikit-Learn library, ensuring a robust implementation of these ML models.[33]

QUANTIFICATION AND STATISTICAL ANALYSIS

The metrics analysis for calculating accuracy, precision, recall, and F1 score was conducted using mmpretrain (formerly mmclassification, https://github.com/open-mmlab/mmclassification) or Python (version 3.6, https://www.python.org/).

ADDITIONAL RESOURCES

This study did not create or expand any websites or resources, and it does not involve clinical experiments.