

RESEARCH

Open Access



Study on deep learning-based detection of viable cell count in dialysis fluid images

Michihiro Kawasaki^{1*}, Toshiki Shimozawa² and Satoshi Suzuki¹

Abstract

Background and aims Control of dialysate fluid quality is critical to secure the safety of dialysis treatment. The number of colonies is manually counted when determining viable cell count in dialysis fluid, but errors and subjective interpretation on the part of the measurer can be problematic. This prompted us to examine the potential for using deep learning to detect viable cells and count their numbers.

Methods In this study we prepared 5360 images for detecting viable cell count and classified them into four categories using the VGG-16 model. These images were resized to 224 × 224 px; 90% of them were used for learning, and 10% were used for validation. In an alternative approach, we also created 110 annotated images from images to detect viable cell count in dialysis fluid and subjected them to learning using the YOLOv5 model.

Results VGG-model had a detection accuracy using the test data was 43%. YOLOv5 model had a mAP (Mean Average Precision) was 0.842. The detection accuracy using the test data was 90%.

Conclusions The method using the VGG-16 model had problems with overfitting, suggesting that the model was not sufficiently expressive. The detection of viable cells using the YOLOv5 model showed high accuracy.

Keyword CNN, VGG-16, YOLO, Deep learning, Viable cell

Introduction

The cleanliness of the dialysate and the quality of the water used to make up dialysis fluid are critical factors in securing the safety of dialysis treatment. The Dialysis Fluid Quality Standards as laid out by The Japan Society for Dialysis Therapy in 2016 [1] require that dialysis fluid be evaluated for endotoxin concentrations and viable cell counts. One or more dialysis consoles should be checked at least once a month, and all consoles must be checked at least once a year. According to these Standards, the minimum water quality required for dialysis treatment is

defined as “standard dialysis fluid,” which has a dialysate ET concentration of less than 0.05 EU/mL and a viable cell count of fewer than 100 cfu/mL. Dialysis fluid with a dialysate ET concentration of less than 0.001 EU/mL (i.e., undetectable) and that with a viable cell count of less than 0.1 cfu/mL is defined as “ultrapure dialysis fluid (UPD).” Water quality control has become more important with the increasing use of online haemodiafiltration [2].

Specific measuring methods are indicated in the Procedure Manual for Achieving the 2016 Dialysis Fluid Quality Standards, published by the Japan Association for Clinical Engineers [3]. The Manual states that “the detection of viable cell count must be carried out by measuring and recording the number of colonies viable to the naked eye. At the same time, the shape and color tone of the colonies and the number of days required to form a colony should be recorded. Recording images using a digital camera or other means is recommended.”

*Correspondence:

Michihiro Kawasaki
kawasaki@cet.kanagawa-it.ac.jp

¹ Department of Clinical Engineering, Kanagawa Institute of Technology, Atsugi, Japan

² Department of Clinical Engineering, Toho University Ohashi Medical Center, Meguro, Japan



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

The number of colonies has hitherto been counted manually, a subjective process that is prone to error [4]. Colonies must also be identified based on their morphological characteristics, such as size, color, and texture. However, with manual counting, results from skilled persons and from beginners are likely to differ.

In recent years, machine learning has been used to analyze biological images.

Convolutional Neural Networks (CNN), the most basic neural network for image analysis, are widely used. The network structure is not designed from scratch, but existing models such as Visual Geometry Group-16 (VGG-16) and You-Only-Look-Once (YOLO) are applied and used.

VGG-16 is a model used for image classification, which classifies (predicts) whether a given image belongs to a specific class. YOLO is a model used for object detection, which locates objects in an image and identifies object classes. YOLO is suitable for real-time object detection and can simultaneously perform detection and classification in a single network. In view of the characteristics of each model, for the use in determining viable cell count, we consider that VGG-16 model can provide a binary classification of images, with and without viable cells, for the use in determining viable cell count. YOLO is expected to deeply observe viable cells in images and detect their number.

Purpose

This study aims to investigate the possibility of using deep learning image and object recognition to determine the presence or absence of viable cells and to count their numbers.

Methods (study 1)

Dataset

The 670 images used to detect viable cell count and their colony counts obtained from a dialysis treatment facility (photos taken: 2015–2022) were classified into four categories: 0, 1–10, 11–29, and 30 or more colonies. Then, from the images used to detect viable cell count obtained from another facility, seven images with zero colonies and seven other images in which colonies had been observed were selected randomly and used as the test data. All the images were cropped such that the membrane filter was positioned in the center of the image.

Since the learning data contained a small number of images, we diversified the dataset and expanded the image data to improve the generalization performance of the model.

5360 images were used as the dataset for learning in addition to the original images, with the seven types of image enhancements: rotation (30°, 60°, 90°), horizontal

Table 1 Input image classification

Classification (Number of colonies)	Number of images
0	4984
1–10	184
11–29	104
30 or more	88

Images were divided into four categories: 0, 1–10, 11–29, and 30+ colonies. Number of images per category.

flip, vertical flip, and brightness adjustment (0.7x, 1.3x), (Table 1).

5% out of the learning dataset was excluded as validation test data.

Learning

We used Visual Geometry Group 16 (VGG-16) [5], a type of Convolutional Neural Network (CNN). When using VGG-16 as is, the model weights are usually initialized with pre-trained weights from ImageNet datasets or other sources. In this study, we constructed a model (Model 1) to perform transition learning using VGG-16 as a feature extractor, and compared it with another model (Model 2) using some of the weights learned by VGG-16 (fixed up to 15 layers). If VGG-16 is used as is, the final layer of the model is designed for a 1000-class classification. To classify the number of colonies into four categories (0, 0–10, 11–29, and 30 or more), the existing final layer was invalidated, and the last layer was modified using the Global Average Pooling (GAP), Dense, and SoftMax functions for the output layer (Fig. 1). In this study, in which the number of learning images is small, we used GPA [6], which is designed to reduce the number of model parameters while preventing overfitting.

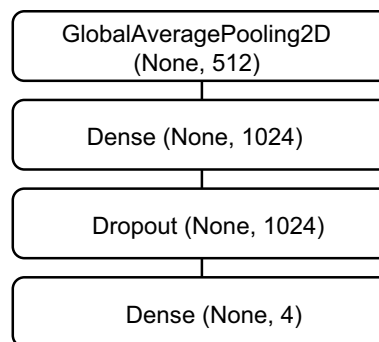


Fig. 1 Composition of the final layer. The existing final layer was invalidated, and the last layer was modified using the Global Average Pooling (GAP), Dense, and SoftMax functions for the output layer

The input image was resized to 224×224 px and normalized while maintaining the aspect ratio. The optimization function was Stochastic Gradient Descent (SGD), and the loss function was categorical_crossentropy. We used 90% of the dataset for learning and 10% for validation, and performed deep learning with a batch size of 8, 30 epochs, and Dropout of 0.5. We then changed Dropout to 0.7 and carried out learning again. We also performed learning in the same way by binarizing the input images. Under the above conditions, we performed learning of model1 and model2 learning.

Evaluation and testing

Using the loss function and the Evaluate method to evaluate the models, we input 268 pieces of previously excluded validation test data and compared the loss function, accuracy correct rate, precision, recall, and F1-score. Using a model (model2, Dropout=0.5) that minimizes the loss function and has the highest detection F1-score, we input the test data using the Predict method and classified the images.

Methods (study 2)

Dataset

We randomly selected 200 images from 670 images (photos taken: 2015–2022) obtained from a dialysis treatment facility, annotated them with colony and membrane filter (MF) labels, and coordinated the information using annotation tool labelling. The dataset consisted of learning data (140 images) and validation data (60 images). From the images used for determining viable cell count obtained from another facility, five images with zero colonies and five other images in which colonies were observed were selected randomly and used as test data (Same image as Study 1). All the images were cropped so as to place the membrane filter at the center of the image.

Models

In this study, we used the You Only Look Once (YOLOv5) models [7], which are used for object recognition. The YOLO system consists of multiple convolutional and pooling layers, and the final output is a feature map corresponding to a grid cell. The input image was resized to 224×224 px while maintaining the same aspect ratio. The optimization function was set to SGD, batch size to 16, and the number of epochs to 300.

Evaluation and testing

YOLOv5 has several models with different architecture and configuration-related parameters that provide

different detection accuracies and which are designed to cope with different computational loads. This study used four models (the n-, s-, m-, and l-size models, in ascending order of size). Learning was performed at the default high parameter settings, and the results were compared using Mean Average Precision (mAP), an index of accuracy. The mean Average Precision was calculated for multiple classes.

The test data were inferred using the learned weights of each model.

Results

The loss function for learning VGG-16 was Train 0.17 (Fig. 2a), Train 0.21 (Fig. 2b), Train 0.17 (Fig. 3a), and Train 0.24 (Fig. 3b) at Epoch 25 for Model 1, and did not decrease during any of the learning process. However, there is no obvious sign of overlearning, as the overall decreasing trend continues in both learning processes of Model 1, although there is a temporary increase in Epoch 18 with a validate of 0.21 (Fig. 3a). Model 2 shows a sharp increase in validation at certain epochs: validation 0.21 (Fig. 2d) at Epoch 7, validation 0.18 (Fig. 3c) at Epoch 20, and validation 0.13 (Fig. 3d) at Epoch 15. Also, validate is increasing around Epoch 10 (Fig. 2d), Epoch 10 (Fig. 3c), and Epoch 12 (Fig. 3d), and has not decreased from validate 0.05, but Train is converging to 0, indicating an overlearning trend. Using the Evaluate method, Model 2 (Dropout of 0.5, color images) had the smallest loss function and the highest detection accuracy, F1-score (Table 2 and 3). Table 4 shows the classification results using this model. Images with a colony count of 0 (OK01-OK07.jpg) were classified at an accuracy of more than 90%. On the other hand, images in which colonies had been observed were not correctly classified, showing a maximum detection accuracy of approximately 5% (NG03.jpg). From the above, only the images with 0 colony count were properly classified in the 14 test data. Thus, the test data's detection accuracy was 43% (Table 4).

While learning using the YOLO system, mAP did not differ significantly by model. The n-size model had the highest mAP (Table 5). The PR Curve and F1 Curve also did not differ significantly (Figs. 4 and 5). The inference results after inputting the test data into each model are shown in Table 6. The percentage of accuracy in the test data varied from model to model. mAP was highest for the n-sized model, but the detection accuracy was highest for the l-size model in the inference. Figure 6 shows an example of an inference result image. Colonies were detected in the Bounding Box. The object class label and object confidence were also above 0.7.

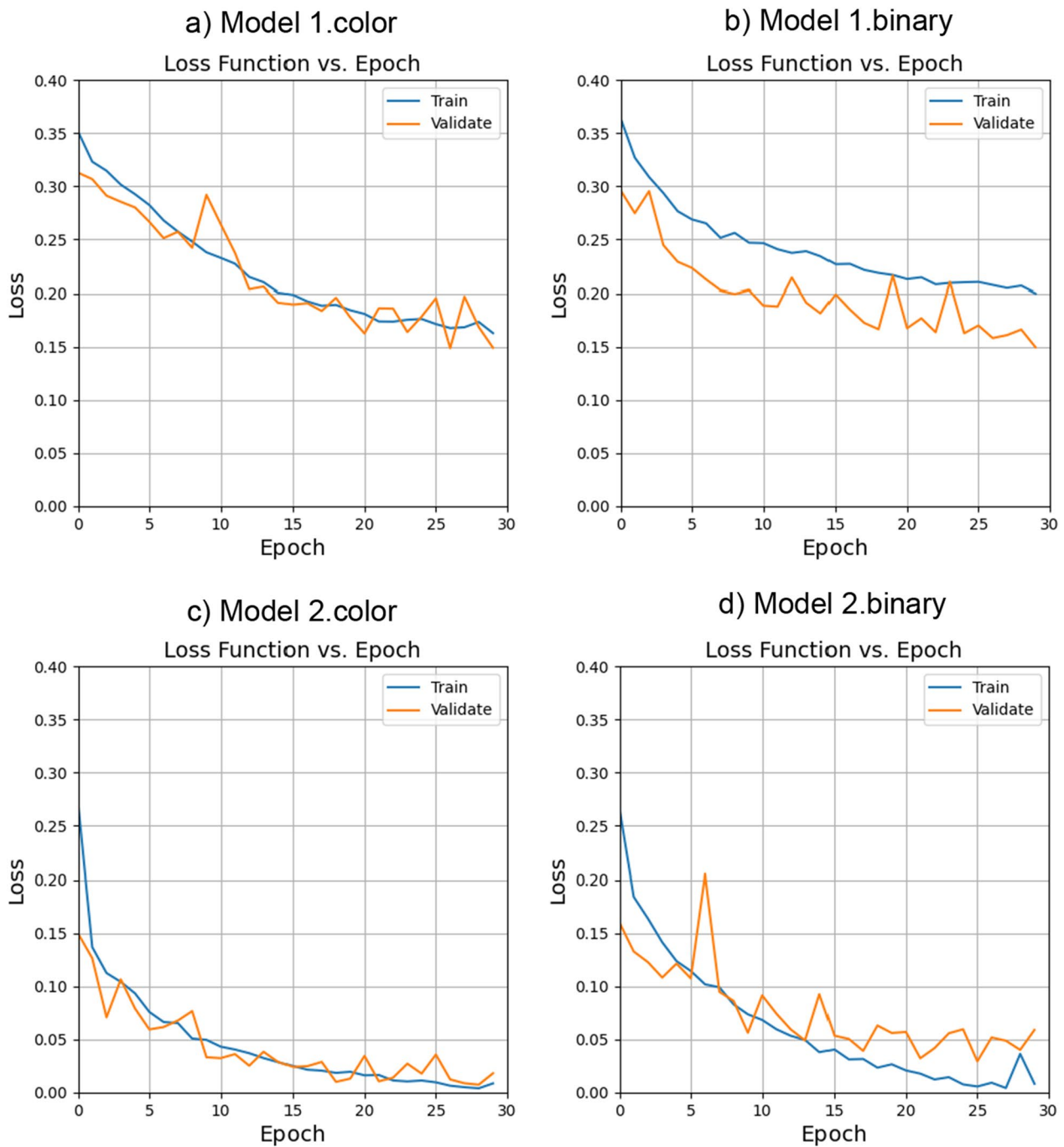


Fig. 2 Learning Loss (Dropout 0.5). The training process is shown for a model using VGG-16 as a feature extractor (model 1) and a model using some weights (fixed up to 15 layers) learned by VGG-16 (model 2). 0.5 Dropout is shown for color and binary input images

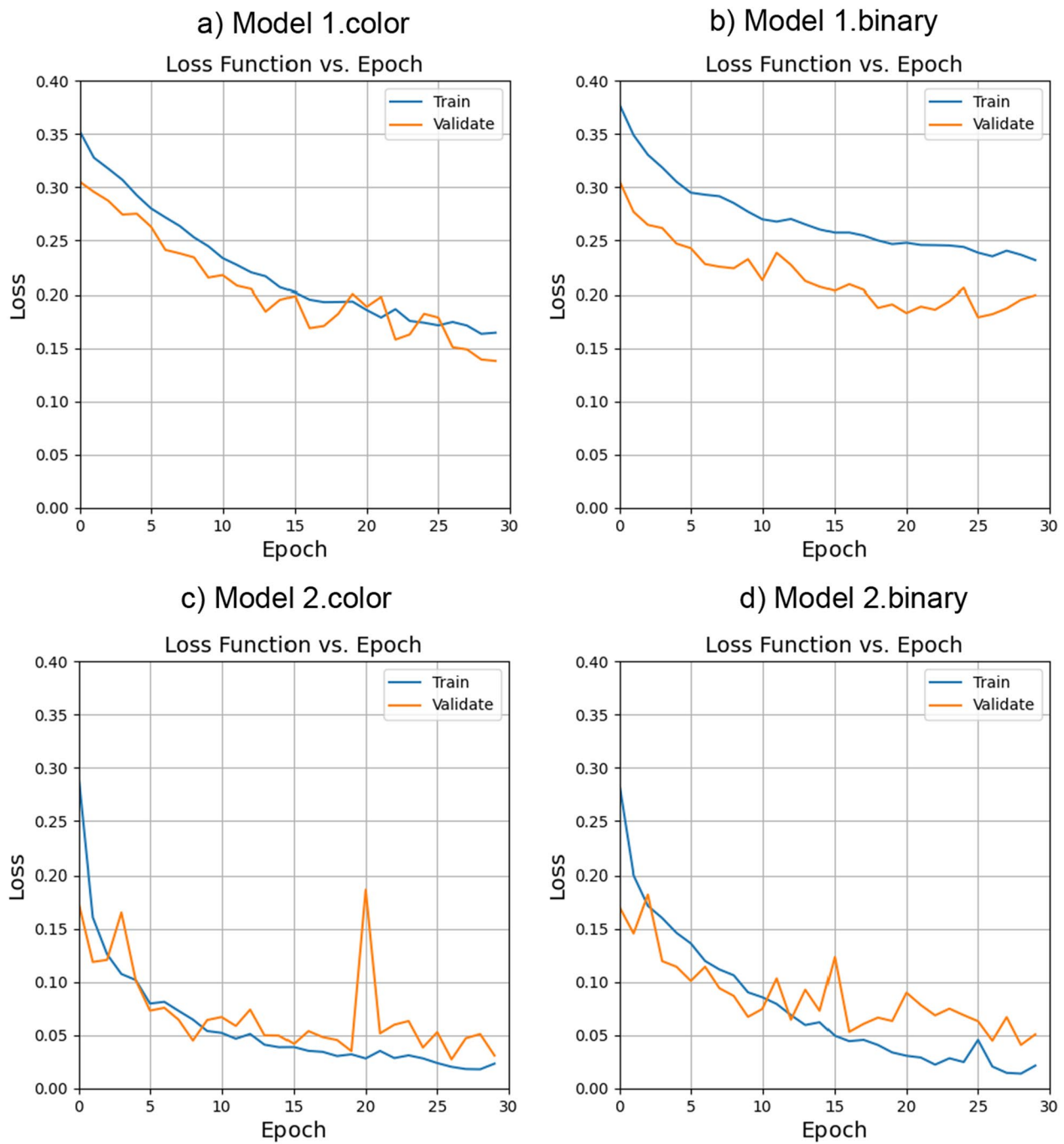


Fig. 3 Learning Loss (dropout 0.7). The training process is shown for a model using VGG-16 as a feature extractor (model 1) and a model using some weights (fixed up to 15 layers) learned by VGG-16 (model 2). 0.7 Dropout is shown for color and binary input images

Table 2 Evaluate results (dropout 0.5)

Input image type	Model1		Model2	
	Color	Binary	Color	Binary
Loss	0.1379	0.1941	0.0305	0.0883
Accuracy	0.9590	0.9336	0.9851	0.9805
Precision	0.9761	0.9518	0.9851	0.9842
Recall	0.9179	0.9258	0.9851	0.9766
F1-Score	0.9461	0.9386	0.9851	0.9801

The following figures show the Loss, Accuracy, Precision, Recall, and F1-score of a model (Model 1) using VGG-16 as a feature extractor and transition learning, and a model (Model 2) using some of the weights (fixed up to 15 layers) learned by VGG-16. Dropout is 0.5.

Table 3 Evaluate results (dropout 0.7)

Input image type	Model 1		Model 2	
	Color	Binary	Color	Binary
Loss	0.1708	0.2498	0.0487	0.0871
Accuracy	0.9328	0.9297	0.9813	0.9766
Precision	0.9609	0.9331	0.9813	0.9803
Recall	0.9179	0.9258	0.9813	0.9766
F1-Score	0.9389	0.9294	0.9813	0.9784

The following figures show the Loss, Accuracy, Precision, Recall, and F1-score of a model (Model 1) using VGG-16 as a feature extractor and transition learning, and a model (Model 2) using some of the weights (fixed up to 15 layers) learned by VGG-16. Dropout is 0.7.

Table 4 Classification results (model2.Color)

File name	Classification label				Correct label
	0	1–10	11–29	30 or more	
NG01.jpg	0.869	0.000	0.131	0.000	1–10
NG02.jpg	1.000	0.000	0.000	0.000	1–10
NG03.jpg	0.000	0.050	0.995	0.000	1–10
NG04.jpg	0.988	0.000	0.012	0.000	1–10
NG05.jpg	0.990	0.000	0.010	0.000	1–10
NG06.jpg	0.021	0.000	0.979	0.000	1–10
NG07.jpg	0.013	0.000	0.987	0.000	1–10
OK01.jpg	0.171	0.000	0.829	0.000	0
OK02.jpg	0.990	0.000	0.000	0.000	0
OK03.jpg	1.000	0.000	0.000	0.000	0
OK04.jpg	1.000	0.000	0.000	0.000	0
OK05.jpg	1.000	0.000	0.000	0.000	0
OK06.jpg	1.000	0.000	0.000	0.000	0
OK07.jpg	1.000	0			
Completion rate					43%

This is the result of classifying color test data using Model2.

Table 5 mAP

Model	mAP		
	Colony	MF	All
n	0.689	0.995	0.842
s	0.658	0.995	0.826
m	0.665	0.995	0.830
l	0.664	0.995	0.830

The mAPs for the four model sizes of YOLOv5 are shown.

Figure 7 shows an example of an undetected and over-detected inference result image.

Discussion

In Study 1, we evaluated the potential for using VGG-16 to classify images with and without colonies using picture recognition. Neural networks have the problem of overfitting, which could have been caused by a lack of learning data and excessive expressive power of the model [8]. In this study, there was little learning data, which may have contributed to a pattern of overfitting. This was especially noticeable in Model 1, in which we used VGG-16 as a feature extractor; however, VGG-16 is a model learned on a dataset comprising more than 10 million images across about 1000 classes of Image Net, suggesting learning using already-learned parameters to be unsuitable for feature extraction from the images used in this study. In Model 2, when a color image was input, the loss function converged, and there was less tendency toward overfitting, suggesting that learning had progressed appropriately. It appears that in CNN, the shallower the layer, the greater the likelihood that general features, such as edges and blobs, will be extracted, while the deeper the layer, the more features specific to learning data tend to be extracted [9, 10]. In Model 2, layers were fixed up to 15. Learning progressed appropriately, since general feature extractors in the shallow layers were fixed as they were, and only the weights in the deep layers were readjusted to suit the image recognition at this time. In addition, the use of Dropout [11] could have reduced overfitting. Dropout is a regularization technique used to temporarily reduce overfitting: it randomly disables network units at each learning step.

The detection accuracy was 43% when the test data was inferred by Model 2 (Dropout of 0.5), although the detection accuracy was over 90% at the time of learning. This was a case of the model being overfitted to the training data and becoming no longer generalizable to new data. When the Dropout value was increased, the network

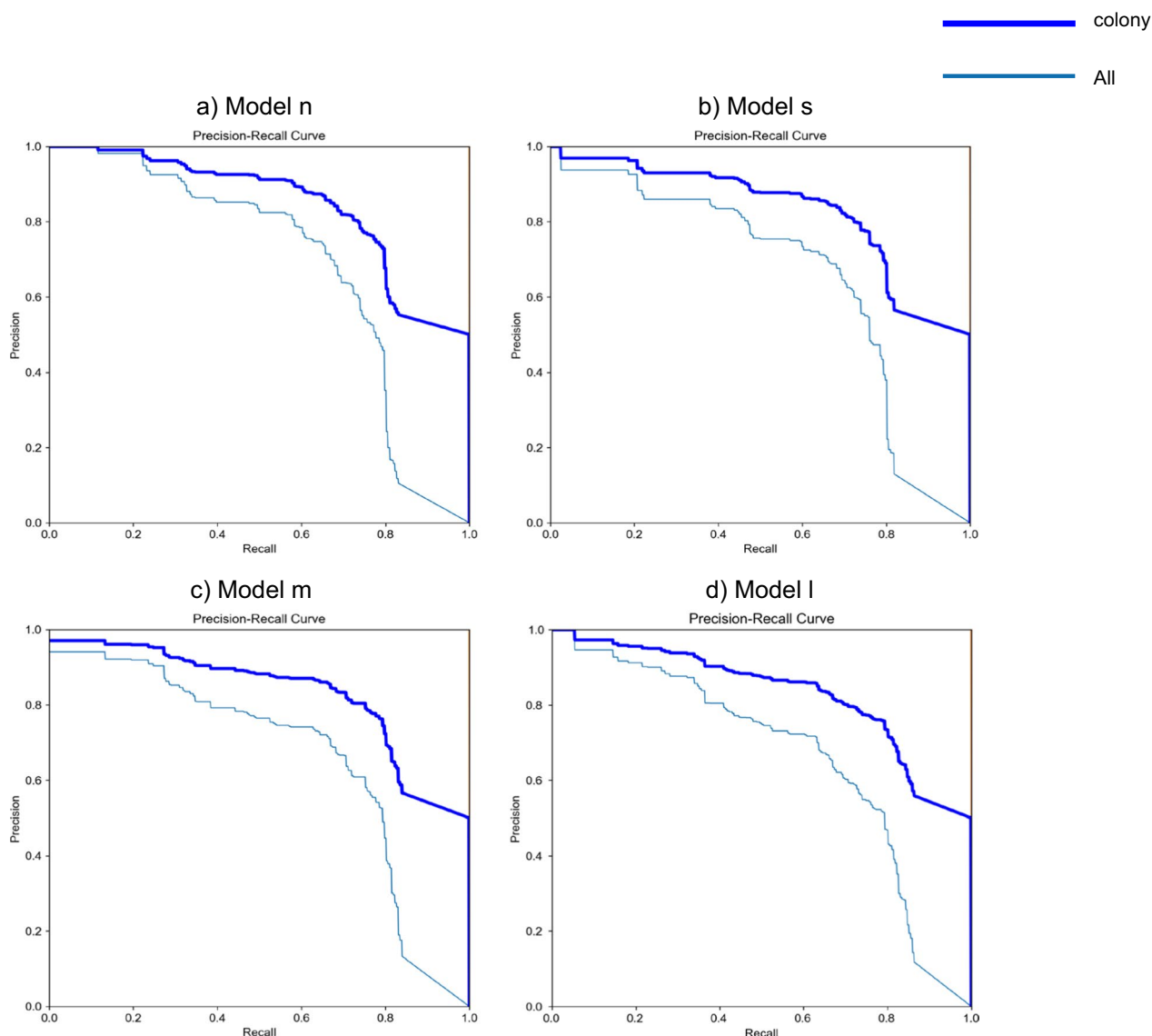


Fig. 4 PR Curve. The PR curves of Recall on the horizontal axis and Precision on the vertical axis show the validation progress of YOLOv5 at four different model sizes

became less expressive the number and the characteristics of input images used in this study, as it progressed to over-training. It was suggested that the network had difficulty capturing complex patterns in the data and became unable to adapt to the training data, resulting in over-learning due to excessive uncertainty.

In Study 2, YOLOv5 was used to evaluate the detection accuracy of viable cells. Using different-sized (n,

s, m, l) models, detection accuracy was evaluated by mAP as determined by the threshold of Intersection over Union (IoU), an index that indicates the overlap of the respective Bounding Box of predictions and correct answers. There were no differences among the models. The higher the detection accuracy, the higher was the mAP value.

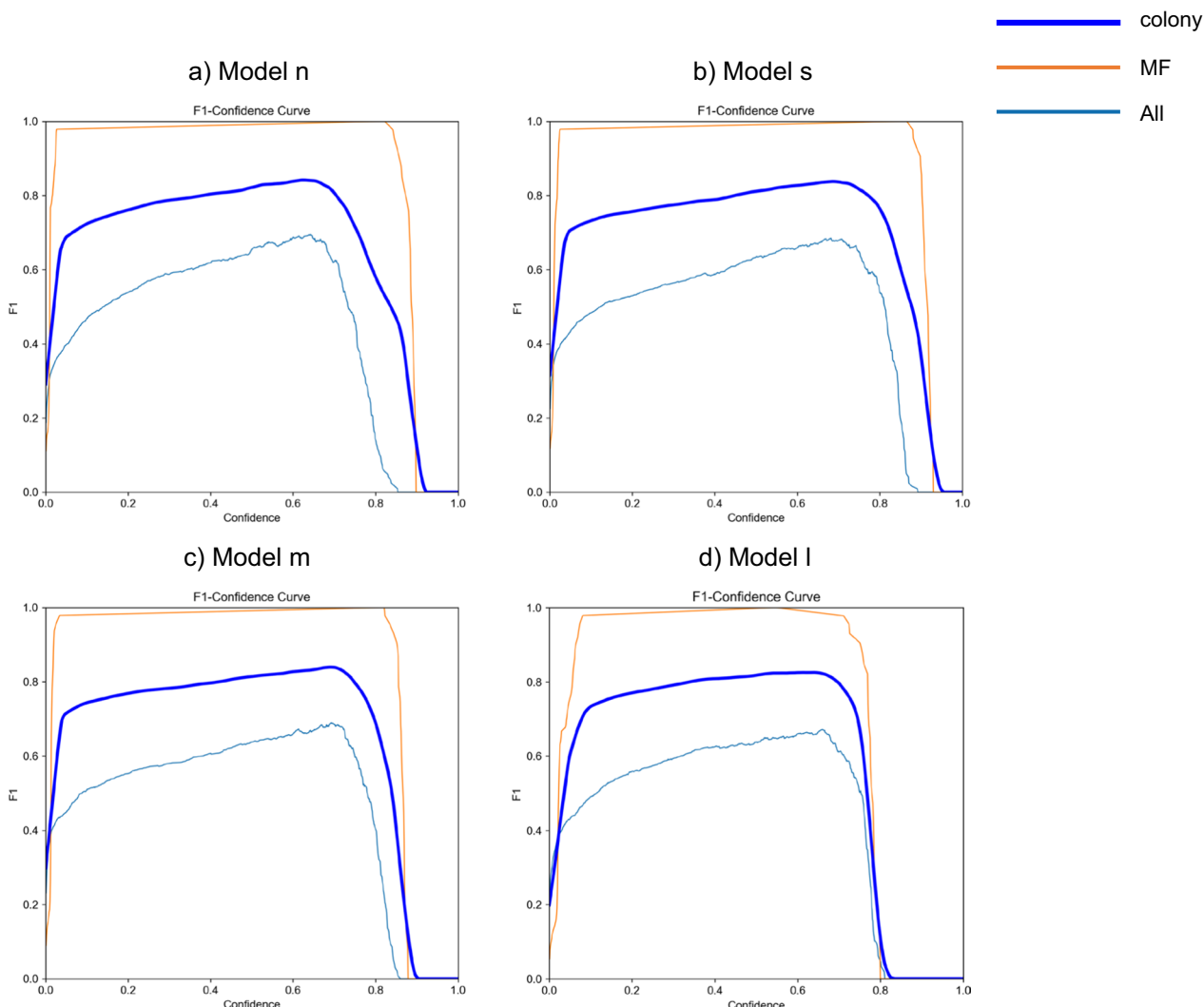


Fig. 5 F1 Curve. The validation progress of YOLOv5 at four different model sizes is shown by the F1 curve with Confidence on the horizontal axis and F1 on the vertical axis

YOLOv5 is designed based on architecture and a specific backbone (CSPDarknet53 or EfficientNet). EfficientNet is a network architecture for automatically optimizing model scaling, which minimizes differences between different models. Fewer significant differences are found in the results if the model architecture, weight initialization method, and learning method are consistent [12].

The inference results for the test data showed a greater than 80% accuracy for all models, with the l-size model showing the highest accuracy. The n-size model

had the highest value for the mAP, but the l-size model was more accurate in the actual inference. Larger models have more parameters and may learn more complex features. This is what enabled the l-size model to detect finer features and more complex patterns that the smaller models did not capture. In the images of the inference results, colonies were seen in the Bounding Box, and the class labels and confidence levels of the objects were appropriately estimated, suggesting that it is possible to count numbers of colonies using YOLOv5.

Table 6 Detection results

File name	Number of colonies (detection)				Correct
	Model n	Model s	Model m	Model l	
NG01.jpg	1	1	1	1	1
NG02.jpg	3	3	3	2	3
NG03.jpg	4	4	4	4	4
NG04.jpg	0	0	0	1	1
NG05.jpg	2	5	4	3	3
NG06.jpg	2	1	1	1	1
NG07.jpg	2	3	2	2	2
OK01.jpg	0	0	0	0	0
OK02.jpg	0	0	0	0	0
OK03.jpg	0	1	0	0	0
OK04.jpg	0	1	0	0	0
OK05.jpg	0	0	0	0	0
OK06.jpg	0	0	0	0	0
OK07.jpg	0	0	0	0	0
Completion rate	78.6%	64.3%	85.7%	92.9%	

The following table shows the inference results (number of colonies) and the number of correct colonies after inputting test data into each model.

Conclusion

Our results suggest that VGG-16 and YOLOv5 can identify viable cell images and count the number of viable cells. However, VGG-16 has a problem with overfitting and needs to be improved by increasing the number of learning images for image discrimination, adjusting high parameters, etc. On the other hand, YOLOv5 showed good performance in detecting viable cells.

Further research is needed to improve VGG-16 and to evaluate and expand the dataset. Evaluation of different datasets will also be an essential challenge for YOLOv5.

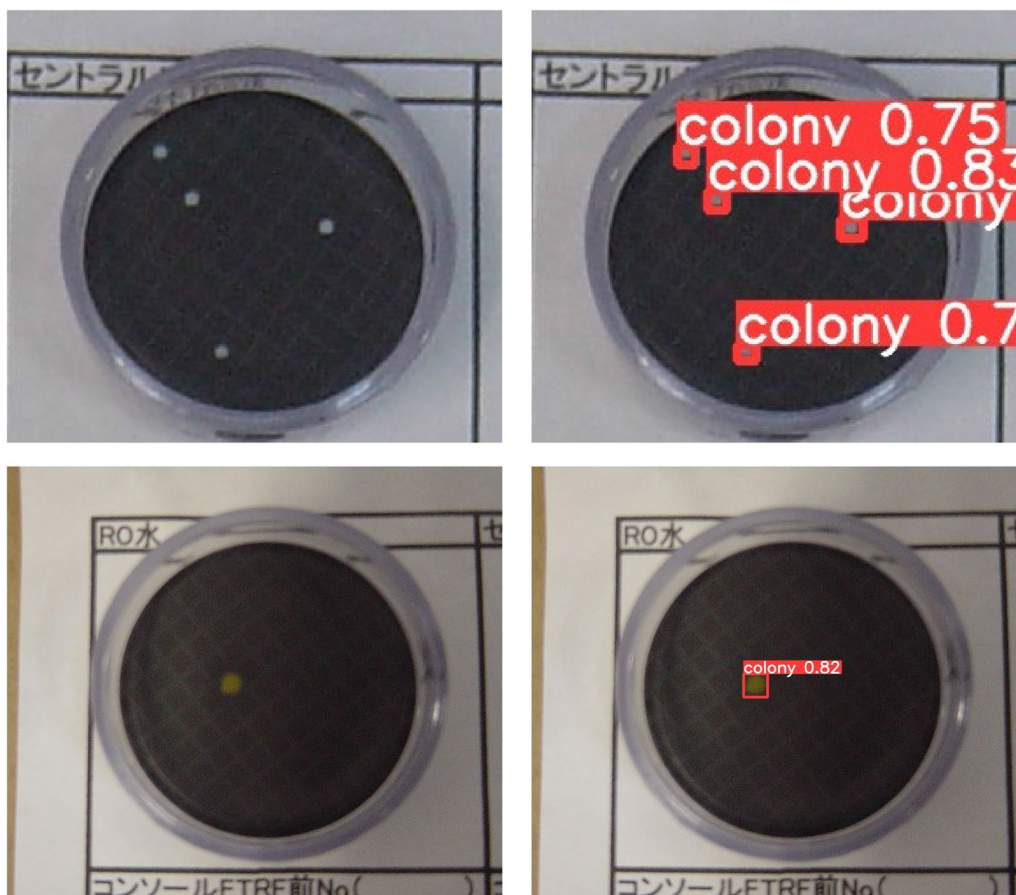


Fig. 6 Detection Images. On the left is the test image. The right is the result image of correct inference. The number is the confidence level, ranging from 0 to 1. The closer to 1, the more correct the inference result

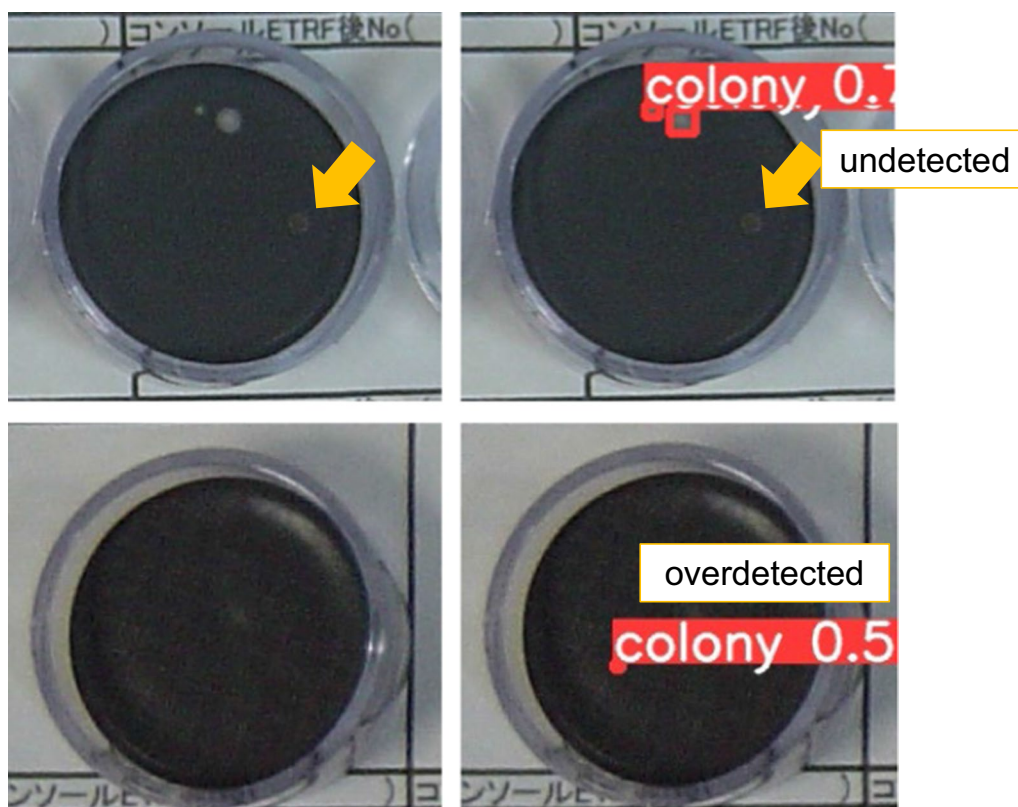


Fig. 7 Detection Failure Images. On the left is the test image. On the right is an image where inference is not appropriate. The upper image is undetected. The lower image shows the overdetected state, where the inference is that there is a colony even though there is no colony

Abbreviations

CNN	Convolutional neural network
ET	Endotoxin
EU	Endotoxin unit
GPA	Global average pooling
JSDT	The Japanese Society for Dialysis Therapy
mAP	Mean average precision
MF	Membrane filter
PR	Precision recall
SGD	Stochastic gradient descent
UPD	Ultrapure dialysis fluid
VGG-16	Visual Geometry Group 16
YOLO	You only look once

Author contributions

Not applicable.

Funding

Not applicable.

Availability of data and materials

The datasets during and/or analysed during the current study available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 15 June 2023 Accepted: 14 January 2024

Published online: 20 February 2024

References

1. Mineshima M, Kawanishi H, Ase T, Kawasaki T, Tomo T, Nakamoto H. 2016 update Japanese Society for Dialysis Therapy Standard of fluids for hemodialysis and related therapies. *Renal Replace Therapy*. 2018;4(1):1–14.
2. Hanafusa N, Abe M, Joki N, Ogawa T, Kanda E, Kikuchi K, Goto S, et al. Annual dialysis data report 2019, JSDT renal data registry. *Renal Replace Therapy*. 2023;9(1):1–37.
3. Japan Association for Clinical Engineers. Procedures for achieving dialysate water quality standards 2016 Ver 1.01. 2017.
4. Zhu G, Yan B, Xing M, Tian C. Automated counting of bacterial colonies on agar plates based on images captured at near-infrared light. *J Microbiol Methods*. 2018;153(October):66–73.
5. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1409.1556>
6. Lin M, Chen Q, Yan S. Network in network. 2013. arXiv [cs.NE]. arXiv. <http://arxiv.org/abs/1312.4400>
7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. 2015. arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1506.02640>
8. Saitoh K. Deep Learning from the basics: python and deep learning: theory and implementation. Birmingham: Packt Publishing Ltd; 2021.

9. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. 2013. arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1311.2901>
10. Mahendran A, Vedaldi A. Understanding deep image representations by inverting them. 2014. arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/1412.0035>
11. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res JMLR*. 2014;15(56):1929–58.
12. Benjumea A, Teeti I, Cuzzolin F, Bradley A. YOLO-Z: improving small object detection in YOLOv5 for autonomous vehicles. 2021. arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/2112.11798>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.