Deep Learning for Dense Colony Detection and Classification in Microbial Cultures

Kai-Yuan Hsiao, Yung-Nien Sun, and Ming-Huwi Horng

***Abstract*—Since bacteria are microscopic, they need to grow and stack on solid culture media, gradually forming colonies visible to the naked eye. In addition to identifying bacterial species, calculating the number of colony-forming units (CFU) enables estimation of the bacterial count in the original sample. CFU enumeration in culture media is particularly critical for industries such as pharmaceuticals and food production. Accurate CFU counts are essential for ensuring product safety and quality, making this process a key step in manufacturing. The used AGAR dataset involves the detection and identification of countable colonies, including Staphylococcus aureus, Bacillus subtilis, Pseudomonas aeruginosa, Escherichia coli, and Candida albicans. By leveraging the imaging characteristics of bacterial colonies on different culture media, this study integrates an improved RCNN detection model and a Transformer classifier to provide more accurate judgments.**

***Keywords*—Bacterial Colonies, Culture Medium, Convolutional Neural Networks, Multi-Scale Object Detection, Classification**

I. Introduction

To effectively detect and control bacterial infections, the **German microbiologist Heinrich Hermann Robert Koch** developed microbial culture techniques in the 19th century. He introduced the use of solid culture media instead of liquid media to prevent bacterial cross-contamination, allowing each bacterium to form distinct and easily distinguishable colonies on an appropriate solid medium. However, if the sample concentration is too high, individual colonies may not form properly. In such cases, the quadrant streaking method shown in Figure 1) can be used to gradually dilute the sample, allowing isolated colonies to form in the third or fourth quadrant of the culture plate, as illustrated in Figure 2.

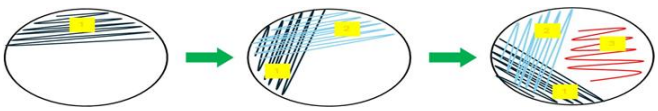

Fig. 1 Illustration of the Three-Quadrant Streaking Method

Beyond bacterial species identification, colony counting has significant applications in industries such as pharmaceuticals, food production, and contamination monitoring. The colony-forming unit (CFU) method is commonly used to estimate the bacterial count in an original sample. Accurate CFU counting is crucial for ensuring product safety and quality, as it directly indicates whether a product meets hygiene standards and helps monitor contamination risks in manufacturing processes. Consequently, automated colony counting technology has become an essential tool for improving efficiency and accuracy.

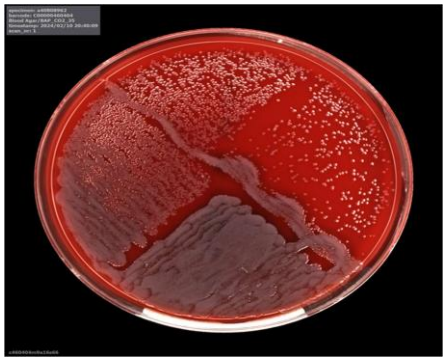

Fig. 2. Four-Quadrant Streaking Method, S. aureus

A bacterial colony is a raised structure formed by the accumulation of numerous bacterial cells. Due to variations in stacking patterns, different structural forms may emerge, as shown in Figure 3. These variations affect light refraction, leading to diverse external features of the colonies in images, as illustrated in Figure 4. A single bacterial species typically forms colonies with specific sizes and appearances [1-5]. By analyzing changes in colony brightness, density, and other characteristics, classification can be achieved.

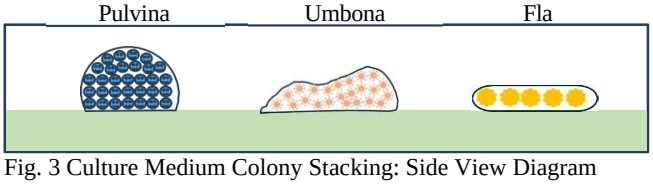

Fig. 3 Culture Medium Colony Stacking: Side View Diagram

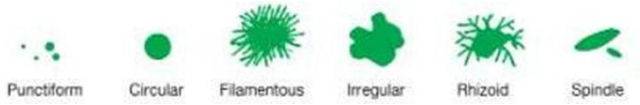

Fig. 4 Colony Morphology on Culture Medium [1]

Since colony counting requires a specialized quantification process, we selected the AGAR dataset provided by NeuroSYS

Manuscript received May 9, 2025. (Write the date on which you submitted your paper for review.) This work was supported in part by the Ministry of Science and Technology, ROC (project numbered: NSC-111-2221-E-006-233-MY3 and NSC-112-2218-E-006 -018).

KY. Hsiao, Y.N. Sun and M.H. Horng, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

to evaluate the performance of automated colony detection. The AGAR dataset was constructed following the guidelines of the Pharmacopoeia and the American Type Culture Collection (ATCC), selecting five bacterial and fungal strains as samples: *Staphylococcus aureus* (*S. aureus*), *Bacillus subtilis* (*B. subtilis*), *Pseudomonas aeruginosa* (*P. aeruginosa*), *Escherichia coli* (*E. coli*), and *Candida albicans* (*C. albicans*).

To ensure accurate colony counting, 100 microliters (µl) of the sample were extracted from the dilution solution and inoculated onto tryptic soy agar (TSA) plates using sterile glass beads. The plates were then shaken for 40 seconds to ensure uniform distribution across the culture medium. Afterward, the glass beads were removed, and the inoculated plates were incubated at 37°C for 18 to 24 hours to allow the formation of distinguishable colonies.

The AGAR dataset includes images captured at various resolutions and under different lighting conditions. However, images were deemed unusable if the number of colonies on the culture medium exceeded 300 or if colonies failed to form due to other factors. To ensure dataset consistency and adequacy, this study specifically selected high-resolution (4000 × 4000 pixels) low-light images from the AGAR dataset for experimental analysis.

II. Material and Proposed Methods

A. Related works

In early studies, colony counting was predominantly performed using traditional image processing techniques. For example, Quentin Geissmann et al. [6] proposed OpenCFU, a detection method based on a multi-step processing approach. First, they applied a median filter to de-noise each channel of the color image, reducing the impact of high-frequency noise. Next, the normalized image was converted into grayscale, and Otsu's method was used to automatically compute the optimal segmentation threshold for distinguishing colonies from the background.

For the initially segmented regions, a particle filter was employed to determine whether the region contained an object, thereby filtering out potential colonies. To address the challenge of overlapping or adjacent colonies, the distance transform was applied to the binarized image to compute the distribution of pixel distances to the nearest boundary within each target region. This step enhances the central features of each colony and provides a more precise foundation for the subsequent watershed algorithm, enabling the effective separation of overlapping or closely packed colonies.

However, OpenCFU is highly sensitive to parameter settings and image quality. Variations in colony size, non-circular colony shapes, or complex backgrounds may lead to detection errors, as shown in Figure 5.

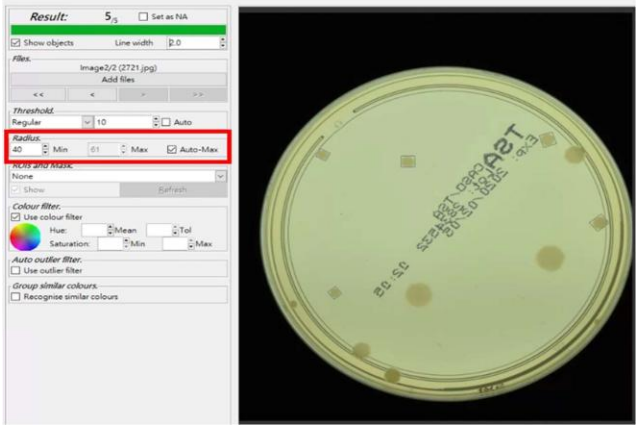

Fig. 5 OpenCFU Colony Counting

B. Materials

The AGAR dataset consists of 9,648 high-resolution, low-light images. Among them, 623 images contain no colony growth, and 3,028 images have an excessive number of colonies, making them unsuitable for colony counting (Figure 6). After excluding images with contamination or defective culture media, a total of 4,700 culture medium images were used in this study. Each culture plate contains only one identifiable type of colony. Table 1 presents the number of culture medium images for each bacterial species.

Fig. 6 Uncountable Culture Medium Images

TABLE I NUMBER OF BACTERIAL CULTURE MEDIUM IMAGES

Class	S.aureus	B.subtilis	P.aeruginosa	E.coli	C.albicans
# of Sample	900	500	1200	1200	900

In the experiment, the dataset was split into training, validation, and test sets in a 3:1:1 ratio, followed by preprocessing for each subset. The training and validation images were first cropped into multiple sub-images, and only sub-images containing colonies were selected based on the annotation files. Since the culture medium is circular, colonies never appear in the corners of the images. Additionally, as the number of colonies per plate ranges between 1 and 300, images with fewer colonies often result in sub-images without colonies.

To prevent an imbalance between positive and negative samples, all training and validation sub-images contain at least

one annotated colony. Although the test set also requires image cropping, it undergoes a post-processing step to reconstruct the sub-images and compare them with the Ground Truth. To achieve this, a sliding window strategy with a 50% overlap is applied to segment the images into sub- images, ensuring that all sub-images are retained and processed by the model for prediction. The principles of 5-fold cross-validation, we conduct five experiments, with one fold serving as the test set for final model evaluation, while the remaining four folds are further divided into three folds for training and one fold for validation. The training set is used to learn the model parameters, while the validation set monitors the training process, fine-tunes hyper-parameters, and prevents overfitting. After training, the model is evaluated on the test set to measure its generalization performance on unseen data.

C. Methods

Following the approach of Fatih Cagatay Akyon et al. [7], we adopted a sliding window mechanism to divide the original image into 768 × 768 pixel patches. To preserve as much colony information as possible along the window edges, each sliding step overlaps 50% of the original window width. This overlapping design ensures that every part of the image is effectively analyzed by the model.

To meet the input requirements of the model, the extracted image patches are resized to 384 × 384 pixels. Subsequently, the bounding boxes are reconstructed to restore their coordinates in the original image. Due to overlapping regions, the reconstruction process results in multiple duplicate bounding box predictions. To eliminate redundant or fragmented colony bounding boxes, Non-Maximum Suppression (NMS) is applied by ranking the bounding boxes based on their area and cumulative probability scores. This approach ensures accurate prediction results, as illustrated in

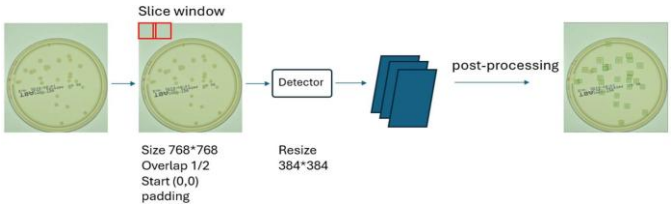

Fig. 7. Colony Detection Process

For the object detection task, we employed ResNet-50 with a Feature Pyramid Network (FPN) [8] as the feature extraction backbone (Figure 8). The scale of colonies varies significantly depending on species, concentration, and incubation time. For instance, under the same culture conditions, the size difference between C. albicans and B. subtilis colonies can reach up to 100-fold, with C. albicans colonies averaging only 40 × 40 pixels. If only the final output layer of ResNet-50 is used for feature extraction, the remaining semantic information, after passing through convolutional and max pooling layers, would be insufficient for C. albicans compared to larger colonies. Therefore, detecting small colonies requires leveraging lower-level features for better representation. In addition to colony size differences, precise localization also relies on low-level feature information.

FPN propagates high-level features through upsampling and merges them with adjacent low-level features, effectively preserving information across multiple scales. In FPN, predictions are made at each level independently using the fused feature maps. To address the Region of Interest (RoI) assignment problem across different scales, FPN ensures that larger objects are mapped to lower-resolution feature levels, while smaller objects are mapped to higher-resolution feature levels. The original paper proposed a strategy based on Equation 1, which assigns each RoI to the appropriate feature level. Here, k represents the feature level to which the RoI should be mapped, $k0$ is the reference level (typically set to 4, corresponding to $P4$ in Figure 2.5), 224 is the standard image size used in ImageNet pre- trained models, and w and h denote the width and height of the RoI, respectively.

$$k = ⌊k_0 + \log_2 \sqrt{wh}/224⌋ \quad (1)$$

The size of the Region of Interest (RoI) is highly dependent on the size of the input image. Using 224 as the reference standard may introduce bias in object scale estimation, thereby affecting the accuracy of RoI assignment. Additionally, semantic information between non-adjacent layers cannot be effectively integrated. For example, the information from C5, after undergoing three upsampling operations, becomes diluted when reaching C2, resulting in suboptimal utilization of multi-scale features. To address these issues, we incorporated the Channel Attention Mechanism [9] into the feature extraction process. By applying channel-wise weighting to each feature map, this mechanism enhances the representation of multi-scale features.

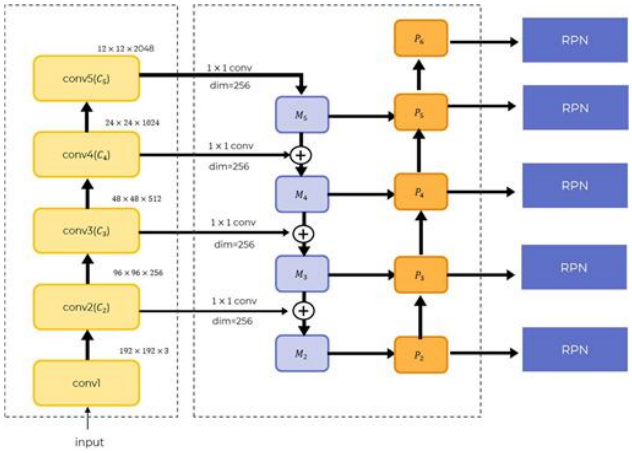

Fig.8 Feature Extraction Network Architecture

The core function of the Channel Attention Mechanism is to dynamically weight the channels within each feature map, enhancing features relevant to object detection while suppressing interference from irrelevant channels. Specifically, Global Average Pooling (GAP) is first applied to compress the spatial dimensions of the feature map, generating a global feature descriptor for each channel. The global feature is then passed through two fully connected layers, where the ReLU and Sigmoid activation functions generate channel-wise

attention weights. Finally, the computed channel weight Wc is applied to the input feature map, producing a refined, weighted feature map. This process further enhances the feature representation of candidate regions, allowing the model to focus more effectively on object-related information.

In the colony recognition task, we apply the Conformer architecture, which combines local features captured by convolutional networks with global semantic information provided by Transformers. Compared to T2T-ViT, which solely relies on Transformers, Conformer's convolutional module exhibits stronger capability in extracting fine-grained local details, such as colony edges and textures. This allows it to learn effective features more quickly in small-data scenarios. Meanwhile, its built-in self-attention module effectively captures the global dependencies of colonies within the image.

Within the MLP module of the Transformer, we enhance the traditional two fully connected layers and activation function by introducing Swish-Gated Linear Units (SWIGLU) [10]. This modification improves the nonlinear feature representation and enhances the overall model performance. SWIGLU is an improved version of the standard Gated Linear Units (GLU), utilizing a gating mechanism to suppress redundant or irrelevant features while amplifying important target-related features through the Swish activation function (Equation 2). This enables the effective integration of feature selection and nonlinear transformation (Equation 3), where σ functions as a gating operation to filter input features, and $W1$, $W2$, $b1$, $b2$ represent the weight and bias matrices, respectively.

$$Swish\,(x) = x \cdot sigmoid(\beta x) \quad (2)$$

$$SWIGLU(X) = (\,\sigma(XW1 + b1) \odot Swish\,(XW2 + b2)\,) \quad (3)$$

Position Encoding is used to supplement the self-attention mechanism by enhancing its ability to perceive the positional relationships between elements within a sequence. Since the self-attention mechanism inherently performs global association computations across all elements in a sequence, position encoding is introduced to encode positional information, enabling the model to capture the structural dependencies within the sequence.

Among various approaches, relative position encoding emphasizes the relative distance between any two elements in a sequence. By incorporating relative position encoding into each attention layer, the model can dynamically adjust the attention distribution based on the relative relationships between elements. This enhances the model's ability to accurately represent structural information and long-range dependencies within the sequence.

In the colony counting experiment, we adopt a sliding window approach to segment the culture medium images into multiple sub-images, which are then processed individually by the model. The final detection results for the entire culture medium are obtained through post-processing of all sub-image predictions. To minimize disruption to the original image features, a larger window size is preferred to preserve more fine-grained details. However, considering computational resource constraints and the impact of colony scale variations, this study crops images into 768 × 768 sub-images and resizes them to 384 × 384 during model inference. This approach achieves an optimal balance between resource efficiency and detection performance.

III. Experimental Results

This experiment was conducted on Ubuntu 20.04, with preprocessing and post-processing procedures written in Python, and the deep learning model implemented using the PyTorch framework. The hardware specifications include an Intel(R) Core(TM) i7-13700K CPU @ 5.40GHz and an NVIDIA GeForce RTX 4090 GPU with 24GB of memory. Detection networks are trained for 36 epochs with the AdamW optimizer, a batch-size of 4, and. the learning rate starts at $1 \cdot 10{-}3$ and decreases by a factor of 10 after 27 and 33 epochs.

Classification networks were trained for 300 epochs using the AdamW optimizer with a batch size of 8 and a weight decay of 0.05. The initial learning rate was set to 0.001 and adjusted using a cosine decay schedule.

The Confusion Matrix is a crucial tool for computing various metrics in object detection tasks. Its primary purpose is to analyze the model's prediction performance across different classes by categorizing each detection result.

Table 2 presents the final prediction results for colony detection. Our objective is to ensure that the predicted colony count closely matches the ground truth while maintaining high precision, thereby preventing an excessive number of incorrect bounding boxes that could mislead the experiment.

When the IoU threshold in the post-processing algorithm is increased, precision tends to improve, but recall may significantly decrease. To achieve a balance between precision and recall, we set the IoU threshold to 0.7 in the post-processing algorithm.

TABLE II. EVALUATION METRICS FOR EACH BACTERIAL SPECIES

	Recall (%)	Precision (%)
S.aureus	96.84±0.28	96.29±0.25
B.subtilis	96.81±0.24	97.43±0.22
P.aeruginosa	95.63±0.64	95.68±0.73
E.coli	96.58±0.17	97.32±0.26
C.albicans	93.45±0.38	94.44±0.48
Average	95.82±0.32	96.23±0.35

The number of RoI directly affects the overall computation time. Reducing the number of RoI can effectively decrease training time and resource consumption. To achieve this, we experimented with the Guided Anchor method, which allows the model to automatically generate anchors and filter potential regions of interest, further reducing RoI numbers to enhance efficiency. The core idea of Guided Anchor [11] is to learn to generate high- quality anchors rather than relying on predefined anchor configurations. Its advantage lies in its adaptability to different datasets, reducing the burden of

hyperparameter tuning.

However, experimental results indicate that Guided Anchor has significant drawbacks in object detection, particularly in handling densely overlapping objects, where detection accuracy is noticeably insufficient. Based on these issues, we instead adopted Sparse R-CNN [12], which localizes and classifies objects using a sparse set of queries and applies the Hungarian algorithm for object matching. This approach eliminates the need for a dense detector that generates numerous candidate boxes and requires NMS post-processing. However, its performance remains suboptimal in certain critical scenarios.

First, the query mechanism relies on random initialization in its early stages, leading to query distributions that deviate from actual object locations, further affecting training effectiveness and overall model performance. Additionally, the limited number of sparse proposal boxes imposes restrictions on detection tasks that require high coverage in dense scenes. This is particularly problematic for small-object detection, where significant improvement is still needed.

To address the low recall rate of Sparse R-CNN, we further adopted DDQ R-CNN [13], which integrates the concept of Guided Anchor to improve recall through a dense query mechanism. Additionally, feature enhancement techniques were introduced to mitigate the challenges of detecting low-contrast targets. Furthermore, DDQ R-CNN simplifies the iterative refinement process, reducing computational overhead.

In comparison, Deformable DETR [14] introduces a deformable attention mechanism, which effectively handles multi-scale features and densely distributed objects. Unlike Sparse R-CNN, which depends on randomly initialized queries, Deformable DETR generates reference points from the global features of the Encoder, significantly reducing the impact of initial query bias and enhancing detection stability. Additionally, it employs an Iterative Bounding Box Refinement strategy, in which each Decoder layer progressively refines the reference points, reducing misalignment with ground truth boxes and improving recall. Furthermore, Deformable DETR is not constrained by a fixed number of queries, allowing it to flexibly adapt to various object distributions. As a result, it outperforms Sparse R-CNN in both detection accuracy and generalization ability.

Although traditional RPN generates a large number of candidate boxes, it maximizes coverage of all potential targets, ensuring that colonies are not missed due to an insufficient number of proposals. Ultimately, in NWD-RPN, we set k=2, meaning that for each candidate box, only the top-k RoIs closest to the target are selected. This approach effectively reduces the number of candidate boxes, lowering computational costs while preserving the most relevant object features. Consequently, it maintains high detection accuracy without significantly affecting model performance. This strategy demonstrates that in resource- constrained environments, moderately reducing the number of RoIs has minimal impact on performance robustness. Table 3 presents the results.

TABLE III. DETECTION METRICS FOR DIFFERENT MODELS

	Recall (%)	Precision (%)
FasterRCNN with Guided Anchor	92.18	92.74
Sparse-RCNN	93.36	93.62
DDQ RCNN	93.57	93.97
Deformable DETR	95.24	94.98
Ours	95.82	96.23

IV. CONCLUSIONS

In this study, we proposed automated methods for various colony detection and classification tasks, leveraging deep learning techniques to significantly reduce the time required for traditional manual inspection. These advancements provide a fast, efficient, and accurate solution for clinical microbiological detection.

For colony detection, we experimented with different R-CNN architectures and introduced improvements to Faster R-CNN, particularly addressing its limitations in small- object detection. By incorporating NWD into the Region Proposal Network (RPN), we enhanced the model's ability to detect small colonies. Additionally, we refined the multi- scale RoI selection strategy, enabling the model to focus more precisely on clinically significant colony regions. Lastly, we integrated a Transformer-based detection head, which, according to our experimental results, effectively improved detection performance and achieved a recall rate exceeding 95%, demonstrating its potential for clinical applications.

For colony classification, we considered variability among bacterial species as well as genotypic differences within the same species. We adopted an improved Conformer model as the core architecture, enhancing the MLP module to better handle complex colony classification tasks. This approach yielded excellent performance in the classification of E. coli, K. pneumoniae, and S. aureus. Additionally, we found that directly classifying the entire culture medium image helped mitigate the impact of localized anomalies (e.g., uneven bacterial growth or contaminant interference), significantly improving classification accuracy and demonstrating robust classification capability.

In clinical research, as the sample size increases and bacterial diversity expands, we anticipate facing more complex challenges in the future. These challenges extend beyond colony identification to include the ability to classify mixed bacterial samples within the same specimen. We aim to develop more robust models to further enhance adaptability to complex clinical samples.

The visual characteristics of bacterial colonies can vary due to a range of external factors, such as the bacterial growth process on the culture medium, the angle of light projection, and the parameter settings of imaging equipment. These factors may lead to variations in colony morphology, color, and lighting in the images, thereby increasing the difficulty of classification. Consequently, future research should focus on establishing comprehensive standardization protocols, including uniform conditions for image acquisition, light

source calibration, and standardized bacterial culture procedures. These measures will help reduce the impact of external variables on model performance and ensure the stability of image data.

To address the challenges of dense colony detection, we believe that combining the Transformer architecture with the Hungarian Algorithm represents a promising research direction. The Hungarian Algorithm, known for its effectiveness in optimizing bounding box matching, can improve the precise localization of dense colonies. By refining the bounding box annotations for dense colonies, we aim to further validate and enhance detection accuracy in future studies.

ACKNOWLEDGMENT

The authors thank the Ministry of Science and Technology, ROC (project numbered: NSC-111-2221-E-006-233-MY3 and NSC-112-2218-E-006 -018) for supporting this work.

REFERENCES

[1] J.M. Willey, L.M. Sherwood and C.J. Woolverton, Harley and Klein's Microbiology. 7th Edition, McGraw-Hill. Higher Education, p. 133-138, 2008.

[2] J. Freeman, et al., The changing epidemiology of Clostridium difficile infections. Clin Microbiol Rev, p. 529-49, 2010. https://doi.org/10.1128/CMR.00082-09

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M, Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv, abs/2010.11929, 2020.

[4] Alessandro Ferrari, Stefano Lombardi, Alberto Signoroni,Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging,Pattern Recognition,Volume 61, Pages 629-640,ISSN 0031-3203, 2017. https://doi.org/10.1016/j.patcog.2016.07.016

[5] S. Majchrowska, J. Pawlowski, G. Gula, T. Bonus, A. Hanas, A. Loch, A.S. Pawlak, J. Roszkowiak,, T. Golan, & Z. Drulis-Kawa. AGAR a microbial colony dataset for deep learning detection. ArXiv, abs/2108.01234, 2021.R. A. Scholtz, "The Spread Spectrum Concept," in *Multiple Access*, N. Abramson, Ed. Piscataway, NJ: IEEE Press, 1993, ch. 3, pp. 121-123. https://doi.org/10.21203/rs.3.rs-668667/v1

[6] Q. Geissmann, OpenCFU, a new free and open-source software to count cell colonies and other circular objects. PloS one, 8(2), e54072. https://doi.org/10.1371/journal.pone.0054072, 2013. https://doi.org/10.1371/journal.pone.0054072

[7] F.C. Akyon, S.Q. Altinuc, & A. Temizel, Slicing Aided Hyper Inference and Fine- Tuning for Small Object Detection. 2022 IEEE International Conference on Image Processing (ICIP), 966-970, 2022. https://doi.org/10.1109/ICIP46576.2022.9897990

[8] T. Lin, P. Dollár, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 936-944, 2016. https://doi.org/10.1109/CVPR.2017.106

[9] S. Woo, et al. CBAM: Convolutional block attention module. in Proceedings of the European conference on computer vision (ECCV). 2018.

[10] N.M. Shazeer, GLU Variants Improve Transformer. ArXiv, abs/2002.05202, 2020. J. P. Wilkinson, "Nonlinear resonant circuit devices," U.S. Patent 3 624 12, July 16, 1990.

[11] J. Wang, K. Chen, S. Yang, C.C. Loy, & D. Lin, Region Proposal by Guided Anchoring. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2960-2969, 2019. https://doi.org/10.1109/CVPR.2019.00308

[12] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, & P. Luo, Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14449-14458, 2020. https://doi.org/10.1109/CVPR46437.2021.01422

[13] S. Zhang, W. Xinjiang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo & K. Chen, Dense Distinct Query for End-to-End Object Detection. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7329-7338, 2023. https://doi.org/10.1109/CVPR52729.2023.00708

[14] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, & J. Dai, Deformable DETR: Deformable Transformers for End-to-End Object Detection. ArXiv, abs/2010.04159, 2020.