Transformer-based Detection of Microorganisms on High-Resolution Petri Dish Images

Nikolas Ebert[1,2] Didier Stricker[2] Oliver Wasenmüller[1]
[1]Mannheim University for Applied Science, Germany
[2]RPTU Kaiserslautern-Landau, Germany
n.ebert@hs-mannheim.de, didier.stricker@dfki.de, o.wasenmueller@hs-mannheim.de

Abstract

Many medical or pharmaceutical processes have strict guidelines regarding continuous hygiene monitoring. This often involves the labor-intensive task of manually counting microorganisms in Petri dishes by trained personnel. Automation attempts often struggle due to major challenges: significant scaling differences, low separation, low contrast, etc. To address these challenges, we introduce AttnPAFPN, a high-resolution detection pipeline that leverages a novel transformer variation, the efficient-global self-attention mechanism. Our streamlined approach can be easily integrated in almost any multi-scale object detection pipeline. In a comprehensive evaluation on the publicly available AGAR dataset, we demonstrate the superior accuracy of our network over the current state-of-the-art. In order to demonstrate the task-independent performance of our approach, we perform further experiments on COCO and LIVECell datasets.

1. Introduction

Regulatory bodies such as the European Medicines Agency (EMA) and the U.S. Food and Drug Administration (FDA) mandate strict guidelines for continuous hygiene monitoring in the pharmaceutical, cosmetics and food industries. As a result, a large number of Petri dishes must be examined for microbial colonies on a daily basis by experienced biologists, which is time-consuming and error-prone. Automating this process presents several challenges. One is the high resolution required to reliably detect tiny colonies. Another is that colonies vary widely in size and shape and can overlap, making automated detection difficult (see Figure 1). There are several open-source approaches [17, 22, 39] that use classical computer vision techniques such as image filters and intensity variations to differentiate colonies from the agar-medium. However, these processes are based on hand-crafted features and laborious to use.

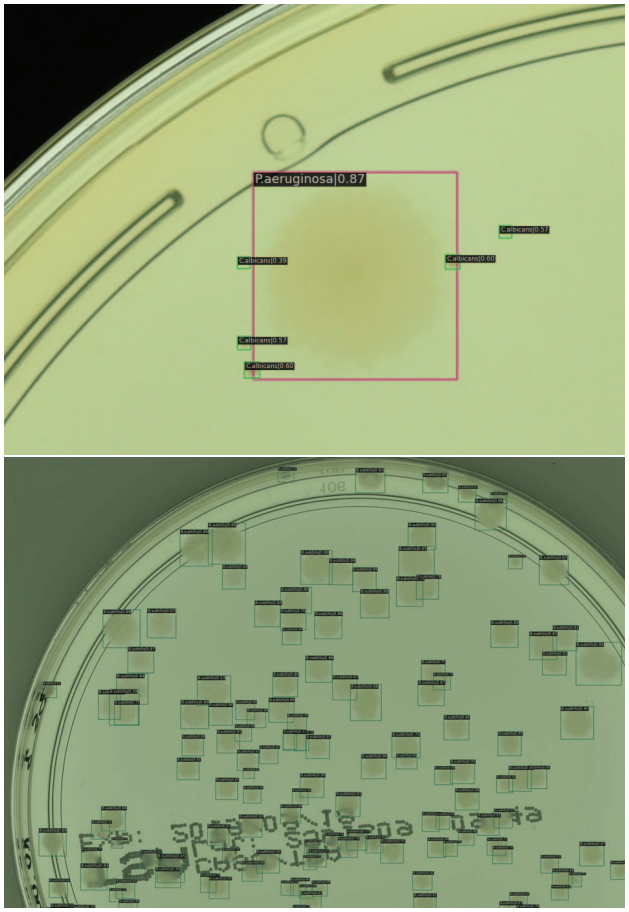


Figure 1: The biggest challenges in hygiene monitoring are the detection of particularly small organisms, the significant variation in colony size, low contrast between foreground and background, as well as a high number of colonies with large overlap. The images show typical inference results of our method on the test data.

Colony detection can be automated through the use of neural networks, such as Faster-RCNN [29], which have proven to be more accurate and robust than traditional

computer vision methods. Recently, transformer networks [40] were introduced, outperforming their convolutional-counterparts in most tasks [28, 46]. This success is partly due to the self-attention mechanism, which enables transformers to model information spatial dependencies within large receptive fields. A drawback of standard self-attention is its quadratic complexity, resulting in large memory requirements and computational costs, especially when applied to high-resolution images for hygiene monitoring.

In this paper, we present an innovative approach to colony detection in the field of computer vision. Our method, called AttnPAFPN, leverages a novel efficient-global self-attention mechanism to improve the performance of a path aggregation feature pyramid network (PAFPN) [26] for object detection. In combination with further optimizations, our efficient-global self-attention achieves superior accuracy and performance, especially when processing high-resolution images. Furthermore, we introduce new high-resolution prediction-heads to improve the detection of tiny objects. A hallmark of our AttnPAFPN is its flexibility, as it can be integrated into almost any top-down object detection method. To demonstrate this flexibility, we integrate our method into two general object detectors [15, 33, 19]. Augmented with our AttnPAFPN, these networks show superior performance in terms of accuracy over the current SoTA on the AGAR dataset [29] for colony detection. In addition, we include an extensive ablation study of our method with varying image resolutions. To demonstrate the task-independent performance of our approach, we also conduct experiments on COCO [25] for general object detection and on LIVECell [13] for the segmentation of cells in microscope images.

2. Related Works

2.1. Detecting colonies

Automated colony counting has been of interest since the late 1950s [1, 30]. Nowadays, there are several tools available, such as OpenCFU [17] and AutoCellSeg [39], which assist in the detection of microorganisms, based on conventional computer vision methods. The main drawback of these tools is their limited automation, requiring handcrafted features for colony detection. Setting these features requires expert knowledge, similar to manual counting.

In addition to these conventional methods, several deep learning-based approaches [14, 16, 29, 18, 36] have been proposed for detecting colonies of microorganisms on agar plates. Ferrari et al. [16] utilize convolutional neural networks (CNNs) for bacterial classification, resulting in significant improvements compared to handcrafted feature-based support vector machine (SVM) systems. Andreini et al. [2] use k-means clustering to perform foreground-background segmentation, rather than classification or counting colonies. Multiple methods [4, 14, 32] approach colony detection by using modified U-Net [35] structures. Mask-RCNN [19] has also been adapted multiple times [27, 31] for detecting and segmenting microorganisms in agar dishes. Majchrowska et al. [29] used an image-patch approach, dividing high-resolution images into smaller overlapping areas to perform individual object detection [6, 33] and then merging the resulting bounding boxes. However, a common drawback of these methods is that they were developed either for low-resolution images or for image slices.

2.2. Object detection

In recent years, deep learning approaches have made significant progress in the field of object detection [24, 33, 10], outperforming classical methods by a large margin, highlighting the potential of the current SoTA to improve accuracy and speed of colony detection. Two stage detectors such as Faster-RCNN [33] and its variants [6, 19] first define regions of interest and then perform object detection. RetinaNet [24] introduced Focal loss to address the class imbalance problem in one-stage detectors. FCOS [38] and VariFocalNet [44] locate objects of interest by using anchor points and point-to-boundary distances. TOOD [15] presented a task-aligned learning strategy for explicitly aligning the two tasks of classification and localization in a learning-based manner. All these methods have in common that they focus on the prediction-head. As a neck, a Feature Pyramid Network (FPN) [23] is usually used to improve accuracy by creating multi-scale features. The Path Aggregation Feature Pyramid Network (PAFPN) [26] extends the FPN approach by adding a bottom-up path to enhance FPN features with accurate localization signals from low levels. YOLOv4 [5] introduces further bottlenecks into the PAFPN for more diverse representations. ResFPN [34] enhances FPN by integrating multiple residual skip connections to leverage information from higher scales for stronger and more localized features. The transformer-based DETR [7, 46] works entirely without FPN and achieves still SoTA-results. The methods mentioned are designed for the COCO dataset [25], which is known for its diversity and mainly consists of medium-sized images and objects. Therefore, the benchmark does not adequately represent the challenge of high-resolution hygiene monitoring, with its numerous tiny colony growths and homogeneous backgrounds. Accordingly, the aforementioned methods are only conditionally suitable for solving the task of colony detection.

To address these drawbacks, we investigate cutting-edge object detection techniques and incorporate a specialized Attention-based Path Aggregation Feature Pyramid Network (AttnPAFPN) for high-resolution feature extraction in order to detect colonies on agar dishes (see Figure 2). The goal of our work is to provide a solution specific to the chal-

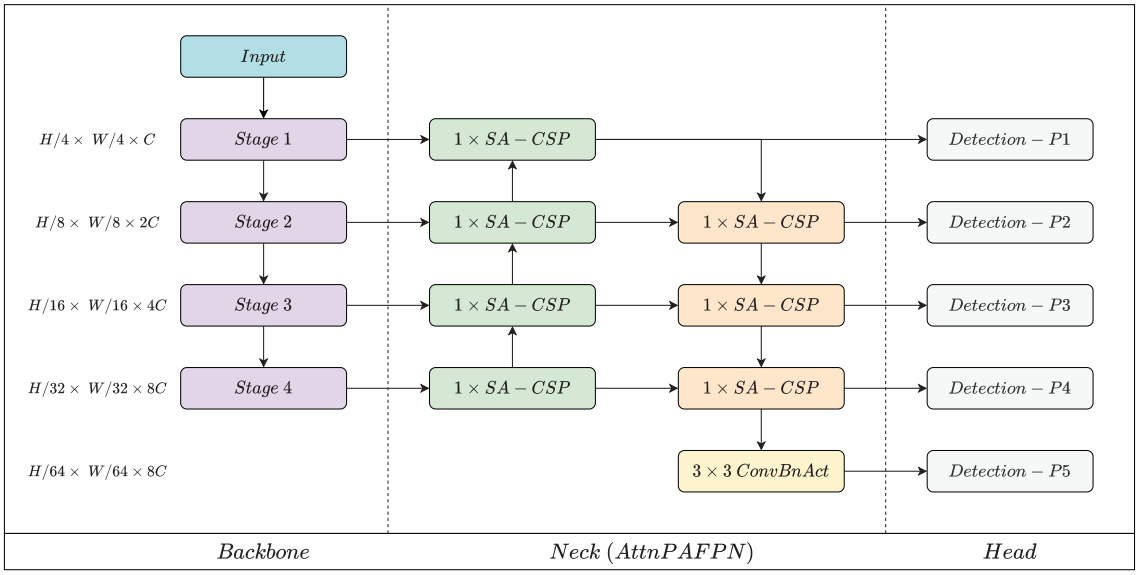


Figure 2: **Architecture overview.** Our object detection network consists of a backbone network, a neck and a prediction-head. We use our AttnPAFPN as the neck, which consists of self-attention extended CSP-Bottlenecks (SA-CSP). Almost any method can be used for the final prediction by the head (e.g. TOOD [15]).

lenges of colony detection, improving both the accuracy and efficiency compared to SoTA methods.

3. Method

This section outlines the design choices of our proposed AttnPAFPN to specifically address the limitations of current SoTA methods in processing high-resolution images. The proposed detection network consists of three key components: a backbone for extracting image features from the input, our neck (AttnPAFPN) for generating a hierarchical feature representation at different scales, followed by a detection head for the final predictions (e.g. TOOD [15]).

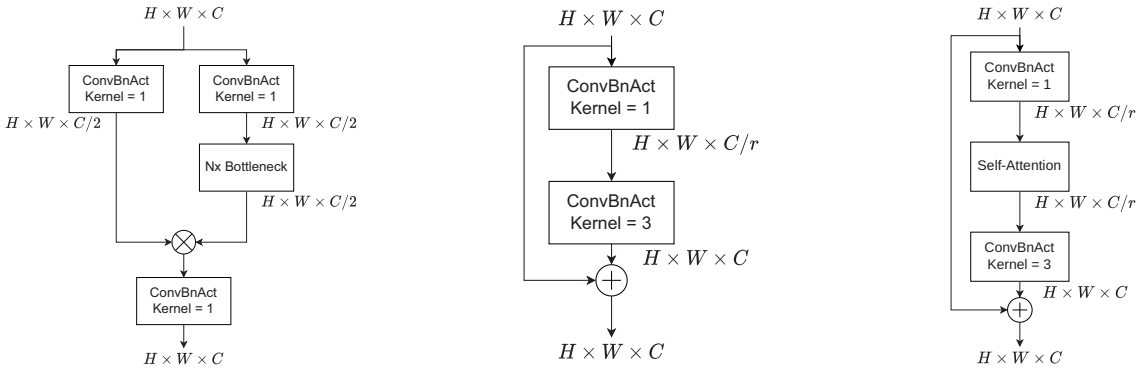
3.1. AttnPAFPN

Our primary contribution is the novel AttnPAFPN network neck, tailored to high-resolution images and small objects. AttnPAFPN utilizes our efficient-global self-attention mechanism and a new high-resolution output, allowing the network to focus on essential features, even for extremely small objects. Our streamlined method is further optimized using concepts from CSP-Net [41], resulting in improved performance, lower parameter counter, and reduced complexity. The end-to-end trainable encoder-decoder is shown in Figure 2.

At the initial stage of our AttnPAFPN, we use the lowest resolution backbone features (e.g. with a total stride of 32). These features are passed through a CSP-Bottleneck block to create high-level features, which are then used in both the top-down and bottom-up pathways. In the top-down pathway, the features are first upsampled by a factor of 2, then concatenated with the backbone features of corresponding size, before being processed again by a subsequent CSP-Bottleneck. This process is repeated until the last stage (stride of 4) is reached, which enables AttnPAFPN to recognize tiny objects due to its high-resolution features. To reduce computational complexity and the number of parameters, we compress the depth of the backbone features by applying a 1×1 convolutional layer before passing them to the feature pyramid. The bottom-up path of our AttnPAFPN also utilizes CSP-Bottleneck blocks, but instead of upsampling, a strided convolutional layer is used to process the features. This path also includes a final strided 3×3 convolutional layer to generate an output with a factor of $\frac{1}{64}$ of the original image size and enables the network to recognize large objects. Our final AttnPAFPN predicts objects at five different scales, with total strides of $\{4, 8, 16, 32, 64\}$.

3.2. Self-Attention augmented CSP-Bottlenecks

One of the key contributions of our work is the integration of transformers [9, 40] into CSP-Bottlenecks [41] (Figure 3a,3c), similar to the approach taken by BoTNet [37] integrating transformers into ResNet for image classification [20]. The structure of CSP-Bottlenecks can be seen in Figure 3a. First, the incoming featuremaps are divided into two parts in depth. The first part is passed directly to the output after a single pointwise-convolution operation. The other half is processed N times by a residual bottleneck (see Figure 3b) and then concatenated with the first half. Finally, a pointwise convolution is performed to enable



(a) CSP-Bottleneck (b) Bottleneck (c) SA-Bottleneck

Figure 3: **Illustrations of the used network modules.**

communication between the channels. To integrate self-attention mechanisms into these structure, we replace the convolutional bottleneck with our self-attention augmented version (see Figure 3c. However, the use of standard self-attention is limited by its quadratic complexity, especially when applied to high-resolution images, such as those of hygiene monitoring. To address this challenge, we compare two resolution-optimized transformers: our novel efficient-global self-attention and local-window self-attention similar to Swin Transformer [28]. In general, a transformer-layer [40] can be described as

$$\begin{aligned} y^* &= \text{Self-Attention}(\text{LN}(x)) + x, \\ y &= \text{FFN}(\text{LN}(y^*)) + y^*, \end{aligned} \tag{1}$$

with x as its input and y as output features. LN refers to layer normalization [3], and FFN to a linear feed-forward layer. Self-attention [28] can be formulated as

$$\text{Self-Attention}(q, k, v) = \text{Softmax}(\frac{qk^T}{\sqrt{d}} + b)v, \tag{2}$$

where q, k, v are query, key and value matrices generated from input-features, d is a scaling factor and b is a trainable relative position bias term. Inspired by SegFormer [43], we extend our feed-forward-network (FFN) by CNN-layers, adding an inductive bias for finer localization using additional positional information:

$$\begin{aligned} y^* &= \text{GeLU}(\text{LN}(\text{PWConv}(x))), \\ y &= \text{PWConv}(\text{GeLU}(\text{LN}(\text{DWConv}_{3\times3}(y^*)))) + x, \end{aligned} \tag{3}$$

where GeLU [21] corresponds to Gaussian Error Linear Unit activation, PWConv to a point-wise convolution and $\text{DWConv}_{3\times3}$ to a depth-wise 3×3 convolution.

Local-window self-attention splits input features into non-overlapping windows with limited receptive fields, before applying multihead self-attention. As a result, the computational effort of self-attention is linear to the window-size. One downside is that information cannot pass between the windows within a layer. Several successive layers with shifting windows is necessary to create a global receptive field. In our experiments we follow the window partitioning strategy of Swin Transformer [28]. In contrast, our efficient-global self-attention reduces the spatial resolution of the input to a fixed global size by performing adaptive max-pooling on the input. The size of the global window is freely selectable, but we have set the window size in all our networks to $\frac{1}{64}$ of the original resolution. In case of 1024×1024 resolution, the fixed global window would be 16×16. Regardless of the input resolution, it is also possible to set the global window to a fixed size. This results in a network complexity that is completely independent of the image resolution. With our efficient-global self-attention we create a single window with a global receptive field to which self-attention is subsequently applied. These and many more transformer variants can be easily inserted into the bottleneck structure as shown in Figure 3c.

4. Evaluation

Our evaluation focuses on demonstrating the benefits of our AttnPAFPN in high-resolution object detection. For this purpose we use the public AGAR dataset [29], containing high-resolution images of five different types of bacteria on agar plates. The data is divided into the higher-resolution (HR) and lower-resolution subsets (LR). The HR subset contains approximately 5k training images and 2k test images, with a resolution of around $4,000^2$ pixels. The LR subset has around 3.5k training images and 1k test images with a resolution of $2,048^2$ pixels. In addition to these two subsets, a third mixed-resolution subset is created by combining both subsets.

We perform an ablation study to determine the effect individual components have on our methods accuracy. Results are shown in Tables 1 and 2. Furthermore, we implement our AttnPAFPN in current SoTA methods (e.g. TOOD

Table 1: **Ablation study** on the effectiveness of the components of our AttnPAFPN. TOOD [15] is used as head for evaluation on AGAR-dataset [29]. We evaluated onthe high-resolution and low-resolution subset with a image-size of 1536×1536.

Method	Params	HR-Subset				LR-Subset			
		mAP	AP^{50}	AP^{75}	R^{50}	mAP	AP^{50}	AP^{75}	R^{50}
TOOD-Baseline [15]	**32.0 M**	57.7	82.6	68.2	83.0	67.2	95.9	80.0	96.5
+ CSP-PAFPN	74.3 M	63.3	90.2	74.8	90.8	68.0	95.8	81.1	96.5
+ Attn-Bottleneck	64.4 M	66.5	95.3	78.1	96.0	69.1	97.5	82.6	98.3
+ Extra Detection-Scales	66.3 M	67.7	96.1	80.3	**96.8**	**69.7**	**98.0**	**83.8**	**98.9**
+ Feature-compression-layer	32.8 M	68.1	96.2	81.0	**96.8**	69.3	97.6	82.8	98.4
+ Multiscale-Training	32.8 M	**68.2**	**96.3**	**81.1**	**96.8**	69.5	**98.0**	83.4	98.6

Table 2: **Ablation study** on the effectiveness of the local-window (v1) and efficient-global self-attention (v2) for our AttnPAFPN. TOOD [15] is used as the detection head for evaluation on the AGAR-dataset [29]. We evaluated on both subsets with a image-size of 1536×1536.

Method	Params	HR-Subset				LR-Subset			
		mAP	AP^{50}	AP^{75}	R^{50}	mAP	AP^{50}	AP^{75}	R^{50}
TOOD-Baseline [15]	**32.0 M**	57.7	82.6	68.2	83.0	67.2	95.9	80.0	96.5
+ AttnPAFPNv1	84.1 M	66.5	95.1	78.5	95.7	69.0	97.4	82.6	98.1
+ Extra Detection-Scales	87.0 M	67.2	95.7	79.5	96.5	**69.7**	**98.0**	83.3	98.8
+ AttnPAFPNv2	64.4 M	66.5	95.3	78.1	96.0	69.1	97.5	82.6	98.3
+ Extra Detection-Scales	66.3 M	**67.7**	**96.1**	**80.3**	**96.8**	**69.7**	**98.0**	**83.8**	**98.9**

[15]) and compare it with five different object detection models, listing the results in Table 3. All methods are implemented in the MMDetection-Framework [8] and we use the mAP metric to evaluate their performance. mAP provides a comprehensive assessment of accuracy and recall, averaging the maximum precision score for each recall value of all classes. In hygiene monitoring, detecting all colonies is a priority over precise localization. Hence, we use the Recall at an IoU threshold of 0.5 (R^{50}) as an additional metric.

4.1. Ablation Study

In our first experiment, we assess the impact of our method by comparing AttnPAFPN with a baseline model (TOOD [15] + FPN [23]) as shown in Table 1. All networks are trained for 20 epochs with the SGD optimizer, a batch-size of 8, and use a pre-trained ResNet50 [20] as their backbone. The learning rate starts at $5 \cdot 10^{-3}$ and decreases by a factor of 10 after 8 and 16 epochs.

Replacing the standard FPN in TOOD with the convolutional CSP-PAFPN leads to an improvement in mAP ($+5.6/+0.8$) and Recall ($+7.8/\pm 0$), but also increases the number of parameters by more than 100%. By introducing our efficient-global self attention (SA) into the CSP-Bottlenecks, we were able to reduce the parameters by over 15% and further boost mAP ($+3.3/+1.1$) and R^{50} ($+5.2/+1.8$) compared to the previous step. In these initial experiments, all network necks use only the backbone scales $\{8, 16, 32\}$ for predictions. To ensure better recognition of particularly large and tiny colonies, we add two more scales, so that we ultimately perform detection across five resolutions: $\{4, 8, 16, 32, 64\}$. To address the heavyweight nature of our network, we implemented 1×1 feature-compression layers in our AttnPAFPN, reducing the depth C of backbone-features to $C^* = 256$. Through this feature reduction our method achieves a parameter count comparable to the baseline FPN, while still achieving a stronger performance in terms of mAP and R^{50}. For a final increase in performance, we utilize multi-scale training. Overall AttnPAFPN increases mAP by $+10.5/+2.3$ and Recall by $+13.8/+2.1$ in comparison to the baseline.

In Section 3.1 of our study, we present two variants of efficient transformer layers that are specifically designed for high-resolution images. Table 2 compares the performance of local-window SA (v1) and efficient-global SA (v2). The results indicate that efficient-global SA, which provides a coarse-grained overview of the entire image, leads to a significant improvement in accuracy. The differences in mAP are only marginal on the HR subset; on the LR subset, both networks achieve almost identical accuracy. The decisive point here is the significantly lower complexity and the lower number of weights of the global self-attention.

4.2. Quantitative Evaluation

In our final experiment, as listed in Table 3, we compare the performance of our proposed method, AttnPAFPN, with SoTA object detection methods [15, 24, 33, 44, 46]. The training process of all the networks is equal to the description in Section 4.1. The first few rows which are titled with "Patches: 512×512" present the results of Majchrowska et al. [29]. They divide the images into patches of size 512×512 and then detect the colonies in each of these patches using Faster-RCNN [33] and Cascade-RCNN [6] with ResNet50 [20] as the backbone, similar to our setup. The following lines contain the results of the SoTA and our method using the full image under different resolutions. Upon comparison with Faster-RCNN, our AttnPAFPN shows lower performance for lower resolution, especially 1024×1024 for the HR-Subset. However, as the resolution increases, AttnPAFPN outperforms all baselines by a large margin. Furthermore, our AttnPAFPN

Table 3: **Comparison of detection accuracy** on the AGAR [29] validation set. Different SoTA methods [15, 24, 33, 44, 46] are compared with our approach. We use TOOD [15] and Faster-RCNN [33] as head. The comparison is performed with different data-subsets.

Method	Params	HR-Subset				LR-Subset				MR-Subset			
		mAP	AP^{50}	AP^{75}	R^{50}	mAP	AP^{50}	AP^{75}	R^{50}	mAP	AP^{50}	AP^{75}	R^{50}
Patches: 512 × 512													
Faster-RCNN [33, 29]	41.5 M	49.3	76.7	54.8	-	56.0	86.5	63.6	-	-	-	-	-
Cascade-RCNN [6, 29]	69.2 M	51.6	79.2	57.0	-	58.4	88.6	68.3	-	-	-	-	-
1024 × 1024													
Faster-RCNN [33]	41.5 M	45.7	65.6	53.9	65.9	62.2	89.7	74.3	90.1	50.0	71.7	59.4	72.0
RetinaNet [24]	37.7 M	42.5	68.1	46.9	74.5	59.2	90.8	67.9	93.7	50.6	77.1	58.2	81.9
TOOD [15]	32.0 M	57.3	82.4	67.6	82.5	66.9	95.4	79.1	96.0	59.8	85.8	70.5	86.3
Def. DETR [46]	41.3 M	49.8	81.3	55.0	82.8	64.4	95.2	76.4	96.5	57.3	86.3	66.8	87.2
VariFocalNet [44]	32.7 M	56.4	81.4	66.0	82.1	66.5	94.8	80.2	95.6	59.5	85.1	73.6	85.8
Faster-RCNN [33] + Ours	42.7 M	49.1	72.4	57.7	73.0	62.6	91.3	75.0	91.7	52.5	77.2	61.9	77.6
TOOD [15] + Ours	32.8 M	67.5	95.8	80.6	95.8	68.9	97.6	82.6	98.4	68.4	96.6	81.4	96.6
1536 × 1536													
Faster-RCNN [33]	41.5 M	56.0	80.2	66.0	80.8	64.7	93.3	77.0	93.8	57.9	82.9	68.4	83.3
RetinaNet [24]	37.7 M	50.3	77.7	57.1	80.2	59.5	90.8	68.8	92.7	54.6	82.6	62.9	84.2
TOOD [15]	32.0 M	57.7	82.6	68.2	83.0	67.2	95.9	80.0	96.5	61.8	86.8	73.9	87.3
Def. DETR [46]	41.3 M	51.9	82.2	58.4	83.5	65.3	94.9	76.8	96.6	56.8	86.5	66.1	87.3
VariFocalNet [44]	32.7 M	59.7	83.1	71.0	83.6	67.6	95.8	80.2	96.6	61.8	86.4	73.6	87.0
Faster-RCNN [33] + Ours	42.7 M	61.5	89.2	72.4	89.8	65.8	95.1	79.0	95.1	62.6	91.2	74.3	91.7
TOOD [15] + Ours	32.8 M	68.2	96.3	81.1	96.2	69.5	98.0	83.4	98.7	68.0	96.2	81.6	97.0
2048 × 2048													
Faster-RCNN [33]	41.5 M	58.0	82.2	68.5	82.5	66.6	95.4	79.4	95.9	60.2	85.6	71.6	85.9
RetinaNet [24]	37.7 M	56.0	81.9	65.2	83.1	64.2	93.8	75.6	95.0	56.4	84.1	65.3	85.4
TOOD [15]	32.0 M	60.6	84.3	72.5	84.6	67.4	95.9	80.5	96.5	62.7	87.4	75.2	87.8
Def. DETR [46]	41.3 M	53.9	83.0	53.9	83.9	64.9	95.5	76.4	96.8	57.8	86.8	67.5	87.4
VariFocalNet [44]	32.7 M	60.5	83.6	72.2	84.1	68.2	95.9	81.0	96.7	63.0	87.0	74.9	87.5
Faster-RCNN [33] + Ours	42.7 M	64.0	93.0	75.3	93.5	67.5	96.9	81.2	97.3	64.4	93.7	76.4	94.1
TOOD [15] + Ours	32.8 M	68.9	96.8	82.1	97.5	70.5	98.1	84.7	98.9	68.4	96.1	82.0	97.4

achieved best results for TOOD [15] at a final resolution of 2048×2048, but it also shows excellent results even at moderate resolutions and therefore does not necessarily require very high resolutions with high computational overhead.

4.3. Further Experiments

Extending the evaluations in Section 4.1 and 4.2, we perform several more experiments on the AGAR dataset [29]. We investigating ability of generalization only using a small number of training data and examined various backbones. Furthermore, we evaluated the performace of our network on COCO [25] for general object detection and on LIVECell [13] for detection of cells on low-resolution images.

4.3.1 Limited Data Analysis

In our first additional experiment, we investigate how a reduction of the amount of data affects the training of our networks. For this reason, we created three evenly distributed subsets from the higher-resolution (HR) set, each containing 10 % (524 images), 5 % (262 images), and 1 % (53 images) of the training data. For evaluation, we use the complete validation set of the HR subset as described in Section 4. In contrast to the training in Section 4.2, we increase the number of epochs to 100 and reduce the learning rate after 50 and 80 epochs by a factor of 10.

The results listed in Table 4 show a drop between 3 % to 5 % of the mAP with respect to networks, trained on all data when using 10 % of the training data. The drop from

Table 4: **Comparison of detection accuracy** of our method to the SoTA on the AGAR [29] validation. For training, different data-splits with 10 %, 5% and 1% of the original 5000 training images are used. For this experiment, we use TOOD [15] and Faster-RCNN [33] as the network heads and evaluate at a resolution of 1536×1536.

Method	Params	Metrics			
		mAP	AP^{50}	AP^{75}	R^{50}
Subset 10 %					
Faster-RCNN [33]	41.5 M	53.4	78.8	62.3	79.4
TOOD [15]	32.0 M	54.0	80.1	63.2	81.9
Faster-RCNN [33] + ours	42.7 M	57.0	87.2	65.7	88.2
TOOD [15] + ours	32.8 M	62.9	92.3	73.8	93.8
Subset 5 %					
Faster-RCNN [33]	41.5 M	51.7	77.7	59.8	78.6
TOOD [15]	32.0 M	51.3	77.6	59.4	79.5
Faster-RCNN [33] + ours	42.7 M	55.4	86.4	63.6	87.9
TOOD [15] + ours	32.8 M	61.8	92.0	72.5	94.1
Subset 1 %					
Faster-RCNN [33]	41.5 M	41.2	70.6	43.6	73.7
TOOD [15]	32.0 M	36.8	61.2	40.5	68.3
Faster-RCNN [33] + ours	42.7 M	42.7	75.4	43.7	84.9
TOOD [15] + ours	32.8 M	42.6	72.5	46.1	80.9

TOOD [15] extended by our AttnPAFPN shows a larger loss in mAP due to the added complexity of the data-hungry transformer layers, but it still shows better accuracy than the pure TOOD trained on all data. When using 5 % of the training data, a similar picture emerges. When training with only 1 % of the image data, a very strong drop in accuracy (approximately 20 % to 25 %) of all networks can be seen. However, our AttnPAFPN still shows an above-average performance here.

Table 5: **Comparison of detection accuracy** of our method on the AGAR [29] val set. Different backbones [12, 20, 28, 42, 45] are compared. For this experiment, we use TOOD [15] as head and evaluate at a resolution of 1536×1536.

Method	Params	Metrics			
		mAP	AP^{50}	AP^{75}	R^{50}
ResNet50 [20]	**32.8 M**	68.2	96.3	81.1	96.2
ResNet101-dcnv2 [45]	54.3 M	69.2	96.9	82.9	97.5
Swin Tiny [28]	36.4 M	69.9	96.8	84.1	97.3
PVTv2-b2 [42]	33.7 M	70.2	96.8	83.9	97.4
PLG-ViT Tiny [12]	34.8 M	**70.4**	**97.0**	**84.2**	**97.6**

4.3.2 Backbone Analysis

During all previous experiments we have used a pretrained ResNet50 [20] as the network backbone, since it is still considered as one of the most important baselines in computer vision. Further improvements in accuaracy can be achieved by using modern CNNs or transformer backbones. For this reason we want to compare ResNet50 with a stronger deformable convolution backbone (ResNet101-dcnv2) [45] and three transformer-based backbones. For the transformer backbones we use Swin-T [28], PVTv2-b2 [42], and the high-resolution optimized PLG-ViT-T [12, 11]. All transformer backbones are similar in size to ResNet50 and training takes place exclusively on the higher-resolution subset of AGAR [29] at a resolution of 1536×1536. We trained ResNet101-dcnv2 with the same hyperparameters as ResNet50. For the transformer backbones, we adapted the training recipes proposed by the authors from COCO [25] to AGAR.

The results in Table 5 confirm the trend of recent years, with transformers outperforming their CNN counterparts. Even the larger ResNet101-dcnv2 backbone cannot keep up with the transformers. These manage to outperform ResNet50 and ResNet101-dcnv2 by about $+2$ and $+1$ mAP, respectively. It is also shown that the differences between transformer networks in terms of accuracy are small. However, this experiment shows the major drawback of the standard SA used by PVTv2. Even if the number of parameters is the same, the computational cost is significantly higher compared to Swin and PLG-ViT. PVTv2 requires about 200 % more GPU memory than the other two networks during training. The computational effort is also significantly higher during the inference [12]. For this reason, PLG-ViT will be used as the backbone of choice in the final experiment to achieve the best possible trade-off between accuracy and performance.

4.3.3 Beyond Colony Detection

In addition to detecting bacteria colonies in high-resolution images, we also want to evaluate our method on medium-resolution images of other areas of application. For this purpose we use the COCO dataset [25], which is a widespread baseline for object detection.For training the networks on COCO we use the standard settings [8] proposed by the authors and train for 12 epochs. As network heads we use TOOD [15] and Mask-RCNN [19] as an additional method for instance segmentation.

Table 6: **Comparison of detection and segmentation accuracy** on the COCO [25] validation set. Different methods [15, 19] are compared with our approach. We use TOOD [15] and Mask-RCNN [19] as the head and ResNet50 [20] and PLG-ViT [12] as the backbone.

Method	Backbone	Params	Metrics	
			mAP^{bb}	mAP^{seg}
TOOD [15]	ResNet50 [20]	32.0 M	42.4	-
TOOD [15] + ours	ResNet50 [20]	32.8 M	42.6	-
TOOD [15] + ours	PLG-ViT [12]	34.8 M	48.0	-
Mask-RCNN [19]	ResNet50 [20]	43.7 M	38.2	34.7
Mask-RCNN [19] + ours	ResNet50 [20]	45.9 M	39.6	35.9
Mask-RCNN [19] + ours	PLG-ViT [12]	48.4 M	45.4	41.4

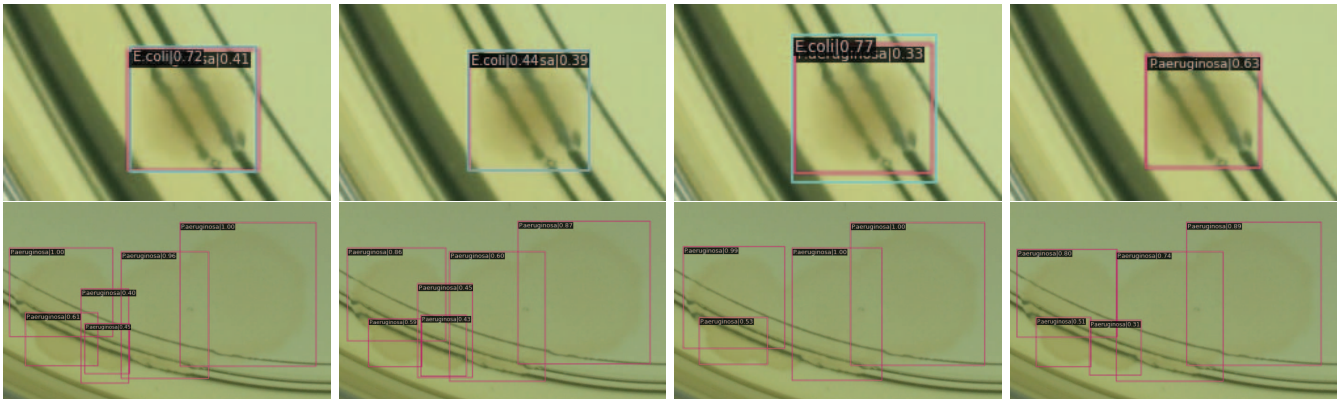
Table 7: **Comparison of detection and segmentation accuracy** on the LIVECell [13] test set. Different methods [15, 19] are compared to our approach. We use TOOD [15] and Mask-RCNN [19] as heads and ResNet50 [20] as the backbone for all models.

Method	Params	Metrics	
		mAP^{bb}	mAP^{seg}
TOOD [15]	32.0 M	29.4	-
TOOD [15] + ours	32.8 M	33.8	-
Mask-RCNN [19]	43.7 M	36.8	37.3
Mask-RCNN [19] + ours	45.9 M	38.0	38.0

The results of the evaluation on COCO can be seen in Table 6. In contrast to AGAR, only a slightly improvement of the accuracy stemming from AttnPAFPN can be seen. As already noted in Section 2, this can be explained by the different characteristics of the COCO dataset, such as the relatively small number of tiny objects, in contrast to the AGAR dataset. Using Mask-RCNN, on the other hand, the impact of our neck is more significant. We achieve $+1.4/+1.2$ mAP for detection and segmentation, respectively. The extension of the methods by a stronger transformer backbone increases the accuracy considerably.

We also performed experiments on the LIVECell dataset [13], which is used to detect and segment cells in microscopy images. For this we also use TOOD and Mask-RCNN, which were previously pre-trained on COCO. Additionally, we made some adjustments regarding the anchor-boxes of Mask-RCNN and TOOD as suggested by the authors of the dataset [13]. As a result, the networks are better adapted to the characteristics of the dataset.

The results on the LIVECell data set are listed in Table 7. Here we can see that especially Mask-RCNN performs much better on the dataset than TOOD, which is a pure detection network. But especially TOOD benefits strongly from the extension by AttnPAFPN, which outperforms the baseline by $+3.4$ mAP. Mask-RCNN achieves with AttnPAFPN an increase in accuracy of $+1.2/+0.7$



(a) Faster-RCNN [33] (b) TOOD [15] (c) Faster-RCNN [33] + Ours (d) TOOD [15] + Ours

Figure 4: **Qualitative comparison** of Faster-RCNN [33] (a), TOOD [15] (b), Faster-RCNN + AttnPAFPN (c) and TOOD + AttnPAFPN (d). All networks are trained on the AGAR dataset [29] under equal conditions. A small colony is visible in the first row with low contrast and distracting texture in the background. The second row shows a cluster of colonies with low contrast.

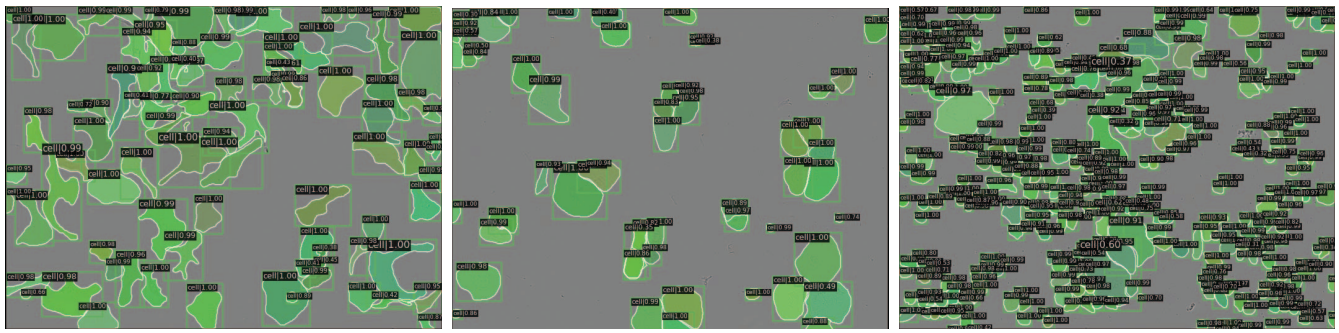


Figure 5: **Qualitative result** of Mask-RCNN [19] with our AttnPAFPN and ResNet50 [20] on LIVECell [13].

mAP for detection and segmentation, respectively. Figure 5 shows some visual results of our method with Mask-RCNN as head and ResNet50 as backbone.

4.4. Visual Evaluation

In addition to a quantitative evaluation we also present a qualitative evaluation on a greatly enlarged section of the image in Figure 4. Here it can be seen that the conventional method has difficulties with particularly small and overlapping colonies, in contrast to our method. In addition to the visual results on AGAR [29], typical results on the LIVECell dataset [13] can be seen in Figure 5.

5. Conclusion

In this paper, we presented AttnPAFPN, a high-performance feature pyramid for high-resolution object detection. Our AttnPAFPN uses our state-of-the-art efficient-global self-attention layers for better visual understanding. Moreover, the efficient-global self-attention can be easily interchanged with any other self-attention mechanism. Furthermore we add a additional scales to our PAFPN for predicting tiny and large objects on high- and low-resolution featuremaps, respectively. In order to be executable even on resource-constrained hardware, we have considered efficiency and parameter count during the optimization of our method. We have performed a comprehensive evaluation on a large scale public dataset [44] for detecting bacterial colonies on agar dishes and proved the surpassing accuracy of our method compared to the current state-of-the-art. In addition, we have performed experiments on the standard object detection baseline COCO [25], as well as on LIVECell [13] for biomedical image analysis.

Acknowledgments

This work was supported by funding from the the Federal Ministry of Education and Research Germany in the project M^{2}Aind-DeepLearning (13FH8I08IA). Additional funding was provided by the German Research Foundation under grant number INST874/9-1 and by the Albert and Anneliese Konanz Foundation.

References

[1] NE Alexander and DP Glick. Automatic counting of bacterial cultures-a new machine. *IRE Transactions on Medical Electronics*, 1958. 2

[2] Paolo Andreini, Simone Bonechi, Monica Bianchini, Alessandro Mecocci, and Franco Scarselli. A deep learning approach to bacterial colony segmentation. In *International Conference on Artificial Neural Networks (ICANN)*, 2018. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[4] Thomas Beznik, Paul Smyth, Gaël de Lannoy, and John A Lee. Deep learning to detect bacterial colonies for the production of vaccines. *Neurocomputing*, 2022. 2

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 6

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5, 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[10] Nikolas Ebert, Patrick Mangat, and Oliver Wasenmuller. Multitask network for joint object detection, semantic segmentation and human pose estimation in vehicle occupancy monitoring. In *Intelligent Vehicles Symposium (IV)*, 2022. 2

[11] Nikolas Ebert, Laurenz Reichardt, Didier Stricker, and Oliver Wasenmüller. Light-weight vision transformer with parallel local and global self-attention. *arXiv preprint arXiv:2307.09120*, 2023. 7

[12] Nikolas Ebert, Didier Stricker, and Oliver Wasenmüller. Plg-vit: Vision transformer with parallel local and global self-attention. *Sensors*, 2023. 7

[13] Christoffer Edlund, Timothy R Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. Livecell—a large-scale dataset for label-free live cell segmentation. *Nature methods*, 2021. 2, 6, 7, 8

[14] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 2019. 2

[15] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 5, 6, 7, 8

[16] Alessandro Ferrari, Stefano Lombardi, and Alberto Signoroni. Bacterial colony counting with convolutional neural networks in digital microbiology imaging. *Pattern Recognition*, 2017. 2

[17] Quentin Geissmann. Opencfu, a new free and open-source software to count cell colonies and other circular objects. *PloS one*, 2013. 1, 2

[18] Oleg Gorokhov, Ramazan Fazylov, Maria Kazachuk, Ivan Lazukhin, Igor Mashechkin, Liudmila Pankratyeva, and Ivan Popov. Bacterial colony detection method for microbiological photographic images. In *International Joint Conference on Neural Networks (IJCNN)*, 2022. 2

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 7, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 5, 7, 8

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[22] Michael R Lamprecht, David M Sabatini, and Anne E Carpenter. Cellprofiler™: free, versatile software for automated biological image analysis. *Biotechniques*, 2007. 1

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 2, 6, 7, 8

[26] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[27] Shi-Jian Liu, Pin-Chao Huang, Xing-Sheng Liu, Jin-Jia Lin, and Zheng Zou. A two-stage deep counting for bacterial colonies from multi-sources. *Applied Soft Computing*, 2022. 2

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 4, 7

[29] Sylwia Majchrowska, Jarosław Pawłowski, Grzegorz Guła, Tomasz Bonus, Agata Hanas, Adam Loch, Agnieszka Pawlak, Justyna Roszkowiak, Tomasz Golan, and Zuzanna

Drulis-Kawa. Agar a microbial colony dataset for deep learning detection. *arXiv preprint arXiv:2108.01234*, 2021. 1, 2, 4, 5, 6, 7, 8

[30] HP Mansberg. Automatic particle and bacterial colony counter. *Science*, 1957. 2

[31] Tanguy Naets, Maarten Huijsmans, Paul Smyth, Laurent Sorber, and Gaël de Lannoy. A mask r-cnn approach to counting bacterial colony forming units in pharmaceutical development. *arXiv preprint arXiv:2103.05337*, 2021. 2

[32] Nisha Ramesh and Tolga Tasdizen. Cell segmentation using a similarity interface with a multi-task convolutional neural network. *Journal of Biomedical and Health Informatics (JBHI)*, 2018. 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 2, 5, 6, 8

[34] Rishav, René Schuster, Ramy Battrawy, Oliver Wasenmüller, and Didier Stricker. Resfpn: Residual skip connections in multi-resolution feature pyramid networks for accurate dense pixel matching. In *International Conference on Pattern Recognition (ICPR)*, 2021. 2

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*. Springer, 2015. 2

[36] Michael Shamash and Corinne F Maurice. Onepetri: accelerating common bacteriophage petri dish assays with computer vision. *Phage*, 2(4):224–231, 2021. 2

[37] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[39] Angelo Torelli, Ivo Wolf, Norbert Gretz, et al. Autocellseg: robust automatic colony forming unit (cfu)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques. *Scientific reports*, 2018. 1, 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3, 4

[41] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2020. 3

[42] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 2022. 7

[43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4

[44] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5, 6, 8

[45] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Conference on Computer Vision and Pattern Recognition*, 2019. 7

[46] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 5, 6