Supplementary Materials

Model overview and selection rationale (image classification)

FastViT models leverage the power of the Transformer architecture, which has proven highly effective in capturing long-range dependencies within data. These models are designed with varying depths and parameter counts, allowing us to explore different trade-offs between performance and computational requirements. FastViTs are optimized for speed while maintaining strong predictive capabilities, making them suitable for real-time analysis of TMJ pathologies.

EfficientViT models further refine the Transformer architecture by focusing on efficiency without compromising on accuracy. They incorporate advanced techniques such as attention mechanisms and efficient block designs, enabling the models to learn intricate patterns in the TMJ ROIs with reduced computational overhead. EfficientViTs are tailored for scenarios where high performance must be achieved under strict resource constraints.

As a lightweight CNN, MobileNetV3-small is engineered for optimal efficiency, featuring innovations such as inverted residuals and linear bottlenecks. It is particularly advantageous for its minimal computational footprint, which allows it to run efficiently on devices with limited processing power, while still delivering robust performance in detecting TMJ pathologies. EfficientNet-B1 represents a scalable CNN family that achieves a favorable balance between model size and performance. By utilizing compound scaling, it systematically scales the network width, depth, and resolution, resulting in a model that offers excellent accuracy while being computationally efficient. EfficientNet-B1 is an ideal choice for tasks requiring a balance between precision and resource utilization.

Implementation details

In this study, we employed five variants of the YOLO object detection model, YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x, to perform ROI detection in medical images. Each variant of the YOLO model was characterized by different architectural complexities and computational requirements, which allowed us to explore their performance in terms of both accuracy and efficiency in a clinical setting.

During the training process of the YOLOv11 model for TMJ detection, we adhered to the official configuration settings. We trained the model for 10 epochs using an input image size of 3 × 640 × 640 pixels. The Stochastic Gradient Descent (SGD) optimizer was employed with an initial learning rate of 0.01 and a momentum of 0.937.

We selected a combination of Transformer-based architectures and lightweight CNN models to investigate their effectiveness in the binary classification of TMJ disc displacement from CBCT images. Specifically, we explored four variants of the FastViT Transformer architecture—FastViT-t8, FastViT-t12, FastViT-s12, and FastViT-sa36—which were chosen for their differing depths and parameter counts to meet varying performance and efficiency needs. Additionally, we included three versions of the EfficientViT Transformer series: EfficientViT-m0, EfficientViT-m2, and EfficientViT-m5, each chosen to balance model complexity with computational efficiency. To complement the Transformer models, we also incorporated two resource-efficient lightweight CNN architectures: MobileNetV3-small and EfficientNet-B1. MobileNetV3-small is optimized for minimal computational cost while maintaining competitive accuracy, making it suitable for resource-constrained

scenarios. EfficientNet-B1, in contrast, offers a scalable architecture that balances model size and performance, serving as an efficient yet powerful option for image classification tasks.

By evaluating this diverse model set, our study aimed to identify the most suitable architecture for accurate TMJ disc displacement prediction in clinical applications. This selection enabled us to compare the impact of model size and architecture on classification accuracy, while also accounting for computational efficiency in view of practical clinical deployment. To ensure a fair comparison, all images were resized to 3 × 224 × 224 prior to training each network, after which they were normalized and standardized.

During the training process, we employed the AdamW optimizer with an initial learning rate of 0.001 and used Cross-Entropy Loss to optimize the model parameters. All networks were implemented and trained using the PyTorch framework. Each model was trained for 50 epochs with a batch size of 32.