MedSegText

A Multi-Modal Medical AI for Automated CT Interpretation

Demo | Detailed Report | GitHub

Tamilarasee Sethuraj

What is MedSegText?

MedSegText is an intelligent platform that assists radiologists by automating two of the most critical and time-consuming aspects of CT scan interpretation.

It functions through a simple, secure workflow: a user uploads a patient's CT scan, and the system processes it to deliver two synchronized outputs in a single step:

1. A **Segmentation Mask:** A visual overlay that precisely highlights and delineates potentially infected or abnormal regions within the CT scan.
2. An **Auto-Generated Finding:** A concise, clinically relevant textual description of the findings, ready for review and integration into the final radiology report.

Essentially, it's a dual-task system that learns to both *see* and *describe* abnormalities, bridging the gap between visual analysis and clinical reporting.

What is the real-world impact for radiology departments?

The primary impact of MedSegText is its potential to significantly enhance diagnostic efficiency and consistency, especially in high-volume environments. By automating the initial segmentation and drafting of findings, the platform aims to:

- **Streamline Clinical Workflows:** MedSegText handles the foundational tasks of localizing and describing findings, allowing radiologists to focus their expertise on higher-level interpretation, verification, and complex case analysis.
- **Enhance Consistency:** By using a standardized model, the generated text for common findings can be made more consistent, potentially reducing inter-observer variability in reports.
- **Support High-Volume Diagnostics:** In situations that strain resources, such as pandemics or in understaffed facilities, MedSegText can act as a powerful productivity tool, helping to manage caseloads more effectively. It provides a reliable "first-pass" analysis that accelerates the entire reporting process.

Is MedSegText dependent on open-source large language model (LLM) like GPT?

No, and this is a critical architectural distinction. MedSegText is **not** a dependent on any external, general-purpose LLM API

It is a **self-contained, integrated deep learning system**. The entire framework is designed to run locally or within a private cloud environment, completely offline from third-party generative AI services.

While we leverage powerful, publicly available pre-trained models as foundational components (like **ConvNeXt** for the vision encoder and a tokenizer from **BiomedVLP** for the text pathway), these are integrated directly into our single, unified MedSegText architecture. The model is then trained end-to-end on the specific radiological dataset.

This self-contained approach has several important implications:

- **Data Privacy and Security:** Because the entire model runs as a single package, no patient data ever needs to be sent to an external, third-party API. This is a paramount consideration for clinical applications and ensures full control over sensitive information.
- **Operational Independence:** The system does not depend on the availability, cost, or potential API changes of external LLM providers. It offers a more stable, predictable, and fully offline-capable operational footprint.
- **Domain-Specific Tuning:** The integrated Transformer decoder learns to perform its specific task—generating concise findings—by directly cross-attending to the visual features from our encoder, all within the same training loop. It is fine-tuned for this precise medical task rather than relying on a general-purpose model.

In short, MedSegText is a purpose-built medical AI system that *integrates* pre-trained components into a novel framework, not an application that *calls* an external service.

Are there similar products on the market, and what makes MedSegText unique?

Yes, the field of AI in medical imaging is active, and several tools offer either segmentation or report-generation capabilities. However, the key differentiator for MedSegText lies in its **unified, single-model architecture**.

Many existing commercial solutions likely employ a *modular* or *pipeline* approach: one AI model performs segmentation, its output is then passed to a separate, often disconnected, language model (like an LLM via an API) to generate a report. This can create integration complexities and potential inconsistencies between the visual findings and the text.

MedSegText is fundamentally different. It is built on a **single, end-to-end deep learning model** that is trained to perform both tasks simultaneously. The "thinking process" for segmenting the image and describing it happens within a unified framework, not in separate models requiring integration of results

Why is a single, unified model a significant advantage?

Having one model handle both tasks, even with parallel decoders, offers tangible product advantages over a multiple models pipeline:

- **Synergistic Performance Through a Shared "Vision":** While the segmentation and text decoders operate in parallel, they both learn from a **single, shared visual encoder**. This is the core of the synergy. The encoder is forced to learn a richer, more robust set of visual features because its representations must be useful for *both* the spatially-demanding segmentation task and the semantically-demanding text generation task. This joint optimization process refines the encoder's "vision," potentially leading to better performance on both outputs compared to training two separate models from scratch.

- **Inherently Grounded Outputs:** Because both decoders interpret the *exact same* set of extracted visual features, the outputs are inherently linked. The text generation is naturally conditioned on the same visual understanding that drives the segmentation, ensuring the findings are "grounded" in the features the model deemed important. This reduces the risk of generating text that contradicts the visual evidence.

- **Simplified Deployment and Maintenance:** Managing, updating, and scaling a single, cohesive model (one encoder, two decoders) is far less complex than maintaining two entirely separate models and the infrastructure to run them. This reduces operational overhead and ensures greater system reliability.

- **Efficiency and Tunability:**
    - **At Inference:** The most computationally expensive part—processing the image through the encoder—is done only once. The resulting features are then efficiently passed to the two lightweight decoders.

  - **During Training:** We have a unified framework where we can tune the learning process (e.g., by adjusting loss weights) to balance the priorities of both tasks. This provides a direct path for future enhancements.

- **A Foundation for Deeper Integration:** This shared-encoder architecture is a crucial stepping stone. It establishes the feasibility of joint learning and serves as the foundation for our ongoing research into implementing more **advanced cross-modal attention** mechanisms *between* the decoders. This next step will allow the decoders to directly influence each other, enabling the segmentation mask to explicitly guide text generation and vice-versa, further enhancing the model's synergistic capabilities.

How reliable are the results?

The results from our prototype are highly encouraging and establish a strong performance baseline, especially considering the efficiency of the training process. When evaluated on the MosMedData+ test set, the framework demonstrated significant capabilities.

To put the evaluated metrics into clear terms:

- **High Segmentation Accuracy:** For the primary task of highlighting lesions, the model achieved an **average Dice Score of 0.7279 (approximately 73%)**. This metric measures the overlap between our model's prediction and the expert's ground truth mask, indicating that the model is highly effective at correctly identifying the location and general shape of abnormalities.

- **Excellent Text Generation Quality:** Simultaneously, the model produced high-quality text, achieving a **BLEU-4 score of 0.8449 (84%)** and a **METEOR score of 0.9405 (94%)**. These scores are nearly identical to those from a dedicated model trained *only* for text generation (which scored 0.8036 and 0.9279 respectively), demonstrating that the joint-task framework can generate highly relevant and coherent text without compromising quality.

Key Context:

It's important to note that this strong baseline performance was achieved with a relatively modest training duration of just **50 epochs**. We have a clear architectural path for the next phase of research—implementing **cross-learning** between the decoders and extending training—which we anticipate will lead to **significant further improvements** on these already promising results.

How is MedSegText architected for real-world deployment?

Beyond the core AI model, it's crucial to consider how such a system would operate in a real-world clinical IT environment. To that end, we developed a **cloud-native prototype** that demonstrates a viable, scalable, and secure deployment strategy.

This prototype is not just a model script; it's a proof-of-concept for an end-to-end system built on a modern microservice architecture using Amazon Web Services (AWS):

- **User Interface (Proof-of-Concept):** A user-facing application with **Streamlit**, which was containerized using **Docker** and prototyped for deployment on **AWS Fargate (ECS)**. This demonstrates a pathway for a secure, isolated web application for user interaction could be automatically scaled to run multiple containers when there is more traffic.
- **Backend API (Serverless Design):** A serverless backend was designed using **AWS Lambda** and **FastAPI**. This handles data orchestration—storing patient and image data securely in **S3** and metadata in **DynamoDB**—and triggers the AI inference.
- **Scalable Model Inference:** The core PyTorch model is hosted on **AWS SageMaker**. This provides a managed, scalable endpoint for on-demand processing, ensuring the AI component can handle varying workloads efficiently.
- **Secure Data Flow:** The prototype's design incorporates security best practices, such as using **IAM roles** and **presigned URLs** to manage access to sensitive patient data stored in S3.

This serverless design demonstrates how MedSegText could be deployed in a way that is highly scalable, cost-effective (pay-per-use), and straightforward to maintain. While this is a prototype, it validates a clear and robust architectural blueprint for moving the MedSegText concept from research to a production-ready clinical tool.