DOI: 10.65322/j.srd.06202026.53

GRAPHICAL SOFTWARE FOR MEDICAL IMAGE CLASSIFICATION AND ENDOSCOPIC POLYP SEGMENTATION

Arabboyev M.M., *Associate Professor;*
Begmatov Sh.A., *Associate Professor;*
Nishanov A.X., *Professor,*
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Abstract. *A graphical system based on Tkinter integrates AGF-C deep learning classification of chest X-rays and AGF-S endoscopic polyp segmentation. The workflow comprises model loading, preprocessing, inference and visualization. The segmentation module provides threshold adjustment and blue-overlay export. A 256 × 256 example at a threshold of 0.500 returned a displayed mask area ratio of 0.1146. Pneumonia and COVID-19 predictions were displayed with rounded confidence scores of 0.9993 and 1.0000.*

Annotatsiya. *Tkinter asosidagi grafik tizim AGF-C chuqur o‘qitish modeli yordamida rentgen tasvirlarini sinflashtirish va AGF-S yordamida endoskopik poliplarni segmentatsiyalashni birlashtiradi. Jarayon modelni yuklash, dastlabki ishlov berish, bashoratlash va vizuallashtirishdan iborat. Segmentatsiya moduli qaror chegarasini o‘zgartirish va ko‘k niqobli tasvirni saqlash imkonini beradi. 256 × 256 o‘lchamli misolda 0,500 chegara uchun niqob maydonining ulushi 0,1146 bo‘lgan. Pnevmoniya va COVID-19 rentgen misollarida yaxlitlangan ishonch ballari 0,9993 va 1,0000 sifatida ko‘rsatilgan.*

Аннотация. *Графическая система на основе Tkinter объединяет классификацию рентгенограмм моделью глубокого обучения AGF-C и сегментацию эндоскопических полипов методом AGF-S. Процесс включает загрузку модели, предварительную обработку, прогнозирование и визуализацию. Модуль сегментации позволяет изменять порог и сохранять изображение с синей маской. В примере размером 256 × 256 при пороге 0,500 доля площади маски составила 0,1146. Для предсказаний пневмонии и COVID-19 отображены округлённые оценки уверенности 0,9993 и 1,0000.*

Keywords: *medical imaging, graphical interface, AGF-C, AGF-S, polyp segmentation, chest X-ray classification.*

Kalit so‘zlar: *tibbiy tasvirlar, grafik interfeys, AGF-C, AGF-S, polip segmentatsiyasi, rentgen tasnifi.*

Ключевые слова: *медицинские изображения, графический интерфейс, AGF-C, AGF-S, сегментация полипов, классификация рентгенограмм.*

Introduction. Medical image classification and segmentation produce different outputs: an image-level category and a spatial mask. Deep learning supports both tasks [1], but using a trained network outside its training environment also requires model loading, consistent preprocessing and an interpretable presentation of results. These implementation steps influence whether a specialist can review a prediction without operating a programming notebook. Clinical impact additionally depends on evaluation in the intended workflow and population [2].

This work presents the graphical implementation of the classification and segmentation functions associated with ChestVision AI and MedSegment AI [3, 4]. The system exposes Adaptive Global–Local Feature Fusion for Classification (AGF-C) and for Segmentation (AGF-S) through a common desktop interface. The contribution is the integration of these models into a defined sequence of user actions, task-specific preprocessing and output visualization. The paper reports the architecture and individual software demonstrations; it does not introduce a new training comparison or a clinical diagnostic study.

DOI: 10.65322/j.srd.06202026.53

Literature review. The U-Net architecture established an encoder–decoder design that combines contextual features with spatial information for biomedical segmentation [5]. EfficientNet studies the joint scaling of network depth, width and resolution [6], while MobileNetV2 uses inverted residual blocks and linear bottlenecks for efficient feature extraction [7]. These network families provide background for the encoder–decoder segmentation and global–local classification components exposed by the present software.

Software frameworks address a different part of the development process. MONAI supplies medical-image transformations, model components and utilities for building reproducible imaging workflows [8]. A desktop interface built around selected trained models serves a narrower purpose: it makes model selection, inference and visual inspection accessible through explicit controls. The segmentation tab examined here is labelled Kvasir-SEG, a dataset of gastrointestinal polyp images with annotated masks [9]. This label identifies the demonstrated task as polyp segmentation.

Interpreting the displayed output also requires care. A high class score does not establish that a prediction is correct or that probabilities are calibrated; Guo et al. distinguish confidence calibration from classification performance [10]. Likewise, a visible mask or its area does not measure agreement with an expert annotation. Metrics Reloaded links metric selection to the image-analysis task and its reference standard [11]. These distinctions guide the interpretation of the software examples below.

Software architecture and methods. ***Interface and processing layers.*** The application uses Tkinter, the Python interface to the Tcl/Tk graphical toolkit [12]. Its Classification (CXR) and Segmentation (Kvasir-SEG) tabs share an image-display area and task-specific controls. The user selects a module, loads a trained model, opens an image and activates prediction. The output panel records the loading state and inference result. Table 1 summarizes the division of responsibilities; Figure 1 shows the two processing branches.

Table 1

Processing layers of the graphical system.

Layer	Implemented responsibility
User interface	Select the task, model, image and prediction command.
Input	Load chest X-ray or endoscopic images and trained model files.
Preprocessing	Resize the image, normalize intensities and adapt channels.
Inference	Run AGF-C classification or AGF-S segmentation.
Visualization	Display class scores or a mask superimposed on the image.
Output	Save a segmentation overlay and allow analysis of another image.

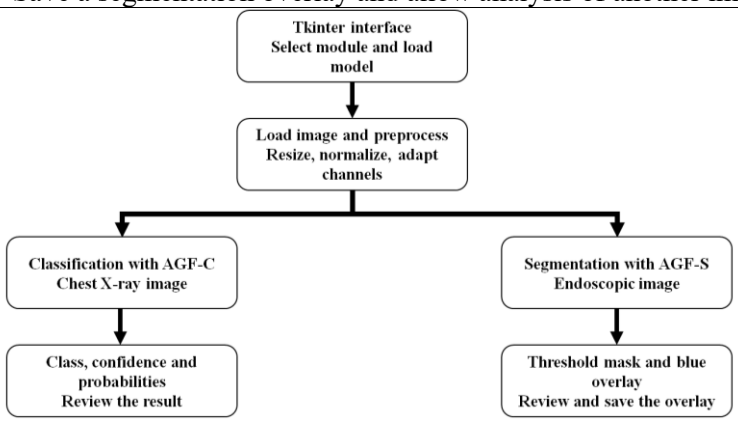


Figure 1. Workflow of the medical image analysis software

Model loading and image preparation. The classification controls distinguish a full model from a weights file, and the segmentation controls load a full model. The interface accepts .h5 and .keras model representations. A weights-only file requires the matching network architecture; a

DOI: 10.65322/j.srd.06202026.53

full model containing custom layers requires the corresponding definitions during deserialization [13]. Loading a file therefore remains tied to the architecture used during training.

Image loading controls explicitly list PNG, JPG and DICOM. The system description additionally includes BMP and TIFF. These are input-format capabilities, not evidence of compatibility with every DICOM object or network service. DICOM defines medical imaging information exchange and associated conformance requirements [14]. The examples in this paper concern single-image loading and display.

For an input image I, preprocessing is represented by a model-dependent operator P_m. It includes resizing, intensity normalization and adaptation of the channel structure. The resulting tensor must match the selected model:

$$I^* = P_m(I) \quad (1)$$

The classification interface shows an input size of 224 and identifies EfficientNetB0 as the global backbone and MobileNetV2 as the local backbone, with ROI attention enabled. The segmentation log shows an input shape of (None, 256, 256, 3) and an output shape of (None, 256, 256, 1). The leading unspecified dimension represents the batch dimension. These settings describe the demonstrated models; preprocessing must remain consistent with each model's training configuration.

Classification and segmentation outputs. AGF-C maps a prepared chest X-ray image to class probabilities p_k for COVID-19, pneumonia and normal. The predicted class ĉ is the label with the largest probability, and c is the displayed confidence:

$$p = f_{AGF-C}(I^*), \quad \hat{c} = arg\,max_k\, p_k, \quad c = max_k\, p_k \quad (2)$$

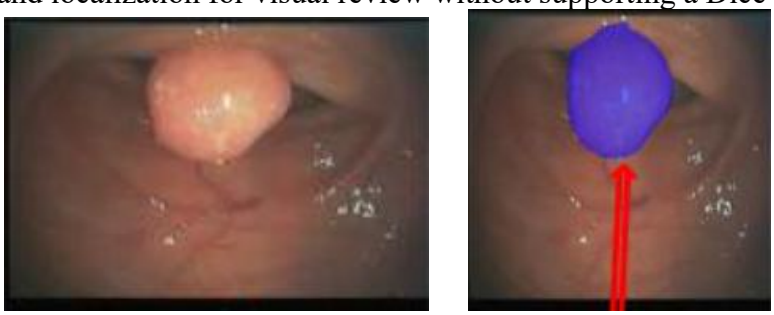
The classification output panel presents the selected class, the class-probability distribution and the adaptive gate weights. These values describe the model's output for the loaded image. The maximum probability alone does not provide a population-level error rate.

AGF-S produces a pixel-probability map Q. Threshold t converts it into a binary mask M, where the indicator is one for pixels reaching the threshold and zero otherwise:

$$Q = f_{AGF-S}(I^*), \quad M(x,y) = \mathbb{1}[Q(x,y) \geq t] \quad (3)$$

A slider controls the threshold, and the resulting foreground is displayed as a blue overlay. A separate command saves the overlay as a PNG image. The output panel also reports the mask area ratio, defined as the number of foreground pixels divided by the total number of pixels in the mask. This ratio describes predicted coverage, not segmentation accuracy. Altering the threshold changes this coverage without retraining the model.

Functional demonstration results. ***Endoscopic polyp segmentation.*** The segmentation demonstration shows the sequence from loading a full model to displaying an endoscopic image and generating a blue foreground overlay. Figure 2 reproduces the image-display regions of the original interface. The mask covers the protruding region visible in the example. The output log records a threshold of 0.500 and a mask area ratio of 0.1146, corresponding to 11.46% of the predicted mask grid. No reference mask is displayed alongside this case, so the example establishes output generation and localization for visual review without supporting a Dice or IoU estimate.

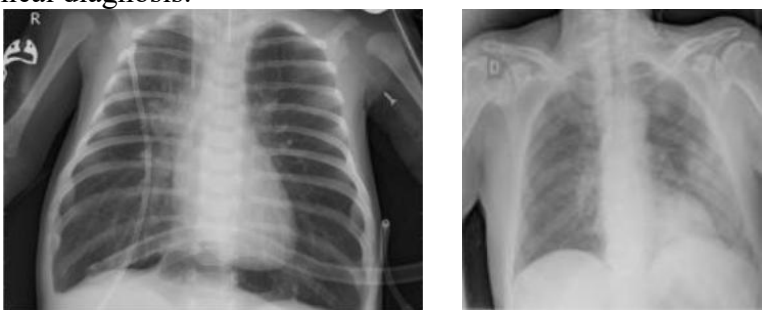


(a) Displayed input (b) Displayed overlay

DOI: 10.65322/j.srd.06202026.53

Figure 2. Endoscopic image and the blue segmentation overlay extracted from the software display. The red arrow in panel (b) is retained from the original annotation. Display proportions follow the original interface views

Chest X-ray classification. The classification demonstration first loads the model weights and then a chest X-ray image. After prediction, the output panel lists a class and its score. Figure 3 shows the two radiographs used in the illustrated outputs. The first is assigned to pneumonia with a displayed confidence of 0.9993. The second is assigned to COVID-19 with a displayed confidence of 1.0000. The latter is rounded to four decimal places and must not be interpreted as certainty of the clinical diagnosis.



(a) Pneumonia prediction (b) COVID-19 prediction

Figure 3. Chest X-ray image regions extracted from the classification demonstrations. Panel labels refer to model predictions; no independent reference diagnoses are established by these screenshots

Table 2

Values displayed in the individual software demonstrations.

Demonstration	Displayed output	Interpretation
Endoscopic image	Threshold 0.500; mask area ratio 0.1146	Foreground coverage in the predicted mask.
Chest X-ray A	Pneumonia; confidence 0.9993	Maximum class probability for this image.
Chest X-ray B	COVID-19; confidence 1.0000	Maximum class probability, rounded in the interface.

Table 2 summarizes the values visible in the interface rather than aggregating them into an accuracy statistic. The normal category is part of the classifier's output space, but the illustrated results do not include a normal-case prediction. The demonstrations show model loading, image display and production of task-specific outputs. The segmentation interface also provides an export command; no timing or file-integrity benchmark is reported here.

Discussion. The shared interface presents classification and segmentation results in forms suited to each task. For chest X-rays, it displays the predicted class and its score; for endoscopic images, it shows how the selected threshold produces a mask overlay that can be reviewed and saved. The mask area ratio provides a concise measure of the region highlighted in the displayed image. Together, these functions make the model outputs accessible within a single workflow and allow users to inspect each result in its image context.

Conclusion. A Tkinter-based system integrates AGF-C chest X-ray classification and AGF-S endoscopic polyp segmentation into a shared model-loading, preprocessing and review workflow. The demonstrated implementation displays class scores and produces a thresholded blue segmentation overlay with an export control. The available examples establish these software functions and their outputs. Diagnostic accuracy, segmentation agreement and clinical utility remain separate evaluation questions requiring reference data and a defined study protocol.

DOI: 10.65322/j.srd.06202026.53

References

1. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

2. Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., & King, D. (2019). Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine, 17*, Article 195. https://doi.org/10.1186/s12916-019-1426-2

3. Arabboyev, M. M., Begmatov, S. A., & Nishanov, A. X. (2026). *ChestVision AI* (Certificate of official registration of a computer program No. DGU 62797). Republic of Uzbekistan.

4. Arabboyev, M. M., Begmatov, S. A., & Nishanov, A. X. (2026). *MedSegment AI* (Certificate of official registration of a computer program No. DGU 62720). Republic of Uzbekistan.

5. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Lecture Notes in Computer Science, Vol. 9351, pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

6. Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (PMLR, Vol. 97, pp. 6105–6114). https://proceedings.mlr.press/v97/tan19a.html

7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474

8. Cardoso, M. J., Li, W., Brown, R., et al. (2022). *MONAI: An open-source framework for deep learning in healthcare* (arXiv:2211.02701). arXiv. https://doi.org/10.48550/arXiv.2211.02701

9. Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., de Lange, T., Johansen, D., & Johansen, H. D. (2020). Kvasir-SEG: A segmented polyp dataset. In *MultiMedia Modeling* (Lecture Notes in Computer Science, Vol. 11962, pp. 451–462). Springer. https://doi.org/10.1007/978-3-030-37734-2_37

10. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (PMLR, Vol. 70, pp. 1321–1330). https://proceedings.mlr.press/v70/guo17a.html

11. Maier-Hein, L., Reinke, A., Godau, P., et al. (2024). Metrics reloaded: Recommendations for image analysis validation. *Nature Methods, 21*, 195–212. https://doi.org/10.1038/s41592-023-02151-z

12. Python Software Foundation. (n.d.). *tkinter – Python interface to Tcl/Tk*. Python documentation. https://docs.python.org/3/library/tkinter.html

13. Keras. (n.d.). *Save, serialize, and export models*. Keras developer guides. Retrieved November 15, 2026, from https://keras.io/guides/serialization_and_saving/

14. National Electrical Manufacturers Association. (2026). *DICOM PS3.1 2026d: Introduction and overview*. NEMA.