When Pixels Talk Back:
How Watermarks Disrupt Medical Image Analysis AI

Sean Moulton, Omkar Kulkarni, Anupreet Singh
May 8, 2025

University of Maryland, Baltimore County

Introduction and Motivation

How do watermarks disrupt medical image analysis AI?

- **Use an empirical approach:**
  - Three AI models.
  - Covid-19 CT scan dataset.
  - Three watermarking algorithms.
- Build on (Apostolidis and Papakostas 2022).
  - *Some key shortcomings…*
  - Only tests krawtchouk moments (one watermarking algorithm).
  - **Few details on why or how analysis fails and how it relates to watermarking.**
  - Nit: Model selection lacks variation.

Hypothesis: **As structural similarity (SSIM) between the original and watermarked images decreases the classification accuracy also decreases.**

The details are where this gets interesting.

Overview

- Introduction and Motivation
- Background
- Krawtchouk Watermarking
- Guo and Zhuang Algorithm
- RONI Algorithm
- Methods and Results
- Conclusion

Code available on GitHub: https://github.com/WindowsVista42/CMSC-652-Group-Project

Background

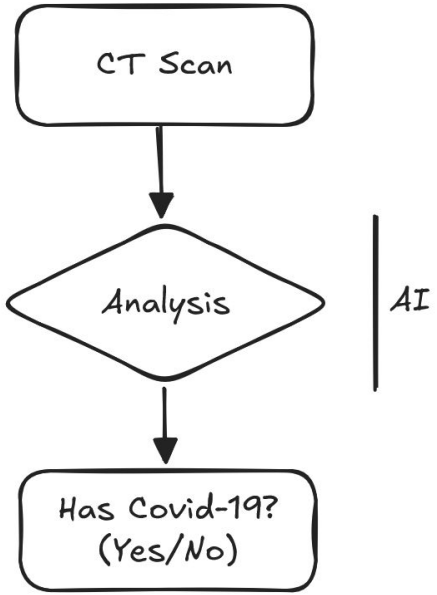
Reference Paper

K. D. Apostolidis and G. A. Papakostas, “**Digital Watermarking as an Adversarial Attack on Medical Image Analysis with Deep Learning**,” *J Imaging*, vol. 8, no. 6, p. 155, May 2022, doi: 10.3390/jimaging8060155.

Medical Image Analysis

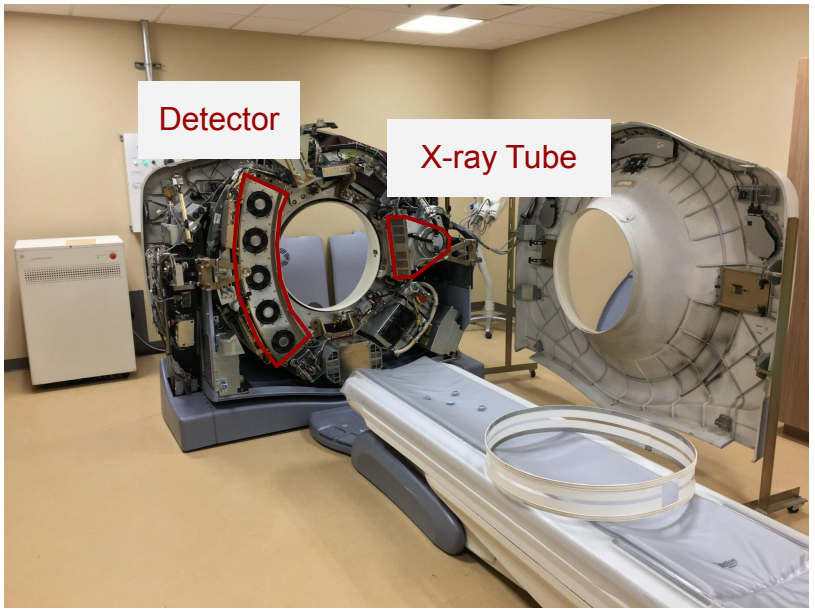
Use computer vision to automate processing, analysis, and interpretation of medical images.

(Li et al. 2023)



CT Scans

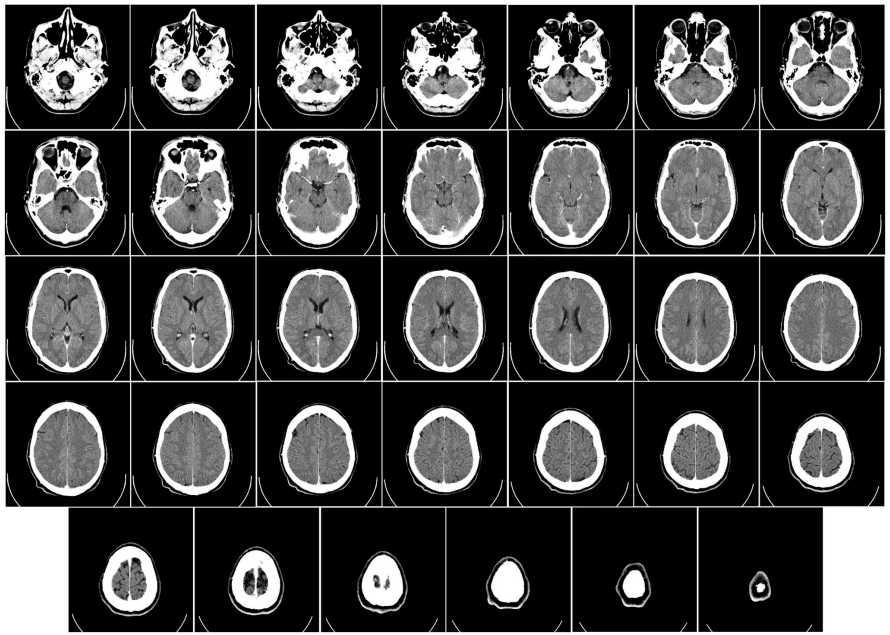
CT scan machine



CT scan machine (opened)

https://www.topdoctors.co.uk/medical-dictionary/ct-scan-cat
https://www.medicalimagingsource.com/computed-tomography

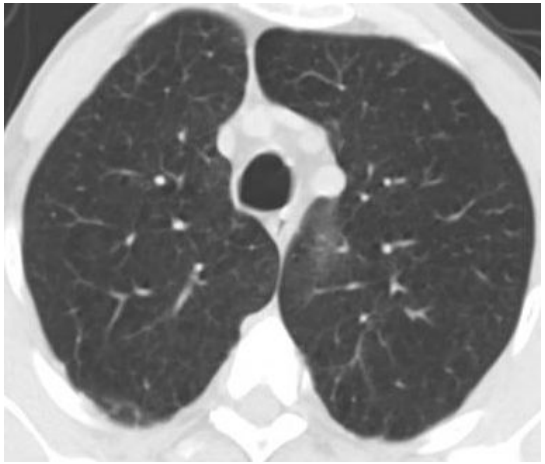
CT Scans



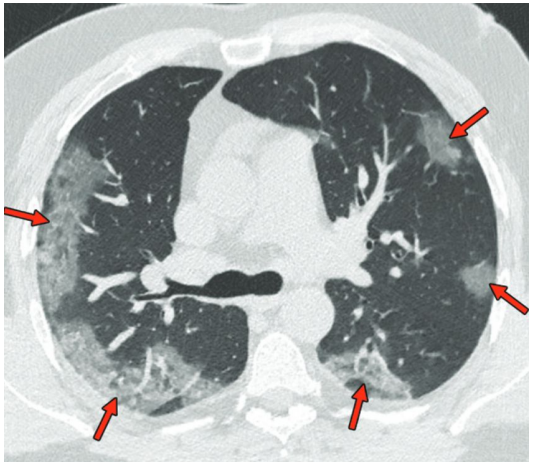
CT scans of the head.

https://en.wikipedia.org/wiki/Computed_tomography_of_the_head

CT Scans



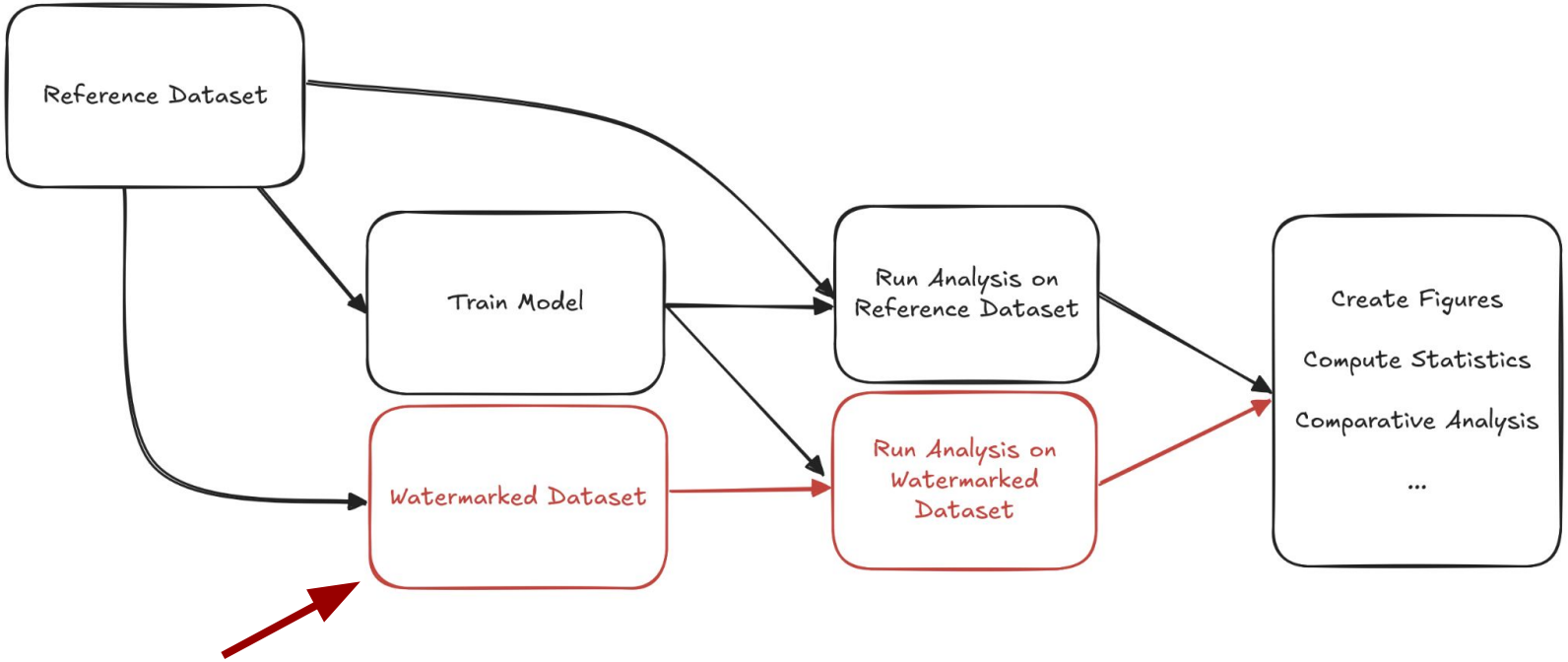
CT scan of lungs without Covid-19.



With Covid-19.

https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset
(Kwee and Kwee 2020)

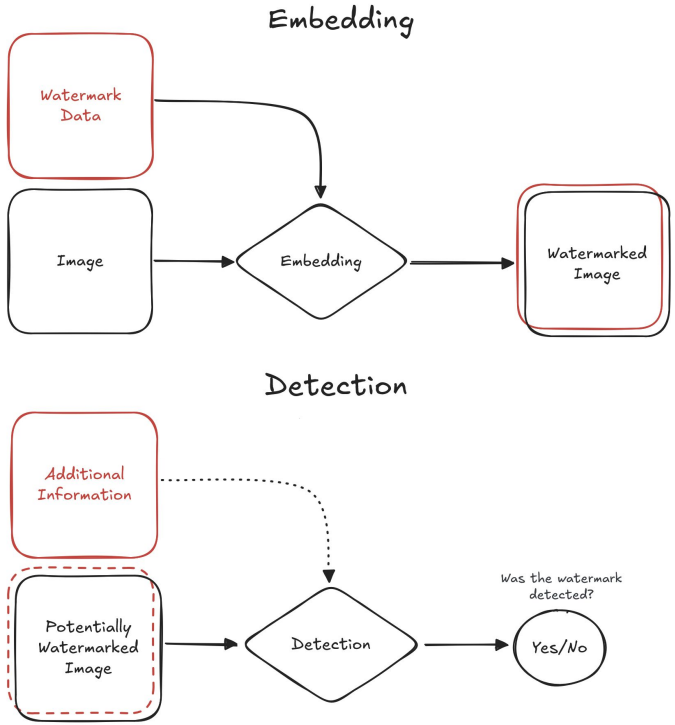
Method


Reference Dataset
Train Model
Watermarked Dataset
Run Analysis on Reference Dataset
Run Analysis on Watermarked Dataset
Create Figures
Compute Statistics
Comparative Analysis
...

Digital Watermarking

Watermarking use cases:

- Data embedding
- Copyright protection
- Traceability
  - C2PA



Watermarking Algorithms Chosen

1. Papakostas, G.A., E.D. Tsougenis, and D.E. Koulouriotis. 2014. "**Moment-Based Local Image Watermarking via Genetic Optimization.**" Applied Mathematics and Computation 227 (January):222–36. https://doi.org/10.1016/j.amc.2013.11.036. ★

2. Guo, Xiaotao, and Tian-ge Zhuang. 2009. "**A Region-Based Lossless Watermarking Scheme for Enhancing Security of Medical Data.**" Journal of Digital Imaging: The Official Journal of the Society for Computer Applications in Radiology 22 (1): 53–64. https://doi.org/10.1007/s10278-007-9043-6.

3. Memon, Nisar A., S.A.M. Gilani, and Asad Ali. 2009. "**Watermarking of Chest CT Scan Medical Images for Content Authentication.**" In 2009 International Conference on Information and Communication Technologies, 175–80. https://doi.org/10.1109/ICICT.2009.5268167.

Krawtchouk Moments Based Watermarking

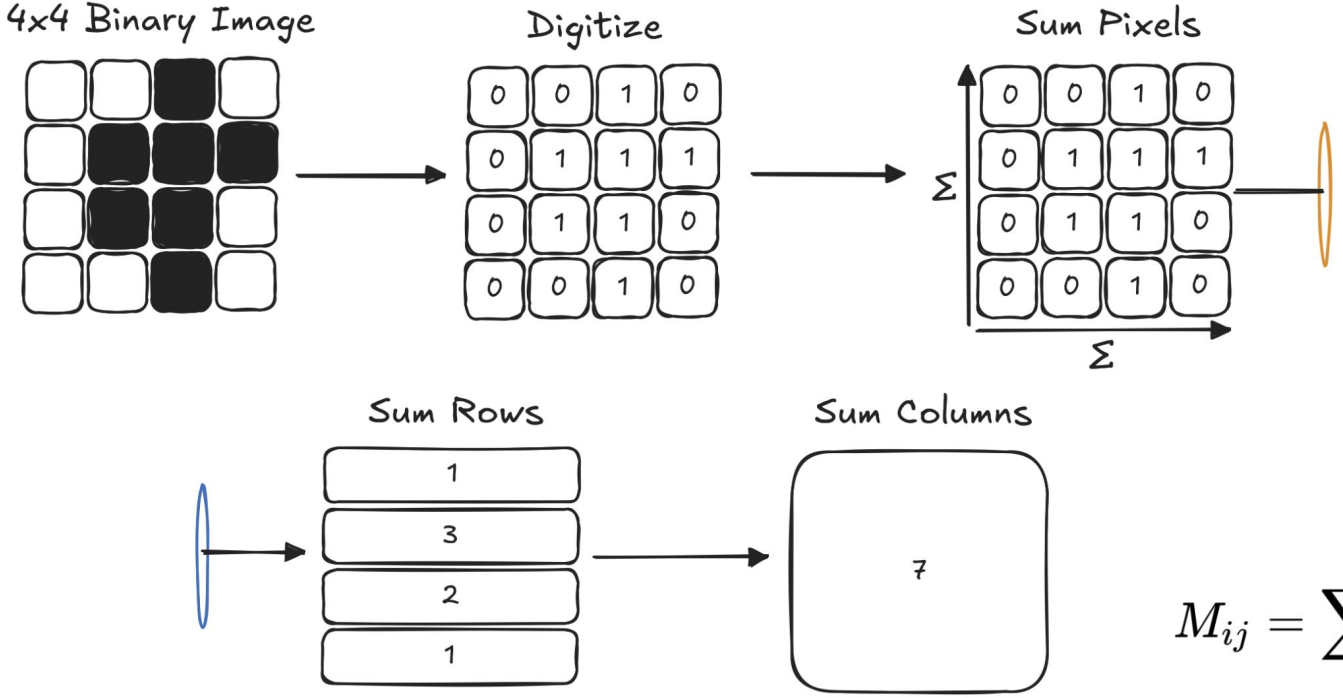
Krawtchouk Polynomials

Mykhailo Pilipovich Krawtchouk

- Soviet Ukrainian mathematician
- Introduced Krawtchouk polynomials in 1929.

(“Mikhail Krawtchouk - Biography,” n.d.)

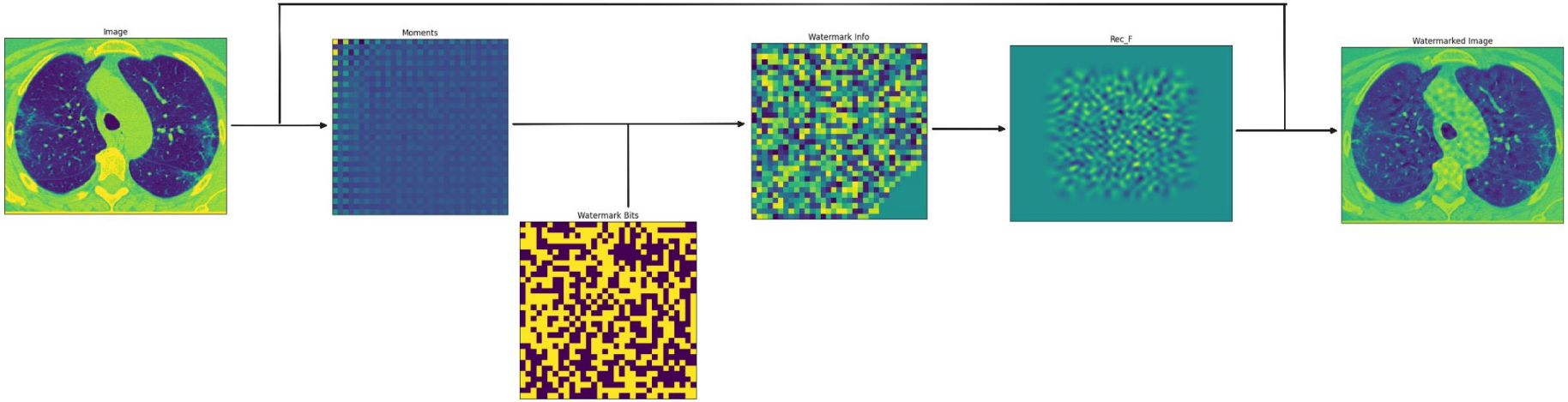
Image Moments

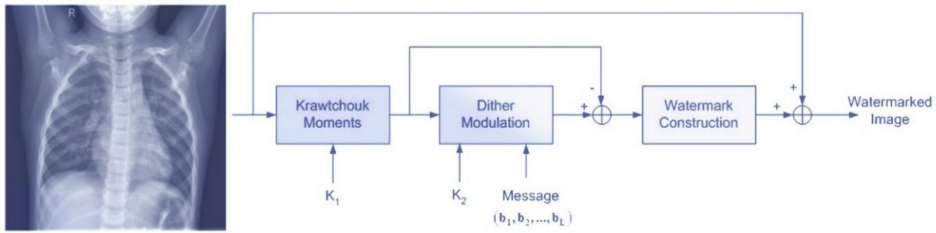


$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

Diagram of M_{00} Moments

Krawtchouk Watermark Embedding Algorithm





(Apostolidis and Papakostas 2022)

Watermark Embedding

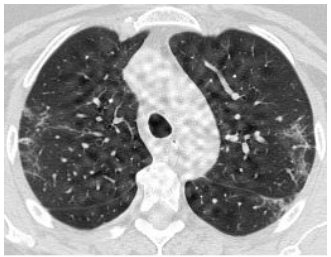
Watermarked images with various embedding strengths.

Embedding strengths: (a) = 0, (b) = 50, (c) = 100, (d) = 200, (e) = 300.

(a)

(b)

(c)



(d)

(e)

Watermark Embedding

Embed strength = 0

Embed strength = 300

Can you still see the Covid-19 features of the image?

Yes!

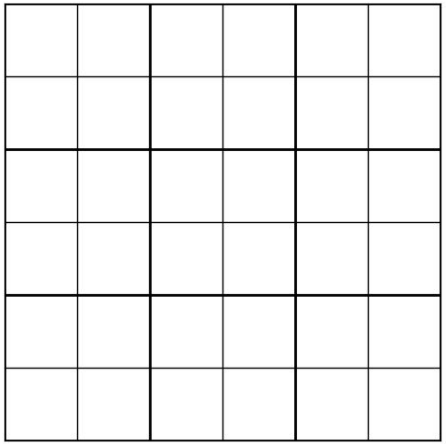
Guo and Zhuang Algorithm

Brief description

- Lossless watermarking
- Minimal noise introduced in the ROI
- Embedding of the digital signature and the hash of the image
- Can embed identifiers from an EPR, or even fingerprint information.

Concepts around this algorithm

- Quads
- Difference Expansion Transform / Inverse
- Embedding Capacity of a region



Difference Expansion Transform

Forward difference expansion transform:

$$v_0 = \left\lfloor \frac{u_0+u_1+u_2+u_3}{4} \right\rfloor$$
$$v_1 = u_1 - u_0$$
$$v_2 = u_2 - u_0$$
$$v_3 = u_3 - u_0$$

Expandable quad:

$$v_0 = \left\lfloor \frac{u_0+u_1+u_2+u_3}{4} \right\rfloor$$
$$\tilde{v}_1 = 2 \times v_1 + b_1$$
$$\tilde{v}_2 = 2 \times v_2 + b_2$$
$$\tilde{v}_3 = 2 \times v_3 + b_3$$

‘Expandable’ practically means that information may be encoded in the region where this quad was selected from.

Here v1, v2, v3, and v4 are all below the limit value (255 for 8-bit).

Inverse Difference Expansion Transform

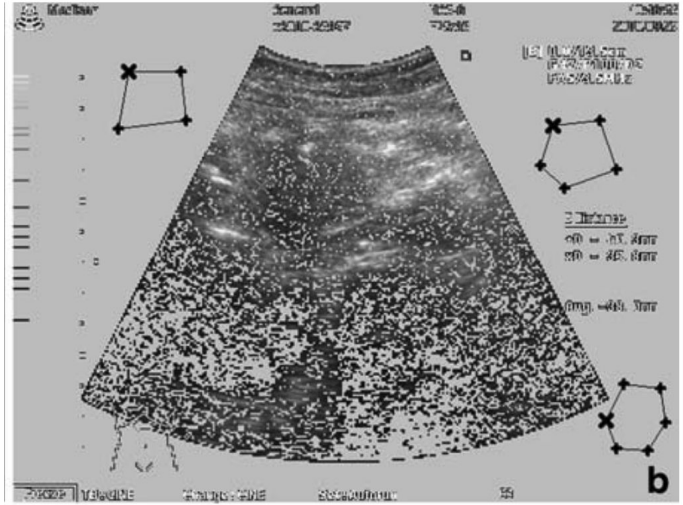
Used in the final embedding step - reconstruction of the image so the watermarking is non obvious.

$$u_0 = v_0 - \left\lfloor \frac{v_1+v_2+v_3}{4} \right\rfloor$$

$$u_1 = v_1 + u_0$$
$$u_2 = v_2 + u_0$$
$$u_3 = v_3 + u_0$$

Region of Embedding (ROE) & Embedding Capacity

A region inside which all the elements of the quads are below a certain threshold. Embedding capacity depends upon the total number of expandable quads in the region and the shape and size of the region.



$$I_E = 3N_E - \|n_v\| - n_v \times \|v(x,y)\|$$

N_E is the number of expandable quads in the ROE.

Watermarking Steps - Payload generation

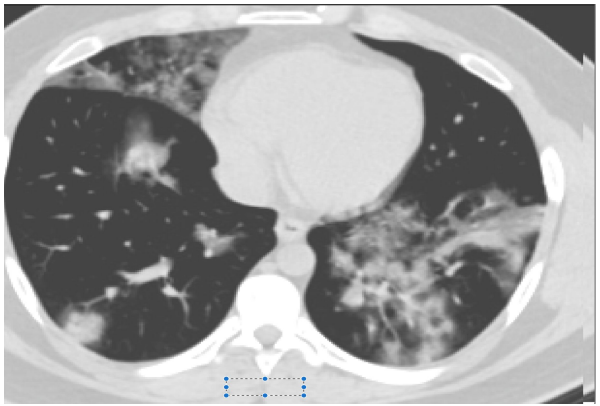
1. Hashing the Image
2. Computing a digital signature (of the hash) using the RSA cryptosystem
3. Concatenating the digital signature with the secret patient information to generate the final payload

$$DS = RSA_E\left(K_{priv}, H\right)$$

$$P = D \oplus DS$$

Watermarking steps - Embedding the data

1. Computing the possible ROEs by using the DET and the expandable quads
2. From the ROE, obtain a subset which has the required amount of embedding capacity as per the embedding capacity equation
3. The payload computed is embedded 3 bits at a time within each quad of the ROE



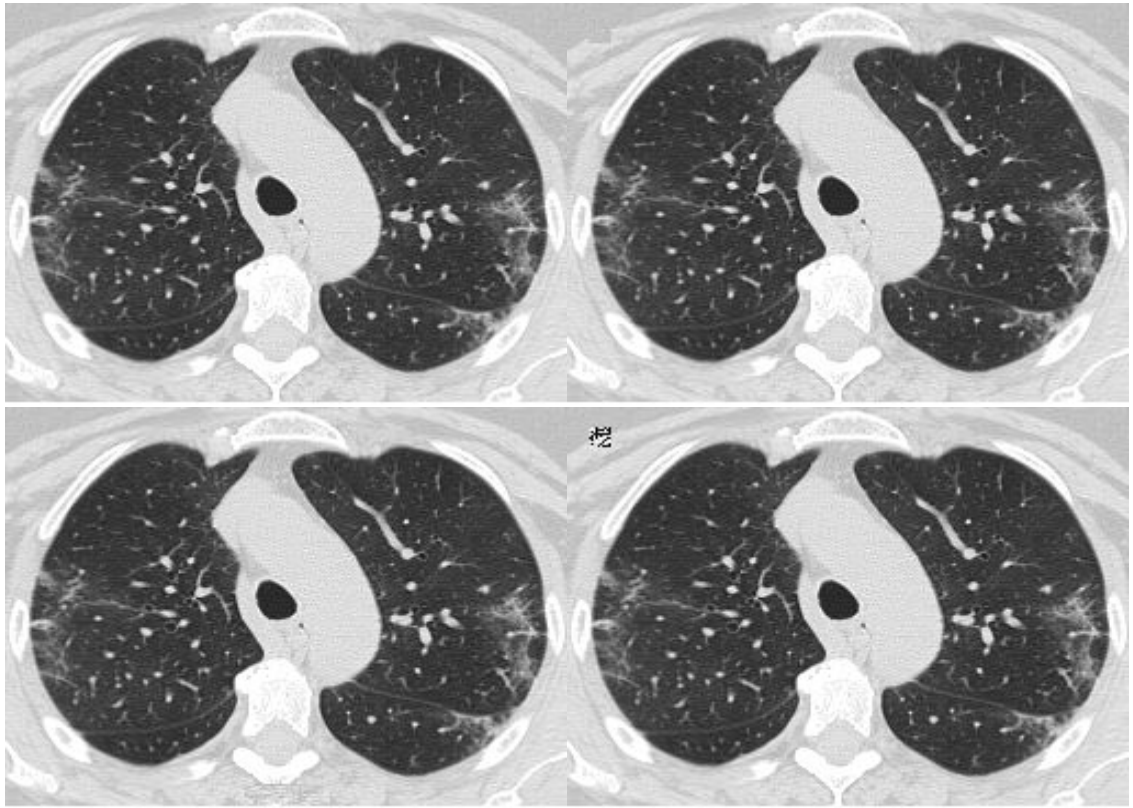
Example - Ultrasound

Original Image

Watermarked Image

Difference

Example: CT Scan



Guo Zhuang pros

1. Completely lossless
2. Quad system makes data extraction non-intuitive
3. The payload is encrypted multiple times, with multiple different systems

Guo Zhuang drawbacks

1. Finding a good ROE tends to be a manual process
2. The amount of information that can be embedded depends on the image resolution

Region of Non-Interest Watermarking (RONI)

RONI Pipeline

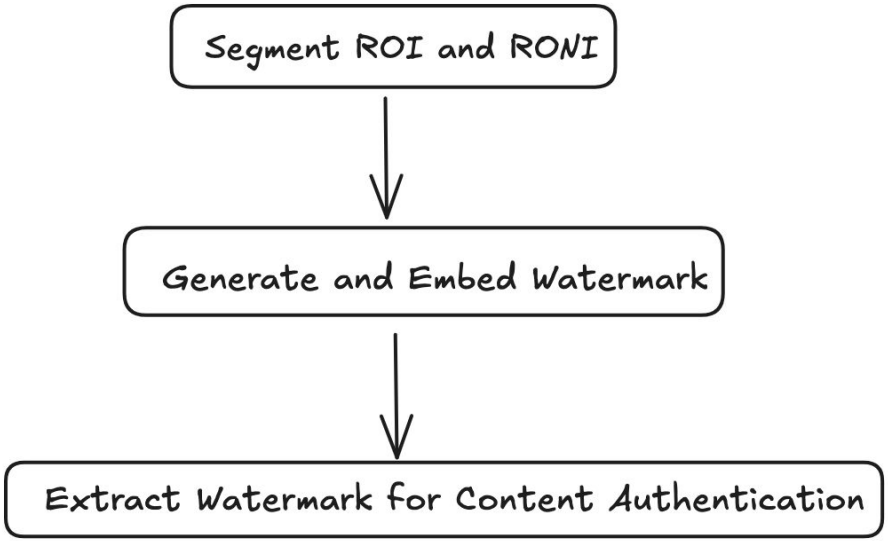
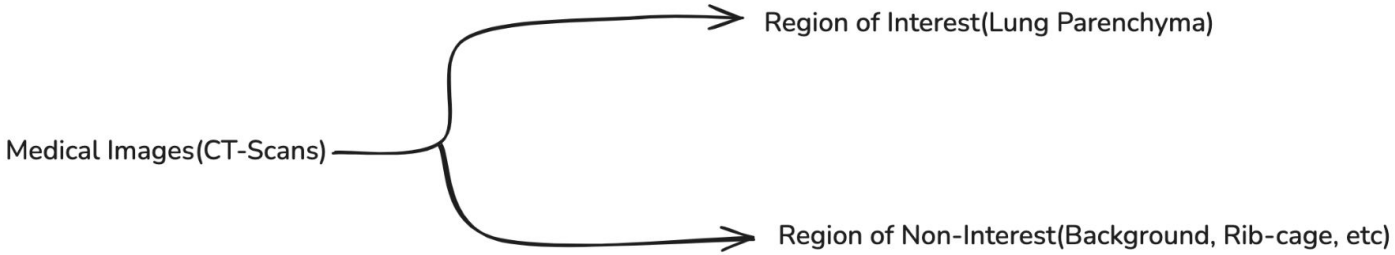


Figure Source: Menon et.al, IEEE, 2009

Segmenting ROI and RONI

Segment ROI and RONI



Region of Interest(ROI)

- Contains Information crucial for Diagnoses.
- Should suffer Least distortion from Watermark

Region of Non Interest(RONI)

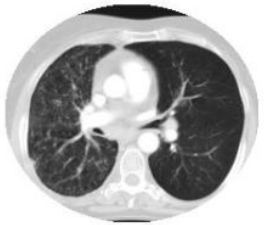
- Ideal for Including Watermark

Naive Approaches to Segmenting ROI and RONI

- Draw a Square Boundary around Lungs
- Draw an Elliptical Boundary around Lungs

Drawback: Wasted RONI area which could be used to store more cryptographic information, like patient and hospital log.

ROI using Square

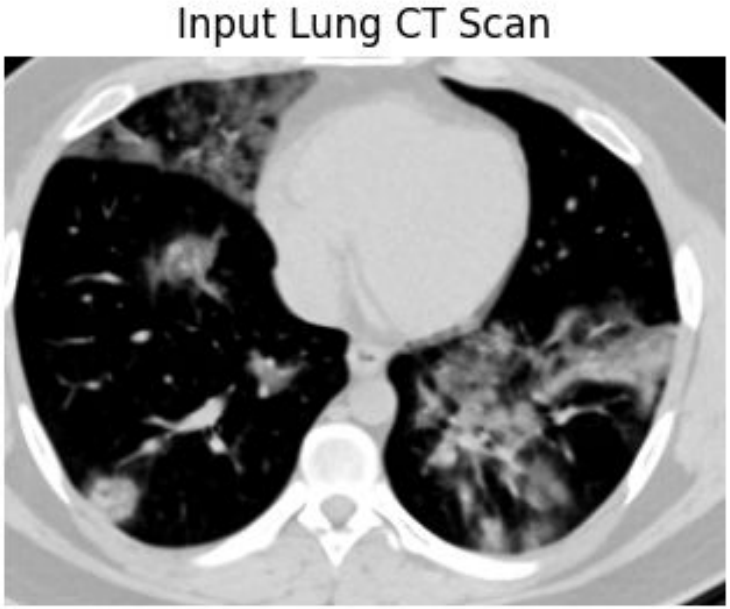


ROI using Ellipse

Figure Source: Menon et.al, IEEE, 2009

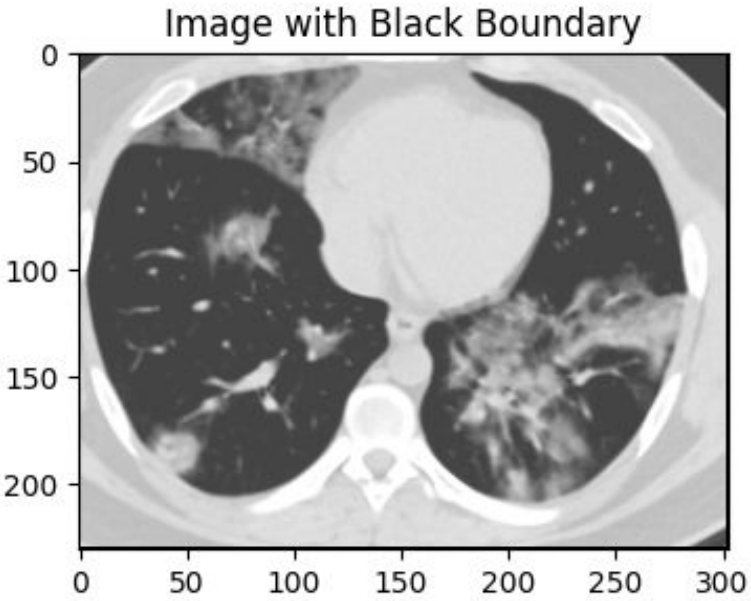
Segmenting ROI and RONI

Step 1: Read Input Image



Segmenting ROI and RONI

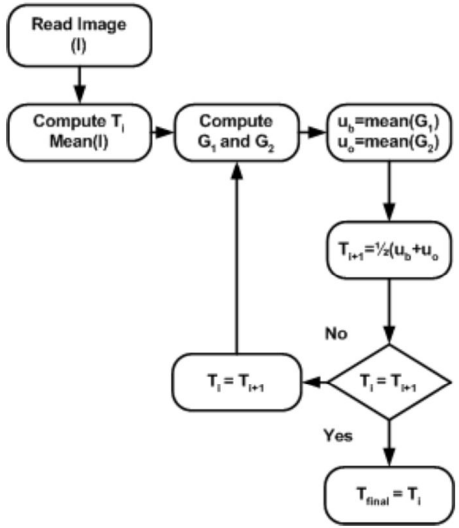
Step 2: Draw a Black Boundary on edge of Input Image



Segmenting ROI and RONI

Step 3: Find Gray Threshold of the input image using Otsu's Method.

Otsu's method is a segmentation Algorithm aimed at maximizing Inter-class variance and Lowering Intra-class Variance.

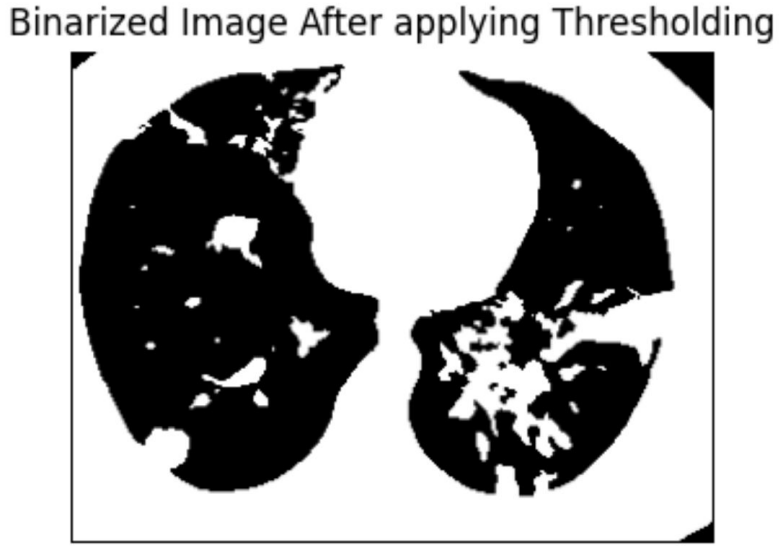


Flowchart of Otsu's Algorithm.

Figure Source-Memon et.al,IEEE, (2009)

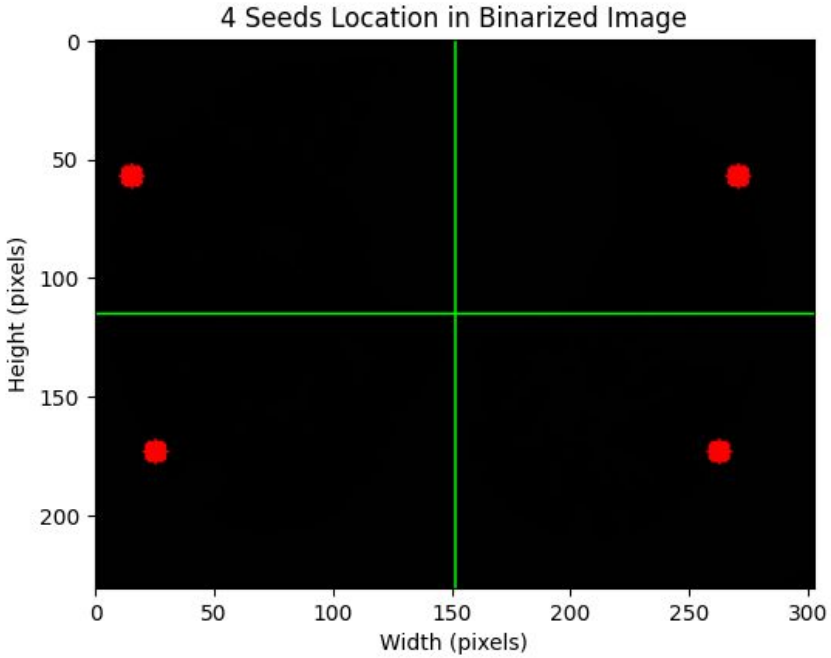
Segmenting ROI and RONI

Step 4: Get binarized image by turning pixels above threshold white and below threshold black.



Segmenting ROI and RONI

Step 5: Identify Seeds for 4 Quadrants

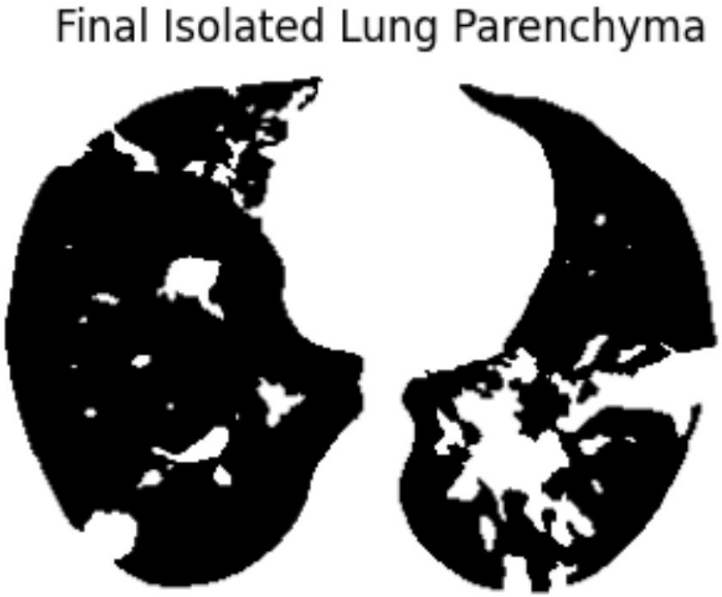


Segmenting ROI and RONI

Step 6: Make a tagged image from binarized Image in which each white pixel (255) is set to 1, and black pixels are set to 0. Used for creating visited mask to keep track of visited pixels.

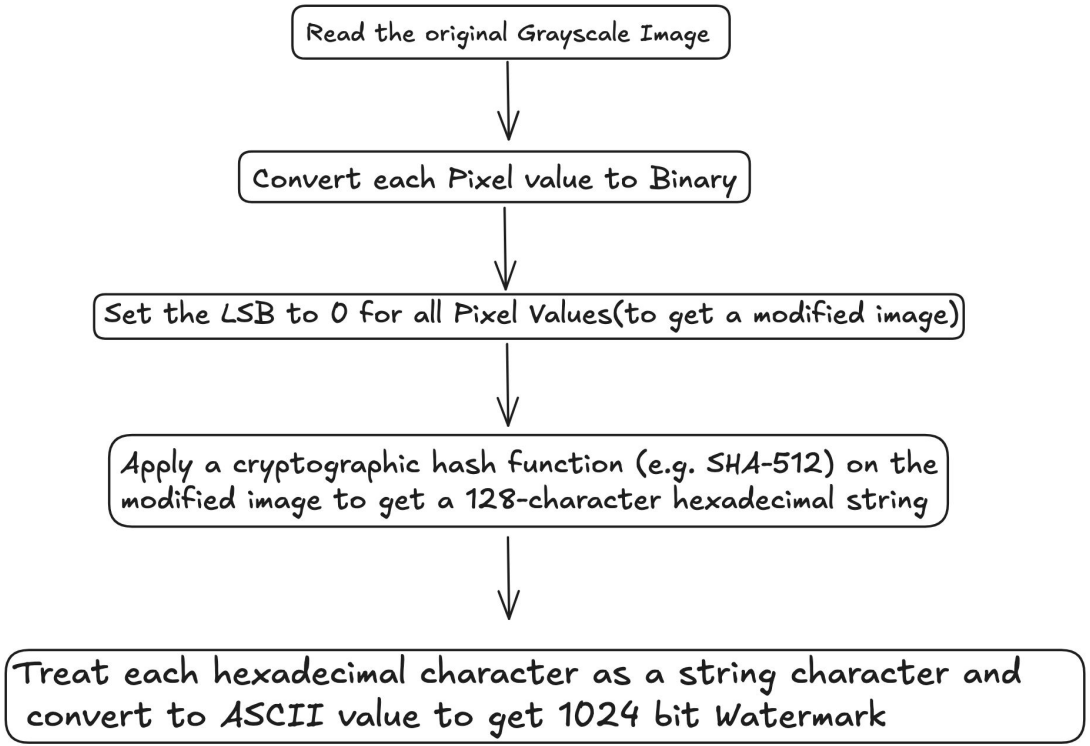
Segmenting ROI and RONI

Step 7: Region growing process using 4 seeds and a Dequeue to get ROI

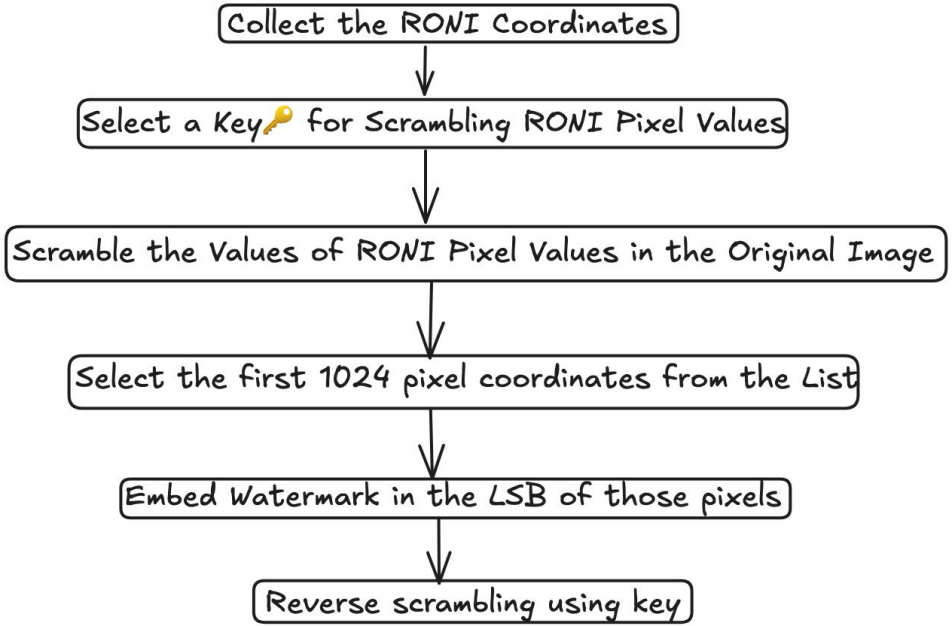


Generating and Embedding the Watermark

Generating Watermark



Embedding the Watermark



Use of Key in scrambling

x1,y1	227
x2,y2	78
x3,y3	63
x4,y4	128
x5, y5	150

Scramble(Using Key 1)

x1,y1	78
x2,y2	150
x3,y3	128
x4,y4	63
x5, y5	227

x1,y1	227
x2,y2	78
x3,y3	63
x4,y4	128
x5, y5	150

Scramble(Using Key 2)

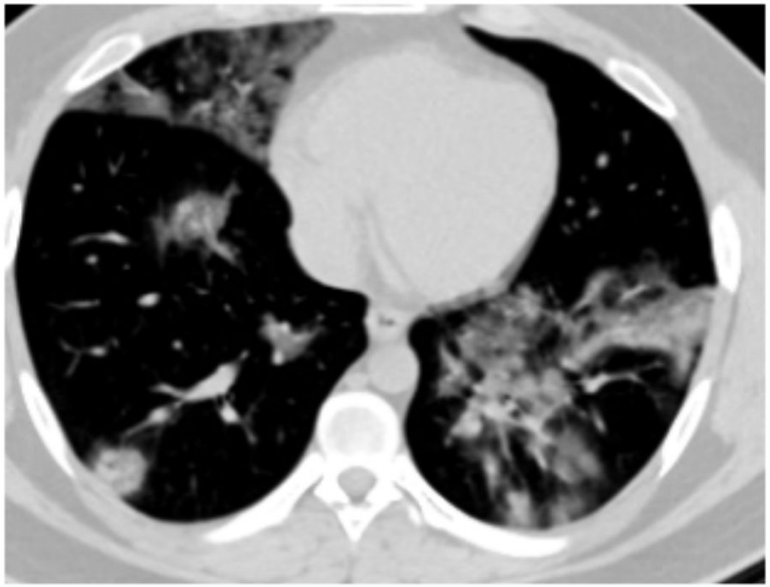
x1,y1	63
x2,y2	227
x3,y3	150
x4,y4	78
x5, y5	128

Why even do the scrambling?

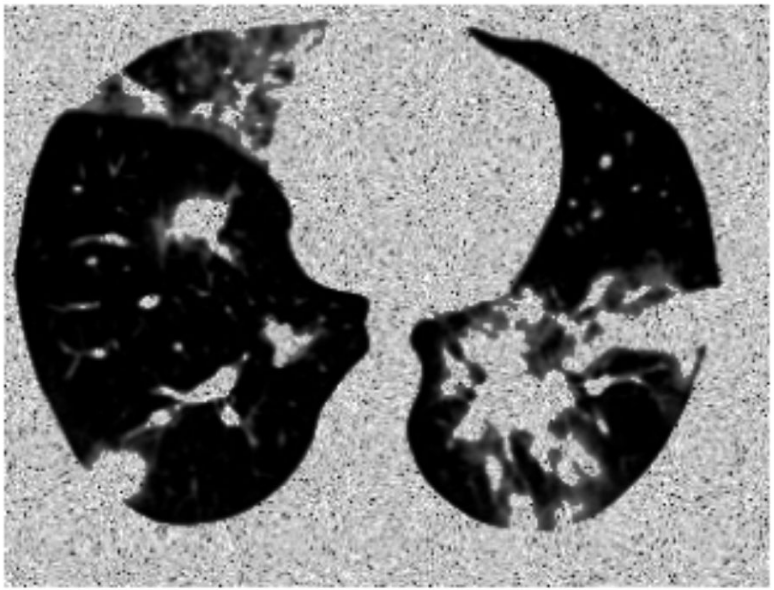
- In case adversary has Otsu Method, your approach and essentially has the RONI list
- Scrambling Key🔑 keeps the coordinates whose LSB's contain the watermark a secret

Scrambling Result

Input Lung CT Scan

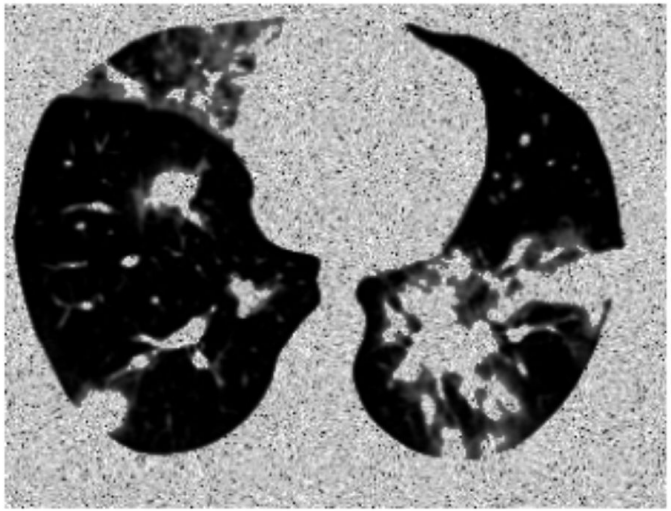


Scrambled RONI Image

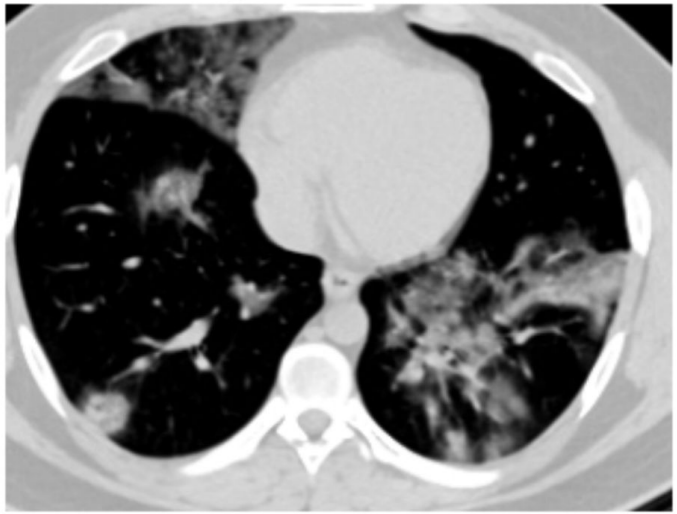


Final Result

Scrambled Image with Watermark Embedded(LSB)

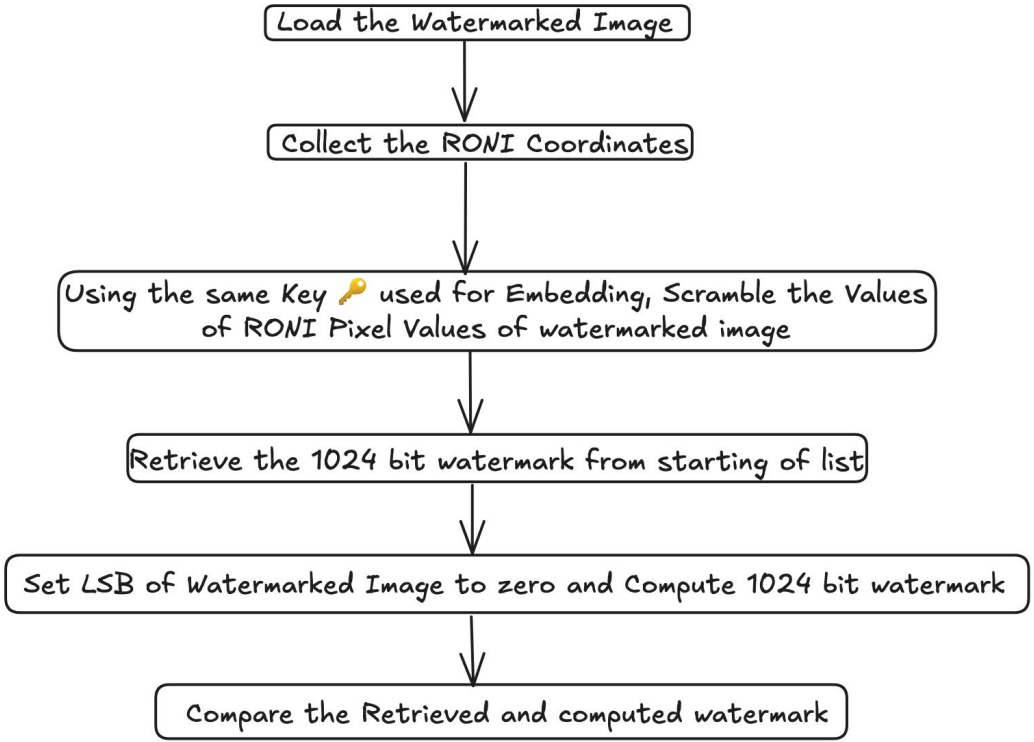


Unscrambled Image with watermark Embedded



Extracting the Watermark

Extracting the Watermark



Why Use this Technique?

- Embedded **128 characters** directly into the image (e.g., patient data)
- Great for Image Authentication purpose.
- A Blind Fragile Watermarking Technique
- Structural Similarity Index (SSIM) remains virtually unchanged at **0.99998**
- All changes confined to the **RONI (Region of Non-Interest)**
- Ensures no interference with **computer vision–based image analysis**
- With a few tweaks, the characters embedded could also include confidential patient and hospital info

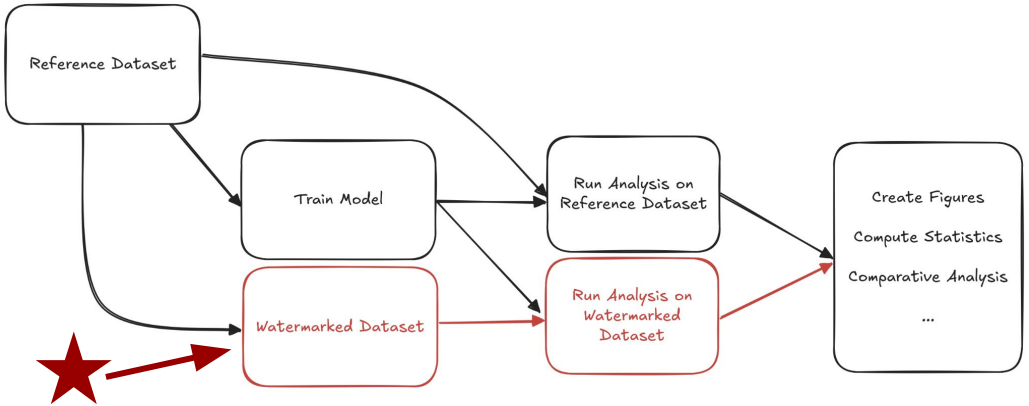
Methods

Project Setup

- **git** - Collaboration
- **uv** (https://github.com/astral-sh/uv) - Python environment setup
- **jupyter** (https://jupyter.org/) - Interactive Python environment
- **kaggle** (https://www.kaggle.com/) - Datasets

- **Dataset:** https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset
- **Repo:** https://github.com/WindowsVista42/CMSC-652-Group-Project

Method



1. Watermark dataset.
2. Run analysis using watermarked dataset.
3. Repeat steps 1 and 2 for all datasets.
4. Analyze results.

Results

Results

How do watermarks disrupt medical image analysis AI?

Hypothesis: As structural similarity (SSIM) between the original and watermarked images decreases the classification accuracy also decreases.

Observation: Accuracy decreases overall.
The number of false negatives increases while the number of false positives does not change.

Reference Paper Results

X-rays — DenseNet169 — Original Accuracy = 95.9%												
L-Bits	**Embed. Strength = 50**			**Embed. Strength = 100**			**Embed. Strength = 200**			**Embed. Strength = 300**		
	SSIM (%)	$p1, p2$	Acc. (%)	SSIM (%)	$p1, p2$	Acc. (%)	SSIM (%)	$p1, p2$	Acc. (%)	SSIM (%)	$p1, p2$	Acc. (%)
100	99.4	0.8, 0.4	95.3	99.2	0.9, 0.3	95.0	98.4	0.8, 0.5	94.0	97.5	0.8, 0.7	94.3
200	99.3	0.8, 0.4	95.0	98.7	0.1, 0.8	95.0	97.0	0.8, 0.5	93.7	95.2	0.8, 0.5	91.2
300	99.0	0.8, 0.4	95.3	98.2	0.1, 0.5	94.3	95.6	0.7, 0.5	92.5	93.0	0.8, 0.5	89.3
400	98.9	0.8, 0.8	95.0	97.6	0.8, 0.5	94.0	94.2	0.7, 0.5	91.5	90.9	0.8, 0.5	89.3
500	98.7	0.8, 0.5	94.6	97.1	0.7, 0.5	93.7	92.9	0.7, 0.5	90.3	88.9	0.7, 0.4	88.4
600	98.5	0.8, 0.4	95.0	96.5	0.7, 0.5	94.0	91.5	0.7, 0.4	89.6	86.9	0.6, 0.5	85.9
700	98.3	0.9, 0.3	94.6	95.9	0.7, 0.4	93.4	90.2	0.7, 0.3	88.4	84.9	0.6, 0.5	85.3
800	98.2	0.9, 0.3	95.0	95.3	0.2, 0.3	93.7	88.8	0.7, 0.5	87.8	83.0	0.6, 0.5	84.0
900	97.9	0.9, 0.4	94.6	96.7	0.7, 0.4	91.5	87.4	0.7, 0.5	87.1	81.0	0.6, 0.5	80.6
1000	97.6	0.8, 0.6	94.6	94.0	0.7, 0.4	91.5	85.9	0.6, 0.5	85.0	79.0	0.6, 0.5	80.3

(Apostolidis and Papakostas 2022)

Reference Paper Results

X-rays – DenseNet169 – Original Accuracy = 95.9%												
L-Bits	**Embed. Strength = 50**			**Embed. Strength = 100**			**Embed. Strength = 200**			**Embed. Strength = 300**		
	SSIM (%)	*p*1, *p*2	Acc. (%)	SSIM (%)	*p*1, *p*2	Acc. (%)	SSIM (%)	*p*1, *p*2	Acc. (%)	SSIM (%)	*p*1, *p*2	Acc. (%)
1000	97.6	0.8, 0.6	94.6	94.0	0.7, 0.4	91.5	85.9	0.6, 0.5	85.0	79.0	0.6, 0.5	80.3

	Name	Strength	Position	L-Bits	Mean SSIM	Accuracy
0	Unaltered	N/A	N/A	N/A	1.000000	0.930481
1	Krawtchouk	50	(0.5, 0.5)	1024	0.988022	0.925134
2	Krawtchouk	100	(0.5, 0.5)	1024	0.963324	0.898396
3	Krawtchouk	200	(0.5, 0.5)	1024	0.910654	0.866310
4	Krawtchouk	300	(0.5, 0.5)	1024	0.868889	0.834225

(Apostolidis and Papakostas 2022)

Our Results (DenseNet169)

	Name	Strength	Position	L-Bits	Mean SSIM	Mean PSNR (dB)	Accuracy	Precision	Recall	F1 Score
0	Unaltered	N/A	N/A	N/A	1.000000	∞	0.930481	0.948905	0.955882	0.952381
1	Krawtchouk	50	(0.5, 0.5)	1024	0.988022	42.098871	0.925134	0.941606	0.955556	0.948529
2	Krawtchouk	100	(0.5, 0.5)	1024	0.963324	36.593767	0.898396	0.927007	0.933824	0.930403
3	Krawtchouk	200	(0.5, 0.5)	1024	0.910654	31.217936	0.866310	0.927007	0.894366	0.910394
4	Krawtchouk	300	(0.5, 0.5)	1024	0.868889	28.401042	0.834225	0.963504	0.835443	0.894915
5	Guo-Zhuang	N/A	N/A	75	0.998816	52.394887	0.930481	0.948905	0.955882	0.952381
6	Guo-Zhuang	N/A	N/A	300	0.994847	35.106653	0.925134	0.948905	0.948905	0.948905
7	RONI	N/A	N/A	1024	0.999941	68.406802	0.930481	0.948905	0.955882	0.952381

Our Results (DenseNet169)

	Name	Strength	Position	L-Bits	Number of Test Images	True Positives	False Negatives	True Negatives	False Positives
0	Unaltered	N/A	N/A	N/A	187	44	6	130	7
1	Krawtchouk	50	(0.5, 0.5)	1024	187	44	6	129	8
2	Krawtchouk	100	(0.5, 0.5)	1024	187	41	9	127	10
3	Krawtchouk	200	(0.5, 0.5)	1024	187	35	15	127	10
4	Krawtchouk	300	(0.5, 0.5)	1024	187	24	26	132	5
5	Guo-Zhuang	N/A	N/A	75	187	44	6	130	7
6	Guo-Zhuang	N/A	N/A	300	187	43	7	130	7
7	RONI	N/A	N/A	1024	187	44	6	130	7

!!!

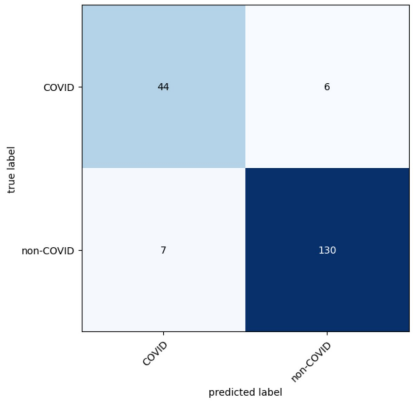
	precision	recall	f1-score	support
COVID	0.83	0.48	0.61	50
non-COVID	0.84	0.96	0.89	137
accuracy			0.83	187
macro avg	0.83	0.72	0.75	187
weighted avg	0.83	0.83	0.82	187

!!!

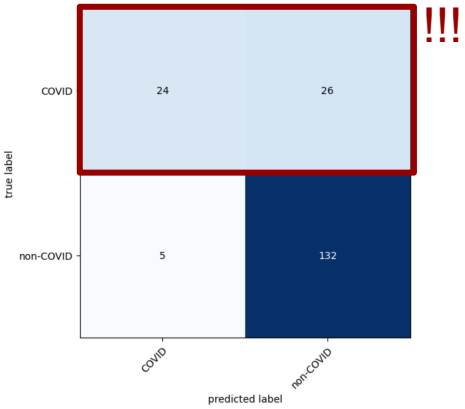
Krawtchouk S=300

DenseNet169 Confusion Matrix

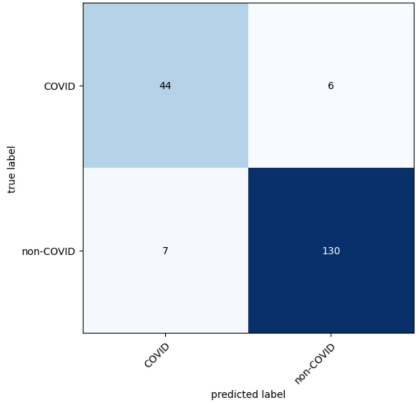
Unmodified



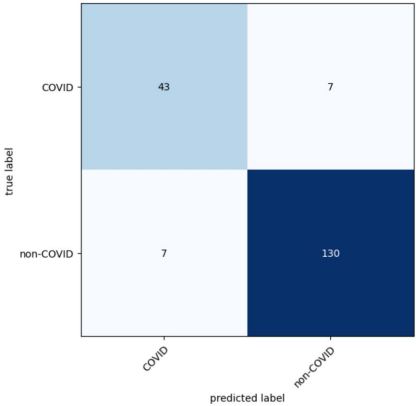
Krawtchouk S=300 !!!



RONI

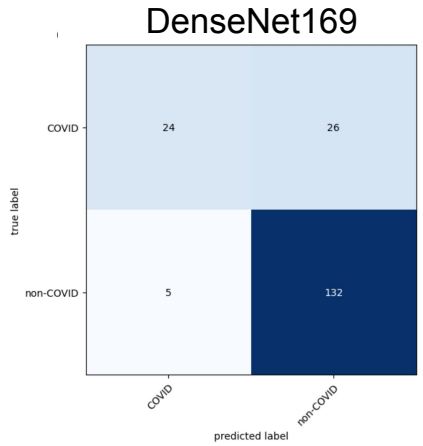


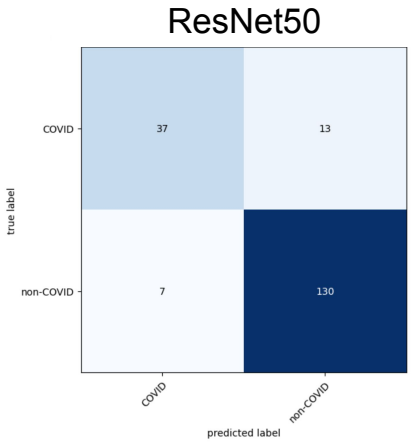
Guo-Zhuang L=300

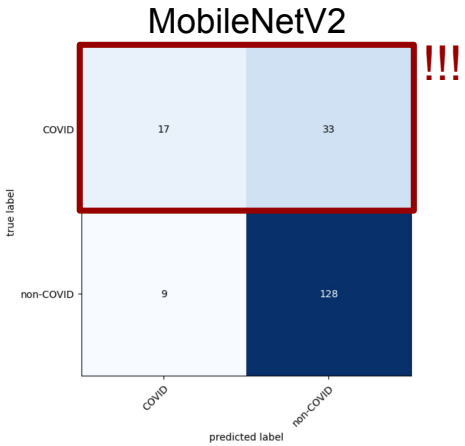


High rate of false negatives.

Coincidence?







Sharp increase in false negatives with all models.

This is only with Krawtchouk moments watermarks.

With Krawtchouk S=300

Our Results (DenseNet169)

	Name	Strength	Position	L-Bits	Mean SSIM	Mean PSNR (dB)	Accuracy	Precision	Recall	F1 Score
0	Unaltered	N/A	N/A	N/A	1.000000	∞	0.930481	0.948905	0.955882	0.952381
1	Krawtchouk	50	(0.5, 0.5)	1024	0.988022	42.098871	0.925134	0.941606	0.955556	0.948529
2	Krawtchouk	100	(0.5, 0.5)	1024	0.963324	36.593767	0.898396	0.927007	0.933824	0.930403
3	Krawtchouk	200	(0.5, 0.5)	1024	0.910654	31.217936	0.866310	0.927007	0.894366	0.910394
4	Krawtchouk	300	(0.5, 0.5)	1024	0.868889	28.401042	0.834225	0.963504	0.835443	0.894915
5	Guo-Zhuang	N/A	N/A	75	0.998816	52.394887	0.930481	0.948905	0.955882	0.952381
6	Guo-Zhuang	N/A	N/A	300	0.994847	35.106653	0.925134	0.948905	0.948905	0.948905
7	RONI	N/A	N/A	1024	0.999941	68.406802	0.930481	0.948905	0.955882	0.952381

The worst performers had a lowest SSIM.

Conclusion

Conclusion

What can we make of this?

Lower SSIM leads to reduced classification accuracy in AI models.

- *Should not be surprising.*
- **Use-case awareness is *critical* in choosing watermarking methods.**
- Trade-off between robustness and image fidelity is real and significant.
- False negatives increase as SSIM drops.
  - **These are missed cases.**
  - **Make sense:** “Has Covid-19” is a small subset of “Does not have Covid-19”.

Mitigations

- Understand your data.
  - Data preprocessing.
  - Choosing the right model.
    - ResNet50 > MobileNetV2.
- Train for robustness.
  - Add noise or transformations.
- **Understand how your model fails.**
  - Does it degrade gracefully?

Unaltered

Krawtchouk 50

RONI

Guo-Zhuang

Acknowledgements

Additional tools that made this possible:

- **Zotero** (https://www.zotero.org/) - Bibliography
- **Excalidraw** (https://excalidraw.com/) - Figures
- **ChatGPT** (https://chatgpt.com/) - Editing/Coding
- **Perplexity AI** (https://www.perplexity.ai/) - Coding

References

Apostolidis, Kyriakos D., and George A. Papakostas. 2022. “Digital Watermarking as an Adversarial Attack on Medical Image Analysis with Deep Learning.” *Journal of Imaging* 8 (6): 155. https://doi.org/10.3390/jimaging8060155.

“Brain Tumor MRI Dataset.” n.d. Accessed March 9, 2025. https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

Coatrieux, G., H. Maitre, B. Sankur, Y. Rolland, and R. Collorec. 2000. “Relevance of Watermarking in Medical Imaging.” In *Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine. ITAB-ITIS 2000. Joint Meeting Third IEEE EMBS International Conference on Information Technology Applications in Biomedicine (ITAB’00). Third Works*, 250–55. Arlington, VA, USA: IEEE. https://doi.org/10.1109/ITAB.2000.892396.

References

“Covid-19 Binary Classification | DenseNet169 | 98%.” n.d. Accessed March 9, 2025. https://kaggle.com/code/ahmedtronic/covid-19-binary-classification-densenet169-98.

Guo, Xiaotao, and Tian-ge Zhuang. 2009. “A Region-Based Lossless Watermarking Scheme for Enhancing Security of Medical Data.” *Journal of Digital Imaging: The Official Journal of the Society for Computer Applications in Radiology* 22 (1): 53–64. https://doi.org/10.1007/s10278-007-9043-6.

Memon, Nisar A., S.A.M. Gilani, and Asad Ali. 2009. “Watermarking of Chest CT Scan Medical Images for Content Authentication.” In *2009 International Conference on Information and Communication Technologies*, 175–80. https://doi.org/10.1109/ICICT.2009.5268167.

“MachineLearningVisionRG/KMsWA2.” (2022) 2024. Python. MachineLearningVisionRG. https://github.com/MachineLearningVisionRG/KMsWA2.

References

"SARS-COV-2 Ct-Scan Dataset." n.d. Accessed March 9, 2025. https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset.

Venkataramana, A., and P. Ananth Raj. 2007. "Image Watermarking Using Krawtchouk Moments." In 2007 International Conference on Computing: Theory and Applications (ICCTA'07), 676–80. https://doi.org/10.1109/ICCTA.2007.72.

"SARS-COV-2 Ct-Scan Dataset." n.d. Accessed April 13, 2025. https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset.

Nyeem, Hussain, Wageeh Boles, and Colin Boyd. 2012. "On the Robustness and Security of Digital Image Watermarking." In . https://doi.org/10.1109/ICIEV.2012.6317496.

Mademlis, Athanasios, Petros Daras, Dimitrios Tzovaras, and Michael G Strintzis. n.d. "3D VOLUME WATERMARKING USING 3D KRAWTCHOUK MOMENTS."

References

“Digital Watermarking.” 2024. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Digital_watermarking&oldid=1256940744.

“Hypergeometric Function.” 2025. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Hypergeometric_function&oldid=1283589438.

“Image Moment.” 2025. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Image_moment&oldid=1282294709.

“Medical Image Computing.” 2024. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Medical_image_computing&oldid=1254933632.

Mousavi, Seyed Mojtaba, Alireza Naghsh, and S. A. R. Abu-Bakar. 2014. “Watermarking Techniques Used in Medical Images: A Survey.” *Journal of Digital Imaging* 27 (6): 714–29. https://doi.org/10.1007/s10278-014-9700-5.

References

Papakostas, G.A., E.D. Tsougenis, and D.E. Koulouriotis. 2014. “Moment-Based Local Image Watermarking via Genetic Optimization.” *Applied Mathematics and Computation* 227 (January):222–36. https://doi.org/10.1016/j.amc.2013.11.036.

Kwee, Thomas C., and Robert M. Kwee. 2020. “Chest CT in COVID-19: What the Radiologist Needs to Know.” *RadioGraphics* 40 (7): 1848–65. https://doi.org/10.1148/rg.2020200159.