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

May 11, 2025

Sean Moulton, Omkar Kulkarni, Anupreet Singh
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Cir, Baltimore, MD 21250, USA
seanmoulton@umbc.edu, omkar.kulkarni@umbc.edu, anuprel1@umbc.edu

***Abstract*—With the rise of AI systems capable of generating synthetic media, the integrity of digital content has become a critical concern. Efforts like C2PA aim to use watermarking to embed provenance and authorship data directly into media to counter misinformation. While promising for general media integrity, the implications of watermarking in high-stakes domains such as healthcare should be explored further. This work investigates the impact of various watermarking algorithms on medical images analysis, focusing on how they influence the performance of deep learning–based computer vision tasks such as image classification. We apply three different watermarking algorithms to a Covid-19 lung CT scan dataset and evaluate the performance of three different deep learning models on a classification task. We perform a detailed comparative study of how each watermarking algorithm affects the performance of the deep learning models. Our results show that fragile watermarking does not lead to a significant degradation in performance, while robust watermarking significantly degrades the performance of some deep learning medical image analysis systems, specifically through an increase in the number of false negatives. In the context of Covid-19 detection, this degradation could lead to missed diagnosis, delayed treatment, and increased risk of disease spread. These findings highlight the need for watermarking methods to be evaluated not just in terms of visual imperceptibility and robustness, but also in terms of their downstream effects on model decision boundaries and diagnostic reliability. As AI tools become increasingly integrated into healthcare, understanding these interactions is essential.**

***Keywords*—Digital signatures, Watermarking, Medical Image analysis, Scans, Computer tomography, Cryptography, Computer vision, generative AI, public key cryptography.**

I. Introduction

With the rise of artificial intelligence systems capable of generating synthetic medical imagery, the integrity of medical scans has become a critical concern. AI-generated CT scans have been shown to improve image detection models [1], but this same technology could be used to create falsified CT scans of real patients, leading to serious medical and ethical implications. To counter this, digital watermarking has emerged as a method to verify the authenticity of medical images.

Efforts like the Coalition for Content Provenance and Authenticity (C2PA) aim to address growing concerns around the provenance and integrity of digital media. The C2PA specification emphasizes flexibility in provenance storage, allowing the option of embedding the information directly within the asset file [2]. Digital watermarking aligns with this framework by presenting a method to store provenance and author data directly in asset files without significantly altering their visual quality. In exploring watermarking in the context of medical imagery, we can begin to understand how standards like C2PA might affect downstream tasks automated by computer vision and AI.

Computer vision plays a crucial role in medical image analysis, aiding in the detection, classification, and diagnosis of diseases. Again, the work by Mangalagiri et al. [1] shows a direct application of computer vision in medical image analysis on the detection of COVID-19 with a computer vision model. However, these models are highly dependent on the integrity of their input data and can be triggered by trivial changes such as the one-pixel attack [3]. Thus, it is important to study watermarking in the context of computer vision to ensure that it's effect is well understood.

Some of this effect has been explored by Apostolidis and Papkostas [4] in the context of medical image analysis. Their findings indicate that digital watermarking can be used to degrade the accuracy of deep learning based computer vision algorithms used in medical image analysis. However, their findings are limited to the context of a single algorithm. We extend their method by analyzing multiple watermarking algorithms and their impact on the medical image analysis task used. In doing so, we provide a deeper understanding of the trade-offs between security, image quality, and algorithmic performance.

II. Related Work

Various researchers have focused on exploring digital watermarking techniques in images and other media. Some of the first research we explored was from Zhou, Huang, and Lou [5], who presented a method to verify the authenticity and integrity of digital mammography. Their work, while having potential security issues today [6], set the foundation

for future exploration. Jian Ren and Tongtong Li [7] proposed a computationally efficient cryptographic watermarking technique using signal processing techniques. Kuang, Zhang, and Han [8] proposed a system for authenticating medical images using reversible digital watermarking based on the RSA cryptosystem. This method proves to be effective, but public access to data proves to be a security concern.

With the advent of new AI-driven content generation technologies, it is now easier than ever for anyone to generate fake medical imagery. A study by Mangalagiri et al. [1] presents a method to improve computer vision algorithms using AI-generated images. While this study shows promising results, it also shows that AI-driven content generation is improving at a rapid pace. It shows that there is a growing need for systems such as C2PA to trace the author and origin of media. Watermarking is a potential method for embedding the required provenance and authorship information.

Apostolidis and Papakostas in 2022 [4] explored the effects of watermarking on medical image analysis. They implemented a moment-based local image watermarking method on three modalities: Magnetic Resonance Imaging (MRI), Computed Tomography (CT) scans and X-ray images. The particular watermarking technique used focuses on embedding watermarks into specific regions of the image, utilizing image moments to ensure the watermark is both imperceptible and robust. Three state-of-the-art deep learning models, DenseNet201, DenseNet169, and MobileNetV2 were used for the medical image analysis. Their work shows that digital watermarking reduces the accuracy of deep learning-based computer vision algorithms used in medical image analysis, with MobileNetV2 being the most vulnerable suffering over a 50% reduction in accuracy with CT scans being associated with the highest degradation in performance. We believe a comprehensive understanding of how other similar watermarking techniques impact such deep learning based computer vision algorithms remains crucial.

We studied research proposing and implementing numerous watermarking algorithms as seen in [7]–[13]. However, no prior work has systematically compared whether these watermarking techniques produce different effects than the method in [4]. In particular, Guo and Zhuang [12] propose a difference expansion transform-based algorithm, both of which offer promising alternative approaches. This gap motivates our investigation into how these techniques compare in terms of their impact on deep learning-based medical image analysis models.

III. Specific Aims

This project aims to analyze the effects that different watermarking algorithms have on computer vision based algorithms used in medical image analysis. We will be extending the existing work by Apostolidis and Papakostas [4] to analyze more watermarking algorithms. The new algorithms we have chosen include a difference expansion region-selection method introduced by Guo and Zhuang [12] and a blind fragile watermarking scheme [14] that only implements the watermark in Region of Non Interest(RONI) in the lung CT scan. These two algorithms, along with the algorithm originally used by Apostolidis and Papakostas [11] represent distinctly different approaches used in digital watermarking. From our analysis we present our findings about the effects that digital watermarking has on computer vision based algorithms used in medical image analysis. **Our Specific aims are:**

- Replicate the findings of [4] using the same CT scan dataset [15] and similar computer vision classification models: ResNet50, DenseNet169, and MobileNetV2.
- Perform the prior analysis using two additional watermarking algorithms [12], [14].
- Identify the characteristics of a watermarking algorithm that affect classification accuracy.

IV. Krawtchouk Moment Based Watermarking

Image moments are a weighted average (moment) computed from the value of pixels in an image. The discrete formula takes the following form:

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

This formula can be extended using krawtchouk moments [16] of the following form:

$$K_n(x; p, N) = {}_2F_1\left(-n, -x; -N; \frac{1}{p}\right)$$

Where $x, n = 0, 1, 2, ..., N$, $N > 0$, $p \in (0, 1)$, and ${}_2F_1$ is the hypergeometric function. Additional information, included weighted krawtchouk moments as used here is described in [4].

Krawtchouk moments are very effective for describing the local properties of the image which can be described by position parameter [4].

Embedding takes place by computing the krawtchouk moments of the image, dithering by a modulation provided by a bit sequence, and subtracting the modulation back out to construct a watermark. The constructed watermark can then be added to the original image to produce a watermarked image [4].

Figure 1 shows the watermark embedding technique as outlined in [4].

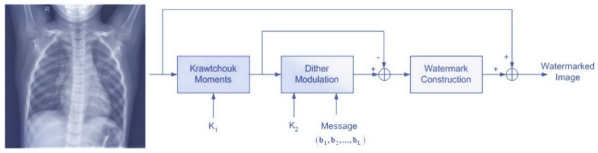


Fig. 1: Krawtchouk Watermarking Method

More details on watermarking with krawtchouk moments are given in [4] and [17].

V. Guo and Zhuang Watermarking

The watermarking technique developed by Guo and Zhuang [12] has the ability to embed data without introducing visible distortions in medical images. By strategically selecting embedding regions outside the region of interest (ROI), the method preserves diagnostic quality while incorporating critical security features. The algorithm uses the entire image along with sensitive/secret patient information to generate a secure embedding, such that it can be determined with a high probability if the image has been tampered with. The secret patient information can also be the patient's fingerprint, thus allowing biometric authentication when accessing the medical image.

We describe some concepts that the algorithm uses when generating the embedded payload.

1) Quads: A quad is a vector $u = (u_0, u_1, u_2, u_3)$ formed from 2x2 adjacent pixel values according to a predetermined order. For our implementation, we follow the raster order. Since we use such quads, and there are 24 possible permutations to choose an order, the order itself may serve as a security key. Quads are non-overlapping, so each pixel belongs to only one quad.

2) Difference Expansion Transform: The difference expansion transform $v = f(u)$ for a vector u is defined as follows:

$$\begin{aligned} v_0 &= \lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \rfloor \\ v_1 &= u_1 - u_0 \\ v_2 &= u_2 - u_0 \\ v_3 &= u_3 - u_0 \end{aligned}$$

The elements of the vector v as described above should be inserted in the same order as that of u.

The inverse difference expansion transform for a transformed quad v is defined by $u = f^-1(v)$. f^-1 is given by:

$$\begin{aligned} u_0 &= v_0 - \lfloor \tfrac{v_1+v_2+v_3}{4} \rfloor \\ u_1 &= v_1 + u_0 \\ u_2 &= v_2 + u_0 \\ u_3 &= v_3 + u_0 \end{aligned}$$

3) Expandable quads: The quad $u = (u_0, u_1, u_2, u_3)$ is said to be expandable if, for all values of $b_1, b_2, b_3 \epsilon$ {0,1}, $v = f(u)$ can be modified to produce $\tilde{v} = (v_0, v_1, v_2, v_3)$ as per the equation below without causing underflow or overflow in $\tilde{u} = f^-1(\tilde{v})$.

$$\begin{aligned} v_0 &= \lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \rfloor \\ \tilde{v_1} &= 2 \times v_1 + b_1 \\ \tilde{v_2} &= 2 \times v_2 + b_2 \\ \tilde{v_3} &= 2 \times v_3 + b_3 \end{aligned}$$

Here, $\tilde{v}$ is now a 1-bit left shifted version of the original v. For a quad with 4 elements, 3 bits of information (b_1, b_2, b_3) can be reversibly embedded.

4) Region of Embedding (ROE) and Embedding Capacity: Most of the medical images exhibit a high spatial correlation among the values of neighboring pixels. In the smooth area of the image, the difference between the values of two adjacent pixels is rather small. We exploit this to define a region of embedding within the image. In their paper [12], the authors mention how intervention by a radiologist can be needed to define a ROE; in our implementation we look at the distribution of the locations of smooth regions in a given image, and use that to define ROEs. If there are N_E expandable quads in a region, $3N_E$ bits can be reversibly embedded inside the ROE. We also need to have information about the ROE during the watermark extraction process, so we need to account for that as well. This will be vertex information - number of vertices, and the vertex coordinates. Thus the net embedding capacity of a region, I_E is given by:

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

where $||x||$ denotes the number of bits required to describe x. We now look at the steps involved in generating and embedding the watermark.

A. Generating the payload

Generating the payload involves three steps:

1) Image Hash: Since it is computationally difficult to find two images with the same hash value or even to generate another with the same hash value, we compute the image hash of the entire image.

$$H = H_{MD}(I)$$

Here, H_{MD} is the MD5 hash function. We have better hash algorithms today, the original paper uses MD5 and for this implementation so do we.

2) Digital signature: Compute the digital signature based on the above hash value H.

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

using the RSA public-key encryption system, and K_{priv} is the private key.

3) Confidential message: If S is the confidential message to be embedded, the final payload P is the concatenation of the S and DS (computed above).

$$P = S \oplus DS$$

We now encrypt the payload P using another encryption scheme (such as AES) to obtain the final bitstream to be embedded. In the original paper, the authors encode ROE information (location and length) as well along with this payload P. For the purposes of this implementation, the ROE location is known, and we consider only one ROE in one image.

B. Embedding the payload in the ROE

Embedding the payload in the ROE involves the following steps:

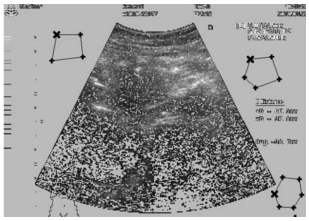
1) Choosing a ROE: A ROE is chosen such that all the quads inside the ROE are expandable quads. This can be done by a visual inspection of the image, and most smooth sections are suitable ROEs. We ensure that the ROE chosen does not intersect with the ROI. We conduct two experiments, one where the ROE has 25 expandable quads (a square at the upper left corner of the image) and another where the ROE has 100 expandable quads (a rectangle along the center of the image at the bottom edge) to embed 75 and 300 bits respectively.

2) Forming sets of quads in the ROE: The image is now scanned in a predefined order. We use the raster order. We then form a set of expandable quads inside the ROE. For each scanned quad u, we compute its forward transform $v = f(u)$, embed 3 bits of the encrypted payload P as (b_1, b_2, b_3) using difference expansion and obtain $\tilde{v}$. Next, we compute the inverse transform of $\tilde{v}$ to produce the watermarked quad $\tilde{u} = f^{-}1(\tilde{v})$. We now replace the pixel values in u with corresponding values from $\tilde{(u)}$. This last step is repeated until the desired number of bits (75 or 300 in our case) from the payload (or the entire payload) is embedded. The image generated after this process is the watermarked image.

Figure 2 shows an example of watermarking an ultrasound image. As is evident from the figures, the watermark is invisible to the naked eye, and the ROE chosen is in the black background which has nothing to do with the actual ultrasound.

Figure 3 shows our steps to watermark a 8-bit CT scan image of the lungs. CT scans are typically 12-bit images with a much higher resolution than what we currently have here (smaller than 300×300), however the dataset used by our reference paper had CT scans that were trimmed down, which is partially the reason why upon close observation, the watermarks are visible as mentioned in the figure descriptions. We also forced an overflow of bits in one of the images, and the result of that is seen in section d of the figure.

(a) 640x480 US image



(b) ROEs on US image

(c) Watermarked US image

(d) Pixel-wise differences between the watermarked and original images

Fig. 2: A reference figure from the paper, showing the watermarking of a 8-bit ultrasound image.

(a) Original CT scan image

(b) CT scan watermarked at the top left corner with 75 bits of payload

(c) CT scan watermarked at the bottom edge center with 300 bits of payload

(d) Effects of overflow/underflow at the ROE

Fig. 3: An example of watermarking a 8-bit CT scan image using the Guo & Zhuang [12]algorithm

VI. RONI Watermarking

A. Segmenting ROI and RONI

Medical images like CT scans are usually comprised of Region of Interest(ROI) and Region of Non Interest(RONI). ROI is the region that contains information crucial from diagnosis/analysis perspective, so it should suffer the least distortion possible from the watermark. Naturally, it means that the watermark must be applied in the RONI then. This approach is inspired by the methodology proposed in prior works by Memon et al. [14], [18]. We will be using CT scan images of lungs to separate the ROI and RONI.

Our approach takes the following steps to segment the ROI and RONI:

1) Read the input image.
2) Draw the black boundary on the edge of the input image.
3) Find the gray threshold of the input image using the Otsu's method, which seeks to maximize Inter-class variance while minimizing Intra-class variance.

4) Form a binarized image by turning all pixels above that threshold white and below that threshold black.
5) Segment the binarized image into four quadrants. For each quadrant, find a seed value. In this case, the seed value is a black pixel found in the middle row of the quadrant, starting from the edge of the image and skipping the outermost border to avoid selecting the boundary drawn in step 2.
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.
7) Use a dequeue to perform a region-growing process, starting from a seed point in each of the four quadrants. Each seed acts as the starting pixel, and we expand outwards by checking its neighboring pixels—specifically, the ones directly above, below, left, and right. If any of these neighboring pixels are black in the tagged image and haven't been visited yet, we add them to the queue for further exploration and mark them as visited. This way, the black region connected to the seed gets fully expanded in the final resulting image, turning those pixels black as well. Repeat this process for each quadrant's seed, ensuring that only the connected black areas are included in the final result.

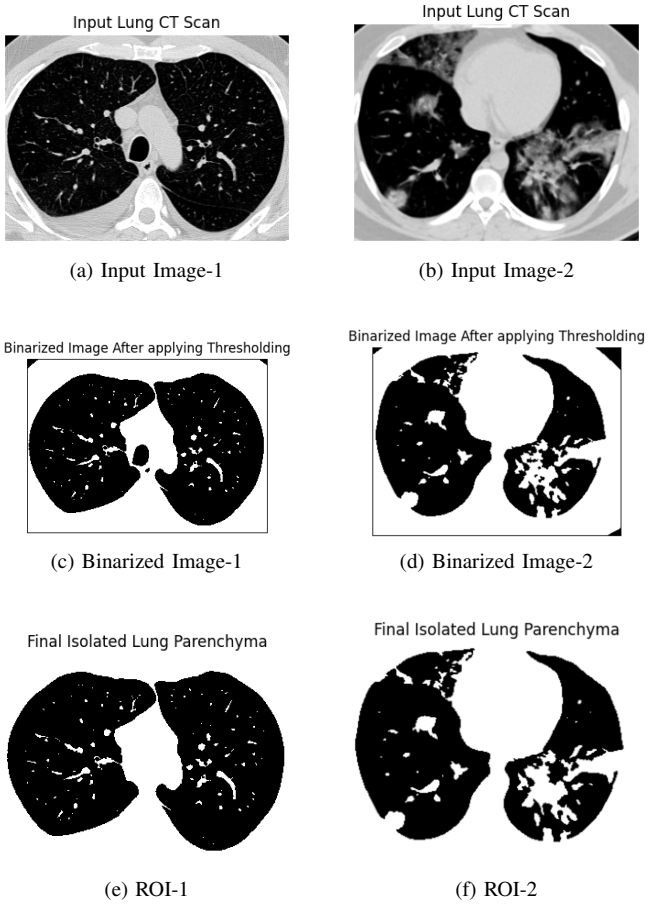
1) Increased Space for Watermark Embedding: The most naive approach for segmenting the ROI that one could think of is to form a close boundary around the lungs. A square boundary may suffice, but an elliptical one can fit more closely to the lung contours. However, even with an elliptical boundary, a significant amount of empty space remains—such as the large cavity between the two lungs—which is not part of the actual ROI. For maximum efficiency, we need a method that isolates only the lung parenchyma.

Our approach addresses it by accurately separating the lung parenchyma. In the resulting image, the black pixels represent the ROI, while the white pixels correspond to the RONI. It not only ensures that we avoid embedding the watermark into the ROI but also maximizes the available area in the RONI for watermarking. By doing so, we can embed more data into the image, which not only aids in verifying authenticity but also allows us to store important information related to the hospital and patient.

2) Visual Analysis of ROI/RONI Segmentation for Two CT Scans: Figure 4 shows the three crucial steps—arranged vertically—for two separate CT scan cases, illustrating the process of obtaining the final ROI and RONI.

For Case-1, we observe that in Input Image-1 we have have two small black smudges at the top corners of the image. These are clearly not part of the lung, but due of their Intensity, they appear as black regions in the binarized image-similar to the lungs. The final ROI successfully removes the smudges but it also removes a round black segment in between the cavity of the two lungs that may be related to lung parenchyma. What this means is, in this case the area corresponding to the round black segment may also contain watermark information. It may disturb our analysis slightly, but we still did not include watermark information in majority of the lung parenchyma region so overall we did far better in terms of possibly preventing degradation of AI analysis due to interference from watermark in the image.

For Case-2, we see the Input Image-2, we observe three black smudges on three corners of the image, which aren't part of the lung paranchyma. Due to their low intensity, these regions are also highlighted as black in the binarized image—similar to actual lung areas. However, the final segmentation step correctly excludes these smudges from the Region of Interest (ROI) while preserving the complete structure of the lung parenchyma. As a result, the watermark is embedded entirely within the Region of Non-Interest (RONI), ensuring that no diagnostic lung region is affected, and thus avoiding interference with any AI-based diagnosis.



(a) Input Image-1 (b) Input Image-2

(c) Binarized Image-1 (d) Binarized Image-2

(e) ROI-1 (f) ROI-2

Fig. 4: Step-wise visualization showing input images, binarized Images, and segmented ROI's.

B. Embedding Watermark in RONI

Once the RONI coordinates have been identified for an image, this work implements a blind fragile watermark in that part. The choice of this specific kind of watermark is was governed by the fact that it adds the functionality of content

authentication of the medical scan just from the watermarked image without requiring access to original non-watermarked image. Hence, the term ”blind”. If there is even almost any kind of perceptible change(more than 1) to a even a single pixel's value, once could identify that the image has been tampered with.

The specifics of watermark generation and embedding are outlined below:

1) Watermark Computation: The input image is loaded in grayscale mode, where each pixel value ranges from 0 to 255. Then each pixel value is represented in binary form. To ensure the watermark generated for authentication is not affected by any previously embedded data or possible noise in the Least Significant bits(LSBs) in the binary form, the LSBs of all pixel values across the image are set to 0, resulting in a modified version of the input image.

The modified image is then flattened and all pixel values(in binary form) are concatenated to be passed through a cryptographic hash function, specifically SHA-512. The output is a 128-character hexadecimal string. Each character of this hexadecimal string is treated as if it were a string character and represented as its 8 bit ASCII value resulting finally in a 1024 bit sequence which will be used as a unique watermark to authenticate content of a particular image.

2) Scrambling RONI Pixels: A list is created using the coordinates corresponding to RONI as extracted by segmenting ROI and RONI in seciton A. The intensity values corresponding to the RONI pixels are retrieved from the original grayscale input image. Then, to enhance security and prevent watermark location predictability, these intensity values are scrambled using a key known only to the watermarking party. Then whatever values land on the first 1024 RONI coordinates in the list are selected for embedding one bit of watermark sequence into each of them.

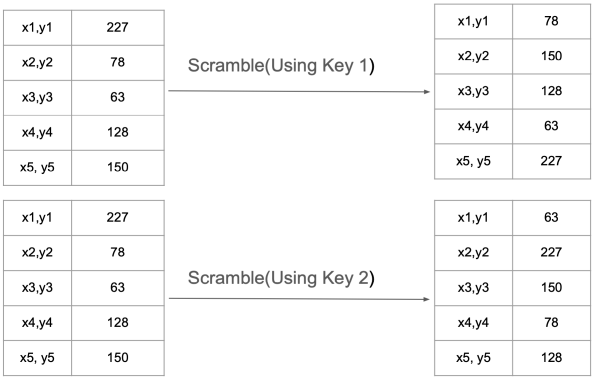


Fig. 5: Scrambling using different keys

The choice of a secure key is critical as it governs the unpredictability of the final arrangement of pixel values after scrambling. As illustrated in Figure 5, even when starting with the same list of coordinates, different keys produce different scrambled outputs. Moreover, the scrambling process is reversible only with access to the original key, which enables unscrambling required for producing the final watermarked image. This key acts as a vital security element: even if an adversary replicates the segmentation process and obtains the same RONI coordinates, they cannot determine which pixels contain the embedded watermark without knowledge of the scrambling key

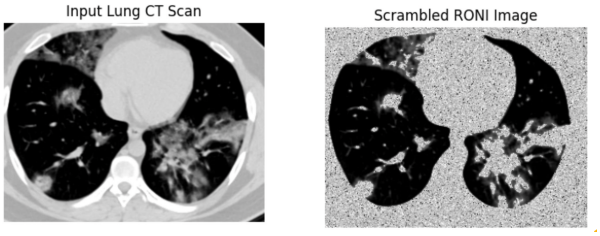


Fig. 6: Scrambling RONI values in original image

Figure 6 illustrates the effect of scrambling the intensity values at the RONI coordinates in the original input image.

3) Watermark Embedding and unscrambling to get final Output: For each of the first 1024 RONI coordinates, the corresponding pixel value is taken from the scrambled version of the original image. One bit of the 1024-bit watermark sequence is then embedded into the LSB of the binary representation of each of these pixel values.

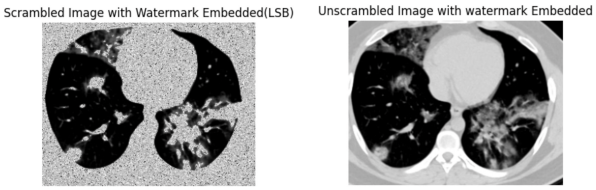


Fig. 7: Unscrambling to get watermarked image

After embedding, the RONI pixel values are unscrambled and restored to their original RONI coordinates, now carrying the watermark bits. This results in the final watermarked image. As shown in Figure 7, the output image is visually indistinguishable from the original input image. The Structural Similarity Index(SSIM) of the watermarked image and input image is 0.99998 showing that the proposed technique successfully embeds 1024 bits into the image with minimal distortion, confined entirely to the RONI region. The algorithm can be adapted to embed additional confidential information, such as patient or hospital data, by increasing the bit length—though significant perceptual changes may occur if the embedded data exceeds five figures bits in length.

C. Extracting Watermark for Content Authentication

Although this paper primarily focuses on the interference of the watermark with computer vision analysis models, it is appropriate to briefly discuss the watermark extraction process, as it highlights the practical use case and additional functionality enabled by embedding such a watermark in the image. This will also show why we won't need the original image for content authentication in this blind fragile watermarking scheme

1) Watermark Extraction: The grayscale medical image containing the embedded watermark is first loaded. The ROI and RONI segments are re-identified using the same segmentation method used during embedding. The pixel values within the RONI are scrambled using the same secret key as before. From this scrambled RONI, the least significant bits (LSBs) of the first 1024 pixel values are extracted to reconstruct the 1024-bit watermark.

2) Recalculating Hash from watermarked image: A modified copy of the watermarked image is created by setting the LSBs of all pixel values to zero. The same cryptographic hash function (e.g., SHA-512) is then applied to this modified image to generate a new 1024-bit binary sequence.

3) Watermark Verification: The extracted watermark is compared with the computed hash. If the two sequences match, the image is considered authentic, indicating that its content has not been altered. A mismatch between them signals potential tampering or unauthorized re-watermarking, thereby compromising the image's integrity. This is because a change to even a single pixel value in the watermarked image would result in a different hash computed at the time of verification.

VII. Methods

To answer the question of how watermarks disrupt medical image analysis AI, we use an empirical approach. This approach utilizes three state-of-the-art computer vision models [19]–[21], a Covid-19 CT scan dataset [22], and three watermarking algorithms [12], [17], [18]. The watermarking algorithms are those previously discussed, a krawtchouk moments based algorithm [17], the algorithm discussed by Guo et al. [12], and the RONI algorithm [14], [18]. The Covid-19 CT scan dataset is sourced from Kaggle [22] and is used for testing the binary classification performance of computer vision models. The computer vision models chosen are state-of-the-art and have shown high performance on many computer vision tasks.

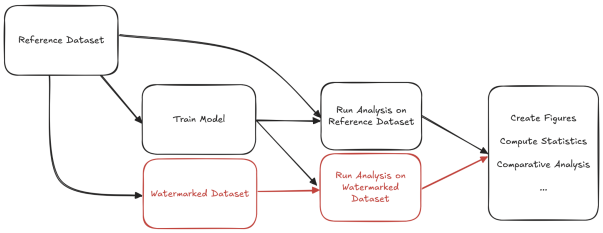


Fig. 8: Method

Figure 8 shows the two phases used in our approach. The steps performed are as follows:

1) The computer vision models are trained using the reference dataset.
2) Analysis is ran using the trained model and the reference dataset.
3) The watermarked datasets are generated.
4) Analysis is ran using the trained model and the watermarked dataset.

Steps 3 and 4 are vital for providing the insights that will help answer our guiding question.

This method differs from that performed in [4] in two major ways. The first is to focus on one data modality, CT scans, while expanding the method to three watermarking algorithms. This method produces the same number of total data points, but gives insight into the affects of different watermarking algorithms. The second key difference is the removal of DenseNet201 and the addition of ResNet50. ResNet50 was chosen as an alternative since its architecture differs significantly from the other chosen models. DenseNet201 derives from the same architecture as DenseNet169 and thus including it offers little additional insight.

The method devised builds on the method used in [4], but extends it in a different, more watermark-focused direction. Furthermore, our method addresses some key limitations of [4].

A. Model Selection

1) MobileNetV2: MobileNetV2 [21] is a lightweight convolutional neural network designed for efficient performance on mobile and IoT devices. It uses an inverted residual structure with thin bottleneck layers and lightweight depthwise convolutions, enabling high accuracy with low computational requirements, making it ideal for real-time and resource-constrained applications. It has only 2.2 million trainable parameters

2) DenseNet169: DenseNet169 [20] is a deep convolutional neural network from the DenseNet family, characterized by dense connectivity where each layer within a dense block receives inputs from all preceding layers. Multiple dense blocks are stacked on top of each other, and the provided input image passes forward. This design improves feature reuse and gradient flow, leading to efficient training and strong performance in image classification and related tasks; DenseNet169 specifically consists of 169 different layers, and has about 20 million trainable parameters.

3) ResNet50: ResNet50 [19] is a 50-layer deep convolutional neural network that introduced residual blocks with skip connections, allowing gradients to flow directly through the network and addressing the vanishing gradient problem. Its architecture includes bottleneck residual blocks, and is the first one to introduce the concept of carrying residuals forward. It was one of the first architectures to achieve strong performance on various IMAGENET tasks, and has been used as benchmark ever since. This architecture has 25 million trainable parameters.

B. Evaluation Metrics

We used a combination of standard classification metrics and image similarity measures. Model performance was evaluated using accuracy, precision, recall, and F1 score per watermarked dataset which gave us a foundational understanding of classification performance. To support these metrics, confusion matrices were constructed based on the counts of true positives, false negatives, true negatives, and false positives.

The confusion matrices help to give a deeper level of insight by showing the specific behavior of the degraded performance seen on the watermarked datasets. Figure 9 shows an example confusion matrix derived from the performance analysis.

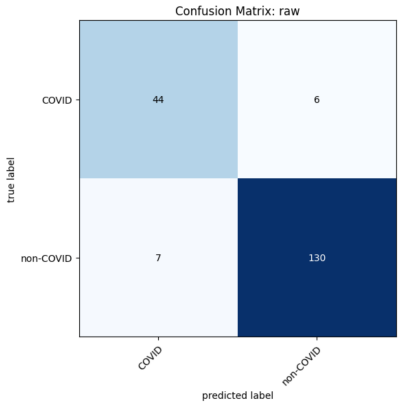


Fig. 9: Example Confusion Matrix

Confusion matrices are a powerful tool for characterizing the nature of model performance and are thus used extensively in the analysis.

C. Technologies Used

This work is meant to serve as a comparative study of the watermarking algorithm implemented in [4]. The deep learning models used are state-of-the-art, have shown high performance on many computer vision tasks, and thus are relevant to our work. These are readily available through PyTorch [23], a popular deep learning library which utilizes the Python programming language. Seamless integration with CUDA [24] allows us to use GPUs to accelerate our deep learning computations. As such, Python, PyTorch, and CUDA has been our choice of language, library, and ecosystem to integrate with.

For the Python environment, a tool called uv [25] is used since it resolves all required libraries on all major platforms (Linux, MacOS, and Windows) with minimal platform-specific configuration required. The uv package manager has support for Jupyter Notebooks which were used for interactive code evaluation. For version control, git and GitHub were used since the tool and platform are standard and well known. The source code with included results for the project is available on GitHub [26].

D. Adversarial Model

In this project, the watermarking algorithms serve as adversarial perturbations to the medical image data. The goal is to evaluate how these watermarking techniques influence the performance of computer vision models, particularly in terms of classification accuracy. Since our aim is to minimize the amount of accuracy loss with the watermarked images, we need to balance the embedding strength of the watermark with the accuracy loss. Our analysis will enable us to determine which watermarking algorithm has the least affect on classification accuracy while maintaining a usable watermark.

VIII. Results and Analysis

Accuracy degradation was found to be in line with the results in [4]. Figure 10 shows the accuracy results of the reference paper. Figure 11 shows the accuracy of our analysis for all watermarks used. The key entries in Figure 11 to look at when comparing to [4] are those labeled "Krawtchouk". Figure 12 shows a comparative analysis in the krawtchouk watermarked image classification accuracy between our results and the results in [4]. The results in [4] show a steeper degradation in accuracy due to the worst-case selection process used. Importantly, our results show a similar, yet less steep, downward trend telling us that we are experiencing the same phenomenon.

CT-Scans – DenseNet169 – Original Accuracy = 95.8%												
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	99.2	0.4, 0.2	89.5	99.0	0.2, 0.3	87.9	98.4	0.4, 0.8	82.9	97.7	0.3, 0.3	77.0
200	99.1	0.7, 0.2	89.5	98.7	0.3, 0.3	87.5	97.4	0.3, 0.1	79.5	95.9	0.3, 0.3	73.7
300	99.0	0.7, 0.2	89.5	98.2	0.3, 0.1	86.6	96.2	0.4, 0.1	79.1	94.2	0.4, 0.3	70.0
400	98.8	0.7, 0.2	90.0	97.8	0.3, 0.1	87.0	95.1	0.4, 0.2	78.7	92.6	0.4, 0.3	70.0
500	98.7	0.5, 0.8	90.0	97.3	0.4, 0.1	86.2	94.0	0.4, 0.1	80.0	91.0	0.4, 0.4	69.1
600	98.5	0.7, 0.2	89.5	96.8	0.3, 0.1	85.4	92.9	0.1, 0.4	75.8	89.4	0.4, 0.5	67.0
700	98.3	0.9, 0.9	89.5	96.4	0.1, 0.4	86.2	91.8	0.1, 0.4	76.6	87.8	0.4, 0.5	65.8
800	98.2	0.1, 0.3	89.1	95.9	0.1, 0.3	86.6	90.7	0.1, 0.4	75.8	86.3	0.4, 0.5	67.0
900	98.0	0.9, 0.9	89.5	95.4	0.1, 0.5	84.5	89.5	0.1, 0.4	74.1	84.6	0.1, 0.5	64.5
1000	97.8	0.9, 0.9	89.1	94.9	0.1, 0.3	85.0	88.3	0.1, 0.5	72.0	83.0	0.1, 0.5	65.0

Fig. 10: Reference Results Using Krawtchouk Moments Watermarking and 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	0.955882	0.948905	0.952381
1	Krawtchouk	50	(0.5, 0.5)	1024	0.988022	42.098871	0.925134	0.955556	0.941606	0.948529
2	Krawtchouk	100	(0.5, 0.5)	1024	0.963324	36.593767	0.898396	0.933824	0.927007	0.930403
3	Krawtchouk	200	(0.5, 0.5)	1024	0.910654	31.217936	0.866310	0.894366	0.927007	0.910394
4	Krawtchouk	300	(0.5, 0.5)	1024	0.868889	28.401042	0.834225	0.835443	0.963504	0.894915
5	Guo-Zhuang	N/A	N/A	75	0.998816	52.394887	0.930481	0.955882	0.948905	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	0.955882	0.948905	0.952381

Fig. 11: Our Results With All Watermarking Algorithms and DenseNet169

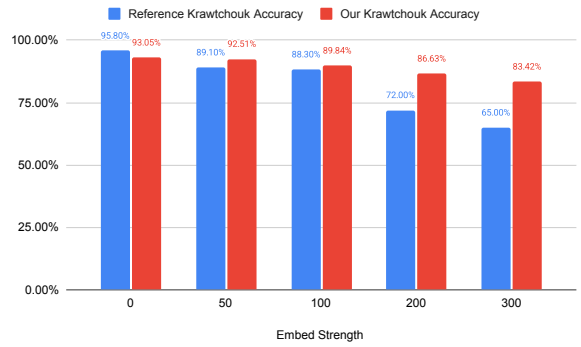


Fig. 12: Accuracy Comparison Using Krawtchouk Moments Watermarking and DenseNet169

In addition to a degradation in the classification accuracy, we can see that this degradation is proportional to the image similarity, as reported by the SSIM. Figure 13 shows the

relationship between SSIM and classification accuracy for all models and all watermarked datasets. Figure 14 shows the accuracy comparison between the watermarked dataset and all models.

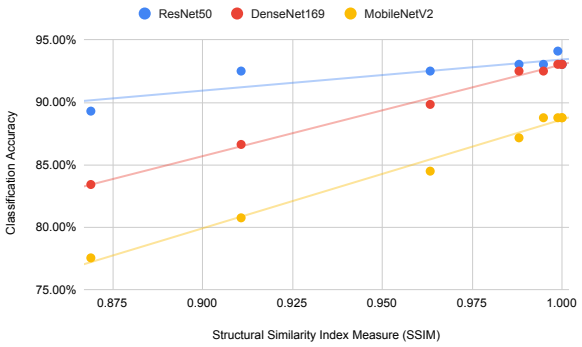


Fig. 13: SSIM and Classification Accuracy With All Models

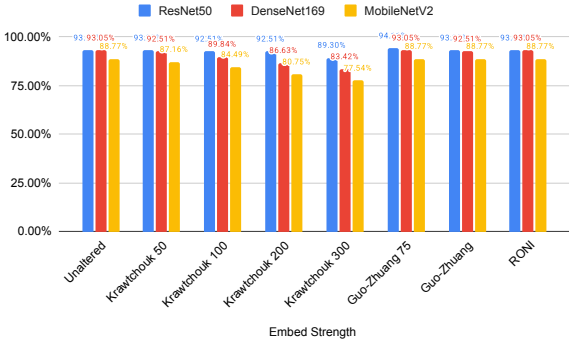
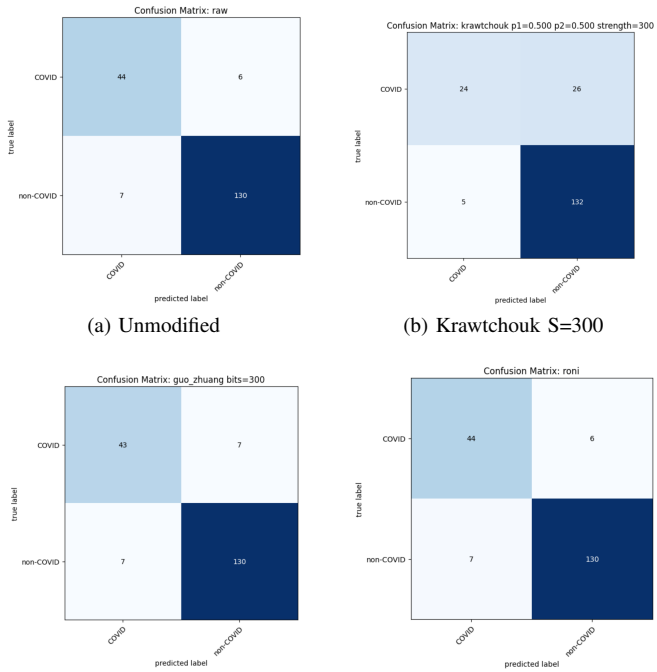


Fig. 14: Accuracy Comparison Between Models

A degradation in the accuracy as SSIM decreases is not a surprising result. It is well known in the machine learning community that as image similarity drops, accuracy is likely to drop as well. Where our results are interesting is in the behavior surrounding the confusion matrices. Figure 15 shows the confusion matrices in the worst case for each watermark type while using DenseNet169.

With DenseNet169, we find generally that the decrease in classification accuracy is due to an increase in the false negative rate. This is most apparent with the krawtchouk moments based watermarking technique where only 48% of positive Covid-19 images are correctly classified. This trend is the same for the other computer vision models, ResNet50, and MobileNetV2. The exact statistics can be found in the GitHub repository [26] used for this project.

This result is surprising and was not mentioned in [4] even though it has potentially substantial implications. An increase in the false negative rate shows that the computer vision models do not fail in a random or uniform manner under watermarking, but instead exhibit a specific bias towards detecting false negatives of the target condition. In the context of Covid-19 detection, this means that many infected



(a) Unmodified (b) Krawtchouk S=300

(c) Guo-Zhuang L=300 (d) RONI

Fig. 15: Confusion Matrices For DenseNet169

cases could be systematically missed, which is particularly concerning for clinical deployment where false negatives can lead to delayed treatment and further spread of infection. These findings highlight the need for watermarking methods to be evaluated not just in terms of visual imperceptibility and robustness, but also in terms of their downstream effects on model decision boundaries and diagnostic reliability.

Alternative watermarking methods such as [12] and [14] exhibit promising results as these methods can be used to embed a similar number of bits as krawtchouk watermarking with a far lower hit to the classification accuracy. Most of the increase in performance seems to be driven by an increase in the SSIM that these algorithms provide. However, these algorithms do not come with the same robustness guarantees as krawtchouk watermarking, so they must be used in applicable contexts. Furthermore, the computer vision model chosen has a significant impact. ResNet50 had the best overall performance, likely driven by it having the most trainable parameters out of all models tested. However, to avoid unnecessarily extending the length of the paper, we have not included the detailed figures and performance analysis for ResNet50 and MobileNetV2 here. Interested readers can refer to the GitHub repository [26]for the corresponding contemporary graphs and detailed results.

IX. Conclusion and Future Work

With the rise of AI systems capable of generating synthetic media, the integrity of digital content has become a critical concern. Efforts like C2PA aim to use watermarking to embed provenance and authorship data directly into media to counter

misinformation. Reliable medical image analysis systems have the potential to improve diagnostic workflows when integrated into routine clinical practice. We explored the impact of various watermarking algorithms on medical images analysis, focusing on how they influence the performance of deep learning–based computer vision classification tasks.

Our comparative analysis suggests that fragile watermarking does not trigger AI systems to significantly misclassify the diagnosis while robust watermarks do. This degradation of performance is seen primarily through an increase in false negative rate during classification. From a patient and healthcare provider perspective, this is troubling as it could lead to more severe symptoms, delayed treatment, and an increased risk of disease spread. Given that this is the case, watermarking effects on downstream AI medical analysis systems are another evaluation metric that should be considered, similar to security and image similarity.

Further understanding of the effects of watermarking on medical image analysis is required. This study can be extended in several directions to further understand the impact of watermarking on medical image analysis. Future research could involve evaluating a broader range of computer vision models, including more recent transformer-based architectures, to assess their robustness to watermarking interference. Additional watermarking algorithms, particularly those aligned with emerging standards like C2PA, could be incorporated to compare their influence on model performance and forensic traceability. Exploring the use of diverse watermarks, specifically those embedded in the RONI (Region of Non-Interest), may offer additional features not explored in our analysis. Furthermore, expanding to other medical imaging modalities, such as MRIs and X-rays would help to evaluate the understanding of the cross-modality impact of watermarking.

In combination, all of this shows that we do not yet have a complete picture. Our findings indicate the behavior of one aspect of digital watermarking, but fall short in other places. As watermarking becomes more prevalent in everyday life, potentially through initiatives such as C2PA, its broader implications must be understood. Future research into the effects of watermarking on AI tasks will be required.

REFERENCES

[1] J. Mangalagiri, D. Chapman, A. Gangopadhyay, Y. Yesha, J. Galita, S. Menon, Y. Yesha, B. Saboury, M. Morris, and P. Nguyen, "Toward generating synthetic CT volumes using a 3d-conditional generative adversarial network," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 858–862. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9458065

[2] C2pa_specification. [Online]. Available: https://c2pa.org/specifications/specifications/1.0/specs/_attachments/C2PA_Specification.pdf

[3] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," vol. 23, no. 5, pp. 828–841. [Online]. Available: http://arxiv.org/abs/1710.08864

[4] K. D. Apostolidis and G. A. Papakostas, "Digital watermarking as an adversarial attack on medical image analysis with deep learning," vol. 8, no. 6, p. 155.

[5] X. Zhou, H. Huang, and S. Lou, "Authenticity and integrity of digital mammography images," vol. 20, no. 8, pp. 784–791, conference Name: IEEE Transactions on Medical Imaging. [Online]. Available: https://ieeexplore.ieee.org/document/938246

[6] X. Wang and H. Yu, "How to break MD5 and other hash functions," in *Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, Ed. Springer Berlin Heidelberg, vol. 3494, pp. 19–35, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/11426639_2

[7] Jian Ren and Tongtong Li, "A cryptographically secure image watermarking scheme," in *MILCOM 2005 - 2005 IEEE Military Communications Conference*. IEEE, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/1605812/

[8] L.-Q. Kuang, Y. Zhang, and X. Han, "A medical image authentication system based on reversible digital watermarking," in *2009 First International Conference on Information Science and Engineering*, pp. 1047–1050, ISSN: 2160-1291. [Online]. Available: https://ieeexplore.ieee.org/document/5455333

[9] S. M. Mousavi, A. Naghsh, and S. A. R. Abu-Bakar, "Watermarking techniques used in medical images: a survey," vol. 27, no. 6, pp. 714–729. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4391065/

[10] Z. Jalil and A. M. Mirza, "A review of digital watermarking techniques for text documents," in *2009 International Conference on Information and Multimedia Technology*, pp. 230–234. [Online]. Available: https://ieeexplore.ieee.org/document/5381212/?arnumber=5381212

[11] M. Ali, C. W. Ahn, M. Pant, S. Kumar, M. K. Singh, and D. Saini, "An optimized digital watermarking scheme based on invariant DC coefficients in spatial domain," vol. 9, no. 9, p. 1428, number: 9 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/9/9/1428

[12] X. Guo and T.-g. Zhuang, "A region-based lossless watermarking scheme for enhancing security of medical data," vol. 22, no. 1, pp. 53–64. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3043669/

[13] F. Rahimi and H. Rabbani, "A dual adaptive watermarking scheme in contourlet domain for DICOM images," vol. 10, no. 1, p. 53. [Online]. Available: https://doi.org/10.1186/1475-925X-10-53

[14] N. A. Memon, S. Gilani, and A. Ali, "Watermarking of chest CT scan medical images for content authentication," in *2009 International Conference on Information and Communication Technologies*, pp. 175–180. [Online]. Available: https://ieeexplore.ieee.org/document/5268167

[15] SARS-COV-2 ct-scan dataset. [Online]. Available: https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset

[16] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by krawtchouk moments," vol. 12, pp. 1367–77.

[17] G. Papakostas, E. Tsougenis, and D. Koulouriotis, "Moment-based local image watermarking via genetic optimization," vol. 227, pp. 222–236. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0096300313012046

[18] N. A. Memon and S. Gilani, "NROI Watermarking of Medical Images for Content Authentication," 2006, unpublished work or internal report.

[19] resnet50 — torchvision main documentation. [Online]. Available: https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html

[20] densenet169 — torchvision main documentation. [Online]. Available: https://docs.pytorch.org/vision/main/models/generated/torchvision.models.densenet169.html

[21] MobileNet v2 – PyTorch. [Online]. Available: https://pytorch.org/hub/pytorch_vision_mobilenet_v2/

[22] Covid-19 binary classification | DenseNet169 | 98%. [Online]. Available: https://kaggle.com/code/ahmedtronic/covid-19-binary-classification-densenet169-98

[23] PyTorch. [Online]. Available: https://pytorch.org/

[24] CUDA toolkit - free tools and training. [Online]. Available: https://developer.nvidia.com/cuda-toolkit

[25] "astral-sh/uv," original-date: 2023-10-02T20:24:11Z. [Online]. Available: https://github.com/astral-sh/uv

[26] A. S. Sean Moulton, Omkar Kulkarni, "Cmsc 652 group project," https://github.com/WindowsVista42/CMSC-652-Group-Project, 2025, [Online; accessed: Mar. 9, 2025].