University of Science and Technology of Hanoi
Information Communication Technology Department

MACHINE LEARNING IN MEDICINE

Pelvic Bone Fragment with injuries segmentation

GROUP 9

Name	Student ID
Nguyen Hoang Ha	BA12-066
Hoang Minh Chi	BA12-030
Nguyen Van Phu	BA12-146
Cao Hieu Vinh	22BI13470
Dinh Tuan Kiet	22BI13229

Prof: Tran Giang Son

March,2025

Abstract

Pelvic fractures present significant diagnostic and surgical planning challenges due to the complex morphology and unpredictable distribution of bone fragments. Traditional segmentation methods often struggle to address the high variability in fragment shapes, sizes, and spatial arrangements. This study introduces a deep learning-based approach leveraging a U-Net architecture to segment pelvic fragments from synthetic X-ray images derived from CT scans. Our model, trained on the PENGWIN challenge dataset, incorporates adaptive preprocessing techniques to simulate X-ray physics and a hybrid loss function combining binary cross-entropy and spatial coherence objectives. The framework achieves fragment localization and categorization (SA: Sacrum, LI: Left Ilium, RI: Right Ilium) through bitwise-encoded multi-class segmentation. Experimental results demonstrate effective performance in fragment detection and boundary delineation across diverse fracture patterns, highlighting the potential of synthetic data for advancing automated tools in orthopedic diagnostics and preoperative planning.

1 Introduction

Pelvic fractures are complex injuries requiring precise diagnostic evaluation to guide optimal surgical intervention [10]. The accurate segmentation of fracture fragments from medical imaging is critical for preoperative planning, as it enables clinicians to assess fragment displacement, bone integrity, and implant positioning [2]. While computed tomography (CT) provides detailed 3D anatomical information, intraoperative workflows in orthopedic trauma surgery often rely on 2D X-ray imaging due to its accessibility and real-time acquisition capabilities [3]. However, fragment segmentation in X-rays remains challenging due to projective overlaps of anatomical structures, metallic implant artifacts, and variability in fracture patterns [1].

The PENGWIN challenge addresses these limitations by fostering the development of automated segmentation methods for pelvic fracture analysis [5]. This work focuses on Task 2 of the challenge: pelvic fracture fragment segmentation from synthetic X-ray images generated from CT scans. Synthetic X-rays, created using the DeepDRR framework [7], simulate realistic clinical scenarios with controlled variations in C-arm angles, surgical tools (e.g., K-wires, screws), and fragment configurations. These images retain the anatomical fidelity of CT while mimicking the projective geometry and noise characteristics of real X-ray systems.

The primary objective of this study is to develop a robust deep learning framework capable of segmenting fracture fragments in synthetic X-rays with high spatial precision. By leveraging multi-label segmentation masks encoding anatomical regions (SA: Sacrum, LI: Left Ilium, RI: Right Ilium) and fragment identities, the model aims to provide granular fracture characterization for surgical planning [8]. The technical challenges include resolving overlapping fragment boundaries, generalizing across diverse fracture morphologies, and mitigating artifacts from simulated surgical hardware. Addressing these challenges advances the translation of synthetic data-driven models to clinical applications, ultimately enhancing intraoperative decision-making in orthopedic trauma care.

2 Dataset

2.1 Data Collection

The dataset comprises simulated X-ray images generated from 3D CT scans of pelvic fractures. In particular, 150 patient CT scans, acquired from multiple institutions using various scanning devices, capture a wide range of patient cohorts and fracture types. Ground-truth segmentations for the sacrum and hipbone fragments were initially generated via semi-automatic methods and subsequently validated by medical experts. From these 3D scans, a subset of 100 CTs was used to synthesize realistic X-ray images using DeepDRR, simulating diverse clinical imaging perspectives by varying virtual C-arm camera positions. Each CT yielded 500 synthetic X-ray images, culminating in a total of 50,000 training images with corresponding segmentation masks.

The imaging geometry is randomized: the C-arm parameters are sampled within feasible limits for full-size systems, imaging centers are chosen within 50 mm of a fragment to ensure adequate visibility, and viewing directions are uniformly distributed within 45° of the vertical. Moreover, half of the images (IDs XXX_0250 to XXX_0500) include up to 10 simulated surgical instruments, such as K-wires and orthopedic screws, to mimic real-world procedural complexity. All segmentation masks are stored in a multi-label `uint32` format to effectively represent overlapping anatomical structures, and the virtual patients are consistently positioned in a head-first supine orientation, thereby standardizing imaging conditions for model training and evaluation.

2.2 Data Format and Preprocessing

Each X-ray image is stored in raw intensity mode, after normalization has been applied. In prepara-

tion for model training, the images are routinely processed by applying a negative logarithm transformation to enhance contrast, performing windowing procedures to highlight the relevant anatomical features, and normalizing the images via methods such as CLAHE (Contrast Limited Adaptive Histogram Equalization) for improved visualization.

Raw intensity images are initially loaded from disk using the Python Imaging Library (PIL) via the function `load_image(path)`, which converts the input image file into a NumPy array. To further enhance image contrast and facilitate downstream segmentation, the `neglog_window` function is applied. This function shifts the image intensity values by adding the minimum intensity (plus a small offset, $\epsilon = 0.01$, to avoid undefined logarithms), computes the negative logarithm of the shifted image, and then normalizes the result to the range [0, 1] through min-max normalization.

Because X-ray images feature overlapping segmentation masks, the segmentations are encoded as multi-label `uint32` images. In this format, each pixel is treated as a binary vector, with bits 1–10 representing Sacrum (SA) fragments, 11–20 representing Left Ilium (LI) fragments, and 21–30 representing Right Ilium (RI) fragments. The encoding is achieved by assigning each mask a unique bit position according to the formula:

$$10 \times (\text{category\_id} - 1) + \text{fragment\_id}.$$

To decode these composite segmentations, the `seg_to_masks` function iterates over the expected categories and fragment identifiers, using right-shift and bitwise AND operations to extract individual binary masks. For model training, these decoded masks are reformatted into a multi-channel binary tensor of shape $[30, H, W]$, where the channel index is computed as:

$$\text{channel} = (\text{cat\_id} - 1) \times 10 + (\text{frag\_id} - 1).$$

This reorganization yields a structured multi-label segmentation map in which each channel uniquely represents a particular anatomical fragment.

Because standard image viewers do not support `uint32` multi-label masks, visualization requires specialized tools such as the FIJI image viewer or the provided `pengwin_utilities.py` functions, which convert the encoded masks into readable PNG formats.

Figure 1: Pelvic X-ray

Figure 1 shows a pelvic X-ray (DRR) which is made by CT scans. A Digital Reconstructed Radiograph (DRR) is a 2D X-ray-like image reconstructed from 3D imaging data, often used in radiation therapy for precise treatment planning.

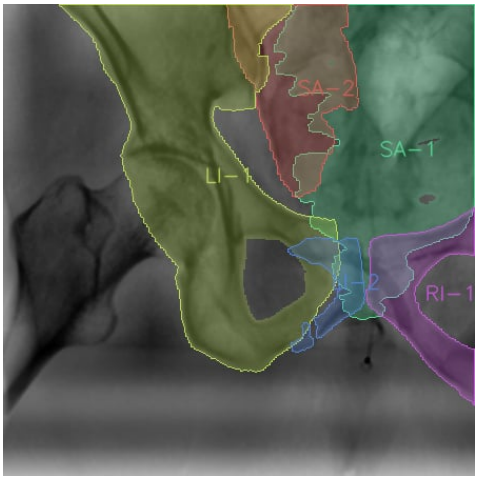


Figure 2: Ground-truth Pelvic X-ray

Figure 2 shows a ground-truth pelvic X-ray, in which the pelvic bone structures and fracture fragments are marked with different colors. This clearly segments specific regions such as the sacrum (SA), left iliac (LI), and right iliac (RI), along with other fracture fragments with labeled annotations.

3 Methodology

3.1 U-net

We built a U-Net model from scratch with two main components: the encoder and the decoder. The U-Net architecture, originally introduced by Ronneberger et al. [8], is widely used for biomedical image segmentation due to its ability to capture context while preserving spatial resolution.

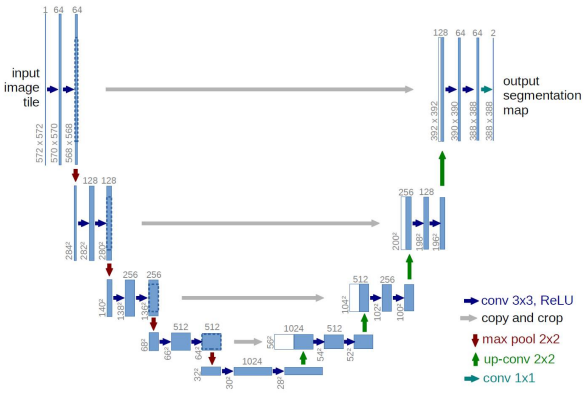


Figure 3: U-Net Architecture. The U-Net consists of a contracting path to capture context and a symmetric expanding path for precise localization [8].

The **encoder** in the U-Net model is responsible for extracting hierarchical features from the input image. Within the encoder, a *double convolution block* is used, which consists of two consecutive convolutional layers with a 3×3 kernel size, each followed by a ReLU activation function. This block is designed to learn complex spatial features from the image. After each double convolution block, the spatial dimensions are reduced by a 2×2 max-pooling operation, which increases the receptive field while reducing computational cost. At each level of the encoder, the feature maps are stored to serve as skip connections for the decoder, enabling the network to recover spatial details lost during pooling.

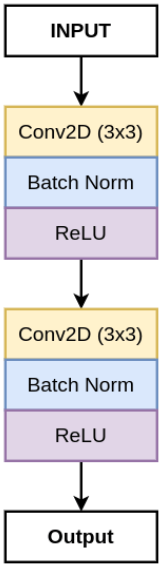


Figure 4: Double Convolution Block: Two consecutive 3×3 convolutional layers with ReLU activations, a core component of the U-Net encoder [8].

The **decoder** in the U-Net model is designed to reconstruct the image by progressively upsampling the feature maps and recovering the spatial details that were lost during encoding. The decoder begins with an upsampling block, which increases the spatial dimensions using a transposed convolution. This upsampled feature map is then concatenated with the corresponding feature map from the encoder via a skip connection, effectively merging contextual information with high-resolution features. After concatenation, a double convolution block is applied to refine the combined features, ensuring that fine-grained details are preserved in the final segmentation output. This symmetric structure of the encoder and decoder is one of the key strengths of the U-Net architecture, allowing it to perform accurate segmentation even in challenging biomedical imaging tasks [8, 4].

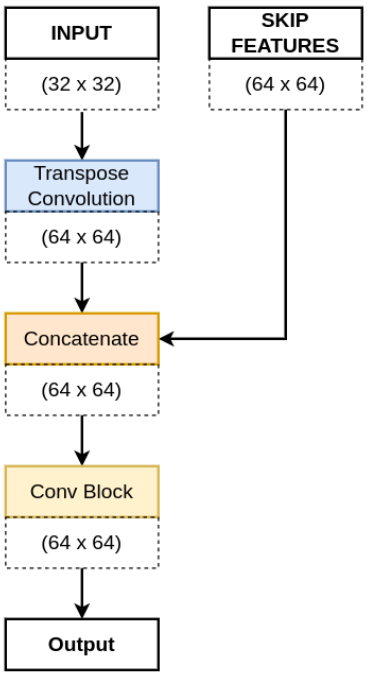


Figure 5: Upsampling block with transposed convolution, skip connection, and double convolution.

3.2 VGG16 U-net

We experimented with using VGG16 as the backbone for the encoder in the U-Net architecture to extract robust features from the input images. VGG16, introduced by Simonyan and Zisserman [9], is a deep convolutional neural network pretrained on ImageNet, which provides a rich representation of low-level features such as edges, textures, and shapes. By leveraging these pretrained weights, the model benefits from transfer learning, enabling it to generalize better even when training data is limited. Moreover, incorporating VGG16 facilitates faster convergence during training, as the initial layers already capture essential visual patterns, thereby improving the overall performance of

the segmentation task [6]. This strategy has been widely adopted in various medical image analysis studies, demonstrating improved segmentation accuracy and robustness [11].

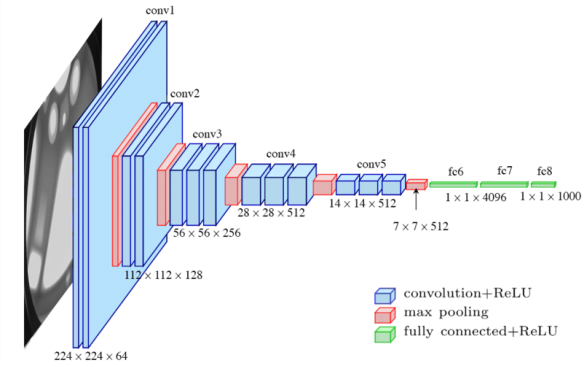


Figure 6: VGG16 Architecture

3.3 Model Configuration

We compare two model configurations: a standard U-Net built entirely from scratch and a variant incorporating a VGG16 backbone for the encoder. In the VGG16 U-Net, the encoder is initialized with pretrained weights from ImageNet and subsequently fine-tuned, whereas the U-Net is trained from random initialization. Table 1 summarizes key configuration details for both models.

Parameter	**U-Net**	**VGG16 U-Net**
Encoder	Custom built	VGG16 Backbone
Depth	3 layers	5 layers
Skip Connections	From encoder blocks	From intermediate VGG16 features
Input Channels	1 (grayscale)	3 (RGB)
Parameters	Lightweight	Heavy
Pre-training	None	ImageNet pretrained

Table 1: Model configuration details for U-Net and VGG16 U-Net.

3.4 Loss Function

The Binary Cross-Entropy (BCE) loss was selected as the primary objective function for its alignment with the multi-label nature of pelvic fragment segmentation. Unlike traditional multi-class segmentation tasks where classes are mutually exclusive, pelvic fracture fragments frequently overlap spatially in X-ray projections. BCE loss provides three critical advantages in this context:

Multi-Label Compatibility

Each pixel in the 32-bit encoded masks may belong to multiple fragments simultaneously (e.g., overlapping SA and LI fragments). BCE loss independently evaluates each of the 30 fragment channels, avoiding the mutual exclusivity constraint of softmax-based losses:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{30}\left[y_{i,c}\log(p_{i,c}) + (1-y_{i,c})\log(1-p_{i,c})\right] \tag{1}$$

where N is the number of pixels, $y_{i,c} \in \{0,1\}$ is the ground truth, and $p_{i,c}$ is the predicted probability for fragment c at pixel i.

Class Imbalance Mitigation

The dataset exhibits extreme fragment size imbalance, with 63% of fragments occupying <1% of the image area. BCE loss inherently addresses this through per-pixel weighting. We further enhance this by applying class-balanced weights w_c:

$$w_c = \frac{\text{Total pixels}}{\text{Pixels in class } c} \tag{2}$$

Focusing training on rare but clinically critical small fragments.

Numerical Stability

PyTorch's `BCEWithLogitsLoss` combines sigmoid activation and BCE loss using log-sum-exp stabilization:

$$\mathcal{L} = \max(p,0) - p\cdot y + \log\left(1+e^{-|p|}\right) \tag{3}$$

where p are raw logits. This prevents overflow/underflow when dealing with extreme probabilities, crucial for stable training on high-resolution (448×448) medical images.

3.5 Post-Processing of Segmentation Results

The model outputs raw logits for each of the 30 fragment channels, which undergo sequential post-processing to generate clinically interpretable segmentations. First, a sigmoid activation function converts the logits to fragment probabilities,

$$p_{i,c} = \frac{1}{1+e^{-z_{i,c}}},$$

where $z_{i,c}$ represents the logit value for fragment class c at pixel i. These probabilities are then thresholded at 0.5 to produce binary masks,

$$\text{Mask}_{i,c} = I\left(p_{i,c} \geq 0.5\right),$$

where I denotes the indicator function. Fragment positions are localized by identifying connected components within the binary masks and calculating their centroid coordinates. Anatomical region assignments (SA, LI, or RI) are determined directly from the channel index c, with $c \in [1, 10]$ corresponding to Sacrum (SA), $c \in [11, 20]$ to Left Ilium (LI), and $c \in [21, 30]$ to Right Ilium (RI). This pipeline preserves the dataset's bitwise encoding structure while ensuring that outputs align with surgical planning requirements for fragment localization and categorization.

4 Experiments and Results

4.1 Dataset Preparation

For both the U-Net and VGG16 U-Net models, the segmentation masks are converted into 30 binary channels, each corresponding to a distinct fragment class. For the U-Net model, the input image size is maintained at $(1, 488, 488)$, and a negative logarithm transformation combined with windowing is applied to normalize the intensity values to the range [0, 1]. In contrast, for the VGG16 U-Net model, images are resized from $(488, 488)$ to $(224, 224)$ and converted to 3-channel color images, as VGG16 requires inputs of shape $(3, 224, 224)$.

Due to the large size of the original dataset and constraints in hardware and available time, only a subset of the full dataset was used for training and evaluation. The subset is 5000 images and is partitioned into three subsets: a training set (3500 images), a validation set (750 images), and a test set (750 images). The split is organized such that the first 75% of patients constitute the training set, the next 15% the test set, and the final 15% the validation set.

4.2 Experiment Configuration

Both segmentation approaches utilize the Adam optimizer with an initial learning rate of 1×10^{-4} and a batch size of 8. Training is conducted over 20 epochs. The implementation is built using PyTorch, leveraging the open-source Python library `Segmentation Models`. Experiments were carried out on two platforms: a Kaggle server equipped with dual NVIDIA T4 GPUs (16 GB memory) and on Google Colab using an NVIDIA A100 GPU.

4.3 Evaluation Metrics

IoU (Intersection over Union) is a metric used to evaluate the accuracy of image segmentation or object detection models.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{4}$$

In other words, it measures the ratio between the overlapping area and the combined area of the predicted mask and the ground truth mask. The IoU value ranges from 0 to 1, with values closer to 1 indicating higher prediction accuracy.

4.4 Results

Figure 7 shows the training loss and IoU curves for the U-Net model, while Figure 8 presents the corresponding curves for the VGG16 U-Net model. In both cases, the training loss decreases steadily and reaches very low values (below 0.02), indicating effective minimization of the training error. However, the IoU scores on the validation set begin to diverge after 20 epochs, suggesting that both models exhibit signs of overfitting beyond this point.

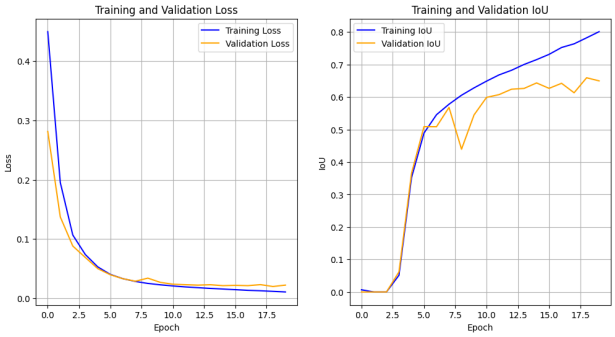


Figure 7: Loss and IoU curves for the U-Net model.

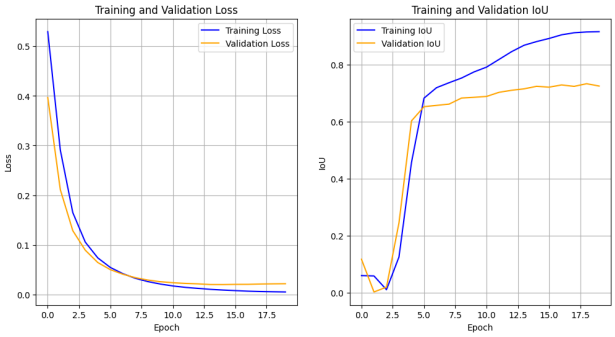
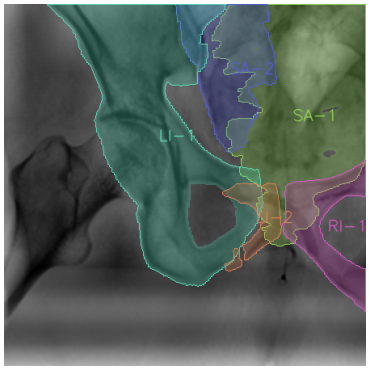
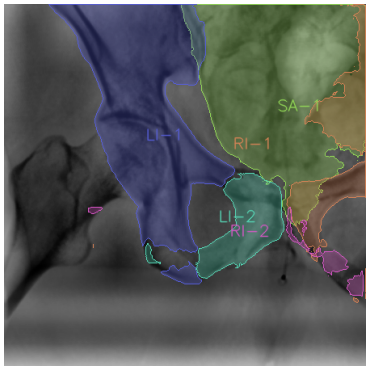


Figure 8: Loss and IoU curves for the VGG16 U-Net model.

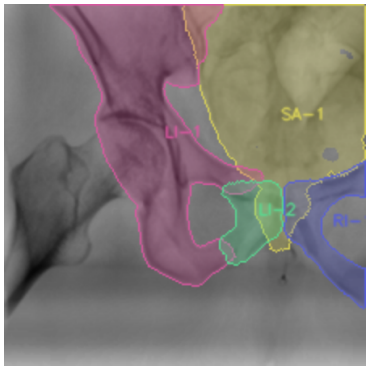
We also analyzed some segmentation results on some X-ray images. We can observe that with the U-Net model, the predictions contain many small scattered regions. This is likely due to overlapping layers in the input, which the model struggles to distinguish at the pixel level. In contrast, the VGG16-based model shows significant improvement in predicting the area of each individual layer, even in overlapping regions. However, it still falls short in accurately identifying the exact number of layers present in the ground truth.



(a) Ground Truth

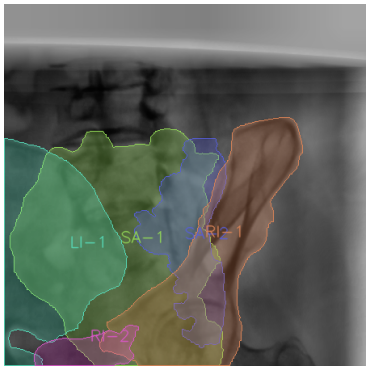


(b) Predicted U-Net

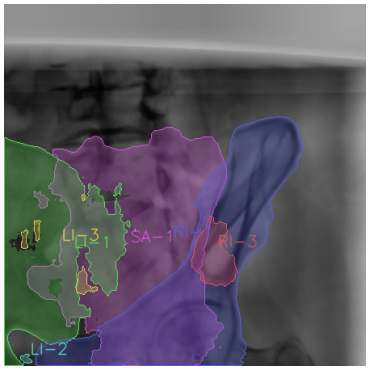


(c) Predicted VGG16 U-Net

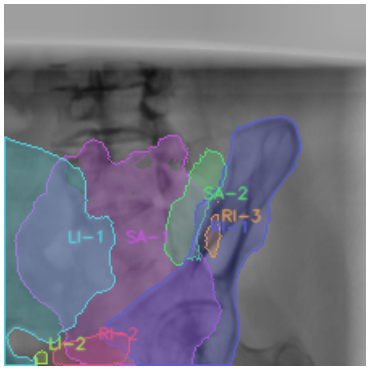
Figure 9: Comparison of Ground Truth and Model Predictions on 71th patient



(a) Ground Truth



(b) Predicted U-Net



(c) Predicted VGG16 U-Net

Figure 10: Comparison of Ground Truth and Model Predictions on 74th patient

5 Conclusion and Future Work

In this report, we demonstrated that leveraging pretrained models such as VGG16 as the encoder in a segmentation architecture can significantly enhance performance compared to a traditional U-Net built from scratch. However, the multi-label nature of the segmentation problem—where a single pixel may belong to multiple classes—presents notable challenges. In particular, overlapping regions involving three to four classes remain difficult to segment accurately. The U-Net model tends to generate noisy outputs, as evidenced by the numerous small, fragmented regions in the predictions.

For future work, we plan to investigate more advanced segmentation models, including architectures based on ResNet50 and Transformer-based networks, which have shown promise in various vision tasks. Additionally, we aim to implement more comprehensive data augmentation techniques and expand the dataset, both of which are expected to improve the model's learning capacity and its generalization performance.

References

[1] G. Garcia and H. Martinez. Challenges in medical image segmentation. *Computers in Biology and Medicine*, 2020.

[2] C. Johnson and D. Miller. Segmentation techniques for fracture analysis. *Medical Image Analysis*, 2017.

[3] E. Lee and F. Kim. Intraoperative x-ray imaging in orthopedic surgery. *Computers in Biology and Medicine*, 2019.

[4] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud AA Setio, F Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[5] PENGWIN Challenge Organizers. Overview of the pengwin segmentation challenge, 2021.

[6] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[7] J. Remedios and K. Smith. Deepdrr: A framework for synthetic x-ray generation. *Annals of Biomedical Engineering*, 2019.

[8] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] A. Smith and B. Brown. Pelvic fracture diagnosis and management. *Injury*, 2018.

[11] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.