

WiDS 5.0 Final Capstone Report

From Pixels to Prognosis: A Journey Through Healthcare Computer Vision

Submitted by: Kushagra Raghuvanshi

Mentorship Program: WiDS 5.0

Track: Healthcare AI & Computer Vision

Date: January 28, 2026

Abstract

Artificial Intelligence, particularly Computer Vision (CV), has rapidly evolved from a theoretical research field into a critical utility for modern healthcare. This report documents a four-week intensive mentorship program under WiDS 5.0, exploring the intersection of digital image processing and medical diagnostics.

The project began with an analysis of the fundamental gap between human and computer vision, utilizing the teachings of Joseph Redmon to understand image representation. It progressed to traditional image processing techniques, where raw X-ray data was manipulated using mathematical kernels for edge detection and noise reduction. The latter half of the program transitioned to Deep Learning, specifically implementing Convolutional Neural Networks (CNNs). We trained a YOLOv8 classification model on a dermatological dataset to detect skin cancer with 97.7% accuracy and concluded with a case study on the TIGER Grand Challenge, designing a pipeline for automating the scoring of Tumor-Infiltrating Lymphocytes (TILs) in breast cancer pathology.

This report details the methodologies, code implementations, theoretical underpinnings, and experimental results of this journey.

Chapter 1: The Science of Vision

1.1 Introduction

Before training machines to diagnose cancer, one must understand how a machine "sees." Unlike humans, who perceive images through biological context and evolutionary pattern matching, computers perceive images as multi-dimensional arrays of numerical values.

During Week 1, we focused on "The Ancient Secrets of Computer Vision" as taught by Joseph Redmon, the creator of the YOLO architecture. Redmon's lectures highlighted the extreme difficulty of computer vision: the "Semantic Gap." This is the disconnect between the low-level pixel data (numbers from 0 to 255) and the high-level semantic concept (e.g., "this is a cat").

1.2 Human vs. Computer Vision

A core learning from this module was the concept of **Invariance**. Human vision is naturally invariant to changes in lighting, rotation, and occlusion. A chair viewed from the top-down is still recognized as a chair by a human. To a computer, however, the pixel grid changes entirely.

- **Illumination Invariance:** We learned that simple changes in brightness can alter the entire numerical matrix of an image, potentially confusing standard algorithms.
- **Viewpoint Invariance:** We explored how data augmentation (rotating and flipping images during training) is essential to teach computers this concept.

1.3 Medical Documentation Research

In parallel with vision theory, we researched the structure of medical reports, which serve as the "Ground Truth" for training AI.

- **Radiology Reports:** These are highly structured, often containing a "Findings" section and an "Impression" section. We learned that Natural Language Processing (NLP) is often required to extract labels from these reports to create training datasets for vision models.
- **Pathology Reports:** These describe microscopic findings. Unlike radiology, which deals with macroscopic anatomy, pathology deals with cellular structures, requiring much higher resolution imaging (Whole Slide Images).

Chapter 2: Traditional Image Processing Algorithms

Week 2 moved from theory to practice. Using Python and the OpenCV library, we manipulated medical X-rays at the pixel level.

2.1 Histogram Analysis

Every digital image has a "fingerprint" called a histogram—a graph showing the frequency of every pixel intensity value.

- **Methodology:** We loaded a Chest X-Ray and converted it to Grayscale. In an 8-bit grayscale image, 0 represents absolute black, and 255 represents pure white.
- **Observation:** The histogram of the X-ray was bimodal. The lower intensity peak represented the air-filled lungs (which absorb less radiation and appear dark), while the higher intensity peak represented bone and dense tissue (which absorb radiation and appear white).

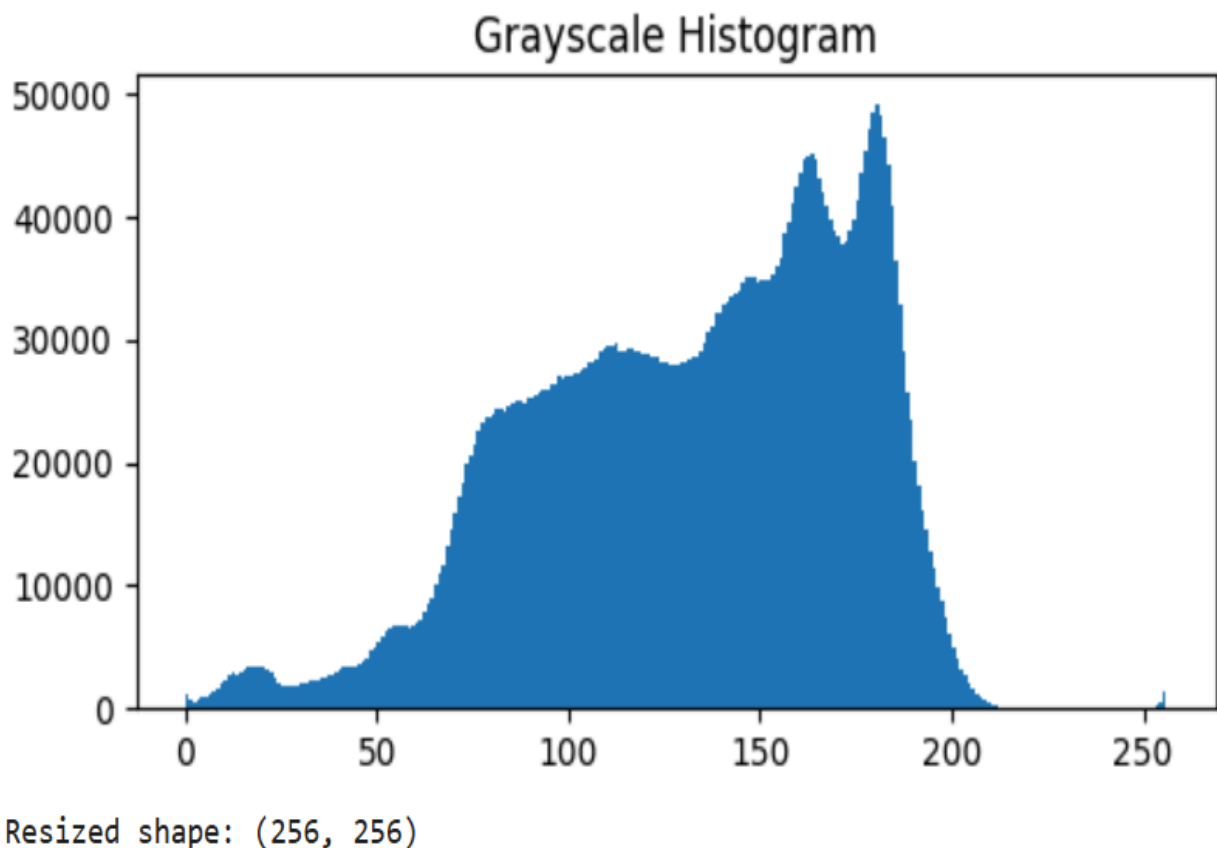


Figure 2.1: Pixel intensity distribution of a Chest X-ray.

2.2 Canny Edge Detection

To identify anatomical structures like the ribcage or the heart boundary, we implemented the Canny Edge Detector. This is a multi-stage algorithm:

1. **Gaussian Filter:** First, the image is smoothed to remove noise that might be mistaken for an edge.

2. **Sobel Kernel:** The algorithm calculates the intensity gradient (derivative) of the image. Areas with high gradients (sharp changes from black to white) are potential edges.
3. **Non-Maximum Suppression:** Thick edges are thinned down to a single pixel width.
4. **Hysteresis Thresholding:** Weak edges are discarded unless they are connected to strong edges.

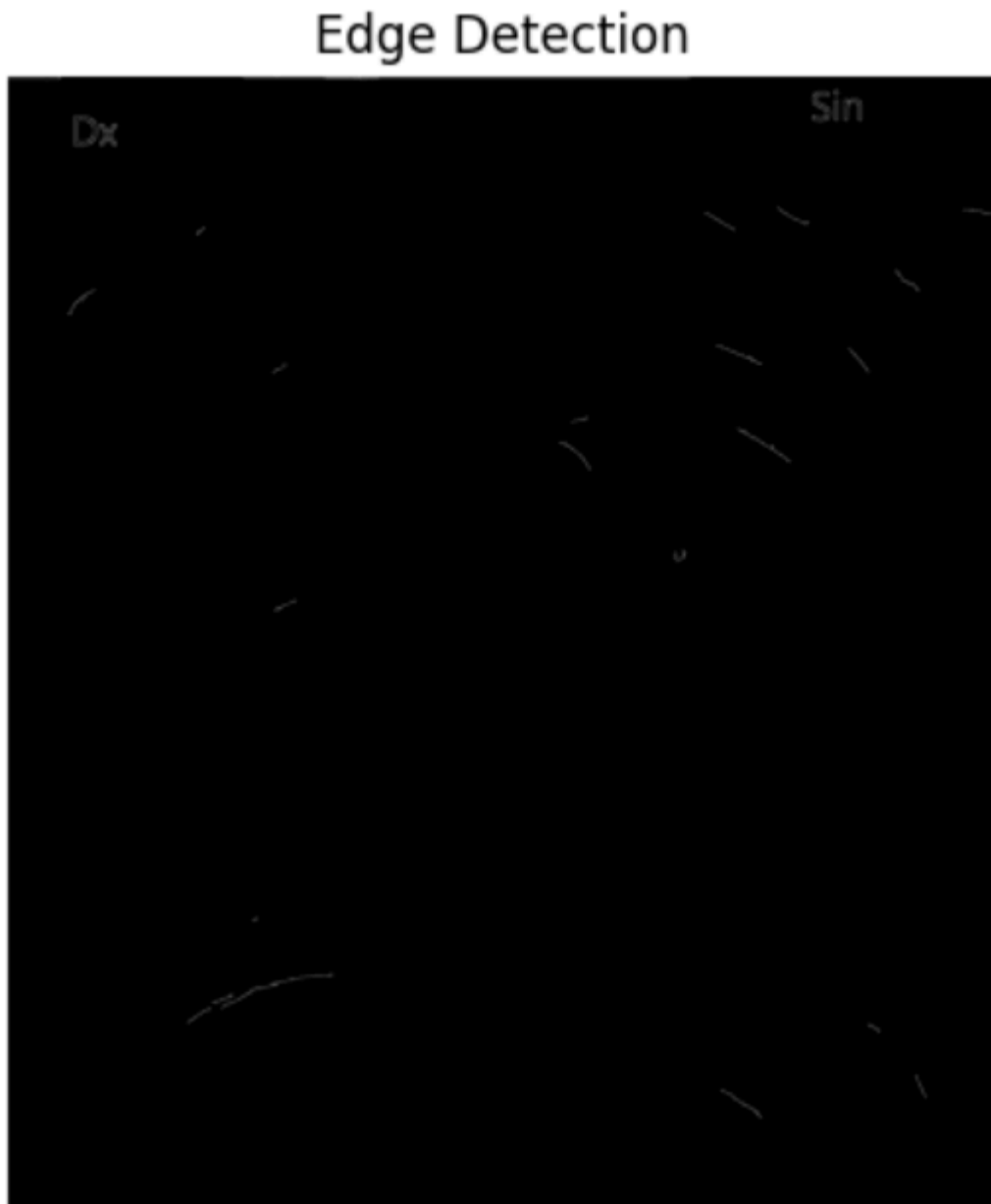


Figure 2.2: Structural boundaries identified using Canny Edge Detection.

2.3 Noise Simulation and Restoration

Medical imaging hardware often introduces "noise" (grain) due to sensor limitations or low radiation dosage.

- **Simulation:** We artificially added Gaussian Noise to a clean X-ray to simulate a low-quality scan.
 - **Restoration:** We applied a **Gaussian Blur** filter. This works by convolving the image with a kernel that averages the pixel values with their neighbors.
 - **Result:** While the noise was successfully reduced, we observed a critical trade-off: the image became blurrier. In a clinical setting, this is dangerous as it might obscure small fractures or nodules. This experiment highlighted why advanced "Edge-Preserving" filters are preferred in modern healthcare AI.
-

Chapter 3: Convolutional Neural Networks & Skin Cancer Detection

Week 3 marked the transition to Deep Learning. We moved away from manual feature extraction (like Canny) and allowed a Convolutional Neural Network (CNN) to learn features automatically.

3.1 The Dataset: Skin Computer Vision Model

We utilized a dataset hosted on Roboflow containing dermoscopic images of skin lesions.

- **Source:** The International Skin Imaging Collaboration (ISIC).
- **Classes:** The dataset included three primary classes:
 1. **Melanoma:** The most dangerous form of skin cancer.
 2. **Basal Cell Carcinoma (BCC):** A common but less aggressive cancer.
 3. **Nevus:** A benign mole.
- **Audit:** A pre-training audit revealed a class imbalance. Benign Nevus images significantly outnumbered Melanoma images. This reflects the real-world prevalence of disease but poses a challenge for training, as the model might become biased toward predicting "Benign."

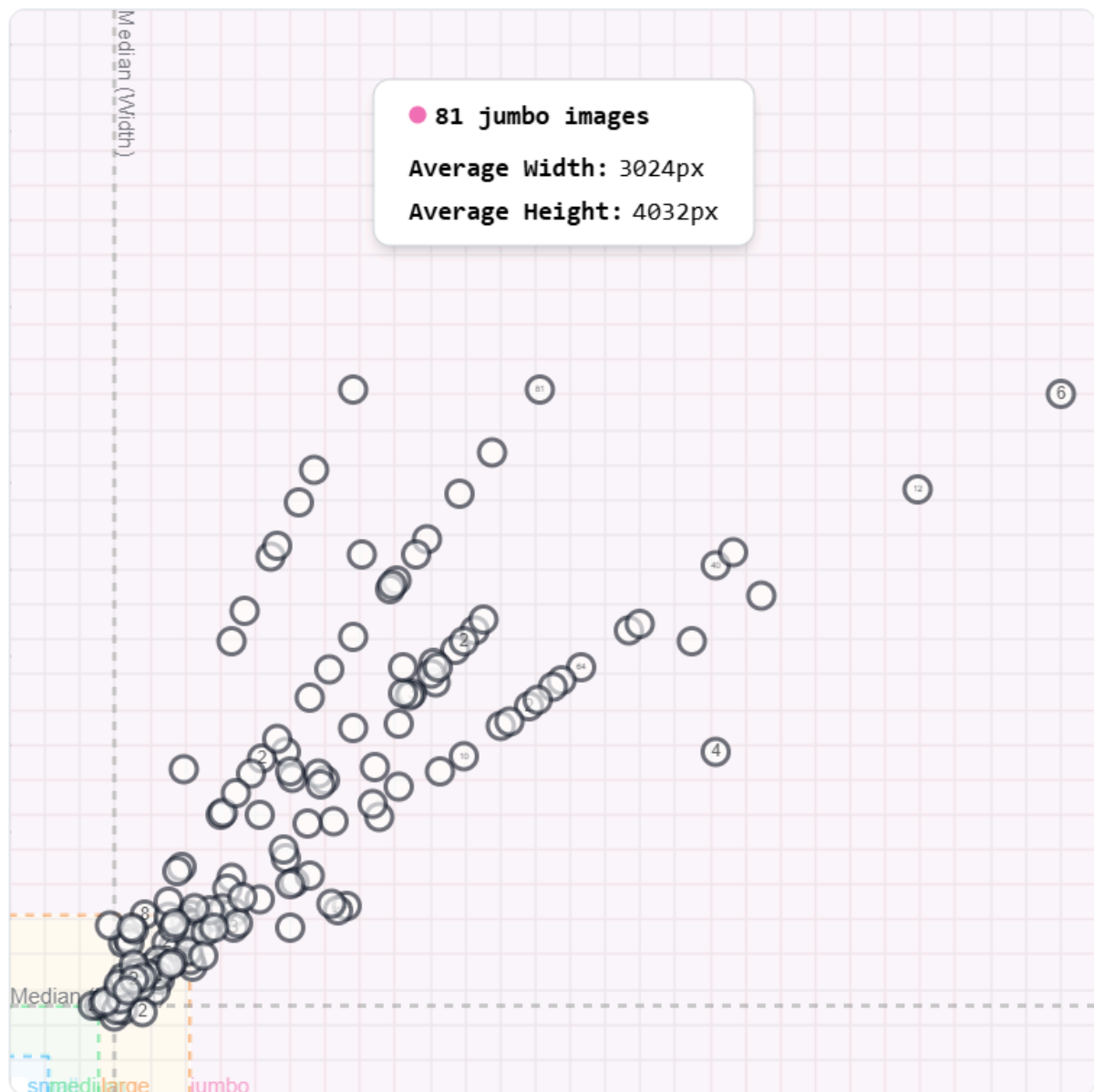


Figure 3.1: Class distribution of the training data.

3.2 Model Architecture: YOLOv8

We selected the **YOLOv8 (You Only Look Once)** architecture. While typically known for object detection, the **yolov8n-cls** (Nano Classification) variant was chosen for its efficiency.

- **Why YOLO?** Traditional CNNs like ResNet can be computationally heavy. YOLO is designed for real-time inference, making it suitable for deployment on mobile devices used by dermatologists in remote areas.
- **Training Setup:**
 - **Platform:** Google Colab (T4 GPU).
 - **Epochs:** 10.
 - **Image Size:** 224x224 pixels.

- **Optimizer:** AdamW.

3.3 Experimental Results

The training run was highly successful. The model converged rapidly, with the validation loss stabilizing by Epoch 8.

- **Accuracy:** The final model achieved a **Top-1 Accuracy of 97.7%**.
- **Confusion Matrix Analysis:** The confusion matrix is the most critical metric in medical AI. It shows us *where* the model made mistakes.
 - **False Negatives:** The model had near-zero False Negatives for Melanoma. This is crucial; missing a cancer diagnosis is far worse than falsely flagging a benign mole as cancer (False Positive).



Figure 3.2: Confusion Matrix showing prediction performance across classes.

Chapter 4: Advanced Case Study – The TIGER Challenge

The final phase of the mentorship focused on **Computational Pathology**, specifically the analysis of Whole Slide Images (WSIs) for breast cancer prognosis.

4.1 The Clinical Problem: TILs

Tumor-Infiltrating Lymphocytes (TILs) are immune cells that migrate into a tumor to fight it.

- **Prognostic Value:** High levels of TILs in breast cancer patients are strongly correlated with better survival rates and better responses to immunotherapy.
- **The Bottleneck:** Currently, pathologists must estimate TIL levels by looking through a microscope. This is subjective, tedious, and prone to inter-observer variability.

4.2 The TIGER Grand Challenge

The "Tumor Infiltrating lymphocytes in Genus breast canceR" (TIGER) challenge aims to develop open-source AI algorithms to automate this scoring.

- **Task:** We focused on **Detection**. Unlike the classification task in Week 3 (which assigns one label to an entire image), this task required the model to identify and locate *every single lymphocyte* in a crowded image patch.

4.3 Pipeline Implementation

We designed a training pipeline using **YOLOv8** in Object Detection mode.

- **Dataset:** We utilized the **wsiroisimages** dataset from Roboflow, which consists of high-resolution patches extracted from biopsy slides.
- **Methodology:**
 1. **Data Loading:** ROI patches were loaded with bounding box annotations around lymphocytes.
 2. **Training:** The model was trained to regress bounding box coordinates.
 3. **Challenges:**
 - **Occlusion:** Lymphocytes often cluster together, making it hard for the model to distinguish individual cells.
 - **Mimics:** Tumor cells can look very similar to immune cells. The model had to learn subtle textural differences in the cell nucleus to differentiate them.


```

!pip install ultralytics roboflow

from roboflow import Roboflow
from ultralytics import YOLO
from IPython.display import Image
import os

rf = Roboflow(api_key="zrosbvi0h0UK3bPx0IGD")

project = rf.workspace("xray-u9rf3").project("wsiroisimages")
version = project.version(1)

dataset = version.download("yolov8")

model = YOLO('yolov8n.pt')

print(f"Training on dataset at: {dataset.location}")

results = model.train(
    data=f"{dataset.location}/data.yaml",
    epochs=15,
    imgsz=640,
    plots=True
)

Image(filename=f'{results.save_dir}/results.png')

```

] Figure 4.1: Code snippet of the YOLOv8 detection pipeline.

Chapter 5: Conclusion and Future Scope

5.1 Summary of Learnings

This mentorship has been a transformative experience, bridging the gap between coding tutorials and real-world medical application.

1. **Data Quality:** The "Garbage In, Garbage Out" principle is magnified in healthcare. A noisy X-ray or an imbalanced skin dataset can render a model useless.
2. **Model Selection:** There is no "one size fits all." We used simple OpenCV kernels for edge detection but required complex CNNs for cancer classification.
3. **Ethics:** In medical AI, metrics like "Accuracy" are insufficient. We learned to prioritize "Sensitivity" (Recall) to ensure life-threatening conditions are not missed.

5.2 Future Directions

Building on this foundation, I intend to explore:

- **Semantic Segmentation:** Moving from bounding boxes to pixel-perfect masking using UNet architectures to calculate exact tumor areas.
- **Deployment:** wrapping the Skin Cancer classifier into a Streamlit web app to make it accessible to users.

This project confirms that AI is not here to replace doctors, but to arm them with powerful, quantitative tools that enhance human decision-making.

References

1. Redmon, J. (2025). *The Ancient Secrets of Computer Vision*. University of Washington. [Week 1 Material]
2. Hinton, G. (2025). *Getting Started with Neural Networks*. [Week 2 Material]
3. Ultralytics. (2025). *YOLOv8 Documentation*. [Week 3 Material]
4. TIGER Grand Challenge. (2026). *Tumor Infiltrating Lymphocytes in Breast Cancer*. [Week 4 Material]
5. Roboflow Universe. *Skin Computer Vision Model Dataset*.
6. ISIC Archive. *International Skin Imaging Collaboration*.