In [1]:
```
from Utils import *
```

In [2]:
```
# load test data
testImages, testOutputxi, testOutputyi, testOutputxw, testOutputyw = readImageDa
```

In [3]:
```
# normalizing pixel values
testImagesNormalized = np.array(testImages) / 255.0
```

In [4]:
```
# show some test images
titles = [f"x={x}, y={y}" for x, y in zip(testOutputxi[:4], testOutputyi[:4])]
display_images(testImages[:16], rows=2, cols=2, titles=titles)
```

In [5]:
```
import os

# Folder where the models are stored
model_folder = "random_forest_models"

# List all the model files in the folder
model_files = [f for f in os.listdir(model_folder) if f.endswith(".pkl")]
```

In [6]:
```
# Prepare the Outputs
testOutputs = np.array([
    testOutputxi,  # x values
    testOutputyi,  # y values
    testOutputxw,  # x_width values
    testOutputyw   # y_width values
], dtype=float).T  # Transpose to align correctly
```

In [7]:
```
# Visualize predictions and ground truth on a sample test image
original_size = (3264, 2448)  # Original image dimensions
resized_size = (128, 128)     # Resized image dimensions
scale_factor_x = resized_size[0] / original_size[0]
scale_factor_y = resized_size[1] / original_size[1]
```

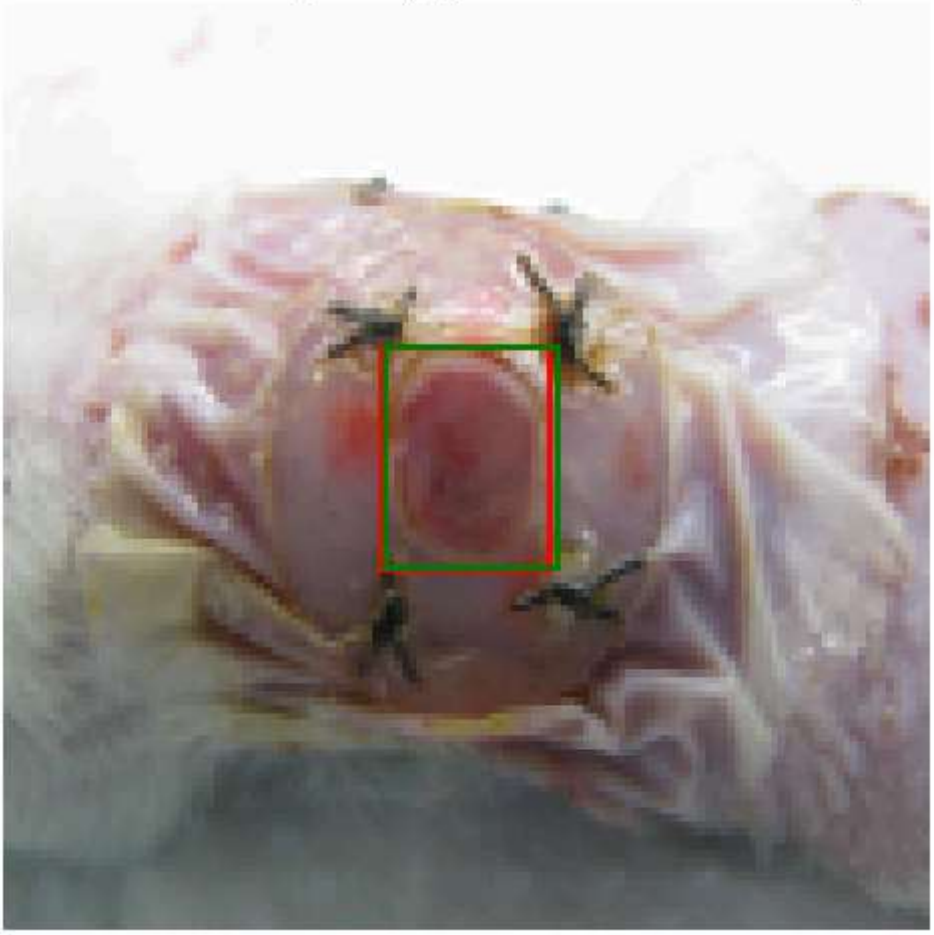
In [8]:
```
import joblib

pca_path = "random_forest_pcas/pca.pkl"
pca = joblib.load(pca_path)

# Apply PCA to the test data
testImages_flattened = np.array([img.flatten() for img in testImagesNormalized])
testImages_reduced = pca.transform(testImages_flattened)
```

In [9]:
```
# import joblib
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
# Iterate over each model file and load it
for model_file in model_files:
    model_path = os.path.join(model_folder, model_file)

    # Load the model
    model = joblib.load(model_path)
    print(f"Loaded model: {model_file}")

    # Make predictions with the loaded model
    if model_file == 'random_forest_model.pkl':
        predictions = model.predict(testImages_flattened)
    else:
        predictions = model.predict(testImages_reduced)

    # Evaluate the performance
    mse = mean_squared_error(testOutputs, predictions)
    mae = mean_absolute_error(testOutputs, predictions)
    r2 = r2_score(testOutputs, predictions)
    print(f"Performance metrics for {model_file}:")
    print(f"Mean Squared Error: {mse:.2f}")
    print(f"Mean Absolute Error: {mae:.2f}")
    print(f"R² Score: {r2:.2f}")

    # Calculate the errors for all predictions
    errors = []
    for i in range(len(testImages)):
        pred = predictions[i]
        gt = testOutputs[i]
        error = calculate_error(pred, gt)
        errors.append((i, error))

    # Sort the errors by ascending order (low error to high error)
    errors.sort(key=lambda x: x[1])
```

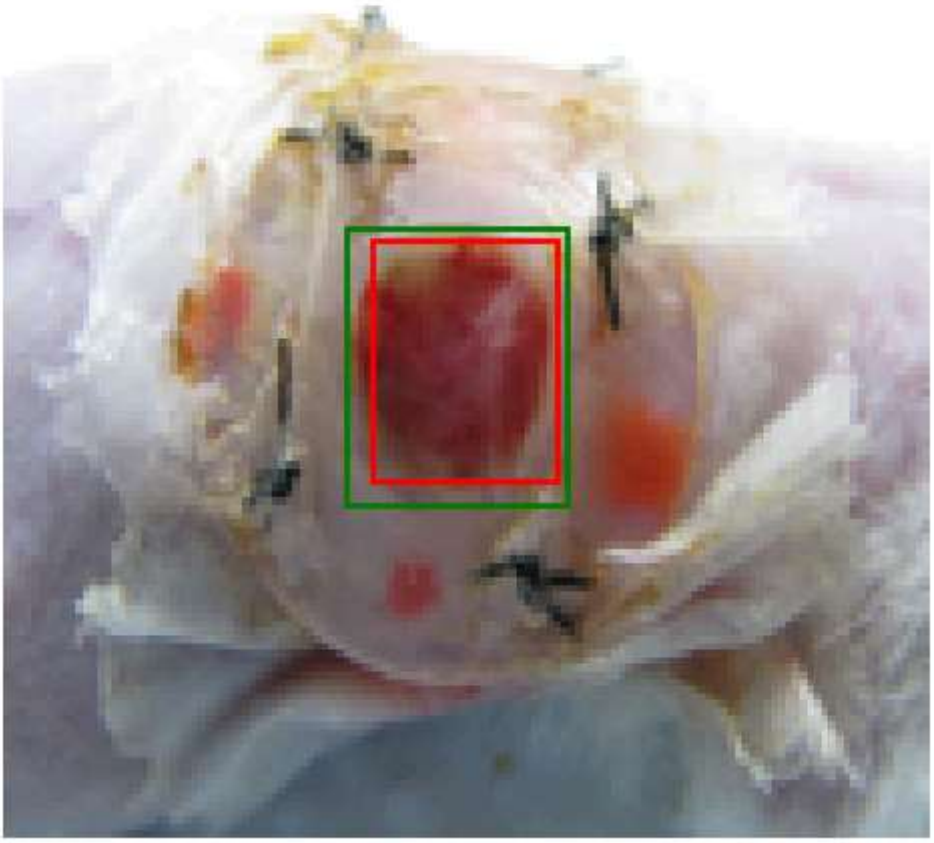
```
    # Select well-predicted images (low error) and hard-to-predict images (high
    num_images = 1  # Number of images to display
    well_predicted_indices = [errors[i][0] for i in range(num_images)]  # First
    hard_predicted_indices = [errors[-i-1][0] for i in range(num_images)]  # Fir

    # Combine indices for mixed display
    selected_indices = well_predicted_indices + hard_predicted_indices

    for i in selected_indices:
        img = testImages[i]
        pred = predictions[i]
        gt = testOutputs[i]  # Replace with your ground truth list or array

        # Unpack predictions and ground truth
        pred_x, pred_y, pred_xw, pred_yw = pred
        gt_x, gt_y, gt_xw, gt_yw = gt

        # Create a plot
        fig, ax = plt.subplots(1, figsize=(6, 6))
        ax.imshow(img, cmap='gray')  # Display the image

        # Draw the predicted rectangle (red)
        draw_rectangle(ax, pred_x, pred_y, pred_xw, pred_yw, 'red', scale_factor

        # Draw the ground truth rectangle (green)
        draw_rectangle(ax, gt_x, gt_y, gt_xw, gt_yw, 'green', scale_factor_x, sc

        plt.title(f"Prediction (Red): {np.round(pred, 2)}\nGround Truth (Green):
        plt.axis('off')
        plt.savefig(f'results/{model_file}_visualization_{i}.jpg')
        plt.show()
```

```
Loaded model: random_forest_model.pkl
Performance metrics for random_forest_model.pkl:
Mean Squared Error: 1195.41
Mean Absolute Error: 26.64
R² Score: 0.21
```

Prediction (Red): [1610.65 1189.99 596.49 586.18]
Ground Truth (Green): [1634. 1189. 602. 585.]

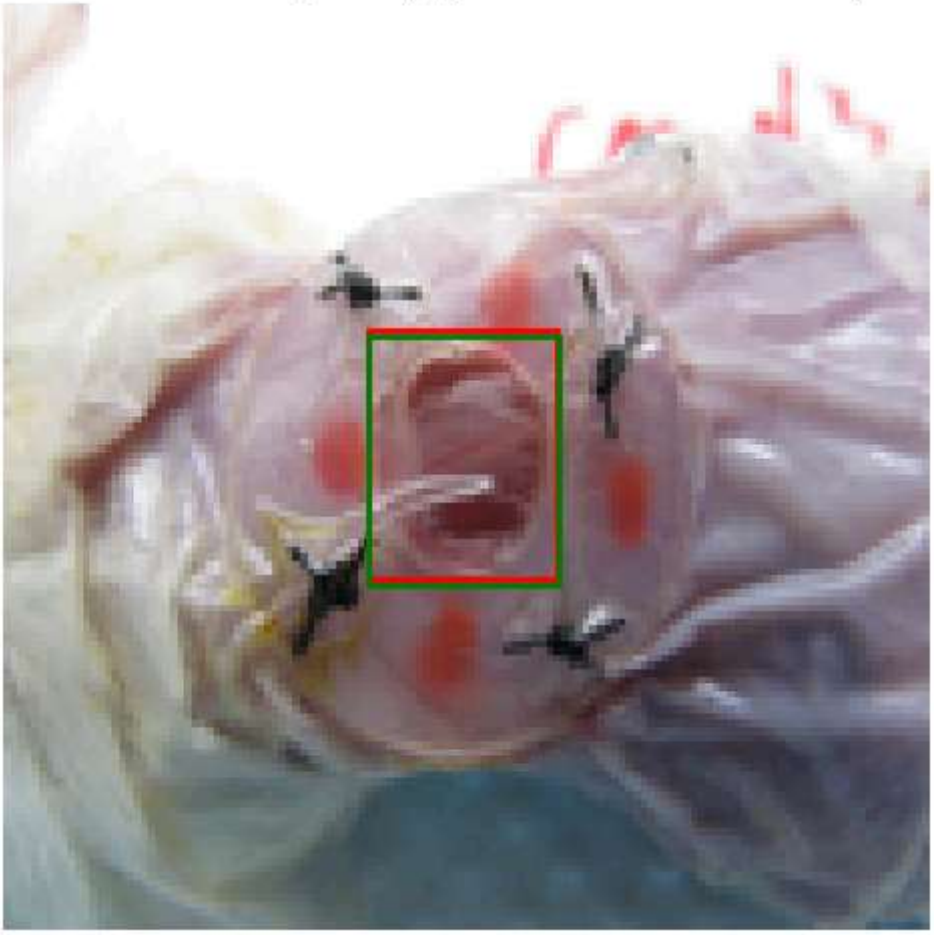


Prediction (Red): [1611.93 1176.53 650.21 639.47]
Ground Truth (Green): [1584. 1193. 780. 736.]

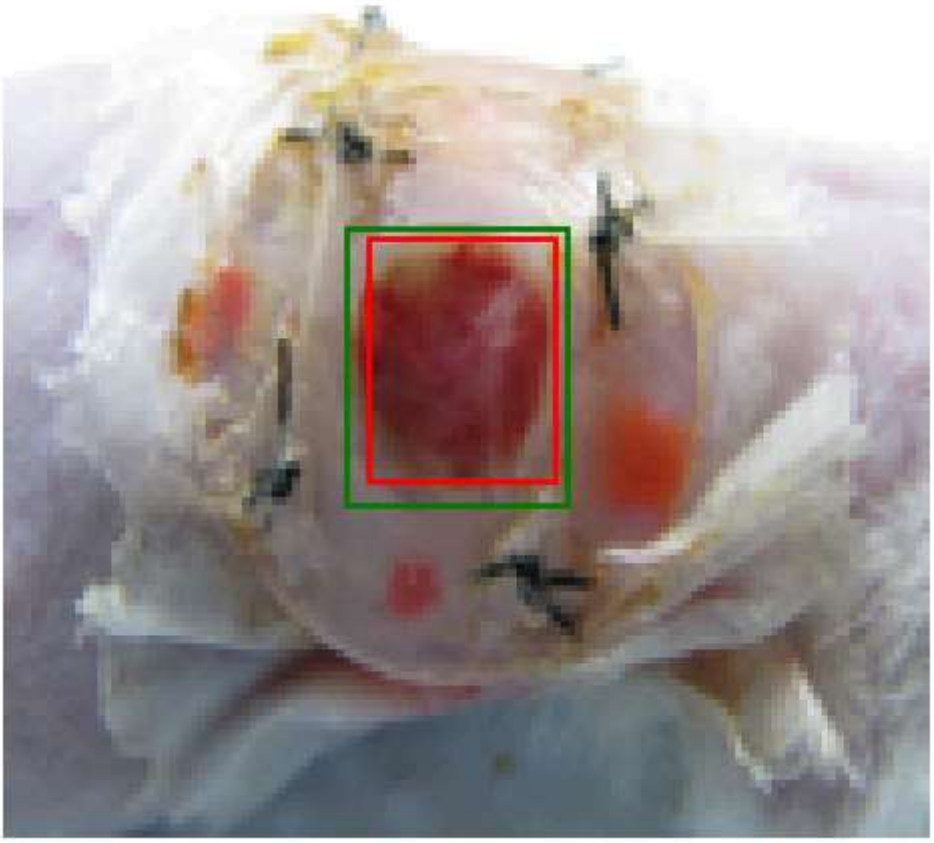


```
Loaded model: random_forest_model_top_1_score_2579.78.pkl
Performance metrics for random_forest_model_top_1_score_2579.78.pkl:
Mean Squared Error: 1614.85
Mean Absolute Error: 31.55
R² Score: 0.05
```

Prediction (Red): [1605.99 1183.97 667.91 658.68]
Ground Truth (Green): [1608. 1198. 666. 657.]

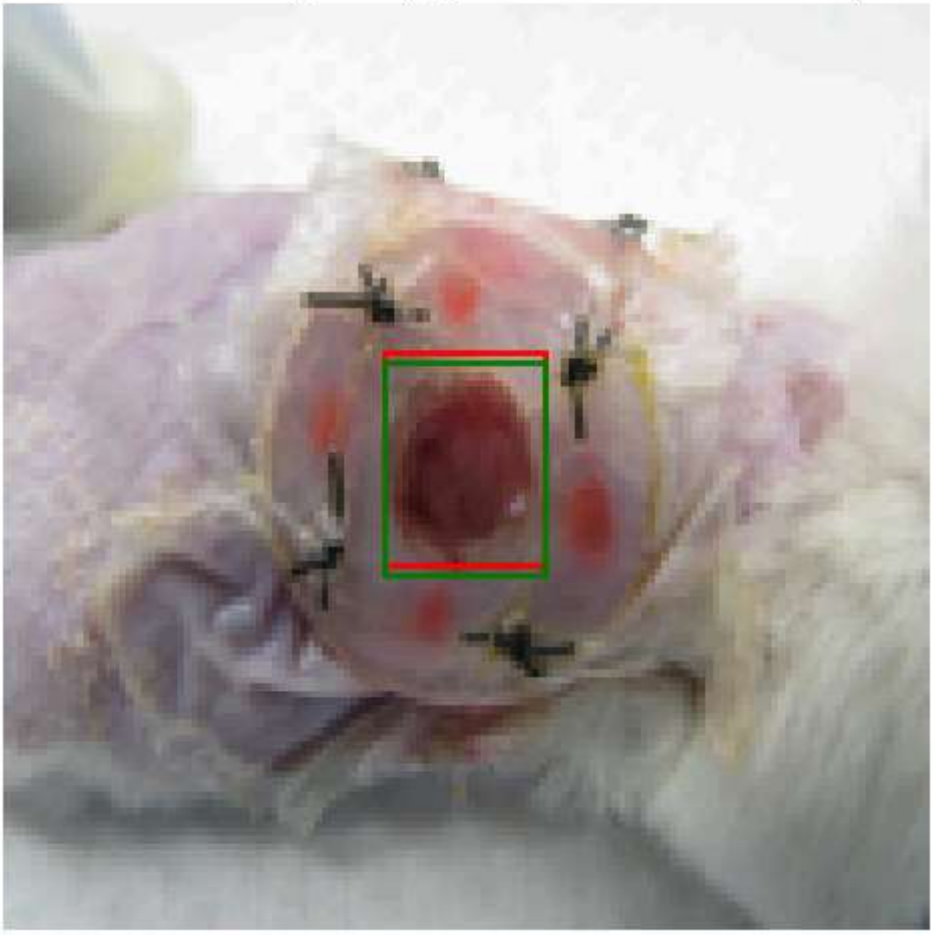


Prediction (Red): [1598.57 1175.99 656.15 643.32]
Ground Truth (Green): [1584. 1193. 780. 736.]

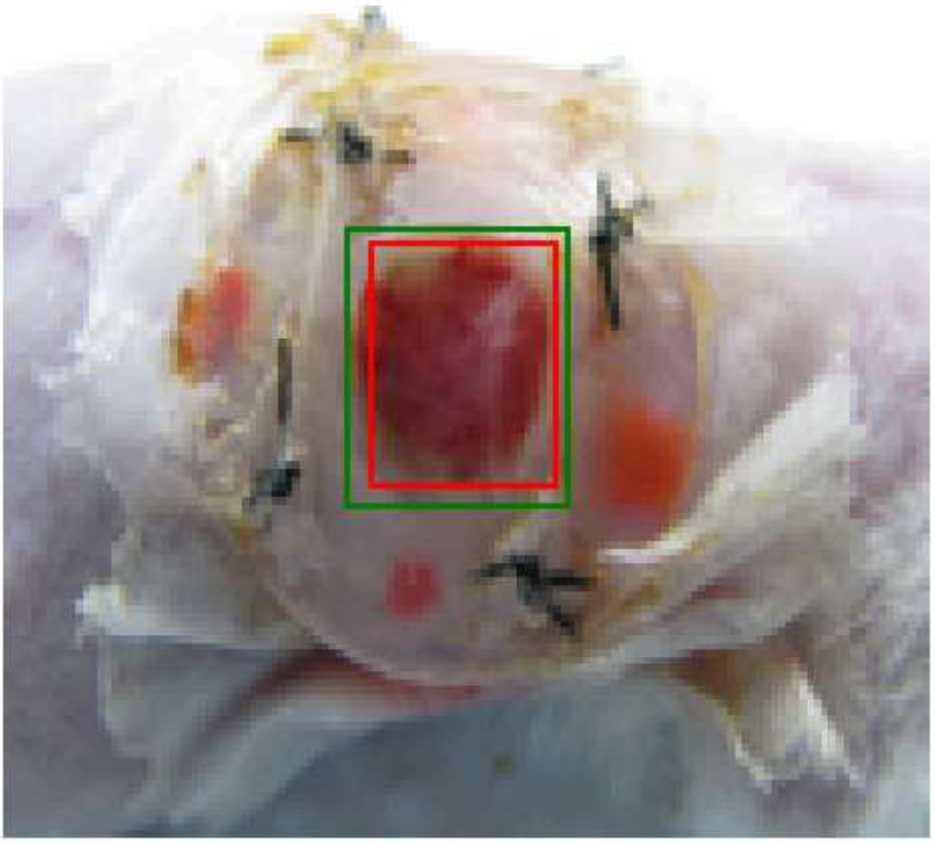


```
Loaded model: random_forest_model_top_2_score_2587.91.pkl
Performance metrics for random_forest_model_top_2_score_2587.91.pkl:
Mean Squared Error: 1644.96
Mean Absolute Error: 31.85
R² Score: 0.03
```

Prediction (Red): [1611.89 1195.17 571.64 561.24]
Ground Truth (Green): [1613. 1220. 569. 559.]

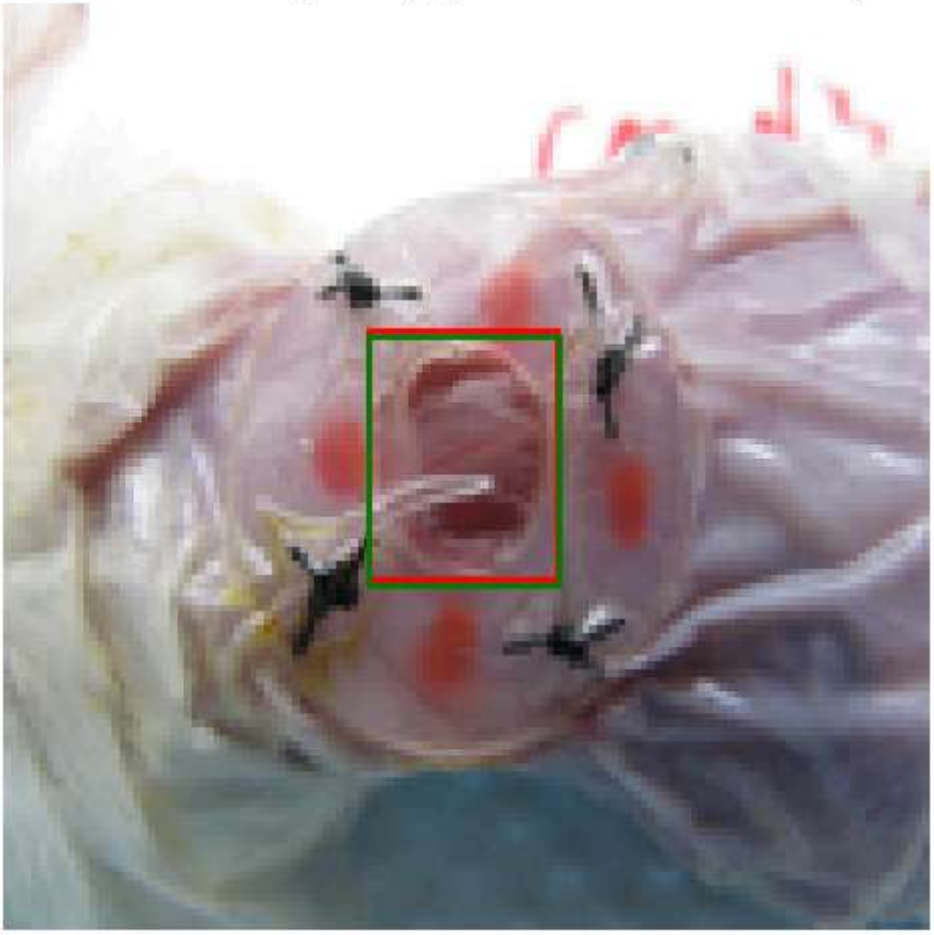


Prediction (Red): [1606.67 1185.68 652.65 640.69]
Ground Truth (Green): [1584. 1193. 780. 736.]

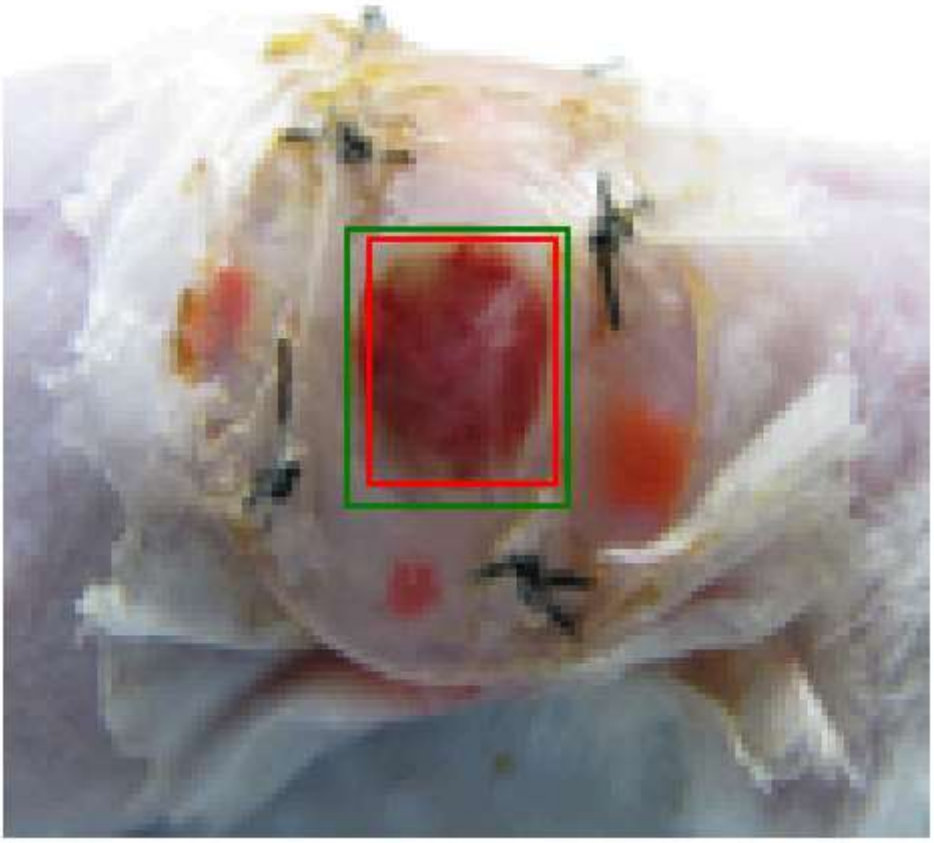


```
Loaded model: random_forest_model_top_3_score_2589.07.pkl
Performance metrics for random_forest_model_top_3_score_2589.07.pkl:
Mean Squared Error: 1623.25
Mean Absolute Error: 31.64
R² Score: 0.04
```

Prediction (Red): [1605.71 1183.38 668.16 658.99]
Ground Truth (Green): [1608. 1198. 666. 657.]

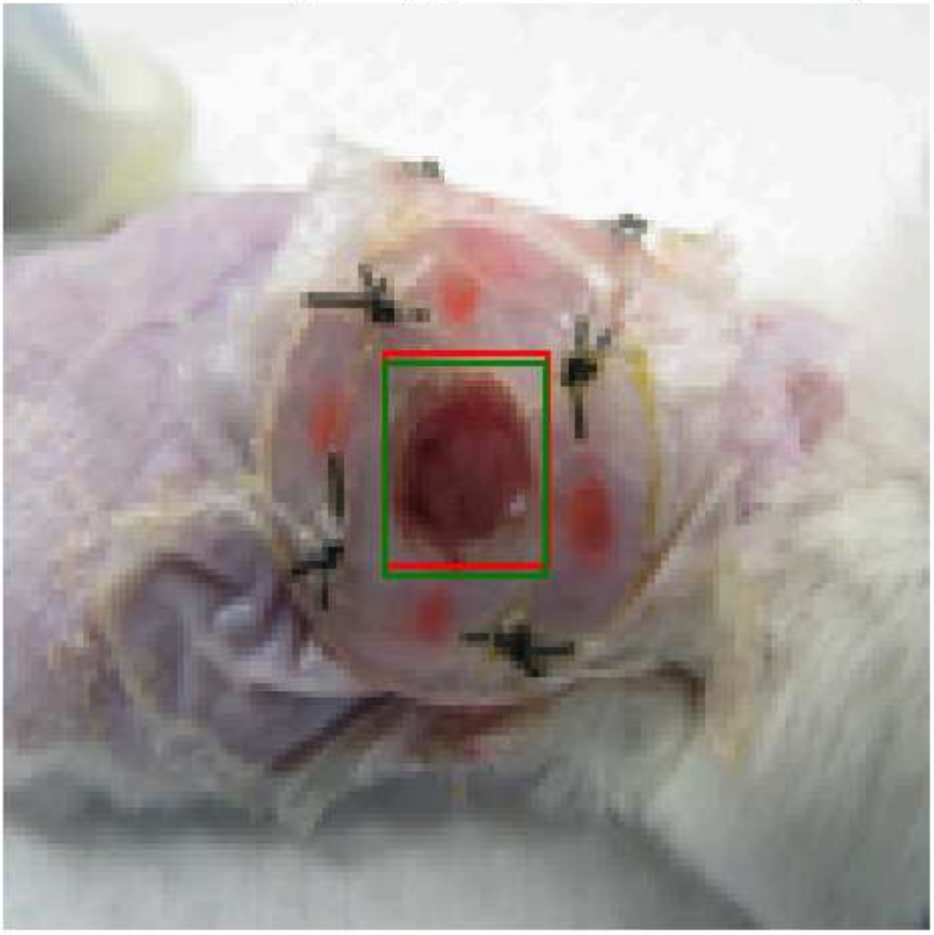


Prediction (Red): [1599.73 1178.37 656.87 644.82]
Ground Truth (Green): [1584. 1193. 780. 736.]

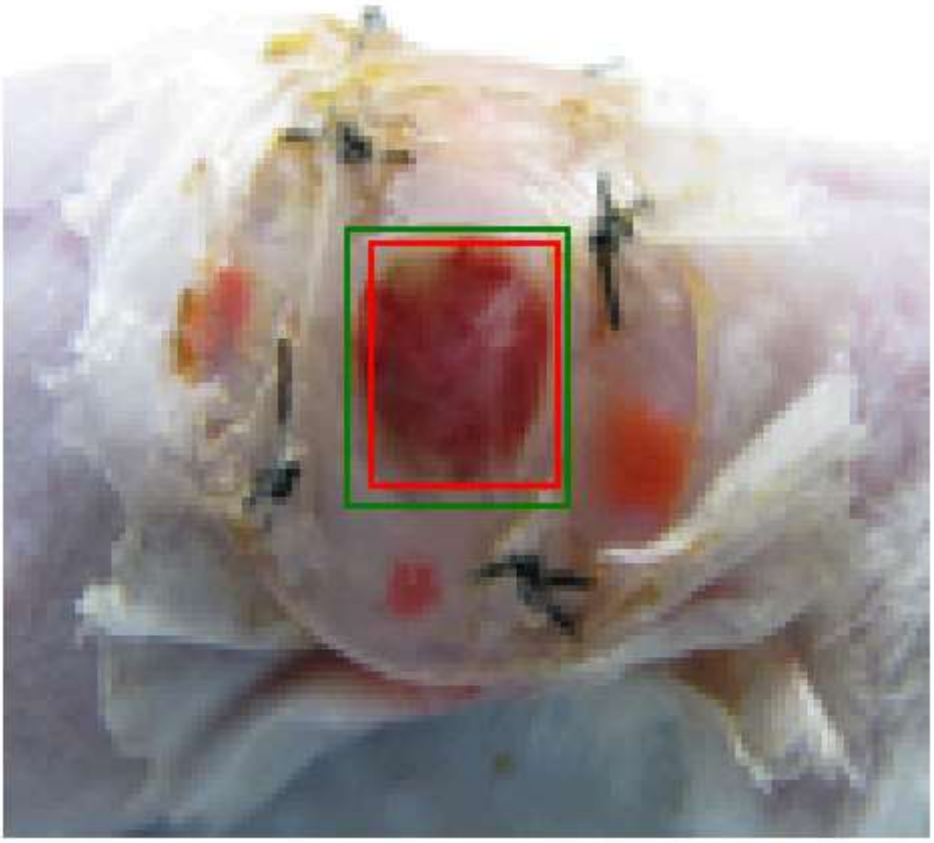


```
Loaded model: random_forest_model_top_4_score_2601.39.pkl
Performance metrics for random_forest_model_top_4_score_2601.39.pkl:
Mean Squared Error: 1653.72
Mean Absolute Error: 31.96
R² Score: 0.04
```

Prediction (Red): [1612.7 1194.4   573.12 561.66]
Ground Truth (Green): [1613. 1220. 569. 559.]

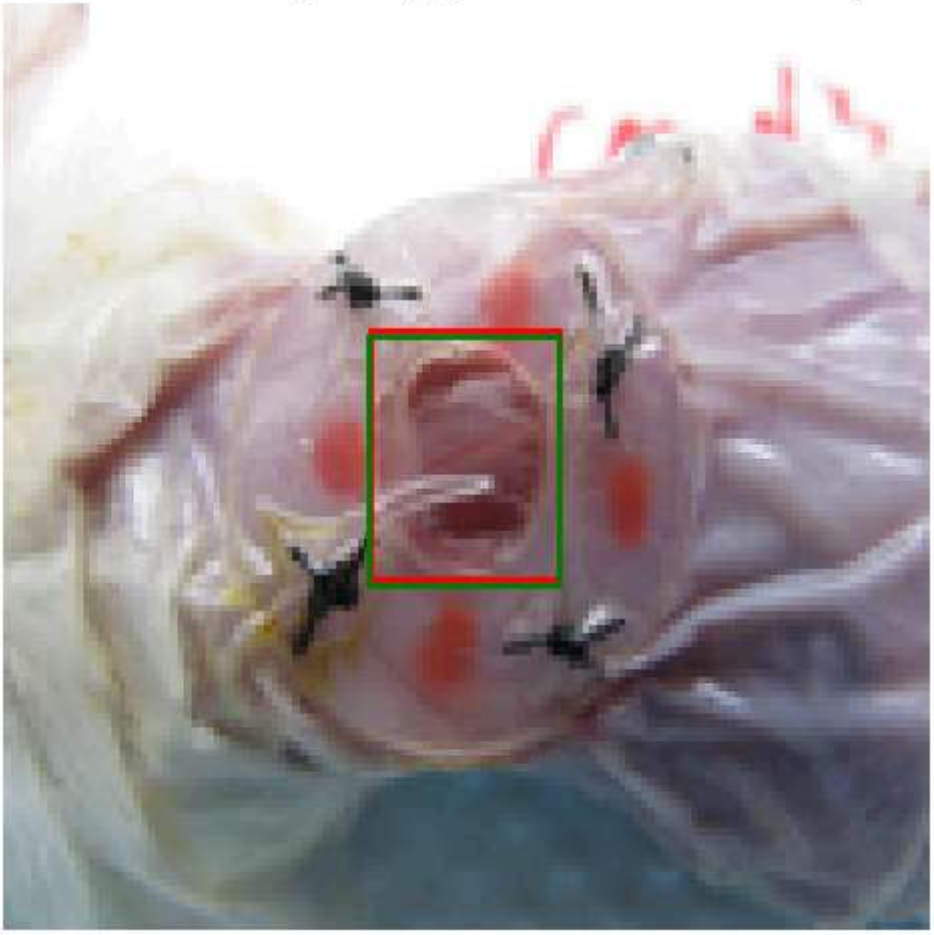


Prediction (Red): [1607.52 1185.95 652.73 640.69]
Ground Truth (Green): [1584. 1193. 780. 736.]

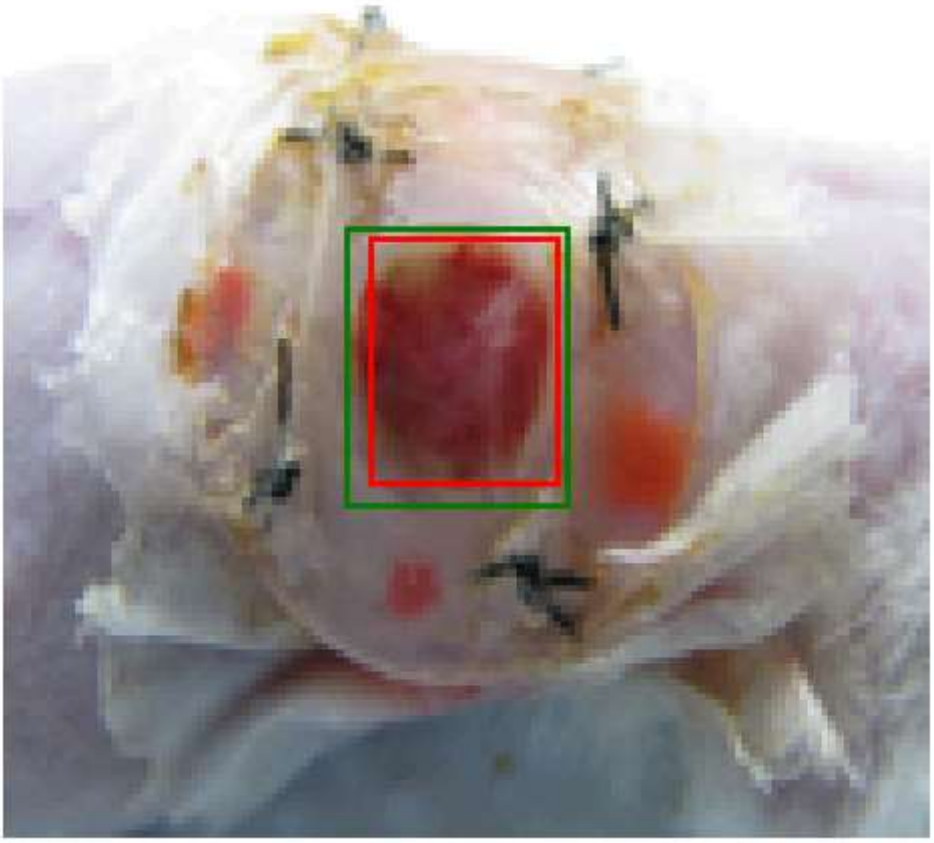


```
Loaded model: random_forest_model_top_5_score_2608.72.pkl
Performance metrics for random_forest_model_top_5_score_2608.72.pkl:
Mean Squared Error: 1616.64
Mean Absolute Error: 31.49
R² Score: 0.04
```

Prediction (Red): [1609.16 1183.81 664.9   654.6]
Ground Truth (Green): [1608. 1198. 666. 657.]



Prediction (Red): [1606.67 1177.16 656.45 644.07]
Ground Truth (Green): [1584. 1193. 780. 736.]



In [ ]: