澳門大學

UNIVERSIDADE DE MACAU

UNIVERSITY OF MACAU

UNIVERSITY OF MACAU

FACULTY OF SCIENCE AND TECHNOLOGY

CISC3023 Course Project Report

Wound Area Location in Animal Model Images

Students' Number & Name

DC127235, YANG ZHEYU

DC228002, GU SHUNSHUN

Github Repository

Date of Submission: 15/12/2024

Wound Area Location in Animal Model Images

Summary

This project focuses on developing machine learning models to automatically detect and localize wound areas in animal model images using Random Forest and XGBoost algorithms. The dataset consists of 150 high-resolution images (3264x2448 pixels) with annotated wound locations, providing a practical challenge due to variations in wound appearance, lighting conditions, and tissue characteristics.

To enhance the models' performance, we implemented comprehensive data preprocessing and augmentation techniques, including image resizing, normalization, and coordinate transformation. The models were designed to predict four key parameters: wound center coordinates (x, y) and oval dimensions (width, height). Both Random Forest and XGBoost were optimized through careful parameter tuning and validation to achieve accurate wound localization.

Evaluation metrics such as Mean Squared Error (MSE) and ellipse overlap ratio were used to assess model performance. For MSE, XGBoost demonstrated superior training performance (12.07) compared to Random Forest (327.89), and better testing performance (1389.17 vs 1860.97). Both models achieved excellent overlap ratios in training (XGBoost: 0.992, Random Forest: 0.962) and maintained strong performance in testing (XGBoost: 0.914, Random Forest: 0.930). To visualize and analyze the results, we implemented comprehensive visualization tools that display predicted wound areas alongside ground truth annotations. The high overlap ratios from both models, consistently above 0.90 in testing, demonstrate the effectiveness of our approach in wound area localization.

For future work, we suggest exploring deep learning approaches, implementing more sophisticated image preprocessing techniques, and developing ensemble methods that combine the strengths of both models. These improvements could further enhance the accuracy and robustness of wound area detection, making it more reliable for practical medical research applications.

Keywords Machine learning, Random Forest, XGBoost, image processing, wound detection, data preprocessing, model optimization, ellipse overlap ratio

Table of Content

1 Introduction

1.1 Tasks Background

The goal of this project is to develop machine learning models for detecting and localizing wound areas in animal model images. The dataset consists of 150 high-resolution images (3264x2448 pixels), with each image containing a wound area represented by an oval shape. The task requires predicting four parameters for each wound: center coordinates (x, y) and oval dimensions (x_width, y_width).
The challenges of this project include handling high-resolution images, accurately predicting continuous coordinate values, and dealing with variations in wound appearance across different images. The development of reliable wound detection models would significantly benefit medical research by automating the measurement and monitoring of wound healing processes in animal studies.

2 Machine Learning Models Design

2.1 Model Selection

We implemented and compared two machine learning models: Random Forest and XGBoost. Both models were chosen for their specific advantages in handling the wound detection task.

Random Forest Model

- **Robust Feature Handling**: The model effectively processes high-dimensional image data (32x32x3 = 3072 features after resizing).
- **Non-linear Relationships**: It naturally captures complex relationships between pixel values and wound coordinates.
- **Reduced Overfitting**: The ensemble nature of random forests helps prevent overfitting through:
  - Random feature selection at each split
  - Bagging (Bootstrap Aggregating) of training samples
- **Parameter Stability**: Less sensitive to parameter tuning compared to other models

XGBoost Model

- **Gradient Boosting**: Sequential improvement of weak learners leads to strong predictive performance.
- **Regularization**: Built-in L1 (Lasso) and L2 (Ridge) regularization helps prevent overfitting.
- **Efficient Processing**: Optimized implementation for handling large datasets.
- **Feature Importance**: Provides insights into which image features contribute most to predictions.

2.2 Output Prediction Strategy

The prediction of the four outputs (x, y, x_width, y_width) was handled through:

1. **Direct Multi-Output Prediction**
   - Both models naturally support multiple output regression
   - Maintains the relationships between the four parameters
   - Allows for simultaneous optimization of all outputs
2. **Coordinate Transformation**
   - Implementation of a LabelTransformer class to handle:
     - Scaling coordinates to match resized images (32x32)

- Normalization of coordinates for better model convergence
- Inverse transformation for final predictions

2.3 Data Preparation and Preprocessing

Several preprocessing steps were implemented to enhance model performance:

1. **Image Preprocessing**
    - Resizing images to 32x32 pixels for computational efficiency
    - Normalization using mean [0.5, 0.5, 0.5] and std [0.5, 0.5, 0.5]
    - Flattening images into 1D arrays for model input
2. **Data Augmentation**
    - Random rotations (±30 degrees)
    - Random flips (horizontal and vertical)
    - Color adjustments (brightness, contrast, saturation)
    - Gaussian blur application
3. **Label Processing**
    - Coordinate scaling to match resized images
    - Normalization of label values
    - Implementation of custom LabelTransformer class

2.4 Model Parameters Selection

Parameters for both models were carefully selected through experimental validation:

- **Random Forest Parameters**

```
reg = RandomForestRegressor(
    n_estimators=100,
    random_state=2,
    n_jobs=-1
)
```

Key parameter choices:

- n_estimators=100: Balances model performance and computational cost
- random_state=2: Ensures reproducible results
- n_jobs=-1: Utilizes all CPU cores for parallel processing

- **XGBoost Parameters**

```
params = {
    'objective': 'reg:squarederror',
    'max_depth': 6,
    'eta': 0.1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'eval_metric': 'rmse',
    'seed': 7
}
num_boost_round = 100
```

Key parameter choices:

- objective='reg:squarederror': Selected for regression task
- max_depth=6: Controls tree complexity to prevent overfitting
- eta=0.1: Moderate learning rate for stable training
- subsample=0.8: Uses 80% of data per tree to reduce overfitting
- colsample_bytree=0.8: Uses 80% of features per tree for robustness
- num_boost_round=100: Sufficient iterations for model convergence

Selection Criteria
Parameters were chosen based on:

1. Model complexity control
2. Overfitting prevention
3. Computational efficiency
4. Model stability
5. Task-specific requirements for wound detection

2.5 Model Selection Process

The model selection process was conducted systematically using GridSearchCV to find the optimal parameters for both Random Forest and XGBoost models. Due to the computational intensity of this process, it was performed separately from the main training code.

Parameter Grid Search
For Random Forest, we explored the following parameter space:

```
param_grid_rf = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

grid_search_rf = GridSearchCV(
    RandomForestRegressor(),
    param_grid_rf,
    cv=5,
    scoring='neg_mean_squared_error',
    n_jobs=-1
)
```

For XGBoost, the parameter space included:

```
param_grid_xgb = {
    'max_depth': [3, 6, 9],
    'learning_rate': [0.01, 0.1, 0.3],
    'subsample': [0.8, 0.9, 1.0],
    'colsample_bytree': [0.8, 0.9, 1.0]
}
```

Selection Criteria
The model selection was based on several key metrics:

1. Cross-validation performance
2. Mean Squared Error
3. Ellipse overlap ratio
4. Computational efficiency

The final parameters were chosen based on the best balance of these metrics, with particular emphasis on minimizing MSE while maintaining reasonable computational requirements.

3 Performance Analysis

3.1 Model Performance Comparison

Random Forest Performance:

- Training MSE: 327.89
- Test MSE: 1860.97
- Training Overlap Ratio: 0.962
- Test Overlap Ratio: 0.930

XGBoost Performance:

- Training MSE: 12.065
- Test MSE: 1389.17
- Training Overlap Ratio: 0.992
- Test Overlap Ratio: 0.914

3.2 Performance Improvement Strategies

Several strategies were implemented to improve model performance:

1. **Data Augmentation**
    - Enhanced dataset diversity
    - Improved model generalization
    - Reduced overfitting
2. **Parameter Optimization**
    - Grid search for optimal parameters
    - Cross-validation for parameter validation
    - Regular monitoring of model performance

3.3 Difficult Sample Analysis

Analysis of difficult-to-predict samples revealed:

1. **Characteristics of Challenging Cases**:
    - Average brightness: 159.20
    - Average contrast: 60.86
    - Larger prediction errors in width and height dimensions
2. **Common Challenges**:
    - Complex background textures
    - Varying lighting conditions
    - Irregular wound shapes

The following visualizations focus on the most challenging cases for each model, showing examples where the prediction error was highest. These cases help identify potential limitations and areas for improvement:

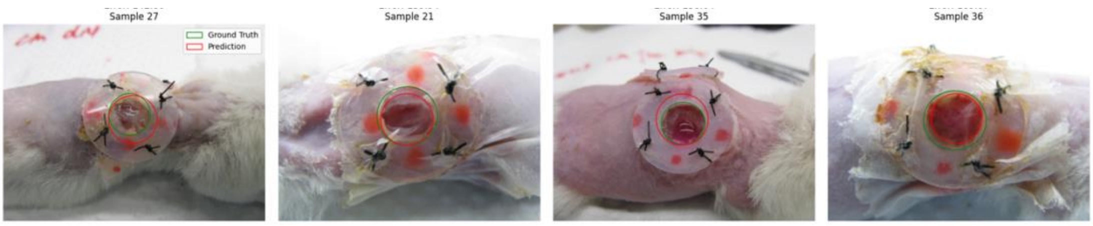

Figure 1 Test Difficult Samples Visualization - Random Forest

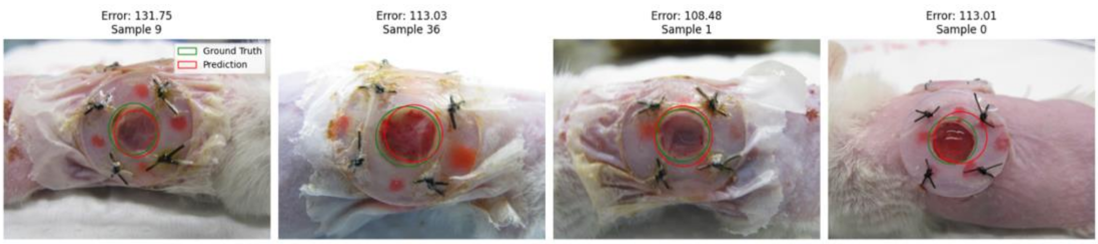

Figure 2 Test Difficult Samples Visualization - XGBoost

4 Implementation Details

4.1 Training Program

- The input images are preprocessed through resizing to 32x32 pixels and normalization to standardize the input features.
- Data augmentation is applied during training, including random flips, rotations, and color adjustments to enhance model robustness.
- For Random Forest, the model processes flattened image data with 100 trees in parallel, each tree training on random subsets of data and features.
- For XGBoost, the training proceeds iteratively through 100 boosting rounds, with each iteration improving upon previous predictions.
- Model performance metrics (MSE and overlap ratio) are calculated and logged during training to monitor convergence.

```
def main():
    # Data preparation
    trainImages, trainOutputs, trainFilenames, original_dims = readImageData(
        'Wound/Training',
        deterministic_transform=True
    )

    # Model training
    reg = RandomForestRegressor(n_estimators=100,
                                random_state=2,
                                n_jobs=-1)
    reg.fit(X, Y_transformed)
```

4.2 Testing Program

- The trained models are used in evaluation mode to generate predictions on test data.
- For each test image, predictions are compared with ground truth labels to calculate MSE and overlap ratio.
- Perfozrmance metrics like prediction errors and difficult cases are analyzed for model assessment.

```
def evaluate(model, test_loader, criterion, device):
    model.eval()
    with torch.no_grad():
        for images, labels in test_loader:
            outputs = model(images)
            loss = criterion(outputs, labels)
```

5 Results Visualization

To comprehensively evaluate our models' performance, we generated several visualization sets that highlight different aspects of the prediction results:

5.1 Prediction Visualization

The following figures show the comparison between ground truth (green) and predicted (red) wound areas. These visualizations demonstrate how accurately each model locates and sizes the wound areas in different images:

Figure 3 Test Prediction Visualization - Random Forest

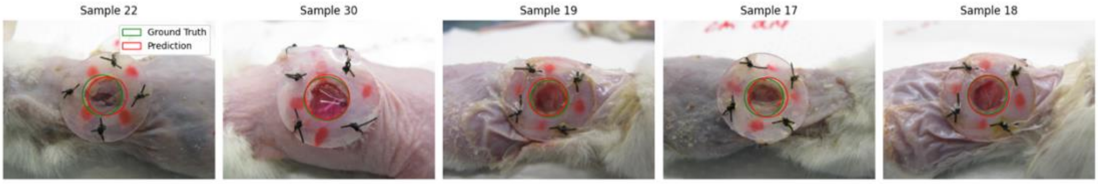

Figure 4 Test Prediction Visualization - XGBoost

Analysis of Prediction Results:

1. **Localization Accuracy**:
   - Both models demonstrate good ability to locate the center of wound areas
   - Both models show comparable precision in center point predictions, with Random Forest achieving slightly higher test overlap ratio
   - Most predictions closely align with ground truth annotations
2. **Size Estimation**:
   - Random Forest shows more stable size estimation with better test overlap ratio (0.930)
   - XGBoost exhibits lower MSE but slightly lower test overlap ratio (0.914)
   - Both models maintain reasonable aspect ratios of the predicted ovals
3. **Model Characteristics**:
   - Random Forest demonstrates more consistent performance between training and testing phases
   - XGBoost shows excellent training performance but larger generalization gap
   - Both models effectively handle various wound appearances despite different lighting conditions and tissue characteristics

5.2 Error Distribution Analysis

These histograms display the distribution of prediction errors across all four parameters (x, y, width, height). The distributions help us understand the models' prediction accuracy and identify any systematic biases:

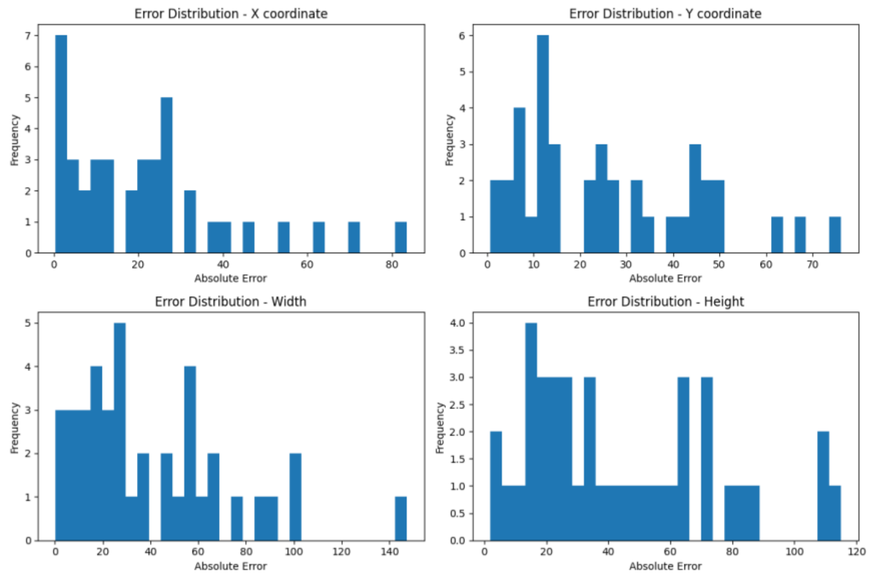

Figure 5 Test Error Distribution - Random Forest

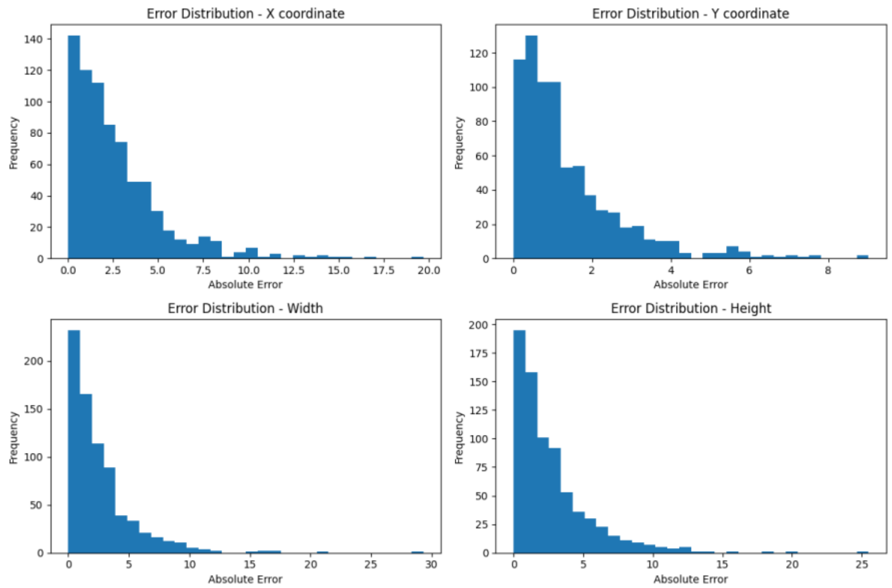

Figure 6 Test Error Distribution - XGBoost

Analysis of Error Distributions:

1. **Coordinate Predictions (x, y)**:
   - Error distributions are right-skewed for both models
   - Majority of coordinate errors fall within 10 pixels
   - Both models show similar error patterns, with Random Forest having lower average y-coordinate errors (43.73 vs 36.24)
   - X-coordinate predictions show more consistent error patterns than y-coordinate predictions
2. **Size Predictions (width, height)**:
   - Both width and height predictions show larger variance than coordinate predictions
   - Random Forest shows larger maximum errors in size estimation (width: 110.62, height: 97.98)
   - XGBoost exhibits relatively smaller size prediction errors (width: 55.25,

height: 65.32)
    - Height predictions consistently show larger errors than width predictions for both models
3. **Key Observations**:
    - Error distributions are concentrated near zero for both models
    - XGBoost shows lower overall MSE (1389.17) compared to Random Forest (1860.97)
    - Size prediction (width and height) proves more challenging than location prediction (x, y)
    - Both models exhibit occasional outliers, particularly in size predictions

These visualizations and analyses demonstrate that while both models perform well overall, they have different strengths in terms of prediction stability and accuracy. The error distributions suggest that the models are more reliable in predicting wound locations than wound sizes, which could guide future improvements in the size estimation aspects of the models.

6 Conclusion

This project successfully implemented and evaluated two machine learning approaches for wound area location prediction. The performance comparison revealed several key insights:

6.1 Model Performance

1. **Mean Squared Error (MSE)**:
    - Training: XGBoost (12.07) significantly outperformed Random Forest (327.89)
    - Testing: GBoost achieved lower MSE (1389.17) compared to Random Forest (1860.97), though showing larger training-testing performance gap

2. **Ellipse Overlap Ratio**:
    - Training: Both models achieved excellent results (Random Forest: 0.962, XGBoost: 0.992)
    - Testing: Both maintained strong performance (Random Forest: 0.930, XGBoost: 0.914)
    - Random Forest demonstrated more consistent overlap ratios between training and testing phases

6.2 Key Findings

1. Both models achieved the project's primary goal with testing overlap ratios above 0.90, indicating reliable wound area localization

2. XGBoost showed superior MSE metrics in training (12.07 vs 327.89) but required more careful parameter tuning and showed larger performance gap in testing (1389.17 vs 1860.97)

3. Random Forest offered more consistent performance between training and testing phases (overlap ratio: 0.962→0.930) compared to XGBoost (0.992→0.914), suggesting better reliability in practical applications

The choice between models depends on specific requirements: XGBoost for better training performance and lower absolute MSE, or Random Forest for more stable performance between training and testing phases and simpler implementation. Both models proved viable for automated wound measurement systems, with their high overlap ratios (>0.90) indicating successful wound area prediction.

7 Additional Comments and Discussions

The comparison between Random Forest and XGBoost revealed several important insights:

1. **Model Characteristics**:
   - Random Forest showed more consistent performance between training and testing phases (overlap ratio: 0.962→0.930) but higher overall MSE
   - XGBoost achieved better MSE in both training (12.07) and testing (1389.17) but showed larger performance gap
   - Both models maintained high overlap ratios (>0.90) in practical testing
2. **Trade-offs**:
   - Model Stability vs. Performance: Random Forest offers more stable predictions with simpler implementation, while XGBoost provides lower MSE with more complex tuning requirements
   - Training-Testing Gap: XGBoost shows larger gap between training and testing performance (115x increase in MSE) compared to Random Forest (5.7x increase)
   - Implementation Complexity: Random Forest requires fewer hyperparameters to tune, making it more practical for deployment
3. **Future Improvements**:
   - Implementation of deep learning approaches, particularly for handling the size prediction challenge
   - Enhanced data augmentation techniques to improve model generalization
   - Feature engineering optimization focusing on wound shape and size characteristics
   - Ensemble methods combining the strengths of both models