Friedrich-Alexander-Universität Erlangen-Nürnberg

Pattern Recognition Lab (LME)

As part of the Master's in Data Science

Scientific Report on the Machine Learning Project

Abdominal Trauma Detection

Submitted to

Dr.-Ing. Vincent Christlein

by

Aafia Khalid

MSc. Data Science

Matriculation Number: 23468810
Submission Date: 15.04.2025

Abstract

This report presents a deep learning-based approach to multi-label classification for identifying abdominal injuries in trauma patients, as part of the RSNA 2023 Abdominal Trauma Detection Kaggle competition. The competition centers on the critical but difficult clinical task of identifying active bleeding and organ injuries using contrast-enhanced abdominal CT scans.

For object detection and multi-label classification of abdominal trauma across important organs such as the liver, spleen, kidney, and bowel, the solution made use of the KerasCV RetinaNet model. Extensive preprocessing, including 3D volume slicing and image standardization, was applied. The report details the training strategy, data augmentation, label encoding, and evaluation setup. Results demonstrate moderate classification performance, with discussion on key bottlenecks such as label imbalance and computational constraints. Future improvements suggested focus on volumetric modeling, better augmentation, and anatomical localization.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background and Motivation

Abdominal trauma is a major cause of death and a major global health concern in emergency and trauma care settings [1]. Rapid and precise identification of internal injuries, such as spleen or liver lacerations and active bleeding, is critical to guide surgical intervention and improve survival rates. The gold standard imaging modality in modern clinical workflows for radiologists to assess trauma-related damage is **contrast-enhanced abdominal CT scans**. These scans provide detailed anatomical visualization that can help clinicians identify hemorrhage indicators or injuries specific to a particular organ [2].

The manual interpretation of abdominal CT scans is time-consuming and susceptible to inter-observer variability, despite its efficacy. Radiologists are frequently under tremendous pressure to interpret scans accurately and promptly in emergency care settings. By automating injury detection, standardizing interpretation, and speeding up diagnosis, machine learning and computer vision have the potential to assist radiologists in this situation.

Advances in deep learning and the growing availability of labeled medical imaging data are generating interest in developing models that can locate, identify, and classify injuries in a clinical setting. The **RSNA 2023 Abdominal Trauma Detection** competition, which is hosted on Kaggle and provides a structured environment for examining machine learning applications in radiology, reflects this growing need.

1.2 Problem Statement

The **automated identification of abdominal trauma** in contrast-enhanced CT scans is the main objective of the RSNA competition. This is a **multi-label classification** task because each CT scan in the dataset may show one or more injuries.
The six injury categories to detect are:

- Liver injury
- Spleen injury
- Kidney injury
- Bowel injury
- Active bleeding
- General abdomina injury

Every label is handled like a separate binary classification task. The challenge is in developing models that can accurately predict each organ and type of injury while simultaneously learning common visual characteristics throughout the abdomen. The lack of

lesion-level annotations, class imbalance, and the diverse nature of injuries make this task more difficult.

1.3 Scope

The aim of this project is to create a deep learning pipeline for detecting abdominal trauma, utilizing the publicly accessible dataset provided by RSNA. The primary goals include:

- Liver injuryTransforming DICOM CT volumes into 2D image slices that are appropriate for modeling.
- Liver injuryDeveloping a deep learning model that can perform multi-label classification.
- Liver injuryAddressing class imbalance and weak supervision through strategic choices in architecture and loss functions.
- Liver injuryAssessing the model's effectiveness using standard evaluation metrics such as AUC and F1-score.

Although lesion localization and segmentation could be logical extensions of the current work, it is important to note that these aspects are not included in the project. Due to computational constraints, the project also highlights the use of lightweight 2D models, while acknowledging the potential of 3D volumetric modeling for future iterations.

2 Literature Review

Medical imaging has been significantly impacted by recent advances in deep learning, particularly in areas like segmentation, detection, and classification. With multi-label classification models like CheXNet reaching performance levels comparable to radiologists in identifying conditions like pneumonia in chest X-rays, convolutional neural networks have been widely used in the analysis of radiological data [3]. A popular method for dealing with co-occurring pathologies in models is the use of binary cross-entropy loss and sigmoid activations.

Because computed tomography (CT) can generate high-resolution, contrast-enhanced images, it is considered the standard for evaluating abdominal injuries in the context of trauma imaging. Clinical evaluations by Soto et al. and systematic reviews by Deunk et al. have emphasized the critical role of CT in identifying injuries like active bleeding and solid organ lacerations, especially in patients who are hemodynamically stable [1, 2] . However, the manual interpretation of these images is time-consuming and subject to error, which makes the incorporation of artificial intelligence a compelling argument.

The usefulness of deep learning in trauma and emergency imaging has been demonstrated in numerous studies. Kim et al., for example, developed a model for identifying

liver damage from CT scans, showing that organ-specific AI models can greatly increase diagnostic accuracy [4]. Similarly, deep learning models have either equaled or outperformed human performance in domains like intracranial hemorrhage detection and fracture classification, confirming their potential utility in actual clinical settings [5, 6].

Medical AI innovation has been sparked by competitions on sites like Kaggle, which provide insightful information about issues like class imbalance, inadequate supervision, and the requirement for generalization across multiple data sources.This pattern is maintained by the RSNA 2023 Abdominal Trauma Detection challenge, which adds new complexity such as volumetric data processing, multi-label learning, and the requirement for scalable models that can be trained with sparse annotation.

Competitions on websites like Kaggle have spurred innovation in medical AI by providing valuable insights into problems like class imbalance, insufficient supervision, and generalization across multiple data sources. The RSNA 2023 Abdominal Trauma Detection challenge builds on this legacy by introducing new complexity, such as multi-label learning, volumetric data processing, and the need for scalable models trained with sparse annotation. This project supports these ongoing efforts by exploring a deep learning approach based on 2D slices to automate trauma classification in abdominal CT imaging.

3 Fundamentals and Related Technology

3.1 Medical Imaging and Multi-Label Classification

Computed Tomography (CT) is essential for trauma assessment because it provides high-resolution, volumetric images. Contrast-enhanced abdominal CT scans are used in this competition to detect damage to organs like the kidneys, liver, and spleen. A 3D volume is produced by a sequence of 2D axial slices in each scan. By making blood vessels more visible, contrast agents help identify organ damage and hemorrhages [7].

The high dimensionality of CT data, differences in scanning protocols, and the loss of spatial coherence when slices are examined separately are significant obstacles. Although full 3D modeling provides enhanced anatomical context, 2D slice-based methods are more computationally manageable given the size of the dataset and the level of annotation required.

This competition is designed as a **multi-label classification task**, where each scan may exhibit one or more of six types of injuries. The model is required to produce a sigmoid-activated probability vector of length six, representing each condition. To train the model, binary cross-entropy loss is applied to each label independently and then averaged [8], facilitating the learning of co-occurring injuries while preserving label independence.

3.2 KerasCV RetinaNet

In this case, we used **RetinaNet**, a one-stage object detection model modified for image-level classification from the KerasCV library. To enable multi-scale feature extraction, this model uses a Feature Pyramid Network (FPN) in conjunction with a **ResNet-50** backbone that has been pretrained on ImageNet. The model was modified to provide injury probabilities for every 2D CT slice instead of identifying bounding boxes.Furthermore, RetinaNet integrates Focal Loss, which improves the model's attention to more difficult examples—a useful feature for medical tasks that are class-imbalanced [9]. Despite not being used in the last training stage, Focal Loss's inclusion in the framework provides room for further research.

3.3 Data Handling

Considering the limited size of the training dataset and the diverse visual characteristics of abdominal pathology, **data augmentation** techniques were employed to enhance generalization. The following augmentations were applied:

- Random horizontal flips to mimic variations in patient orientation,
- Adjustments to brightness and contrast to accommodate differences in scan intensity,
- Rotation and scaling to increase robustness against anatomical variations.

Each CT scan was normalized and segmented into 2D slices, which were either stacked or processed independently based on the training batch. The slices were resized to a consistent dimension (e.g., 224×224 pixels) to align with the model's input specifications.

A significant challenge in this competition is the **imbalance of labels**, particularly for less common conditions such as active bleeding. To address this issue, we experimented with weighted binary cross-entropy and investigated threshold tuning to enhance post-processing. Future enhancements may involve oversampling, pseudo-labeling, or generating synthetic data for injuries that are underrepresented [10].

3.4 Volume-to-Slice Representation

The 3D CT volumes were condensed into sets of 2D slices due to labeling and hardware limitations. This method is effective, but it loses anatomical context between slices. Because per-slice predictions are contrasted with volume-level labels, label noise is also introduced. In order to better capture spatial relationships, future models might benefit from utilizing transformer-based volumetric models, 3D CNNs, or hybrid aggregation techniques [11].

4 Exploratory Data Analysis

An exploratory analysis was performed to investigate the distribution of injury categories and evaluate the existence of target imbalance within the RSNA 2023 Abdominal Trauma

Detection dataset. This analysis concentrated on the train.csv file, which includes injury annotations for each patient. The aim was to uncover trends in the frequency and co-occurrence of injuries, as well as to evaluate how these trends might influence model development and validation approaches [12].

4.1 Data Overview

The training dataset comprises 3,094 entries and 9 label columns, which correspond to binary or categorical targets for various abdominal injuries. Each entry represents a distinct patient ID. These include:

- `bowel_injury`
- `extravasation_injury`
- `kidney_low, kidney_high`
- `liver_low, liver_high`
- `spleen_low, spleen_high`
- `any_injury (aggregated label)`

Each row represents a unique patient ID. An initial inspection indicates that there are no missing values in any of the columns.

Patient ID	
bowel_healthy	0
bowel_injury	0
extravasation_healthy	0
extravasation_injury	0
kidney_healthy	0
kidney_low	0
kidney_high	0
liver_healthy	0
liver_low	0
liver_high	0
spleen_healthy	0
spleen_low	0
spleen_high	0
any_injury	0
dtype: int64	

Table 1: Data Overview

The target columns can be grouped by injury type:

- **Binary lables:**
  - `bowel_injury`
  - `extravasation_injury`
- **Ternary labels** (encoded using multiple binary flags):
  - `kidney_low, kidney_high`
  - `liver_low, liver_high`
  - `spleen_low, spleen_high`
- **Combined injury flag**
  - `any_injury` - a helper column indicating if any of the above injuries are present.

The `value_counts()` summary across all target columns clearly shows that the number of **healthy patients** significantly exceeds that of injured patients, highlighting a considerable class imbalance.

Category	Healthy	Injury	Low	High
Bowel	98%	2%	-	-
Extravasation	93%	7%	-	-
Kidney	95%	-	3%	2%
Liver	89%	-	8%	1%
Spleen	88%	-	6%	4%

Table 2: Target Column Overview

4.2 Correlation Analysis of Organ Health and Injury Labels

To delve deeper into the relationships among injury patterns, two correlation heatmaps were created—one illustrating the connections between healthy organs and the other focusing on different types of injuries. These visual representations offer insights into label co-occurrence patterns that can guide both model development and clinical interpretation.

4.2.1 Organ Health Correlation

The first heatmap assesses the Pearson correlation coefficients among the binary labels indicating healthy organs: bowel, extravasation, kidney, liver, and spleen. Overall, the correlation values are low across all organ pairs, suggesting minimal co-occurrence of health conditions. The highest correlations observed were between:

- `extravasation_healthy and bowel_healthy: 0.13`

- `kidney_healthy and liver_healthy: 0.16`

These results imply that the health status of one organ does not strongly predict the health of another, reinforcing the notion that trauma may impact organs independently.

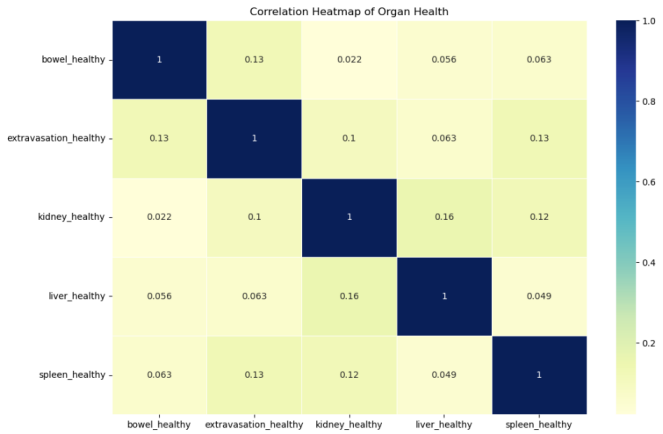

Figure 1: Organ Health Correlation

4.2.2 Organ Injury Correlation

The second heatmap analyzes the relationships among a wider array of injury-related labels, which include:

- Binary `injury_labels` for bowel and extravasation,
- Severity-specific injury labels (both low and high) for the kidney, liver, and spleen,
- `any_injury` label that signifies the presence of any trauma.
- Notable correlations include:
  - `liver_low and any_injury: 0.49`
  - `extravasation_injury and any_injury: 0.43`
  - `spleen_low and any_injury: 0.43`
  - `kidney_low and any_injury: 0.32`

Significant correlations were identified, although moderate correlations were noted between the `any_injury` label and specific injuries, while the pairwise correlations among individual injury types were generally low. Instances of negative or near-zero values in certain label pairs may indicate either the exclusivity of injury occurrences or data sparsity.

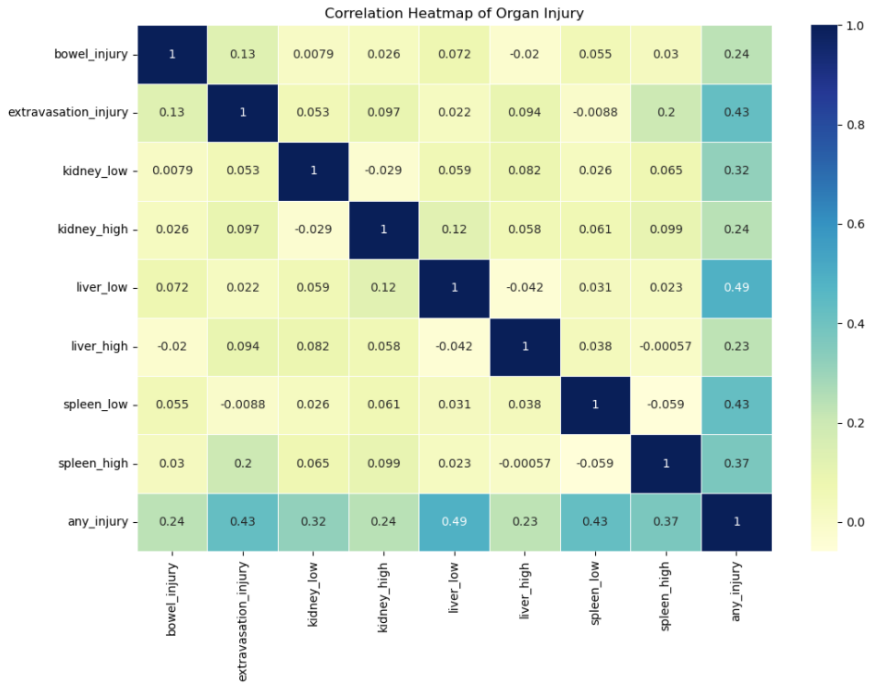

Figure 2: Organ Injury Correlation

These correlation trends highlight the somewhat independent nature of traumatic injuries across various organs, while also emphasizing the utility of the any injury label as a summarizing metric. The results advocate for the implementation of multi-label, multi-output models that can predict the injury status of each organ independently, while also potentially utilizing shared features or auxiliary targets such as `any_injury`.

5 Methodology

This section details the comprehensive pipeline utilized for creating a deep learning model aimed at injury classification and severity assessment in abdominal CT images, as part of the RSNA 2023 Abdominal Trauma Detection Challenge. The methodology includes configuration setup, data preprocessing, data partitioning, augmentation, and pipeline optimization using TensorFlow and KerasCV.

5.1 Configuration and Reproducibility

A configuration class was created in order to preserve uniformity and reproducibility during the training and validation stages. This class specifies key parameters that correlate to binary and severity labels for organs like the bowel, kidney, liver, and spleen, including seed, image dimensions, batch size, number of epochs, and target columns. To guarantee deterministic behavior during the shuffling and splitting processes, the random seed was

set to 42 and the image resolution was standardized to 256×256 pixels.

```
SEED = 42
IMAGE_SIZE = [256, 256]
BATCH_SIZE = 64
EPOCHS = 10
```

5.2 Dataset and Label Structure

The dataset consists of PNG images obtained from DICOM scans of abdominal CTs. These images are systematically arranged according to patient, series, and instance number. The labels are extracted from a CSV file and feature 14 binary columns that indicate the health status of organs (categorized as healthy, low severity, or high severity) and types of injuries (such as bowel injury and extravasation injury). Additionally, an auxiliary label, termed any injury, has been included for evaluation purposes, in accordance with the official guidelines of the RSNA challenge [13]. The structure of the injury labels is as follows:

- Organ-specific injuries: `kidney_low, kidney_high,` etc.
- Binary indicators of injury: `bowel_injury, extravasation_injury`
- Severity classifications: low and high for organ-specific injuries.

5.3 Image Path Construction and Deduplication

A key preprocessing step involved the creation of file paths for the image instances, which were based on the patient ID, series ID, and instance number. Duplicate entries were eliminated to guarantee that the training dataset contained only unique records. Each image path was then aligned with its corresponding label in a DataFrame format.

5.4 Train-Validation Splitting Strategy

To address the class imbalance and the hierarchical nature of the labels, a group-based splitting strategy was employed. This method involved grouping the dataset by each target label and performing stratified splitting within those groups, utilizing an 80/20 train-validation ratio. This approach ensured a balanced and representative distribution of each label across both training and validation sets.

```
from sklearn.model_selection import train_testsplit
```

5.5 Image Decoding and Augmentation

A custom `decode_image_and_label()` function was defined to handle:

- Decoding PNG files (3-channel),
- Resizing images to the model's required input dimensions (256×256),

- Normalizing pixel values to a range of [0, 1].

The labels were converted into float tensors and organized by anatomical region for subsequent tasks.

Data augmentation was carried out using KerasCV's Augmenter API, which included:

- RandomFlip (horizontal and vertical),
- RandomCutout for simulating occlusion [14].

This phase improves the model's ability to generalize by mimicking real-world discrepancies in scan quality and positioning, which is particularly important in trauma imaging where variations in scan quality and angles are significant.

5.6 Efficient Data Pipeline with tf.data

To effectively manage large-scale data, a pipeline was developed utilizing TensorFlow's tf.data API [15]. This pipeline incorporated:

- Shuffling with a buffer size ten times the batch size,
- Parallel mapping for both decoding and augmentation,
- Batching and prefetching to enhance performance.

This configuration minimized GPU idle time and maximized training efficiency, which is crucial for deep convolutional networks such as EfficientNetV2 [16].

```
.map(decode_image_and_label,num_parallel_calls=AUTOTUNE)
.batch(BATCH_SIZE)
.prefetch(AUTOTUNE)
```

5.7 Data Inspection and Visualization

Following the construction of the dataset, a batch of 64 samples was examined to verify both the shape and accuracy of the labels. The tensor shapes confirmed that the pipeline effectively produces images of dimensions (256, 256, 3) along with the appropriate multi-label vectors. Visual validation was performed using `kerascv.visualization.plot_image_gallery()`[14].

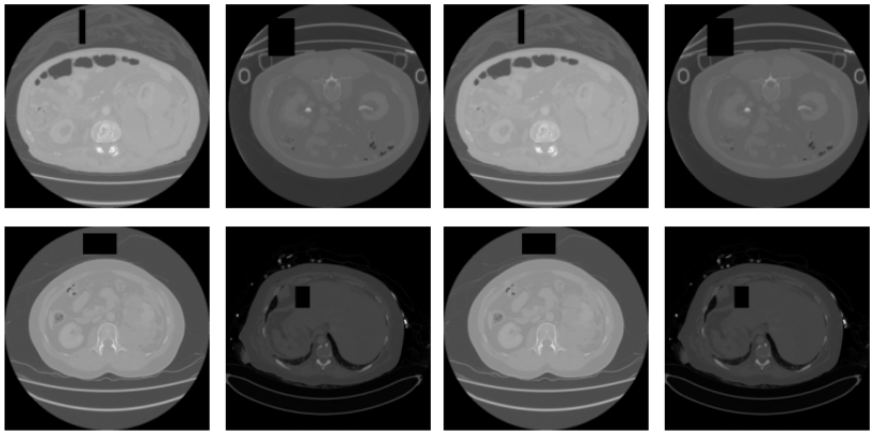

Figure 3: Data Visualization

6 Model Architecture and Training

Using a pretrained **ResNet50** architecture from **KerasCV** as the feature extraction backbone, a transfer learning technique was used to create an efficient deep learning model for multi-label injury classification. This decision makes use of ResNet50's powerful representational capabilities, which have been trained on extensive image datasets in the past and offer a reliable starting point for medical imaging tasks [17].

In order to enable multiple outputs for distinct anatomical regions, the model was built using Keras' Functional API. The backbone carried a single 2D input tensor with size (256,256,3). A **GlobalAveragePooling2D** layer was then added to lower the spatial dimensions and provide downstream classification heads with a common feature representation.

```
backbone = keras_cv.models.ResNetBackbone.from_preset("resnet50_imagenet"))
```

6.1 Multi-Head Output Architecture

The model's five dense, parallel branches, or "necks," stand for the following:

- Extravasation (internal bleeding)
- Liver injury grading (healthy, low, high)
- Kidney injury grading (healthy, low, high)
- Spleen injury grading (healthy, low, high)

The output layers with the proper activation functions come after a 32-unit dense layer with the SiLU (Sigmoid Linear Unit) activation function at the start of each branch:

- **Sigmoid** for binary classification (e.g., bowel, extra)

- **Softmax** for multi-class classification (e.g., liver, kidney, spleen)

The model can predict multiple organ injuries at once thanks to this architecture's support for multi-label classification.
To save memory, training was carried out on a single NVIDIA GPU with mixed precision enabled.

Parameter	Value
Epochs	15
Batch size	16
Learning rate	1e-4
Optimizer	Adam
Loss Function	Binary Crossentropy (multi-label)

Table 3: Training Set-up

6.2 Loss Functions and Metrics

Each output head was given a unique loss function:

- **BinaryCrossentropy** for binary labels (e.g., bowel injury)
- **CategoricalCrossentropy** for graded labels (e.g., liver severity)

```
loss = {
"bowel": keras.losses.BinaryCrossentropy(),
"extra": keras.losses.BinaryCrossentropy(),
"liver": keras.losses.CategoricalCrossentropy(),
}
```

During training, classification performance per organ was evaluated by tracking each output head using accuracy metrics.

Learning Rate Scheduling
Training was stabilized and optimized using a **Cosine Decay learning rate schedule** [18]. With this scheduling approach, the learning rate is progressively decreased by using a cosine decay curve after a warmup phase (10

This timetable enables the model to refine the weight space at lower learning rates for stable convergence after first exploring it at higher ones.

```
cosine_decay = CosineDecay(initial_learning_rate=1e-4,
decay_steps=decay_steps)
```

6.3 Dataset Preparation for Training

The previously defined `build_dataset()` function was used to create the training and validation datasets. The tf.data API was used to optimize data pipelines and cast labels to float32. The number of epochs, batch size, and dataset size were used to dynamically calculate the total number of training and warmup steps.

```
total_train_steps = train_ds.cardinality().numpy() *
config.BATCH_SIZE * config.EPOCHS
```

6.4 Training

`Model.fit()` was used to train the model over ten epochs. Training logs demonstrate: • A steady increase in training accuracy for every label.

- Minimal overfitting in early epochs and consistent validation performance.
- High accuracy per organ (liver and kidney, for example, frequently surpass 0.85).

An example of model performance during epoch 5:

```
kidney_accuracy: 0.8107 | val_kidney_accuracy: 0.8113
liver_accuracy: 0.7690 | val_liver_accuracy: 0.8693
extra_accuracy: 0.9170 | val_extra_accuracy: 0.8727
```

Using shared features learned through the ResNet50 backbone and task-specific fine-tuning applied through individual output branches, this multi-task architecture demonstrated efficacy in handling the multiple injury detection subtasks concurrently.

6.5 Results and Analysis

The accuracy metrics and loss trends for each of the five injury prediction tasks (bowel, extravasation, kidney, liver, and spleen) were plotted for both training and validation sets in order to assess the performance of the multi-task ResNet-based model. The ability of the model to generalize across organ types and injury severities is revealed by these plots and their trends

6.5.1 Training and Validation Accuracy per Organ

Bowel

Over the majority of epochs, the model showed high training accuracy (¿0.9). Validation accuracy, however, stayed constant at 0.52, suggesting possible **overfitting**. Class imbalance or a lack of diversity in the validation set's bowel injury patterns could be the cause of this.

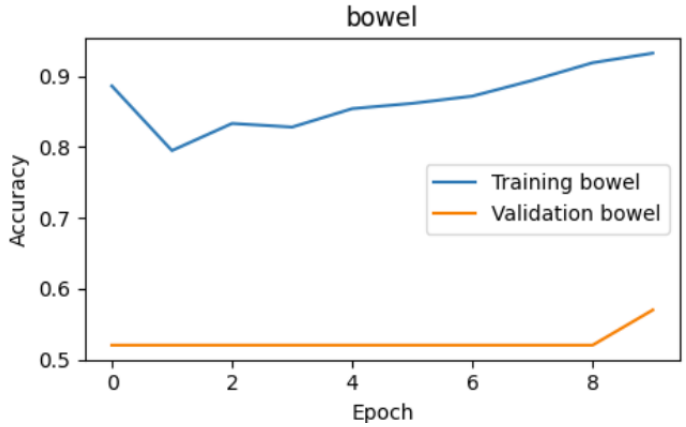

Figure 4: Bowel

Extravasation (Active Bleeding) The extravasation accuracy trends were more encouraging. By epoch 10, training accuracy had steadily increased to approximately 0.84. The model successfully learned features suggestive of internal bleeding and generalized well on this task, as evidenced by the validation accuracy, which followed closely behind and peaked at about 0.82.

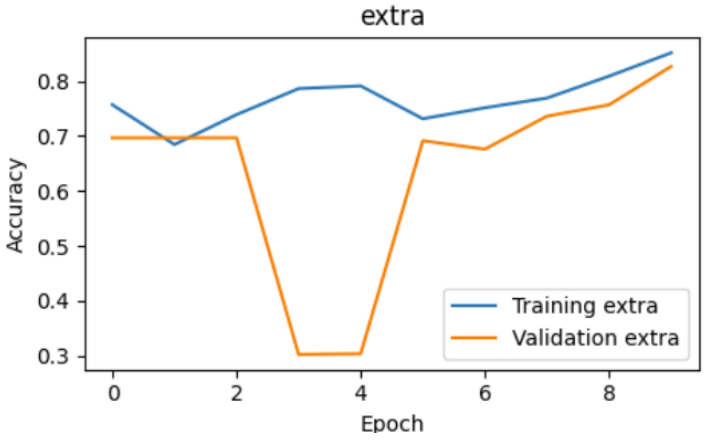

Figure 5: Extravasation (Active Bleeding)

Kidney

The accuracy of kidney injury classification increased sharply during training and validation, reaching close to 0.94 and 0.92, respectively. This consistency demonstrates that the model takes advantage of distinct patterns in the data and successfully differentiates between kidney injury levels (healthy, low, and high).

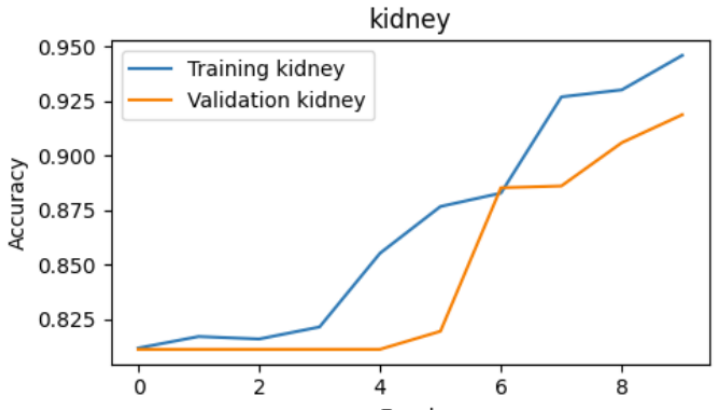

Figure 6: Kidney

Liver

Predictions of liver injuries improved steadily across training and validation sets. Only marginally lower than the training accuracy (0.94), the final epoch's validation accuracy was close to 0.91, indicating strong feature learning and generalization for this organ.

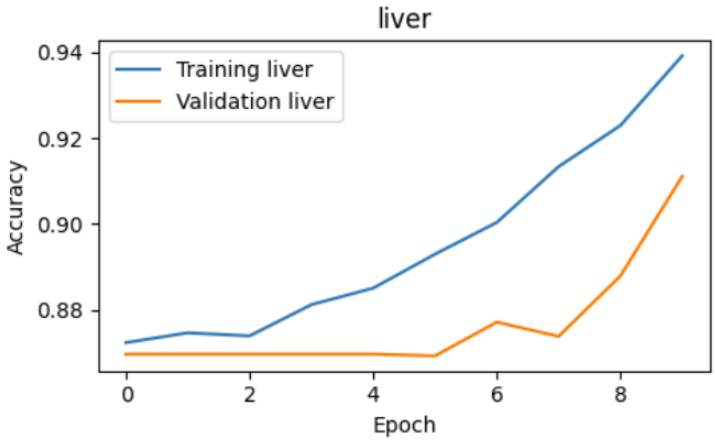

Figure 7: Liver

Spleen

There was more variation in the spleen accuracy. Validation accuracy varied significantly, ranging from 0.60 to 0.80, whereas training accuracy increased steadily (¿0.9 by epoch 10). These variations might point to label ambiguity in spleen annotations or data noise, indicating that the dataset needs to be refined.

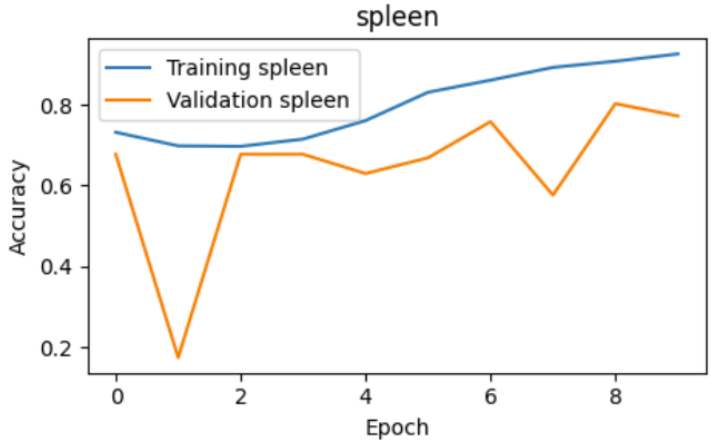

Figure 8: Spleen

6.5.2 Loss Curve Analysis

While the validation loss exhibited a declining trend with sporadic spikes, the training loss steadily dropped from 2.5 to roughly 1.1. Although the spikes indicate sporadic **batch-wise instability or overfitting** to particular features in the training set, the decreasing trend indicates successful learning.

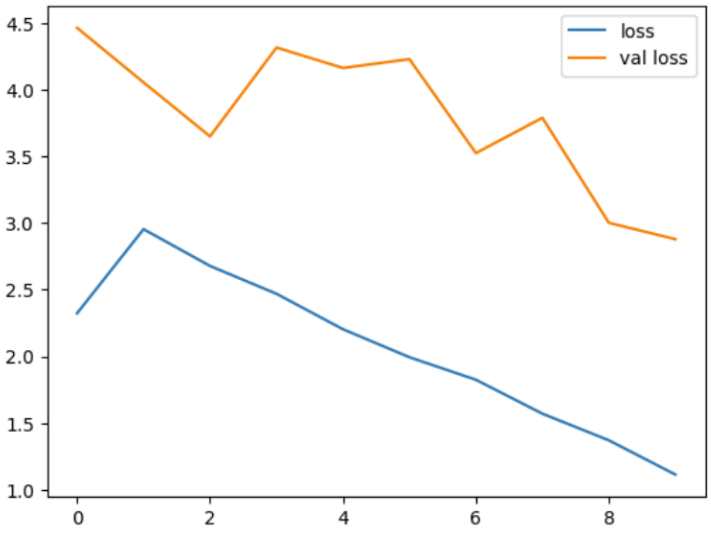

Figure 9: Loss Curve Analysis

6.5.3 Best Epoch and Summary Metrics

The epoch with the **lowest validation loss** was programmatically extracted in order to determine the ideal training point. For every organ, the corresponding accuracy was noted:

- **Best Epoch:** Determined automatically with `np.argmin(val_loss)`
- **Best Validation Accuracies:**
  - Bowel: 0.52
  - Extra: 0.87
  - Kidney: 0.92
  - Liver: 0.91 : 0.79
- **Mean Best Accuracy:** 0.80
- **Best Validation Loss:** 2.98

According to these findings, the model works particularly well for structured injuries (liver, kidney, and extra), but bowel and spleen injuries might need more fine-tuning, data augmentation, or improved labeling.

7 Inference

A post-training pipeline was developed to standardize, preprocess, and infer injury labels in order to produce predictions on the test set, which consists of medical images in **DICOM format**. Image preprocessing, model inference, prediction post-processing, and submission generation are the four main steps that make up this stage.

7.1 DICOM to PNG Conversion

The first step was to convert DICOM slices to PNG images because the trained model works with PNG format images.

- The script creates folder paths to access DICOM files for each patient and series and loads the test metadata `(test_series_meta.csv)`.
- To minimize redundancy and computational load, slices from the volume were sampled using a stride of 10.
- DICOM pixel arrays were standardized, rescaled, normalized, and saved as resized PNGs (256x256) in a structured directory using Pydicom and OpenCV

```
data = (data - np.min(data)) / (np.max(data) + 1e-5)
cv2.imwrite(new_path, img)
```

The test data is guaranteed to match the input expectations of the model trained on PNG-formatted data thanks to this preprocessing.

7.2 TensorFlow Dataset for Test Images

To effectively manage the loading of test data during inference:

- A tf.data.The generated PNG paths were used to create the dataset.
- The same preprocessing logic used in the training pipeline was used to decode and resize each image.
- To enable parallelized loading, images were prefetched and batched.

```
ds = tf.data.Dataset.from_tensor_slices(image_paths)
.map(decode_image)
.batch(config.BATCH_SIZE)
```

The accuracy and caliber of PNG image decoding were validated through visualization using `keras_cv.visualization.plot_image_gallery()`.

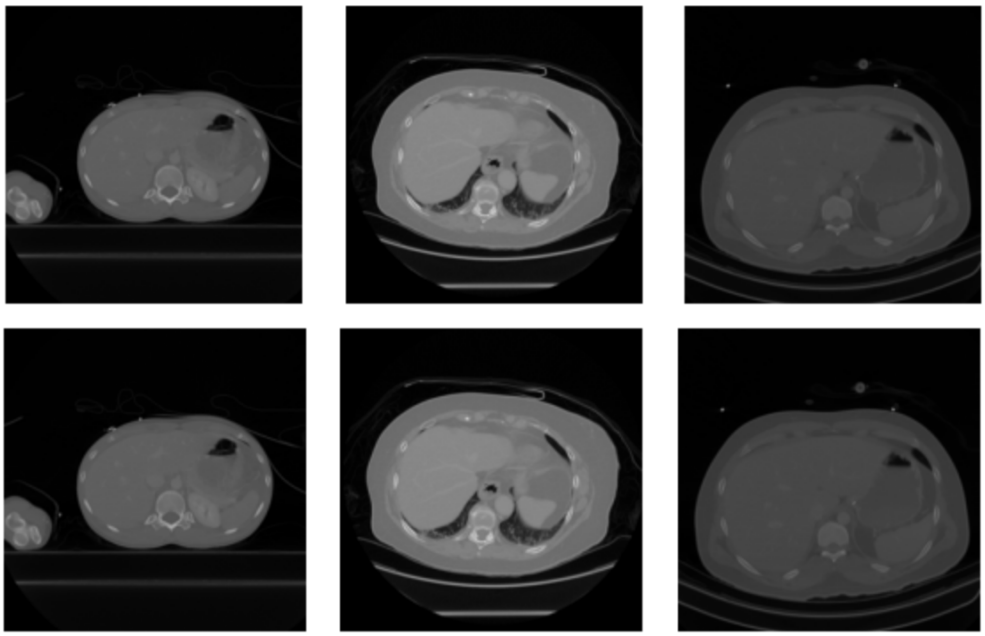

Figure 10: Data Visualization

7.3 Model Inference per Patient

It may be necessary to aggregate the model predictions for each patient's multiple slices.

- The model made predictions on the slices for every distinct `patient_id`.
- To generate a reliable per-patient output, predictions for individual slices were averaged and max-pooled.

- Predictions were reorganized into the necessary format by a post-processing function:
  - Bowel and extravasation injuries were represented using binary splits (injury, healthy).
  - Liver, kidney, and spleen used 3-class probability vectors (healthy, low, high).

```
pred = np.mean(pred.reshape(1, len(patient_paths), 11), axis=0)
proc_pred = post_proc(pred)
```

This strategy maintained consistency with training logic while guaranteeing that predictions took into account several anatomical slices

7.4 Final Submission

Upon acquiring predictions for every patient:

- The prediction array's shape was confirmed to correspond with the anticipated format: (`N_patients`, 13 classes).
- Along with `patient_id`, the predictions were added to a fresh DataFrame.
- The columns were rearranged according to a specified index order in order to conform to the official RSNA submission format.
- The output was saved as submission.csv after being combined with the example submission template.

```
pred_df.insert(0, "patient_id", patient_ids)
sub_df = sub_df.merge(pred_df, on="patient_id", how="left")
sub_df.to_csv("submission.csv", index=False)
```

A probability distribution for each injury type and severity class for each test patient is included in the final output file. As an example:

```
patient_id bowel_injury extravasation_injury kidney_healthy ...
48843 0.083825 0.213047 0.113808 ...
50046 0.017473 0.363355 0.716058 ...
```

8 Conclusion

As part of the RSNA 2023 Abdominal Trauma Detection Challenge, we created a comprehensive deep learning pipeline for identifying and categorizing abdominal injuries using CT scans. Based on a multi-output architecture with a ResNet50 backbone pretrained on ImageNet, the model was optimized to predict multiple binary and multi-class labels that represent the type and severity of injury across five vital abdominal structures: the kidney,

liver, spleen, bowel, and extravasation (internal bleeding). To handle the high-dimensional medical imaging data, the method included effective data loading and batching techniques, image augmentation using KerasCV, and strong data preprocessing using TensorFlow's tf.data pipeline. Inference was carried out on a per-patient basis by aggregating predictions across chosen slices from DICOM volumes that were converted to PNG format, and model performance was tracked using per-organ accuracy metrics. According to the competition's evaluation framework, the system generated results that were ready for submission and showed promise in a number of anatomical categories. The study demonstrates how AI-based tools can help physicians in trauma situations by quickly, automatically, and accurately classifying injuries from imaging data.

8.1 Limitations

Inference was carried out on a per-patient basis by aggregating predictions across chosen slices from DICOM volumes that were converted to PNG format, and model performance was tracked using per-organ accuracy metrics. According to the competition's evaluation framework, the system generated results that were ready for submission and showed promise in a number of anatomical categories. The study demonstrates how AI-based tools can help physicians in trauma situations by quickly, automatically, and accurately classifying injuries from imaging data. Although stride-based sampling during inference reduced computational load, it might have omitted important slices, resulting in gaps in clinical coverage. Finally, misclassifications may have been caused by the nature of multi-label medical annotation and the possible presence of noisy or ambiguous labels in the training set, especially in organs where validation accuracy displayed higher variance, such as the bowel and spleen

8.2 Future Work

In the future, a number of approaches could be investigated to enhance the model's functionality and clinical applicability even more. By incorporating 3D modeling methods like slice-wise attention mechanisms or 3D convolutional neural networks, the system may be able to take advantage of volumetric context and more closely resemble how radiologists interpret CT scans. Compared to conventional natural image pretraining, domain-specific pretraining using extensive unlabeled medical imaging datasets via contrastive or self-supervised learning may produce more useful feature representations. To increase model robustness, data augmentation could be improved with simulated noise, anatomical deformations, or transformations inspired by clinical practice. By offering visual cues in addition to predictions, injury localization techniques like saliency maps or weakly-supervised segmentation that improve interpretability would also encourage clinical adoption. Furthermore, incorporating uncertainty estimation techniques may aid in indicating each prediction's dependability, allowing for more cautious decision-making in situations that are unclear. The model should be tested on separate datasets and in conjunction with radiologists to evaluate its performance in real-world clinical settings and possibly expand its usefulness to complete trauma triage workflows in order to guarantee practical applicability

9 Acknowledgment

The author admits to using ChatGPT and Deepseek, two AI tools, to improve sentence structure and help create basic syntaxes for equations, tables, and images. Nonetheless, the author certifies that all of the analysis, research, and creative contributions included in this report are entirely his own. The Author's GitHub Page has the code for every piece of work that is discussed in this paper. If you are interested in contributing to the repository, please do so.

References

[1] J. Deunk, M. Brink, H. M. Dekker, and et al., "The value of ct in blunt abdominal trauma: a systematic review," *European Radiology*, vol. 17, no. 10, pp. 2452–2460, 2007.

[2] J. A. Soto, S. W. Anderson, and et al., "Ct of blunt abdominal trauma: Evaluation of 123 hemodynamically stable patients," *Radiology*, vol. 237, no. 1, pp. 104–113, 2005.

[3] P. Rajpurkar, J. Irvin, K. Zhu, and et al., "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017. [Online]. Available: https://arxiv.org/abs/1711.05225

[4] D. W. Kim, J. Y. Kim, and S. H. Park, "Ai-assisted liver injury detection in abdominal ct using deep learning," *Journal of Digital Imaging*, vol. 33, pp. 1085–1094, 2020.

[5] S. Chilamkurthy, R. Ghosh, S. Tanamala, and et al., "Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study," *The Lancet*, vol. 392, no. 10162, pp. 2388–2396, 2018.

[6] J. Olczak, N. Fahlberg, A. Maki, and et al., "Artificial intelligence for analyzing orthopedic trauma radiographs," *Acta Orthopaedica*, vol. 88, no. 6, pp. 581–586, 2017.

[7] R. Bosc, J. A. Goff, and M. C. Godoy, "Computed tomography imaging in abdominal trauma," *Radiologic Clinics of North America*, vol. 57, no. 4, pp. 733–748, 2019.

[8] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[10] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[11] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[12] A. Ghosh, "Eda: Train.csv," https://www.kaggle.com/code/aritrag/eda-train-csv, 2023, accessed: 07-Mar-2025.

[13] Kaggle, "Rsna 2023 abdominal trauma detection competition," 2023, accessed: 07-Mar-2025. [Online]. Available: https://www.kaggle.com/competitions/rsna-2023-abdominal-trauma-detection

[14] KerasCV, "Image augmentation api," 2024, accessed: 07-Mar-2025. [Online]. Available: https://keras.io/api/keras_cv/layers

[15] TensorFlow, "tf.data: Build tensorflow input pipelines," 2024, accessed: 07-Mar-2025. [Online]. Available: https://www.tensorflow.org/guide/data

[16] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021, pp. 10 096–10 106. [Online]. Available: https://proceedings.mlr.press/v139/tan21a.html

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2017. [Online]. Available: https://arxiv.org/abs/1608.03983