Predicting Organ Injuries from Blunt Force Trauma using CT Scans and Machine Learning

Problem: We need faster diagnoses of organ injuries

A recent study has estimated the incident rate for trauma patients with solid organ injuries at 5.4 per 100,000. Of these patients, 12.5% die within 30 days (Larsen, J. W., et al., 2022). It is difficult to determine if a patient with abdominal trauma has solid organ injuries that need immediate surgery. "Computed tomography (CT) has become an indispensable tool in evaluating patients with suspected abdominal injuries due to its ability to provide detailed cross-sectional images of the abdomen" (Errol Colak, et al., 2023). However, the process is time-consuming and requires experts to analyze the scans.

Deliverables

- A model that takes a CT scan and produces probabilities for the injury status of a patient's liver.
- A model that can segment out the liver from a CT scan.
- A report and slide deck outlining the process, outcomes, and recommendations.

Stakeholders

Doctors and radiologists who use abdominal CT scans of patients and need to diagnose the injury status of their liver, especially when deciding if surgery is needed.

Proposed Solution

A two-stage deep learning model that takes a CT scan as an input and predicts the injury status of an organ. The predictions will be probabilities for each of three labels: no injury, mild injury, and severe injury. A severe injury is one that requires immediate surgery (Errol Colak, et al., 2023). The first stage of the model predicts the segmentation mask of the organ. The second stage of the model predicts the injury status of the organ.

In this study, I will focus on the liver, as it is the most common injury found in abdominal trauma patients (Larsen, J. W., et al., 2022).

Recommendations

1. Use the predictions to triage patients. If there are many CT scans but few experts to analyze those scans, we could use this model to find the scans that require the most urgent need of analysis.
2. Use bounding box predictions to expedite analysis. We could automatically create visual reports, which contain only the liver, leading to analysis that is faster and easier.

Methodology Overview

Model Description

- **Model Inputs**: For each patient, the model needs a series of DICOM files created from a CT scan.

- **Model Outputs**: A probability for each possible ground truth of the patient's liver condition. For example, {healthy: 0.3, mild_injury: 0.3, severe_injury: 0.4}.
- Metric: Sample-weighted mean of the binary log loss for each class ('liver_healthy', 'liver_low', 'liver_high') with sample weights (1, 2, 4) respectively.

Data

We have de-identified CT scans for 3,147 patients with injury labels concerning the liver, spleen, kidney, bowels, and extravasation (internal bleeding). 1564 patients have 2 scans, and the rest have 1. We have about 200 scans with segmentation masks for the liver, kidney, spleen, and bowels.

The data is curated from 23 research institutions from 14 countries (Errol Colak, et al., 2023). The data is sourced from the Kaggle competition "RSNA 2023 Abdominal Trauma Detection."

Data Files Overview

train.csv - Target labels for the train set. Note that patients labeled healthy may still have other medical issues, such as cancer or broken bones, that don't happen to be covered by the competition labels.

- **patient_id** - A unique ID code for each patient.
- **[bowel/extravasation]_[healthy/injury]** - The two injury types with binary targets.
- **[kidney/liver/spleen]_[healthy/low/high]** - The three injury types with three target levels.
- **any_injury** - Whether the patient had any injury at all.

[train/test]_images/[patient_id]/[series_id]/[image_instance_number].dcm - The CT scan data, in DICOM format.

[train/test]_series_meta.csv - Each patient may have been scanned once or twice. Each scan contains a series of images.

- **patient_id** - A unique ID code for each patient.
- **series_id** - A unique ID code for each scan.
- **aortic_hu** - The volume of the aorta in Hounsfield units. This acts as a reliable proxy for when the scan was. For a multiphasic CT scan, the higher value indicates the late arterial phase.
- **incomplete_organ** - True if one or more organs wasn't fully covered by the scan. This label is only provided for the train set.

segmentations/ - Model-generated pixel-level annotations of the relevant organs and some major bones for a subset of the scans in the training set. This data is provided in the NIfTI file format.

[train/test]_dicom_tags.parquet - DICOM tags from every image, extracted with Pydicom. Provided for convenience.

Unused Data That Could Be Useful for Improving the Model Further

image_level_labels.csv - Train only. Identifies specific images that contain either bowel or extravasation injuries.

- **patient_id** - A unique ID code for each patient.
- **series_id** - A unique ID code for each scan.
- **instance_number** - The image number within the scan. The lowest instance number for many series is above zero as the original scans were cropped to the abdomen.
- **injury_name** - The type of injury visible in the frame.

Exploratory Data Analysis

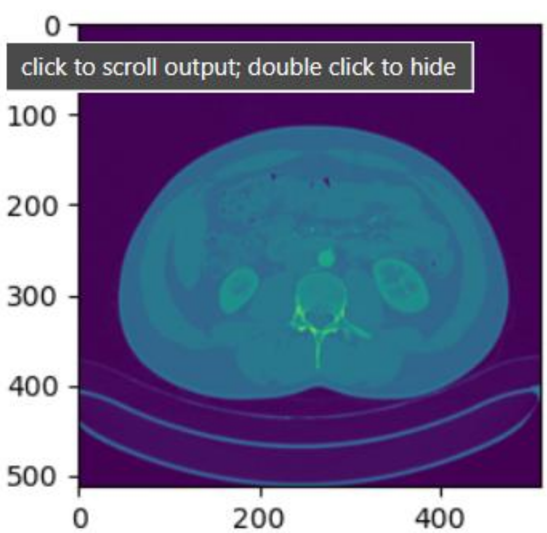
Found that CT scans have between 30 and 500 total DICOM files. Each file has an array 512x512. Each array value represents the density of the tissue in Hounsfield units, ranging 0-4000.

Preprocessing for Segmentation Model

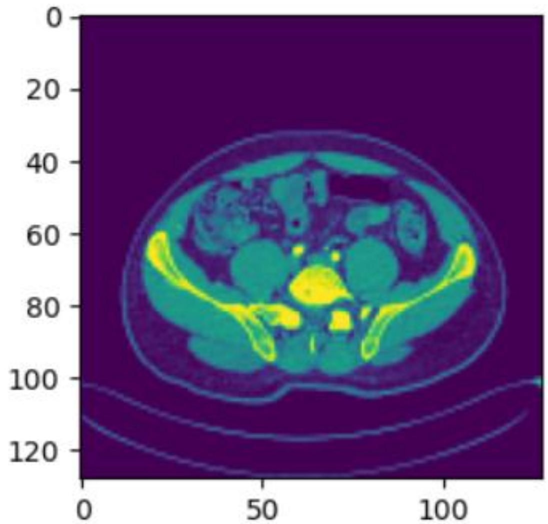
For the segmentation model, we perform the following preprocessing steps:

1. **We ensure that the images are ordered by the z-coordinate in ascending order.**
2. **Then, we resize the 3D image to a uniform size of 128x128x128.**
3. **We also clip all values to be within the radiologist annotated window of relevant values. This way we can see the structures better**

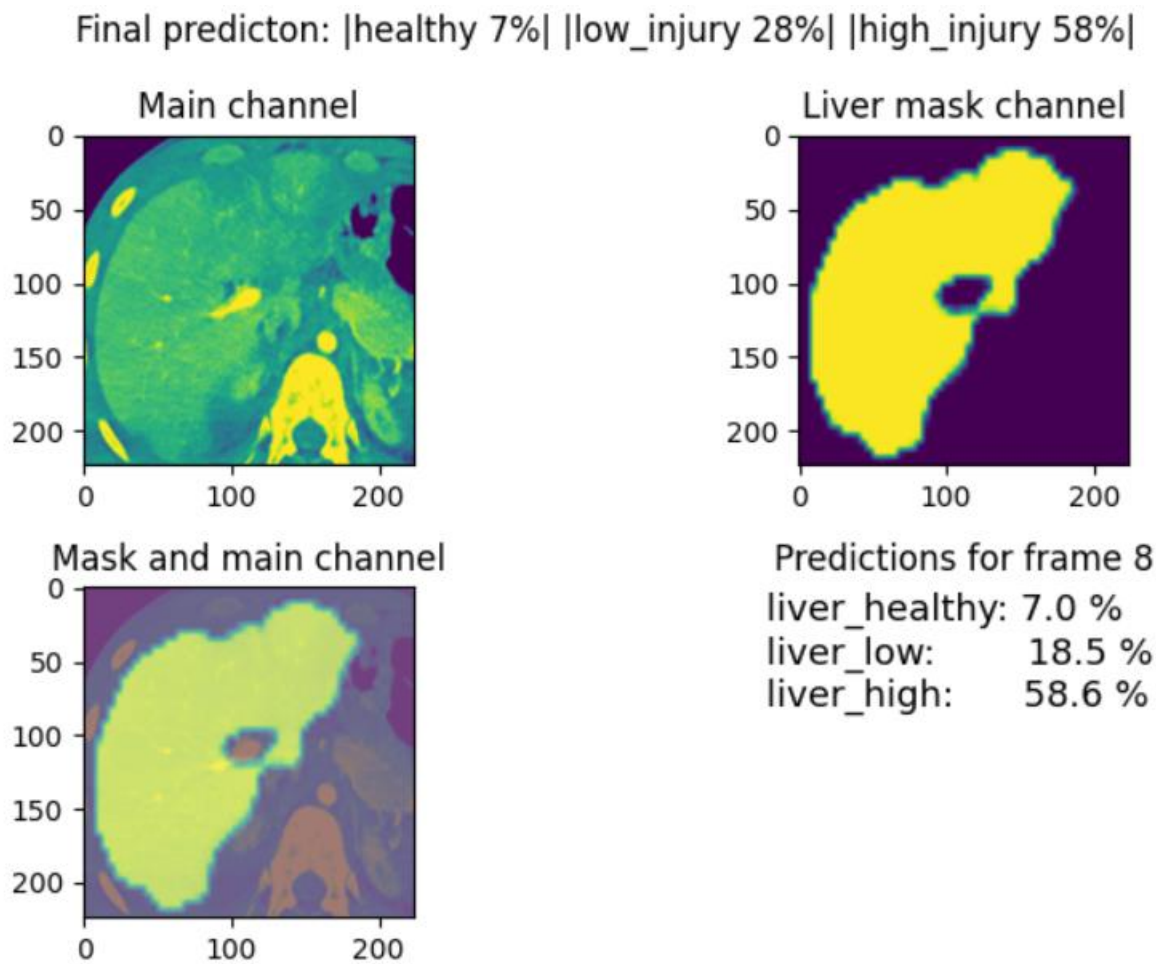
Raw Input



Processed input ready for segmentation (just one slice in image)



The output of the segmentation model is a mask over the predicted liver. We use that mask to identify the bounding cube that surrounds the liver. Then consider only this cube, and take 15 slices evenly spaced. For each slice, we take the image at slice as well as the two preceding and the two following. We stack these five images with a the predicted mask for each image and create a single "image" with 6 channels. So we now have 15 images, each with 6 channels, as inputs to our model. The model makes predictions for each of the 15 images. See the full slideshow here: [rsna23-prediction-visualization | Kaggle].



Postprocessing

For each CT scan, we have 15 probability predictions from the previous step. For our final predictions, we take the min across the healthy predictions and the max across the two injury predictions.

Segmentation model description

Input: 128x128x128 3D image

Output: 5x128x128x128 tensor. 5 masks for liver, kidney, spleen, left kidney, right kidney

Architecture: Encoder/Decoder segmentation model.

- Encoder: Timm model with resnet18 backbone to get global features
- Segmentation decoder: Takes global features and upsamples to create segmentation output
- Segmentation head: takes the features from the decoder and outputs the masks

This model is also converted from a 2-D to a 3-D segmentation model.

Stage 2 classification model

Input: 6 x 224 x 224

Output: 1 x 3 probabilities

Architecture: Encoder – LSTM – Sequential head

- Timm encoder with efficient_net backbone to extract features
- 2 Layer bi-directional LSTM
- Sequential head: 2 linear layers with batchnorm and dropout

Data augmentations

```
transforms_train = albumentations.Compose([
    albumentations.Resize(image_size, image_size),
    albumentations.HorizontalFlip(p=0.5),
    albumentations.VerticalFlip(p=0.5),
    albumentations.Transpose(p=0.5),
    albumentations.RandomBrightness(limit=0.1, p=0.7),
    albumentations.ShiftScaleRotate(shift_limit=0.3, scale_limit=0.3, rota
te_limit=45, border_mode=4, p=0.7),

    albumentations.OneOf([
        albumentations.MotionBlur(blur_limit=3),
        albumentations.MedianBlur(blur_limit=3),
        albumentations.GaussianBlur(blur_limit=3),
        albumentations.GaussNoise(var_limit=(3.0, 9.0)),
    ], p=0.5),
    albumentations.OneOf([
        albumentations.OpticalDistortion(distort_limit=1.),
        albumentations.GridDistortion(num_steps=5, distort_limit=1.),
    ], p=0.5),

    albumentations.Cutout(max_h_size=int(image_size * 0.5), max_w_size=int
(image_size * 0.5), num_holes=1, p=0.5),
])

transforms_valid = albumentations.Compose([
    albumentations.Resize(image_size, image_size),
])
```

Model experiments

<u>Model | Metric scores (</u>

Baseline: .336

Base + remove mixup: .356

Base + remove distortion: .355

Base with backbone tf_efficientnet_b4_ns_jft_in1: .321

Base with backbone tf_efficientnet_b5_ap: .3177

Test set score: .3176 with model " Base with backbone tf_efficientnet_b5_ap"

The baseline framework for my solution is based on work done on a similar problem: predicting spinal fractures from CT scans. This work was done by Qishen Ha, with a write-up and links to code at https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/discussion/362607. This baseline is applied to all organs (liver, kidney, spleen) and achieves 79th place out of 1126 teams in the Kaggle competition, which is the source of data used for this project (Errol Colak, et al., 2023). The write-up for my solution to the competition can be found at https://www.kaggle.com/competitions/rsna-2023-abdominal-trauma-detection/discussion/450279.

References

Errol Colak, Hui-Ming Lin, Robyn Ball, Melissa Davis, Adam Flanders, Sabeena Jalal, Kirti Magudia, Brett Marinelli, Savvas Nicolaou, Luciano Prevedello, Jeff Rudie, George Shih, Maryam Vazirabad, John Mongan. (2023). RSNA 2023 Abdominal Trauma Detection. Kaggle. https://kaggle.com/competitions/rsna-2023-abdominal-trauma-detection

Larsen, J. W., Søreide, K., Søreide, J. A., Tjosevik, K., Kvaløy, J. T., & Thorsen, K. (2022). Epidemiology of abdominal trauma: An age- and sex-adjusted incidence analysis with mortality patterns. *Injury, 53*(10), 3130–3138. https://doi.org/10.1016/j.injury.2022.06.020