**Abdominal Multi-Organ Segmentation (AMOS) Project Report**

**Matan Ziv**

**Yohann Pardes**

**1. Introduction**

The objective of this project is to develop a **machine learning pipeline** for the **segmentation of abdominal organs** in **CT/MRI scans** using the **AMOS dataset**. The task involves predicting **segmentation masks** that delimit multiple organs based on image features extracted from the scans.

This problem is formulated as a **multi-class segmentation task**, where each pixel in the image is assigned to one of several predefined **organ classes** or classified as **background.**

The goal is to accurately distinguish and label each organ as accurately as possible.

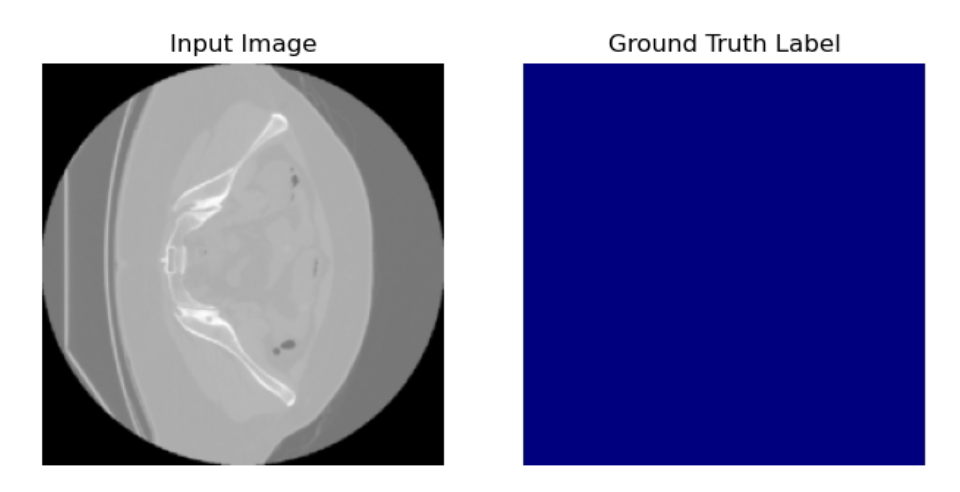
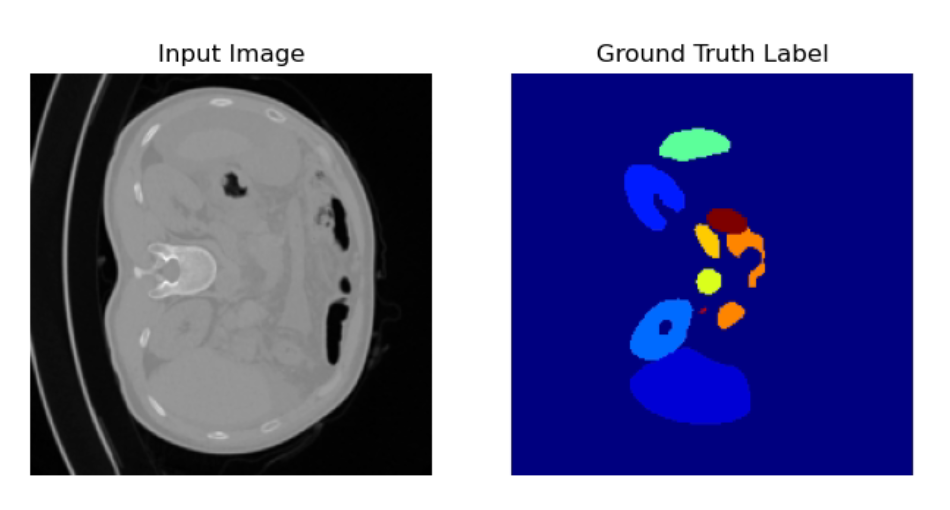
**2. The AMOS dataset**

The dataset comprises **medical imaging data** collected from **multiple patients**, where each scan provides a volumetric representation of the **abdominal region**. The images originate from **CT scans**, capturing a diverse range of anatomical variations, patient demographics, and imaging conditions.

Each pixel within the images represents an **intensity value**, serving as a **feature** used for model training. The corresponding **labels** are structured as **segmentation masks**, where each pixel is assigned to a specific organ class or marked as background (non-organ structures such as flesh, bones, or empty space). These masks are manually or semi-automatically annotated by medical experts to ensure **high-quality ground truth data** for supervised learning.

The dataset includes scans from **patients of varying ages, body compositions, and clinical backgrounds**, providing a **realistic and heterogeneous** representation of abdominal anatomy. This diversity ensures that the segmentation model generalizes well across different patient populations and medical imaging conditions.

The dataset consists of **16 segmentation classes**, including **background, spleen, right kidney, left kidney, gallbladder, esophagus, liver, stomach, aorta, postcava (inferior vena cava), pancreas, right adrenal gland, left adrenal gland, duodenum, bladder, and prostate/uterus.**



**2.1 Dataset filtering and preparation**

The dataset originally comprised full CT scans stored in the **.nii.gz** format. To facilitate model training and evaluation, the scans were decomposed into individual 2D slices, each representing a standalone image.

To ensure consistency across the dataset, all images were **filtered and resized** to a uniform resolution of **640×640 pixels**. This preprocessing step aimed to standardize the proportions of anatomical structures within the images while maintaining a fixed input size for the models.

For model evaluation, the dataset was **divided into three subsets**:

* **Training set**: 70% (240 scans)
* **Validation set**: 15% (60 scans)
* **Test set**: 15% (60 scans)

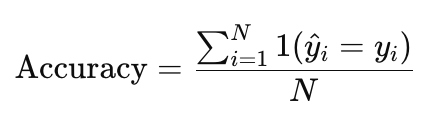
The filtering and resizing processes were applied **after** the dataset split, ensuring that the proportion of data allocated to each subset remained unchanged.

In summary, the final dataset consists of three sets of preprocessed images derived from different CT scans, each paired with their corresponding pixel-wise segmentation labels.

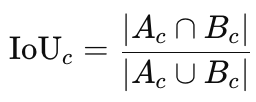
**3. Evaluation Metrics**

To assess the performance of the segmentation models, we employ **pixel-wise accuracy** and **Mean Intersection over Union** (Mean IoU), both of which are commonly used in semantic segmentation tasks.

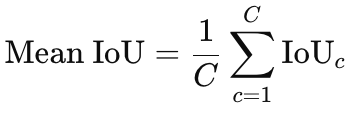
**Pixel-wise accuracy** measures the proportion of correctly classified pixels across the entire dataset. It is defined as:

****

where N is the total number of pixels, y\_i is the **ground truth** label for pixel i, y^i is the predicted label, and 1(⋅) is the indicator function that returns 1 if the prediction is correct and 0 otherwise. While accuracy provides an overall correctness measure, it can be misleading in **class-imbalanced** datasets, as in our current dataset, where the **background** class dominates.

**Mean Intersection over Union (Mean IoU)** evaluates the overlap between predicted and ground truth segmentations for each class and averages the results across all classes. It is defined as:

where Ac​ represents the set of pixels predicted as class c, and Bc​ represents the **set of ground truth pixels** belonging to class c. The Mean IoU is then computed as:



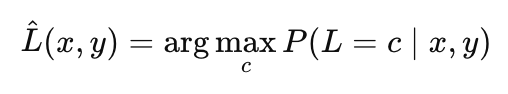
where CC is the number of classes.

Mean IoU provides a more robust evaluation than accuracy, as it considers both **false positives and false negatives**, ensuring that performance is assessed fairly across all classes, including underrepresented organs.

**4.** **Baseline**

The baseline model is implemented as a **probabilistic classifier**, constructed by analyzing the class distributions at **each pixel position** within the **training set**. Specifically, for each pixel location (x, y), the occurrences of each class across all training images are counted to estimate the probability distribution of labels at that position.

During inference, the model assigns each pixel in the test set to the **most probable class** at the corresponding pixel position in the training set. Formally, for each pixel (x, y), the predicted label is given by:



where:

* (x,y) is the predicted label at pixel (x,y),
* c represents possible class labels,
* P(L=c∣x,y) denotes the estimated probability of class c occurring at pixel position (x,y) in the training set.

**4.1** **Expected Results**

Given that the majority of pixels across all images are labeled as **background (zero)**, it was anticipated that the baseline model would classify most pixels as **background**. This expectation arises from the class imbalance in the dataset, where organ pixels represent only a small fraction of the total image content.

In fact ~95% of the pixels are labelled background.

**4.2 Actual Results Analysis**

| **Accuracy (Pixel wise)** | **95.42 %** |
| --- | --- |
| **Mean Iou (over the different labels)** | **5.96%** |

The baseline model classified **ALL** pixels as **background**, exceeding the initial expectation. Instead of merely predicting **background** as the dominant class, the model failed to identify any organ pixels, leading to an extreme case of class imbalance in the predictions.

We can also see that 0.0596 is the result of (0.9542 + 0 + … + 0)/16 = 0.00596 wich is **exactly** predicting always the background class

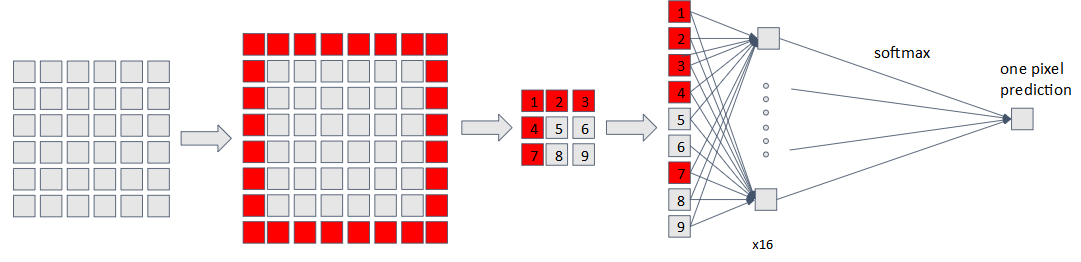
The extreme bias towards the **background** class can be attributed to the nature of the dataset. Even in regions where flesh-filled areas were present, most of the training scans contained **irrelevant structures**, such as bones and non-organ soft tissues, which were also labeled as **background**.

As a result, the most frequent class across nearly all pixel positions remained **background**, reinforcing the model’s tendency to assign this label universally.

This outcome highlights a major limitation of the **probabilistic baseline approach**, demonstrating that a more sophisticated model with spatial awareness and learned feature representations is necessary for effective organ segmentation.

**5. Logistic Regression Attempts**

A logistic regression model was implemented to predict segmentation masks at the pixel level:

1. **Initial Attempt:** A pixel-by-pixel logistic regression was tested. Each pixel’s intensity served as the input feature, and the corresponding organ class was the label. Results:
   1. Extremely poor IoU and precision for organ classes.
   2. Unable to capture spatial relationships between pixels.
2. **Second Attempt:** A 3×3 patch was used to extract localized spatial features before logistic regression. The image was padded (1, 1) padding, and a 3×3 convolution was performed around each pixel. The resulting convolution values were flattened into a vector and fed into the logistic regression model. While this improved IoU slightly, the results were still unsatisfactory due to the lack of global spatial awareness.
3. **Final Attempt:** A 5×5 patch was used, further improving performance marginally. The image was similarly padded (2, 2) padding, and a 5×5 convolution was performed around each pixel, followed by flattening the values into a vector for the model. However, IoU and other metrics indicated that logistic regression lacked the complexity to handle this problem effectively. More than that the performance of the model slightly decreases.

**5.1** **Expected Results**

The regression model was expected to outperform the baseline by capturing localized spatial dependencies using patches. The incorporation of 3×3 and 5×5 patches was anticipated to improve IoU and segmentation performance for non-background classes, albeit with potential limitations in global spatial awareness.

**5.2 Actual Results Analysis**

| **Metric\Attempt** | **Pixel wise prediction** | **3x3 patch** | **5x5 patch** |
| --- | --- | --- | --- |
| **Accuracy** | 85.3% | 85.8% | 85.5% |
| **Mean IoU** | 6.54% | 6.63% | 6.60% |

The results indicate that the **regression-based model** performs **comparably to the baseline**, showing **no significant improvement** in segmentation accuracy. Despite the added complexity of the regression approach, it fails to achieve **statistical relevance** over the baseline model. In fact, the model exhibits a slight decrease in accuracy while not demonstrating any meaningful gains in segmentation performance.

This suggests that the regression model may not be well-suited for this particular segmentation task, potentially due to its inability to effectively capture the spatial dependencies and structured nature of organ boundaries.

**6. Fully Connected Neural Network**

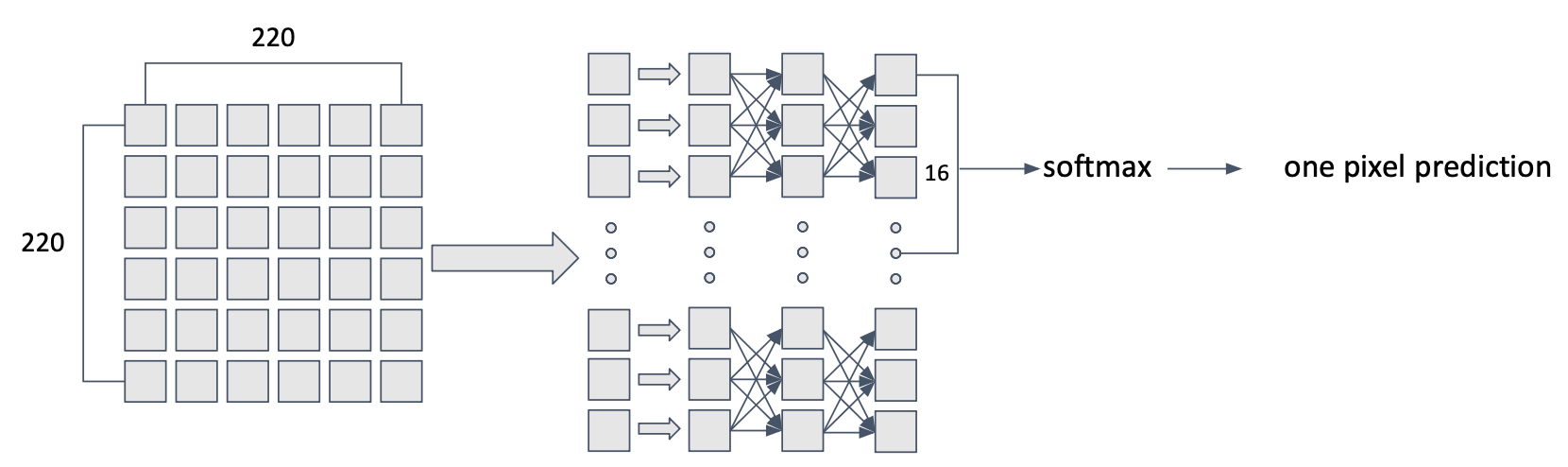
For the fully connected Neural Network approaches (using only dense layers) the data has been downsampled down to 220X220.

The network architecture is as follow:

The input layer dimension is 220x220 = 48,400 input neurons.

The output dimension is 220x220x16 = 774,400 output neurons.

For the actual prediction each 16 output neurons (the number of possible labels) is then feeded into a softmax unit to output the actual predicted class as seen in the below diagram.

****

**6.1** **Expected Results**

We anticipate that the model will achieve a **reasonable level of performance**, allowing it to recognize **obvious patterns** associated with organ structures. However, due to its limited ability to capture **spatial dependencies**, the model is also expected to exhibit **over-sensitivity to insignificant patterns** present in surrounding **soft tissues**.

Since the model lacks a strong encoding of **spatial context**, it may misinterpret certain **flesh-like regions** as organs, leading to an increased number of **false positives**. This behavior stems from the model’s reliance on **local pixel intensities** rather than leveraging broader anatomical structures, which are crucial for accurate organ segmentation.

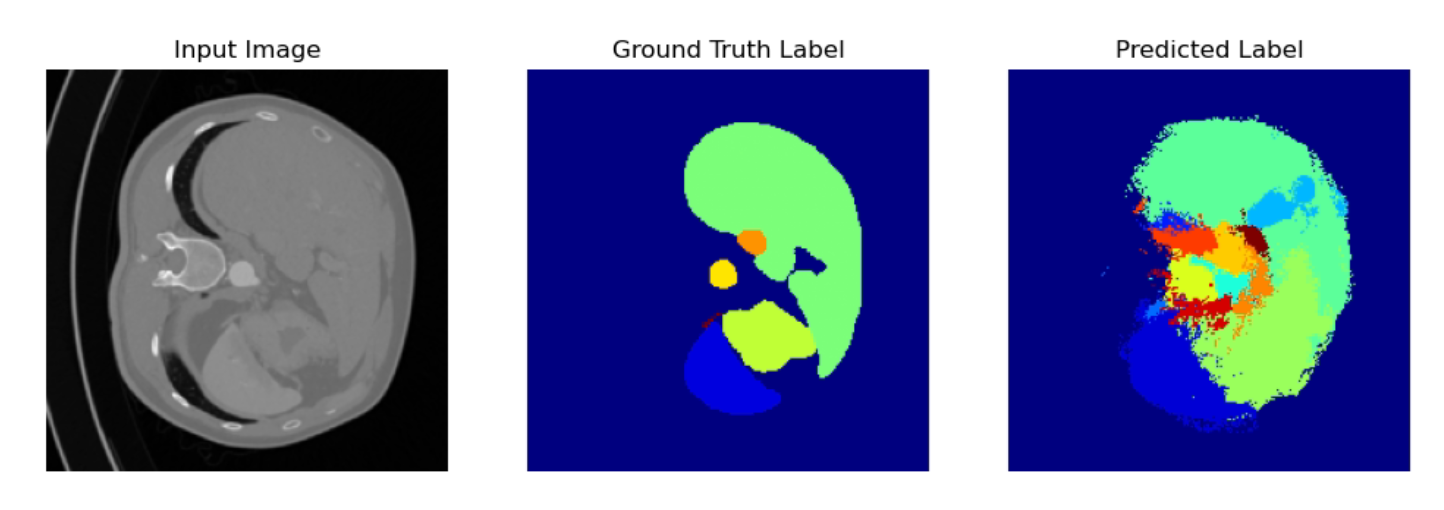
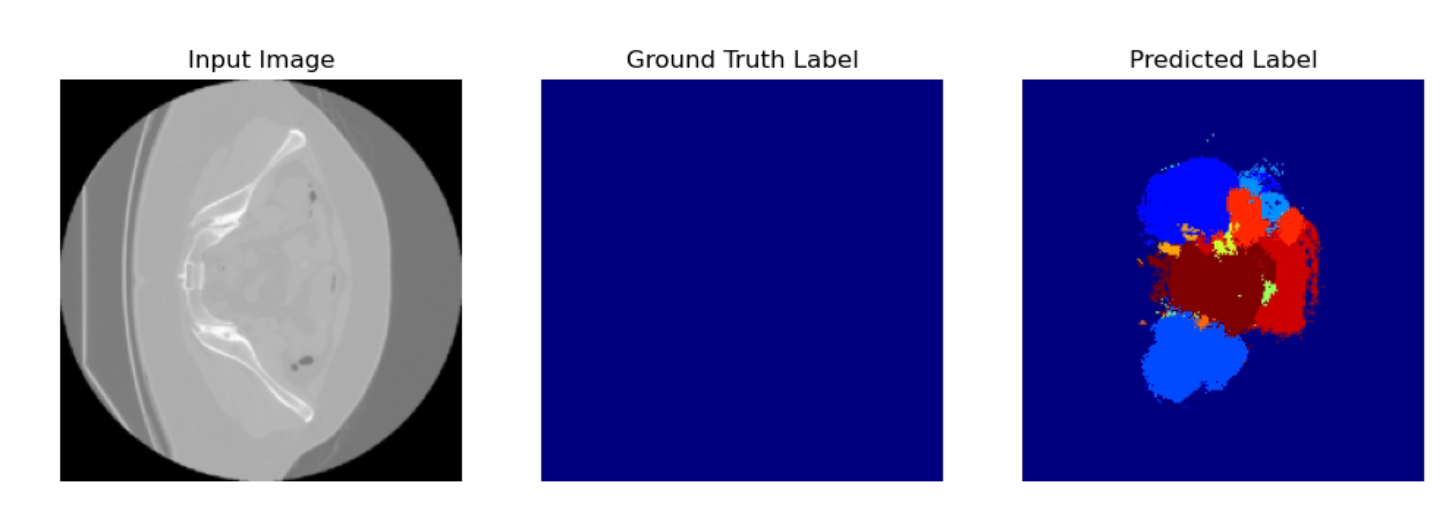
**6.2 Actual Results Analysis**

| **Accuracy (Pixel wise)** | **50.5%** |
| --- | --- |
| **Mean Iou (over the different labels)** | **11.2%** |

The obtained results align with the initial hypothesis; however, the model’s performance is **worse than expected**. While we anticipated the model to recognize **obvious patterns** and misclassify **insignificant structures** in soft tissues as organs, the degree of misclassification and overall segmentation accuracy fell below expectations.

This suggests that the model struggles more than anticipated in differentiating between organs and surrounding tissues, likely due to its **inability to effectively encode spatial relationships**. The increased error rate indicates that the model may be overly sensitive to pixel-wise variations while failing to capture **coherent anatomical structures**, leading to a degradation in performance beyond the initially predicted limitations.

For example in figure 1 the model highly overreacts to some irrelevant patterns and labels them as organs. However as can be seen in the second figure the model is able to mostly find the present organs and shape.

****

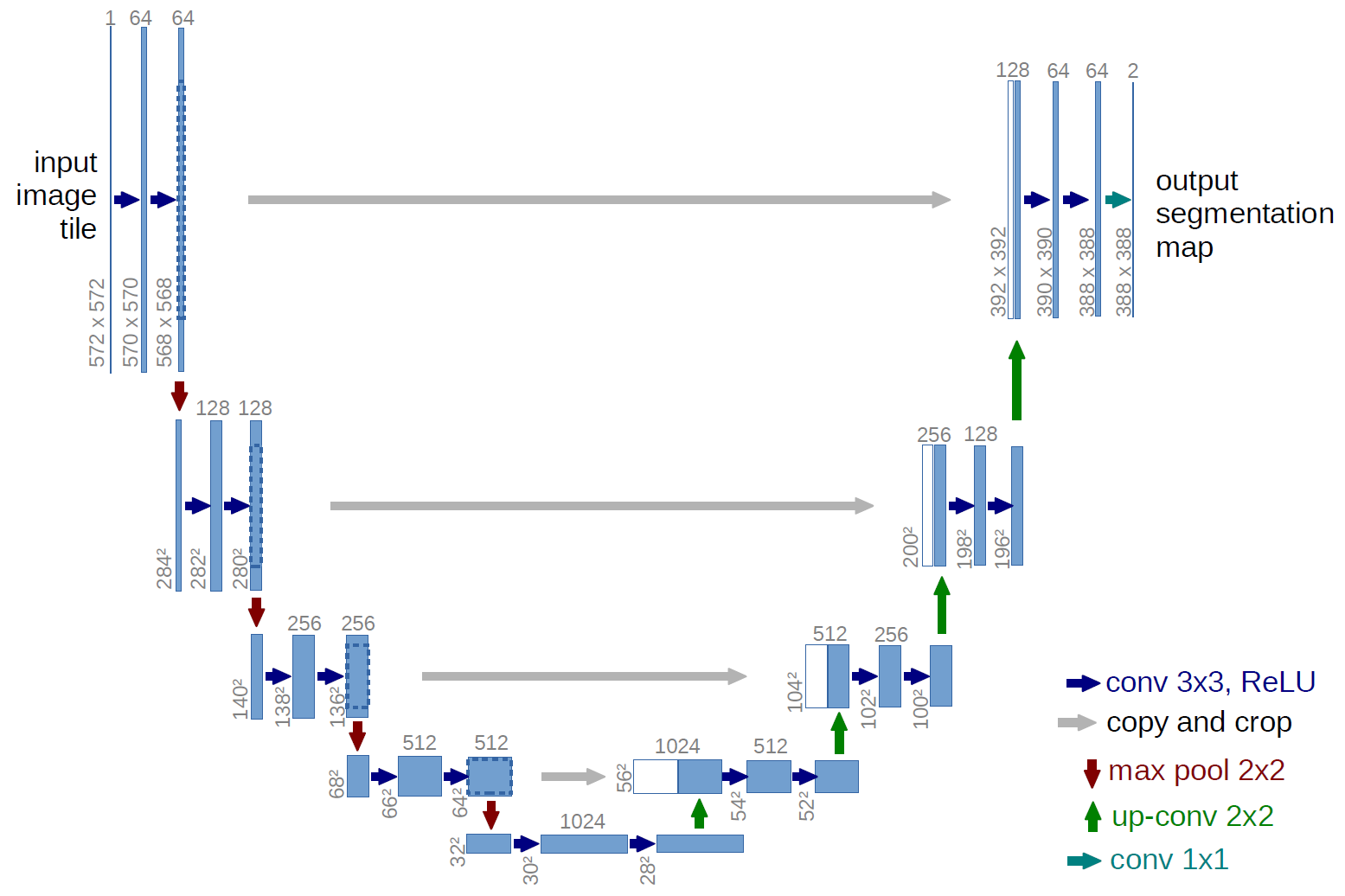
**7. Advanced Network Architecture: U-Net**

A U-Net architecture is used for segmentation, representing a significant advancement over previous approaches. The initial attempts and optimizations are as follows:

#### **7.1 Model Architecture**

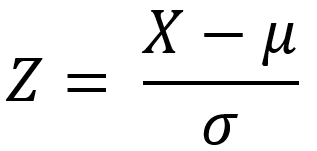
The U-Net model consists of a contracting path (encoder) and an expansive path (decoder):

* **Encoder:** Extracts spatial and hierarchical features using convolutional layers and max-pooling operations. The encoder progressively reduces spatial dimensions while increasing the number of feature maps.
* **Decoder:** Reconstructs the spatial resolution of the segmentation mask by performing upsampling and concatenating encoder features through skip connections. This design allows the model to combine high-level features with localized details.
* **Skip Connections:** Enable the network to recover spatial details lost during the down-sampling process.

****

#### **7.2 Input Normalization**

First input images are normalized to a range of [0, 1] by dividing by 255 (the maximum pixel intensity). Finally the pixel values undergo Z-score normalization using the formula below:



The purpose of this normalisation is to have a means of 0 and a std of 1 to normalise across different images.

This helps with convergence of the model toward a lower loss.

#### **7.3 Loss Function Choice**

In the first attempt, we used normal categorical cross-entropy loss, which resulted in poor performance due to the dominance of the background class. Subsequently, a weighted categorical cross-entropy loss function was implemented to address class imbalance. The weights assigned to each class were inversely proportional to their frequency in the training dataset. This encouraged the model to pay greater attention to underrepresented organ classes while mitigating the dominance of the background class.

#### **7.4 Optimization**

Hyperparameter optimization was performed using Optuna to systematically search for the best configurations. The following parameters were optimized:

* Learning rate: Step size, how much to update the weights of the model according to the gradient.
* Weight decay: Encourage small weights in the model to reduce overfitting.
* Batch size: How many examples to look at once to take a learning step.
* Dropout: Applying a random mask to a layer to prevent overfitting
* Loss Weight Factor: A custom parameter that further decrease the weight of the background class in the loss function

Optuna’s search strategy allowed us to identify an optimal combination of parameters that enhanced the model’s performance and reduced overfitting issues.

**Results:**

* Significant improvement in IoU, Accuracy.
* The model generalized well across validation and test sets.

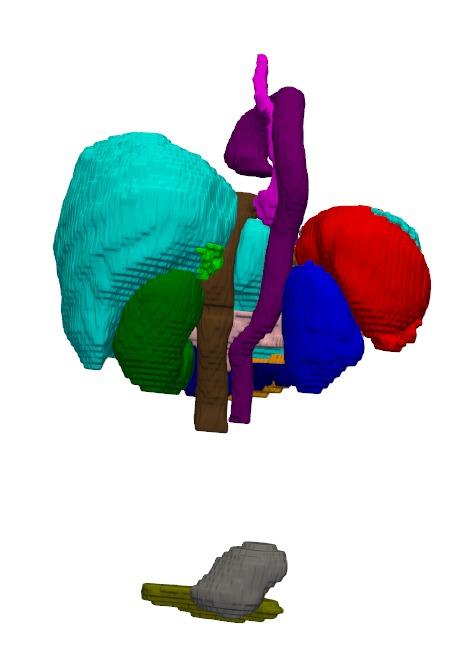
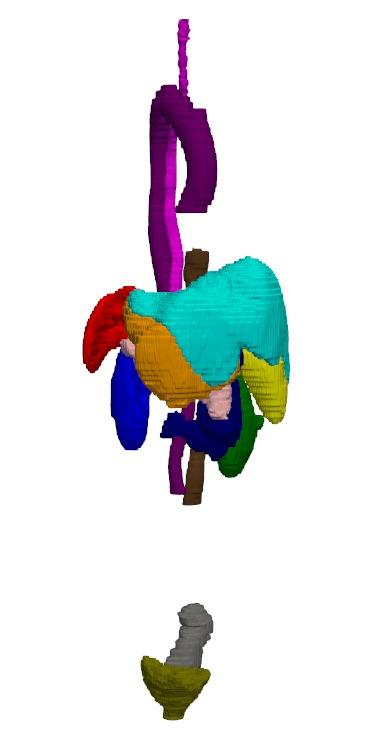
| **Metric\**  **Attempt** | **Manually selected parameters** | **With Optuna parameters**  **No weighted loss** | **With Optuna parameters**  **And weighted loss** |
| --- | --- | --- | --- |
| **Accuracy** | 94.3% | 90.2% | 83.5% |
| **Mean IoU** | 32.3% | 36.6% | 52.6% |

**7. Comparison of Models**

| **Model** | **Accuracy** | **IoU** | **Overall performance** |
| --- | --- | --- | --- |
| Baseline | 95.42% | 5.96% | Poor |
| Logistic regression | 85.8% | 6.63% | Poor |
| Fully connected | 50.5% | 11.2% | Better but Poor |
| U-net | 83.5% | 52.6% | Good in all scale |

**8. 3D Reconstruction From Image Segmentation**

Given a set of labeled CT slices in order we can reconstruct the organs in 3D as shown below:



(This technique is used in hospital today to make a 3D heart models before surgery)

This report highlights the progression from simple models to advanced architectures, demonstrating the critical role of advanced neural networks like U-Net for medical image segmentation.