Deep Learning Final Project

FinalGroup 9

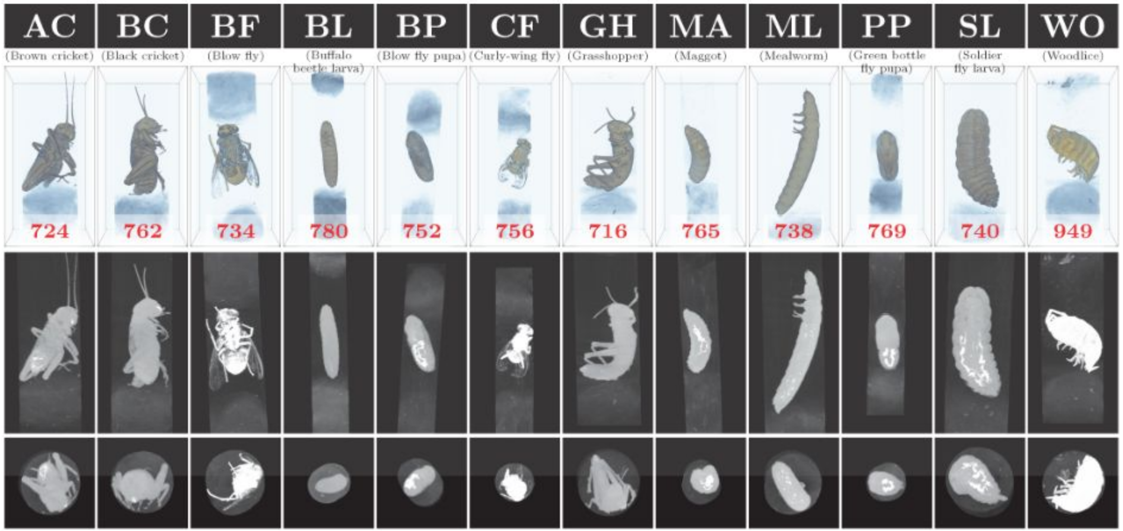
Carroccetto Edoardo, Detogni Federico,
Mameli Dario, Masip Llopis Àngel, Russo Michele

PART 1

Table of contents:

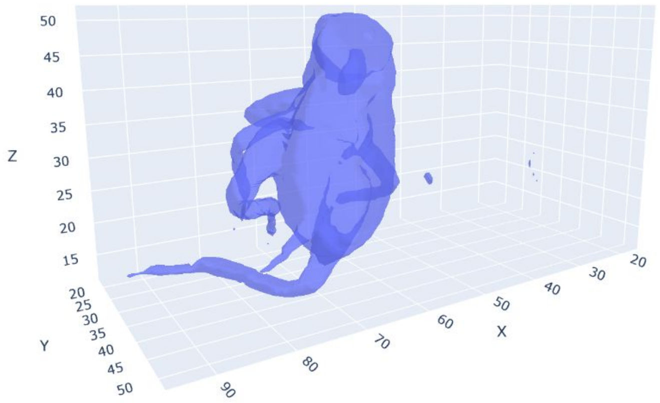
Kaggle competition: BugNIST2024

Domain adaptation for detecting and classifying complex-shaped volumetric objects



Dataset structure

- **Training data:**
  - single-volume scans of a specific class of bug.
- **Validation data:**
  - volume scans of bug mixtures.
- **Test data:**
  - volume scans of bug mixtures.



Dataset splitting

KAGGLE SPLITTING
Training dataset
Validation dataset
Test dataset

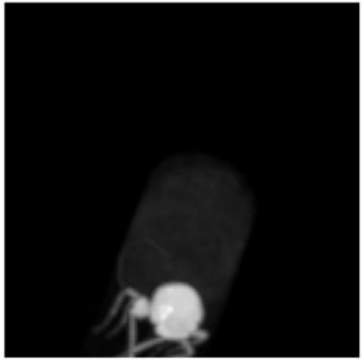
OUR SPLITTING
Training dataset
Validation dataset
Test dataset
Submission

Preprocessing: Data Augmentation

Original image

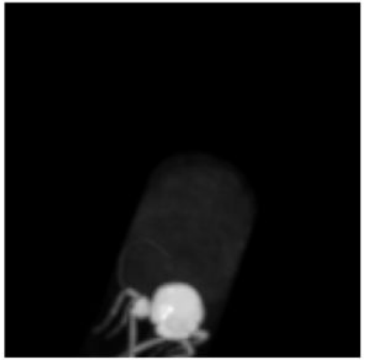
INPUT IMAGE

- Intensity scaling
- Border padding
- Random rotation
- Random translation
- Random flip
- Random zoom



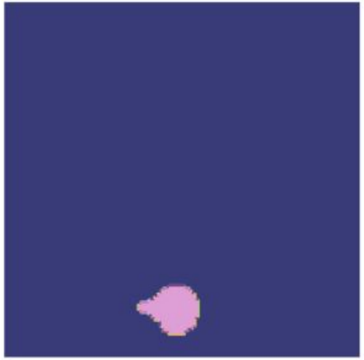
Augmented image

Preprocessing: Data Augmentation

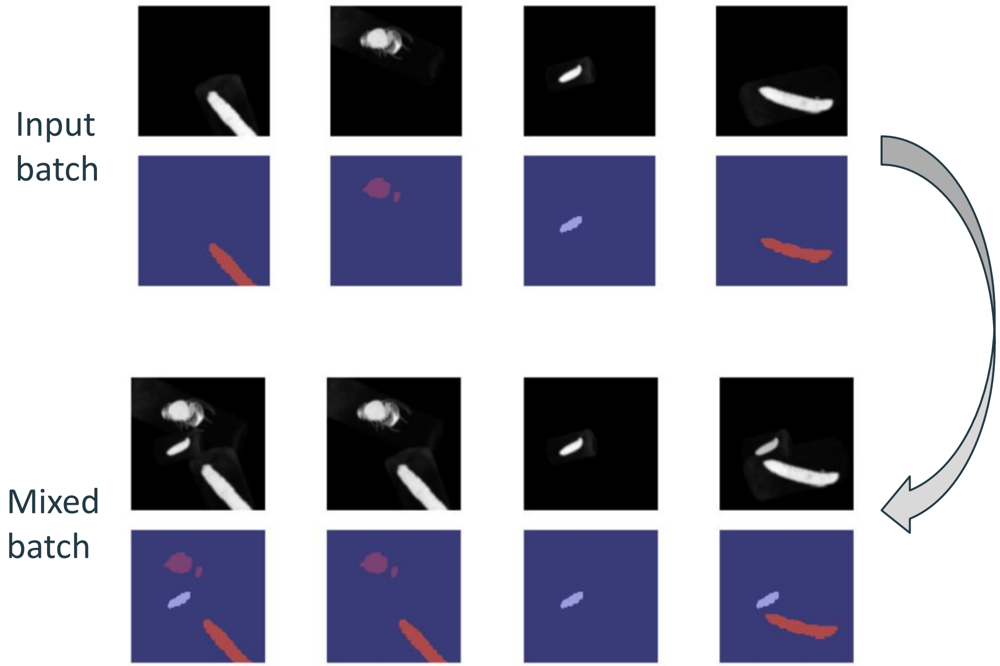

Augmented image

MASK CREATION

- Gaussian smoothing
- Binary thresholding
- Largest connected component


Mask of the image

Preprocessing: Data Augmentation



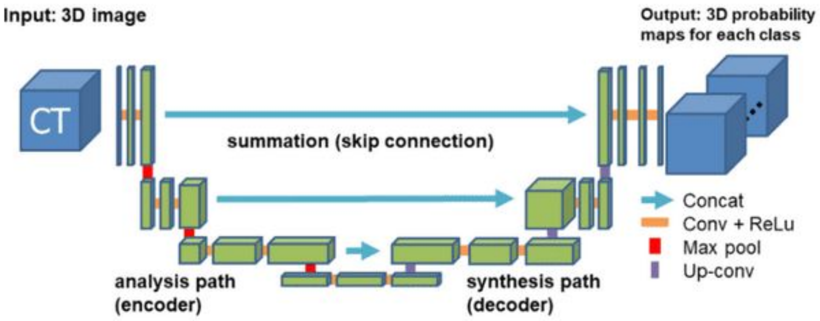
BATCH MIXING

- For each element in the batch
- Take a random number in [0, batch_size - 1] of other elements
- Overlap the scans if the volumes of the bugs do not collide

=> DOMAIN SHIFT ADAPTATION

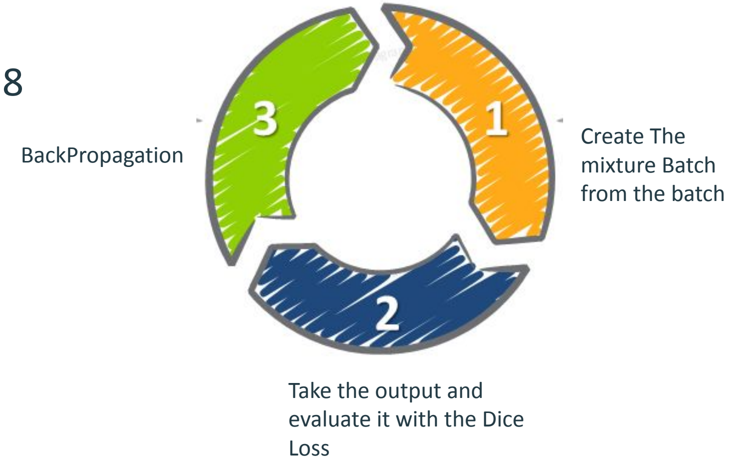
Why 3D-UNET architecture?

- 3D UNET's encoder-decoder architecture allows it to capture **comprehensive contextual information**, including features that span the entire volume.
- **Preserves spatial relationships** across the three dimensions, enabling it to capture the contextual information necessary for precise segmentation.
- **Robust performance in complex environments** (skip connections and hierarchical feature extraction)



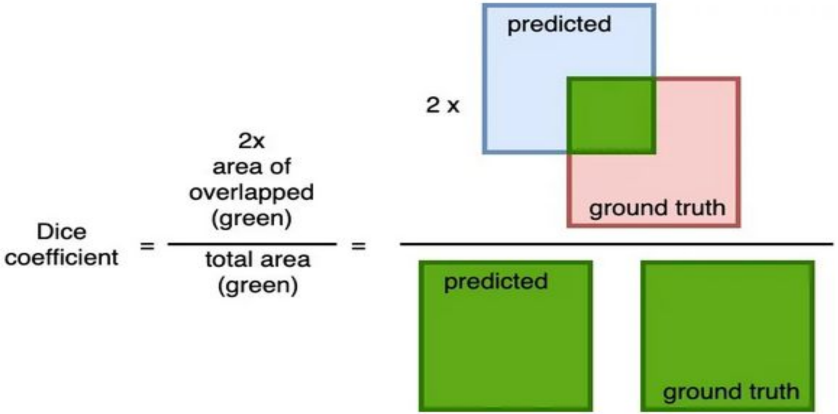
Training loop

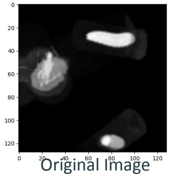
- We **Fine-Tuned** a pretrained Unet for spleen segmentation, trained on Medical Segmentation Decathlon Challenge 2018
- **Training Batch size:** 12 and **Validation batch size:** 8
- **Adam optimizer**
- **Early stopping**
- **Loss function: DiceLoss** (implemented in the package monai)
- For each iteration take the batch of images and create a **mixed batch** of images

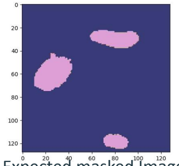


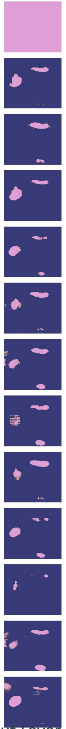
Loss Function

During the training the DiceLoss function was used :



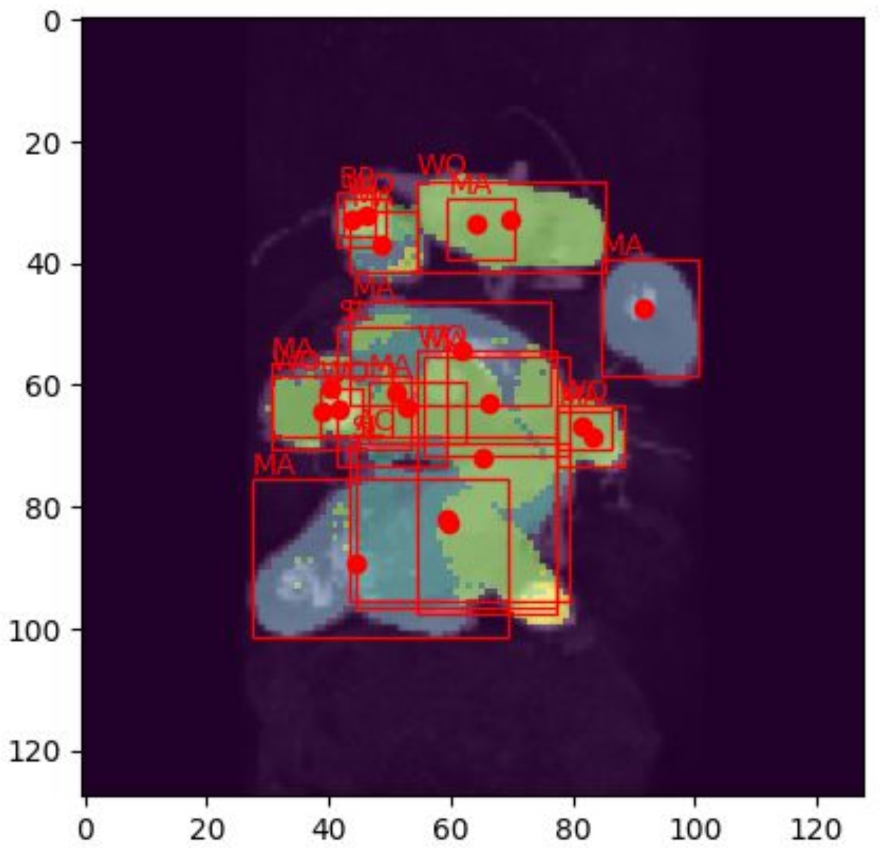

Original Image


Expected masked Image


Output Layer Image

Model prediction

- Prediction of the **Labels**
- Prediction of the **Center Points**
- Prediction of the **Bounding Boxes**



Model evaluation

Leaderboard on kaggle:

#	Team	Score
1		0.41031
2	Our CODE	0.21669
3		0.19953
4		0.15273
5	BASELINE	0.11102
6		0.09992
7		0.09776
8		0.06727

Conclusions

- The **BugNIST2024**: a challenging task for our project
- **Preprocessing**: data augmentation
- **Model**: 3D UNET architecture to segment and classify the bugs
- **Further improvements**: Swin UNETR (with sufficient computational resources)

Potential applications

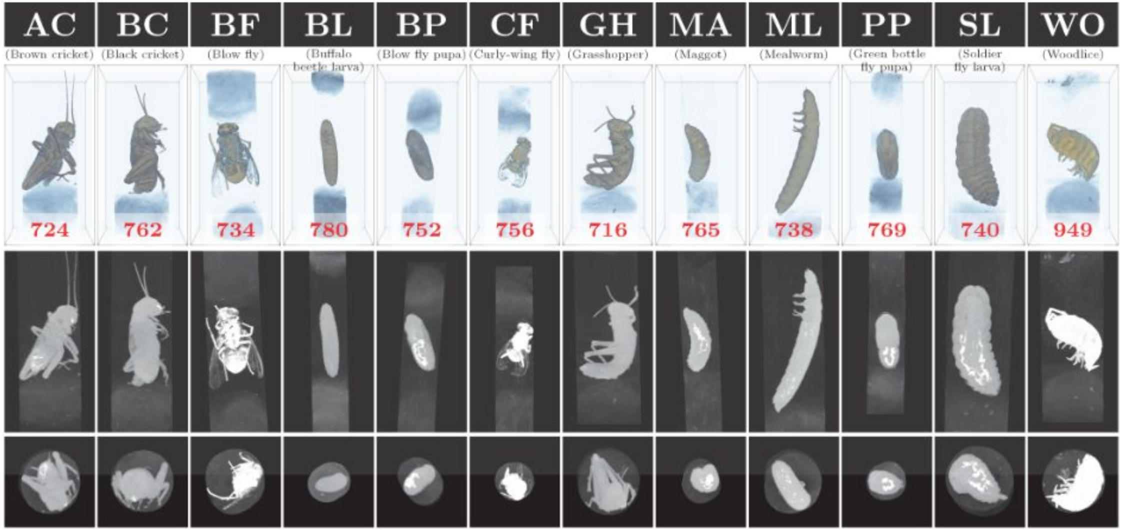
- **Agriculture:** Facilitating pest detection and monitoring in crops
- **Entomology Research:** Supporting studies on insect behavior, population dynamics, and biodiversity conservation
- **Industrial Quality Control**: Inspecting and classifying products with complex shapes and structures

PART 2

Table of contents:

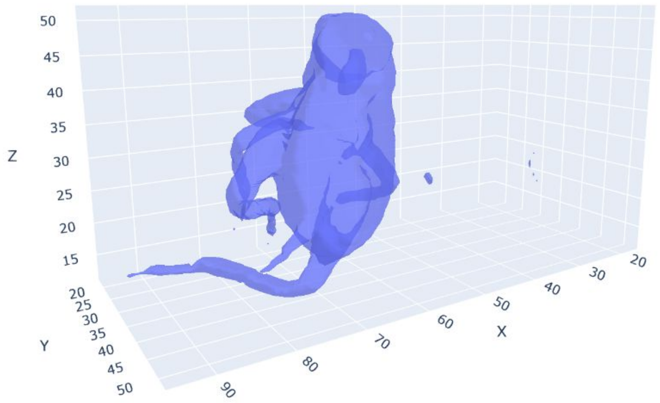
Kaggle competition: BugNIST2024

Domain adaptation for detecting and classifying complex-shaped volumetric objects



Dataset structure

- **Training data:**
  - single-volume scans of a specific class of bug.
- **Validation data:**
  - volume scans of bug mixtures.
- **Test data:**
  - volume scans of bug mixtures.



Dataset structure*

- **Training data:**
  - 12 folders, each containing single-volume scans of a specific class of bug.
  - The dimensions are (128, 64, 64) corresponding to the z,x,y-axis.
- **Validation data:**
  - It contains 78 volumes of bug mixtures.
  - The dimensions are (128, 92, 92) corresponding to the z,x,y-axis.
- **Test data:**
  - It contains 310 volumes of bug mixtures.
  - The dimensions are (128, 92, 92) corresponding to the z,x,y-axis.

Dataset splitting

KAGGLE SPLITTING
Training dataset
Validation dataset
Test dataset

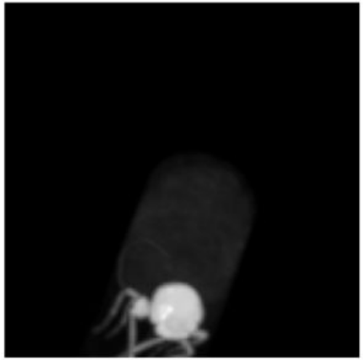
OUR SPLITTING
Training dataset
Validation dataset
Test dataset
Submission

Preprocessing: Data Augmentation

Original image

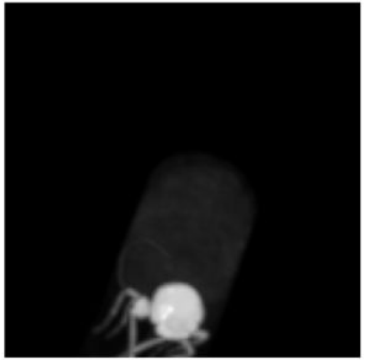
INPUT IMAGE

- Intensity scaling
- Border padding
- Random rotation
- Random translation
- Random flip
- Random zoom



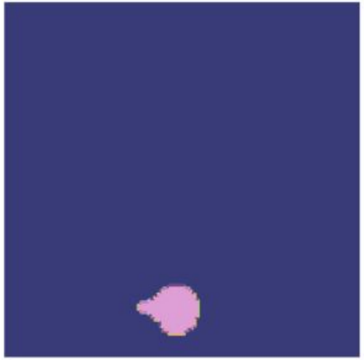
Augmented image

Preprocessing: Data Augmentation


Augmented image

MASK CREATION

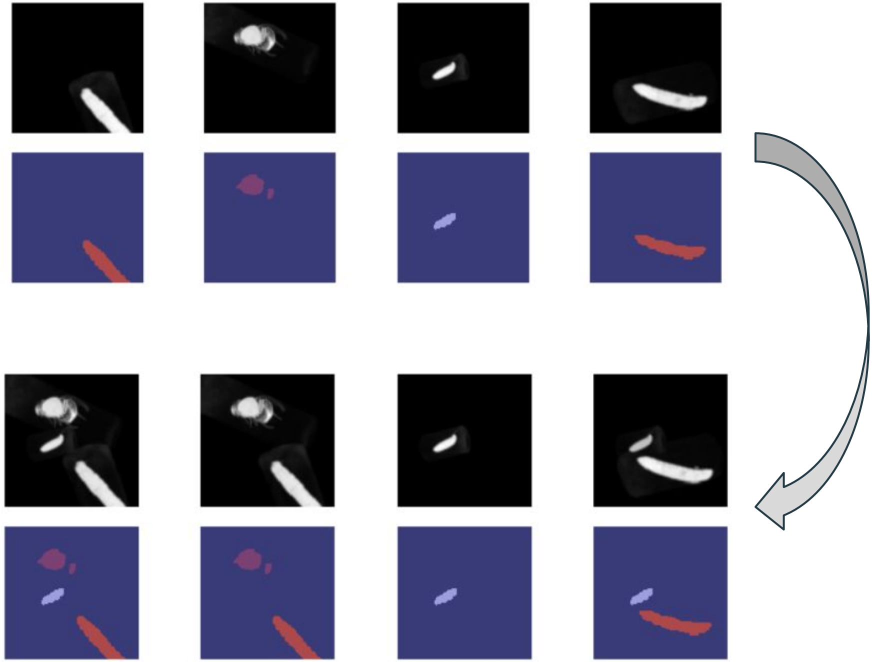
- Gaussian smoothing
- Binary thresholding
- Largest connected component


Mask of the image

Preprocessing: Data Augmentation

Input batch

Mixed batch



BATCH MIXING

- For each element in the batch
- Take a random number in [0, batch_size - 1] of other elements
- Overlap the scans if the volumes of the bugs do not collide

=> DOMAIN SHIFT ADAPTATION

Preprocessing: Data Augmentation*

- **Data augmentation**: increase the model's generalization capability by introducing noise and mixing images together to better match validation and test sets.
- **Inputs:** volumes, represented as images in this presentation for simplicity**.**

  [batch_size, 1, 128, 128, 128] Tensors

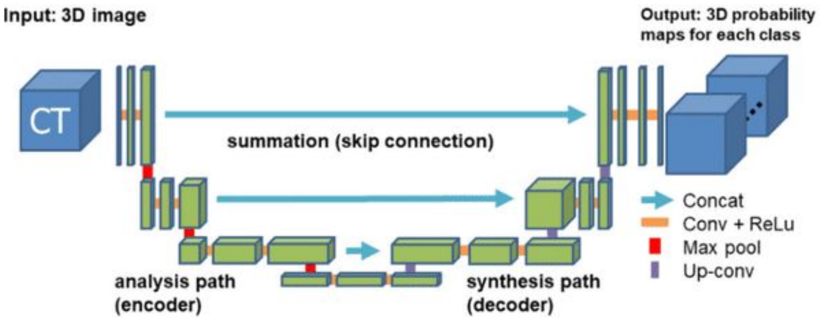
- **Masks:** not given, computed using the information in the volumes scans.

  [batch_size, 128, 128, 128] Tensors. Each element encodes a value in [0,num_classes]

- These operations are performed at **run-time** during training.

Why 3D-UNET architecture?

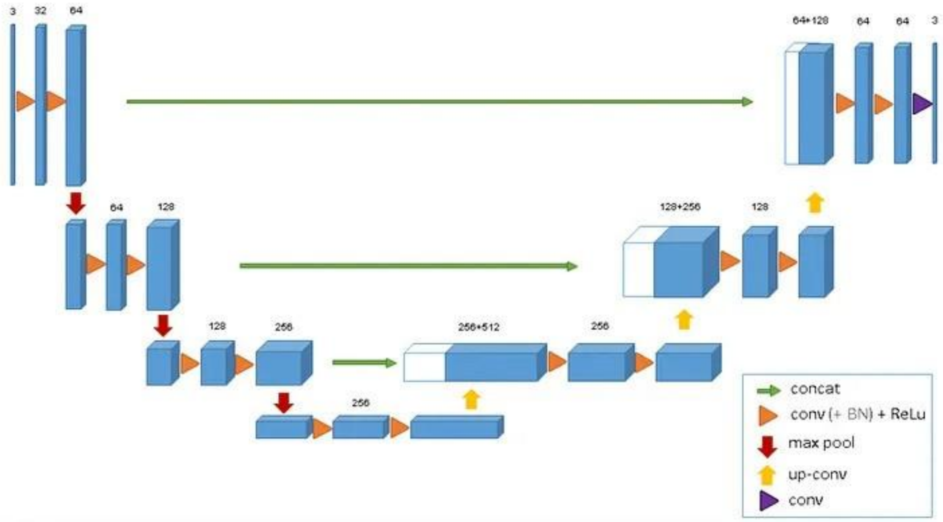
- 3D UNET's encoder-decoder architecture allows it to capture **comprehensive contextual information**, including features that span the entire volume.
- **Preserves spatial relationships** across the three dimensions, enabling it to capture the contextual information necessary for precise segmentation.
- **Robust performance in complex environments** (skip connections and hierarchical feature extraction)



Model architecture*

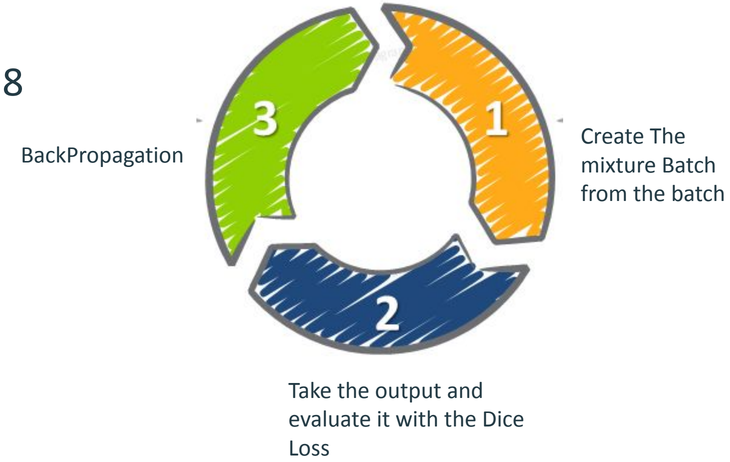
Key components:

- **U-shaped** architecture similar to 2D UNET but tailored for volumetric data.
- **Encoder:** Convolutional layers downsample volumetric data, capturing hierarchical features.
- **Decoder:** Upsamples feature maps to generate high-resolution segmentation masks.
- **Skip connections:** Preserve fine-grained details and spatial context across depths



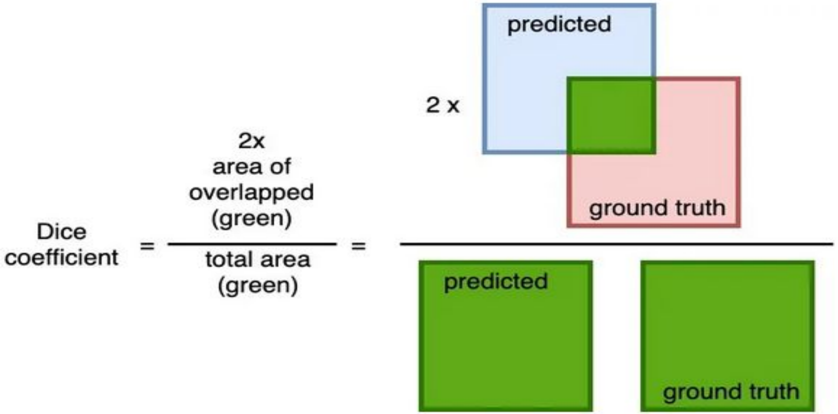
Training loop

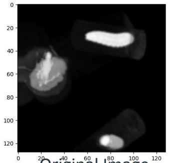
- We **Fine-Tuned** a pretrained Unet for spleen segmentation, trained on Medical Segmentation Decathlon Challenge 2018
- **Training Batch size:** 12 and **Validation batch size:** 8
- **Adam optimizer**
- **Early stopping**
- **Loss function: DiceLoss** (implemented in the package monai)
- For each iteration take the batch of images and create a **mixed batch** of images



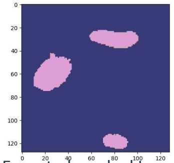
Loss Function

During the training the DiceLoss function was used :

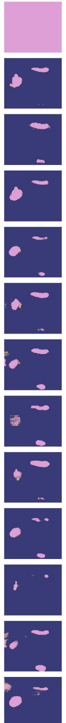




Original Image



Expected masked Image



Output Layer Image

Model evaluation*

Our Intern Evaluation Metrics:

- F1 Score
- F1 Class Score
- Average difference between the predicted center and the real center

```
F1 SCORE OF CLASSES:
Class sl F1 score: 0.00000050
Class bc F1 score: 0.02941225
Class ma F1 score: 0.00000050
Class gh F1 score: 0.12857186
Class ac F1 score: 0.11594247
Class bp F1 score: 0.00000050
Class bf F1 score: 0.48780513
Class cf F1 score: 0.02439073
Class bl F1 score: 0.03846202
Class ml F1 score: 0.39130465
Class wo F1 score: 0.12820556
Class pp F1 score: 0.00000050
Mean F1 score across all classes: 0.12228080

Average centerpoint distance 13.74
```

Conclusions

- The **BugNIST2024**: a challenging task for our project
- **Preprocessing**: data augmentation
- **Model**: 3D UNET architecture to segment and classify the bugs
- **Further improvements**: Swin UNETR (with sufficient computational resources)

Potential applications

- **Agriculture:** Facilitating pest detection and monitoring in crops
- **Entomology Research:** Supporting studies on insect behavior, population dynamics, and biodiversity conservation
- **Industrial Quality Control**: Inspecting and classifying products with complex shapes and structures

Timesheet

NAME	HOURS	TOPIC
Carroccetto Edoardo	15	Model architecture, model training and presentation
Detogni Federico	15	Model architecture, model training, presentation
Mameli Dario	30	Data preprocessing, model architecture, model training
Masip LLopis Angel	25	Data preprocessing, model evaluation
Russo Michele	27	Data preprocessing, model architecture