**Data Overview and EDA Insights**

The data used in the project is the PolypGen dataset (v3) and it provides colonoscopy images from multiple medical centers and hospitals. The hospitals only provide negative images that do not have any polyp in the form of sequences of frames. Each center folder (data\_C1, …, data\_C6) contains single images, mostly with at least one polyp, bbox text files with bounding box annotations, the images with the bounding box and the masks. In addition to single images, data centers also have sequence data made up of positive (contains a polyp) and negative (does not contain a polyp) video frames. Sequence data must be handled carefully to ensure that frames from the same sequence are not split across training, validation, and test sets, in order to avoid data leakage.

In data\_C3, the directory contains 457 images while only 393 of the images have corresponding bounding box annotations. This implies that 64 images do not have annotation files and therefore should be excluded from training. We found that in the positive sequence data, sequences 1 and 7 have empty bbox files. Therefore, we will not be using those sequences since the data could be affected by the missing files. We want the proportion of positive and negative images in each set to be about 50-50, so we will have to leave out some of the negative images in order to keep this ratio.

**Image preprocessing and data augmentation**

A mandatory preprocessing step is to resize all images so they can be collated into a single tensor for GPU training. This can be done using the Resize method in PyTorch, which also supports bounding boxes.

In the PolypGen data set, the bounding boxes are given in the XYXY format (described in the [PyTorch docs](https://docs.pytorch.org/vision/main/generated/torchvision.tv_tensors.BoundingBoxFormat.html)), so any further transformations or data augmentation must respect this format. For example, if we want to rotate an image, we cannot simply rotate the bounding box, because we want the bounding boxes to remain in the axis-aligned XYXY format. Instead, to rotate an image, we must also rotate the ground truth polyp pixel mask, extract the new XYXY bounding box coordinates (using extract\_PolypBoxes.py provided with the PolypGen data set), then provide the new bounding box coordinates as the label for the rotated image. Similarly, even though [RandomResizedCrop](https://docs.pytorch.org/vision/main/generated/torchvision.transforms.v2.RandomResizedCrop.html) can transform bounding boxes, it may not produce the appropriate axis-aligned bounding box on its own; we might have to recompute the bounding box coordinates using extract\_PolypBoxes.py on the transformed pixel mask. Instead of using the RandomResizedCrop method, we could instead use the [Random IoU Crop](https://docs.pytorch.org/vision/main/generated/torchvision.transforms.v2.RandomIoUCrop.html) method in PyTorch, as described in [this paper](https://arxiv.org/abs/1512.02325).

Another possible image preprocessing step would be pixel intensity normalization using the [Normalize](https://docs.pytorch.org/vision/main/generated/torchvision.transforms.v2.Normalize.html) method in PyTorch. This transformation does not affect bounding boxes. For any pretrained model that we want to fine-tune, we can normalize our training data with the same mean and standard deviation as was used to train the original model.

**Train-Test Validation Splitting and Overfitting Handling**

There are 1473 annotated non-sequence images from the 6 data centers. We take advantage of a Scikit-learn function to put 1105 images into the training set, 139 images into the validation set, and 229 images into the test set. To test the generalisability of our model, all the data from data center C6 are assigned to the test set.

We distribute the continuous frames manually, so that images from a complete sequence do not go into both the training set and the test set. In addition, we only add part of the negative sequence images (3174/4275) to the training set and the validation so that the fractions of positive images are roughly about 0.5, respectively. We believe this promotes model training. We put all the sequences from data center C6 to the test set and do not impose a fixed percentage of positive images. This reflects the fact that in a typical clinical scenario, most images are going to be negative.

In sum, the training set includes 4179 images, which consists of 2093 positive images and 2086 negative images. The validation set includes 666 images, with 328 images being positive and 338 images being negative. The test set includes all the data from data center C6 with a small fraction of data coming from other centers. This consists of 646 positive images and 1307 negative images. For a detailed decomposition, please refer to the following tables.

To prevent overfitting, we will use data augmentation techniques and supplement training with additional images from the Real Colon dataset.

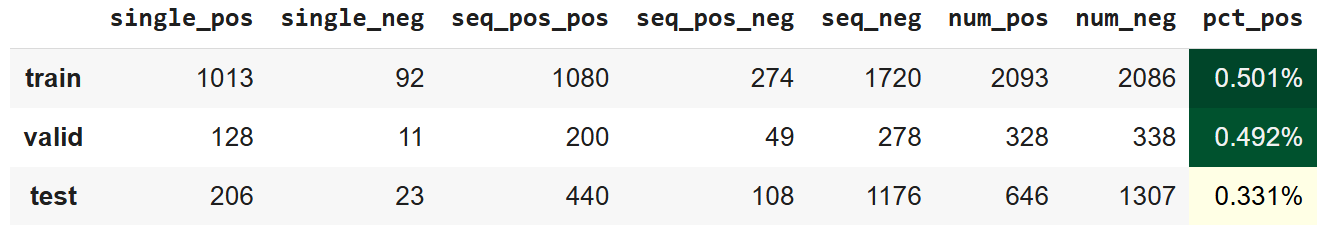


Table 1: number of images in each set

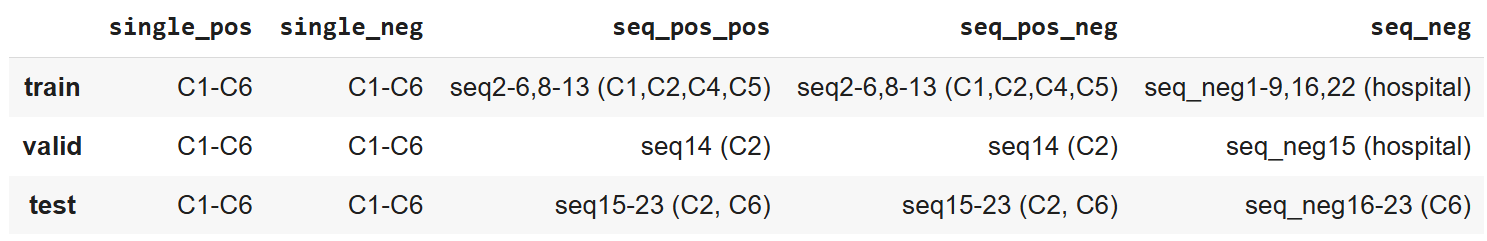


Table 2: Source of the data

8037 frames including single and sequence (positive and negative)

4275 negative: Coming from 4 different hospitals

3762 positive: Coming from 6 centers

C1: 484 = 256 (single) + 228 (seq) – 11

C2: 1166 = 301 (single) + 865 (seq) – 14, 15, 2 of (5, 12, 13)

C3: 457 = 457 (single)

C4: 677 = 227 (single) + 450 (seq) – 1,2,3,4,6,7,8,9,10

C5: 458 = 208 (single) + 250 (seq) – 1 of (5,12,13)

C6: 520 = 432 (seq 16 - 23) + 88 (single) TEST