Weekly Review – [18.04.2023 – 25.04.2023]

Weekly Progress:

- Completed Tasks:
  - Pitch presentation on 25th April
  - Data exploration with number of images per label (train, validation, test), pixel sizes: Notebook
  - Literature study (paper which is linked to the Kaggle dataset source)
  - GitHub Repo created
  - Decide on which Baseline to use (Paper from Kaggle dataset source)
  - Decide Research question and objective: Change parts of Mask R-CNN model to improve mean average precision from the paper. We could switch the backbone CNN model or the mask predictor. We should also try data augmentation first.
  - Define the additional evaluation criteria: Weekly Review.
  - Write weekly schedule based on the Mini-challenge description from Susanne.
  - Everything written in Pitch Presentation.

- In Progress Tasks:
  - None

- Not Started Tasks:
  - Build Skeleton Model and Baseline

Impediments:

- Which model to choose and writing a research question based a hypothesis was harder than expected. We chose a simple approach but while we want to improve Mask R-CNN, we still need to decide on a second model to look at.

Weekly Review:

- Achievements:
  - Know more in detail which models to try out.
  - Understand the theory behind Mask R-CNN
- Areas of Improvement:
  - We need to choose the second model for the Mini-challenge.
- Lessons Learned:
- Action Items for Next Weekly:
  - Skeleton model
  - Overleaf report

Weekly Review – [25.04.2023 – 02.05.2023]

Weekly Progress:

- Completed Tasks:
  - Skeleton model using PyTorch with Mask RCNN- runs on Kaggle GPU P100 with 16GB memory resource – Batch size 12 and uploaded to GitHub (*see Commit dated 02.05.2023 with comment 'Week 2 DoD'*). Training 1 epoch took around 4 mins with evaluation. In total 10 Epochs trained.
  - Overleaf Report documentation started for section – 'Situation', 'Problem statement' and 'Data'.
  - Change the dataloader to include all bounding boxes instead of only one per image.
  - Implement a Dataset class from PyTorch for the task.
    - Calculate Bounding Boxes based on masks
  - Use mixed precision and gradient accumulation for faster training.

- In Progress Tasks:
  - Implement mean average precision (mAP) if possible else we try another metric.
  - Normalization of image pixels.
  - Other preprocessing steps maybe?

- Not Started Tasks:
  - Optional: Choose more metrics.

Impediments:

- Faced problem / error when running locally on Linux with cuda PyTorch (kernel problem). That's why it was tried on Kaggle.
- Mask R-CNN GitHub Repository is old and has 1.8k open issues. Using their library throws errors such as: "AttributeError: module 'keras.engine' has no attribute 'Layer'. Therefore, we consider it rather unusable.
- Faced problem with correct data structure. Data needs to be in correct dictionary form. See: PyTorch Mask R-CNN. The input data for labels needs to be in form of a list of dictionaries (n = batch size) where each dictionary contains the keys "boxes", "labels" and "masks". "Boxes" for example, contains a list of bounding boxes for the image.
- Output of the model is different during model.train() (training) and model.eval() (inference). The output of the model during model.train() returns a dictionary with losses while the output of the model during model.eval() returns the predictions of the model without losses. This means we can't output the evaluation losses during training, which would be interesting. This would also mean that we can't easily compute the mAP for the training data.
- Error when running on MacOS locally – need to fix the error or try alternatively on cscs.

Weekly Review:

- Achievements:
  - Load Data.
  - Understand how data should look like for Mask R-CNN models.
  - Understand PyTorch Implementation of Mask R-CNN and where to find source code: Link.
- Areas of Improvement:
- Lessons Learned:
  - Don’t trust chatgpt blindly and look at the source code.
- Action Items for Next Weekly:
  - Normalize image data
  - Per team member, try a model version
  - Overfit on training data
  - Regularize