**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](https://github.com/navjotzubler/mc_dlbs_fs23_nzu_bkr/blob/main/dlbs_visualize-labels.ipynb)
  + 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.](https://fhnw365-my.sharepoint.com/:p:/r/personal/navjot_zubler_students_fhnw_ch/Documents/Documents/DLBS/DLBS_Pitch_nzu_bkr_20230424.pptx?d=w3d7f1fbf6292424e947257aadda2d993&csf=1&web=1&e=dRd59S)
* 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](https://pytorch.org/vision/main/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html#torchvision.models.detection.maskrcnn_resnet50_fpn). 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.](https://pytorch.org/vision/main/_modules/torchvision/models/detection/mask_rcnn.html#maskrcnn_resnet50_fpn)
* 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

**Weekly Review –** [03.05.2023 – 09.05.2023]

**Weekly Progress:**

* Completed Tasks:
  + Baseline model now uses correct mAP calculation from torchmetrics. Issue was that it wasn’t adding the predictions during inference to a list to calculate mAP from.
  + Skeleton model runs on kaggle with 3.5 mins / epoch and ocally 1.5 mins / epoch
  + Improved memory handling of GPU tensors. Deleting cache and unused variables for each epoch.
  + Do mAP calculation on CPU.
  + Make parameters match the paper’s parameters.
  + Normalize the image to the size as described in the paper.
* In Progress Tasks:
  + Visualize Results
  + Action Items from last week
  + Choose second model from paperswithcode
* Not Started Tasks:
  + Train a second model

**Impediments:**

* Memory problems GPU and CPU (when loading data and processing images)
* Navjot can’t resume minichallenge due to time constraints
* Model metrics are much worse than in the paper (20% mAP vs 70%+ mAP in the paper)

**Weekly Review:**

* Achievements:
  + Baseline runs
* Areas of Improvement:
* Lessons Learned:
  + Kaggle trains the model but the model is saved on a disk that gets deleted after the session has ended.
* Action Items for Next Weekly:
* Like last week:
  + Normalize image data
  + Per team member, try a model version
  + Overfit on training data
  + Regularize

+ Train another type of model

**Weekly Review –** [10.05.2023 – 16.05.2023]

**Weekly Progress:**

* Completed Tasks:
  + Prepared interim presentation for 16.05, therefore no tasks have been completed this week.
  + Navjot wrote Susanne to leave mini challenge
* In Progress Tasks:
  + Same as last week
* Not Started Tasks:
  + Same as last week

**Impediments:**

**Weekly Review:**

* Achievements:
* Areas of Improvement:
  + Interim presentation took a lot of time to prepare but some models could be set up to train while working on the p6 interim presentation.
* Lessons Learned:
* Action Items for Next Weekly:
* Same as last week

**Weekly Review –** [17.05.2023 – 23.05.2023]

**Weekly Progress:**

* Completed Tasks:
  + Train mask r-cnn model fully and add an evaluation notebook for the model for visualization.
  + Chose the 2nd model(s): EVA, Mask DINO, YOLOv8.
  + Add dockerfile and eva testing notebook highlighting that it does not work even with correct cuda and pytorch settings.
  + Test mask dino project (also didn’t work).
  + Used visual studio code to start docker container.
  + Upload data to Roboflow (<https://app.roboflow.com/dlbs-zcxsj/strawberry-hxgaj/>).
  + Used their notebooks to train a YOLOv8 models.
  + Added evaluation until now to the demo presentation for the 23rd.
  + Held the presentation.
  + Got feedback and put it into my TODO list.
* In Progress Tasks:
  + Visualize the results (now not only Mask R-CNN but also YOLOv8 results).
  + Train YOLOv8 with data augmentation.
* Not Started Tasks:
  + Same as last week
  + See if model architecture can be changed
  + Compare model architecture to Mask R-CNN (for report)
  + Compare model training to Mask R-CNN (for report)
  + Add manual evaluation
  + Check if YOLOv8 does their own normalization/augmentation by default.
  + Manually augment dataset (use custom augmentation because Roboflow is not flexible enough)

**Impediments:**

**Weekly Review:**

* Achievements:
  + Successfully held demo even with not many achievements and problems using Mask R-CNN
  + Successfully trained YOLOv8 model without much work.
* Areas of Improvement:
  + Wasted time trying to make other models work.
* Lessons Learned:
  + Don’t trust all Open Source models. Looking at quality of installation guide and open issues is important.

**Weekly Review –** [24.05.2023 – 28.05.2023]

**Weekly Progress:**

* Completed Tasks:
  + Trained more YOLOv8 models.
  + Implemented custom data augmentor.
  + Refactored code to ./src/utils.py
  + Add evaluation notebook for YOLOv8 with training plots and confusion matrices.
  + Wrote the evaluation directly into notebook and added everything to the report.
  + Wrote the report.
  + Sent the report.
  + Clean up code.
  + Rename a bunch of notebooks.
  + Make sure data into PyTorch Mask R-CNN makes sense (visualized in notebook and mentioned in report)
  + Check YOLOv8 base augmentation by asking forum and looking into source code (mentioned in report)
  + Compare model training to Mask R-CNN in existing research paper in report.
  + Add plots to report.
  + Connect project to Karpathy’s guideline in report.
  + Add a much more precise data set evaluation.
* In Progress Tasks:
  + See if model architecture can be changed.
  + Compare model training to Mask R-CNN (for report).
  + Lerntagebuch.
* Not Started Tasks:
  + Compare model architecture to Mask R-CNN (for report).

**Impediments:**

**Weekly Review:**

* Achievements:
  + Successfully trained different YOLOv8 models that beat Mask R-CNN.
  + Write a long evaluation about the YOLOv8 training.
  + Add a lot of evaluation to report and successfully complete it.
* Areas of Improvement:
  + One of the incomplete tasks is the learning journal. It couldn’t be completed because I got distracted by my bachelor’s thesis interim presentation.
* Lessons Learned:
  + Writing reports is