Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting Time: 20th of January

Ghaith:

- [In Progress]: Evaluate the (A General Framework for Jersey Number Recognition in Sports Video) model
- Suggested checking Super Resolution (SR) models
- [TODO]: Send the intermediate results (legible images) so everyone can analyze the results
- [TODO]: Create Repo to host the modified code changes

Tanmaya:

- [TODO]: Evaluating (A General Framework for Jersey Number Recognition in Sports Video) for performance analysis

Karel:

- Compare (Jersey Number Recognition using Keyframe Identification from Low-Resolution Broadcast Videos) with the other model (the SOTA)

Hamza:

- Review STR and Pose detector

Available Resources:

- Ghaith:
  - CPU: AMD Threadripper pro 5995wx (64 cores)
  - GPU: 3x NVIDIA A6000 (48GB each)
  - RAM: 758GB
- Tanmaya
  - CPU(s): 256
  - On-line CPU(s) list: 0-255
  - Model name: AMD EPYC 7742 64-Core Processor
  - Thread(s) per core: 2
  - Core(s) per socket: 64
  - Socket(s): 2
  - Mem: 2.0Ti
  - GPU - Nvidia A100

Next Meeting: 10:30 AM Next Monday

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 27th of January

Ghaith:

- Reproduced the (A General Framework for Jersey Number Recognition in Sports Video) results for the test dataset (link to results)
- Results indicate that need to check for occluded areas and might benefit from larger pixel count (SR)
- Created the Repo (link to repo)
- [TODO]: Implement SR in the pipeline

Tanmaya:

- Edit the environment to setup for A100 to later development/improvements
- The RTX A6000 targets rendering, workstation-based AI workloads, and graphics-heavy applications, while the A100 is designed for data centers and compute-intensive AI tasks.
- Continue replicating the challenge dataset
- Profile the repo and push the changes to the repo

Karel:

- No significant improvement in (Jersey Number Recognition using Keyframe Identification from Low-Resolution Broadcast Videos) compared to the SOTA
- Check for models to check occluded images
- Struggled to set up the OCR library, I was trying to use MMOCR but wasn't able to install it without issues, I spent too long on this, I'm switching ocr library to mindspore.
- ran the occluded images onto DBNet++ model, but was not able to perform on well using the pretrained out-of-the-box weights.

Hamza:

- I found papers for both topics (with models' code and dataset).
- **Scene Text Recognition** Scene Text Recognition | Papers With Code
  A General Framework for Jersey Number Recognition in Sports Video: STR uses the **PARSeq** trained on the SVT dataset. There are more accurate ones now, No.1 is **CLIP4STR-H (DFN-5B).**
- **Pose Detector** Pose Estimation | Papers With Code
  A General Framework for Jersey Number Recognition in Sports Video: Pose Detector uses **ViTPose** (trained on the COCO dataset). It's still the best so far (on the COCO dataset). There are models trained on the **MPII dataset** that might perform better e.g. **PCT** (Paper: Human Pose as Compositional Tokens)
- Continue checking the STR and Pose Models

Possible Improvements/Contributions:

- Update backbone models (e.g., ViTPose -> PCT)
- Fine-tune STR on a new dataset (reduce the Lipschitz constant when fine-tuning)
- Implement SR
- Check for occluded cropped images

Next Meeting: 10:30 AM Next Monday

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 3rd of February

Ghaith

- [Inprogress] Testing DRCT-SR to check if it will boost performance between attempting to fine-tune the model
- [Inprogress] Comparing RGDiffSR vs PARSeq without finetuning to check for preliminary results

Tanmaya:

- Edit the environment to setup for A100 to later development/improvements
- The RTX A6000 targets rendering, workstation-based AI workloads, and graphics-heavy applications, while the A100 is designed for data centers and compute-intensive AI tasks.

Karel:

- Found the winner method described in the SoccerNet report. Hopefully we can get some inspiration from them.
- [TODO] Implement J1 Step 1 on DBNet++ Text Detection. **Update:** the outline given on the SoccerNet report is very vague so I was having trouble recreating the ZZPM's pipeline.

Hamza

- Pose Detector Methods (Code and Models Link)
    - Paper: Human Pose as Compositional Tokens. Method: PCT
    - Good at detecting poses when occluded: *"First, when a large portion of the human body is occluded our method can predict a reasonable configuration for the occluded joints that are in harmony with the visible joints although there are no supporting visual features"*
    - With the MP2 dataset, it used PCKh@0.5 because that's the standard evaluation metric for that dataset. For COCO it's AP50 and AP75
    - On the COCO dataset, the PCT method has comparable or slightly better results than the ViTPose method used in the Jersey Detection paper: *"our largest model also achieves better results than the state-of-the-art ViTPose (huge) with 1.5x faster inference speed. The fast inference speed is mainly because our method does not require any expensive post-processing"*
- STR [Paper: CLIP4STR (Code and Models Link)]
    - Uses 2 branches, visual and cross-modal. Initial prediction based on visual features. Cross-modal branch refines prediction by seeing the discrepancy between visual features and the textual semantics
    - Uses Permuted Sequence Modeling like PARSeq. Permutes the text so it doesn't depend on left-to-right or right-to-left structures.
    - Beats PARSeq and other SOTA.
    - CLIP4STR can robustly read scene text that is curved, occluded, blurred, or rotated, showing its great robustness. It has a strong ability to complement incomplete characters
- [TODO]: Start writing a report on we have done so far.

Next Meeting: 10:30 AM Next Monday

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 10th of February

Ghaith:

- Finished testing DRCT and the performance was improved to 87.11808422791081% (result can be found here)
- [TODO]: Run the results of Karel's TextGestalt SR model as part of the pipeline
- Analyzed to the pipeline result and dataset to find alot of mislabeled data points and data imbalance in the final number and between single digits as well. The general framework paper used a CE with the final number and each digit to address that issue.
- [TODO]: Find data augmentation techniques and maybe use L1 Reg to feature select the valid features since the data is mislabeled.

Tanmaya:

- Reproduce results on Cuda 12.2
- Performance analysis for testing

Karel:

- Ran super-resolution model on the cropped images, TextGestalt to be fed into the General Framework pipeline.

Hamza:

- Discussed the template for the Project Proposal Template
- Viewed requirements to run the PCT code
- Took notes of updates of everyone's work, made a shared Google Doc to collect everyone's work to eventually put into the Project Proposal Template for submission

Next Meeting: 10:30 AM Next Monday

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 17th of February

Ghaith:

- Finished testing TextGestalt and the performance was reduced to 83.07184145334435%. It can be seen that the DRCT had better resolution and introduced less artificats compared to TextGestalt
- [TODO]: Run Bicubic interpolation to check that SR actually helps and not just the addition of extra pixels
- These is a lack of data augmentation in the current pipeline. Related papers for data augmentation techniques:
    - Data Augmentation for Scene Text Recognition: introduced eight different classes of data augmentation. According to their result the best were: Blur and Noise.
    - Self-compositional data augmentation for scene text detection: showed that the data augmentation improved results, in the presence of an unbalanced dataset, mainly cropping and rotation. Also suggested the use of unlabeled data in a self-supervised method.
- Maybe we can follow the randaugm method to incorperate it in the pipeline (paper: RandAugment: Practical automated data augmentation with a reduced search space)

Tanmaya:

- Performance analysis for training
- Need installation of packages not required for testing.

Karel:

- I'm trying to follow their idea of using an ensemble method using several STR model instead of just one like what General Framework pipeline was doing using ParSEQ, so I ran PPOCR-v3, PPOCR-v4, SVTR-TINY, ParSEQ, using PaddleOCR on my NVIDIA 4070 Laptop GPU. got some usable results for PPOCR-v4 and SVTR-TINY. The rest still needs further debugging since it all spits out null prediction for the cropped images.

Hamza:

- Discussed whether to write the proposal using LaTeX/Overleaf or Word/Google Doc
- Updated Tanmaya's LaTeX report template with our work

Amey:

- As a new member to the group, got up to date with what everyone else was doing in the group. Understood the pipeline and planned to profile the entire pipeline to figure out potential areas of the code which could be parallelized further.
- Read the following papers to better understand the project and complete pipeline before profiling it:
    - Soccer Jersey Number Recognition Using Convolutional Neural Networks by Sebastian Gerke, Karsten Muller, etc.
    - A General Framework for Jersey Number Recognition in Sports Video by Maria Koshkina, James H.Elder.
    - Automated player identification and indexing using two-stage deep learning network by Hongshan Liu, Colin Adreon, Noah Wagnon, etc.

Next Meeting: 10:30 AM Next Monday

Meeting time: 24th of February

Ghaith:

- Wrote the literature review section.
- Wrote most of the proposed pipeline section and created the corresponding figures.
- Gathered all the team's proposed improvements in the Proposed Enhancements & Performance Improvements section.
- Wrote the proposed timeline for the project.
- Cleaned up the tables and figures for the whole paper.

Tanmaya:

- Compiled results and plotted graphs to understand performance issues in existing code to understand what parts of the code could be improved computationally and what diff could be made.
- Proof reading and editing report and also compiling steps to reproduce results and analysis.

Karel:

- Wrote the STR models section of the report that talks about the proposed method of using an ensemble method instead of just one model, ParSEQ.
- Wrote the Out-of-the-box Models section of the report where PPOCR-v4 and SVTR-TINY accuracy and response time.
- Compiled the results into a table for the report.

Hamza:

- Wrote the introduction and parts of the literature review section.
- Shifted from the LaTeX doc to the Google Doc. It's in template form and it will be easier to synchronize everyone's work together.
- Formatted others' works, made the report neater and cleaner.
- Cleaned references using the Zotero extension.
- Added Figure and Table numbers.

Amey:

- Downloaded and configured NVIDIA's Nsight Systems and understood how the software worked to profile samples of code.
- Ran the profiler on the main.py file of the repo to get a detailed breakdown of the time taken for various functions in the code to run as well as the percentage of GPU utilized and the memory transfers.
- Created a report with the time splits, percentage of GPU utilized and kernel memory transfers.
- Analyzed the current GPU utilization in training the model.
- Analyzed the profile report to determine parts of the code that could be parallelized further to improve the performance of the model.

Next Meeting: Wednesday after the midterm*

*Everyone still has to work on the project according to the mentioned timeline in the report.

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 12th of March

Ghaith:

- Implemented data augmentation that includes blurring, noise, and simulating different weather conditions
- Was not able to find any unlabeled images that were preprocessed to include self-supervised learning (only ones that were available were a video of the different matches): will skip that part.

Tanmaya:

- Ran training of parseq with 1000 epoch and generated some results

Karel:

- Tried training SVTR and PPOCR with PaddleOCR, having issue reading the documentation as ppocrv4 recog docs are in Chinese :(

Hamza:

- PCT method unusable because their models were unavailable in the links provided in their GitHub repo. They are in organizations' OneDrives that have restricted access
- Planned to implement ViTPose and VitPose+ models instead

Amey:

- Ran Legibility Classifier in serial and evaluated the performance: time taken, amount of GPU usage to determine if it could be parallelized to improve performance.

Next Meeting: 10:30 AM Next Monday

Meeting time: 17th of March

Ghaith:

- Worked with Tanmaya to integrate data augmentation
- Started testing the bicubic interpolation method on the output of the ViTPose with ParSeq without any finetuning

Tanmaya:

- Tried integrating augmentation with Ghaith but ran in some installation difficulties for imagemagic

Karel:

- Turns out you can't train the recognition model first, you need to first train the recognition model which in our case the general pipeline already uses DBNet++ so it's redundant to retrain the detection model.

Hamza:

- Created environments for the VitPose and VitPose+ models
- Many of the checkpoints were unavailable, so we chose the ViTPose+-H model's checkpoints
- Turns out that after a lot of debugging of the environments and trying to run the VitPose models, the code was built for Linux as it used NCCL. I have a Windows machine, so I couldn't run it.
- We can replace NCCL with Gloo, but it will take too much time as we have to reconfigure the whole code. It's not feasible to do that. We won't replicate ViTPose's results

Amey:

- Did research on the NVIDIA's nv-math library which enables us to parallelize linear algebra functions like matrix multiplications and fast fourier transforms as an alternative to performing them in serial using python's numpy library.

Next Meeting: 10:30 AM Next Monday

Meeting Minutes (Progress Journal) for Team 3 (DL course)
GitHub link: https://github.com/Ghaith-Chrit/JerseyNumberRecognitionPipeline

Meeting time: 24th of March

Ghaith:

- Integrated the data augmentation on Hamza's machine
- Finished testing Bicubic interpolation and the performance was improved to 87.20066061106523%. It can be seen that it is even better than the pretrained-DRCT since it introduced less artifacts due to upscaling and down scaling
- Started testing ParSeq (no finetuning) without ViTPose (on the input of ViTPose)
- Started finetuning ParSeq on the input of ViTPose

Tanmaya:

- Compiling results of parallelization and running testing to check new accuracy results

Karel:

- Basically not a good idea to retrain det and recog models from scratch, maybe we should just use Parseq finetune, since it has a better out of the box test accuracy anyways.

Hamza:

- Created environments for the whole jersey number recognition pipeline
- Trained the PARSeq STR with data augmentation techniques with Ghaith. Basically had to train from the beginning. We couldn't use Tanmaya's checkpoints. Training took almost 6 days.

Amey:

- Implemented the Legibility Classifier part of the pipeline in parallel using the nv-math library and swapped out all the numpy functions with the corresponding nv-math functions. Evaluated the final performance of the Legibility Classifier and compared the performance to the original performance of the serial implementation.

Next Meeting: 10:30 AM Next Monday

Meeting time: 31st of March

Ghaith:

- ParSeq without finetuning performed terribly (this can be due to domain generalization since it always assumes that the number is in the middle). Accuracy on the test dataset is 37.902559867877784%.
- Finished preparing my parts of the presentation
- Formatted the presentation to have a coherent flow
- Started writing my parts of the report

Tanmaya:

- Completion of report and my part of the presentation.

Karel:

- Updated the presentation for the introduction, challenges, and off-the-shelf str models section

Hamza:

- Try to test the whole pipeline with all our changes. Tested for about 2 hours then it stopped.
- It got through the 1st few steps (ball detection, features generation). But it stops at the legibility classifier because my PC runs out of disk space to save the outputs. I am unable to free up space due to some restrictions.
- Discussed presentation with team and general planning for it. I will do the challenges part and the live demo. Ghaith will make a Jupyter notebook
- We can finish the report by Friday and proofread it before that

Amey:

- Worked on completing the report and the PowerPoint presentation. Reported my findings using the nv-math library to parallelize the legibility classifier. Explained the nv-math library usage in the presentation along with the profile report for the performance breakdown obtained using NVIDIA's nsight systems in the presentation.