Logbook

Tymo van Rijn

Graduation Project

March 3, 2026

Contents

Week 1 (09 Feb 2026 – 13 Feb 2026)

Monday, 09 Feb 2026

Goals

- Understand the dataset.
- Explore the dataset.

Work performed The way I'm going to approach this is to make a notebook that will give me valuable insights about the dataset. By doing this, I will be able to understand the dataset better and then I will be able to make plans for the next steps. The main point for me in this step is visualizing the dataset, since this makes it easier for me to understand certian patterns and relationships in the data.

Today I got quite some things done, I extended my Exploration notebook, which now has relevant and valuable insights about the dataset. For example the most common resolution, the number of frames per examination, the intensity curve of the examinations, and the temporal change of the examinations.

I also started to work on the feature extraction script, which is a script that will extract the features from the examinations. I started by extracting the features from the first examination, and I saved the features to a file. I also saved a debug image of the first examination, to verify that the crop is correct.

This way, I will be able to make a small experiment that tracks on what frames the accuracy is the highest. So that I will be able to understand which frames are the most important for the classification task.

Overall, I'm quite satisfied with the progress I made today. I will continue to work on the feature extraction script tomorrow, and I will continue to explore the dataset.

Outputs / Evidence

- Exploration notebook updates (frame statistics, resolution distribution, intensity curves).
- Feature extraction script (first exam extracted + debug crop image).

Next steps

- Continue feature extraction script.
- Extend probing experiment to identify informative frames.

Tuesday, 10 Feb 2026

Goals

- Finish the feature extraction script.
- Probe feature usefulness across frames.
- Create additional plots/visualizations to assess need for sequential modeling.

Work performed The extraction script is now finished, and I have a lot of features to work with. I then made a notebook block that gives insight in the importance of the frames over an examination for the classification task. It showed some interesting patterns, for example that window defect classification has more profit in the first frames.

After today I think my understanding of the dataset and the problem of the FA its self has become much better. Also on the physiological and clinical aspects of the FA, I have a much better understanding. I think it would be nice to grab some good studies done in this field already, to get a better understanding of the current state of the art. So that's what I'm going to be doing for the rest of the day, and hopefully tomorrow I will be able to get my hands on mednet and gain knowledge about that library.

I also changed the extraction script, so it places all the exam in dedicated train, val and test splits, given by the mednet hyperftype.json file. This way I can easily use those features for the next steps, so that I can sort of replicate the original experiments done by Roberto for example.

I found some interesting papers, and went in-depth for one of them about using longitudinal data for eye disease prognosis tasks. It has some interesting insights, and I think it will be a good starting point for my own research.

Now the plan was to get the baseline results from Roberto's experiments (which were done on only the last frame from an examination), so that I can compare future results with those. I managed to fork his repository and run the experiment.py to get baseline results, which I wrote down in my slides. Now in the future I will compare my new approaches with these results to see if the temporal aspect of the examinations is beneficial for the classification task.

Outputs / Evidence

- Feature extraction completed; train/val/test split aligned with `hyperftype.json`.
- Probing notebook block: frame importance patterns (e.g., window defect early frames).
- Baseline replication run (`experiment.py`) and recorded results.

Next steps

- Read and summarize key papers.
- Start integrating multi-frame strategy into the pipeline.

Wednesday, 11 Feb 2026

Goals

- Modify `experiment.py` to use averaged embeddings over n last frames.
- Find additional relevant papers.
- Get Approach 1 working end-to-end.

Work performed So today I am starting on trying to understand the mednet library and I will try to modify the experiment.py to use all the features averaged over an examination. The first step for me is to clone this repository and try to understand the code and the structure of the library.

With some adjustments, I think I was able to modify the code, so that it now accepts a N amount of (last) frames from an examination. The results were better than expected on some of the HyperF_Type classes. Especially the window defect class was classified with a higher accuracy and AUC.

I also found 2 more interesting papers which I then read and made a quick summary of in my presentation which I am going to present tomorrow.

The plans for tomorrow are to make clear in the presentation what I actually did to get to these results and to try and modify the amount of frames used, to see if that makes a difference in the results. Also trying to figure out what my next steps will be after this.

Outputs / Evidence

- Modified pipeline supporting last-n frame averaging.
- First results recorded (notable improvement for window defect).
- Two paper summaries added to analysis materials.

Next steps

- Prepare presentation explanation of changes and results.
- Sweep different values of n.

Thursday, 12 Feb 2026

Goals

- Test different n for last-n frame aggregation.
- Continue literature reading.
- Refine Approach 1 and reasoning around temporal sampling.

Work performed So today I am continuing to work on the modified mednet experiment.py, trying to figure out what my next steps will be after this. I think averaging only the last frames isn't really the best idea, since clinicians don't act this way either when examining an FA exam.

From what I've read in the literature, FA examinations are interpreted by clinicians using the different phases of the examination.

1. Choroidal phase: The choroidal hyperfluorescence is present. A cilioretinal artery if there is one iwll fill in this phase. Delayed choroidal filling time happens in ocular ischemic syndrome

2. Arterial phase: Arteries are bright, but the veins remain dark.

3. Arteriovenous phase: Laminar flow in the veins - the walls of the veins are bright while the center of the vein is stilll dark.

4. Venous phase: Complete filling of the veins

5. Late phase: Dye has recirculated. Things that are going to leak or pool will have done so already.

So I guess it does make sense that the last frames contain most of the information, since they should show the most important information for the classification task.

But if these clinicians look at the examination in these phases, they will probably not only look at the last frame, but at the whole examination. So I think it does make sense to use the whole examination for the classification task. And specifically put them in phases (buckets)

Outputs / Evidence

- Notes on FA phase interpretation and implications for temporal sampling.
- Decision to explore bucket/phase-based sampling rather than only late frames.

Next steps

- Define a bucket strategy (early/mid/late) and implement sampling.

Friday, 13 Feb 2026

Work performed Today I am presenting my findings and results to the Andre, Oscar and Roberto, I made a presentation about my analysis of the dataset and my approach to the problem. I also showed them what literature I have read and what I have found interesting in the field of sequential data classification and FA. The goal of this meeting was to get feeedback on my approach and my results and to get some inspiration for my next steps.

The presentation went well, I have got to admit that I was nervous, and that didn't quite go away until the end of the presentation. But overall it went well, and I got some good feedback on my approach and my results. We also discussed some potential next steps, and I will be working on the following: - Trying to make a video of the FA examination process, to get a better understanding of the process and to get inspiration for my next steps. - Try to implement the idea of buckets (phases) for the classification task. (idea to average what's in the buckets or using laplacian formula) - Try to actually extract temporality features from the examinations, since on the frames there is an elapsed time shown (vaguely), this could be extracted and used for a more stable temporal feature. - Run the best model of Roberto's experiments, so I have a better baseline to compare my results with. - Do more research on evaluation metrics, "balanced accuracy" was mentioned by Andre, so I will be looking into that for example.

So for today I will try to get a better understanding of how I am going to transform the frames into a video, and how I am going to showcase this.

Feedback / Decisions

- Explore phase-based buckets (possibly with sharpness selection).
- Investigate temporal feature extraction from elapsed-time overlays.
- Run Roberto's best model as a stronger baseline.
- Look into evaluation metrics such as balanced accuracy.

Next steps

- Create FA examination video visualization.
- Plan bucket-based experiment implementation.

Weekly Reflection

What went well

- Rapid onboarding into the dataset and pipeline; baseline reproduced successfully.
- First multi-frame experiment moved the needle for specific classes (e.g., window defect).
- Built a concrete foundation of artifacts: notebooks, scripts, presentation.

Received feedback

- I should look into the way I evaluate my findings more, be aware of confirmation-bias
- My graphs should contain more information about class distributions, could otherwise be confusing

What did not go well / challenges

- Simple averaging risks losing temporal order and phase-specific signals.
- Understanding and modifying the mednet pipeline required significant time.

Key learning

- Temporal dynamics likely matter differently per class; a single aggregation strategy is insufficient.

- Clinical interpretation suggests phase-aware sampling is more aligned than only late-frame selection.

Action points for next week

1. Implement bucket/phase-based temporal sampling (e.g., 3 buckets).
2. Compare bucket sampling vs last-n averaging using same evaluation protocol.
3. Add at least one order-aware method (e.g., learnable weighted pooling) as a stronger baseline.
4. Take evaluation metrics into account more, be able to explain why I made a certan decision on using a specific metric.
5. Update the legends, so they also provide the class distribution information.
6. Make a video out of the frames from a FA examination.
7. Extract the "elapsed time" feature from the frames inside of an examination.

Week 2 (16 Feb 2026 – 20 Feb 2026)

Monday, 16 Feb 2026

Goals

- Extract time feature from FA exam frames.
- Turn frames into a video

Work performed Today I had a lot of stuff to do, I started with trying to make videos out of the frames, which turned out to be harder than expected, since a lot of the frames did not have stable sequential frames. Several camera shifts appeared, the coloring changed, etc. This made the task quite hard, but I went through several exams for each HyperF_type to get the best video results for each HyperF_type.

So I wrote a script to see which examinations had a good amount of frames to even be able to make a video out of it, this script returned which exams had the most frames for each HyperF_type, which gave me a good idea of which exams I should probably be using.

I also tried using OCR (Optical Character Recognition) to extract the "elapsed time" feature from the frames of an FA exam, but this didn't go well either. Seems like the OCR I'm using right now (EasyOCR) is not always picking up the right digits, which makes it hard to make a solid .csv with all the frame times. This was to be expected since the digits are written in a pretty small font, nevertheless, I will still try to make some modifications to make this work.

At the end of the day I didn't accomplish much, I tried to do too much in one day, when I got home I was planning on doing some more investigating, but then my laptop's SSD broke down so I couldn't reach any of the data anymore...

I think what I learned today is to work on one task at a time, otherwise I'm driving myself crazy, which kind of happened today. Which in the end only makes me do less, instead of more.

Outputs / Evidence

- Information about how OCR works and how it is going to be implemented in our specific case.
- Made an example video of a "leakage" examination, which shifted alot so is not that clear at all.

Next steps

- Work on one thing tomorrow, only move on when it is finished. The one thing would probably be OCR (if my laptop is working)

Tuesday, 17 Feb 2026

Goals

- Extract time feature from FA exam frames.

Work performed So for today I'm only focusing on the feature extraction, well, I thought I was. But me laptop is still broken and the help desk is taking quite long to fix the issue. So for now I'm just reading into more information about positional encoding, which would be the next step to implement (based on buckets).

I am kind of annoyed and not used to these setbacks. Although I realize they are inevitable, I

do not really know how to react to them, I can't help it but feel sad and disappointed. I want to be working on something and now my planning is not working out anymore, I feel like I'm going to be behind of work because of this.

In some way I think it is a good thing that this is happening to me right now, this way I can become more familiar with these things happening to me, but that does not make today a better day.

I made some good progress on the OCR work! I generated a .csv file containing all the frames linked with their elapsed time timestamp. Then I visualized it to see how long most exams last and when most of them "begin" (first timestamp available). This gave me some interesting insights, as well as the sad truth that not all of my OCR reads worked well, some have really odd timestamps which are probably made by mistakes.

I am still trying to get better results, so that I can show the results with confidence in the next meeting! This all took way longer than I had anticipated, but I am glad I took the time and didn't rush into it.

Focusing on one task at a time also had a great impact on today's results, I knew what I had to do and I didn't get tangled up in too many things, which resulted in a clean work-flow (is that a word?)

Outputs / Evidence

- Got a somewhat-okay OCR output (.csv file with timestamps) !
- Tried to make different buckets using these timestamps, which was not going well since they weren't in-line with the clinical phases of FA.

Next steps

- Working on creating videos for each HyperF_Type, probably not going to look great, but just to show that it is very hard to get a stable video.
- Also need to start thinking about how I'm going to be designing a new approach on the model, since the averaging didn't work out that well.

Wednesday, 18 Feb 2026

Goals

- Make a video of FA exam based on the frames
- Design an architecture for using the bucket strategy

Work performed

I just finished making a video sequence for each of the HyperF_Type's, I took exams with at least 30 frames and merged them into one video, with a smooth transition between frames so it seemed more stable (so I thought).

It turned out alright, not quite amused with the results, but I think handpicking everything to get slightly better results is not worth the time for now, it could be something I am going to pick up later on.

Next step for today is to try and figure out an architecture for the bucket strategy, how would something like that look like? I will try and visualize it for myself first, based on the articles I have already read.

After I watched some educational videos about the entire idea of positonal encoding, I think I have got a better idea about the approach now, although it still is a bit sketchy. Especially since our entire bucket idea was based upon the timestamps, which now seem to be a bit useless given that they don't represent the different phases very well. For example, the first phase is between 0 and 15 seconds from the start, but there are only 0.9% of exams that actually have frames during this first phase.

I think the main problem is how we want to bin the exams, which is something I am going to figure out tomorrow and experiment with!

Outputs / Evidence

- Managed to make videos for every HyperF_Type, which didn't turn out great.
- Got a better feeling about how positional encoding in ViT's work and how we can approach the bucket strategy in our case.

Next steps

- Get a concrete idea on how to bin the exams and try to get a test experiment going.
- Do more research on positional encoding with ViT's

Thursday, 19 Feb 2026

Goals

- Try to get a first test going for positional encoding.
- Get a better understanding of positional encoding.

Work performed

After some more research I have figured out how I want to approach this positional binning idea. My first experiment will consist of 3 bins, (since the mode of n_frames is 3, I thought it would be nice) I will add a –n-bins argument which makes it dynamic.

Then in the function RawDataLoader.sample() I will make some changes so an entire exam is divided over 3 bins, so $\frac{n\_frames}{3}$ -> each bin contains $\frac{n\_frames}{3}$ frames (or some bins are empty, if $n\_frames < 3$. Then I need to think of a way to pick 1 frame for each bin, so there are in total 3 frames in total.

An extra parameter(?) is added to the sample output, *bin_indices*, this is a tensor that keeps track of which frames is originating from which bin. This may seem useless, but it is necessary to handle edge cases: an exam with only 2 frames and 3 bins, then there are only 2 bins with actual frames and 1 empty bin. The empty bins would be indicated with -1.

Example 5 bins, 3 frames:
$bin\_indices = tensor([0, 1, 2, -1, -1)]$

A new collate function will be made that processes multiple bin-frames per sample.

The $Hyperftype.py$ also needs to be modified so the n_bins is passed to the DataModule

The $Configuration.py$ will get a new model class choice: ViT_bins and then the n_bins will be passed to the model, so it knows how many bins are being used.

The model also should be changed, so *vit.py* will have the following changes:

- New class $ViT\_bins(ViT\_enhanced)$ with:
- $nn.Embedding(n\_bins, embed\_dim)$ for the positional embeddings (KEY!)
- *BinAttentionPool* module*
- Custom forward(), training_step(), validation_step(), predict_step()

BinAttentionPool is going to be used because: There are K embeddings (one per bin), e.g., 3 vectors with 1024 dimensions. They need to be fit into one vectr of 1024, averaging is "dumb", so I want the model to learn which bins are most important.

With a normal average, each bin would weight 0.33 (if $n\_bins = 3$), everytime, despite the output. By using attention de weights are input-dependent – they change per sample.

The model can learn:

- For leakage the late frames are more important
- For window defect the early frames are more relevant
- For this specific exam the middle frame contains the most information

It is known as the best practice in medical vision analysis for combining multiple views/frames per patient + it is very light, it is only one extra linear layer. You can also inspect the attention weights to see which bins contain the most information.

Outputs / Evidence

- Envisioned a model architecture for temporal data

- Read literature about AttentionPool module
- Made an analysis on the OCR timeframes which were extracted from the script I wrote yesterday

Next steps

- Try to put this in an advisory manner, to present it during the presentation upcoming Monday.
- Build a beginning for the binning idea into the existing pipeline.

Friday, 20 Feb 2026

Goals

- Make the advise of my model visual for the presentation
- Figure out how what binning would actually look like in our existing pipeline

Work performed

Today I started visualizing my ideas about binning. I made some diagrams in my slides to showcase the architecture of the pipeline, this should give people a general idea of how it is going to be functioning. This took quite some time, since you have to think about a lot of things, some concepts may seem obvious to me, but other people might not understand it.

I finished my Week 2 presentation today with the findings and all of my new ideas (also for the upcoming week(s)). I am kind of wondering how to actually structure things as of right now, since in my presentation I understand most of it, but I don't really have the profound (mathematical) knowledge of how everything is actually working "under the hood", this is a big weakness for me.

I feel like I know what I am doing, what architectures I am creating, and why I want to do certain things. But if I really ask myself the question if I understand the things I am doing on a deeper level, I am faced with a lot of confusion surrounding the topics I am covering. This results in me not being as confident as I would like to be in talking about the things I am (planning on) doing.

I think this is a good learning point for me that I have already discovered within 2 weeks of being here, I am more of a practical thinker. When I see a problem, I feel like I know how to fix it, so I jump straight into it without much theoretical knowledge. e.g., the positional embeddings in my architecture are very simply done within the PyTorch library, so I think "lets use it!". But when I need to explain how positional encoding is actually working inside the architecture I have NO clue...

If I really want to become an expert in this field (which I do want to become), I have to dig deeper and ask myself more questions. Which is what I will be doing from now on, I hope to become more confident in my way of doing things this way.

"Sometimes you have to take a step back, to move forward", which is exactly what I will do. I will try to start from the basics and not just out of nowhere try to design an entire working pipeline for this temporality problem.

I will still present my things upcoming Monday, but after that I will dig deeper in the theory and make sure I understand what I am doing.

So today was more based on making the presentation and trying to see where my current knowledge gaps lay, which I think was a very good and productive way to spend my day!

Outputs / Evidence

- Made an advisory presentation
- I dug deeper into the basics of what I am doing by starting to watch some explanatory videos on YouTube

Next steps

- Get feedback from the presentation
- Find some more educative sources to learn from

Weekly Reflection

What went well

- I translated last week's conceptual ideas into concrete implementation steps. The dynamic binning strategy was partially implemented in the pipeline, allowing exams to be split into n_bins and producing multiple embeddings per exam.
- I focused more deliberately on one task at a time compared to the previous week, which improved productivity and clarity.
- The laptop/SSD issue was resolved and I successfully restored my working environment, ensuring no further technical delays.
- I moved from purely exploratory experiments (averaging frames) toward a more structured architectural design (binning + attention pooling).

Progress on last week's action points

- **Implement bucket/phase-based temporal sampling:** A dynamic binning mechanism was implemented that divides exams into n_bins and produces corresponding embeddings.
- **Compare bucket sampling vs last-n averaging:** Conceptual groundwork was laid, but full comparative experiments will be conducted next week.
- **Add order-aware method:** Designed an attention-based pooling strategy (BinAttentionPool) to replace naive averaging, with a clear architectural plan.
- **Improve evaluation awareness:** Reflected more critically on what the metrics (AUC/AP) actually represent and how class imbalance affects interpretation.
- **Extract elapsed time feature:** OCR pipeline was improved and timestamps were extracted, though still imperfect.
- **Create FA videos:** Created representative videos per HyperF_Type to better understand temporal instability and frame variability.

What did not go well / challenges

- The OCR-based timestamp extraction remains noisy and not fully reliable.
- While the binning implementation works structurally, I do not yet fully understand positional encoding at a mathematical level. The mechanism functions in PyTorch, but the theoretical foundation is still developing.
- Temporal phase alignment using timestamps proved less clinically meaningful than expected.

Key learning

- Temporal modeling requires both architectural design and a solid theoretical understanding; implementation without deep comprehension reduces confidence.
- Binning solves computational and variable-length issues but does not automatically encode temporal order.
- Attention-based pooling is a principled improvement over averaging because it allows sample-dependent weighting.
- Technical setbacks (hardware failure) are inevitable; resilience and workflow recovery are part of professional development.

Action points for next week

1. Finalize and validate the binning + attention pooling implementation experimentally.
2. Deepen theoretical understanding of positional encoding (especially how embeddings interact with transformer self-attention).

3. Run controlled comparisons between last-n averaging and bin-based sampling using identical evaluation protocols.
4. Improve robustness of OCR preprocessing or reconsider its role in temporal modeling.
5. Increase mathematical confidence by revisiting linear algebra foundations relevant to positional encoding.

Week 3 (23 Feb 2026 – 27 Feb 2026)

Monday, 23 Feb 2026

Goals

- Give the presentation, obtain feedback.
- Continue working on the binning pipeline

Work performed

So today at 09:00 I had my "Tymo's week 2 progress" presentation, which went alright. Like I have mentioned earlier, I don't really know the technical details about what I am exactly doing, so when I got questions about the pipeline I couldn't answer them properly. At some point I was being honest and said that I didn't feel confident in continuing the presentation since I felt like it would not be of much use.

The confession of this to people who expect things from me was hard, but I think it was a mature thing to do: pointing out my weaknesses.

I got some great advise from Andre and Oscar on how to advance from here, the main takeaway was to just start out in a simpler manner, which they thought was not something to be ashamed of. I am really happy that I spoke to them about my knowledge gaps and I feel like they really understood me.

They were really happy to see the OCR results and the videos I was able to make, at the end of the meeting I got the following feedback:

- Annotate some random exams that have been "OCR-read", to see if we can trust the OCR timestamps extraction.
- To discover the SOIN frame distribution aswell, to see if we can use the timestamps from there for actual phase-based-binning
- For the videos it was important to implement key point registrations, so the video would remain stable.
- Adding the elapsed time + HyperF_Type to the video would also be useful.
- Try to use a LSTM / GRU instead of something else to keep it simple and just to discover if temporal binning data would actually make sense.

So there were a few things I could start working on, I really gained some more structure after the meeting.

The first thing I did was dive deeper into making of the videos, the idea of registering the key points like the optic disc and some vessels to keep the video stable really motivated me to create a better video! I really saw the use in trying to do this.

So that is exactly what I did; I investigated ways in which this was possible, and I stumbled upon "ORB". ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First, it uses FAST to find keypoints (like the optic disc), then applies Harris corner measure to find the top N points among them.

Keypoints detection is all about finding special, easily recognizable spots in an image, and a feature descriptor gives us a numeric summary of what is around each of those spots.

So, this day really revolved around doing some ORB experiments, which was really interesting. I decided not to go too much into detail for the ORB, I just wanted it to be practical, so that is what I did.

I also started manually checking if the extracted timestamps were actually correct, which was not going smoothly. So I decided I needed a little program to evaluate and "annotate" these timestamp using a GUI. I started on this and I am hoping to finish it by tomorrow.

Something I also did was explore the opportunities to learning more about the theoretical side of what I am doing. I stumbled upon this book, which is called "Mathematics for Machine Learning" (MML). I think this could form a great baseline for my knowledge about this topic, already managed to understand quite a bit.

Outputs / Evidence

- Got feedback from my presentation / progress for Week 2.
- Great start for the creation of videos out of examination frames using keypoint detection (think that is what its called)
- GUI for validating the extracted OCR timestamps.
- Studied from the MML book.

Next steps

- Finish the GUI for the OCR validation
- Apply the ORB on the frames and try to stabilize the frames and make a video out of it, containing the useful information; time, HyperF_Type, exam and frame count.
- Continue studying the theory from my book (MML).

Tuesday, 24 Feb 2026

Goals

- Create the first FA exam video
- Validate 500 random frames for the timestamps

Work performed

This morning, Andre, Oscar, Roberto and I had a meeting with Deborah Barnett (my school supervisor) about my graduation project and how it is going to look like. I think it was a very good meeting, since everyone now has a better idea of what I need to be doing in the upcoming months.

After the meeting I immediately went to the store. validating the 500 random frames for timestamp validation. I let AI write the rest of de GUI (since I am not that good at html etc.), and I ended up with a nice little program that would let me validate the timestamps very easily!

One hour passed and I had already validated 500 frames, the results were promosing, 90% of the timestamps were correct. Most of the mixups were because of the number 6 being seen as an 8. So this was quite promising!

After that I went straight back into the ORB task, which then took me a while to figure out how to work with it to make a video, but eventually I created my first "stable" FA exam video!. But this one was without all the other info (timestamp, HyperF_Type, CNV (Choroidal Neo Vascularization and a timeline). So that was going to be my next step.

I used the already existing .csv's to extract the metadata and tried to frame them nicely into the video so everything would be visible at the same time.

This worked out fine and after some time and experimenting with opencv (python library), I managed to create a well framed video!

Outputs / Evidence

- Validated 500 Timestamps
- Created videos for 2 HyperF_Types

Next steps

- Create videos for the other HyperF_Types
- Get an understanding for LSTM's and/or GRU
- Keep studying from the book

Wednesday, 25 Feb 2026

Goals

- Create videos for the other HyperF_Types
- Get a general understanding for LSTM's / RNN

Work performed

Today I am planning on finishing the other FA exam videos, which I should be able to do since I have done quite some work/research already. I already have the keypoints matched and aligned through frames, the big job now is to actually use rotations etc. accordingly.

The first exam I am trying to make a video for today, will be a pooling exam, since I am most curious about this one.

I finished the script, which mainly was just retrieved from a website (linked in my slides). So then, I took some time to find a qualitative pooling exam which had a good amount of frames ≈ 30. I finally picked 2 exams (where pooling occured) and put those frames through my script, which would then automatically create the FA exam video. The results were quite good, I directly sent the video to Oscar to see what he would think of it.

Oscar liked how stable the video looked and said that for the visualization this is good enough. He came up with a great idea; "Maybe in the later stages it would make sense to do this preprocessing of the images before feeding them to the models". I though this was a really good and original idea so I wrote it down (credits to Oscar of course!)

With the videos out of the way (for now), I could start investigating LSTM and RNN. These concepts were new to me, so, of course, I was excited to learn about them, but I also knew that a lot of information was yet to come in my direction.

I started by reading the Geeks4Geeks page about LSTM, which always lays a good and simple foundation for me, just to have some background information. Then I went ahead and watched some YouTube videos about the concept. I realized , again that my mathematical skills were lacking, so I got even more motivated to continue studying from my book.

As the day passed, I watched more and more videos and read more and more articles explaining LSTM and RNN. I have a better understanding of it now, but it is not yet where I want it to be.

Outputs / Evidence

- Created videos for every HyperF_Type
- Did research about LSTM and RNN

Next steps

- If time accepts it, make a pipeline for how the LSTM is going to look like

Thursday, 26 Feb 2026

Goals

- Try to make a pipeline sketch for how the LSTM is going to work
- Get a better understanding about LSTM

Work performed

Today I went and tried to design a LSTM pipeline that would work with our data. I spent almost the entire day doing this, as there are quite some things involved in making this. First of all, a thorough understanding of the current pipeline is necessary, which in my case is somewhat the case now, but still not a 100%. Then the next point is to see where and how this LSTM can be implemented, since it needs to function inside of the existing pipeline.

The main idea was to implement the LSTM after the ViT, so the ViT would make embeddings, and these embeddings would be fed to the LSTM. I decided NOT to freeze the backbone since I didn't have any experience with it. This means that all the ViT parameters can also be optimized during the model training optimizer step. This leads to a lot more trainable parameters, since the LSTM also has trainable parameters.

The dataset is quite small, so this could lead to overfitting, but that is something I have to find out. There are a couple of things that could help with the overfitting, there are already some augmentation techniques applied to the forward method.

I made a presentation on how I am planning on making the pipeline happen, from the way the bins are made up to what happens with the binned frames and so on. Eventually there were be 5 logits again representing the probability distribution of all the HyperF_Type classes with a softmax at the end. I hope I can explain my ideas through this visual representation of the "flow", and get good feedback regarding to my approach on this matter.

I also learned alot from the PyTorch documentation on LSTM, this showed me how they can be used and what is needed to make them work (what the input should look like, and what the output will look like). This was necessary to know, otherwise I wouldn't know how to prepare my data for this job.

I found out that there would be a lot of ways to actually select the frames for these bins, so I will explore the options tomorrow and take them into account when finalizing my architecture sketch.

Outputs / Evidence

- Created a presentation that shows how I am planning on implementing the LSTM in the current pipeline
- Did research about LSTM and RNN

Next steps

- Finalize the architecture of implementing an LSTM into the pipeline
- Start writing the first code to create bins inside the *angioreport.py* dataloader function

Friday, 27 Feb 2026

Goals

- Finalizing the architecture of implementing the LSTM into the pipeline (or just in general)
- Start writing the conceptual binning logic for the dataloader function

Work performed

Today was a meh day, I didn't feel well and that had a huge impact on my motivation. The last couple of days I really started missing my loved ones back in the Netherlands and it projects on my general well being. I feel kind of alone with no one to really communicate to about this in person. I need to find a way to fix this because it is having huge impact on getting my work done...

But this is part of the challenge I put myself into, I knew it was not always going to be easy to be away from home, but the most important thing for me now is to learn how I can deal with these "emotional setbacks". And most importantly, how do I make them not obstruct my performance at work.

Because I couldn't really focus today, I decided to read some educational blogs and watched some educational videos about the subject, this was low-effort for me and I didn't quite feel like I was wasting time.

I would have rather talked about me doing fantastic stuff today and making some really good progress, but I am not perfect and these days exist as well. Hopefully it will get better soon, so I give 100% again!

Outputs / Evidence

- Knowledge

Next steps

- Finalize the architecture of implementing an LSTM into the pipeline
- Start writing the first code to create bins inside the *angioreport.py* dataloader function

Weekly Reflection

What went well

- I actively addressed the feedback from my presentation by shifting from broad architectural ambition to smaller, controlled experiments (OCR validation, video stabilization, LSTM exploration).
- I translated qualitative observations (temporal instability, frame variability) into technical experiments (ORB-based registration, timestamp validation, binning logic exploration).
- I validated 500 random OCR frames and quantified correctness (90%), showing a more data-driven and critical approach instead of assuming correctness.
- I moved from "trying models" toward understanding why temporal modeling might work, by studying LSTM theory and explicitly questioning positional encoding and hidden state dynamics.

Progress on last week's action points

- **Validate OCR timestamps:** Implemented a GUI for manual verification and validated 500 random frames (90% correct). Identified systematic OCR confusions (6 vs 8) and fallback logic using ICGA timestamps + interpolation.
- **Create representative FA videos:** Implemented ORB-based keypoint registration and created stabilized videos for all HyperF_Types including metadata overlays (elapsed time, labels).
- **Explore temporal modeling alternatives:** Studied LSTM and RNN fundamentals; designed an LSTM-based pipeline concept integrated after the ViT backbone.
- **Design binning logic:** Refined conceptual binning strategy and analyzed how frame selection influences temporal learning.
- **Strengthen theoretical foundation:** Started systematic study of "Mathematics for Machine Learning" to reduce conceptual gaps in linear algebra and representation learning.

What did not go well / challenges

- During the Week 2 presentation, I was unable to clearly explain core architectural mechanics (e.g., embedding flow, positional encoding timing), revealing insufficient theoretical grounding.
- OCR remains imperfect and introduces noise, which complicates reliable phase-based binning.
- Emotional setback on Friday reduced productivity; maintaining performance while abroad requires better structure and coping strategies.

Key learning

- Architectural decisions (e.g., LSTM after ViT) require understanding both the data representation and gradient flow, not just implementation syntax.
- Temporal modeling is not only about "adding more frames" but about preserving order and encoding change.
- Empirical validation (e.g., 500-frame OCR audit) strengthens credibility far more than theoretical assumptions.
- Professional growth includes acknowledging knowledge gaps early and converting them into structured learning plans.

Action points for next week

1. Implement first working LSTM prototype integrated after the ViT backbone using controlled bin input.

2. Document mathematically how hidden states h_t and cell states c_t evolve in the LSTM to ensure conceptual clarity.
3. Quantitatively compare logic-based bin selection versus random bin sampling.
4. Evaluate whether timestamp-based phase binning is robust enough for experimental use.
5. Establish a fixed daily theoretical study block (MML + transformer mechanics) to systematically close knowledge gaps.