# Integration of Complex Machine Learning Techniques into Real World Systems


## Swarnim Kalbande

Advised by Professor Bhiksha Raj


## Background

I'm a Junior at CMU with a major in Computer Science and a minor in Machine Learning. During my summer internship at Norilla in the Swartz center, I was working on basic object detection of toys using depth data and found the techniques used very outdated and tedious to work with. During this time I was acquainted to Agot.ai who were also working in the Swartz center and dealing with a much more challenging computer vision task – setting up a robust computer vision framework for food detection in fast food chain restaurants to support cashier-less checkout among a variety of other applications.

I saw this as an opportunity for a very interesting and challenging undertaking. When I was shown their systems at first and what they had working so far, I was very skeptical of whether the idea would work but was completely sold on the fact that if it did it would be an incredible advancement in how "fast" fast food really is. However, as I understood the systems at work better, I saw how feasible the whole task really was. In my excitement, I did – to some extent – underestimate the challenges that they faced.

I was always interested in machine learning systems and had done well in controlled environments like in classes but always craved more control over the process of how everything was done and always saw flaws in the so-called "pipeline." I signed up for an independent study that would let me explore my area of interest while also engaging with Agot, which I came to better define as the title of poster.

This poster goes over what the pipeline was, how well we had gotten it to work and how that all went to waste when the pandemic happened. It is true that a pandemic is a fairly rare occurrence so it might be a niche to focus on recovering from something that goes this wrong – but really it is symbolic of what happens to many in the pursuit of using machine learning techniques for real world applications – both in research and in startups. It symbolizes the hurdles they must face and overcome. So this poster goes over the entire process from the start and more importantly, it goes over how it changed post-pandemic and pretty much boosted our process further than we would have gotten had the pandemic not happened.

## Pre-Pandemic - The original pipeline

The very first thing to do was setting up data collection pipelines for RGB and depth data from Sushi Fuku who let us collect indefinite video data from their store. The most important thing with this endeavor was to have a way of moving current models to a new store (usually of the same chain) with minimal labelling costs and retraining models. This data is used by our object detection and action recognition models. I spent the first three weeks setting up an annotation tool (CVAT) by modifying its source code to setup a new labeling pipeline that supports attributes for each labeled box (like ingredients in a bowl) and added live tracking of boxes to significantly speed up manual labeling speeds. Live tracking tested with 8 different models (BOOSTING, MIL, KCF, CSRT, MEDIANFLOW, TLD, MOSSE, GOTURN) to finally find out that our combination of frame rate with the pillow library's jpeg compression makes it impossible for the tracking to be good enough to speed up labeling.

Despite that setback with tracking integration, CVAT was still unfortunately our best option since we needed support for attribute labelling. We produced a layered method of yolo detection to prevent have the cartesian product of CVAT attribute combinations to be the number of training label values since that would make any model useless. So instead we decided to use one model to separate base labels and then use another set of models – one for each category – with different parameters to determine if a given category (say bowl) has a certain attributes (say rice) or not. This made the learning space a sum of the number of categories and attributes instead of product.

With this and some data format conversion scripts we had a circular pipeline setup so instead of having to label a video from scratch we could provide our labelers with current best model's predicted bounding boxes, which brings us back to main objective of speeding up labelling time when deploying to a new store. Without tracking, this was still slower than our previous pipeline that had tracking working but without attributes. So instead of outputting all frames, I set up a probabilistic sampling of frames that matched certain criteria (like having low confidence bounding boxes or missing tracker ids in frame) to create a dataset that would focus areas of our model where it was weak and thus train it more effectively while needing much fewer frames labelled. Finally we had a good pipeline to retrain the model for new places by sampling data to be labelling from the old model's predictions on the new place, and then using that labelled data to create a model for that place (in this case it was a tested on a new branch of Sushi Fuku).

## Depth Data Fiasco

Next I was assigned the problem of serving size detection which was best solved with the depth video data. I researched all potentially useful ways of solving single vs double scoop problem with depth data and settled on regression. I realized that the nature of the problem was such that if regression can't solve it nothing can since it's a volume-based difference between scoops and this would effectively be differentiating using volume without explicitly calculating volume. Counterintuitive but I expected it to work better than actual calculations for volume using depth data since shadows in bins can affect the base formulae in a systematic manner which the regression will be able to catch. If the regression can't solve it then the difference in the depths is generated from a random error (i.e. insufficient precision of the camera's accuracy) and can't be solve with any other method. Instead of succeeding, I ended up proving that it was simply impossible with our current depth data – the precision was not enough to detect serving sizes. So we got a new camera that supported 16-bit depth data of much better accuracy. Once the data was ready, I would simply perform regression with pooling (trying possible sizes of pools and variants of pooling like min/avg/max to find a combination that works). Instead, we found out that the presumably "best" library for streaming we had setup did not support compression of 16-bit data and it proved incredibly non-trivial to do that. Alex and I spent 2 weeks trying to setup streaming for 16-bit data but the software (Gstreamer) was so convoluted that we decided to outsource it to the people who made Gstreamer. Finally, even their code turned out buggy and we still are waiting on support for 16-bit data streaming.


## Post-pandemic changes – The pipeline online

Then the pandemic happened. Our labeling pipeline previously needed a powerful computer to function and our labeling was done by hiring labelers and having them label on our machines. Our entire data generation system basically stopped and with it the training of our models. There was nothing we could do, and all hope was lost.

We recollected ourselves and realized that this had been coming a long time – we needed our labeling system to be fully online. It would have had to happen once we expanded anyway, and the pandemic only forced us to do it sooner. We set on search of a new labeling pipeline and found Supervise.ly.

This had everything we needed to have solid foundations for setting up an online labeling systems. Not only that, but tracking works, making the new online system significantly faster than our old one. We plan to set up a self-validating workflow of multiple labelers labeling the same data from home which is combined to result in more accurate labels than before. Without the huge setback, we never would have considered switching our entire labeling pipeline and we plan to continue work on it to develop action recognition models.

## Quick Aside

Also did I mention that at some point in the middle of all this I had to write an entire FCE compliant payment processing server that is always live and uses toast integration to support deployment to integrated restaurants seamlessly? Yeah, sometimes you need to do new things when you're working on real world systems that might be worlds away from your original task.

## Conclusion

The bottom line is that things never go as planned in real world systems. Nothing comes easy, not even a simple regression. When you take on a task as open-ended as this there are hundreds of decisions to be made at every level of the pipeline, and each suboptimal decision leads to numerous others which all build up to countless wasted work hours. A task that may initially seem simple enough to take at most an evening can take up a semester. The actual machine learning techniques required turn out to not nearly be as complex as their integration and thus classroom expertise does not translate well at all. I would highly recommend everyone interested in pursuing a career in machine learning technologies to take on an internship or research and try to apply their knowledge usefully – it will be an incredible learning experience and change the way you approach machine learning for the better.

## References and Acknowledgements


Special thanks to Evan and Alex (the co-founders of Agot.ai) for letting me take this amazing peek in what real world machine learning systems looked like.


The CVAT tool so extensively talked about is here (don't use it)

https://github.com/opencv/cvat

The better alternative that works incredibly well is Supervisely

https://supervise.ly/

Rumors say Alex is still working on 16-bit data streaming with gstreamer

https://en.wikipedia.org/wiki/GStreamer

Carnegie Mellon University