**Interviewer 1:** [00:00:00] Could you just confirm it on record that you agreed that this will be taped?

**Interviewee:** [00:00:07] Yes.

**Interviewer 1:** [00:00:08] Okay. Thank you. So our first questions are mostly about your background. So could you tell us what is your current position at your job? I would guess that you are a freelance developer, but..

**Interviewer 2:** [00:00:19] In case.

**Interviewee:** [00:00:21] Yeah, so I am a freelance developer, I also currently doing research Fellowship as a medical student at the institution.

**Interviewer 1:** [00:00:32] Okay, thank you. Could you tell us what was your overall work experience and specifically your experience in Deep Learning Systems in terms of years?

**Interviewee:** [00:00:43] Sure. So I've done approximately three years of research using deep learning in the medical imaging field, specifically radiology and mostly focus on MR and Palomar applications. I've also done some freelance work that includes other types of Medical Imaging and [inaudible] CT. And I've also done some [inaudible] Medical Imaging work that is like natural images and things like that.

**Interviewer 1:** [00:01:17] Okay. So what type of the deep learning networks you have implemented so far, by that I mean like supervised, unsupervised, reinforcement learning and Etc.

**Interviewee:** [00:01:28] Sure, mostly supervised. It's in my specialty, segmentation networks. But supervised of all types of classification and then image translation and then I have done a little bit of unsupervised and as far as the cycle Gann is technically a form of unsupervised image in image translation Network.

**Interviewer 1:** [00:01:52] Okay, thank you. Could you tell us which programming languages and Frameworks have you been using so far?

**Interviewee:** [00:01:58] Python and majority of Keras [inaudible] just pure Tensorflow.

**Interviewer 1:** [00:02:04] Thank you. So we have one general question for this interview, which is what type of bugs, problems, challenges you have faced one while developing systems that use deep learning and we're interested in all kinds of problems and bugs, we are interested in conceptual bugs, in programming bugs, in all kinds of error messages, failures that you have witnessed so far. So if you could start from there and tell us whatever comes to your mind about this, that would be great.

**Interviewee:** [00:02:34] Sure, I might just start from the beginning - just installing software. It's gotten better over the years I've been using that, I think it's matured but installing drivers for the GPUs and Cuda and Cuda toolkits [inaudible] sometimes, you know be buggy If you don't get the right version installed and then installing packages on top of that.

My biggest challenges in any deep learning project is always working with the data, to sort of organize it in a way that's usable for right deep learning networks. [inaudible] heavily runs into bugs especially Medical Imaging when you have the dicom file format, you often run to some certain challenges and reading those appropriately and sorting those.. But for example, like compressed icon images can be hard to work with sometimes and that's often what you work with when you pull images from an actual clinical workflow is the compressed.

And as far as actual writing and using models, you know, the biggest is probably just memory bugs, are memory errors, and making your model fit inside of limited memory space because that's always a concern because I could always use more memory if I wanted to. So I always have to, no matter what I'm doing, keep in mind that there's a limited amount of GPU memory and system memory to be using so that's really the number one bugs for me.

For a segmentation models you have to be concerned about the layers matching up. In a unit structure if using skip connections, the layers have to match up. So I have code written that parameterizes that for you so you can use different input sizes and it automatically will calculate those corresponding layers. So that's helpful, but it's still sometimes the concern especially if I want to edit my models. I have to be careful about getting those to work and since tensorflow uses tensors to define its variable, sometimes it's difficult to dig into exactly what's happening as opposed like using numpy arrays or something like that.

Is that good so far? Do you want me to..?

**Interviewer 1:** [00:05:12] Yes, yes. That is very useful.

**Interviewee:** [00:05:14] Okay, that's one. I am trying to think of another example. So I've been using image data generators recently, which is a way to load images off of a hard drive and use them in batches rather than loading entire datasets in a memory. That's very convenient way to work with large data sets, but that can be tricky to work with because first of all, you're using multi processing for a lot of those, you're trying to use multi-threading to make it more efficient so that your CPU loading the batch is not the bottleneck. Pre-processing for that can be just difficult to debug because it's always hard to know exactly where the error is coming from when you have multiple threads [inaudible, 1 word] the same task.

So any error that occurs in there, it can be difficult to trace back. It can also be difficult to follow bottlenecks because it's complicated code that's trying to run these processes [inaudible, 1 word] and I write custom ones for loading different types of the file formats such as dicom or numpy arrays or PNG masks simultaneously with PNG images.

So I run debugs just trying to get those work properly. I do want to do an image augmentation on those. So apply like random affine transformations every batch as they're loaded and if you use different file formats and just commonly accepted like jpegs and png's and you have to add your own support for augmenting those files and you can run into bugs doing that.

**Interviewer 1:** [00:07:17] Okay, thank you. That was really useful.

**Interviewee:** [00:07:26] Okay.

**Interviewer 1:** [00:07:27] So I wanted to ask, do usually use existing datasets to train your network or did you ever had to collect your own data?

**Interviewee:** [00:07:38] I rarely use existing datasets.I have on occasion. So I think there is a standard datasets you go to in Medical Imaging is a lot different than say like ImageNet is kind of the classic that people use for non Medical Imaging like natural image processing but in medical imaging usually you just don't get that much utility out of dataset because you have such a unique dataset that you work with in general. So for instance, if I trained a [inaudible, 1 word] segmentation model that segments [inaudible, 3 words] images and there's not really a corresponding data set that you can use for pre-training on so I didn't do any pre-training and that's why augmentation is so important because we do have a limited data set and you can't pre-train.

Another example is I am working on Kegel challenge right now that's doing chest radiograph images and I am planned using pre-training, using the NIH publicly available dataset so I have done that but it's not very common for me.

**Interviewer 1:** [00:08:58] Okay, so could you tell us about any problems that you have faced when working with existing data sets?

**Interviewee:** [00:09:09] Sure, it's always just kind of overcoming that hurdle of understanding their format because formatting and organizing data so important in deep learning you have to know exactly what you're feeding in and be sure that you're feeding in the right images and labels whatever form they take, they have to correspond to one another. So you always have to do your double checking. There is a great variety of how these datasets are formatted and even sometimes the same dataset in different forces can have different formats. And that's so you have to be able to, you know, use, you know, icon import OS and start sorting through different file paths.

The ones that come pre-installed with Keras are probably easiest to use such as MNIST dataset of handwritten digits because those are formatted perfectly as is. Whereis there is intermediate ones like just piles of jpegs and different directories that's usually pretty easy to work with and then there's things like piles and icons that are really horrendously organized and you have to sort through to find the corresponding annotations reach dicom using Header information and things like that.

**Interviewer 1:** [00:10:33] Okay. Thank you. And could you also tell us about the challenges of collecting your own data set?

**Interviewee:** [00:10:43] Sure, first of all, if you're doing a clinical study, then you have the barrier is just for the in clinical study of having IRB approval and having approval to run that data,[inaudible, 1 word] collect that data. I have to get consent from patients to collect their data or in my institution we have a retrospective [inaudible, 3 words] in place that allows us to retrospectively go back and collect data stored on clinical servers. And so that's the biggest hurdle. I think the challenge to take care of.. A big thing in the medical imaging field is you have to search through data to know what subjects you want. So with a project such as one I'm currently working on looking at liver cancer and trying to classify images whether they do or do not have liver cancer. First you have to collect a large number of subjects that have liver cancer and large number that don't have liver cancer and that's no trivial task to get a list of the patients that have that, to find where it is. And then attitude [inaudible, 2 words] appropriately. So that's [inaudible, 2 word] about the hurdles to go through.

**Interviewer 1:** [00:12:01] Okay. Okay, thank you. So you mentioned that it's very important to format the data appropriately and I wanted to ask whether you ever had cases when you missed some pre-processing step and you run into trouble when training your model and you went back and added some pre-processing step and if yes, could you mention which one?

**Interviewee:** [00:12:25] Sure, well, normalization is one that comes to mind. That's something I'm constantly tweaking because you don't just have to have like they're arrays in the right shape and everything that [inaudible, 1 word] themselves. Do you have to normalize them too a certain range and it can vary so much based of what type of evidence you working with it. So I go back and change that constantly or sometimes I normalize [inaudible, 2 words] range 0 to 1 for all values and then I'll change it to normalizing by mean and standard deviation, mean of zero and a standard deviation of one or maybe I'll do it by percentiles that's a good example. Something I always am tweaking.

**Interviewer 1:** [00:13:12] Okay. Okay. Thank you. And do you remember any other kinds of problems related to training data that you face? Like, I don't know, like not enough data, that training data is wrong in some way and etc.

**Interviewee:** [00:13:27] Yeah, I never have enough data in medical imaging because you're always limited by how much data you can collect. In general your often limited by that so that's certainly a challenge I always encounter and that's again, why image augmentation is the really important part of a successfully training a model. And there's certainly been times where I had data, It's out of order. Actually. I just fixed a bug yesterday where my data was being loaded in with channel access first instead of last. So that actually was a silent bug and it was running and I actually don't understand how it even ran but it did and then I went back and checked something and I realized that the channel axes was in wrong order and so the data was [inaudible, several words] fine, even though the results weren't good. So [inaudible, 3 words] they had an effect since I had to change the axis order.

**Interviewer 1:** [00:14:39] Okay. Thank you. So I wanted to ask about model structure. So you mentioned that you have problems for the layer sizes to match that you have a code that helps with that. Have you ever faced any other problems related to the role model structure?

**Interviewee:** [00:15:01] Sure. So one problem I've worked with is trying to generalize a model to different input sizes. You can train them [inaudible, on any size?] But then evaluate images of any size and that comes down. Usually the tool uses global average pooling as the last layer so you can take input image of any size that condenses down to some array size. And then when you do the global average pooling is just a 1 by 1 array and I do classification from there. So that allows that flexibility but that does introduce some question as to how best to use that layer and in a one order and whether you should use an average pooling or Max pooling or there's these attention gates that are being used now that say that they're better than average pooling. So that that is a question of model structure where you should be using that.

I also I've done some GANs or image to image translation and the structure of those models can be very challenging to appropriately put together and understand the best way to link the discriminator and the generator models together.

**Interviewer 1:** [00:16:22] Thank you. And another question we have is about hyperparameter tuning. Did you ever had problems because of the very wrong value of some hyperparameters.

**Interviewee:** [00:16:34] Yeah. Actually, I've tried a little bit of 3D image processing which is very challenging to do because there's a number of parameters is so large usually it doesn't fit into memory. And so then you are forced to use small batch sizes, which I think makes the training unstable or a combination of just small batch sizes and really really large parameter spaces. That tends to explode [inaudible] hyperparameters such as learning rate. If you don't use loss regularization or weight regularization. And usually the only way to train the models successfully I found is to use grade [inaudible, 1-2 words] or radiant normalization. Otherwise, the loss just explodes almost instantly, so that's a good example.

**Interviewer 1:** [00:17:31] Okay. Thank you. And about the loss functions. Do you usually use a pre-defined or a custom written loss function? And did you ever had any problems because of the wrongly selected loss function or because the implementation of the loss function was not correct.

**Interviewee:** [00:17:49] Sure, so three years ago when I started doing this. The dice loss function was not quite as common as it was today. So when people are doing segmentation, which wasn't uncommon at all, people are still using like binary cross entropy or categorical cross entropy, which is a far inferior loss function to dice. But that is also something you had to implement by yourself. Nowadays it's very standard, I think, and you can find implementations very easily and that it, but it is still not included and for example the the default Keras API you have to define it yourself and the difference is its dramatic that when you go from doing something like binary cross-entropy to dice. So that's a really good example.

I've also tried to write other custom loss function. For example, in image image translation a really common metric you use to determine how images are similar to each other as SSIM, [inaudible] index and so I thought well, this is such a good metric for that. I [inaudible] and train on it.

That's all really what I'm using to evaluate these images and [also?, although?] very challenging [inaudible] ultimately I gave up on it because I do get on but it wasn't effective [inaudible] compared to just L1 loss. I have also tried [inaudible] because that was something I was using to [vary?, evaluate?] images.

It's nice to just copy the percent error overlap between different segmentation. And that was again effective metric for evaluating images to use applications [inaudible], but very ineffective loss function to use for training the model all as I did not have proper gradients to train the model so dice [inaudible] training [inaudible] evaluating.

**Interviewer 1:** [00:20:02] Okay. Thank you and about hardware and where did you train your models? And did you ever encounter problems related to hardware?

**Interviewee:** [00:20:14] Sure, the most training I've done is on local machines that the institution purchased. They are Alienware machines from Dell that have all the hard work was already installed. We use top line GPU cards to do that and unfortunately the support for those machines, all the software support, comes down to that was my research group. So we end up spraying those ourselves and so it's not technically a hardware issue but solving drivers and things like that can be challenging as I mentioned a beginning.

I've also used Google Cloud compute or more recently and I tend to use it for freelance work as it's very convenient to use, log in remotely. Handling data on there can be challenging but not having to worry about the hardware side of things. Because it makes [inaudible] just works every time you log in that that's a huge advantage.

So yeah, I haven't really had too many specific hardware issues other than just drivers for hardware. As long as the drivers are working on the software is running then the bugs are mostly just are software bugs.

**Interviewer 1:** [00:21:39] Okay, thank you. So I have like one general question. Are there any bug types or error messages that you think, you know, you get very often?

**Interviewee:** [00:21:52] Out of memory on the GPU that's definitely like if I went back and [inaudible] the number one just because I try to find the limit of how big of a model I can use. Another bug is the silent bug with no error message at all and your model just doesn't work and then you have to figure out why it's all whether it's over fitting or it's just not learning or something else. It's just not working, It's super super slow and you don't know why it's because you have a bottleneck somewhere that doesn't have any error messages associated with it. So other than that, I feel like I [inaudible] with a very diverse set of bugs on a daily basis and I'm constantly Googling and then fix them because it's always a new bug. It's never something I've seen before thath often, something I've seen before occasionally.

**Interviewer 1:** [00:22:49] Okay. And do you remember any specific bug which was particularly nasty and you know took a lot of your time. If not, that is alright.

**Interviewee:** [00:23:10] One second..after my model is to you know, do a Google Search and look on stack exchange and if I don't find it then try to find a workaround. So most of my time is spent just writing the code and trying to get to run because [inaudible].

**Interviewer 1:** [00:23:36] Okay. Thanks. Out of curiosity you told that you're training this models that are supposed I guess to detect cancer on livers. So what is the accuracy of that kind of model?

**Interviewee:** [00:23:56] Right now It's not very good. And that's something I'm still in development of. Its doing slice wise classifications or saying whether there is a tumor present on that image slides [inaudible] somewhere around like 60% but I hope to have it somewhere close to 80 to 90% accuracy.

**Interviewer 2:** [00:24:19] One other question, maybe you remember any other issues related to the GPUs, if you use multiple GPUs or some errors related to concurrency?

**Interviewee:** [00:24:32] Sorry, I didn't quite catch all of that about using multiple GPUs.

**Interviewer 2:** [00:24:35] No, just any problems related to GPUs, to concurrency that you remember besides out of memory problem you already told us about.

**Interviewee:** [00:24:45] Sure. I mean, sometimes the driver unexpectedly crash and then they have to be reinstalled and [inaudible]. Yesterday I realized I accidentally installed software that did not support the GPU [inaudible] the wrong version of it. And so I had to go back and reinstall the version of that support the GPU. I don't use multiple GPUs very often just because it's a lot simpler to use one and...Yeah, usually fortunately the drivers are installed correctly. This software is installed correctly it runs. So I think that's something [no?, to?] credit to the packages and frameworks that we use.

**Interviewer 2:** [00:25:40] Okay, thank you for okay.

**Interviewer 1:** [00:25:43] Thank you, thanks a lot. It was really very useful and.

**Interviewee:** [00:25:48] Great. Happy to help.

**Interviewer 1:** [00:25:50] Okay. Have a nice day. Bye bye.