Research Statement

Ragav Venkatesan

I am an applied and research scientist in machine learning and computer vision whose work is always driven by creativity and simplicity with an emphasis on practical applications.

1 Large-Scale ML Platforms

As a research scientist at AWS AI Labs, I was a member of the team that launched AWS Sagemaker, largely credited to be the world's first comprehensive, commercial, end-to-end machine learning platform. Working on such a product allowed me to focus on inventing technologies that scaled to a massive customer base and broad use-cases. Since the launch of Sagemaker, I was involved in the development of several Sagemaker Computer Vision algorithms, with primary ownership stake in AWS Sagemaker Object Detection Algorithms and AWS Sagemaker Semantic Segmentation Algorithms. I also worked on areas spanning large-scale distributed training to privacy-preserving ML [9]. Machine learning products at-scale opens two opportunities of research: 1. Transferability of solutions and 2. Cost-efficiency.

Consider the annotation service AWS Sagemaker GroundTruth. In such a product, the only practical way to offer domain-adaptation solutions is to have high transferability. With hundreds of customers, each with their own data sources, it is not a scalable strategy to produce bespoke domain-adaptation solutions that is typically the norm in this area of research. In the era of deep learning and generative models, simpler techniques such as stochastic neighborhood (SNE) embedding are often cast away. While building GroundTruth, we were able to demonstrate that techniques such as SNE are not only capable of being state-of-the-art in most benchmark datasets, but are also highly transferable across a variety of use-cases [12]. Similarly, most customers choose to warm-start their model training using pre-trained weights. It is not often clear which pre-trained artifact trained with which dataset are transferable. Early indications from my work indicate that there are possibilities of having canonical pre-trained models for images, as we are witnessing in the natural language domains with the Generative Pre-trained Transformer series of models[7].

Analogously, consider ML models for a product such as Amazon Alexa. These models need to deployed to devices that are often USB-bus powered and need to be highly transferable across different hardware. Similarly, for a corporation like GE Healthcare, it is very important to produce models that are highly portable for their low-powered devices and are highly transferable to work across multiple hardware. I worked on RL-based transferable model compression algorithms for such use-cases. This work was designed to learn which channels or attention heads to prune using a reward mechanism, specifically designed to be transferable to unseen models, out-of-the-box [10]. I was able to launch this work as product in the form of model compression algorithms using reinforcement learning that became a significant highlight of a keynote address at re:Invent 2018, the premier event of Amazon Web Services, where the Director of Machine Learning at GE Healthcare demonstrated how they were able to use this product to install compressed deep models into their medical equipment. A related work on the other hand using similar approaches to search for network architectures efficiently, gave us largely negative results, casting aspersions on the practicality of NAS-like algorithms for large language models such as BERT[4]. While this result is not useful to build into a product, the insights we gained directed the course of further research. These works are largely extensions of an earlier research project, where I was able to train student-teacher models with a primary focus on model compression[8].

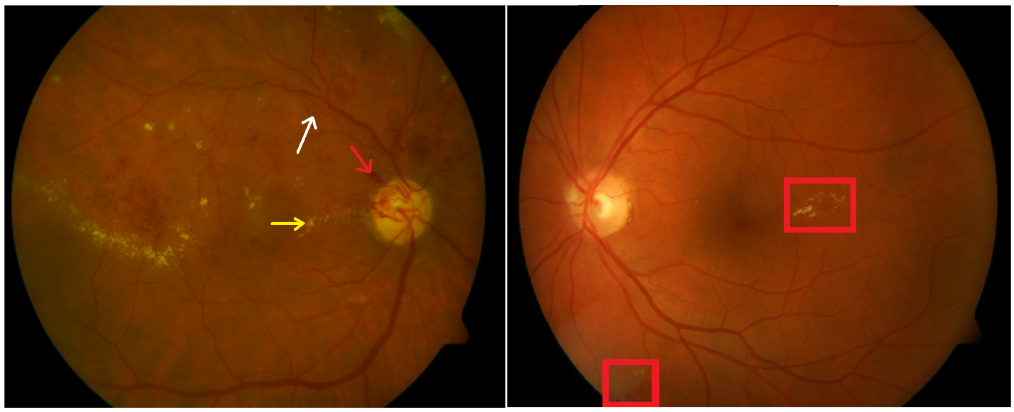
The work of an ML platform does not end with training. Not only are these models expected to efficient, they are also expected to be fine-tuned in a federated manner after being deployed in the wild. Uncompromising incremental training is therefore an important component of such a platform and provides a comprehensive, closed-loop solution. Most incremental trainers in literature compromise on data privacy, where they do not maintain a strict data privacy membrane between the original data source and the incremental sites. I was able to show that simple data encoding techniques with creative uses of generative models is one possible way to migrate information about the original source distribution to the incremental site, while maintaining strict data privacy [11]. A patent for this work was also issued, that demonstrates how to build this work into a closed-loop training system such as Alexa [9].

2 Multiple-Instance Learning

While, machine learning at a platform-level focuses on transferability to disparate use-cases, most machine learning is applied on particular problems. Growing up in a region of the world which is prone to several dietary health issues while simultaneously insisting on retaining a culture known for its decadent diet, I chose to focus my application domain on identifying diabetic

retinopathy during my doctoral study. My work largely focused on simple, fast and efficient non-parametric multiple-instance learning methods applied toward clinically-relevant medical-imaging solutions.

Multiple-instance learning (MIL) is a setting where labels are provided only for a collection of instances called bags. There are two types of instances: negative instances, which are found in either negative bags or positive bags, and positive instances, which are found only in positive bags. While a positive bag should contain at least one inherently positive instance, a negative bag must not contain any positive instances. In MIL, labels are not available at the instance-level. It is interesting to note however that the label space is the same for both at the bag-level and at the instance-level.



MIL is an ideal set-up for medical image-based pathology classification and lesion detection/localization, where an image is labeled pathological just because of one or a few lesions localized to small portions of the image. Medical images collected in a clinical setting may readily have an image-level label (either normal or various levels of pathology) while lacking the exact location of the lesion(s). The figure above illustrates such an example: color fundus images of eyes affected with different pathologies of diabetic retinopathy. It is easy to notice that, although majority of the image looks normal, a small retinal landmark is enough to alter the label of the image from normal to pathological. In a MIL formulation for this problem, each image can be considered a bag and patches of images can be considered instances.

While machines are quite capable of being pathologists when trained appropriately and tested thoroughly, medical regulations and ethics prevent the fully-automated use of technology for diagnosis. Assistive technology, on the other hand is extremely useful to hasten the diagnostic process. The time of nurses and doctors are very limited and therefore need to be used highly efficiently. This was the lesson learned during the global pandemic of 2020, but to the researchers working in the space of medical-imaging, this was apparent from the outset. The simplest solution is to develop a clinically-relevant retrieval mechanism. [2, 1, 3]. A clinically-relevant retrieval system is one where not only is the retrieval expected to be precise on its overall label, but also on the relationship between the labels of the instances themselves. Not only do the retrieved images be similar in pathology to the queried image, but also are the diagnosis. This implies that the medical practitioner gets access to previous therapeutic applications and followups, helping make an informed judgment for the case at hand.

While the clinically-relevant retrieval system that we built assisted healthcare providers with immediate access to similar prior diagnoses, it still did not help with current diagnosis. This was because the images themselves are typically large and the pathological areas too small. To alleviate this, we built fine-lesion detection and classification techniques by inventing special multiple-instance methods. These methods are expected to run on smaller, often older devices and therefore need to be simple and highly efficient. My work in this area pushed multiple-instance learning research into the yet, unseen direction of simpler, non-parametric learning techniques particularly in the diabetic retinopathy feature spaces [5, 6].

3 Future Directions

Cost-efficient and Environmentally-Conscious ML.

Machine learning is a massive industry. At a platform scale cost is becoming a priority. An obvious avenue for avoiding unwanted compute usage is in reduced training time. Saving compute usage and terminating wasteful training jobs directly helps reduce the global carbon footprint left behind by the ML community. While products such as idle-instance termination have enabled AWS and other ML platform service providers save compute costs by identifying machines that are not performing useful tasks, this is merely an entry point for such technology. Early-stopping techniques in literature currently focus on avoiding over-fitting. They are convergence detectors. While these can be used as termination logic, they only work on well-behaved training jobs. There are two obvious scenarios, where a general, platform-scale, early-stopping technique could further compute savings.

1. **Scenario 1: AutoEpochs: Early stop models were the best is already past.** In a typical ML platform more than two-thirds of training jobs have reached the minimum generalization performance earlier than when they were scheduled to terminate, either manually or using standard patience-based early-stopping techniques. A quarter of these

jobs usually could have been stopped within the first 10% of iterations and about half could have been stopped in the first 50% of iterations. One way to mitigate this wastage is by learning from the data of the platform itself and building meta early-stopping technologies. These should essentially predict if the minimum generalization performance of the training job is in the past. If so, the training job could be stopped.

2. **Scenario 2: Stopping futile jobs.** Most training jobs in an enterprise-level ML platform are futile (will never reach expected loss value). More than half the training jobs ends with an user interrupt. Scientists use their intuition and often desperation to terminate training jobs that appear to converge at a loss value more than they expected. Not only is this compute in-efficient, this also implies that most scientists spend a significant portion of their working hours babysitting training jobs leading to lowered productivity. Using past training jobs, we can build models that will be able to estimate the convergence of the current training job. We could use a meta-model to terminate futile jobs early and save massively on compute.

Most mature ML platforms amass large quantities of data. This data is often cheap and readily-available. This data is also easily-simulated. I am focused on leveraging meta-learning techniques to be able to identify futile training jobs and converged models.

Techniques to find transferable, parameter-constrained network architecture.

Most neural network architecture search techniques today are very compute-inefficient and often do not generalize to multiple domains. Bespoke architecture also, do not come with pre-trained weights out-of-the-box, requiring them to be trained from scratch. This implies that these searched architectures are not useful in cases where the datasets are nominal, as is often the case of most customers using ML platform services. Several works including one from my own research have already demonstrated that random architecture work as well as searched ones [4]. It is quite clear that there exists some synergy between model compression using RL such as the out-of-the-box pruned networks and architecture search since these both use similar reward functions and similar algorithmic formulations. I am focused on combining these two areas of research and studying the ability to produce transferable, pre-compressed network architectures.

If AI is the new electricity, we have only just started using it to power light bulbs.

References

[1] Parag S Chandakkar, Ragav Venkatesan, and Baoxin Li. Retrieving clinically relevant diabetic retinopathy images using a multi-class multiple-instance framework. In *Medical Imaging 2013: Computer-Aided Diagnosis*, volume 8670, page 86700Q. International Society for Optics and Photonics, 2013. 2

[2] Parag S Chandakkar, Ragav Venkatesan, and Baoxin Li. Mirank-knn: multiple-instance retrieval of clinically relevant diabetic retinopathy images. *Journal of Medical Imaging*, 4(3):034003, 2017. 2

[3] Parag S Chandakkar, Ragav Venkatesan, Baoxin Li, and Helen K Li. A machine-learning approach to retrieving diabetic retinopathy images. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 588–589, 2012. 2

[4] Ansel MacLaughlin, Jwala Dhamala, Anoop Kumar, Sriram Venkatapathy, Ragav Venkatesan, and Rahul Gupta. Evaluating the effectiveness of efficient neural architecture search for sentence-pair tasks. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 22–31, 2020. 1, 3

[5] Ragav Venkatesan, Parag Chandakkar, and Baoxin Li. Simpler non-parametric methods provide as good or better results to multiple-instance learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2605–2613, 2015. 2

[6] Ragav Venkatesan, Parag Chandakkar, Baoxin Li, and Helen K Li. Classification of diabetic retinopathy images using multi-class multiple-instance learning based on color correlogram features. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1462–1465. IEEE. 2

[7] Ragav Venkatesan, Vijetha Gatupalli, and Baoxin Li. On the generality of neural image features. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 41–45, 2016. 1

[8] Ragav Venkatesan and Baoxin Li. Diving deeper into mentee networks. *arXiv preprint arXiv:1604.08220*, 2016. 1

[9] Ragav Venkatesan and Gurumurthy Swaminathan. United states patent office 10567334b1: Domain mapping for privacy preservation. 1

[10] Ragav Venkatesan, Gurumurthy Swaminathan, Xiong Zhou, and Anna Luo. Out-of-the-box channel pruned networks. *arXiv e-prints*, pages arXiv–2004, 2020. 1

[11] Ragav Venkatesan, Hemanth Venkateswara, Sethuraman Panchanathan, and Baoxin Li. A strategy for an uncompromising incremental learner. *arXiv preprint arXiv:1705.00744*, 2017. 1

[12] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. *d*-sne: Domain adaptation using stochastic neighborhood embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2497–2506, 2019. 1