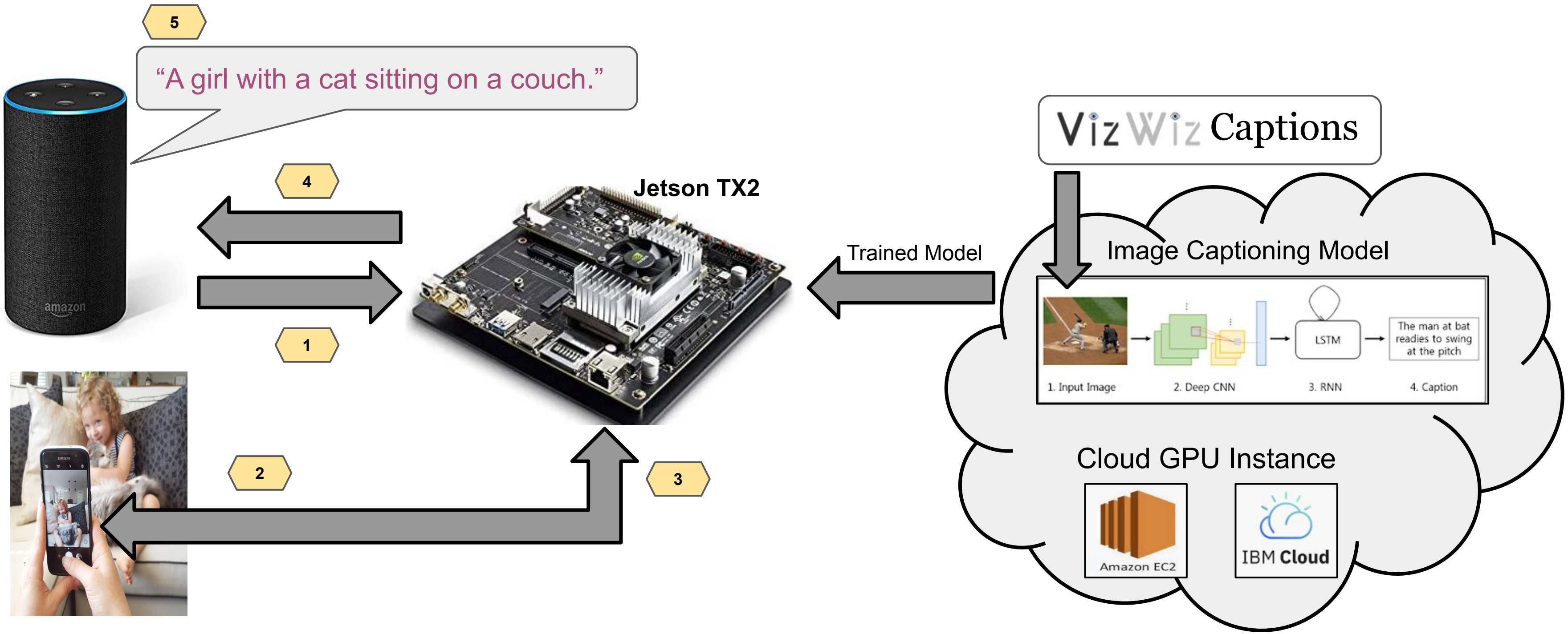
Alexa!
What do you see?

Padmavati Sridhar
Shaji Kunjumohamed
Shwetha Chitta Nagaraj

W251 Final Project Presentation - August 4th 2020

"Alexa! What do you see?"



Dataset - VizWiz Captions

some basil leaves in a container on a counter

a bottle of spices in a plastic container laying on a surface.

a green and white plastic condiment bottle containing basil leaves.

its is a basil leaves container its contains the net weight too.

black counter with canisters, kettle and can of soda.

a black tin of coca cola placed on a black surface

a kitchen counter the various items on top including a can of coca-cola, metal containers, and a teapot.

a black can of coca cola zero calorie soda is on the counter near the coffee maker.

a can of coca cola on a counter is shown for when one can use a nice, cold drink.

image is a can of crushed tomatoes in view.

a price chopper branded can of crushed tomatoes

a can of crushed tomatoes in puree from price chopper.

a can of crushed tomatoes sitting on a beige colored counter.

a can of crushed tomatoes are on a brown surface. the tomatoes read crushed tomatoes on the brand.

- Curated by University of Texas, Austin

- **1st publicly available dataset - images taken by the visually impaired**
  - To meet their daily needs about things around them
  - Presents a **real-use case** for our project - Captioning for the visually impaired!

- Crowdsourced captions through Amazon Mech. Turk
- 31, 161 Train+Val and 8000 Test **Images**
- 157, 905 Train+Val and 40,000 Test **Captions**

- An image annotated with 1 to 5 captions
  - Images with more complex scenes
- Average caption length: 13
  - Greater than that of MS COCO, Flickr etc.
  - Larger vocabulary
    - More nouns, verbs and adjectives

- **VizWiz Image Captioning Challenge**
  - Evaluated on Test images
  - Using CIDEr-D score

Image Captioning Architectures

Show and Tell: A Neural Image Caption Generator

Oriol Vinyals, Google, vinyals@google.com
Alexander Toshev, Google, toshev@google.com
Samy Bengio, Google, bengio@google.com
Dumitru Erhan, Google, dumitru@google.com

Multimodal Neural Language Models

Ryan Kiros RKIROS@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard Zemel ZEMEL@CS.TORONTO.EDU
Department of Computer Science, University of Toronto
Canadian Institute for Advanced Research

From Captions to Visual Concepts and Back

Hao Fang*, Saurabh Gupta*, Forrest Iandola*, Rupesh K. Srivastava*
Li Deng, Piotr Dollár†, Jianfeng Gao, Xiaodong He
Margaret Mitchell, John C. Platt‡, C. Lawrence Zitnick, Geoffrey Zweig
Microsoft Research

DenseCap: Fully Convolutional Localization Networks for Dense Captioning

Justin Johnson*, Andrej Karpathy*, Li Fei-Fei
Department of Computer Science, Stanford University
{jcjohns, karpathy, feifeili}@cs.stanford.edu

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.C
Jimmy Lei Ba JIMMY@PSI.UTORONTO.C
Ryan Kiros RKIROS@CS.TORONTO.EI
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.C
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

Peter Anderson[1]*, Xiaodong He[2], Chris Buehler[3], Damien Teney[4]
Mark Johnson[5], Stephen Gould[1], Lei Zhang[3]
[1]Australian National University [2]JD AI Research [3]Microsoft Research [4]University of Adelaide [5]Macquarie University
[1]firstname.lastname@anu.edu.au, [2]xiaodong.he@jd.com, [3]{chris.buehler,leizhang}@microsoft.com
[4]damien.teney@adelaide.edu.au, [5]mark.johnson@mq.edu.au

Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

Jiasen Lu[2]*†, Caiming Xiong[1]†, Devi Parikh[3], Richard Socher[1]
[1]Salesforce Research, [2]Virginia Tech, [3]Georgia Institute of Technology
jiasenlu@vt.edu, parikh@gatech.edu, {cxiong, rsocher}@salesforce.com

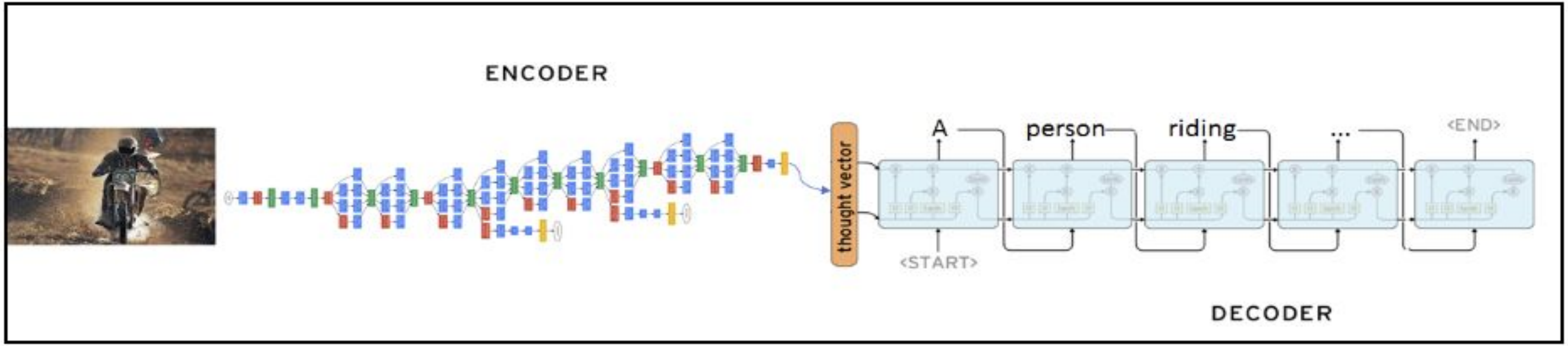
SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning

Long Chen[1], Hanwang Zhang[2], Jun Xiao[1]*, Liqiang Nie[3], Jian Shao[1], Wei Liu[4], Tat-Seng Chua[5]
[1]Zhejiang University [2]Columbia University [3]Shandong University
[4]Tencent AI Lab [5]National University of Singapore

Attention on Attention for Image Captioning

Lun Huang[1], Wenmin Wang[1,3]*, Jie Chen[1,2], Xiao-Yong Wei[2]
[1]School of Electronic and Computer Engineering, Peking University
[2]Peng Cheng Laboratory
[3]Macau University of Science and Technology
huanglun@pku.edu.cn, {wangwm@ece.pku.edu.cn, wmwang@must.edu.mo}, {chenj, weixy}@pcl.ac.cn

Encoder-Decoder



Source: https://www.themtank.org/multi-modal-methods

Encoder:

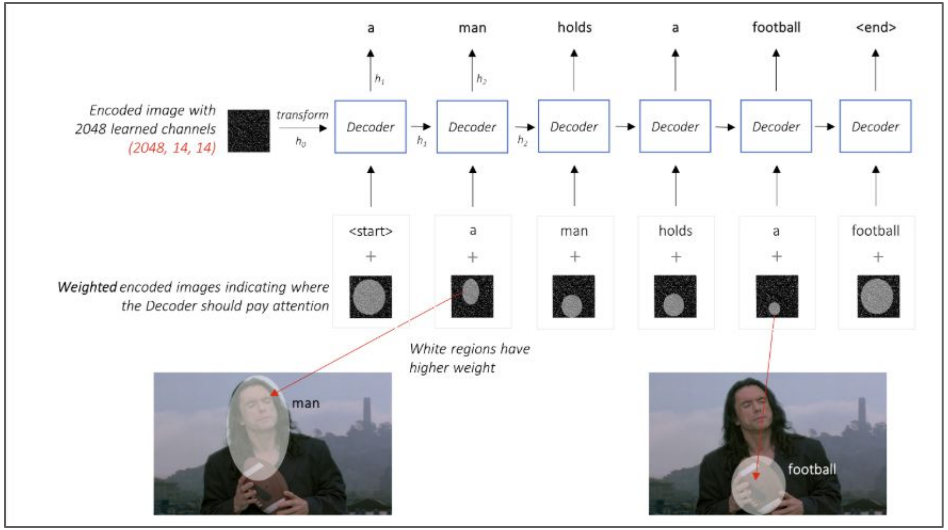
- Pre-trained Deep CNN like VGG16, Inception V3, ResNet variants on image classification tasks
- Takes an input image -> Generate feature representations (fixed length vectors): objects, attributes, regions
- Last hidden layer used as input to decoder

Decoder:

- Language model to generate captions
- Could be LSTM, GRU, Bidirectional LSTM etc.
- Next words are generated based on current time step and previous hidden state till end of sequence

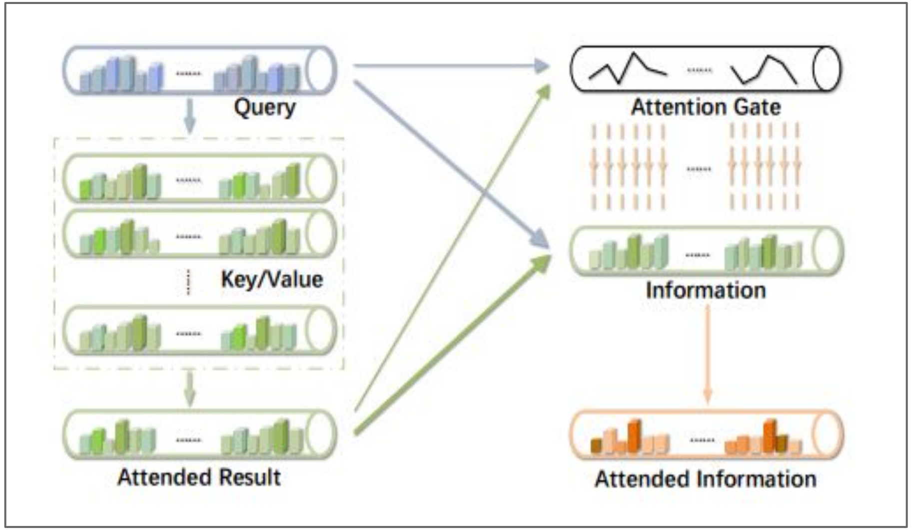
Attention on Attention(AoA)

Attention Mechanism



Source:https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning

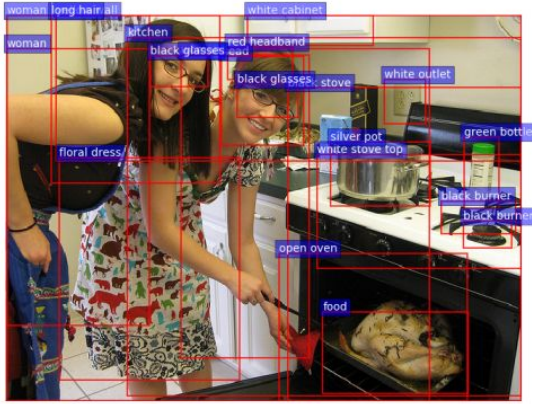
Attention-on-Attention (AoA)



Source：https://arxiv.org/pdf/1908.06954.pdf

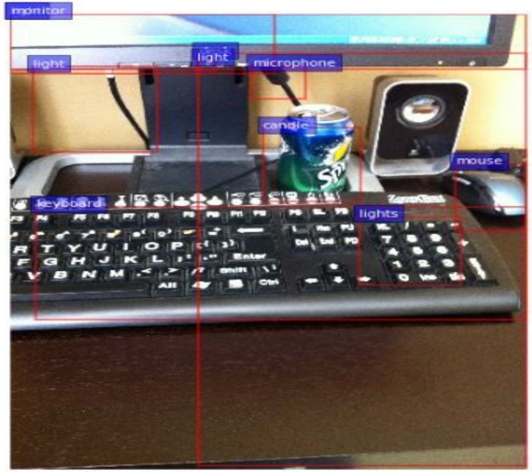
AoA Net

- **Image features** are extracted using a **pre-trained Faster R-CNN**(ResNet-101) model on ImageNet and Visual Genome datasets using **Bottom-up mechanism**
    - Each image feature **encodes a spatial region** of the image
    - The spatial regions consist of identifying instances of objects belonging to certain classes and **localize them with bounding boxes**(pic. on top-right).
    - Each bounding box is labeled with an **attribute** class and an **object** class.
    - Built with an older version of **Caffe**(on Ubuntu 14.4) which could not be compiled due to various dependencies with newer versions of python libraries.
    - Partially extract features vectors using Facebook's AI Research called **Detectron2**(previously called Pythia).
        - This <u>does not</u> extract the attribute classes(pic. on bottom-right).
- Used the provided pre-extracted image vectors from bottom-up for VizWiz to train the model.
- **Decoder:**
    - LSTM with 2 layers
        - 1024 hidden nodes each
- **Model parameters:**
    - Loss function: Cross-entropy
    - Optimizer: Adam with a learning rate(LR) of 2e-4
    - LR decay after 0.5 every 3 epochs
    - Batch Size: 20, Epochs: 25
- Self-Critical Sequence Training(**SCST**) optimization to optimize CIDEr-D score
    - Epochs: 40, LR=2e-5,
- **Evaluation:**
    - Beam Search: 3, Batch Size: 100
- **Trained from scratch** on VizWiz and **also fine-tuned** model with MS COCO dataset.

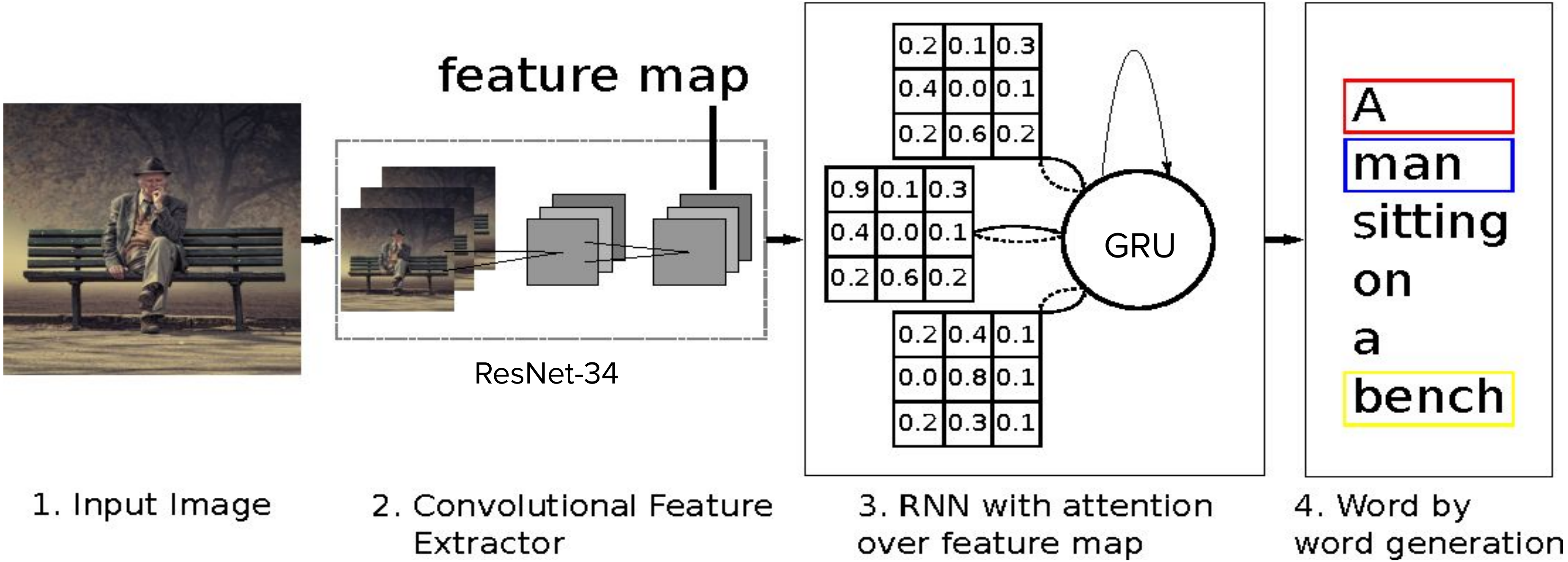


Features from Bottom-up(Caffe model)

Features from Detectron2



Show Attend and Tell



Source: https://arxiv.org/pdf/1502.03044.pdf

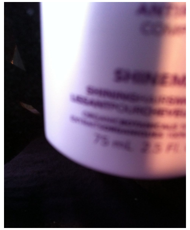
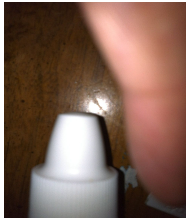
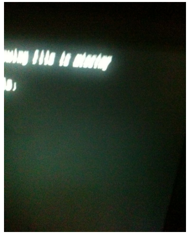
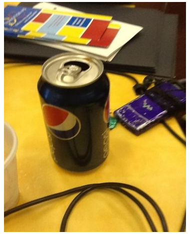
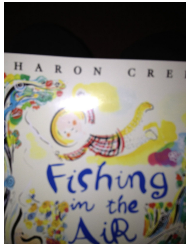
Training and Inference - Overview

- Pytorch based Model
- Final model
    - For encoder - Used resnet34 as pre-trained model.
    - Decoder - Used GRU.
    - Used teacher forcing, Beam search to generate reasonably accurate prediction
- Mixed COCO 2014 image caption dataset and Vizwiz dataset for getting better results.
- Inference is done in Jetson.
    - Model selection depended on training time@cloud, Jetson RAM utilization and GPU activity
- Fast.ai framework is used in model development.
    - Leveraged pre-defined machinery to build efficient training structure.

Training highlights

- Each image had multiple captions - randomly selected a caption per image epoch (Resulted in low BLEU metric, but yielded less overfitting)
- Made use of one cycle policy to figure out learning rate.
- Multiple techniques were used to improve training time.
    - Image stored as pickle files, (Local storage of picke in SSD)
    - number of workers.
    - To save time, store processed training data and setup files in a nfs mount.
- Dask framework - worked best for large scaled parallel processing/wrangling.
- Batch size adjusted for max GPU utilization, had to re-adjust for different instances of same type of machine to avoid CUDA errors.

Generated Captions(test sample)

	**AoA(Scratch)**: a bottle of lotion is on top of a table **AoA(FT):** a bottle of lotion is on top of a table **SAT:** a white bottle with a white label on it		**AoA(Scratch):** a hand holding a white bottle with a white cap **AoA(FT):** a white bottle is on top of a wooden table **SAT:** a person is holding a bottle of lotion in their hand
	**AoA(Scratch):** a computer screen with white text on a black background **AoA(FT):** a green screen with the words signal on it **SAT:** a computer screen with a black background and white text .		**AoA(Scratch):** a jar of something is on top of a table **AoA(FT):** a mug with a blue cap on top of a table **SAT:** the top side of a red and white food container with a red and white label
	**AoA(Scratch):** a can of pepsi sitting on a desk with a desk **AoA(FT):** a can of coke is on top of a table **SAT:** a can of coca cola sitting on a table next to a computer keyboard .		**AoA(Scratch)**: a children ' s book with colorful and blue and yellow and blue **AoA(FT):** a children ' s book with a colorful butterfly on it **SAT:** a birthday card with a cartoon character on it

AoA(Scratch) : AoA Net with VizWiz only **AoA(FT):** AoA Net with VizWiz+COCO **SAT:** Show, Attend and Tell with VizWiz+COCO

Evaluation Scores

Rank	Team	Bleu1	Bleu-2	Bleu-3	Bleu-4	ROUGE-L	METEOR	CIDEr	SPICE
1	IBM Research AI	-	-	-	-	-	-	81.13	-
2	SRC-B-VCLab	-	-	-	-	-	-	72.89	-
3	ABurns(Boston Univ.)	-	-	-	-	-	-	64.27	-
-	Baseline(AoANet)	65.91	47.77	33.68	23.41	46.56	20.00	59.77	15.11
-	**Team SSP** (AoA Net Scratch)*	65.70	47.15	32.97	22.80	46.38	19.79	**59.56**	14.88
-	**Team SSP** (AoA Net Fine-tuned)*	65.68	46.94	32.85	**22.83**	**46.46**	19.82	58.50	**14.91**
-	**Team SSP** (Show, Attend And Tell)*	58.58	39.57	26.49	17.71	40.75	16.57	37.18	11.06

For Inference on Jetson

* - Post Competition Scores.   Scores for other evaluation metrics not available.   Post Competition Ranks not available.

Inference highlights

- Inference on the Jetson TX2 → leveraged edge architecture for POC end-to-end system

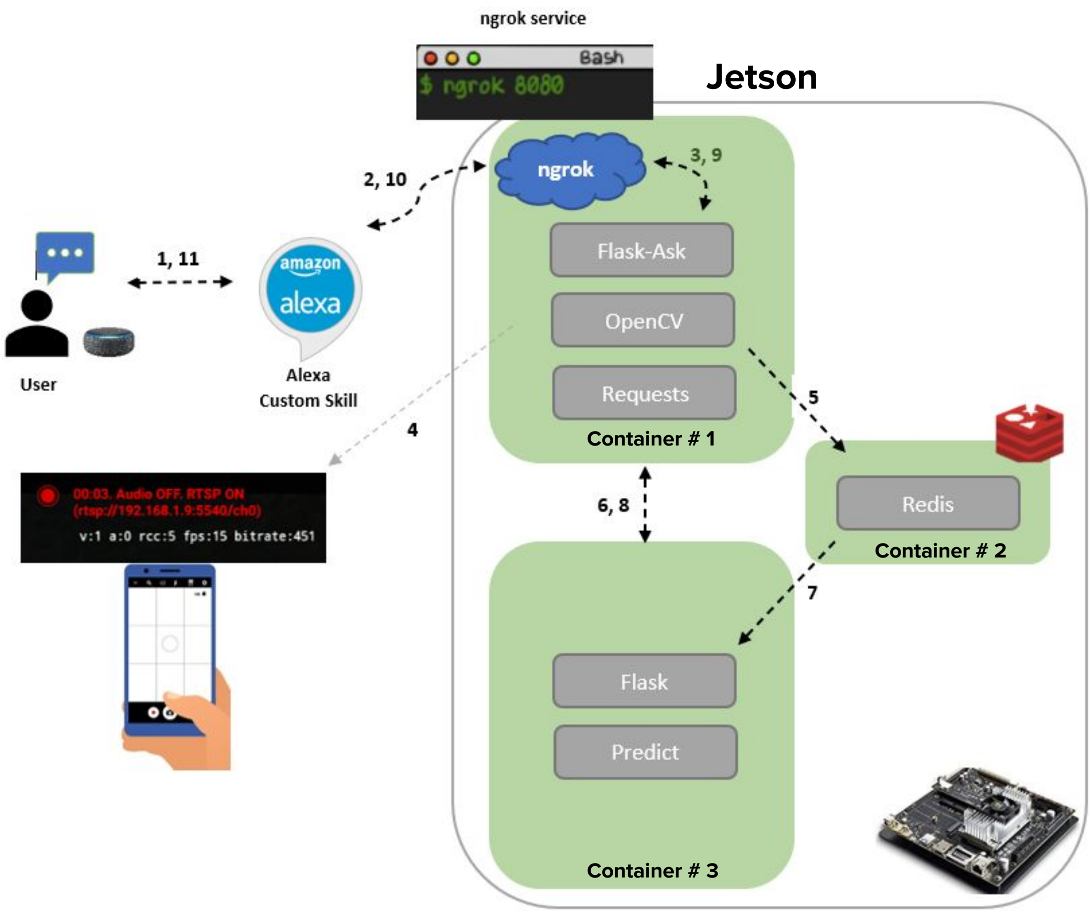
- Inference using CPU yielded less accurate captions vs GPU.

- Fast.ai learner worked on Jetson flawlessly.

- Used nvidia pytorch docker image for inference. Precompiled wheels to speed up Docker build.

- Tried on demand inference in Jetson Vs continuous inference.
    - On demand is currently implemented in final demo
    - Continuous inference (5 fps) was an overkill and not practical for use case

Edge Architecture



1. User invokes Alexa custom skill, Echo dot processes audio and sends it to Alexa Cloud
2. Alexa Cloud sends message to ngrok endpoint (setup to be *www.251final.com*)
3. ngrok forwards JSON to Flask-Ask (processes skill intents)
4. OpenCV takes a picture on-demand and encodes it as a JPG into memory buffer
5. Buffer saved in base64 encoding in redis
6. Request sent to prediction container via Flask endpoint to make a prediction
7. Image retrieved from redis and decoded, prediction model invoked, image is captioned
8. Caption sent back to ngrok container
9. Caption sent back via Flask-Ask to ngrok
10. ngrok sends caption back to Alexa Cloud
11. Alexa Cloud sends caption back to Echo Dot

Demo

Everyday items:
laptop, can of soup, coffee mug, teddy bear

Kitchen:
refrigerator, stove

Other:
vitamins, can of water

Future Improvements

- Productionalize edge architecture
  - Nginx, Kubernetes, security
  - Multiple users
  - Latency
  - Connectivity
  - Use-case can determine CPU vs. GPU for inference
- Pre-processing of Images for text processing (~63% contain text):
  - So it would be ideal to use an OCR mechanism like [scene text detection model.]
  - Image augmentation techniques like rotation coupled with scoring the captions generated at different image angles.
  - Apply blurring filters to only in-focus images for better object detection
- Try a different pre-trained model for image feature extraction which is trained on pictures similar to VizWiz(like Instagram photos)
- Refine AoA Net
  - Use a different RNN like Transformer in the Decoder.
  - Use [Bayesian-SCST] for fine tuning.

Scene text detection

Thank you!

References

- VizWiz Captions: https://vizwiz.org/tasks-and-datasets/image-captioning/
- A Comprehensive Survey of Deep Learning for Image Captioning: https://arxiv.org/abs/1810.04020
- Multi-Modal Methods: Recent Intersections Between Computer Vision and Natural Language Processing https://www.themtank.org/multi-modal-methods
- Show and Tell: A Neural Image Caption Generator: https://arxiv.org/abs/1411.4555
- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention: https://arxiv.org/abs/1502.03044
- Attention on Attention(AoA) for Image Captioning: https://arxiv.org/pdf/1908.06954.pdf
- Bottom-up and Top-Down Attention for Image Captioning: https://arxiv.org/pdf/1707.07998.pdf
- Detectron2: https://github.com/facebookresearch/detectron2
- Self-Critical Sequence Training for Image Captioning: https://arxiv.org/abs/1612.00563
- CIDEr: Consensus-based Image Description Evaluation: https://arxiv.org/abs/1411.5726