95-891: Introduction to Artificial Intelligence

Session 10: Applications of Computer Vision

David Steier

steier@andrew.cmu.edu

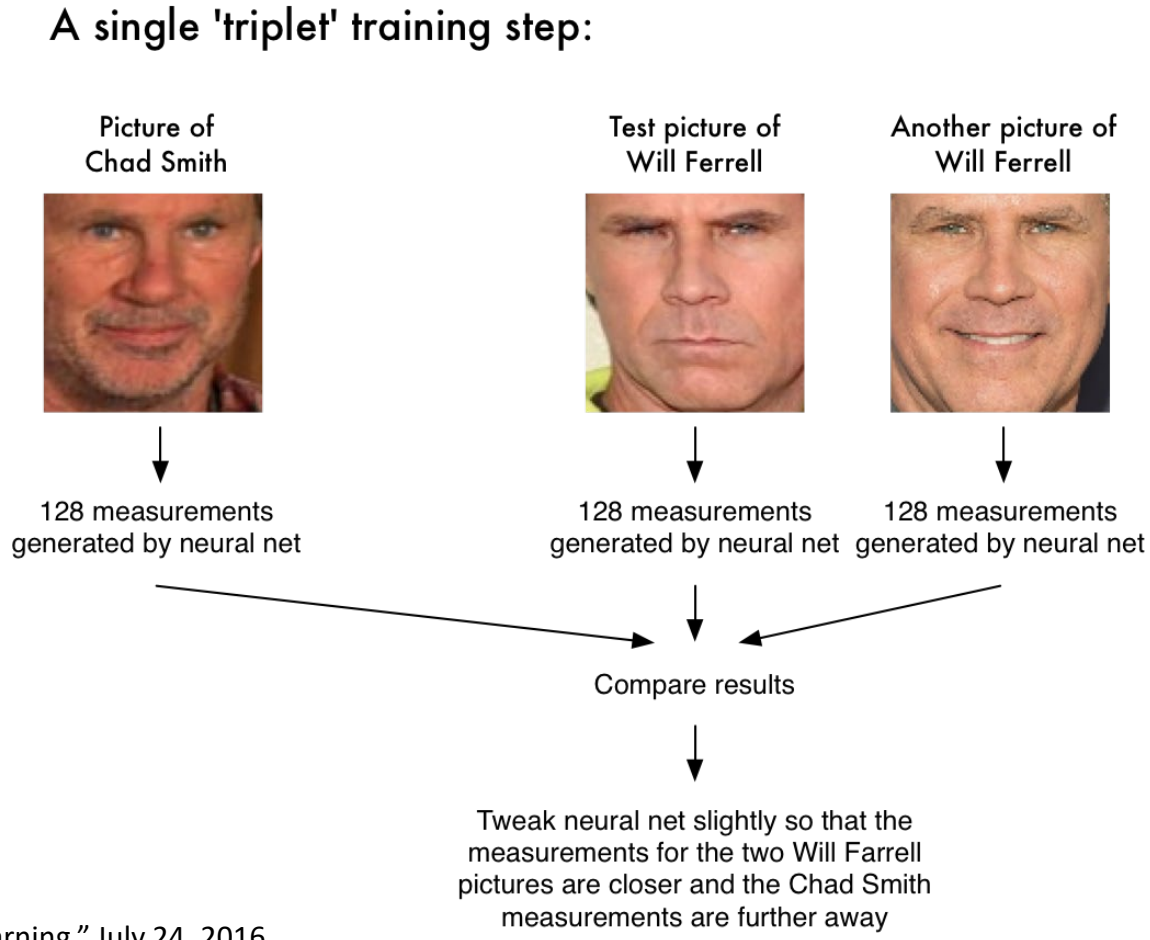
September 25, 2025

Agenda

- Contrastive learning
- Image synthesis using DALLE-2 and other models
- Case study: Computer vision for parasite detection

Remember Triplets in Facial Recognition

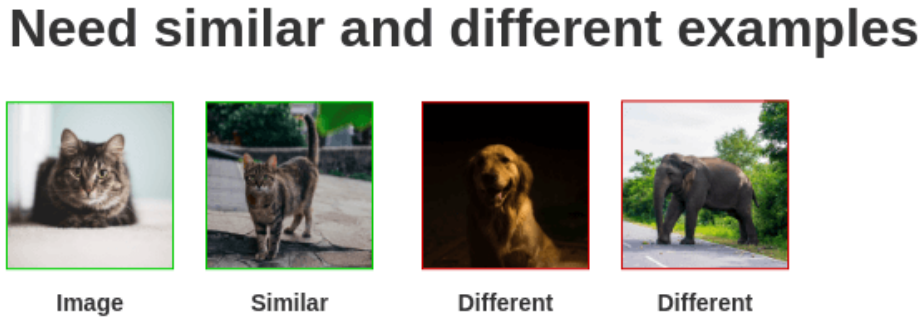
- Use CNN to compute face embeddings as 128-element vector
- Need to do this millions of times for millions of images
- Build on pretrained networks such as OpenFace
  https://github.com/cmusatyalab/openface



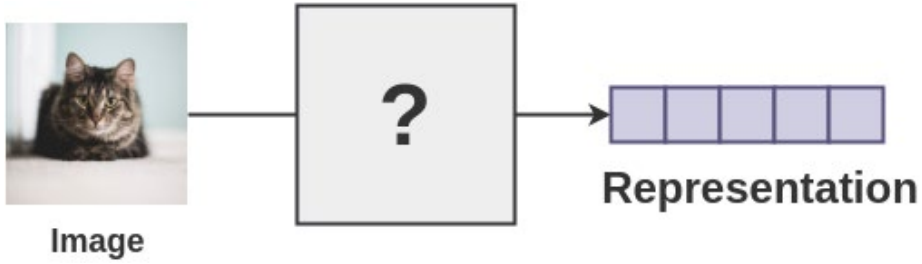
A Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning," July 24, 2016, https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

Self-Supervised Learning (Without Labels)

- Use available images to generate labels based on similarity



- Represent images as vectors



- Quantify similarity of images

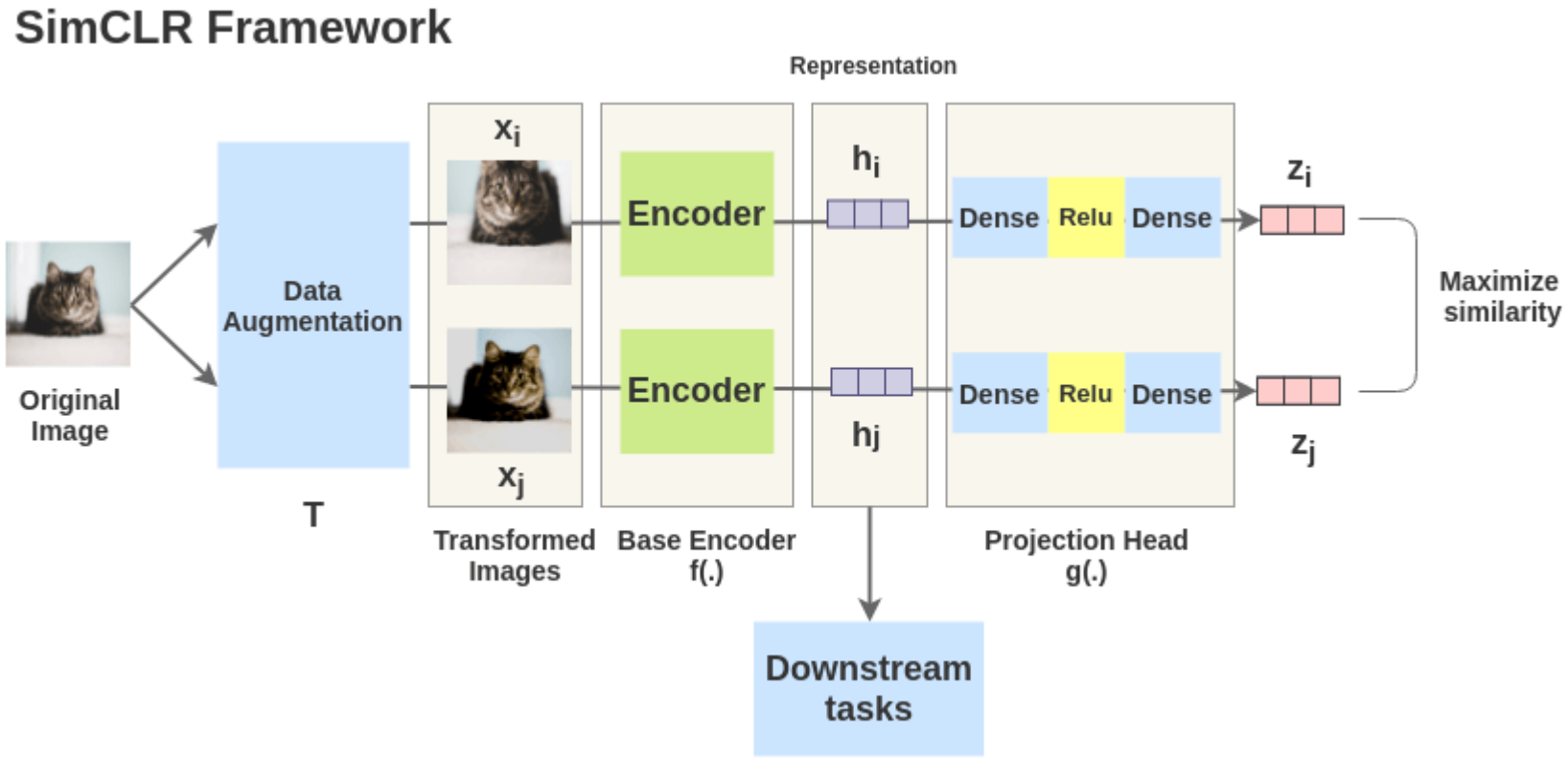
Chaudhary, A, "The Illustrated SimCLR2 Framework," March 2020, https://amitness.com/2020/03/illustrated-simclr/

Contrastive Learning

- Learn the general features of a data set without labels by training a network to recognize which points are similar
- Three steps
    1. Augment the data to create two similar points (for images, cropping, resizing, recoloring etc.)
    2. Create vector representations for each data point in the augmented set
    3. Maximize the similarity of the augmented data points by minimizing a contrastive loss function

E. Tiu, “Understanding Contrastive Learning,” January 7, 2021, https://towardsdatascience.com/understanding-contrastive-learning-d5b19fd96607

SimCLR Framework for Contrastive Learning

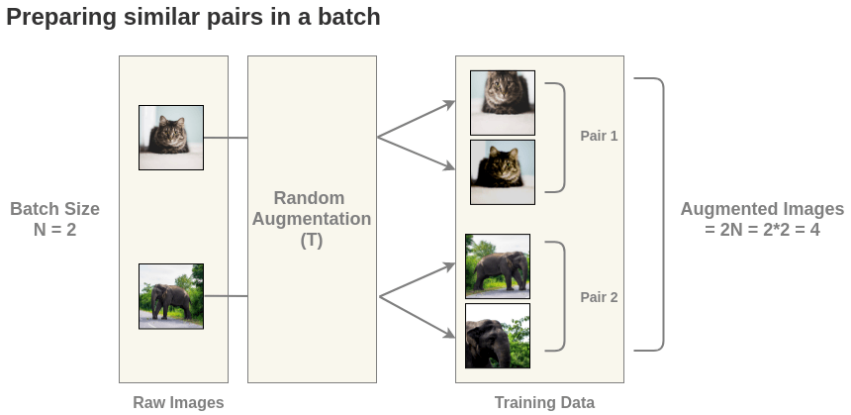


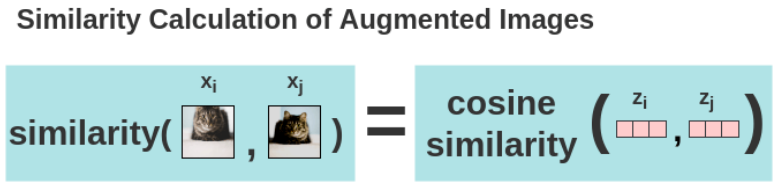
Chaudhary, A, "The Illustrated SimCLR2 Framework," March 2020, https://amitness.com/2020/03/illustrated-simclr/

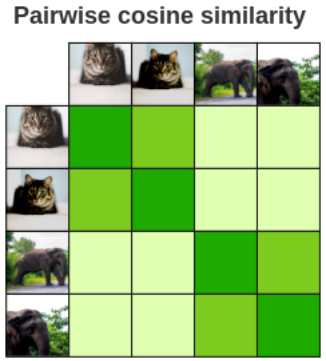
Getting Similar Images

- Augment (color, resize, rotate) data sets so that augmented images are known to be similar

- Use cosine similarity on vector representations

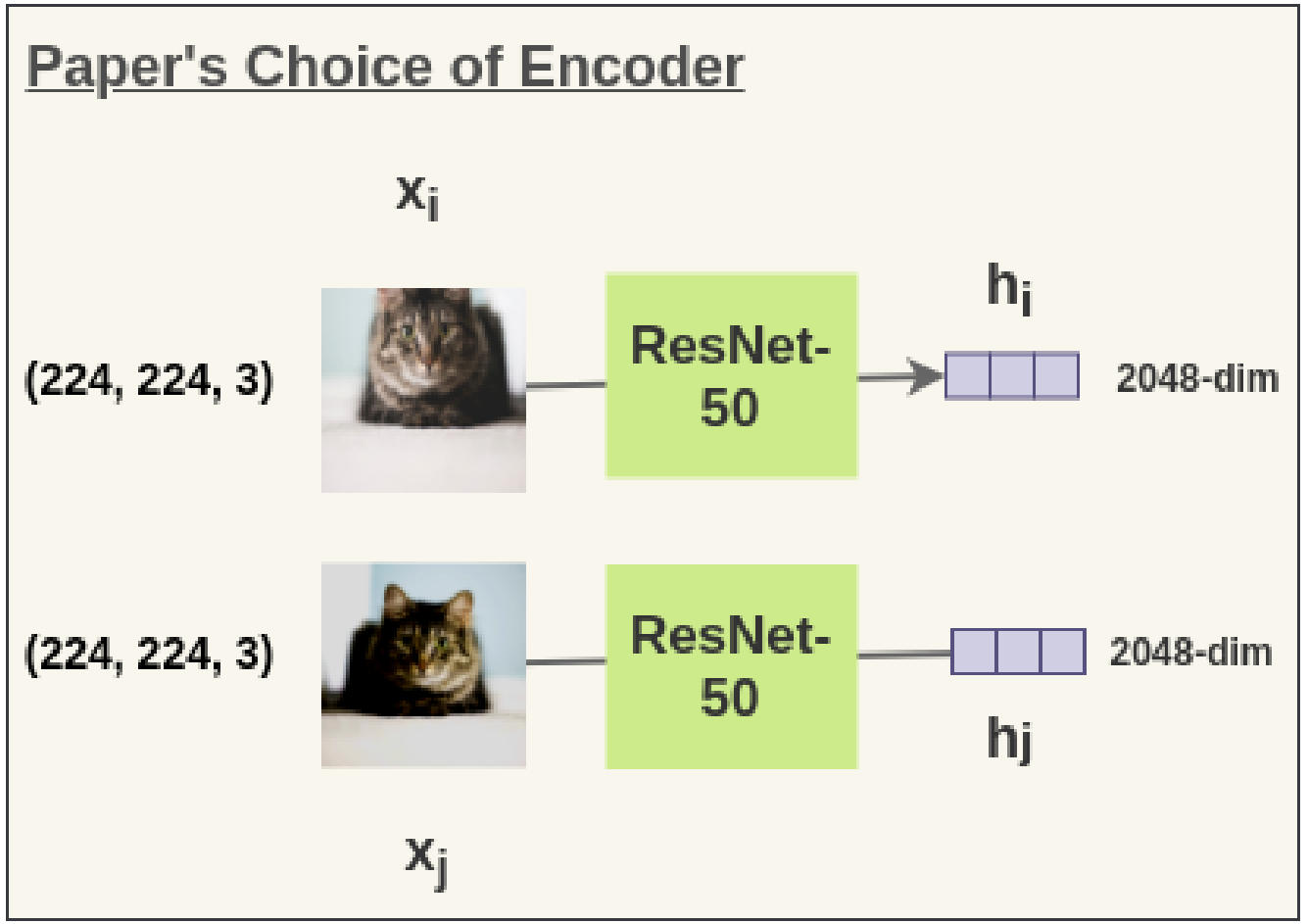






Chaudhary, A, "The Illustrated SimCLR2 Framework," March 2020, https://amitness.com/2020/03/illustrated-simclr/

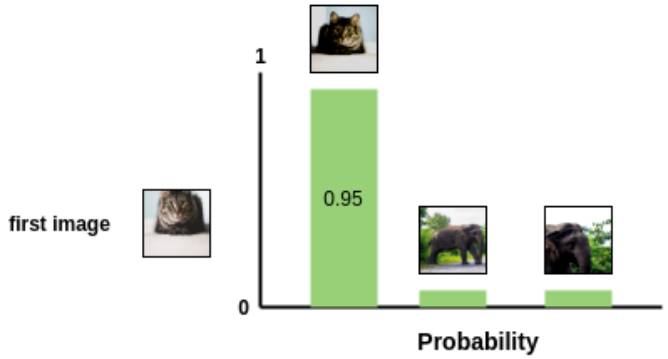
Encoding the Images



Chaudhary, A, "The Illustrated SimCLR2 Framework," March 2020, https://amitness.com/2020/03/illustrated-simclr/

Minimizing the Loss Function

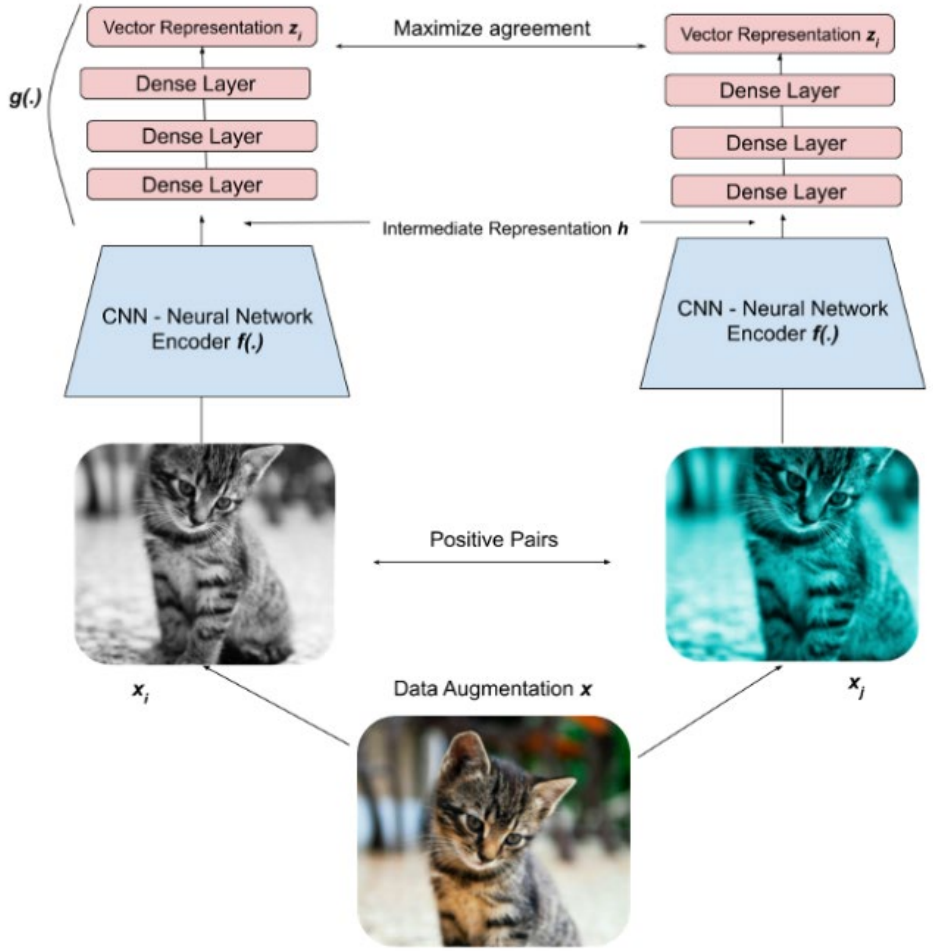
- The goal is to represent the images to maximize the probability of the augmented second image being most similar to the first image
- The loss function is smallest when the augmented image is most similar to the original image and most dissimilar from the remaining images
- Minimize overall loss as average across all pairs



$$l(\text{cat}_1, \text{cat}_2) = -\log\left(\frac{e^{\text{similarity}(\text{cat}_1, \text{cat}_2)}}{e^{\text{similarity}(\text{cat}_1, \text{cat}_2)} + e^{\text{similarity}(\text{cat}_1, \text{elephant})} + e^{\text{similarity}(\text{cat}_1, \text{panther})}}\right)$$

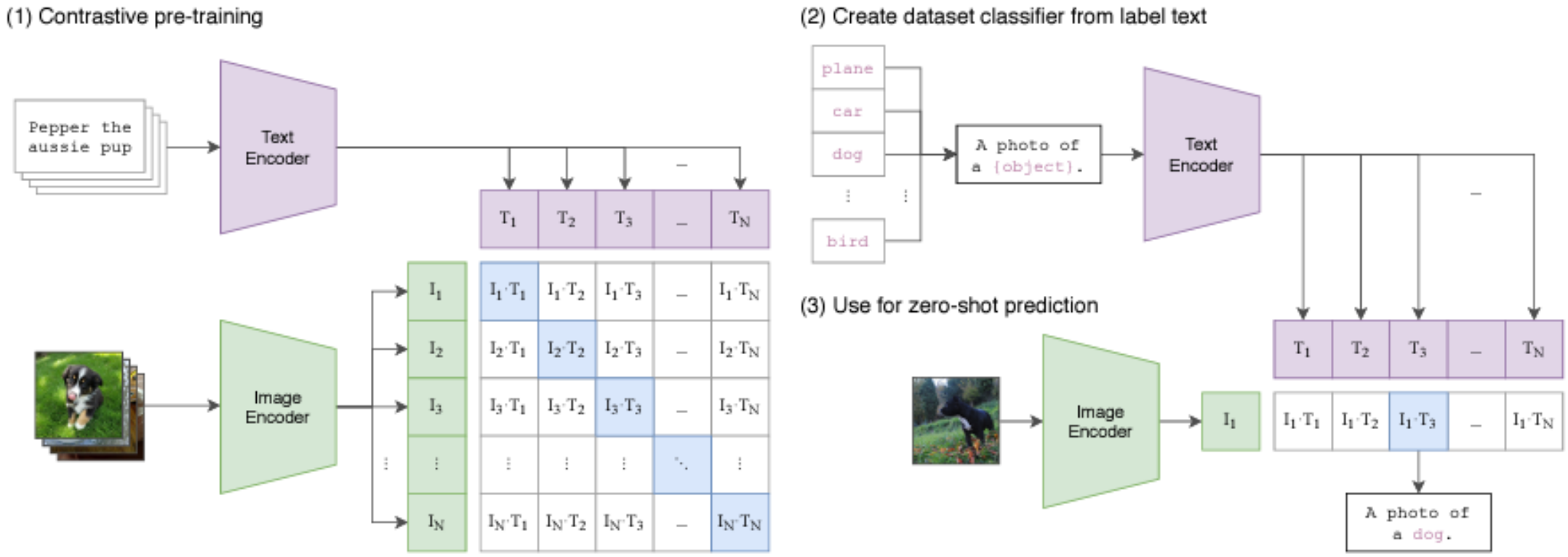
Chaudhary, A, "The Illustrated SimCLR2 Framework," March 2020, https://amitness.com/2020/03/illustrated-simclr/

Google's SimCLRv2 for Contrastive Learning



E. Tiu, "Understanding Contrastive Learning," January 7, 2021, https://towardsdatascience.com/understanding-contrastive-learning-d5b19fd96607
Also see https://amitness.com/2020/03/illustrated-simclr/

Contrastive Learning Image Pre-Training (CLIP)



- Encodes images and natural language descriptions to create image embeddings

A. Radford, "Learning Transferable Visual Models From Natural Language Supervision," 26 Feb 2021, https://arxiv.org/abs/2103.00020

How CLIP Learns Text-Image Pairings

- Standard object recognition jointly trains an image feature extractor and a linear classifier to predict some label
- During pre-training, given a batch of N image-text pairs, CLIP predicts which of N*N (image-text) pairings actually occurred
  - CLIP jointly trains an image encoder and text encoder to maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the $N^2 - N$ incorrect pairings
- At test time, the text encoder creates a zero-shot linear classifier by embedding the natural language description of the target classes
- Pre-trained on 400M image-text pairs, CLIP can perform OCR, activity recognition from videos, geo-localization, and outperforms single-task classifiers on supervised ImageNet recognition based on ResNet-50

A. Radford, "Learning Transferable Visual Models From Natural Language Supervision," 26 Feb 2021, https://arxiv.org/abs/2103.00020

DALL-E 2.0 (https://openai.com/dall-e-2/)

- Combines pre-training on CLIPS' 400M image-text pairs, curated to exclude harmful content, with image synthesis to produce variations
- Prompt: "Vibrant portrait of Salvador Dali with a robotic half-face"
- DALLE-2.0 released in summer 2022 to a million users, but source code and models are not released

A. Ramesh, "Hierarchical Text-Conditional Image Generation with CLIP Latents," 13 April 2022, https://arxiv.org/abs/2204.06125

Figure 18: Random samples from unCLIP for prompt "Vibrant portrait painting of Salvador Dali with a robotic half face"

DALL-E 2.0 Combines CLIP and unCLIP

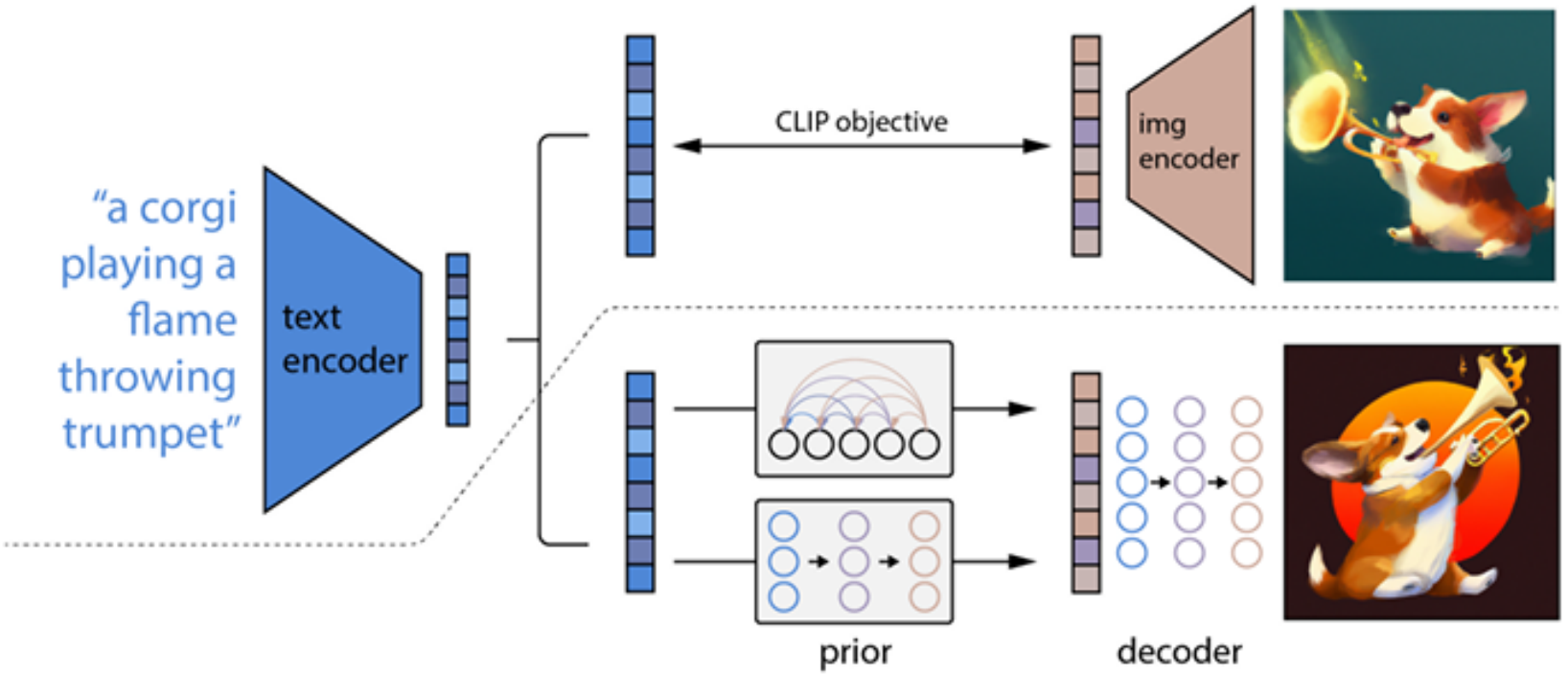
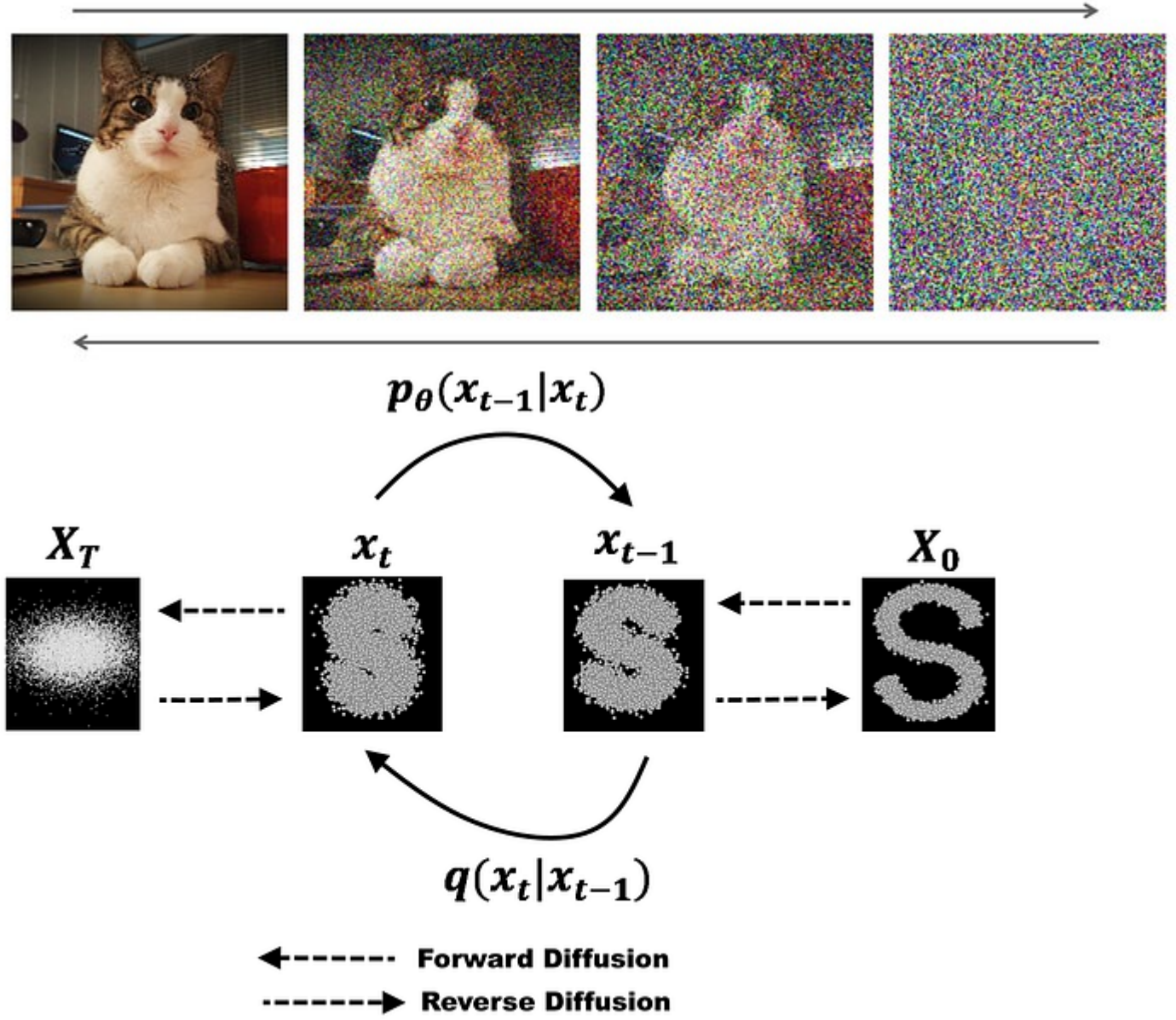


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

A. Ramesh, "Hierarchical Text-Conditional Image Generation with CLIP Latents," 13 April 2022, https://arxiv.org/abs/2204.06125

Diffusion Decoders

- Imagine adding noise to a coherent image at each time-step
- Diffusion works by reversing the process, to **denoise** an image



S. Wang, "Adding Noise Until The Model Can Generate My Head Exploding (.png): A Brief Overview of Diffusion Models and DALL-E 2", Sep 24, 2022 , https://medium.com/demistify/adding-noise-until-the-model-can-generate-my-head-exploding-png-7e31f672dd3c#:~:text=To%20explain%20diffusion%20models%20as,since%20it%20satisfies%20this%20requirement.

DALLE 2.0 Captures Semantics and Style

- Two components
  - A **prior** $P(z_i|y)$ that produces CLIP image embeddings z_i conditioned on captions y
  - A **decoder** $P(x|z_i,y)$ that produces images x conditioned on CLIP image embeddings z_i (and optionally text captions y).
- Stacking the two components yields the generative model

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y)$$

A. Ramesh, "Hierarchical Text-Conditional Image Generation with CLIP Latents," 13 April 2022, https://arxiv.org/abs/2204.06125

DALLE 3.0 Adds More Detail

https://openai.com/dall-e-3

Re-captioning to train DALL-E 3.0

	Image		
Alt Text	now at victorian plumbing.co.uk	is he finished...just about!	23 (19 of 30) 1200
SSC	a white modern bathtub sits on a wooden floor.	a quilt with an iron on it.	a jar of rhubarb liqueur sitting on a pebble background.
DSC	this luxurious bathroom features a modern freestanding bathtub in a crisp white finish. the tub sits against a wooden accent wall with glass-like panels, creating a serene and relaxing ambiance. three pendant light fixtures hang above the tub, adding a touch of sophistication. a large window with a wooden panel provides natural light, while a potted plant adds a touch of greenery. the freestanding bathtub stands out as a statement piece in this contemporary bathroom.	a quilt is laid out on a ironing board with an iron resting on top. the quilt has a patchwork design with pastel-colored strips of fabric and floral patterns. the iron is turned on and the tip is resting on top of one of the strips. the quilt appears to be in the process of being pressed, as the steam from the iron is visible on the surface. the quilt has a vintage feel and the colors are yellow, blue, and white, giving it an antique look.	rhubarb pieces in a glass jar, waiting to be pickled. the colors of the rhubarb range from bright red to pale green, creating a beautiful contrast. the jar is sitting on a gravel background, giving a rustic feel to the image.

Betker, K. et al, "Improving Image Generation with Better Captions", https://cdn.openai.com/papers/dall-e-3.pdf

MidJourney (https://www.midjourney.com/home/)

- Prompt: “horse galloping at sunset <artist> style”
- Can also specify camera/lens, effects, rendering, resolution, themes

L. Nielsen, Sep. 3, 2022, https://medium.com/mlearning-ai/an-advanced-guide-to-writing-prompts-for-midjourney-text-to-image-aa12a1e33b6

StableDiffusion (https://stablediffusionweb.com/)

- Open source!
- Built-in “AI Safety Classifier”
- Trained on 5 billion scraped images from the internet LAION-5B

- Rombach, et.al. April 13, 2022, “High-resolution Image Synthesis with Latent Diffusion Models”, https://arxiv.org/abs/2112.10752

Potential Ethical Issues in Image Synthesis

- Generating toxic/biased/adult content
- Generating deepfake images/video
- Use of artistic intellectual property without consent or compensation
- Charging for image synthesis built on the work of others' scraped images

S. Rosenberg, Sep 12, 2022, "AI-generated images open multiple cans of worms", https://www.axios.com/2022/09/12/ai-images-ethics-dall-e-2-stable-diffusion

DALL-E 2 Preview – Risks and Limitations, April 2022, https://github.com/openai/dalle-2-preview/blob/main/system-card.md

One Artist's Reaction to DALL-E 2.0

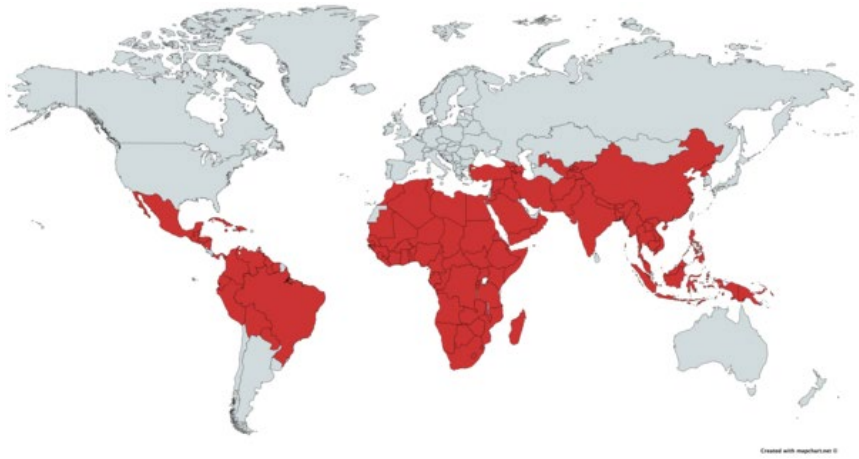
"I don't say this lightly: this new AI system is not just on-par with human artists. It is definitively better than humans in almost every sense of the word…. This should sound the alarm for the vast majority of current artists to start looking for new work. If you do commissions, business designs, illustrations — hell, even animations — your value in the marketplace is about to plummet…. The result? There will simply be no place for human artists, aside from maybe 'artisan' markets that are treated the same way we treat antiques today."

Nick Saraev, Apr 9, 2022

https://medium.com/geekculture/yesterday-marked-the-death-of-art-as-an-industry-1472b9c49c63

Computer Vision to Diagnose Parasitic Diseases

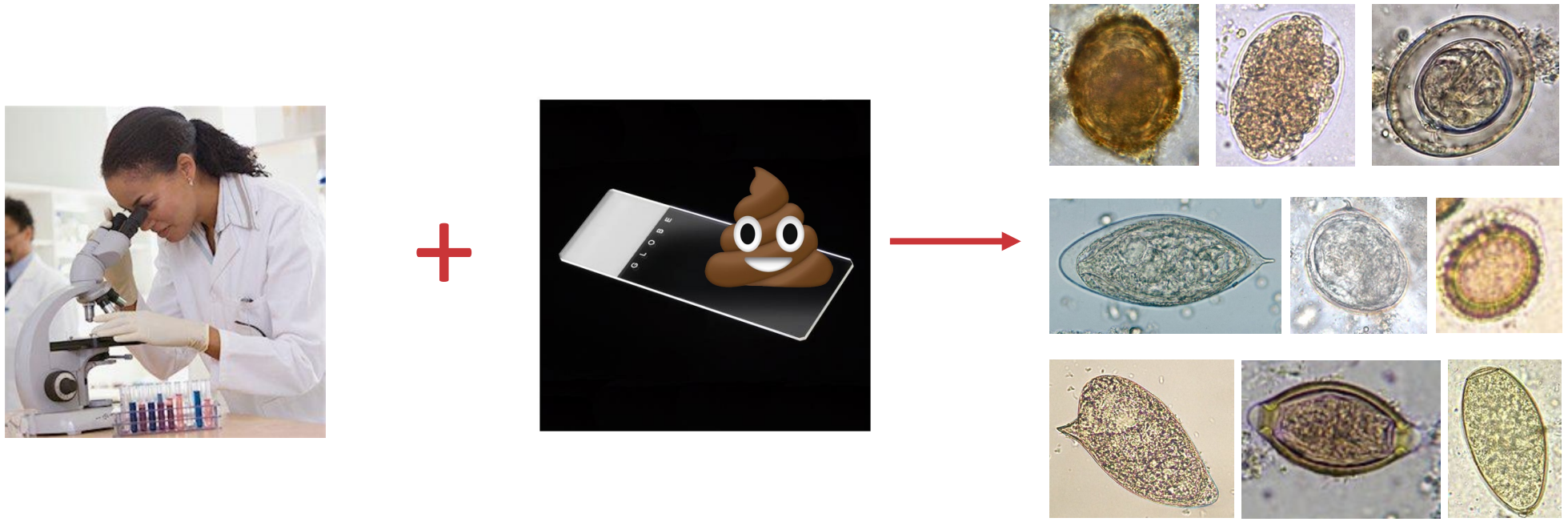
- Affects 2 billion people (1 in 4) worldwide
- Causes disabilities and kills more than 500,000 people annually
- Treatment is easy (5 cents per application, few side effects), but diagnosis is hard because symptoms could be caused by multiple organisms



https://parasite.id/

Current Diagnosis Process is Expensive and Hard to Scale

Generally diagnosed through microscopy but good microscopes are expensive and diagnosticians are scarce in the developing world



https://parasite.id/docs/ParasiteID-Presentation--UC-Berkeley-MIDS-Capstone-Showcase-12-17-2018.pdf

Parasite.ID Addresses This Problem

- First web-based diagnostic aid for parasitic infection
- Built with the most comprehensive set of parasitic worm egg images ever assembled
- Built by 4 Berkeley MIDS (Masters in Information and Data Science) students in a semester using CNNs, transfer learning, cloud computing
  - Kiersten Henderson
  - Nat Schub
  - Cameron Bell
  - Vicki Foss

https://parasite.id/

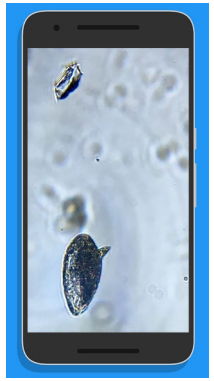
https://medium.com/berkeleyischool/capstone-project-winner-parasiteid-aims-to-facilitate-diagnoses-in-developing-world-1f1d860f51b0

Parasite.ID Approach

Entire village

Almost Anyone

+



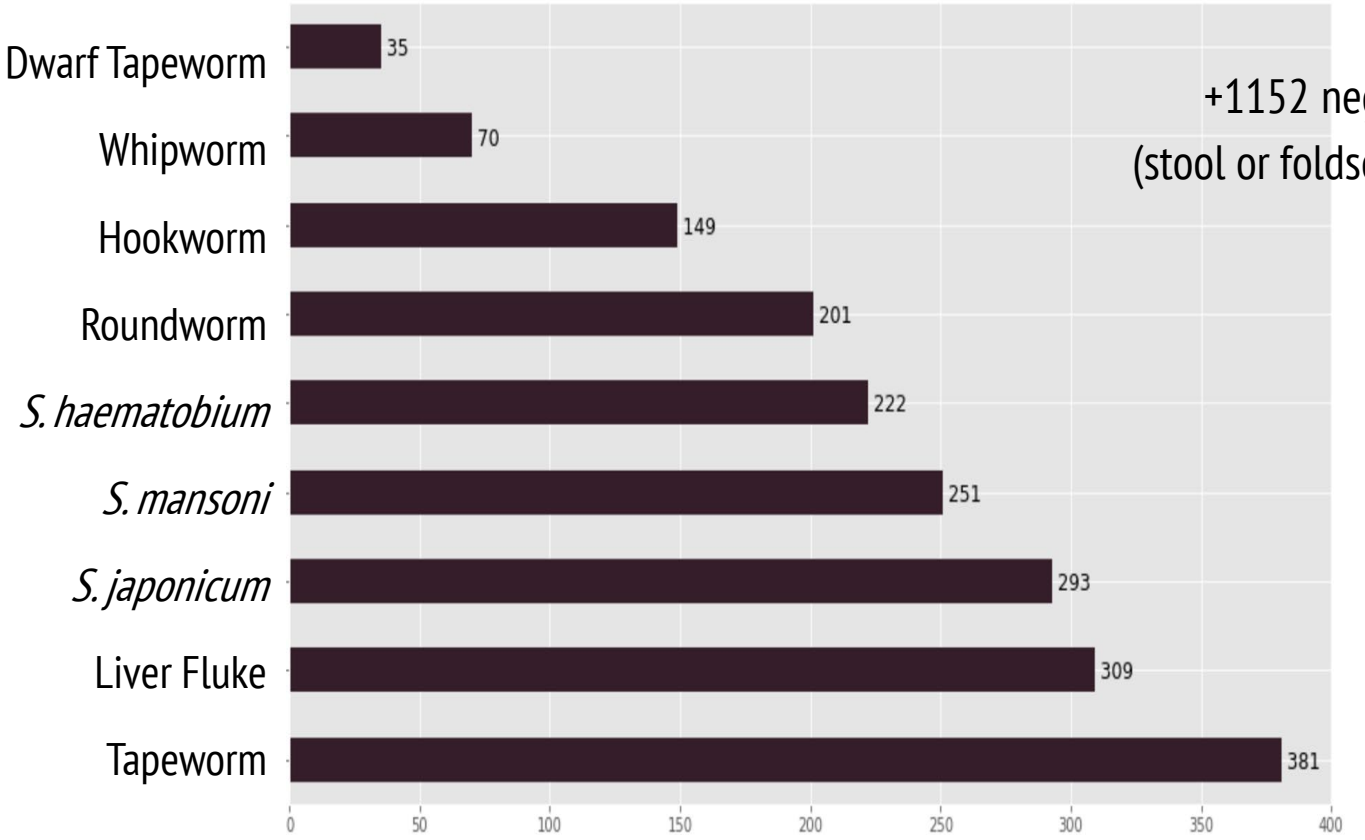
Foldscope and cell phone

ParasiteID

Schistosoma mansoni **identified!**

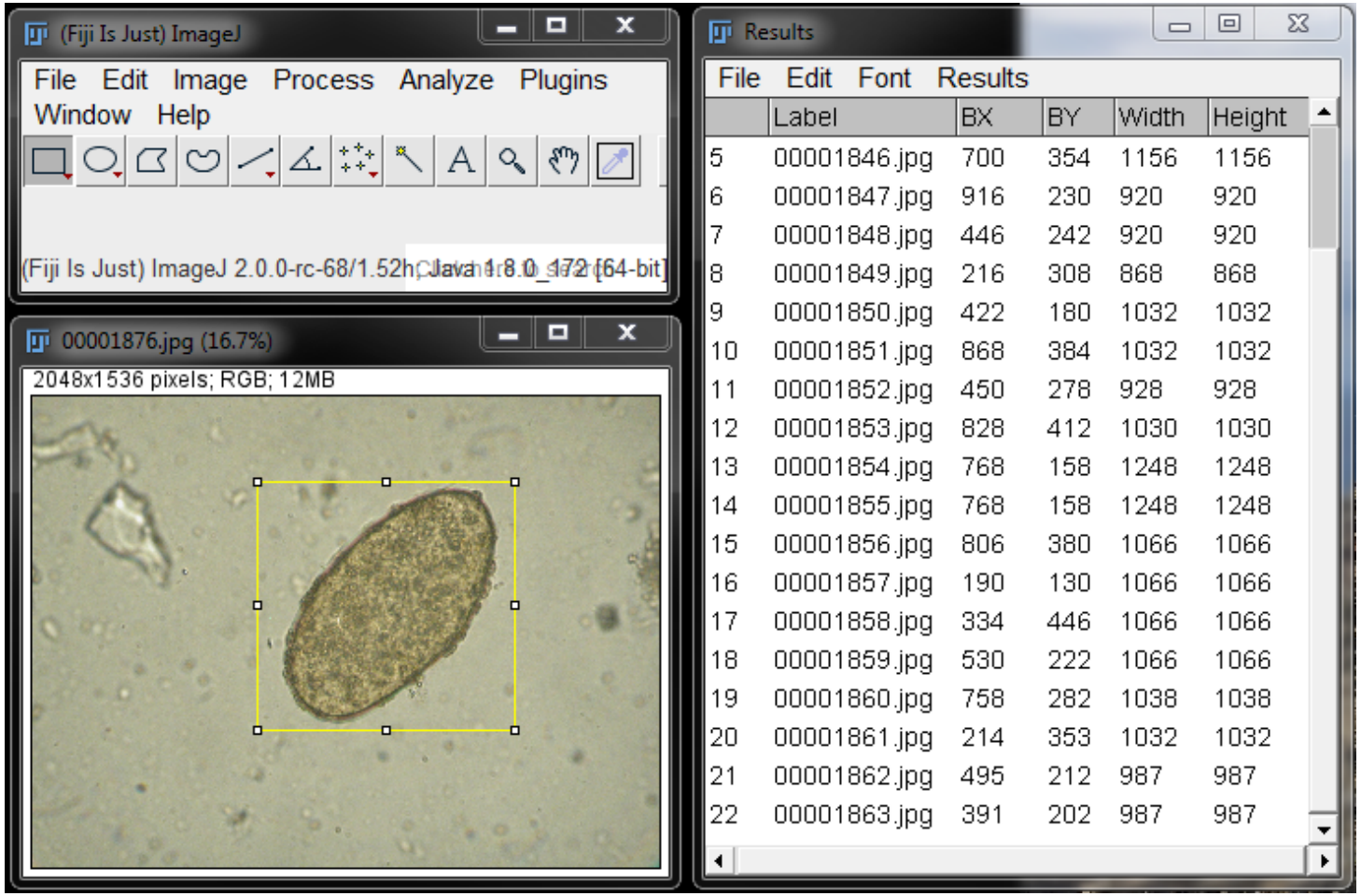
Diagnosis:
Schistosomiasis

2000+ Image Data Set for Parasite.ID



+1152 negative images
(stool or foldscope background)

Manually Drawing Bounding Boxes



	Label	BX	BY	Width	Height
5	00001846.jpg	700	354	1156	1156
6	00001847.jpg	916	230	920	920
7	00001848.jpg	446	242	920	920
8	00001849.jpg	216	308	868	868
9	00001850.jpg	422	180	1032	1032
10	00001851.jpg	868	384	1032	1032
11	00001852.jpg	450	278	928	928
12	00001853.jpg	828	412	1030	1030
13	00001854.jpg	768	158	1248	1248
14	00001855.jpg	768	158	1248	1248
15	00001856.jpg	806	380	1066	1066
16	00001857.jpg	190	130	1066	1066
17	00001858.jpg	334	446	1066	1066
18	00001859.jpg	530	222	1066	1066
19	00001860.jpg	758	282	1038	1038
20	00001861.jpg	214	353	1032	1032
21	00001862.jpg	495	212	987	987
22	00001863.jpg	391	202	987	987

Choices to make in Designing CNNs

- Kernel size of the filter
- Number of filters
- Types of intermediate hidden layers
- Size of the intermediate hidden layers
- Whether and where to add dropout or max pooling layers

Baseline Model- Simple CNN

Baseline CNN Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 205, 205, 32)      12832
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 41, 41, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 41, 41, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 53792)             0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1721376
_________________________________________________________________
dense_2 (Dense)              (None, 10)                330
=================================================================
Total params: 1,734,538
Trainable params: 1,734,538
Non-trainable params: 0
_________________________________________________________________
```

- Test accuracy = ~74%
- Poor intuition around what param choice
- Kernel size, # filters, # of layers, types of layers, size of layers, dropout, max pooling, etc.
- Took a long time to train (>4 hrs), difficult to randomly test params

Transfer Learning Based on VGG16

Pre-trained VGG16 Model

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	(None, None, None, 3)	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0

Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688

Baseline Model

Layer (type)	Output Shape	Param #
input_shape (InputLayer)	(None, 224, 224, 3)	0
vgg16 (Model)	multiple	14714688
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 10)	40970

Total params: 134,301,514
Trainable params: 119,586,826
Non-trainable params: 14,714,688

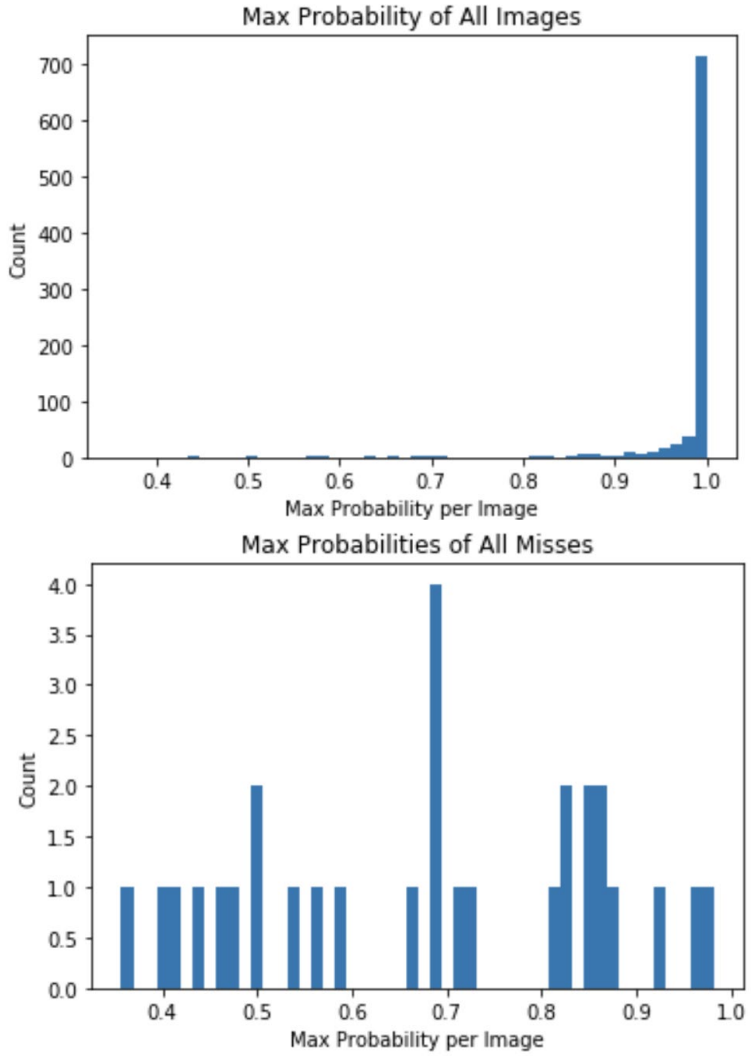
Results: Evaluation metrics

Class	Accuracy	Error rate	Specificity	Precision	Recall	F1-score
Hymenolepsis_nana	0.999	0.001	0.999	0.833	1.000	0.909
Trichuris_trichiura	0.995	0.005	0.998	0.818	0.818	0.818
Hookworm	0.993	0.007	0.995	0.862	0.926	0.893
Ascaris_lumbricoides	0.994	0.006	0.995	0.922	0.979	0.949
Schistosoma_haematobium	0.983	0.017	0.993	0.846	0.786	0.815
Schistosoma_mansoni	0.991	0.009	0.998	0.963	0.897	0.929
Schistosoma_japonicum	0.992	0.008	0.996	0.946	0.930	0.938
Fasciola_hepatica	0.998	0.002	0.998	0.967	1.000	0.983
Taenia	0.997	0.003	1.000	1.000	0.959	0.979
Negative	0.992	0.008	0.987	0.990	0.996	0.993
Total (macro-average metric)	**0.993**	**0.007**	**0.996**	**0.915**	**0.929**	**0.921**
Total (micro-average metric)			**0.996**	**0.967**	**0.967**	**0.967**

The Hammock Problem

- Data quality often matters more than model hyperparameter tweaks
- Train how you test, ensure training data looks like expected test data
- Consider model probability thresholds

Cutoff Thresholds



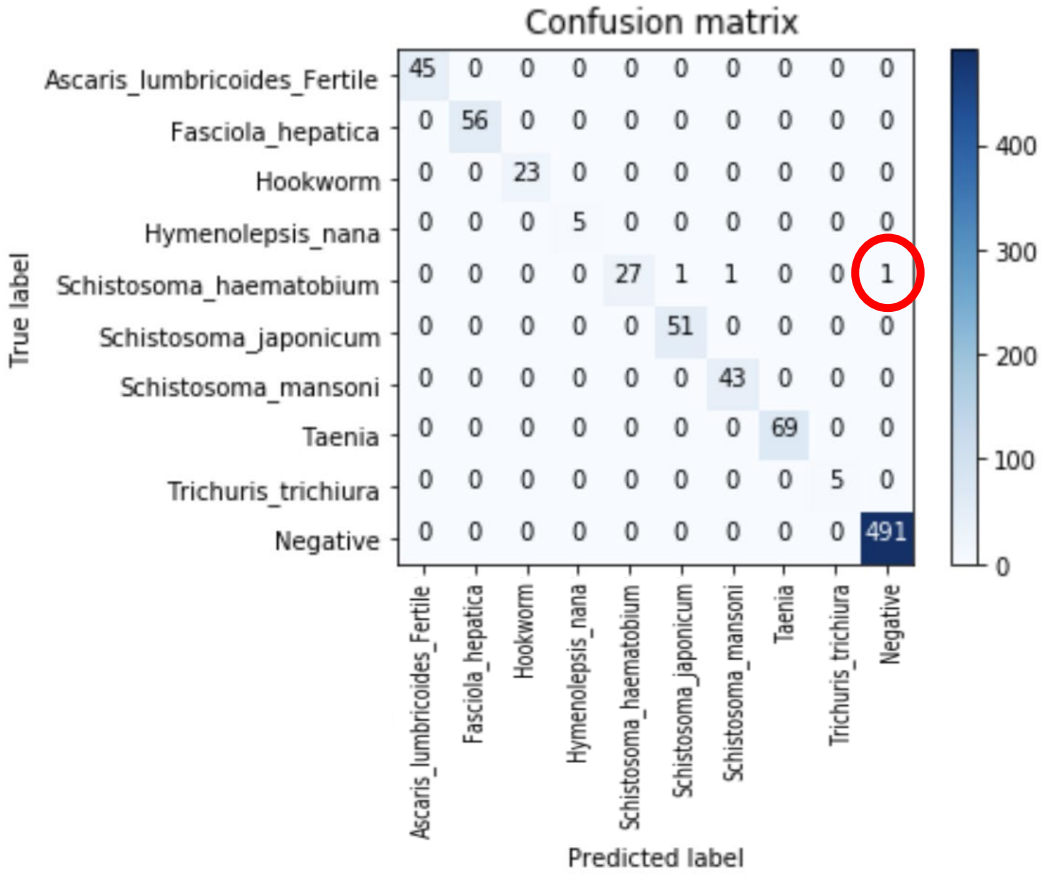
Original approach:

- Model results = [0.1, 0.05, 0.4, …. 0.1]
- We returned class with highest probability, regardless of model confidence
- Most misses/low probability images were quite difficult to classify, even by humans

New approach:

- Only return results if model is >90% accurate
- Better to ask user to upload better quality image

Results on All Images with >90% Cutoff



- User-facing accuracy: 99.6% (816 correct/818 total images) on new set of 818 held out images, when excluding low-confidence images
- <90% confidence on 62/880 original images, model returns “Please upload better image”
- Only 1 false negative, no false positives

Differing Error Costs

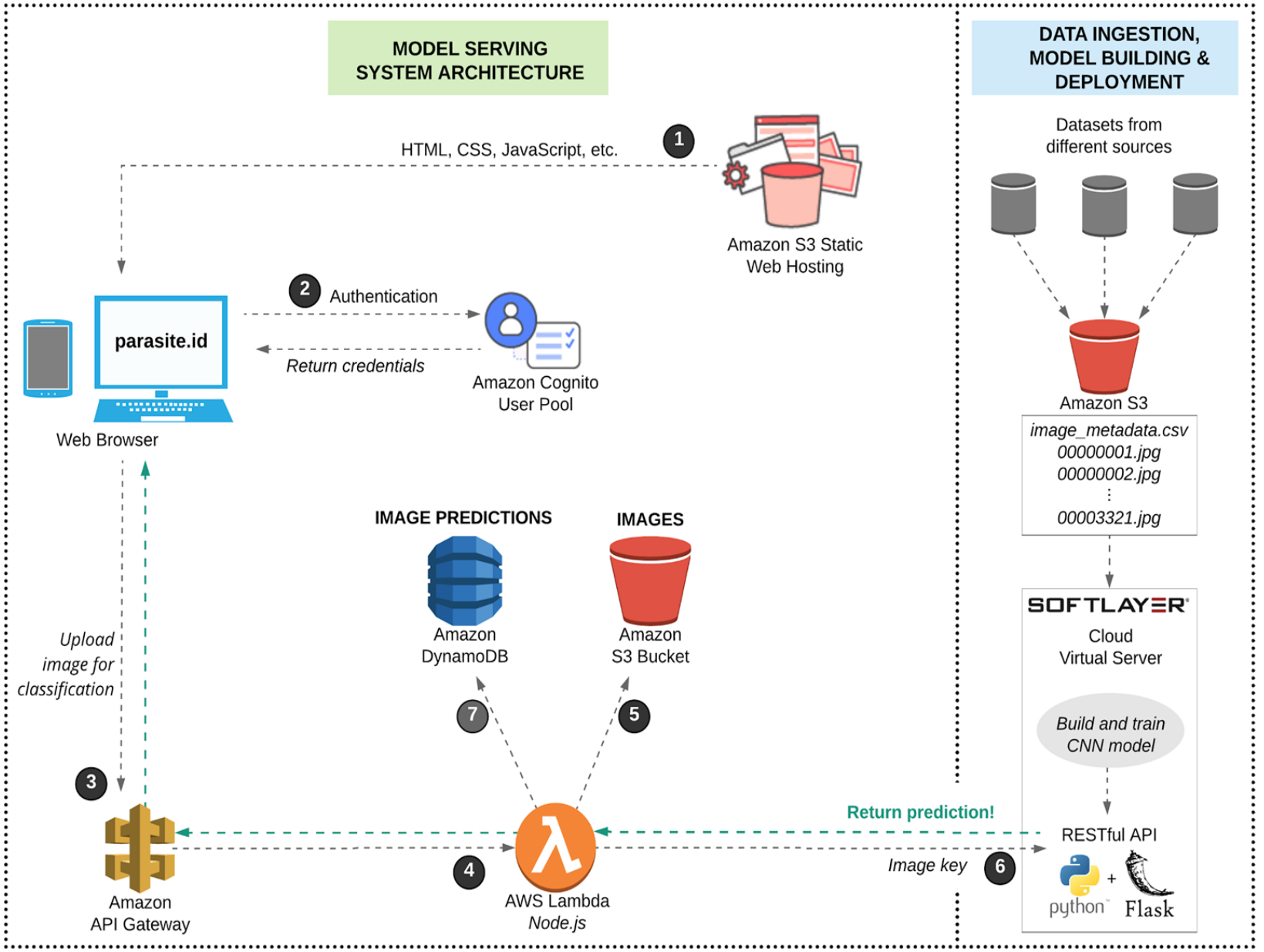
- **False positives:** Few side effects (<<1%)
- **False negatives:** Patient goes untreated (<1%)
- **Misdiagnosis:** Patient receives the wrong drug

Humans: 74-95%

ParasiteID: 96.4%

https://www.ncbi.nlm.nih.gov/pubmed/24992655

Parasite.ID System Architecture



Conclusion

- Contrastive learning is a self-supervised, task-independent deep learning technique that allows a model to learn about data, even without labels.
- The model learns general features about the dataset by learning which types of images are similar, and which ones are different.
- The practicality of computer vision applications will depend on consideration of the deployment environment, including image quality and skills needed to use the app.
- **Next class (September 30):** Natural language understanding