Improving few shot object classification using contrastive learning

Abhishek Rajendra Prasad
Jishnu Jaykumar Padalunkal

CS 6301.004 - Deep Learning For NLP

Group-11 | Spring 2023

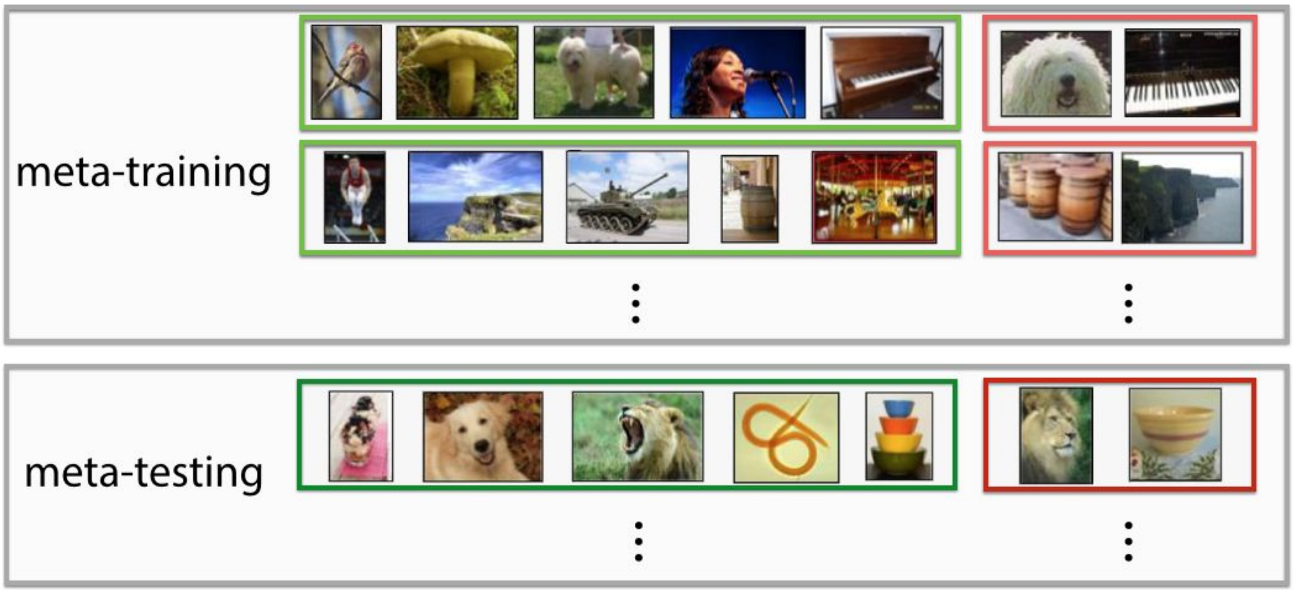
Improving **few shot** object classification using contrastive learning

- Few-Shot Learning is a sub-area of machine learning. It's about classifying **new data** when you have only a **few training samples** with **supervised information** (neptune.ai).
- Formulated as an N-way-K-shot problem (**Episodes**)
  - N := number of classes
  - K := number of samples per class
    - In a fixed setup, this remains same for all classes
    - In a variable setup, this varies across classes

Improving **few shot** object classification using contrastive learning

Support Set

Query Set

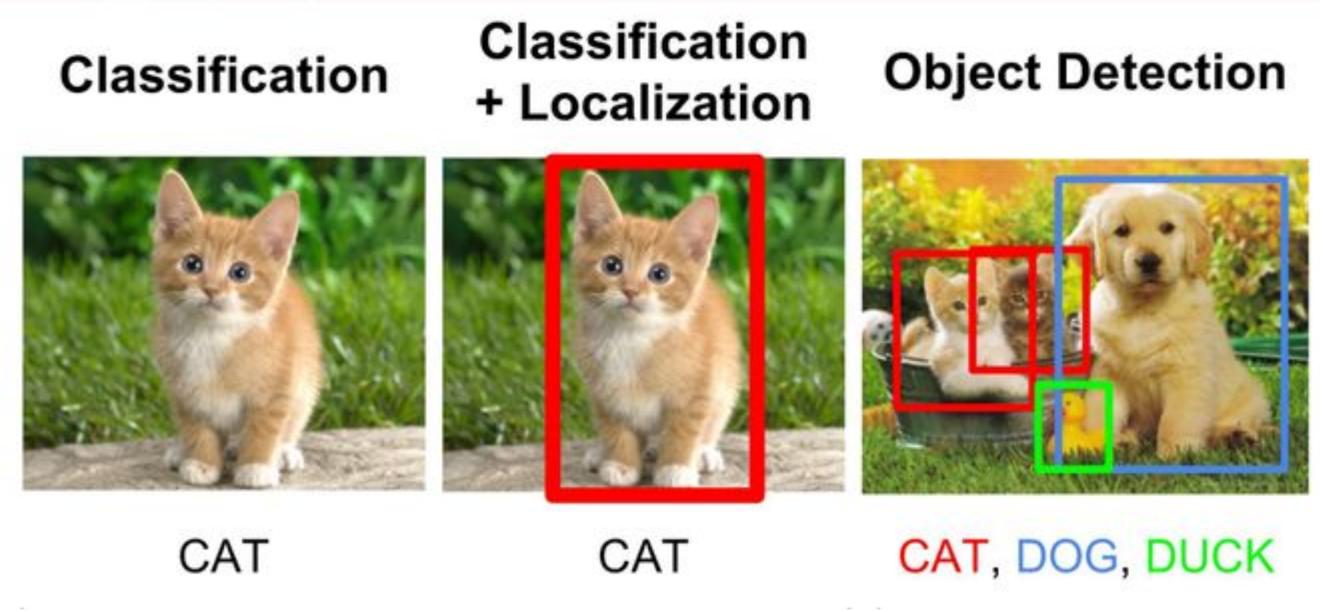


Each row is an
Episode
(Support + Query)

Here, it's a **5-way-1-shot** setup (**fixed** episode variant)

Image: https://bair.berkeley.edu/blog/2017/07/18/learning-to-learn

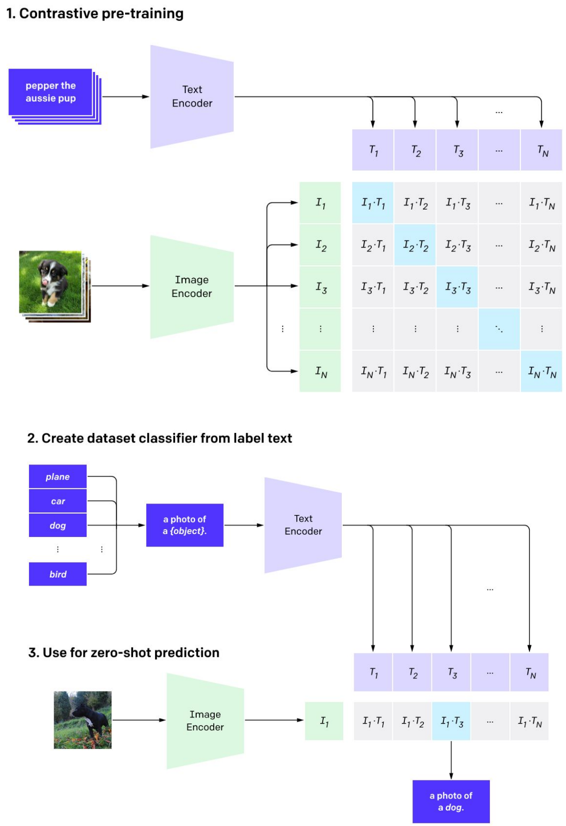
Improving few shot **object classification** using contrastive learning



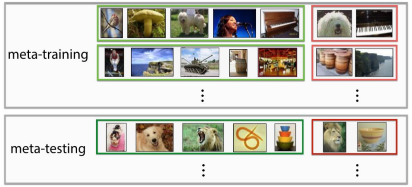
We will be dealing with **classification** only. i.e. Given an image containing a single object, classify it.

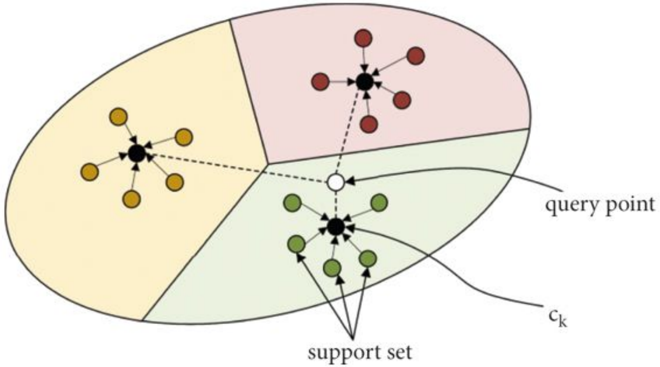
Image: https://www.kaggle.com/getting-started/169984

Related Works



Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *ICML* 2021.





$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x)$$

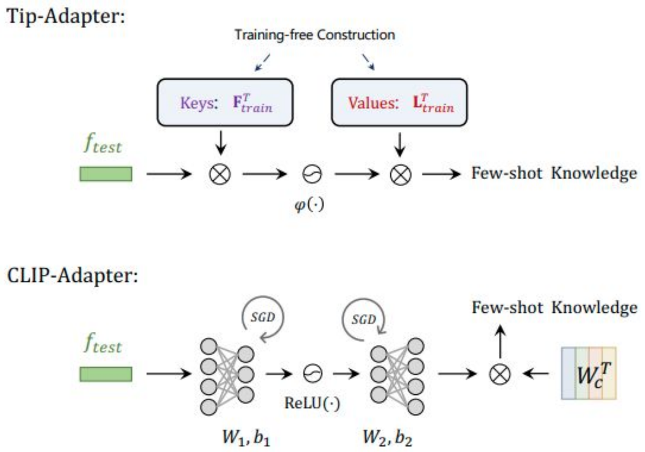
$$p_\phi(y = k|x) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{k'} \exp(-d(f_\phi(x), c_{k'}))}$$

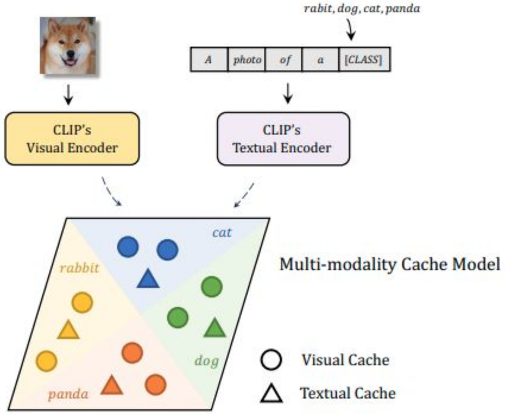
$$\min J(\phi) = -\log p_\phi(y = k|x)$$

Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." NeurIPS 2017.

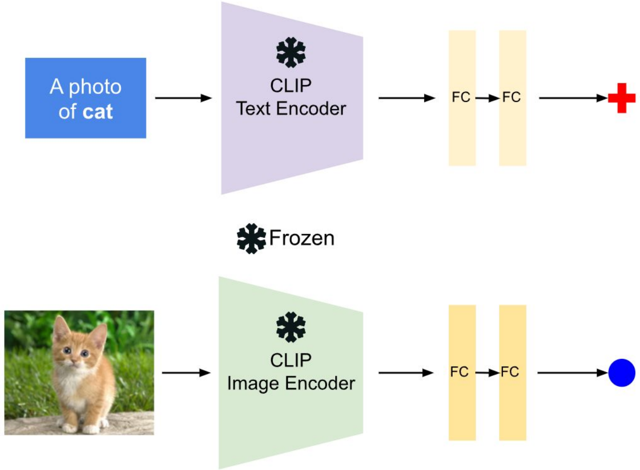
Related Vs Ours

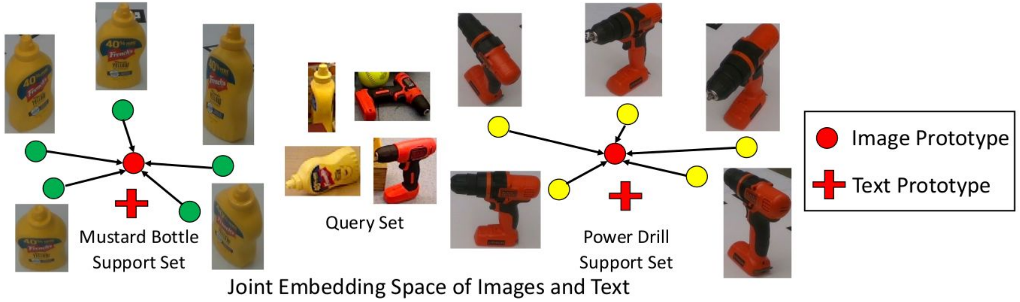
Method	Use Support Sets	Adapt Image Encoder	Adapt Text Encoder	Align Image and Text
Zero-shot CLIP [18]	✗	✗	✗	✓
Linear-probe CLIP [18]	✓	✗	✗	✗
CoOp [26]	✓	✗	✓	✗
CLIP-Adapter [8]	✓	✓	✗	✗
Tip-Adapter [25]	✓	✓	✗	✗
Proposed Method	✓	✓	✓	✓





Proposed Method
Proto-CLIP (Model#1)





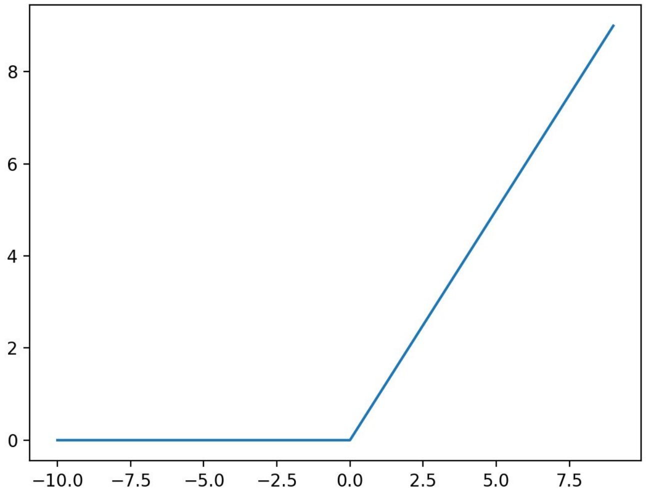
$$P(y=k|\mathbf{x}^q,\mathcal{S}) = \alpha P(y=k|\mathbf{x}^q,\mathcal{S}_x) + (1-\alpha)P(y=k|\mathbf{x}^q,\mathcal{S}_y)$$

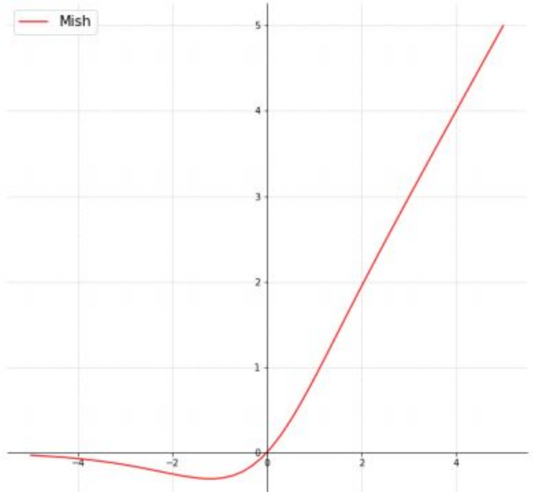
Loss: Negative Log Likelihood

Our proposed **Proto-CLIP** model learns a *joint embedding space of images and text,* where ***image prototypes*** and ***text prototypes*** are learned using ***support sets*** for few-shot classification.

Metric: Accuracy

Activation Fn.: ReLU vs Mish





Source: https://krutikabapat.github.io/Swish-Vs-Mish-Latest-Activation-Functions/

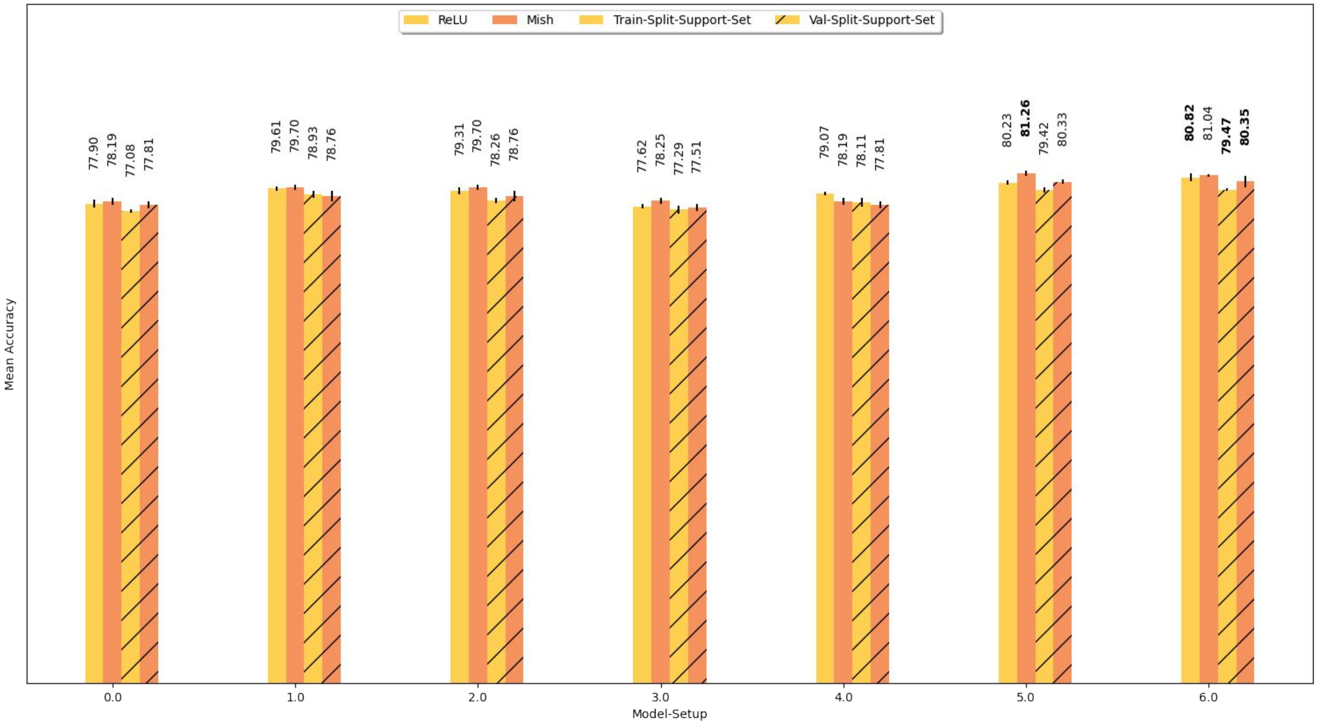
Model#1 Variants

Model-Setup	Res	Tau	CosineAnneal	Output_Embedding_Size		{Input,hidden,output} sizes	
						ViT-B/32	ViT-L/14
0	0	0	0	Config_Output_Size		512, 256, 128	768, 512, 256
1	1	0	0	Config_Input_Size		512, 512, 512	768, 768, 768
2	1	1	0	Config_Input_Size		512, 512, 512	768, 768, 768
3	0	1	0	Config_Input_Size		512, 512, 512	768, 768, 768
4	0	1	0	Config_Output_Size		512, 256, 128	768, 512, 256
5	0	1	1	Config_Output_Size		512, 256, 128	768, 512, 256
6	1	1	1	Config_Input_Size		512, 512, 512	768, 768, 768

CosineAnneal	
0	Adam Opt
1	AdamW Opt + CosineAnnealing (eps=1e-4 same as Tip-A)

Tau	
0	1
1	sqrt(output_embedding_size)

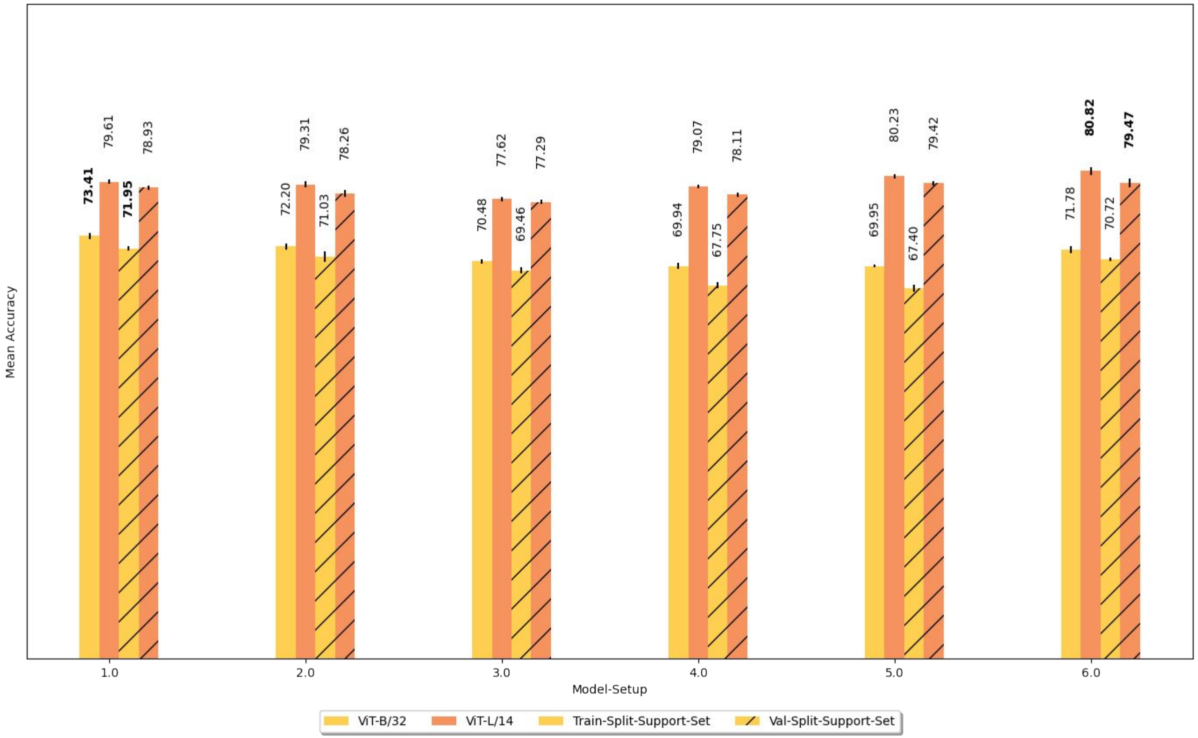
Backbone: ViT-L/14 | ReLU vs Mish



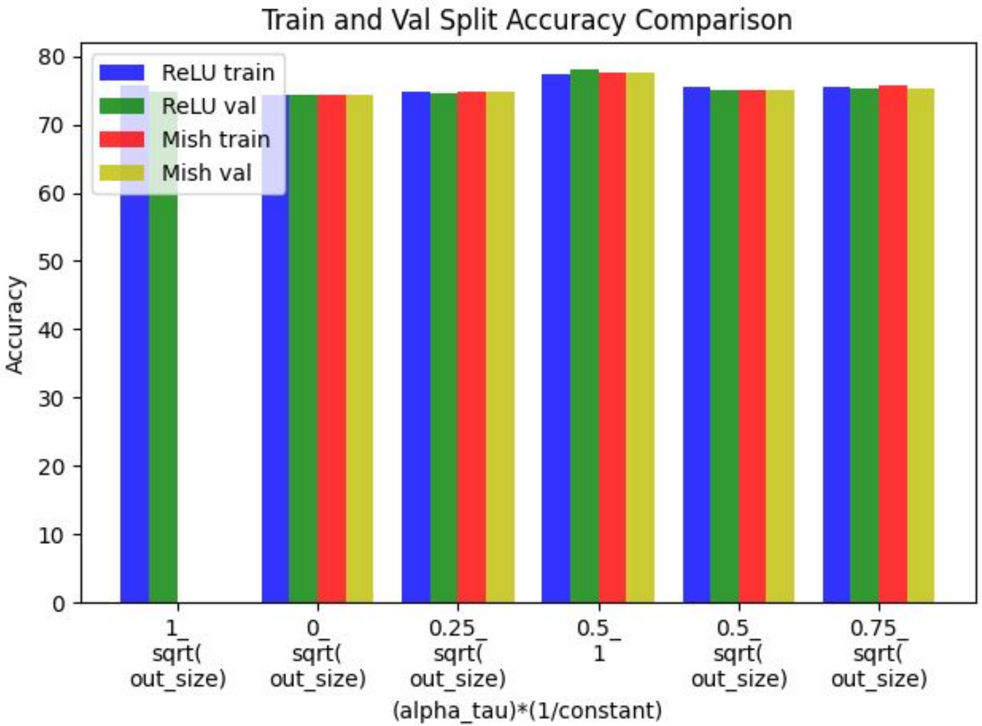
Sample t-SNE plot for Model#1

(Setup#1)

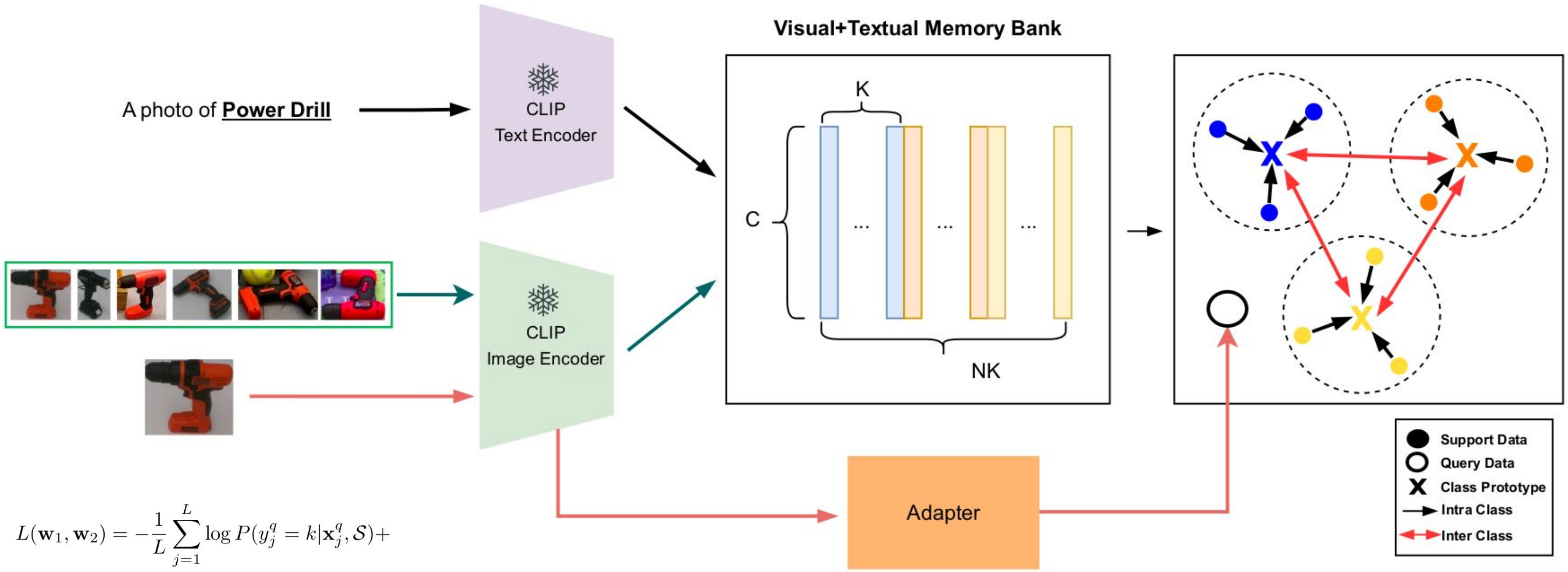
ReLU: ViT-B/32 vs ViT-L/14



DTD Results (Model-Setup#6)



Proposed Method:Proto-CLIP (V2)



$$L(\mathbf{w}_1, \mathbf{w}_2) = -\frac{1}{L}\sum_{j=1}^{L} \log P(y_j^q = k | \mathbf{x}_j^q, \mathcal{S}) +$$

$$\frac{1}{N}\sum_{k=1}^{N} \left(L_2^k(\mathbf{c}_k^x, \{\mathbf{c}_{k'}^y\}_{k'=1}^N) + L_3^k(\mathbf{c}_k^y, \{\mathbf{c}_{k'}^x\}_{k'=1}^N) \right)$$

alpha

$$P(y = k | \mathbf{x}^q, \mathcal{S}) = \alpha P_i + (1 - \alpha) P_t \qquad (1)$$

$$P_i = P(y = k | \mathbf{x}^q, \mathcal{S}_x), P_t = P(y = k | \mathbf{x}^q, \mathcal{S}_y)$$

Beta (temperature)

$$P(y = k | \mathbf{x}^q, \mathcal{S}_x) = \frac{\exp(-\|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_k^x\|_2^2)}{\sum_{k'=1}^{N} \exp(-\|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_{k'}^x\|_2^2)} \qquad (2)$$

$$P(y = k | \mathbf{x}^q, \mathcal{S}_y) = \frac{\exp(-\|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_k^y\|_2^2)}{\sum_{k'=1}^{N} \exp(-\|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_{k'}^y\|_2^2)}$$

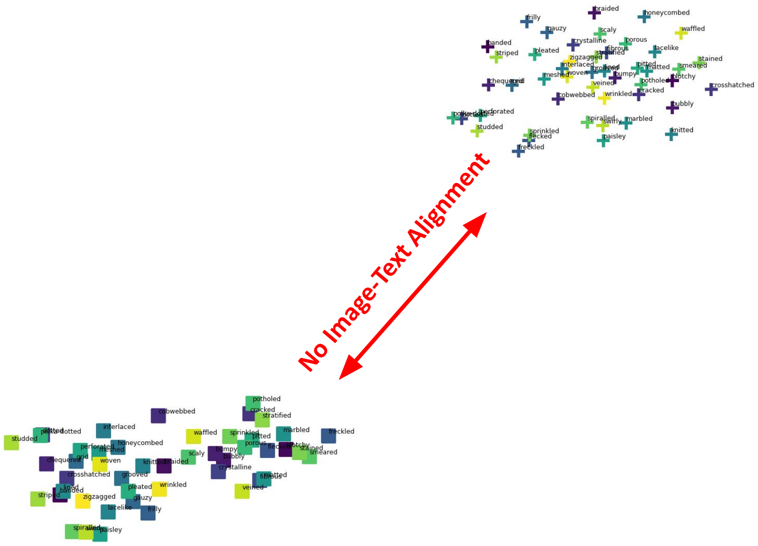
$$L_2^k(\mathbf{c}_k^x, \{\mathbf{c}_{k'}^y\}_{k'=1}^N) = -\log \frac{\exp(\mathbf{c}_k^x \cdot \mathbf{c}_k^y)}{\sum_{k'=1}^{N} \exp(\mathbf{c}_k^x \cdot \mathbf{c}_{k'}^y)}, \qquad (3)$$

$$L_3^k(\mathbf{c}_k^y, \{\mathbf{c}_{k'}^x\}_{k'=1}^N) = -\log \frac{\exp(\mathbf{c}_k^y \cdot \mathbf{c}_k^x)}{\sum_{k'=1}^{N} \exp(\mathbf{c}_k^y \cdot \mathbf{c}_{k'}^x)}$$

Config + Loss Ablation Study

```
# ------ root_path/dataset_name ------
root_path: 'DATA'

# ------ Basic Config ------
shots: 16
backbone: 'RN50'

lr: 0.0001
augment_epoch: 10
train_epoch: 100
delta: .5

# loss comments based on stanford_cars dataset
losses: ['L1', 'L2', 'L3'] # better that L1, L2, L3, L4

# losses: ['L1', 'L3', 'L4'] # just below L1, L2, L3

# losses: ['L1', 'L2'] # on par with L1+L2+L3
# losses: ['L1', 'L3'] # on par with L1+L2+L3

# losses: ['L1'] # overfits; test accuracy near 0
# losses: ['L2'] # not very effective, test accuracy just below few pc w.r.t. zero-shot
# losses: ['L3'] # helps more than L2 alone. not very effective, test accuracy just below few pc w.r.t. zero-shot
# losses: ['L4'] # doesn't help; decreases learning performance

# losses: ['L2', 'L3'] # helps more than L2, L3 alone. Not very effective, test accuracy just below few pc w.r.t. zero-shot

# losses: ['L2', 'L3', 'L4'] # doesn't help; worsens the training
```

t-SNE visualisation



Alpha: 0.5 | Beta: 1 | Acc.: 58.81%

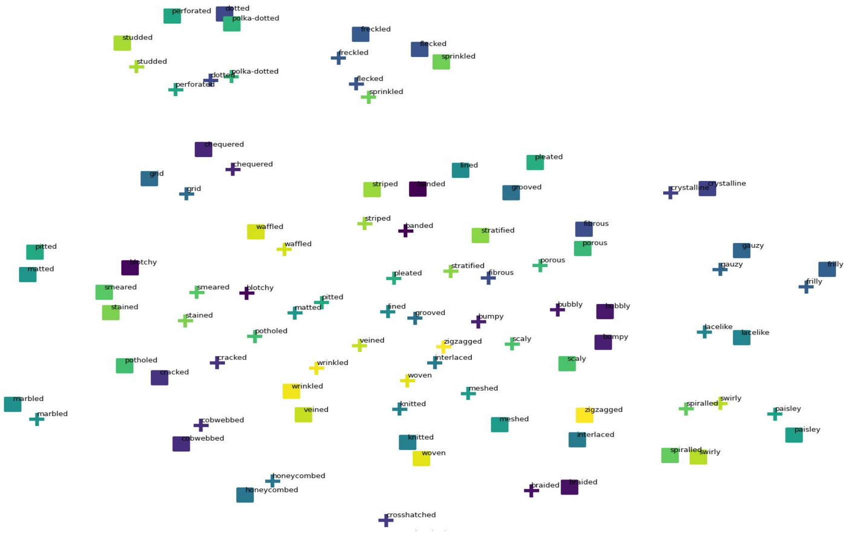


Image-Text Alignment Improvement

Alpha: 0.7 | Beta: 8 | Acc.: 68.79%

Results (Model#2)

Dataset	Zero-shot						Fine-tune					
	CLIP	Proto-CLIP	Tip-A	Tip-A^R	$\mathbf{\Delta^R} \downarrow$	$\mathbf{\Delta} \uparrow$	Proto-CLIP	Tip-A	Tip^R	$\mathbf{\Delta^R} \downarrow$	HP Searched Tip^R	$\mathbf{\Delta_F} \uparrow$
Eurosat	37.56	**73.06**	70.54	70.57	0.03	2.49	81.46	84.54	84.52	0.02	**84.56**	-3.10
DTD	42.32	**61.58**	60.93	60.93	0.00	0.65	**68.79**	66.55	66.19	0.36	66.61	2.18
UCF101	61.38	**73.14**	70.58	70.66	0.08	2.48	**78.09**	78.03	77.16	0.87	77.50	0.59
SUN397	58.56	**68.07**	66.85	66.82	0.03	1.25	**71.96**	71.47	71.35	0.12	71.35	0.61
Stanford Cars	55.63	**67.95**	66.77	66.76	0.01	1.19	**75.24**	75.74	74.93	0.81	75.09	0.15
Oxford Pets	85.83	**88.80**	88.14	88.20	0.06	0.60	88.93	89.7	**89.62**	0.08	**89.62**	-0.69
Oxford Flowers	66.06	**92.90**	89.89	89.93	0.04	2.97	**95.17**	94.8	93.87	0.93	94.60	0.85
Food101	77.33	**78.00**	77.83	77.86	0.03	0.14	78.98	77.89	79.16	1.27	**79.39**	-0.41
FGVC	17.16	29.64	**29.82**	29.82	0.06	-0.18	34.83	35.55	34.56	0.99	**35.07**	-0.24
Caltech101	85.92	**91.00**	90.18	90.18	0.00	0.82	**93.59**	92.86	92.86	0.00	92.86	0.73
Imagenet	60.34	**62.73**	62.01	61.81	0.20	0.93	**65.49**	65.51	65.40	0.11	65.48	0.01

```
eurosat, 0.7, 10, 81.31, 81.46
dtd, 0.7, 8, 68.79, 68.74
UCF101, 0.7, 10, 78.09, 78.01
sun397, 0.7, 10, 71.96, 71.93
stanford_cars, 0.8, 10, 75.24, 75.21
oxford_pets, 0.4, 10, 88.93, 88.55
oxford_flowers 0.2, 10, 86.44, 95.17
food101, 0.1, 10, 78.40, 78.98
fgvc, 0.7, 20, 34.44, 34.83
caltech101, 0.9, 10, 93.59, 93.39
imagenet, 0.4, 13, 65.49, 65.28
```

Different datasets have different alpha, beta requirements.

Questions?