Personalizing Pretrained Models

Anonymous Authors[1]

Abstract

Self-supervised or weakly supervised models trained on large-scale datasets have shown sample-efficient transfer to diverse datasets in few-shot settings. We consider how upstream pretrained models can be leveraged for downstream few-shot, multilabel, and continual learning tasks. Our model *CLIPPER* (CLIP PERsonalized) uses image representations from CLIP, a large-scale image representation learning model trained using weak natural language supervision. We developed a technique, called *Multi-label Weight Imprinting* (MWI), for multi-label, continual, and few-shot learning, and CLIPPER uses MWI with image representations from CLIP. We evaluated CLIPPER on 10 single-label and 5 multi-label datasets. Our model shows robust and competitive performance, and we set new benchmarks for few-shot, multi-label, and continual learning. Our lightweight technique is also compute-efficient and enables privacy-preserving applications as the data is not sent to the upstream model for fine-tuning. Thus, we enable few-shot, multilabel, and continual learning in compute-efficient and privacy-preserving settings.

1. Introduction

Data-efficiency and generalization are key challenges in deep learning, and representation learning has been at the heart of deep learning (Bengio, 2012). Recently, self-supervised or weakly supervised models have been leveraged to learn from large-scale uncurated datasets and have shown sample-efficient transfer (Chen et al., 2020b; Radford et al., 2021; Henaff, 2020; He et al., 2020; Devlin et al., 2019; Radford et al., 2019). However, commonly used transfer techniques, e.g., fine-tuning or distillation, do not currently support few-shot, multilabel, and continual learning.

Few-shot learning (FSL) has made great strides in the area of sample-efficient learning (Wang et al., 2020). However, FSL models are pretrained on large, domain-specific, and expensive-to-label datasets and have not leveraged pretrained models to avoid training on large and domain-specific labeled datasets. Also, FSL methods do not outperform pretrained models when domain shift is present (Chen et al., 2019; Kornblith et al., 2019).

We consider the problem of enabling *few-shot, multilabel, and continual learning* for real-world downstream tasks, and investigate combining representation learning from pretrained self-supervised or weakly supervised models with few-shot, multilabel, and continual learning techniques.

Our model **CLIPPER** (CLIP PERsonalized) uses image representations from CLIP, a weakly-supervised image representation learning model, for FSL. Inspired by Weight Imprinting (Qi et al., 2018), an FSL method, we develop an approach called Multilabel Weight Imprinting (MWI) for few-shot, multilabel, and continual learning. CLIPPER combines image representations from CLIP with MWI for continual and multilabel few-shot learning.

We evaluated CLIPPER on 10 single-label and 5 multi-label datasets. CLIPPER shows robust and competitive performance with state-of-the-art methods, e.g., FSL for MiniImagenet. We set benchmarks for few-shot, continual, and multilabel learning on several different datasets.

We make 3 key contributions.
1. A new methodology combining the flexibility of few-shot learning methods with the sample-efficiency and generalizability of transfer learning methods using self-supervised or weakly supervised pretrained models. Our method eliminates the need for data- and compute-intensive pretraining on large, domain-specific, and labeled datasets for FSL.
2. A FSL technique, leveraging pretrained representations for few-shot, continual, and multilabel learning.
3. Evaluations and benchmarks for few-shot, continual, and multilabel learning on 15 multilabel and single-label datasets, showing robust and competitive performance.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Related Work

2.1. Few-shot Learning (FSL)

There are 3 types of approaches for FSL: model-, metric-, and optimization-based. Unlike previous work, we use a pretrained models for data- and compute-efficient training.

Our Multilabel Weight Imprinting technique lies in the category of metric-based approaches (Koch et al., 2015; Snell et al., 2017; Sung et al., 2017; Vinyals et al., 2017). More specifically, we use a prototype-based metric-learning approach, as they assign trainable proxies to each category and enable faster convergence via element-category comparison, instead of element-wise comparisons. Our work extends a previous FSL technique, called weight imprinting (Qi et al., 2018). We not only use a pretrained base model (Qi et al., 2018), instead of training a base network from scratch, but also extend weight imprinting to enable multilabel and continual learning. Other metric-based methods are complementary to our approach and our model can be extended, e.g., with MatchingNet attention (Vinyals et al., 2017) or RelationNet relations (Sung et al., 2017).

Model-based methods (Santoro et al., 2016; Munkhdalai & Yu, 2017) use especially designed models for rapid parameter updates, and optimization-based techniques (Ravi & Larochelle, 2016; Finn et al., 2017; Nichol et al., 2018) adjust the optimization method to meta-learn efficiently. Recent research indicates that learning a good embedding model can be more effective than sophisticated meta-learning algorithms (Tian et al., 2020) and efficient meta-learning may be predominantly due to the reuse of high-quality features (Raghu et al., 2019). Nonetheless, these techniques, though relatively training-intensive, are complementary to our work and may be used to improve both upstream and downstream models.

2.2. Self-supervised Representation Learning

Self-supervised and weakly supervised models have been used in natural language processing (Dai & Le, 2015; Radford et al., 2018; Devlin et al., 2019) and computer vision (Henaff, 2020; He et al., 2020; Chen et al., 2020a;b; Radford et al., 2021) to learn from large-scale unlabeled or weakly labeled datasets. Though pre-training is still imperfect (Ericsson et al., 2020; Mahajan et al., 2018), pretrained models trained on large-scale datasets have shown robust and sample-efficient transfer to diverse tasks (He et al., 2020; Henaff, 2020; Chen et al., 2020b; Radford et al., 2021).

Transfer learning is related to few-shot learning, but FSL does not use a pretrained method. Instead, FSL is trained and evaluated using the same distribution and does not necessarily outperform transfer learning when domain shift is present (Chen et al., 2019). Transfer learning, however, could benefit from the specialized FSL techniques (Kornblith et al., 2019). Also, to the best of our knowledge, unlike our work, transfer learning using pretrained models has not been combined with multi-label and continual learning.

Like previous work, we use self-supervised data augmentation to boost FSL (Gidaris et al., 2019; Qi et al., 2018).

2.3. Multilabel and Continual Learning

Continual learning techniques (Mai et al., 2021) include regularization-based methods (e.g., Elastic Weight Consolidation), memory-based methods (e.g., Incremental Classifier and Representation Learning (Rebuffi et al., 2017)), and parameter isolation (like Continual Neural Dirichlet Process Mixture). Previously used continual techniques, however, did not use pretrained models for few-shot, multilabel, and continual learning.

Common multilabel classification techniques include ML-kNN, Multi-label DecisionTree, etc (Devkar & Shiravale, 2017). Multi-Label Image Classification has also been done using knowledge distillation from weakly supervised detection (Liu et al., 2018a). However, none of the existing methods combine multilabel, continual, and few-shot learning, especially using pre-trained models. Several multilabel and continual learning techniques, nonetheless, are complementary to our work and can be extensions of our work.

3. Approach

3.1. Desiderata

We outline 3 desiderata for real-world computer vision applications. First, **few-shot learning** so that the applications can start well in data-scarce scenarios and can also be customized and personalized for different needs. Second, **continual learning** to incrementally learn new information and avoid catastrophic forgetting, e.g., replacing of older classes when new ones are added. Third, **multilabel learning** as the right label may not be just one label but a subset of all the given labels, including 0 to all labels. The multi-label case is important for not only assigning multiple labels to a particular data point but also for assigning zero labels, in case we get data points that we currently do not have labels for, i.e., the continual learning case. Continual learning often considers the addition of data points along with their respective labels. However, we consider the more realistic continual case when a point may be added even before their label is added and thus, the model needs to assign no label.

3.2. Decisions

We made the following three design choices to enable few-shot, multilabel, and continual learning.

Pretrained base model: FSL models are typically pre-

trained on large domain-specific training sets, which contain examples not in the support/test set. The models are then trained and tested on support and test sets, which have the same classes. Large-scale, domain-specific, and labeled datasets, however, may not always be available in real-world settings. Also, FSL models trained on domain-specific sets may not generalize well to domain shifts (Hu et al., 2021). Large-scale self-supervised or weakly supervised models, on the other hand, learn good representations and can be fine-tuned for data-efficient and diverse downstream tasks (Chen et al., 2020b; Radford et al., 2021). We use pretrained models trained on diverse datasets as base models for FSL, instead of training base models from scratch on domain-specific datasets. As a result, unlike FSL methods, we only train with a support test, which we call train set.

Weight Imprinting (WI): Weight imprinting (Qi et al., 2018) is a FSL learning method that learns a linear layer on top of the embeddings, where the columns of the linear layer weight matrix are prototype embeddings for each class. Many self-supervised or weakly supervised models have been shown to learn linearly-separable embeddings using linear probes (Radford et al., 2021) and a linear layer can be added to pretrained embeddings to classify different classes. Compared to traditional transfer learning techniques with a fixed number of classes, WI adds new classes as new columns of the linear layer weight matrix, making adding classes computationally and conceptually simpler and avoiding catastrophic forgetting. Thus, weight imprinting supports prototype-based few-shot and continual learning.

Sigmoids, not Softmax: The original weight imprinting model uses softmax and thus is compatible with single-label classification. We replace the softmax with sigmoid activations for each class in weight imprinting to enable multi-label learning. Sigmoids also support an output of 0 labels for continual learning, i.e., when the label for a given data point has not yet been added to the label set.

3.3. Details

We created a multilabel version of weight imprinting (Qi et al., 2018), called **Multilabel Weight Imprinting (MWI)**. Our model has two parts. First, an embeddings extractor, $\phi : \mathbb{R}^N \rightarrow \mathbb{R}^D$, maps input image $x \in \mathbb{R}^N$ to a D-dimensional embedding vector $\phi(x)$, followed by an L_2 norm. Second, a sigmoid function, $f(\phi(x))$, maps the embedding using sigmoid activations for each category.

$$f_i(\phi(x)) = \frac{1}{1 - exp(-w_i^T \phi(x))}$$

where w_i is the i-th column of the weight matrix normalized to unit length (with no bias term).

Each column of the WI matrix is a template of the corresponding category. The linear layer computes the inner product between the input embeddings $\phi(x)$ and each template embedding w_i. The result represents 'close-by' templates in the embedding space using a threshold function.

$$\hat{y} = sgn(w^T \phi(x) - \vartheta)$$

where sgn is the sign function and ϑ is the threshold.

4. Implementation

We share our implementation details below and model architecture in Fig 1, and algorithm in Appendix.

Embeddings Generator: Weight imprinting (Qi et al., 2018) uses a base classifier trained on "abundant" labeled training samples. We replace the base classifier with CLIP (ViT B/32), a pretrained weakly supervised model (Radford et al., 2021). We do not re-train or fine-tune the weights of the pretrained CLIP model. As shown in section 6 (Figure 2), we compared embeddings from different supervised, self-supervised, and weakly supervised models, and chose CLIP because it had the best FSL performance using WI.

Image Embeddings: We embed images using CLIP's vision transformer and then use the normalized embeddings for multilabel weight imprinting. Compared to weight imprinting (Qi et al., 2018), which used 64-dimensional embeddings, we use 512-dimensional embeddings from CLIP. Qi et al. (Qi et al., 2018) also tried 512-dimensional embeddings and reported no significant effects on the results.

Multilabel Weight Imprinting (MWI): The MWI layer is a single dense layer with an input size equal to the embedding size of the embeddings generator and output equal to the number of classes. We initialize the MWI weights as an average of the embeddings for each class corresponding to the weight column. We normalize the weights columns and use sigmoid activations with a threshold. When training the MWI layer, we use the binary cross-entropy loss with an Adam optimizer (Kingma & Ba, 2014).

MWI+ = MWI + Training (T) + Augmentations (A): When training with non-trivial (nt) augmentations, we use 3 types of augmentations (Chen et al., 2020a): i. random crop, resize, and random horizontal flip; ii. random color jitter; iii. random Gaussian blur. Trivial (t) augmentations refer to repeating the image.

Continual Learning (CL) We use Experience Replay (ER) (Lin, 1992), which involves keeping a memory of old data and rehearsing it. ER has been used for CL (Rolnick et al., 2018; Chaudhry et al., 2019; Hayes et al., 2019) and has been shown to outperform many CL approaches with and without a memory buffer (Chaudhry et al., 2019). In our multilabel continual learning setting, we retrain the old data with *having/not having* the new label when new labels are received.

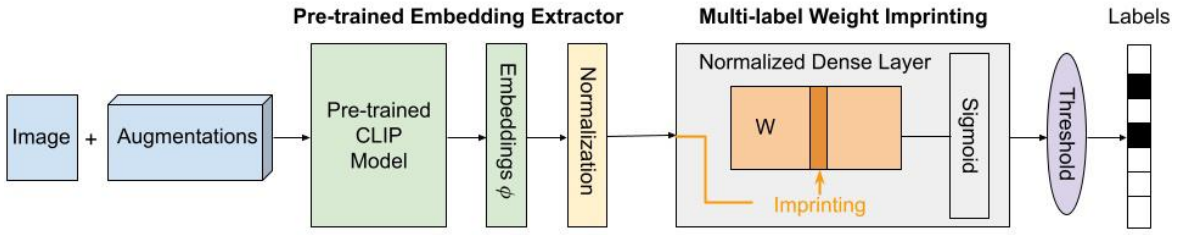

Figure 1. CLIPPER uses pre-trained embeddings with Multilabel Weight Imprinting for few-shot, multilabel, and continual learning.

Table 1. Our datasets and their abbreviations

Dataset Name	Abbr.
Single	
Omniglot (Few-shot)	OM
MiniImagenet (Few-shot)	MI
Labeled Faces in the Wild	LFW
UCF101 (Action videos)	UCF
Imagenet-R (Art)	IR
Imagenet-Sketch	IS
Indoor Scene Recognition	ISR
CIFAR10	C10
Imagenet-A (Adversarial)	IA
Colorectal Histology (Medical)	CH
Multi-label	
CelebA Attributes	CAA
UTK Faces	UTK
Yale Faces	YF
Common Objects in Context	COCO
iMaterialist Fashion (Fine-grained)	IM

5. Experiment Study

5.1. Datasets

We selected 10 single-label and 5 multi-label datasets based on 5 reasons: i. *Few-shot learning:* We added commonly used datasets for FSL; ii. *Diversity:* We included diverse datasets to evaluate performance under distributional and task shifts; iii. *Robustness:* We also picked an adversarial example dataset to evaluate robustness; iv. *Multilabel settings:* We chose multilabel datasets, including object detection, fine-grained detection, and overlapping labels; v. *AI for good:* We included a medical dataset to illustrate the broader impact of our work. Our dataset list is in Table 1.

5.2. Evaluations

We compared FSL in 7 settings: i. using different embeddings generators; ii. using sigmoid (MWI) versus softmax (WI) activations; iii. with and without training (T) and augmentations (A), both trivial (t) and non-trivial (nt) augmentations; iv. in 4 FSL settings like (Sung et al., 2017)): (5-way 5-shot, 15 test; 20-way 5-shot, 5 test; 5 way 1 shot, 19 test; 20-way 1-shot, 10 test); v. in continual learning settings; vi. with CLIP's zero-shot and FSL linear probe; vii. with state-of-the-art (SOTA) results – there are no previous few-shot, multi-label, and continual learning evaluations, but we compare with FSL and also full training/test set evaluations. All evaluations are 5-way 5-shot, except for Ch (results in Appendix). We randomly sample classes and data points from each dataset 100 times and average the results.

5.3. Metrics

Commonly used single-label classification metrics, e.g., top-1 accuracy, are not applicable in multilabel settings. Multilabel evaluations have used different metrics, including class and overall precision, recall, and F1, as well as mean average precision (mAP) (Wang et al., 2016). We calculated a total of 13 diverse metrics for each of our evaluations and included all the results in the supplementary materials.

We primarily use overall **F1-score** in this paper since F1-score accounts for class imbalance, which may be present in multilabel datasets, especially in real-world settings. The only downside of F1-score is that compared to mAP, it is threshold-dependent. However, in real-life situations, the threshold is also important, and therefore, we also discuss the optimal cut-off thresholds for our evaluations.

To compare our results with state-of-the-art (SOTA) results, we also report the metrics used by different SOTA results, i.e., top-1 accuracy for single-label datasets and average class accuracy (cAc) for multi-label datasets, except COCO, which uses mAP. We report these metrics along with F1-scores so that the F1-scores can be compared to the different SOTA metrics. SOTA references are in Table 3.

6. Results

We share our main results in this section and ablations in the next. First, CLIP+WI performance is similar to CLIP's linear probe performance, possibly because both are linear

layers (Fig 2). Second, without training and augmentations, MWI performs worse than WI for single-label classifications – MWI F1-score is worse than WI F1-score (Fig 2), even though the accuracies are comparable (Fig 3). Third, MWI with training (50-80 epochs) and augmentations (10 trivial/non-trivial), i.e., MWI+, does at least as well as WI using CLIP (Fig 2-3), CLIP's linear probes (Fig -2), and state-of-the-art baselines (Fig 3). MWI and MWI+ are also compared with SOTA and CLIP's baselines in Table 2.

Comparing CLIP's embeddings: We compared embeddings from different pretrained supervised (Resnet50 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), Inception V3), self-supervised (SimCLR v2 (Chen et al., 2020b), MoCo v2 Resnet50 (Chen et al., 2020a;c), PCL Resnet50 (Tang et al., 2018), SwAV Resnet50 (Caron et al., 2020)), and weakly supervised (CLIP (Radford et al., 2021)) models (Figure 2 left). CLIP is trained on 400 million images, while the others are on 14 million Imagenet images. We have three key findings. First, CLIP gives the best results, possibly because CLIP is trained on a bigger dataset than other pretrained models. Second, SimCLR's performance is closest to CLIP, even though it was trained only on a smaller dataset than CLIP. Third, Resnet50-based models performed much worse than the other models, even though all models, except CLIP, were trained on Imagenet.

SOTA caveats: There are 3 caveats to our SOTA comparisons (Table 2). First, to the best of our knowledge, there is no prior work on multilabel few-shot learning and hence, we are setting new benchmarks and have no direct prior work to compare with. Second, even though we list the SOTA 5-way 5-shot results for OM and MI, there are two main differences: i. Previous few-shot results were pre-trained on large-scale, domain-specific, and labeled datasets, whereas our model is trained only on the few-shot set. Thus, performance for new domains like OM may not be as good as few-shot models pre-trained on OM; ii. Also, previous few-shot works did not do multilabel few-shot learning. Third, we list SOTA for other datasets, which have previously not been evaluated for few-shot learning, so the SOTA results are for full datasets and we only list them as a reference. [1]

CLIP baselines: Since we use embeddings from CLIP, we also compare our CLIP + MWI results to CLIP's linear probe, zero-shot, and CLIP + WI performance. We use both F1-score and accuracy, and all comparisons, other than zero-shot, are 5-way 5-shot. MWI+ using CLIP is comparable to CLIP's baselines, but unlike the linear probe, also enables few-shot, multilabel, and continual learning.

[1]Both 1. Ia and Ir & 2 in Table 2 use CLIP but 2 uses the ViT B/32 architecture whereas 1. Ia and Ir use the ViT L/14-336px architecture. L/14-336px is a bigger and better performing architecture but is not public (Radford et al., 2021)

7. Ablations

7.1. MWI: Without Training and Augmentations

We compare CLIPPER's 5-way 5-shot performance on 9 **single-label datasets** (Figure 4 left) and 5 **multilabel datasets** (Figure 4 right). For single-label, we perform two evaluations: i. Weight imprinting with softmax activations (f1-score and accuracy); ii. Multilabel weight imprinting with sigmoids (f1-score, top-1 accuracy, and per-class accuracy). For multilabel datasets with bounding boxes, i.e., COCO and iMaterialist, we compared full-full and patch-patch configurations, where 'full' represents the full image and 'patch' represents the bounding box of the relevant object. In n-m, n represents the training configuration and m represents the testing configuration.

We had three key findings (Fig 4). First, for single-label datasets, WI accuracy is comparable to MWI top-1 accuracy, which means that the sigmoid activation can get us comparable results to the softmax activation. Though, as expected, MWI F1-scores are much lower in value than MWI Top-1 accuracy. Second, multi-label datasets on average have much lower performance than single-label datasets, which is expected as they have more labels than single-label datasets. Third, the patch-patch configuration works best for iMaterialist whereas the full-full configuration works best for COCO, possibly because the background is meaningful in COCO but mostly white in iMaterialist.

7.2. MWI+: With Training and Augmentations

We evaluated the performance of Multilabel Weight Imprinting by adding **training (T) and augmentations (A)** (Figure 5). We had three key findings. First, CLIPPER's performance improved with both training and augmentations – after training and augmentations, F1-scores for multi-label weight imprinting were comparable to the F1-scores for weight imprinting with softmax. Second, the performance saturates around 50-80 epochs, and trivial (t) augmentations, i.e., image repetitions, are as good or sometimes even better than non-trivial (nt) augmentations. Third, with training, the best threshold values stabilized around 0.5 for most datasets (Figure 8 (left). We also compared 4 **few-shot learning settings** (Figure 6): 5-way 5-shot, 20-way 5-shot, 5 way 1 shot, 20-way 1-shot. The performance worsens with decreasing shots and with increasing classes.

7.3. Continual learning

We evaluated 5 way 5 shot continual learning. We incrementally added the number of labeled classes and their respective training data and labels, while keeping the test set fixed. We had three key findings. First, CL performance (Figure 7) varies with the number of classes but reaches approximately the same 5-way 5-shot value with continual

Table 2. Comparing CLIPPER Multilabel Weight Imprinting (MWI) with SOTA and CLIP baselines[1]

	C10	Ia	Ir	Is	Isr	Lfw	Mi	Om	Ucf	Ca	Co	Im	Ut	Yf	Ch
1	.91	.85	.89	.60	.74	1.0	.92	1.0	.99	.82	.84	.72	.86	.85	.93
2	.91	.75	.82	.91	.95	1.0	.95	.96	.95						
3	.89	.75	.79	.89	.94	1.0	.94	.94	.94	.69	.88	.79	.74	.88	
4	.70	.54	.60	.72	.83	.94	.78	.66	.83	.62	.75	.48	.60	.66	
5	.90	.74	.80	.90	.94	1.0	.94	.95	.94	.69	.73	.56	.78	.79	.69
6	.96	.90	.92	.96	.98	.99	.98	.98	.98	.76	.89	.83	.86	.91	.92
7	.91	.77	.83	.92	.96	1.0	.95	.97	.95	–	.87	–	–	–	

SOTA(1); **CLIP** Lin. Probe Ac(2); **MWI** Ac(3),F1(4); **MWI+T+A** F1(5),CAc(6),Top1/mAP(7)

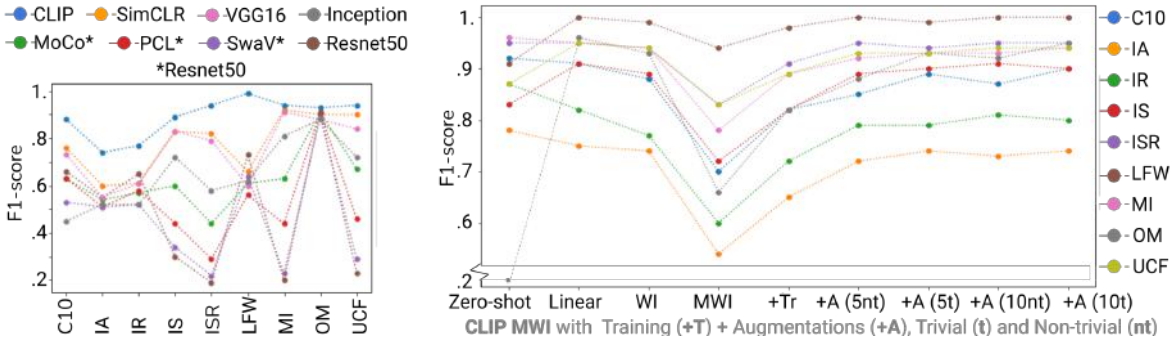

Figure 2. Comparing embeddings models (left) and Multilabel Weight Imprinting results (right).

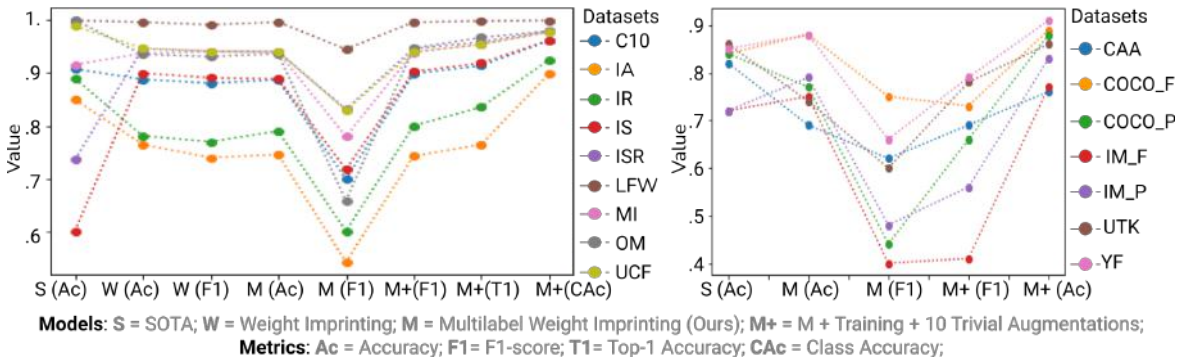

Figure 3. Comparing SOTA, WI, and MWI for single-label (left) and multi-label (right) datasets.

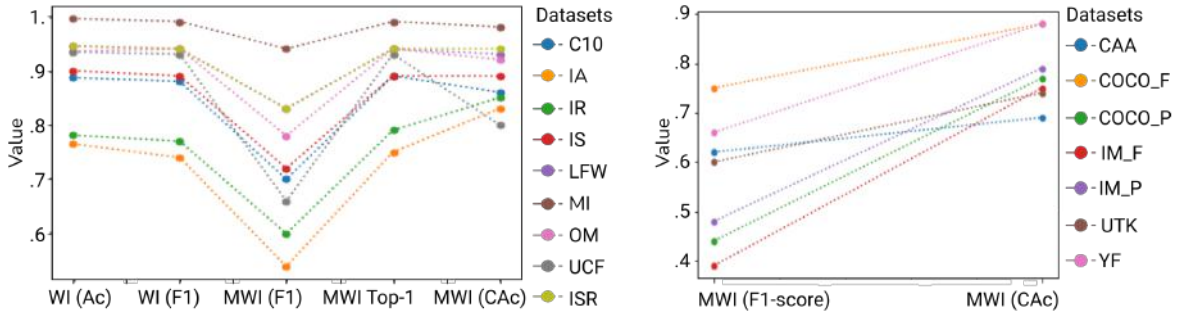

Figure 4. Comparing metrics, without training and augmentations, for single- and multi-label datasets.

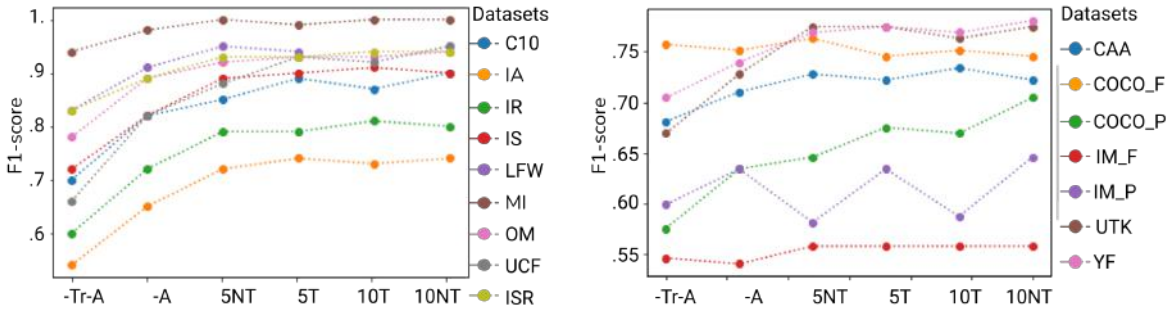

Figure 5. Comparing MWI with training and augmentations for single- and multi-label datasets

.

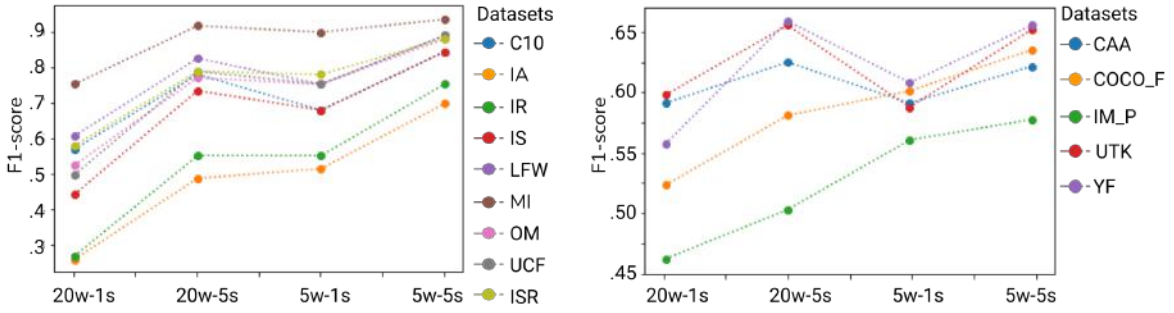

Figure 6. Comparing different few-shot settings with MWI+ (L:single-label, R:multi-label datasets)

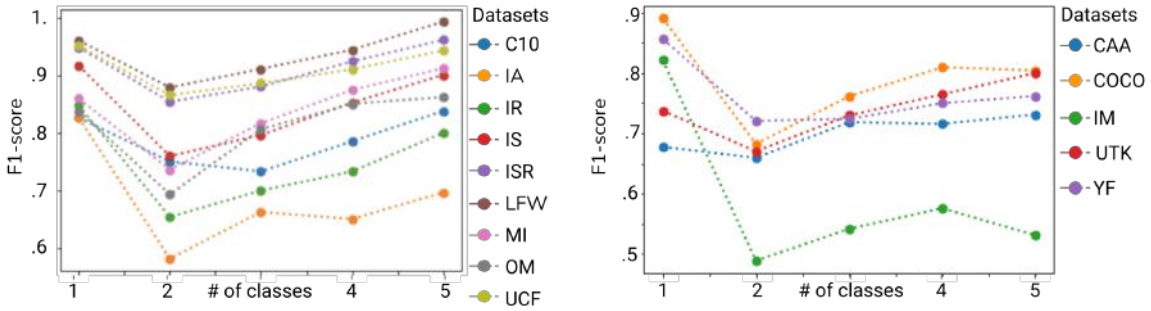

Figure 7. MWI+ Continual learning results for increasing classes (L:single-, R:multi-label dataset)

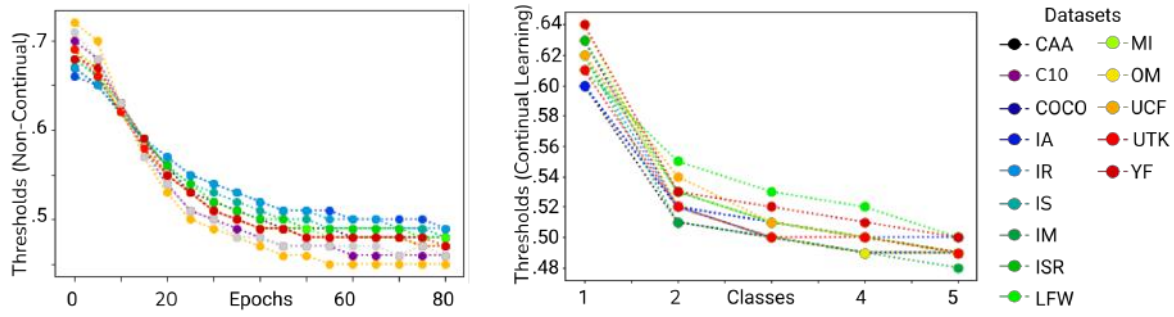

Figure 8. Optimal thresholds with (left) and without (right) continual learning for all datasets.

learning as it does without continual learning (5). Second, the optimal-performance thresholds vary with the number of classes and we share the best accuracies and their respective thresholds for each dataset for different number of classes (Figure 8 right). Third, the thresholds are higher with lower number of classes, possibly because of lesser training data, but converge to approximately the same 5-way 5-shot value with and without continual learning (Fig 8).

8. Discussion and Limitations

With advances in representation learning, the question arises: *how to best use the representations in downstream tasks*. Previous work suggests, "combining the strength of zero-shot transfer with the flexibility of few-shot learning is a promising direction" (Radford et al., 2021) and "obtain better results...by combining few-shot learning methods with fine-tuning" (Kornblith et al., 2019).

We outline few-shot, continual, and multilabel learning as the desiderata for downstream tasks and introduce a technique, called Multilabel Weight Imprinting, to meet the desiderata. Our model uses embeddings from a pertained CLIP model and shows promising performance on diverse and challenging tasks. We set few-shot, multilabel, and continual learning benchmarks for many datasets.

Our work has 3 **key findings**. First, using pretrained models with an existing FSL technique, i.e., weight imprinting (Qi et al., 2018), enables sample-efficient learning with 2 additional benefits: i. Unlike commonly-used transfer learning techniques like fine-tuning and distillation, we have a prototype for each class and can flexibly add/update each class prototype without influencing (e.g., forgetting) the other class prototypes; ii. Unlike commonly-used FSL methods, the base model need not be trained with computationally-intensive techniques involving large, domain-specific, and expensive-to-label datasets. Second, replacing weight imprinting's softmax function with a sigmoid and threshold function enables multilabel weight imprinting, and using training and augmentations helps improve performance. Third, adding experience reply enables continual learning.

Our work has 3 **key limitations**: i. Multilabel learning has poorer and threshold-dependent performance compared to single-label learning, but multi-label learning is still more realistic than single-label classification as even single-label datasets have multiple labels (Yun et al., 2021); ii. Prototype-based few-shot learning scales the number of prototypes with the number of classes and comparing with every single prototype may not be efficient. Thus, efficient and scalable methods, e.g., hierarchical prototypes, are needed; iii. Experience replay for multilabel continual learning is memory-inefficient and memory-efficient continual learning, e.g., prototype-based contrastive learning, could be leveraged.

We have 3 **key future directions**: i. Use downstream few-shot learning for error correcting labels from upstream models; ii. Make few-shot, multilabel, and continual learning memory-efficient, robust, and deployable; iii. Deploy and test in real-world settings, e.g., human-in-the-loop personalized applications.

9. Broader Impact

We highlight 3 key areas of positive impact. First, we designed our model for few-shot, multilabel, and continual learning to enable real-world sample-efficient applications, including personalized and AI for good applications (more details in appendix). Second, since we do not train the upstream model, the data does not have to be sent to the upstream model, affording privacy-preserving and offline model training. Third, since we only train a linear layer, our model affords easy and lightweight real-world training and deployment, including on mobile and wearable devices, especially if the pretrained base model are mobile-optimized (Howard et al., 2017) as in (Khan & Maes, 2021). We have made our model flexible, easy-to-use, and easy-to-train – it can be used with any state-of-the-art pre-trained model, trained and run using free Google Colab notebooks, and personalized using only a few examples. Few-shot and personalized learning may also help mitigate data/labeling bias. Our work will hopefully enable stakeholders to ethically design and deploy personalized, privacy-preserving, and meaningful real-world deep learning applications.

10. Conclusion

Data-efficiency and generalization are key challenges for deep learning. Self-supervised or weakly supervised models trained on unlabeled or uncurated datasets have shown promising transfer to few-shot tasks. Few-shot learning methods have also demonstrated sample-efficient learning.

We highlight the need for few-shot, multilabel, and continual learning, and developed Multi-label Weight Imprinting (MWI) for few-shot, continual, and multi-label learning. Unlike previous FSL techniques, our model, CLIPPER, uses MWI with pretrained representations from a weakly-supervised model, i.e., CLIP. Thus, CLIPPER combines the sample-efficiency and generalizability of transfer learning with the flexibility and specialization of FSL methods.

CLIPPER shows robust and competitive performance and is a step in the direction of using pretrained models for few-shot, multilabel, and continual learning. Our model is also lightweight and the data does not have to be sent back to the upstream model, enabling privacy-preserving and on-device downstream training. Thus, our model enables few-shot, multilabel, and continual learning, especially for easy-to-train, light-weight, and privacy-preserving applications.

References

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv:2002.05709 [cs, stat]*, June 2020a. URL http://arxiv.org/abs/2002.05709. arXiv: 2002.05709.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv:2006.10029 [cs, stat]*, October 2020b. URL http://arxiv.org/abs/2006.10029. arXiv: 2006.10029.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Chrysos, G. G., Moschoglou, S., Bouritsas, G., Panagakis, Y., Deng, J., and Zafeiriou, S. P-nets: Deep polynomial neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7325–7335, 2020.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. *arXiv preprint arXiv:1511.01432*, 2015.

Devkar, R. and Shiravale, S. A survey on multi-label classification for images. *International Journal of Computer Application*, 162(8):39–42, 2017.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv: 1810.04805.

Ericsson, L., Gouk, H., and Hospedales, T. M. How well do self-supervised models transfer? *arXiv preprint arXiv:2011.13377*, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*, July 2017. URL http://arxiv.org/abs/1703.03400. arXiv: 1703.03400.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23 (6):643–660, 2001.

Ghosh, S., Bandyopadhyay, A., Sahay, S., Ghosh, R., Kundu, I., and Santosh, K. Colorectal histology tumor detection using ensemble deep neural network. *Engineering Applications of Artificial Intelligence*, 100:104202, 2021.

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., and Cord, M. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8059–8068, 2019.

Guo, S., Huang, W., Zhang, X., Srikhanta, P., Cui, Y., Li, Y., Adam, H., Scott, M. R., and Belongie, S. The imaterialist fashion attribute dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Hayes, T. L., Cahill, N. D., and Kanan, C. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9769–9776. IEEE, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hu, D., Lu, Q., Hong, L., Hu, H., Zhang, Y., Li, Z., Shen, A., and Feng, J. How well self-supervised pre-training performs with streaming data? *arXiv preprint arXiv:2104.12081*, 2021.

Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

Kalfaoglu, M., Kalkan, S., and Alatan, A. A. Late temporal modeling in 3d cnn architectures with bert for action recognition. *arXiv preprint arXiv:2008.01232*, 2020.

Kärkkäinen, K. and Joo, J. Fairface: Face attribute dataset for balanced race, gender, and age. *arXiv preprint arXiv:1908.04913*, 2019.

Kather, J. N., Weis, C.-A., Bianconi, F., Melchers, S. M., Schad, L. R., Gaiser, T., Marx, A., and Z"ollner, F. G. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.

Kehrenberg, T., Bartlett, M., Thomas, O., and Quadrianto, N. Null-sampling for interpretable and fair representations. In *European Conference on Computer Vision*, pp. 565–580. Springer, 2020.

Khalili Mobarakeh, A., Cabrera Carrillo, J. A., and Castillo Aguilar, J. J. Robust face recognition based on a new supervised kernel subspace learning method. *Sensors*, 19(7):1643, 2019.

Khan, M. and Maes, P. Pal: Intelligence augmentation using egocentric visual context detection. *arXiv preprint arXiv:2105.10735 [cs]*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koch, G., Zemel, R., and Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, J., Chao, F., Yang, L., Lin, C.-M., and Shen, Q. Decoder choice network for meta-learning. *arXiv preprint arXiv:1909.11446*, 2019.

Liu, Y., Sheng, L., Shao, J., Yan, J., Xiang, S., and Pan, C. Multi-label image classification via knowledge distillation from weakly-supervised detection. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 700–708, 2018a.

Liu, Z., Luo, P., Wang, X., and Tang, X. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15 (2018):11, 2018b.

Luo, Y., Wong, Y., Kankanhalli, M., and Zhao, Q. Direction concentration learning: Enhancing congruency in machine learning. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 181–196, 2018.

Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., and Sanner, S. Online continual learning in image classification: An empirical survey. *arXiv preprint arXiv:2101.10423*, 2021.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Munkhdalai, T. and Yu, H. Meta Networks. *arXiv:1703.00837 [cs, stat]*, June 2017. URL http://arxiv.org/abs/1703.00837. arXiv: 1703.00837.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Qi, H., Brown, M., and Lowe, D. G. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5822–5830, 2018.

Quattoni, A. and Torralba, A. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413–420. IEEE, 2009.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *OpenAI*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Rahimzadeh, M., Parvin, S., Safi, E., and Mohammadi, M. R. Wise-srnet: A novel architecture for enhancing image classification by learning spatial resolution of feature maps. *arXiv preprint arXiv:2104.12294*, 2021.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. *https://openreview.net/forum?id=rJY0-Kcll*, November 2016.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. P., and Wayne, G. Experience replay for continual learning. *arXiv preprint arXiv:1811.11682*, 2018.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical Networks for Few-shot Learning. *arXiv:1703.05175 [cs, stat]*, June 2017. URL http://arxiv.org/abs/1703.05175. arXiv: 1703.05175.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017. URL http://arxiv.org/abs/1711.06025.

Tang, P., Wang, X., Bai, S., Shen, W., Bai, X., Liu, W., and Yuille, A. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., and Isola, P. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching Networks for One Shot Learning. *arXiv:1606.04080 [cs, stat]*, December 2017. URL http://arxiv.org/abs/1606.04080. arXiv: 1606.04080.

Wang, H., Ge, S., Xing, E. P., and Lipton, Z. C. Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549*, 2019.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2285–2294, 2016.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Yun, S., Oh, S. J., Heo, B., Han, D., Choe, J., and Chun, S. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *arXiv preprint arXiv:2101.05022*, 2021.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

A. Method and Algorithm

We describe our problem definition below and the algorithm in Algorithms 1 - 6.

We have a train set of N labeled examples: $S = \{(x_1, y_1), ..., (x_N, y_N)\}$ where each $x_i \in \mathbb{R}^D$ is the D-dimensional feature vector of an example and y_i is a subset of labels $L = \{1, ..., K\}$. S_k denotes the set of examples labeled with class k and S_j and S_k are not necessarily disjoint sets for $j \neq k$.

Moreover, in continual learning setting, the training set S and the labels L can grow over time. $S^t = \{(x_1, l_1), ..., (x_M, l_M)\}$ and $L^t = \{1, ..., P\}$ and $S^t = \{(x_1, l_1), ..., (x_N, l_N)\}$ and $L^t = \{1, ..., Q\}$ where $N > M$ and $Q > P$, i.e., the size of labels and the training set are non-decreasing over time.

B. Real-world Applications

We discuss the real-world applications of our work using a personalized photo-labeling application and few-shot results for a medical imaging dataset.

Personalized photo labeling: We designed a personalized photo labeling application to enable personalized phot labeling using CLIPPER's. The interface (Figure 9) shows how users can label specific images and also, how clustering can help efficient labeling by clustering similar images. We share the clustering accuracies in Figure 10 – image embeddings from each embedding generator were passed to UMAP (McInnes et al., 2018), resulting in 2-dimensional embeddings, which we used for clustering.

Real-world Medical Imaging: We tested multilabel few-shot learning on the colorectal cancer histology dataset (Kather et al., 2016). Our model, trained with 5 examples per class, shows competitive performance compared to the state-of-the-art model, which was trained using the full dataset (Ghosh et al., 2021) (Fig 11).

C. Experiments and Results

We share references for our datasets and their respective SOTA results in Table 3. Our metric details in Table 4.

For UCF and ISR, the images are resized such that the smaller dimension is 224 followed by a center crop of 224x224. For all other datasets, the images are resized to 224x224. Since UCF has videos, we select the middle frame as the target image.

In the rest of the tables, we show the detailed numerical results for all our evaluations, including error bars (Table C) for our few-shot evaluations.

Algorithm 1 Augmentations Function

Input: Original image, **x**
Output: Augmented image, **x'**

function AUGMENT($\mathbf{x}$)
  $\mathbf{x}' = \text{RANDOMCROPANDRESIZE}(\mathbf{x})$
  $\mathbf{x}' = \text{RANDOMHORIZONTALFLIP}(\mathbf{x}')$
  $\mathbf{x}' = \text{RANDOMCOLORJITTER}(\mathbf{x}')$
  $\mathbf{x}' = \text{RANDOMGAUSSIANBLUR}(\mathbf{x}')$
  return $\mathbf{x}'$

Algorithm 2 Embedding Function

Input: Dataset to embed D, Number of augmentations for each image N_A
Output: Embedded dataset E

function CLIPEMBEDDATASET(D, N_A)
  $E \leftarrow \{\}$ ▷ Embedded dataset
  for $(\mathbf{x_i}, y_i) \in D$ **do**
    for $j \in \{1, 2, \cdots N_A\}$ **do**
      $\mathbf{x_i'} \leftarrow \text{AUGMENT}(\mathbf{x_i})$ ▷ If augmenting(ablation)
      $\phi_i \leftarrow \text{CLIP}(\mathbf{x_i'})$
      $\phi_i' \leftarrow \text{NORMALIZE}(\phi_i)$
      $\text{APPEND}(E, (\phi_i', y_i))$
  return E

Algorithm 3 Training Function

Input: Embedded dataset E, Weight Imprinting single layer f, Number of training epochs e_{train}
Output: Trained weight imprinting layer f

function TRAIN(E, f, e_{train})
  $J \leftarrow 0$
  for $(\phi_i', y_i) \in E$ **do**
    $y_i' = \sigma(f(\phi_i')$
    $J \leftarrow J + \text{BINARYCROSSENTROPY}(y_i, y_i')$
  $J \leftarrow \text{MEAN}(J)$
  $\text{OPTIMIZE}(f, J, Adam, e_{train})$
  return f

Algorithm 4 Predicting Function

Input: Embedding ϕ', Weight Imprinting single layer f, Evaluation threshold $threshold$
Output: Predicted labels $labels$

function PREDICT($\phi', f, threshold$)
  $labels \leftarrow \{\}$
  $y_i' = \sigma(f(\phi_i')$
  for $j \in \{1, 2, \cdots W\}$ **do**
    if $y_{ij}' \geq threshold$ **then**
      $\text{APPEND}(labels, j)$
  return $labels$

Algorithm 5 Sampling Function

Input: Number of ways/labels n_{ways}, Number of shots per label n_{shot}, Number of test images per label n_{test}, Number of episodes $n_{episodes}$, Dataset to sample from D
Output: Few shot train dataset D'_{train}, Few shot test dataset D'_{test}

function SAMPLE($n_{way}, n_{shot}, n_{test}, n_{episodes}, D$)
  $D'_{train} = \{\}$
  $D'_{test} = \{\}$
  $L \leftarrow$ GETUNIQUELABELS(D)
  for $i \in \{1, 2, \cdots n_{episodes}\}$ **do**
    $d'_{train} \leftarrow \{\}$
    $d'_{test} \leftarrow \{\}$
    $L_{sampled} \leftarrow$ RANDOMSAMPLE(L, n_{way}) ▷ Sample n_{way} labels
    for $l \in L_{sampled}$ **do**
      $D_l \leftarrow \{(\mathbf{x_i}, y_i) \in D \mid l \in y_i\}$
      $d_1 \leftarrow$ RANDOMSAMPLE($D_l \setminus (d'_{train} \cup d'_{test}), n_{shot}$)
      APPEND(d'_{train}, d_1)
      $d_2 \leftarrow$ RANDOMSAMPLE($D_l \setminus (d'_{train} \cup d'_{test} \cup d_1), n_{test}$)
      APPEND(d'_{test}, d_2)
    APPEND(D'_{test}, d'_{test})
    APPEND(D'_{train}, d'_{train})
  return D'_{train}, D'_{test}

Algorithm 6 Continual Learning Evaluation

Input: Embedded few shot train dataset E_{train}, Embedded few shot test dataset E_{test}, Evaluation threshold $threshold$, Number of training epochs e_{train}

function CONTINUALLEARNING($E_{train}, E_{test}, threshold, e_{train}$)
  $L \leftarrow$ GETUNIQUELABELS(E_{train})
  $LabelsAdded = \{\}$
  for $l \in L$ **do**
    $E_l \leftarrow \{(\phi'_i, l) \mid (\phi'_i, y_i) \in E_{train}, l \in y_i\}$
    if ISEMPTY($LabelsAdded$) **then**
      $f =$ INITIALIZEWEIGHTIMPRINTING(E_l)
    if !ISEMPTY($LabelsAdded$) **then**
      $f =$ ADDCLASSWEIGHTIMPRINTING(f, E_l)
    $E'_{train} \leftarrow \{(\phi'_i, \{l_j\}) \mid (\phi'_i, y_i) \in E_{train}, l_j \in LabelsAdded\}$
    TRAIN(E'_{train}, f, e_{train})
    $E'_{test} \leftarrow \{(\phi'_i, \{l_j\}) \mid (\phi'_i, y_i) \in E_{test}, l_j \in LabelsAdded\}$
    for $(\phi'_i, y_i) \in E'_{test}$ **do**
      $labels \leftarrow$ PREDICT($\phi'_i, f, threshold$)
      EVALUATEMETRIC($labels, y_i$)
    APPEND($LabelsAdded, l$)

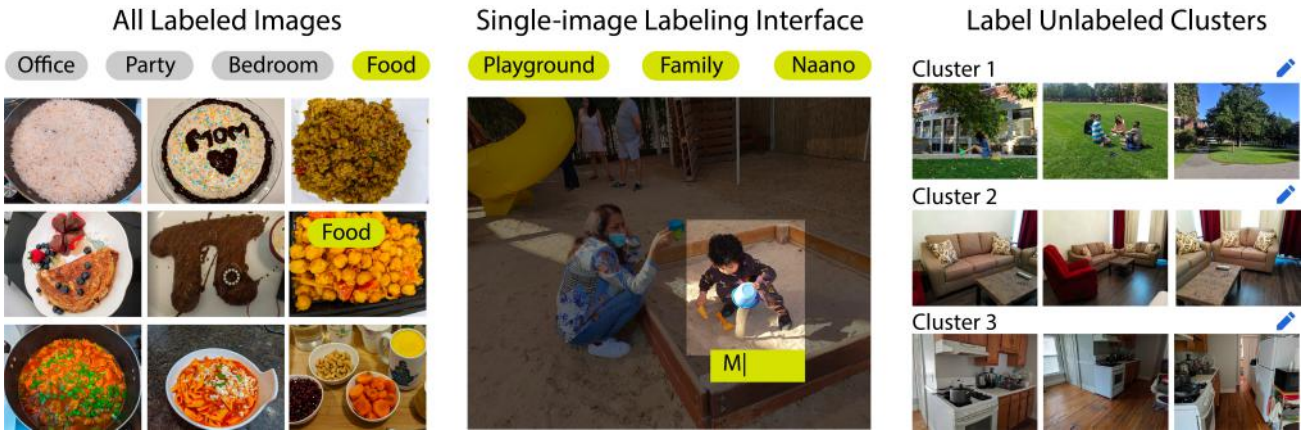

Figure 9. Design for personalized photo labeling application, showing all labeled images, image-labeling interface, and cluster labeling interface

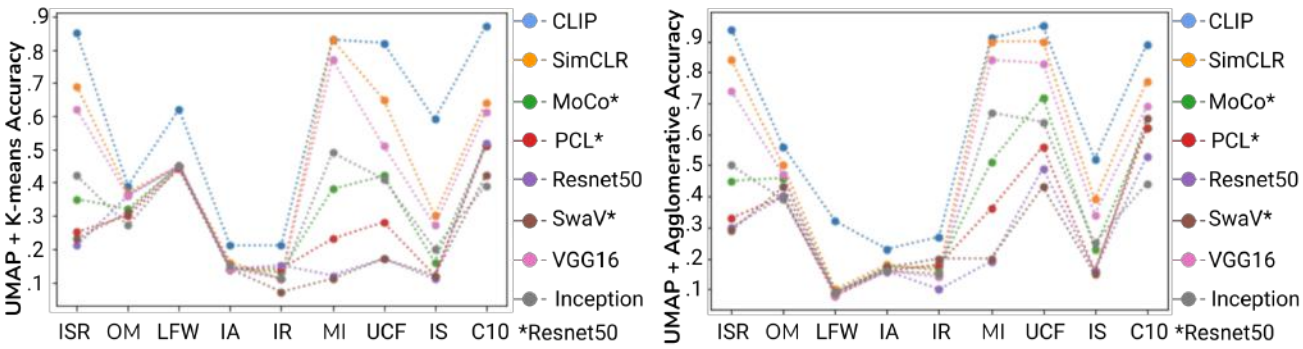

Figure 10. Clustering using different embeddings: Kmeans (Left), Aggolomerate Clustering (Right)

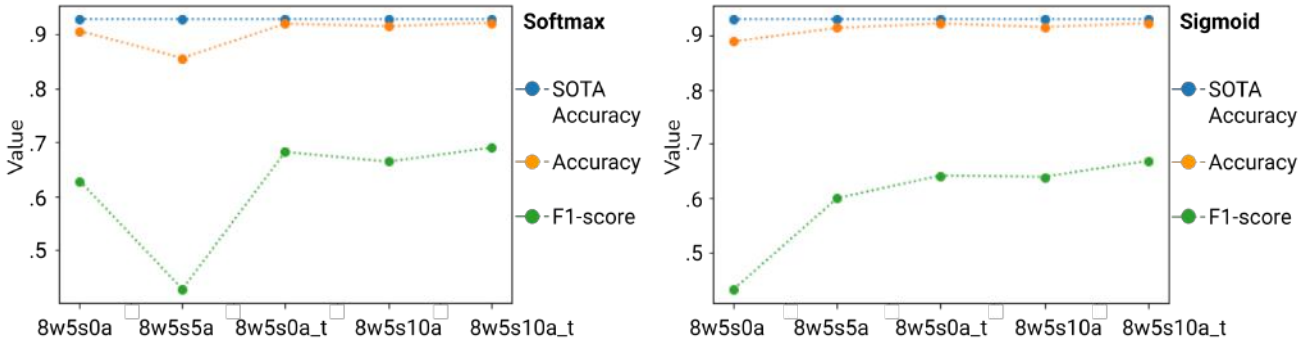

Figure 11. Comparing CLIPPER's FSL for colorectal cancer histology with state-of-the-art

Table 3. Details about our chosen datasets, including their abbreviations (Abbr.).

Dataset Name	Abbr.	SOTA
Single		
Omniglot (Lake et al., 2015)	OM	(Liu et al., 2019)
MiniImagenet (Vinyals et al., 2017)	MI	(Luo et al., 2019)
Labeled Faces in the Wild (Huang et al., 2008)	LFW	(Chrysos et al., 2020)
UCF101 (Soomro et al., 2012)	UCF	(Kalfaoglu et al., 2020)
Imagenet-R (Hendrycks et al., 2020)	IR	(Radford et al., 2021)
Imagenet-Sketch (Wang et al., 2019)	IS	(Wang et al., 2019)
Indoor Scene Recognition (Quattoni & Torralba, 2009)	ISR	(Rahimzadeh et al., 2021)
CIFAR10 (Krizhevsky et al., 2009)	C10	(Foret et al., 2020)
Imagenet-A (Hendrycks et al., 2019)	IA	(Radford et al., 2021)
Colorectal Histology (Kather et al., 2016)	CH	(Ghosh et al., 2021)
Multi-label		
CelebA Attributes (Liu et al., 2018b)	CAA	(Kehrenberg et al., 2020)
UTK Faces (Zhang et al., 2017)	UTK	(Kärkkäinen & Joo, 2019)
Yale Faces (Georghiades et al., 2001)	YF	(Khalili Mobarakeh et al., 2019)
Common Objects in Context (Lin et al., 2014)	COCO	(Ridnik et al., 2021)
iMaterialist (Fashion) (Guo et al., 2019)	IM	(Guo et al., 2019)

Table 4. Details of metrics used for single-label and multi-label classifications.

Metrics	Single-label Softmax	Single-label Sigmoid	Multi-label
Hamming score		✓	✓
Jaccard		✓	✓
Subset accuracy		✓	✓
Mean Average Accuracy (mAP)		✓	✓
Class F1	✓	✓	✓
Overall F1	✓	✓	✓
Class precision	✓	✓	✓
Overall precision	✓	✓	✓
Class recall	✓	✓	✓
Overall recall	✓	✓	✓
Top-1 accuracy	✓	✓	
Top-5 accuracy	✓	✓	
Class accuracy	✓	✓	✓

Table 5. Comparing FSL (F1-score) using Weight Imprinting for different embeddings models

	CLIP	SimCLR	MoCo	PCL	SwAV	Resnet50	VGG16	Inception V3
C10	0.88	0.76	0.63	0.63	0.53	0.66	0.73	0.45
IA	0.74	0.6	0.54	0.51	0.51	0.55	0.55	0.52
IR	0.77	0.61	0.57	0.58	0.52	0.65	0.61	0.52
IS	0.89	0.83	0.6	0.44	0.34	0.3	0.83	0.72
ISR	0.94	0.82	0.44	0.29	0.22	0.19	0.79	0.58
LFW	0.99	0.66	0.61	0.56	0.64	0.73	0.6	0.62
MI	0.94	0.92	0.63	0.44	0.23	0.2	0.91	0.81
OM	0.93	0.9	0.91	0.91	0.89	0.91	0.88	0.88
UCF	0.94	0.9	0.67	0.46	0.29	0.23	0.84	0.72

Table 6. Comparing Multilabel Weight Imprinting F1-scores for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
CLIP Zero Shot	0.92	0.78	0.87	0.83	0.95	0.91	0.96	0.2	0.87
CLIP Linear Probe	0.91	0.75	0.82	0.91	0.95	1.0	0.95	0.96	0.95
WI	0.88	0.74	0.77	0.89	0.94	0.99	0.94	0.93	0.94
MWI	0.7	0.54	0.6	0.72	0.83	0.94	0.78	0.66	0.83
+ T	0.82	0.65	0.72	0.82	0.91	0.98	0.89	0.82	0.89
+ T + A (Non-trivial 5)	0.85	0.72	0.79	0.89	0.95	1.0	0.92	0.88	0.93
+ T + A (Trivial 5)	0.89	0.74	0.79	0.9	0.94	0.99	0.93	0.93	0.93
+ T + A (Non-trivial 10)	0.87	0.73	0.81	0.91	0.95	1.0	0.93	0.92	0.94
+ T + A (Trivial 10)	0.9	0.74	0.8	0.9	0.95	1.0	0.94	0.95	0.94

Table 7. Comparing Multilabel Weight Imprinting, SOTA, and WI for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
SOTA (Acc)	0.91	0.85	0.89	0.6	0.74	1.0	0.92	1.0	0.99
WI (Acc)	0.89	0.76	0.78	0.9	0.94	1.0	0.94	0.93	0.95
WI (F1)	0.88	0.74	0.77	0.89	0.94	0.99	0.94	0.93	0.94
MWI (Acc)	0.89	0.75	0.79	0.89	0.94	1.0	0.94	0.94	0.94
MWI (F1)	0.7	0.54	0.6	0.72	0.83	0.94	0.78	0.66	0.83
+T +A (F1)	0.9	0.74	0.8	0.9	0.94	1.0	0.94	0.95	0.94
+T +A (Top-1 Acc)	0.91	0.76	0.84	0.92	0.96	1.0	0.95	0.97	0.95
+T +A (Class Acc)	0.96	0.9	0.92	0.96	0.98	1.0	0.98	0.98	0.98

Table 8. Comparing Multilabel Weight Imprinting, SOTA, and WI for multi-label datasets

	CAA	COCO˙F	COCO˙P	IM˙F	IM˙P	UTK	YF
SOTA (Acc)	0.82	0.84	0.84	0.72	0.72	0.86	0.85
MWI (Acc)	0.69	0.88	0.77	0.75	0.79	0.74	0.88
MWI (F1)	0.62	0.75	0.44	0.4	0.48	0.6	0.66
+T +A (F1)	0.69	0.73	0.66	0.41	0.56	0.78	0.79
+T +A (Acc)	0.76	0.88	0.88	0.76	0.83	0.86	0.91

Table 9. Comparing metrics, without training and augmentations, for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
SOFT ACC	0.89	0.77	0.78	0.9	0.94	1.0	0.94	0.93	0.95
SOFT F1	0.88	0.74	0.77	0.89	0.94	0.99	0.94	0.93	0.94
SIG F1	0.7	0.54	0.6	0.72	0.83	0.94	0.78	0.66	0.83
SIG TOP1	0.89	0.75	0.79	0.89	0.94	0.99	0.94	0.93	0.94
SIG CACC	0.86	0.83	0.85	0.89	0.93	0.98	0.92	0.8	0.94

Table 10. Comparing metrics, without training and augmentations, for multi-label datasets

	CAA	COCO˙F	COCO˙P	IM˙F	IM˙P	UTK	YF
SIG F1	0.62	0.75	0.44	0.39	0.48	0.6	0.66
SIG CACC	0.69	0.88	0.77	0.75	0.79	0.74	0.88

Table 11. Comparing MWI with training and augmentations for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
MWI	0.7	0.54	0.6	0.72	0.83	0.94	0.78	0.66	0.83
+ T	0.82	0.65	0.72	0.82	0.91	0.98	0.89	0.82	0.89
+ T + A (Non-trivial 5)	0.85	0.72	0.79	0.89	0.95	1.0	0.92	0.88	0.93
+ T + A (Trivial 5)	0.89	0.74	0.79	0.9	0.94	0.99	0.93	0.93	0.93
+ T + A (Non-trivial 10)	0.87	0.73	0.81	0.91	0.95	1.0	0.93	0.92	0.94
+ T + A (Trivial 10)	0.9	0.74	0.8	0.9	0.95	1.0	0.94	0.95	0.94

Table 12. Comparing MWI with training and augmentations for multi-label datasets

	CAA	COCO˙F	COCO˙P	IM˙F	IM˙P	UTK	YF
MWI	0.62	0.75	0.44	0.39	0.48	0.6	0.66
+ T	0.67	0.74	0.54	0.38	0.55	0.7	0.72
+ T + A (Non-trivial 5)	0.7	0.76	0.56	0.41	0.45	0.78	0.77
+ T + A (Trivial 5)	0.69	0.73	0.61	0.41	0.54	0.78	0.78
+ T + A (Non-trivial 10)	0.71	0.74	0.6	0.41	0.46	0.76	0.77
+ T + A (Trivial 10)	0.69	0.73	0.66	0.41	0.56	0.78	0.79

Table 13. Comparing CLIPPER's FSL for colorectal cancer histology with state-of-the-art

		Single-label Evaluation		Multi-label Evaluation	
	SOTA	Accuracy	Overall F1	Accuracy	Overall F1
8w5s0a	0.93	0.90	0.63	0.89	0.43
8w5s5a	0.93	0.85	0.43	0.91	0.6
8w5s5a_trivial	0.93	0.92	0.68	0.92	0.64
8w5s10a	0.93	0.92	0.66	0.91	0.64
8w5s10a_trivial	0.93	0.92	0.69	0.92	0.67

Table 14. Best performance (10 trivial augmentations) on all datasets for 5-way 5-shot with training - mean metric values with error across 100 episodes.

	MWI+T+A F1	MWI+T+A CAc
C10	0.9±0.04	0.96±0.02
IA	0.74±0.09	0.9±0.03
IR	0.8±0.07	0.92±0.03
IS	0.9±0.05	0.96±0.02
ISR	0.94±0.04	0.98±0.01
LFW	0.1±0.01	0.99±0.01
MI	0.94±0.03	0.98±0.01
OM	0.95±0.03	0.98±0.01
UCF	0.94±0.05	0.98±0.02
CAA	0.69 ± 0.07	0.76±0.05
COCO˙F	0.73±0.08	0.89±0.1
IM˙P	0.56±0.1	0.83±0.14
UTK	0.78 ± 0.06	0.86±0.04
YF	0.79±0.14	0.91±0.05

Table 15. Comparing different few-shot settings with MWI+ for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
5 way 1 shot	0.72	0.54	0.58	0.72	0.8	0.96	0.8	0.8	0.83
20 way 1 shot	0.6	0.26	0.27	0.46	0.64	0.8	0.55	0.52	0.61
5 way 5 shot	0.9	0.74	0.8	0.9	0.95	1.0	0.94	0.95	0.94
20 way 5 shot	0.83	0.51	0.58	0.78	0.88	0.98	0.82	0.84	0.84

Table 16. Comparing different few-shot settings with MWI+ for multi-label datasets

	CAA	COCO˙F	IM˙P	UTK	YF
5 way 1 shot	0.6	0.63	0.51	0.59	0.65
20 way 1 shot	0.6	0.40	0.22	0.62	0.5
5 way 5 shot	0.69	0.73	0.56	0.78	0.79
20 way 5 shot	0.7	0.57	0.34	0.79	0.8

Table 17. MWI+ Continual learning results for increasing classes for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
1	0.83	0.83	0.85	0.92	0.95	0.96	0.86	0.84	0.95
2	0.75	0.58	0.65	0.76	0.85	0.88	0.74	0.69	0.87
3	0.73	0.66	0.7	0.8	0.88	0.91	0.82	0.81	0.89
4	0.79	0.65	0.73	0.85	0.92	0.95	0.88	0.85	0.91
5	0.84	0.7	0.8	0.9	0.96	0.99	0.91	0.86	0.94

Table 18. MWI+ Continual learning results for increasing classes for multi-label datasets

	CAA	COCO	IM	UTK	YF
0	0.68	0.89	0.82	0.74	0.86
1	0.66	0.68	0.49	0.67	0.72
2	0.72	0.76	0.54	0.73	0.72
3	0.72	0.81	0.58	0.76	0.75
4	0.73	0.8	0.53	0.8	0.76

Table 19. Optimal thresholds without continual learning for single-label datasets

	C10	IA	IR	IS	ISR	LFW	MI	OM	UCF
0	0.7	0.66	0.67	0.68	0.68	0.69	0.69	0.72	0.69
5	0.68	0.65	0.65	0.66	0.66	0.67	0.67	0.7	0.66
10	0.63	0.62	0.62	0.63	0.62	0.62	0.62	0.63	0.62
15	0.58	0.59	0.59	0.59	0.59	0.58	0.58	0.57	0.58
20	0.54	0.57	0.57	0.56	0.56	0.55	0.55	0.53	0.55
25	0.51	0.55	0.55	0.54	0.54	0.53	0.53	0.5	0.53
30	0.5	0.54	0.54	0.53	0.52	0.51	0.51	0.49	0.51
35	0.49	0.53	0.53	0.52	0.51	0.5	0.5	0.48	0.5
40	0.48	0.52	0.52	0.51	0.5	0.49	0.49	0.47	0.49
45	0.47	0.51	0.51	0.5	0.5	0.49	0.49	0.46	0.49
50	0.47	0.51	0.51	0.5	0.49	0.49	0.48	0.46	0.48
55	0.47	0.51	0.5	0.49	0.49	0.48	0.48	0.45	0.48
60	0.46	0.5	0.5	0.49	0.49	0.48	0.48	0.45	0.48
65	0.46	0.5	0.5	0.49	0.49	0.48	0.48	0.45	0.48
70	0.46	0.5	0.49	0.49	0.49	0.48	0.48	0.45	0.48
75	0.46	0.5	0.49	0.49	0.48	0.48	0.47	0.45	0.47
80	0.46	0.49	0.49	0.48	0.48	0.48	0.47	0.45	0.47

Table 20. Optimal thresholds without continual learning for multi-label datasets

	CAA	COCO˙F	COCO˙P	IM˙F	IM˙P	UTK	YF
0	0.67	0.7	0.7	0.68	0.68	0.68	0.71
5	0.65	0.68	0.68	0.66	0.66	0.67	0.68
10	0.62	0.63	0.63	0.63	0.63	0.63	0.63
15	0.59	0.58	0.58	0.59	0.59	0.59	0.57
20	0.56	0.54	0.54	0.56	0.56	0.55	0.54
25	0.53	0.51	0.51	0.54	0.54	0.53	0.51
30	0.52	0.5	0.5	0.53	0.53	0.51	0.5
35	0.51	0.49	0.49	0.52	0.52	0.5	0.48
40	0.5	0.48	0.48	0.51	0.51	0.49	0.48
45	0.49	0.47	0.47	0.5	0.5	0.49	0.47
50	0.49	0.47	0.47	0.5	0.5	0.48	0.47
55	0.49	0.47	0.47	0.49	0.49	0.48	0.47
60	0.48	0.46	0.46	0.49	0.49	0.48	0.47
65	0.48	0.46	0.46	0.49	0.49	0.48	0.47
70	0.48	0.46	0.46	0.49	0.49	0.48	0.46
75	0.48	0.46	0.46	0.49	0.49	0.48	0.47
80	0.48	0.46	0.46	0.48	0.48	0.47	0.46