[Learning to Discover Novel Visual Categories via Deep Transfer Clustering]

주제 : 새로운 카테고리 발견 및 분류

Contribution :
1. transfer learning을 통한 clustering. : DEC(Deep Embedded Clustering)을 clustering이 known class에 의해 guide되도록 수정함. + representational bottleneck, temporal ensembling, consistency.
2. unlabeled data에서 클래스 수 예측. : known class의 일부를 probe set로 사용하여 이들을 unlabeled 인것처럼 unlabeled set에 추가하고 이 extended unlabeled set에서 clustering함. probe set의 clustering accuracy와 unlabeled set의 cluster quality index에 기반하여 선택할 클래스 수를 교차 검증하여 unknown class 개수를 신뢰성 있게 추정함.

세팅 :
fully unsupervised setting 은 아님.
machine이 특정 카테고리에 대해서는 이미 알고 있는 상황에서, 새로운 카테고리에 해당하는 이미지가 주어졌을 때 machine 이 새로운 카테고리가 몇개인지, 그리고 그들 각각을 분류. (Figure1)

실생활 활용 예시 :
'*예를 들어, 시장 조사를 위해 슈퍼마켓에서 제품을 인식하는 문제를 고려해보면, 매주 수백 개의 새로운 제품이 도입되며 모든 제품에 수동으로 주석을 다는 것은 엄청나게 비용이 많이 듭니다. 그러나 알고리즘은 수천 개의 제품에 대한 지식을 바탕으로 새로운 제품이 데이터 스트림에 들어오자마자 이를 발견할 수 있습니다.*'

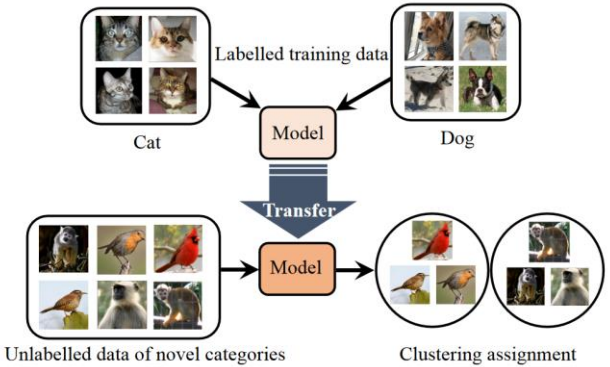

Figure 1. Learning to discover novel visual categories via deep transfer clustering. We first train a model with labelled images (e.g., cat and dog). The model is then applied to images of unlabelled novel categories (e.g., bird and monkey), which transfers the knowledge learned from the labelled images to the unlabelled images. With such transferred knowledge, our model can then simultaneously learn a feature representation and the clustering assignment for the unlabelled images of novel categories.

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

Task:

이미 Labeled set이 있는 상황에서, (L : known class 개수)

$$D^l = \{(x_i^l, y_i^l), i = 1, \ldots, N\} \quad y_i^l \in \{1, \ldots, L\}$$

unlabeled image(:input) 가 들어왔을 때, 클래스(:output) 부여하기. (K : unknown class개수)

$$D^u = \{x_i^u, i = 1, \ldots, M\} \quad y_i^u \in \{1, \ldots, K\}$$

Labeled set과 Unlabeled set은 클래스 종류와 개수가 다 다름.
따라서 labeled set으로부터 특정 클래스에 관해 학습하는 것이 아니라, 좋은 클래스를 만드는 속성이 무엇인지를 학습하고,
이렇게 얻은 지식을 unlabeled set에서 새로운 클래스와 그 수를 발견하는데 활용.

3. Deep transfer clustering
3.1. Transfer clustering and representation learning : deep clustering algorithm의 확장 : transfer knowledge from a known set of classes to a new one.
3.1.1. Joint clustering and representation learning
-Transferring knowledge from known categories
-Bottleneck
3.1.2. Temporal ensembling and consistency
3.2. Estimating the number of classes : unlabeled class 개수(:K) 예측.
-Cluster quality indices

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

3.1. Transfer clustering and representation learning : deep clustering algorithm의 확장 : transfer knowledge from a known set of classes to a new one.

기반이 된 clustering 방법 : DEC(Deep Embedded Clustering) : 좋은 data representation을 학습하는 것과 data clustering을 동시 수행.
(: *It is trained in two phases. The first phase trains an autoencoder using reconstruction loss, and the second phase finetunes the encoder of the autoencoder with an auxiliary target distribution.*)

Representation 추출 : 데이터(: x)에 신경망 f_θ 를 통과시켜서 embedding vector(: z) 추출.

$$z = f_\theta(x) \in \mathbb{R}^d$$

Representation은 labeled data를 사용하여 초기화되고, unlabeled data를 사용하여 fine-tune됨.

우리의 clustering 방법 : 위에서 본 DEC에 3가지 수정을 가함. (Algorithm1)
1. labeled data를 처리할 수 있도록 방법을 확장.
2. 일반화를 개선하기 위해 타이트한 bottleneck을 포함.
3. temporal ensembling과 consistency를 통합.

Algorithm 1 Transfer clustering with known cardinality

Initialization:
Train the feature extractor f_θ on the labelled data D^l. Apply f_θ to the unlabelled data D_u to extract features, use PCA to reduce the latter to K dimensions, and use K-means to initialize the centers U. Incorporate the PCA as a final linear layer in f_θ. Construct target distributions q.
Warm-up training:
for $t \in \{1, \ldots, N_{\text{warm-up}}\}$ **do**
    Train θ and U on D_u using q as target.
end for
Update target distributions q.
Main loop:
for $t \in \{1, \ldots, N_{\text{train}}\}$ **do**
    Train θ and U on D_u using q as target.
    Update target distributions q.
end for
Predict $p(k|i)$ for $i = 1, \ldots, M$ and $k = 1, \ldots, K$.
Return $y_i^u = \text{argmax}_k\, p(k|i)$ for $i = 1, \ldots, M$.

3.1.1. Joint clustering and representation learning

DEC란?
k-means와 비슷하게 cluster들이 vectors 또는 prototypes(: cluster centers : U)로 표현됨.

$$U = \{\mu_k, k = 1, \ldots, K\}$$

다만, k-means와 다른 점은 목표가 cluster를 결정하는 것 뿐만 아니라 data representation인 f_θ 도 학습해야함.

'Representation learning(구별 과제)과 clustering(생성 과제)을 단순히 결합하는 것은 도전적입니다. 예를 들어, k-means 목표 함수를 직접 최소화하면 학습된 표현 벡터가 가장 가까운 클러스터 중심으로 즉시 수렴할 것입니다. DEC [38]은 클러스터 중심과 데이터 표현을 서서히 수렴시키는 방법을 통해 이 문제를 해결합니다.'

$p(k|i)$ = 데이터 포인트 $i \in \{1, \ldots, N\}$ 가 클러스터 $k \in \{1, \ldots, K\}$에 할당될 확률.

$$p(k|i) \propto \left(1 + \frac{\|z_i - \mu_k\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \qquad (1)$$

$p(i,k) = p(k|i)/N$ (데이터 포인트가 균일하게 샘플링된다고 가정하면, $p(i) = 1/N$)

좋은 해에 도달하기 위해, 모델 p의 likelihood를 바로 maximize하지 않고, 대신에 모델 p를 적절한 형태의 분포 q에 맞춤.

구체적으로는, $q(i,k) = q(k|i)/N$ 와 $p(i,k) = p(k|i)/N$ 간의 KL divergence 를 minimize 하는 방식.

$$E(q) = KL(q||p) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q(k|i)\log\frac{q(k|i)}{p(k|i)}$$

[Learning to Discover Novel Visual Categories via Deep Transfer Clustering]

타겟 분포 q를 현재 분포 p의 점진적으로 더 예리한 버전으로 구성.

$$q(k|i) \propto p(k|i) \cdot p(i|k)$$

(데이터 포인트 i가 클러스터 k에 속할 확률 p(k|i)와 클러스터 k에 속한 데이터 포인트가 i일 확률p(i|k)를 결합하여 더 예리한 목표 분포를 구성)

'이 방식으로 이미지 i를 클러스터 k에 할당하는 것은 현재 분포 p가 i에서 k로 가는 높은 확률과 k에서 i로 가는 높은 확률을 할당할 때 강화됩니다. 후자의 경우 클러스터 k가 너무 크지 않은 경우에만 클러스터 k에서 데이터 포인트 i를 샘플링할 확률이 높기 때문에 균형 효과가 있습니다. 따라서 목표 분포는 먼저 p(k|i)를 제곱하여 이를 예리하게 만든 다음, 클러스터별 빈도로 정규화하여 이를 균형 있게 만듭니다.'

$$q(k|i) \propto \frac{p(k|i)^2}{\sum_{i=1}^{N} p(k|i)}$$ (Bayes rule 사용)

정리하면,

$$E(q) = KL(q||p) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|i) \log \frac{q(k|i)}{p(k|i)}. \quad (2)$$

<- 교대 최적화 방식으로 KL divergence를 최소화함. 즉, 타겟 분포 q(k|i)를 고정한 상태에서, representation f_θ 을 SGD 또는 유사한 방법을 사용하여 일정 횟수(보통, 에포크) 동안 이 식을 최소화.

$$q(k|i) \propto \frac{p(k|i)^2}{\sum_{i=1}^{N} p(k|i)}. \quad (3)$$

<- 얘를 사용하여 타겟 분포 q(k|i)를 예리하게 만들고, 이 과정을 반복.

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

-Transferring knowledge from known categories

앞서 본 클러스터링 알고리즘 fully unsupervised.
그러나, 우리의 목표는 known class들을 사용하여 unknown class 발견에 도움 주고싶음.
그러한 정보는 labeled set에서 pre-trained된 image representation f_θ에 있다.

-Bottleneck

알고리즘1 에서 cluster centers U의 초기화.
Unlabeled data로부터 추출한 feature vector에,

$$\mathcal{Z}^u = \{z_i = f_\theta(x_i^u), i = 1, \ldots, M\} \qquad z_i \in \mathbb{R}^d$$

PCA를 통해 차원 축소 후, (unlabeled class 개수 K와 동일한 수의 성분 유지)

차원 축소 layer : $\hat{z}_i = Az_i + b \qquad A \in \mathbb{R}^{K \times d}$
이 linear layer는 신경망의 head에 영구적으로 추가하고, 파라미터 A, b는 클러스터링 중 다른 매개 변수와 함께 fine-tune됨.

k-means 돌려서 초기화.

3.1.2. Temporal ensembling and consistency

DEC의 key idea = 클러스터를 천천히 수렴시켜 데이터의 의미 있는 분할을 학습.
Ours : DEC에 temporal ensembling을 접목시켜서 수렴과정에서의 smoothness를 증가시킴.
방법 : 서로 다른 에포크에서 계산된 클러스터링 모델 p가 이전 분포의 EMA를 통해 합쳐짐.
(앙상블 : 기계 학습에서 여러 개의 모델을 결합하여 더 나은 예측 성능을 얻는 방법)

네트워크의 예측 p를 앙상블 예측 P로 축적시킴.

$$P^t(k|i) = \beta \cdot P^{t-1}(k|i) + (1-\beta) \cdot p^t(k|i), \qquad (4)$$

β : 앙상블이 과거의 학습에 얼마나 멀리 도달할지 제어하는 모멘텀 항
t : 시간단계

EMA의 zero initialization을 수정하기 위해, P^t는 다음과 같이 재조정되어 부드러운 모델 분포를 얻음.

$$\tilde{p}^t(k|i) = \frac{1}{1-\beta^t} \cdot P^t(k|i). \qquad (5)$$

'식 (5)는 새로운 목표 분포 $\tilde{q}^t(k|i)$ 를 얻기 위해 식 (3)에 대입됩니다. 이로 인해 식 (2)의 변형이 정의되며, 이를 최적화하여 모델을 학습합니다.'

$$\tilde{p}^t(k|i) = \frac{1}{1-\beta^t} \cdot P^t(k|i). \qquad (5)$$

$$q(k|i) \propto \frac{p(k|i)^2}{\sum_{i=1}^{N} p(k|i)}. \qquad (3)$$

$$E(q) = KL(q||p) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|i) \log \frac{q(k|i)}{p(k|i)}. \qquad (2)$$

cf)
Deep Embedded Clustering (DEC)의 핵심 아이디어는 클러스터를 천천히 수렴시켜 데이터의 의미 있는 분할을 학습하는 것입니다. 이를 천천히 수렴시키려는 이유는 다음과 같습니다:

1. **안정적이고 점진적인 클러스터링**: 클러스터를 천천히 수렴시키면 모델이 한 번에 급격하게 변화하지 않고 점진적으로 변하게 됩니다. 이는 클러스터링 결과가 안정적이고 일관되게 되도록 돕습니다.
2. **로컬 최적화 방지**: 급격한 변화는 모델이 로컬 최적화에 빠질 위험을 높입니다. 천천히 수렴시키면 더 많은 탐색이 가능해져 글로벌 최적화로 수렴할 가능성이 높아집니다.
3. **표현 학습과 클러스터링의 균형**: DEC는 데이터의 잠재 표현을 학습하고 이를 바탕으로 클러스터를 형성합니다. 이 과정에서 클러스터를 천천히 수렴시키면, 표현 학습과 클러스터링 사이의 균형을 맞출 수 있습니다. 즉, 모델이 데이터를 잘 표현할 수 있는 잠재 공간을 학습하는 동안 클러스터도 점진적으로 개선됩니다.
4. **노이즈와 이상치 처리**: 천천히 수렴하는 과정에서는 노이즈나 이상치가 클러스터링에 미치는 영향을 줄일 수 있습니다. 급격한 변화는 이러한 요소들에 민감하게 반응할 수 있지만, 점진적인 수렴은 보다 견고한 클러스터링을 가능하게 합니다.
5. **모델의 일반화 성능 향상**: 천천히 수렴시켜 모델이 다양한 데이터 포인트에 대해 충분히 학습할 시간을 가지면, 학습된 모델이 새로운 데이터에 대해 더 잘 일반화할 수 있습니다.

이와 같이, 클러스터를 천천히 수렴시키는 것은 모델이 보다 안정적이고 견고하게 데이터의 구조를 학습하고, 의미 있는 분할을 학습하는 데 중요한 역할을 합니다.

Consistency constraint 추가하면, 식(2)가 식(6)으로 됨.

$$E(q) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q(k|i)\log\frac{q(k|i)}{p(k|i)} + \omega(t)\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\|p(k|i) - p'(k|i)\|^2 \quad (6)$$

$p'(k|i)$: 변형된 샘플의 예측 또는 시간 앙상블 예측 (: $\tilde{p}^t(k|i)$)

$\omega(t)$: ramp-up function : consistency constraint의 weight이 0부터 1까지 점진적으로 증가.

3.2. Estimating the number of classes : unlabeled class 개수(:K) 예측. : Unlabeled data에서 클래스 개수 예측.

Labeled probe classes를 사용하여 unlabeled data 내에 클래스 개수 예측.
Probe classes 는 unlabeled data와 결합되고, 이 결과 세트에 여러번 k-means clustering 함. (할때마다 클래스 개수 다름)
결과 클러스터들은 2가지 품질 지표로 검사함. (이 중 하나는 gt가 있는 probe classes가 잘 식별되었는지를 확인)
이 품질 지수를 최대화하는 클러스터 수를 unlabeled class 개수로 추정.

1. L 개의 known classes를 probe subset D_r^l(L_r classes) 와 training subset $D^l \setminus D_r^l$($L - L_r$ classes) 로 split함.

$L - L_r$ classes : supervised feature representation learning 에 사용.

L_r probe classes : unlabeled data와 결합하여 클래스 개수 추정에 사용.

2. L_r probe classes를 D_{ra}^l(L_r^a classes) 와 D_{rv}^l(L_r^v classes) 로 split. $(\text{e.g., } L_r^a : L_r^v = 4 : 1)$

(anchor probe set, validation probe set으로 split)

D^u 의 클래스 개수 추정 위해, $D_r^l \cup D^u$ 에 constrained (semi-supervised) k-means clustering 수행.

구체적으로, k-means 동안 anchor probe set D_{ra}^l에 있는 이미지들은 gt label을 따라 cluster 되도록 하고, validation probe set D_{rv}^l 에 있는 이미지들은 추가적인 unlabeled data로 간주함.

$D_r^l \cup D^u$ 에서의 전체 카테고리 개수 C를 변화시키며 이 constrained k-means를 여러번 실행하고, $D_r^l \cup D^u$ 에 constrained clustering quality를 측정. 각 C 값에 대해 두 가지 품질 지수를 고려함.

1. Labelled validation probe set L_r^v에서 cluster quality 측정.

2. Unlabeled data D^u 에서 cluster quality 측정.

각 인덱스는 최적의 클래스 개수 결정하는 데 사용되고, 결과들은 평균 시킴.

마지막으로, 이 값을 클래스 수로 하여 k-means를 한 번 더 실행하고, D^u 에서 outlier 클러스터(가장 큰 클러스터 질량의 일정 비율(예: τ=1%)보다 작은 질량을 가진 클러스터) 와 클래스들은 제거함. (Algorithm 2)

-Cluster quality indices

1. ACC (average clustering accuracy)
Validation probe set D_{rv}^l 에 있는 labeled classes L_r^v 에 적용.

$$\max_{g \in \mathrm{Sym}(L_r^v)} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left\{\bar{y}_i = g\left(y_i\right)\right\}, \qquad (7)$$

$\bar{y}_i$ and y_i : 각 데이터 포인트 $x_i \in D_{rv}^l$ 의 gt label과 clustering assignment.

$\mathrm{Sym}(L_r^v)$: L_r^v 요소들의 순열 조합 (클러스터링 알고리즘이 임의의 순서로 클러스터 만드니까)

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

2. CVI (cluster validity index)
'클러스터 내 응집'과 '클러스터 간 분리' 개념을 포착하여 unlabeled data D^u에 적용
'CVI metrics : Silhouette [26], Dunn [13], DaviesBouldin [10], and CalinskiHarabasz [5]'

Silhouette index.

$$\sum_{x \in \mathcal{D}^u} \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}, \qquad (8)$$

x : 데이터 샘플
a(x) : x 와 동일 클러스터 내 다른 데이터 샘플들 간 평균 거리
b(x) : x 와 다른 클러스터 내의 모든 데이터 샘플들 간 가장 작은 평균 거리

Algorithm 2 Estimating the number of classes

Preparation:
Split the probe set D_r^l into D_{ra}^l and D_{rv}^l.
Extract features of D_r^l and D^u using f_θ.
Main loop:
for $0 \leq K \leq K_{\max}$ **do**
    Run k-means on $D_r^l \cup D^u$ assuming $L_r + K$ classes in semi-supervised mode (i.e. forcing data in D_{ra}^l to map to the ground-truth class labels).
    Compute ACC for D_{rv}^l and CVI for D^u.
end for
Obtain optimal:
Let K_a^* be the value of K that maximise ACC for D_{rv}^l and K_v^* be the value that maximise CVI for D^u and let $\hat{K} = (K_a^* + K_v^*)/2$. Run semi-supervised K-means on $D_r^l \cup D^u$ again assuming $L_r + \hat{K}$ classes.
Remove outliers:
Look at the resulting clusters in D^u and drop any that has a mass less than τ of the largest cluster. Output the number of remaining clusters.

4. Experimental results.
Known scenario : 새로운 클래스 개수 아는 상황. OmniGlot, ImageNet, CIFAR-10, CIFAR-100, SVHN
Unknown scenario : 새로운 클래스 개수 모르는 상황. OmniGlot, ImageNet, CIFAR-100

- OmniGlot.
전체 데이터셋 : 50개 알파벳(1623개 손글씨 문자)(32460개 이미지).

아래와 같이 split됨.
background set : 30개 알파벳(954개 문자)(19080개 이미지) -> labeled data
evaluation set : 20개 알파벳(659개 문자)(13180개 이미지) -> unlabeled data

각 문자가 카테고리를 의미하고, 각 문자마다 20개의 예제 이미지 존재.

background set에서 랜덤선택한 5개 알파벳(169개 문자)(3380개 이미지) -> probe
(남은 25개 알파벳(795개 문자)(15900개 이미지)를 feature extractor 학습시키는데 사용)

- ImageNet
전체 데이터셋 : 1000개 클래스(1000개 예제 이미지)(각 클래스당 1개씩)

아래와 같이 split됨.
882개 클래스(882개 이미지) -> labeled data
남은 118개 클래스(118개 이미지) 중 3번 랜덤선택한 30개 클래스 (90개 이미지) -> unlabeled data

labeled data에서 랜덤선택한 82개 클래스(82개 이미지) -> probe
(남은 800개 클래스(800개 이미지)는 feature extractor 학습시키는데 사용)

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

- CIFAR-10
전체 데이터셋 : 50000개 train 이미지, 10000개 test 이미지, 10개 클래스.
(각 이미지는 32x32 크기)

split 방법 :
50000개 train 이미지를 labeled, unlabeled set으로 split함.
처음 5개 클래스(i.e., airplane, automobile, bird, cat, deer)(25000개 이미지) : labeled set
나중 5개 클래스(i.e., dog, frog, horse, ship, truck)(25000개 이미지) : unlabeled set

- CIFAR-100
CIFAR-10보다 클래스마다 이미지 개수가 10배 작다는 것을 제외하고는 다 동일함.

처음 80개 클래스(40000개 이미지) : labeled set
나중 10개 클래스(5000개 이미지) : unlabeled set
남은 10개 클래스(5000개 이미지) : probe

- SVHN
전체 데이터셋 : 73257개 train 이미지, 26032개 test 이미지.

split 방법 :
73257개 train 이미지를 labeled, unlabeled set으로 split함.
digit 0~4(45349개 이미지) : labeled set
digit 5~9(27908개 이미지) : unlabeled set

Evaluation metric :
-클러스터링 성능 : *'conventionallv used clustering accuracy (ACC) and normalized mutual information (NMI)'* : 0과 1사이 값, 값이 클수록 좋은거.
-new class 개수 예측 오차 : $|K_{gt} - K_{est}|$

K_{gt} : 클래스 개수 gt, K_{est} : 클래스 개수 예측 값

Learning to Discover Novel Visual Categories via Deep Transfer Clustering

네트워크 구조.
'For a fair comparison, we follow [15, 16] and use a 6-layer VGG like architecture [27] for OmniGlot and CIFAR-100, and a ResNet18 [14] for ImageNet and all other datasets.'

Training configurations.
- Omniglot
Omniglot은 카테고리 수는 많고, 카테고리 당 예제 이미지 수는 적어서 few-shot learning에서 많이 사용됨.

전체 데이터셋 : 50개 알파벳(1623개 손글씨 문자)(32460개 이미지).

아래와 같이 split됨.
background set : 30개 알파벳(954개 문자)(19080개 이미지) -> labeled data
evaluation set : 20개 알파벳(659개 문자)(13180개 이미지) -> unlabeled data

각 문자가 카테고리를 의미하고, 각 문자마다 20개의 예제 이미지 존재.

background set에서 랜덤선택한 5개 알파벳(169개 문자)(3380개 이미지) -> probe
(남은 25개 알파벳(795개 문자)(15900개 이미지)를 feature extractor 학습시키는데 사용) <-- prototypical loss 사용.

- ImageNet 및 다른 데이터셋들
cross-entropy loss

4.2. Learning with a known number of categories (Known scenario)

DTC-Baseline: DEC 손실로 훈련된 우리의 기본 모델.
DTC-Π: '샘플의 예측과 그 변형된 대응물의 예측 간의 일관성 제약'을 적용하여 DEC 손실로 훈련된 우리의 모델.
DTC-TE: '각 샘플의 현재 예측과 시간적 앙상블 예측 간의 일관성 제약'을 적용하여 DEC 손실로 훈련된 우리의 모델.
DTC-TEP: '시간적 앙상블 예측으로부터 구성된 target을 사용한' DEC 손실로 훈련된 우리의 모델.

'To measure the performance of metric learning based initialization, we also show the results of k-means [21] on the features of unlabelled data produced by our feature extractor trained with the labelled data.'
'k-means shows reasonably good results on clustering the unlabelled data using the model trained on labelled data, indicating that the model can transfer useful information to cluster data of unlabelled novel categories.'
'All variants of our approach substantially outperform kmeans, showing that our approach can effectively finetune the feature extractor and cluster the data.'

Table 1. Visual category discovery (known number of categories).

	CIFAR-10		CIFAR-100		SVHN		OmniGlot		ImageNet	
Method	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means [21]	65.5%	0.422	66.2%	0.555	42.6%	0.182	77.2%	0.888	71.9%	0.713
DTC-Baseline	74.9%	0.572	72.1%	0.630	57.6%	0.348	87.9%	0.933	**78.3%**	0.790
DTC-II	**87.5%**	**0.735**	70.6%	0.605	**60.9%**	**0.419**	**89.0%**	**0.949**	76.7%	0.767
DTC-TE	82.8%	0.661	**72.8%**	**0.634**	55.8%	0.353	87.8%	0.931	78.2%	**0.791**
DTC-TEP	75.2%	0.591	72.5%	0.632	55.4%	0.329	87.8%	0.932	**78.3%**	**0.791**

[Learning to Discover Novel Visual Categories via Deep Transfer Clustering]

'It can be seen that our learned representation is sufficiently discriminative for different novel classes, clearly demonstrating that our approach can effectively discover novel categories.'

Figure 2. Representation visualization on CIFAR-10. Left: t-SNE projection on our learned features of unlabelled data (colored with GT labels); Middle: failure cases of clustering horses as dogs; Right: failure cases of clustering trucks as ships.

'We compare our approach with traditional methods as well as state-of-the-art learning based methods on OmniGlot and ImageNet in table 2.'
'In addition, we also compare with KCL and MCL on CIFAR-10, CIFAR-100, and SVHN in table 3 based on their officially-released code.'

Table 2. Results on OmniGlot and ImageNet with known number of categories.

	OmniGlot		ImageNet	
Method	ACC	NMI	ACC	NMI
k-means [21]	21.7%	0.353	71.9%	0.713
LPNMF [4]	22.2%	0.372	43.0%	0.526
LSC [8]	23.6%	0.376	73.3%	0.733
KCL [15]	82.4%	0.889	73.8%	0.750
MCL [16]	83.3%	0.897	74.4%	0.762
Centroid Networks [17]	86.6%	-	-	-
DTC	**89.0%**	**0.949**	**78.3%**	**0.791**

Table 3. Comparison with KCL and MCL on CIFAR-10/CIFAR-100/SVHN.

	CIFAR-10		CIFAR-100		SVHN	
	ACC	NMI	ACC	NMI	ACC	NMI
KCL [15]	66.5%	0.438	27.4%	0.151	21.4%	0.001
MCL [16]	64.2%	0.398	32.7%	0.202	38.6%	0.138
DTC	**87.5%**	**0.735**	**72.8%**	**0.634**	**60.9%**	**0.419**

4.3. Finding the number of novel categories (Unknown scenario)

KCL, MCL : 카테고리 개수를 굉장히 크게(i.e., 100) 가정함.
Ours : 알고리즘2를 사용하여 transfer clustering 하기 전에 미리 카테고리 개수 예측함.(우리 실험