Sharif University of Technology
Computer Engineering Department

MSc Thesis

Deep Zero-Shot Learning

Seyed Mohsen Shojaee

supervised by

Dr.Mahdieh Soleymani

Summer 2016

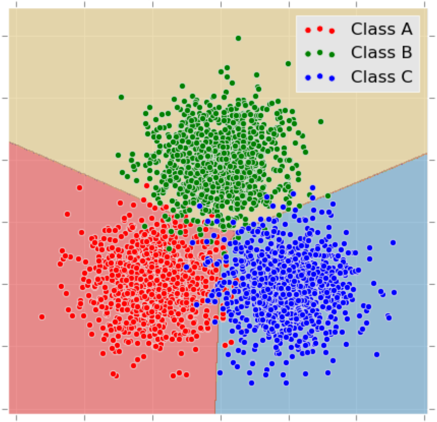
Plan

Plan

Introduction

Standard Supervised Learning Paradigm: Discover the pattern for each class from abundant labeled samples.

- Using SVM, Decision Tree, KNN, etc.



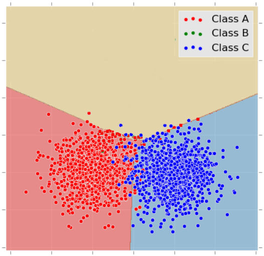
Extening the Standard Paradigm

Sometimes samples from all classes is not available

- Example: Novel Categories, Fine-grained classification.

Zero-Shot Learning addresses the problem of classification.

when no training sample is available for some classes.



Extening the Standard Paradigm

Identifying Classes without Samples:

- Each category is identified some *auxiliary information* also called *signature*.
- Examples of class signatures include:
  - Attribute Vectors
  - Text Articles
  - Category Names
- Signatures exist for all classes.

Extening the Standard Paradigm

As a sample, an animal species like Zebra can have these signatures:

- The Vector (four legs, fast, striped, gallops, non-domestic, ...).
- The Wikipedia Entry for zebra.
- The word *'Zebra'* itself.

Problem Definition

At training time:

- there are N_s labeled samples: $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_s}$.
- These samples are from n_s classes that are called *seen classes*.
- Class signatures C_s for seen classes is also available.
- There are also n_u classes with no labeled sample. These are called *unseen classes.*
- It is assumed in most works that signatures of unseen classes, C_u, is also available.

Problem Definition

At test time:

- N_u samples from unseen classes are presented: $\{(\mathbf{x}_i)\}_{i=N_s+1}^{N_s+N_u}$.
- The Goal is to classify test samples into unseen categories.
- In other words finding

$$\arg\min_{\mathbf{y}^*{}_i} \mathbf{y}^*{}_i \neq \mathbf{y}_i, \quad i = N_s + 1, \ldots, N_s + N_u$$

Solution Steps

Most existing solutions for zero-shot learning consist of these three steps:

1. Embed images in a semantic space

Solution Steps

Most existing solutions for zero-shot learning consist of these three steps:

1. Embed images in a semantic space
2. Embed class signatures to same semantic space

Solution Steps

Most existing solutions for zero-shot learning consist of these three steps:

1. Embed images in a semantic space
2. Embed class signatures to same semantic space
3. Assign images to those classes (e.g. using nearest neighbor classifier)

Solution Steps

Most existing solutions for zero-shot learning consist of these three steps:

1. Embed images in a semantic space
2. Embed class signatures to same semantic space
3. Assign images to those classes (e.g. using nearest neighbor classifier)

Plan

Prior Works

Existing works can be categorized by the semantic space they use:

- Space of signatures (Attribute Prediction).
- Space of images.
- A third space.

We review some selected works from each category.

Attribute Prediction

- A large body of work in zeroshot learning belongs to this category.
- The mapping from signature space is considered identity mapping.
- Attribute Estimator/Classifier are learned on train images (standard supervised problem).
- The Estimator/Classifier is used on test images to find $\mathbf{c}_i^*$ for image $\mathbf{x}_i$
- $\mathbf{x}_i$ is assigned to class with most similar signature:

$$\ell(\mathbf{x}_i) = \underset{j=n_s+1\ldots,n_s+n_u}{\arg\min} \; distance(\mathbf{c}_i^*, \mathbf{c}_j)$$

Examples from this category include:
[Akata et al., 2013, Jayaraman and Grauman, 2014, Lampert et al., 2009]

Mapping to Image Space

- In training time, Learn a mapping from class signatures to image space:

$$\phi : \mathbb{R}^a \rightarrow \mathbb{R}^d$$

- This can bee seen predicting linear one-vs-all classifier for each class from its signature.
- In test time, classify test images using classifiers predicted from unseen class signatures.
- Assign each sample to class whose classifier produces maximum score:

$$\ell(\mathbf{x}) = \operatorname*{arg\,max}_{j=n_s+1\ldots,n_s+n_u} \langle \phi(\mathbf{c}_j), \mathbf{x} \rangle \tag{1}$$

Examples from this category include:
[Elhoseiny et al., 2015, Reed et al., 2016]

Mapping to a middle Space

- In training time mappings $\theta(\mathbf{c})$ and $\phi(\mathbf{x})$ are learned.
- The mapping should map image $\mathbf{x}$ close to its true signature $\mathbf{c}$
- And with a margin from other signatures.
- In this way we expect θ and ϕ to map signature and samples of same unseen classes also close to each other.
- Space of seen classes has been a successful choice for middle space

Mapping to a middle Space

- In training time mappings $\theta(\mathbf{c})$ and $\phi(\mathbf{x})$ are learned.
- The mapping should map image $\mathbf{x}$ close to its true signature $\mathbf{c}$
- And with a margin from other signatures.
- In this way we expect θ and ϕ to map signature and samples of same unseen classes also close to each other.
- Space of seen classes has been a successful choice for middle space
- Bilinear mappings fall into this category too.

Mapping to a middle Space

- In training time mappings $\theta(\mathbf{c})$ and $\phi(\mathbf{x})$ are learned.
- The mapping should map image $\mathbf{x}$ close to its true signature $\mathbf{c}$
- And with a margin from other signatures.
- In this way we expect θ and ϕ to map signature and samples of same unseen classes also close to each other.
- Space of seen classes has been a successful choice for middle space
- Bilinear mappings fall into this category too.

Examples from this category include: [Ba et al., 2015, Reed et al., 2016]

Mapping to a middle Space

- In training time mappings $\theta(\mathbf{c})$ and $\phi(\mathbf{x})$ are learned.
- The mapping should map image $\mathbf{x}$ close to its true signature $\mathbf{c}$
- And with a margin from other signatures.
- In this way we expect θ and ϕ to map signature and samples of same unseen classes also close to each other.
- Space of seen classes has been a successful choice for middle space
- Bilinear mappings fall into this category too.

Examples from this category include: [Ba et al., 2015, Reed et al., 2016]

Semi-supervised Zero-shot Learning

- Use Unsupervised information in structure of unlabeled images.
- This information helps finding better mappings
- Semi-supervised methods can alleviate *domain shift problem* by using unlabeled samples.

Semi-supervised Zero-shot Learning

- Use Unsupervised information in structure of unlabeled images.
- This information helps finding better mappings
- Semi-supervised methods can alleviate *domain shift problem* by using unlabeled samples.

Examples include: [Schuurmans and Tg, 2015, Li and Guo, 2015, Kodirov et al., 2015, Fu et al., 2014]

Semi-supervised Zero-shot Learning

- Use Unsupervised information in structure of unlabeled images.
- This information helps finding better mappings
- Semi-supervised methods can alleviate *domain shift problem* by using unlabeled samples.

Examples include: [Schuurmans and Tg, 2015, Li and Guo, 2015, Kodirov et al., 2015, Fu et al., 2014]

Domain shift problem

- Attributes are represented with different visual features in different classes.
- Mapping Learned on seen classes would not do as good on unseen classes.

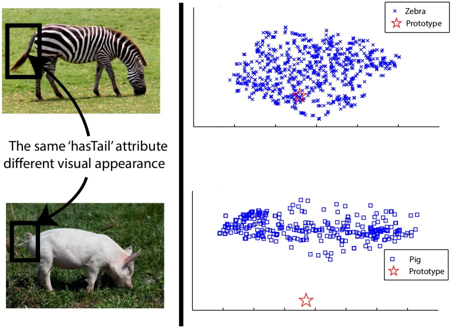


Figure: Different visual representation of attribute "has tail" [Fu et al., 2014].

Plan

Proposed Methods

Here we present four proposed methods for the problem of zero-shot Image Classification.

In our methods we consider class signatures of type attribute vectors.

- Attribute Prediction with Multi-task Deep Neural Networks.
- Mapping to Histograms of Seen Classes with Deep Neural Network.
- Independent Embedding and Clustering
- Joint Embedding and Clustering

Multi-task Neural Network

We propose a network architecture for attribute prediction from images. The network:

- predicts for train and test images at the same time (hence multi-task).
- can mitigate the domain shift problem that appears when only samples from seen classes is used.
- uses 17 layers from famous VGG-19 network [Simonyan and Zisserman, 2014] for feature extraction.
- is trained fast using Stochastic gradient descent algorithms family.

Multi-task Neural Network

Let f denote the mapping modeled by the multi-task network.

Then $\hat{\mathbf{c}}_i = f(\mathbf{x}_i)$ would be attributes predicted by network for $\mathbf{x}_i$

We learn f such that:

$$\underset{f}{\text{minimize}} \frac{1}{N_s} \sum_{i=1}^{N_s} loss(\hat{\mathbf{c}}_i, \mathbf{c}_{y_i}) + \frac{\gamma}{N_u} \sum_{i=N_s}^{N_s+N_u} \Big(\min_{j=n_s,\ldots,n_s+n_u} loss(\hat{\mathbf{c}}_\mathbf{i}, \mathbf{c}_\mathbf{j}) \Big). \quad (2)$$

The second term enforces that prediction for test samples to be close to an unseen class signature

Therefore, mitigating domain-shift problem

Multi-task Neural Network

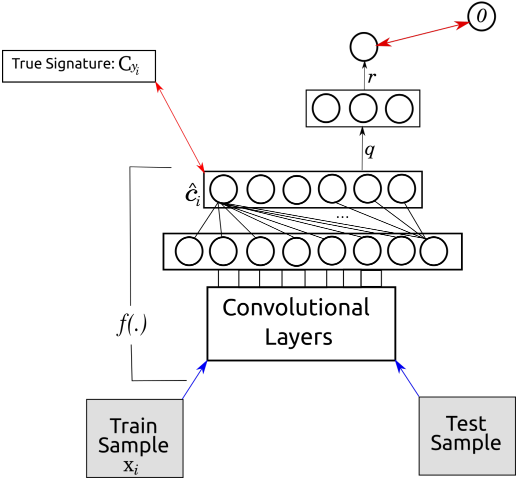


Figure: Proposed Multi-task network Architecture

Multi-task Neural Network

The second term in Eq. (2) is modeled by two layers, q and r:

$$\big(q(\mathbf{v})\big)_j = \|f(\mathbf{v}) - \mathbf{c_j}\|_2^2\,, \tag{3}$$

$$r(\mathbf{z}) = \min_{j=1\ldots n_u} (\mathbf{z})_j. \tag{4}$$

- The $j-$th element of q shows distance of prediction made by network to signature of $j-$th unseen category.
- r selects the minimum element of its input
- Hence using q and r successively produces distance of prediction to nearest unseen class signature.
- This is exactly same as the second term in Eq. (2)

Comparison with other attribute prediction methods

Table: Multi-calss accuracy in form of average $\pm$ std

method	AwA	CUB-2011	aPY	SUNA
[Jayaraman and Grauman, 2014]	43.01 ± 0.07	-	26.02 ± 0.05	56.18 ± 0.27
[Lampert et al., 2009]	41.4	-	19.1	22.2 ± 1.6
[Lampert et al., 2009]	42.2	-	16.9	18.0 ± 1.5
[Akata et al., 2013]	37.4	18.0	-	-
Baseline network (1 layer)	56.78 ± 1.29	32.60 ± 0.82	24.57 ± 1.36	58.33 ± 1.52
Baseline network (2 layer)	52.14 ± 0.31	31.65 ± 0.41	22.56 ± 1.29	62.00 ± 2.64
Multi-task network (1 layer)	$\mathbf{74.52 \pm 1.93}$	$\mathbf{33.91 \pm 0.21}$	$\mathbf{33.10 \pm 1.36}$	$\mathbf{66.13 \pm 0.50}$
Multi-task network (2 layers)	57.10 ± 0.47	31.27 ± 0.87	22.32 ± 0.48	66.83 ± 1.52

Mapping to Histogram of Seen Classes

- Motivated by good performance of methods using historgram of similarity to seen classes as semantic space [Zhang and Saligrama, 2015].
- We present a deep neural network that maps images to this space.
- This network also uses convolutional layers from VGG-19 network.
- The network is a modification of a typical CNN used in standard supervised classification problems.

Mapping to Histogram of Seen Classes

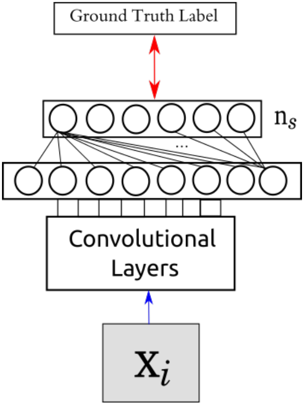


Figure: Proposed network architecture for mapping to histograms

Mapping to Histogram of Seen Classes

In Training Time:

- Labeled samples from seen classes is used.
- Activation function in last layer is softmax:

$$softmax(\mathbf{z})_j = \frac{e^{\mathbf{z}_j}}{\sum_k e^{\mathbf{z}_k}}, \quad j = 1, \ldots, n_s. \tag{5}$$

- Training criteria is correct label prediction of labeled samples.
- Let ϕ denote the mapping modeled by the network

$$\underset{\phi}{\text{minimize}} \sum_{i=1}^{N_s} \sum_{j=1}^{n_s} (\mathbf{y}i)_j \times log(\phi(\mathbf{x}_i)_j) + (1 - (\mathbf{y}i)_j) \times log(1 - \phi(\mathbf{x}_i)_j) \tag{6}$$

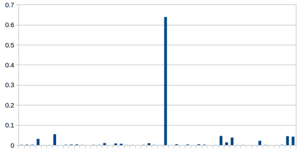
Mapping to Histogram of Seen Classes

In Test Time:

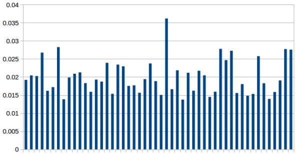
- Activation function in last layer is *temprature softmax*:

$$softmax_T(\mathbf{z})_j = \frac{e^{\mathbf{z}_j}/T}{\sum_k e^{\mathbf{z}_k}/T}, \quad T > 1, \quad j = 1, \ldots, n_s. \tag{7}$$

- The softmax layer is trained to produce distribution of true label which is a discrete delta function.
- When setting $T > 1$ the output becomes smoother.



(a) $T = 1$



(b) $T = 10$

Mapping to Histogram of Seen Classes

- Singanures are mapped to space of histogram by similarity of their signatures to signature of seen classes:

$$\theta_j(\mathbf{c}) = \frac{1}{\|\mathbf{c} - \mathbf{c_j}\|_2}, \quad j = 1, \ldots, n_s. \tag{8}$$

- finally prediction can be done using nearest neighbor (or our proposed compatibility function)

Independent Embedding and Clustering (IEaC)

Observation: There is a clustering structure in image space when features are extracted using Deep CNNs.

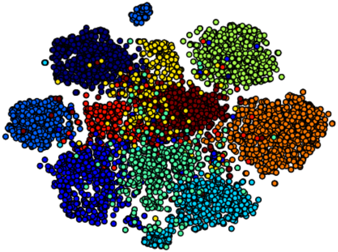


Figure: Samples of unseen classes from AwA dataset. Classes are shown in colors.

Independent Embedding and Clustering (IEaC)

Motivated by this observation we propose a novel compatibility function for zero-shot learning:

1. Cluster test samples.
2. Assign each cluster to an unseen class.
3. All items in a cluster inherit the label received by cluster

We have used two variation for the second step:

- Classify all samples and then use majority vote:
  used on output of two previous methods.
- Classify just cluster centers.
  we will present a method for this type.

Independent Embedding and Clustering (IEaC)

To assing label to cluster centers μ_k we propose:

- Embed class signatures to image space using linear mapping D from:

$$D = \underset{D}{\arg\min} \left\| X_s - DZ_s \right\|_{Fro}^2 + \alpha \left\| D \right\|_{Fro}^2 . \tag{9}$$

  X_s: matrix of train samples, Z_s true attribute vector for samples in X_s.

- Assign each μ_k to an unseen class using:

$$\ell(\boldsymbol{\mu_k}) = \underset{u=1,\dots,n_u}{\arg\min} \left\| \boldsymbol{\mu_k} - D\mathbf{c}_u \right\|_{Fro}^2 \tag{10}$$

Joint Embedding and Clustering (JEaC)

In the previous method:

- Classification Accuracy is bottlenecked by the clustering accuracy.
- Separate learning mapping and mapping prevents information flow.
- Each one is learned with a different criteria

To over come this shortcomings, we propose a *Joint Embedding and Clustering* method.

Joint Embedding and Clustering

The method is formulated as:

$$\min_{R,D} \|X_s - DZ_s\|_{Fro}^2 + \lambda \left\|X_u - DC_uR^T\right\|_{Fro}^2 + \eta \|D\|_{Fro}^2 , \qquad (11)$$
$$s.t. \quad R \in \{0,1\}^{N_u \times n_u}.$$

- The first term is same as in Eq. (9).
- The second term is essentially a clustering criteria. this will be more clear if re-written as:

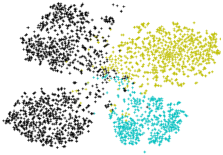
$$\sum_{n=N_s+1}^{N_s+N_u} \sum_{k=1}^{n_u} r_{nk} \|\mathbf{x_n} - D\mathbf{c_k}\|_2^2 .$$

Plan

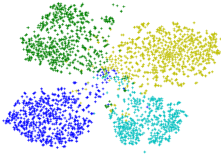
Experimental Results

method	AwA	CUB-2011	aPY	SUN
[Li and Guo, 2015]	38.2 ± 2.3	-	-	18.9 ± 2.5
[Schuurmans and Tg, 2015]	40.05 ± 2.25	-	24.71 ± 3.19	-
[Akata et al., 2015]	66.7	50.1	-	-
[Xian et al., 2016]	71.9	45.5	-	-
[Kodirov et al., 2015]	73.2	39.5	26.5	-
[Akata et al., 2015]	61.9	50.1	-	-
[Zhang and Saligrama, 2015]	76.33 ± 0.53	30.41 ± 0.20	46.23 ± 0.53	82.50 ± 1.32
[Zhang and Saligrama, 2016]	80.46 ± 0.53	42.11 ± 0.55	$\mathbf{50.35 \pm 2.97}$	$\underline{83.83 \pm 0.29}$
proposed (mapping to histograms)	76.50 ± 1.02	33.29 ± 0.21	47.46 ± 0.31	79.88 ± 0.42
proposed(IEaC - k-means)	86.34 ± 0.13	52.48 ± 0.60	48.03 ± 1.56	75.75 ± 1.06
proposed (IEaC - semisupervised)	$\underline{86.38 \pm 0.56}$	$\underline{53.10 \pm 0.43}$	48.52 ± 0.29	80.66 ± 0.76
proposed (JEaC- init D)	83.03	57.55	42.62	72.50
roposed (JEaC - init R)	$\mathbf{88.64 \pm 0.04}$	$\mathbf{58.80 \pm 0.64}$	$\underline{49.77 \pm 2.02}$	$\mathbf{86.16 \pm 0.57}$

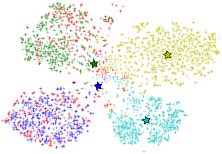
Demonstrating on Real Data



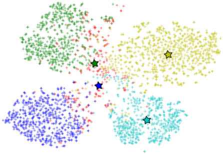
(a) seen/unseen



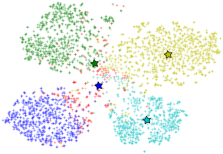
(b) ground truth



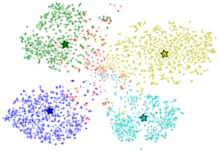
(c) nearest neighbor com-pabtibility



(d) IEaC (k-means)



(e) IEaC (semi-supervised)



(f) JEaC

Conclusion

In this presentation we:

- introduced the problem of zero-shot learning
- categorized and reviewed a selection on prior works
- proposed a Deep Neural network to predict attributes from images.
- proposed a Deep Neural network to map images to histogram of seen classes
- proposed a novel compatibility function and semi-supervised algorithm for zero-shot learning.
- used above propositions in an Independent Embedding and Clustering method
- extended our Independent Embedding and Clustering to do this steps jointly.
- Demonstrated performance of our methods through experiments.
- Discussed effect of different parts in our models by experimenting on a real dataset.

References I

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. (2013).
Label-embedding for attribute-based classification.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 819–826.

Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. (2015).
Evaluation of Output Embeddings for Fine-Grained Image Classification.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ba, J., Swersky, K., Fidler, S., and Salakhutdinov, R. (2015).
Predicting Deep Zero-Shot Convolutional Neural Networks using Textual Descriptions.
IEEE Conference on Computer Vision (ICCV).

Elhoseiny, M., Elgammal, A., and Saleh, B. (2015).
Tell and Predict: Kernel Classifier Prediction for Unseen Visual Classes from Unstructured Text Descriptions.
arXiv preprint arXiv:1506.08529.

References II

Fu, Y., Hospedales, T. M., Xiang, T., Fu, Z., and Gong, S. (2014).
Transductive Multi-view Embedding for Zero-Shot Recognition and Annotation.
In *European Confrence on Computer Vision (ECCV)*, volume 8690, pages 584–599.

Jayaraman, D. and Grauman, K. (2014).
Zero-shot recognition with unreliable attributes.
In *Advances in Neural Information Processing Systems (NIPS) 27*, pages 3464–3472.

Kodirov, E., Xiang, T., Fu, Z., and Gong, S. (2015).
Unsupervised Domain Adaptation for Zero-Shot Learning.
In *IEEE Conference on Computer Vision (ICCV)*, pages 2927–2936.

Lampert, C., Nickisch, H., and Harmeling, S. (2009).
Learning to detect unseen object classes by between-class attribute transfer.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958.

References III

Li, X. and Guo, Y. (2015).
Max-margin zero-shot learning for multi-class classification.
In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 626–634.

Reed, S., Akata, Z., Schiele, B., and Lee, H. (2016).
Learning Deep Representations of Fine-grained Visual Descriptions.
pages 49–58.

Schuurmans, D. and Tg, A. B. (2015).
Semi-Supervised Zero-Shot Classification with Label Representation Learning.
In *IEEE International Conference on Computer Vision (ICCV)*, pages 4211–4219.

Simonyan, K. and Zisserman, A. (2014).
Very deep convolutional networks for large-scale image recognition.
CoRR.

References IV

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. (2016).
Latent Embeddings for Zero-shot Classification.
pages 69–77.

Zhang, Z. and Saligrama, V. (2015).
Zero-Shot Learning via Semantic Similarity Embedding.
In *IEEE Conference on International Computer Vision (ICCV)*, pages 4166–4174.

Zhang, Z. and Saligrama, V. (2016).
Classifying Unseen Instances by Learning Class-Independent Similarity Functions.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Sermi-Supervised Clustering Algorithm

Performance of proposed compatibility function depends on clustering.
We present a semi-supervised clustering algorithm matching assumptions of zero-shot Learning.
Formulated with an Optimization Problem:

$$\min_{R,\mu_{\mathbf{1}},\ldots,\mu_{\boldsymbol{k}}} \sum_{n=1}^{N_s+N_u} \sum_{k} r_{nk} \left\|\mathbf{x_n} - \mu_{\boldsymbol{k}}\right\|_2^2 + \beta \sum_{n=1}^{N_s} \mathbb{1}(\mathbf{r_n} \neq \mathbf{y_n}). \tag{12}$$

The first term is inherited from k-means clustering

The second term produces a penalty of β if a labeled instance from seen classes is assigned to cluster with different number.

Experimental Results for Clustering

method	AwA	CUB-2011	aPY	SUNA
k-means	65.93 ± 1.73	34.48 ± 1.00	65.37 ± 3.73	16.83 ± 0.76
Proposed semi-supervised clustering	70.74 ± 0.32	42.63 ± 0.07	69.93 ± 3.40	45.50 ± 1.32

Plug Proposed Compatibility Function to other methos

Table: Results for Multi-task Neural Network using two different compatibility functions

	AwA	CUB-2011	aPY	SUNA
Nearest Neighbor	74.52 ± 1.93	33.91 ± 0.21	33.10 ± 1.36	66.13 ± 0.50
Proposed Compatibility	74.68 ± 0.73	33.92 ± 0.07	38.26 ± 1.27	67.50 ± 0.00

Discussion

In IEaC:

- Using proposed clustering function based on clustering performs better than nearest neighbor compatibility.
- This is by taking in account the unsupervised information available in images space.

In JEaC:

- Keeps Strong points in IEaC
- Jointly learning cluster assignments and linear mapping improves results.
- Considers good clustering while learning the mapping.
- Considers proximity of cluster centers and mapping of signatures while assigning clusters.
- This mitigates Domain-shift problem