Transfer Learning

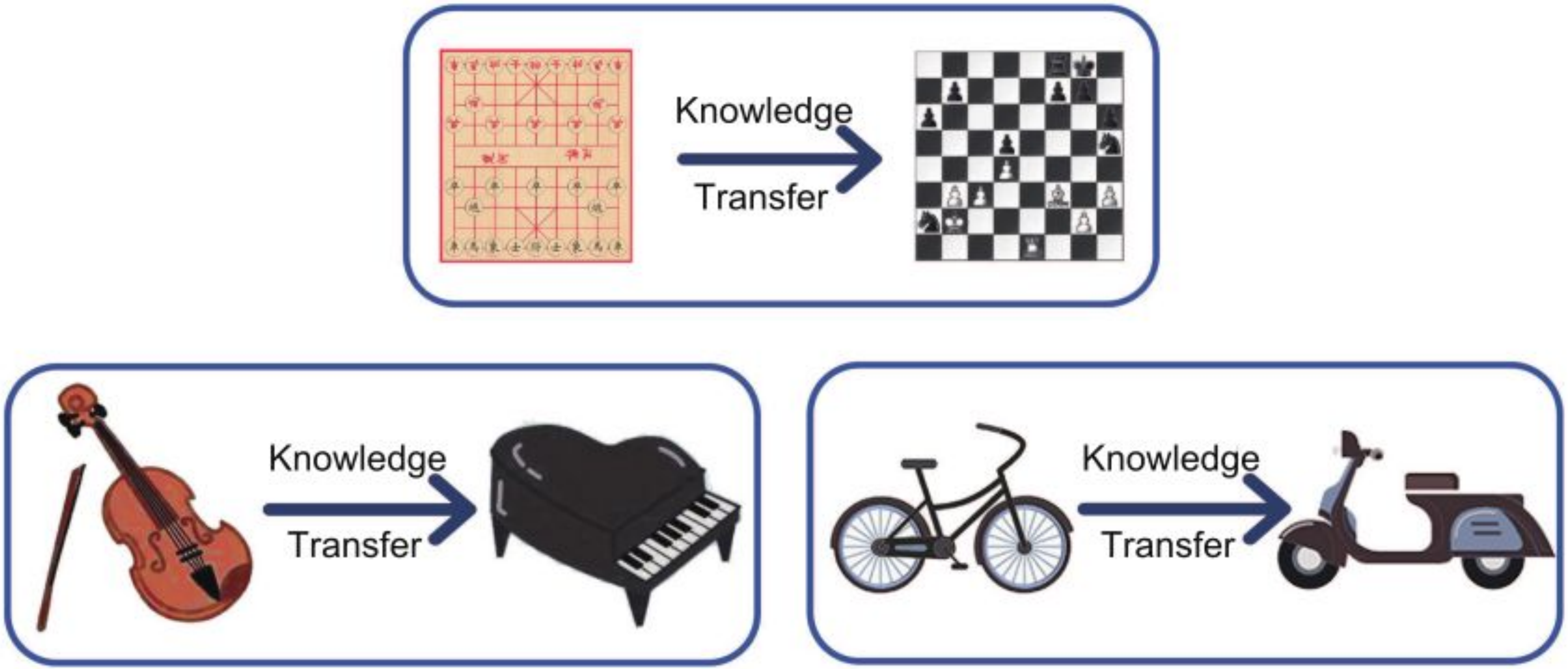


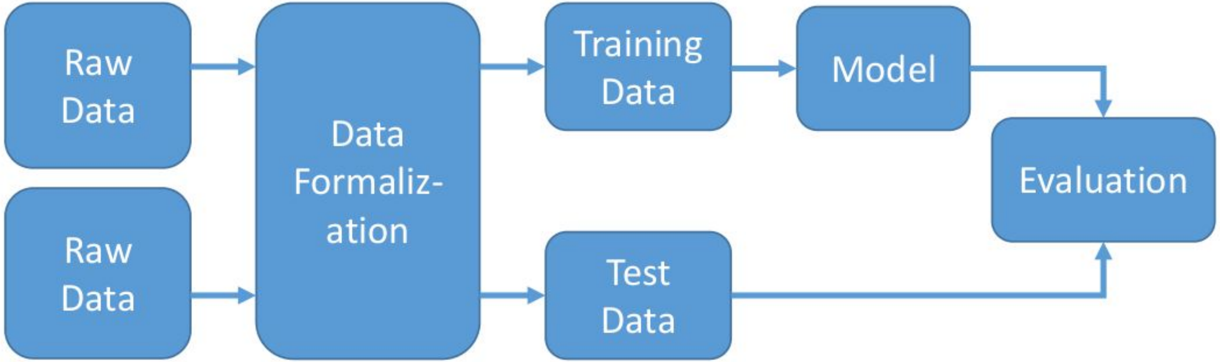
Fig. 1. Intuitive examples about transfer learning.

What is Transfer Learning?

- Definition: A technique in machine learning that uses knowledge from one area to enhance performance in another.
- Process: Involves transferring insights from a source domain to improve outcomes in a target domain.
- Origin: Inspired by educational psychology's theory on the generalization of learning.

Machine Learning Process

$$\min_{\theta} \frac{1}{N} \sum_{(x_i, y_i) \in D_{\text{train}}} \mathcal{L}(y_i, f_\theta(x_i)) + \lambda ||\theta||_2^2$$



$$\text{Test Error} = \frac{1}{N} \sum_{(x_i, y_i) \in D_{\text{test}}} \mathcal{L}(y_i, f_\theta(x_i))$$

- Assumption: training and test data has the same distribution

Strong Assumption

- Consider the learning framework of empirical risk minimization

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y)\sim P_{tst}}[\ell(x, y; \theta)]$$

Strong Assumption

- Consider the learning framework of empirical risk minimization

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y)\sim P_{tst}}[\ell(x,y;\theta)]$$

$$= \arg\min_\theta \mathbb{E}_{(x,y)\sim P_{tst}}\left[\frac{P_{trn}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right]$$

$$= \arg\min_\theta \int_y \int_x P_{tst}(x,y)\left(\frac{P_{trn}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right)dx\,dy$$

$$= \arg\min_\theta \int_y \int_x P_{trn}(x,y)\left(\frac{P_{tst}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right)dx\,dy$$

$$= \arg\min_\theta \mathbb{E}_{(x,y)\sim P_{trn}}\left[\underbrace{\boxed{\frac{P_{tst}(x,y)}{P_{trn}(x,y)}}}_{=1}\ell(x,y;\theta)\right]$$

If $P_{tst}(x,y) = P_{trn}(x,y)$

$$= \arg\min_\theta \mathbb{E}_{(x,y)\sim P_{trn}}[\ell(x,y;\theta)]$$

Notations

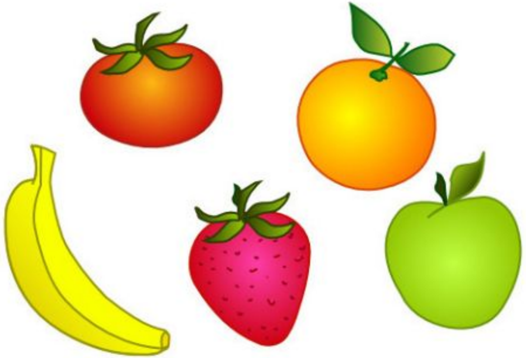
- A **domain** $\mathcal{D} = \{\mathcal{X}, p(x)\}$
  - Feature space $\mathcal{X}$
  - Data distribution $p(x)$
- A **task** $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$
  - Label space $\mathcal{Y}$
  - Objective predictive function $f(\cdot)$

Practical Cases

- Data distributions $p(x)$ change across different domains or vary over time

$$\mathcal{X}_S \neq \mathcal{X}_T \quad \text{or} \quad p_S(x) \neq p_T(x)$$

Real images



Cartoon images

Practical Cases

- Data dependencies $p(y|x)$ could be also different

$$\mathcal{Y}_S \neq \mathcal{Y}_T \quad \text{or} \quad p_S(y|x) \neq p_T(y|x)$$

Apple recognition

Pear recognition

Transfer Learning

- **<u>In practice:</u>** training and test data come from different domains
  - Represented in different feature spaces, OR
  - Follow different data distributions
- $\arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{tst}}[\ell(x,y;\theta)] \neq \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{trn}}[\ell(x,y;\theta)]$

Data Challenge in Deep Learning (TL Motivation)

Disadvantages

- Deep learning models require a large amount of labeled data for effective training.
- Obtaining and preparing sufficient and high-quality data can be a challenging task.
- The performance of machine learning models can be affected by the data challenge.

Transfer Learning

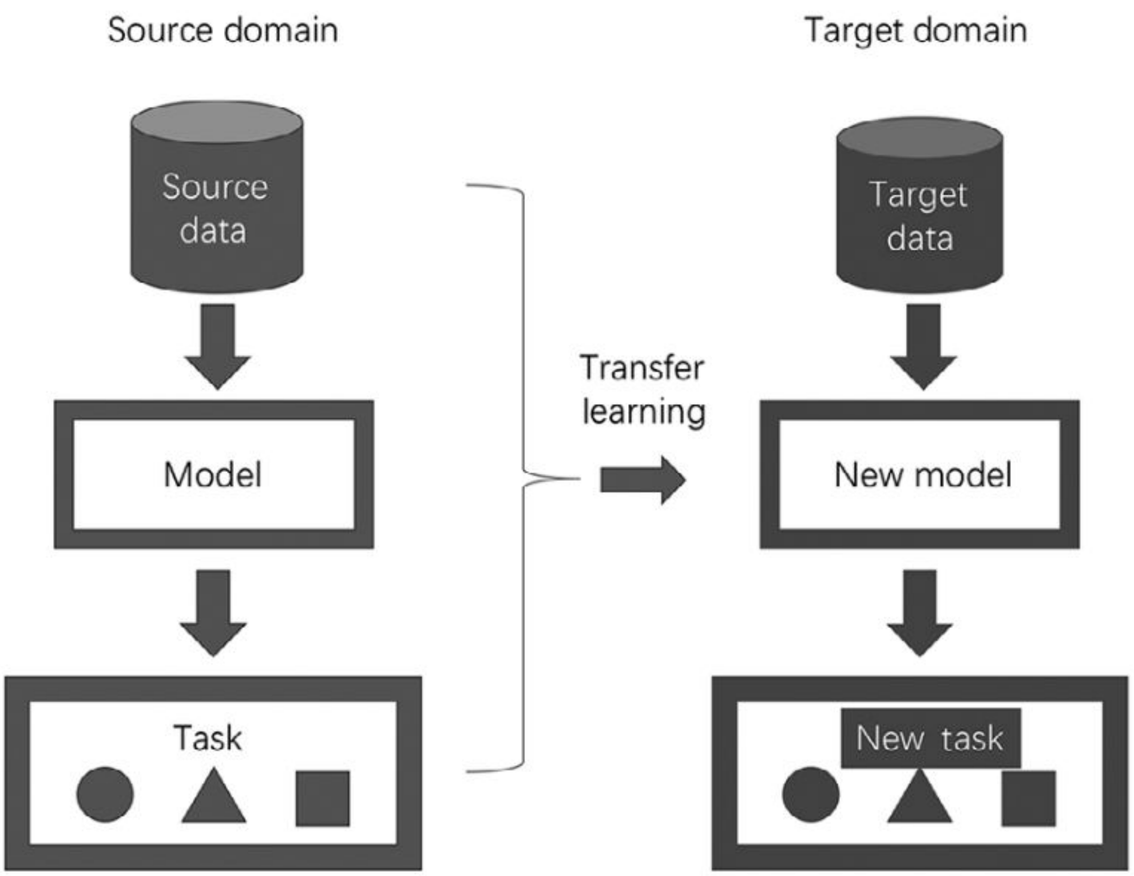
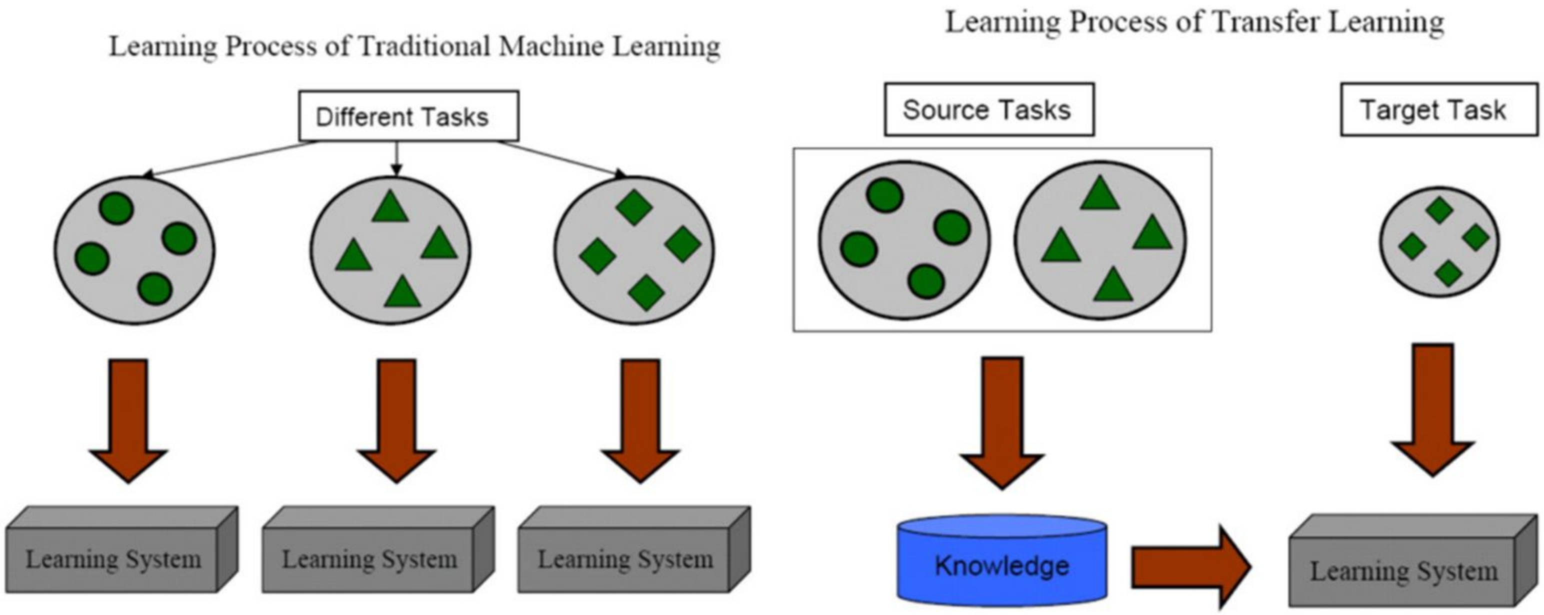


Figure 1.1 An illustration of a transfer learning process

Transfer Learning

- Given a target domain/task, transfer learning aims to
  1) identify the commonality between the target domain/task and previous domains/tasks
  2) transfer knowledge from the previous domains/tasks to the target one such that human supervision on the target domain/task can be dramatically reduced.

Transfer Learning



(a) Traditional Machine Learning

(b) Transfer Learning

Notations

Source domain:

$$\mathcal{P}(X_S), \text{ where } X_S = \{x_{S_1}, x_{S_2}, ..., x_{S_{n_S}}\} \in \mathcal{X}_S$$

Task in the source domain:

$$\mathcal{P}(Y_S|X_S), \text{ where } Y_S = \{y_{S_1}, y_{S_2}, ..., y_{S_{n_S}}\} \text{ and } y_{S_i} \in \mathcal{Y}_S$$

Target domain:

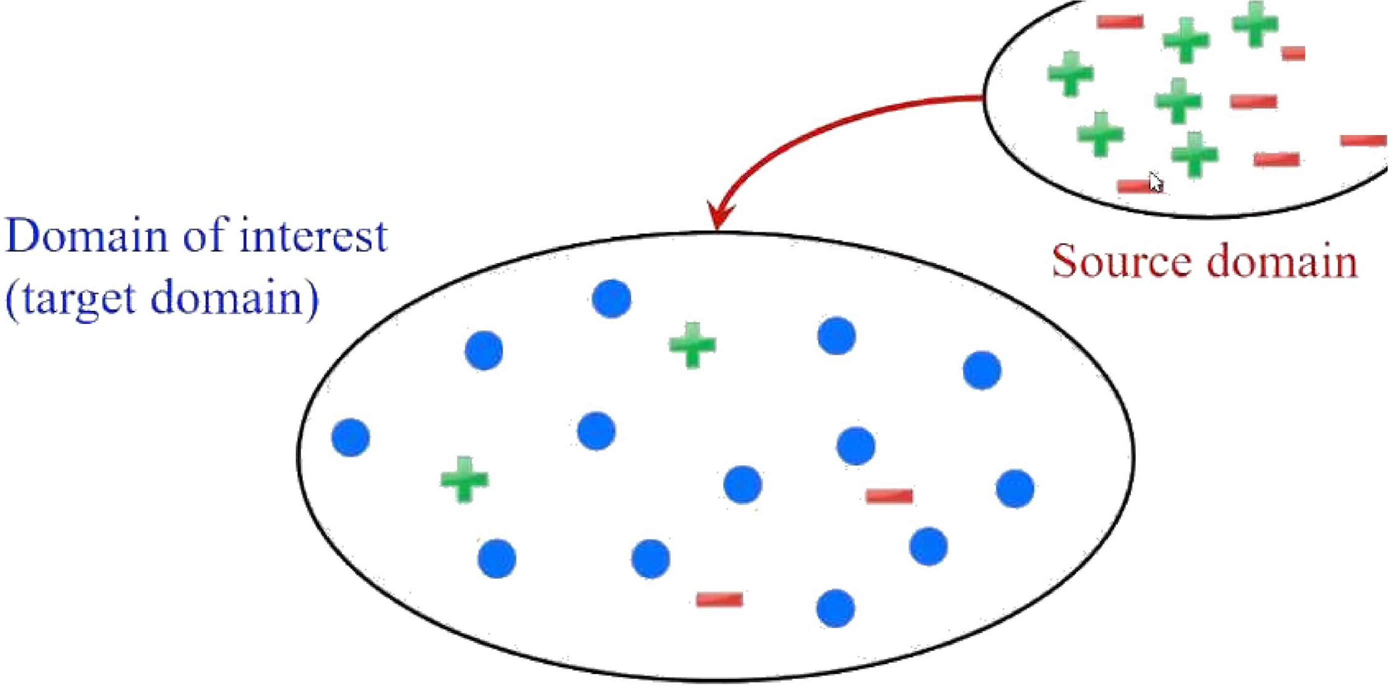
$$\mathcal{P}(X_T), \text{ where } X_T = \{x_{T_1}, x_{T_2}, ..., x_{T_{n_T}}\} \in \mathcal{X}_T$$

Task in the target domain

$$\mathcal{P}(Y_T|X_T), \text{ where } Y_T = \{y_{T_1}, y_{T_2}, ..., y_{T_{n_T}}\} \text{ and } y_{T_i} \in \mathcal{Y}_T$$

Formal Definition

- Given a **source domain** $\mathcal{D}_S$ with corresponding learning task $\mathcal{T}_S$ and a **target domain** $\mathcal{D}_T$ with corresponding learning task $\mathcal{T}_T$
- **transfer learning** is the process of improving the target predictive function $f_T(\cdot)$ by using the related information from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$



Assumption:

1. A little labeled or/and some unlabeled data is available on the target domain
2. Plenty labeled data is available on related source domain(s)
3. Source-domain data can be borrowed to learn a target classifier after some adaptation

Fine-Tuning

- Pre-trained Model: Start with a model that has been pre-trained on a large dataset, such as ImageNet, which has already learned a rich hierarchy of features.
- Initialization: Use the weights of the pre-trained model as the initial weights for the new task rather than random initialization, providing a better starting point.
- Selective Re-training: Decide which layers of the model will be fine-tuned:
- Fine-tune only the top layers while freezing the earlier layers.
- Fine-tune all layers for a more adaptable model, if sufficient data is available.
- Loss Function: Use a loss function appropriate for the new task and compute the loss based on the model's output for the target data.

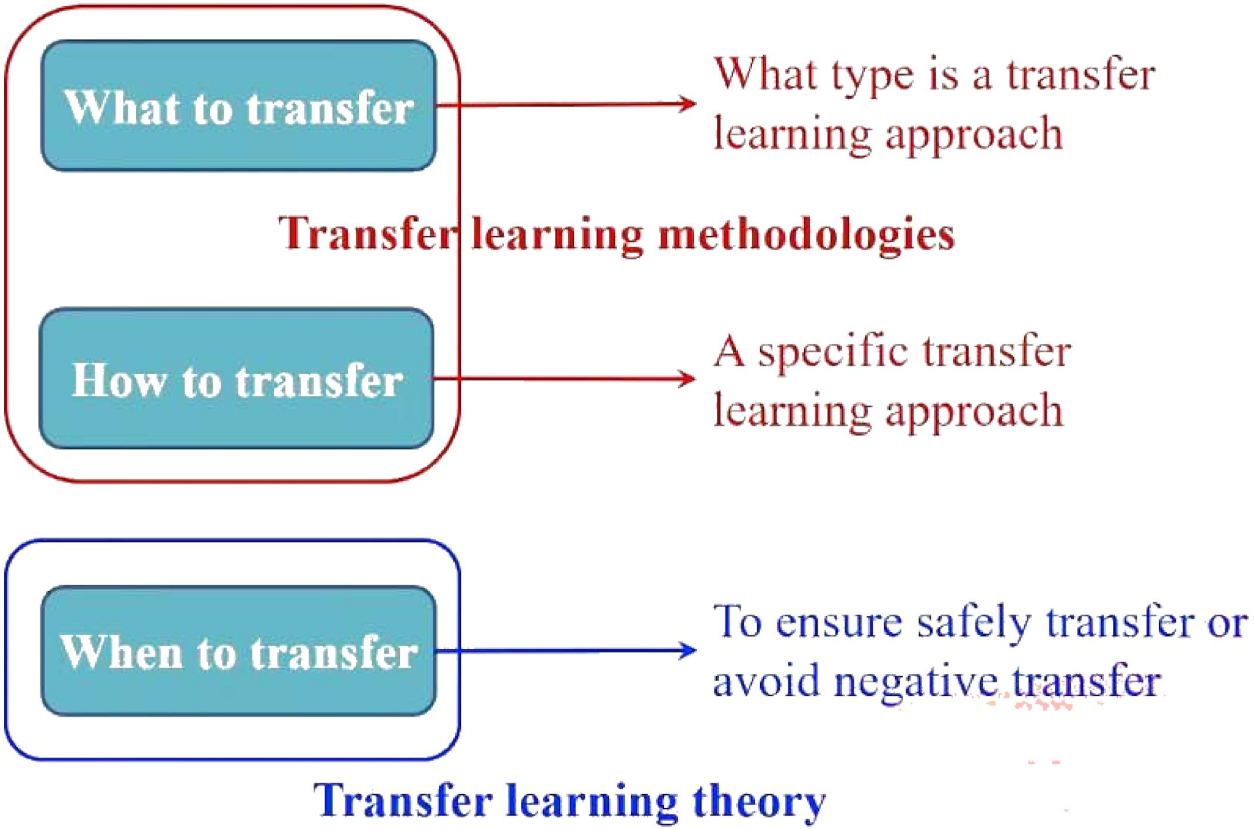
Feature extraction

- Selecting a pre-trained model.
- Identifying and freezing relevant feature layers.
- Passing the target dataset through these layers to extract features.
- Creating a new, smaller model for the specific task using extracted features.
- Training this new model on the target dataset.

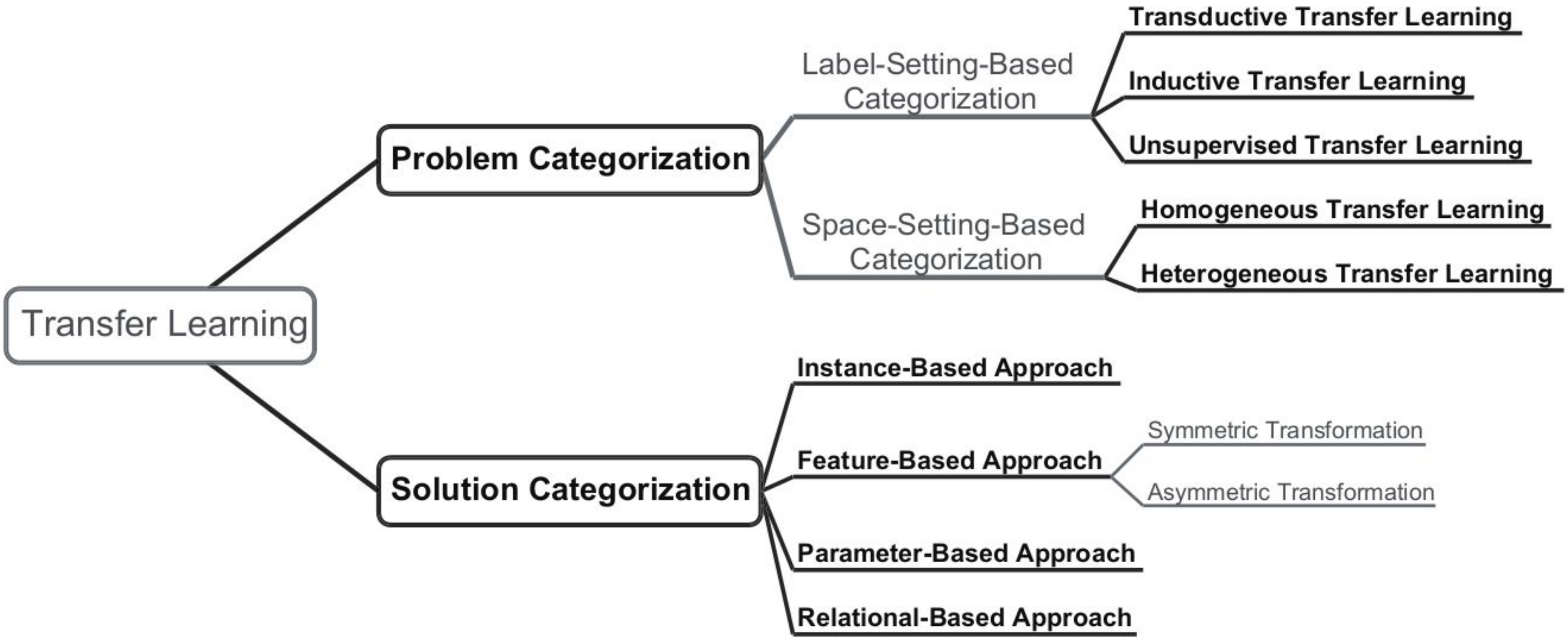
Transfer Learning

- **What to Transfer**: Identifies which part of knowledge (e.g., instances, features, models, relations) can be transferred across domains or tasks.
- **How to Transfer**: Deals with the methodology of transferring knowledge, including instance-based, feature-based, model-based, and relation-based algorithms.
- **When to Transfer**: Determines in which situations transferring knowledge is beneficial and avoids negative transfer, where the transfer might degrade performance.

Transfer Learning



Categorizations of transfer learning



Transfer Learning Problem Setting

Transfer learning settings	Labeled data in a source domain	Labeled data in a target domain	Tasks
Inductive Transfer Learning	×	√	Classification Regression …
	√	√	
Transductive Transfer Learning	√	×	Classification Regression …
Unsupervised Transfer Learning	×	×	Clustering …

Approaches to Transfer Learning

Transfer learning approaches	**Description**
Instance-transfer	*To re-weight some labeled data in a source domain for use in the target domain*
Feature-representation-transfer	Find a "good" feature representation that reduces difference between a source and a target domain or minimizes error of models
Model-transfer	Discover shared parameters or priors of models between a source domain and a target domain
Relational-knowledge-transfer	Build mapping of relational knowledge between a source domain and a target domain.

Approaches to Transfer Learning

	Inductive Transfer Learning	**Transductive Transfer Learning**	**Unsupervised Transfer Learning**
Instance-transfer	√	√	
Feature-representation-transfer	√	√	√
Model-transfer	√		
Relational-knowledge-transfer	√		

Approaches to Transfer Learning

Definition 1.2 (homogeneous transfer learning) Given a source domain $\mathbb{D}_s$ and a learning task $\mathbb{T}_s$, a target domain $\mathbb{D}_t$ and a learning task $\mathbb{T}_t$, *homogeneous transfer learning* aims to help improve the learning of the target predictive function $f_t(\cdot)$ for $\mathbb{D}_t$ using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$, where $\mathscr{X}_s \bigcap \mathscr{X}_t \neq \emptyset$ and $\mathscr{Y}_s = \mathscr{Y}_t$, but $\mathbb{P}^{X_s} \neq \mathbb{P}^{X_t}$ or $\mathbb{P}^{Y_s|X_s} \neq \mathbb{P}^{Y_t|X_t}$.

Definition 1.3 (heterogeneous transfer learning) Given a source domain $\mathbb{D}_s$ and a learning task $\mathbb{T}_s$, a target domain $\mathbb{D}_t$ and a learning task $\mathbb{T}_t$, *heterogeneous transfer learning* aims to help improve the learning of the target predictive function $f_t(\cdot)$ for $\mathbb{D}_t$ using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$, where $\mathscr{X}_s \bigcap \mathscr{X}_t = \emptyset$ or $\mathscr{Y}_s \neq \mathscr{Y}_t$.

Approaches to Transfer Learning

Approach	Homogeneous TL	Heterogeneous TL
Instance-Based	Adapt instances; same features, different distributions.	Map instances; different features.
Feature-Based	Align features and distributions within the same space.	Bridge feature gaps; find common space.
Parameter-Based	Share model parameters; leverage structural knowledge.	Adapt parameters across different features.
Relational-Knowledge	Transfer logic/rules; same features, different tasks.	Align relational structures; different features.

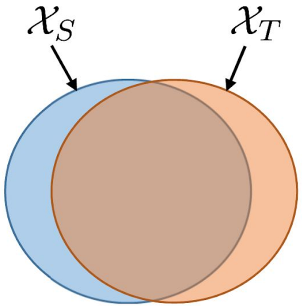
Instance-based Transfer Learning

- General assumption
  - Source and target domains have a lot of overlapping features or even share the same feature spaces

$$\mathcal{X}_S \simeq \mathcal{X}_T$$

  - Label space should be the same

$$\mathcal{Y}_S \simeq \mathcal{Y}_T$$



- Example applications
  - Electronic medical record across different departments
  - Sentiment analysis over different topics

Instance TL Case 1: Domain Adaption

- Problem setting
  - Given source domain labeled data $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$ and target domain data $D_T = \{x_{T_i}\}_{i=1}^{n_T}$
  - learn f_T such that the loss on target data is small

$$\sum_i \mathcal{L}(f_T(x_{T_i}), y_{T_i})$$

  - where y_{T_i} is unknown.
- Assumption
  - The same label space $\mathcal{Y}_S = \mathcal{Y}_T$
  - The same dependency $p(y_S|x_S) = p(y_T|x_T)$
  - (Almost) the same feature space $\mathcal{X}_S \simeq \mathcal{X}_T$
  - Different data distribution $p_S(x) \neq p_T(x)$

Importance Sampling for Domain Adaption

- Importance sampling

$$
\begin{aligned}
\theta^* &= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_T}[\mathcal{L}(y, f_\theta(x))] \\
&= \arg\min_\theta \int_{(x,y)} p_T(x)\mathcal{L}(y, f_\theta(x))dx \\
&= \arg\min_\theta \int_{(x,y)} p_S(x)\frac{p_T(x)}{p_S(x)}\mathcal{L}(y, f_\theta(x))dx \\
&= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_S}\left[\frac{p_T(x)}{p_S(x)}\mathcal{L}(y, f_\theta(x))\right]
\end{aligned}
$$

- Re-weight each instance by $\beta(x) = \dfrac{p_T(x)}{p_S(x)}$

Importance Sampling for Domain Adaption

- How to estimate $\beta(x) = \dfrac{p_T(x)}{p_S(x)}$

- A simple solution would be to first estimate $p_S(x)$ and $p_T(x)$ respectively, and then calculate $\beta(x)$
  - May suffer from huge variance problem

- A more practical solution is to estimate $\dfrac{p_T(x)}{p_S(x)}$ directly

Importance Sampling for Domain Adaption

- How to estimate $\beta(x) = \frac{p_T(x)}{p_S(x)}$

- Build the estimator with a list of basis functions

$$\hat{\beta}(x) = \sum_{l=1}^{b} \alpha_l \psi_l(x)$$

- Minimize squared error

$$\min_{\{\alpha_l\}_{l=1}^{b}} \int_x \left(\hat{\beta}(x) - \beta(x)\right)^2 p_S(x) dx$$

Kanamori et al., A Least-squares Approach to Direct Importance Estimation, JMLR 2009

Instance TL Case 2: Labels in 2 Domains

- Problem setting
  - Given source domain labeled data $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$
  - and very limited target domain data $D_T = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_T}$
  - learn f_T such that the loss on target data is small

$$\sum_i \mathcal{L}(f_T(x_{T_i}), y_{T_i})$$

- Assumption
  - The same label space $\mathcal{Y}_S = \mathcal{Y}_T$
  - Different dependency $p(y_S|x_S) \neq p(y_T|x_T)$
  - (Almost) the same feature space $\mathcal{X}_S \simeq \mathcal{X}_T$
  - Different data distribution $p_S(x) \neq p_T(x)$

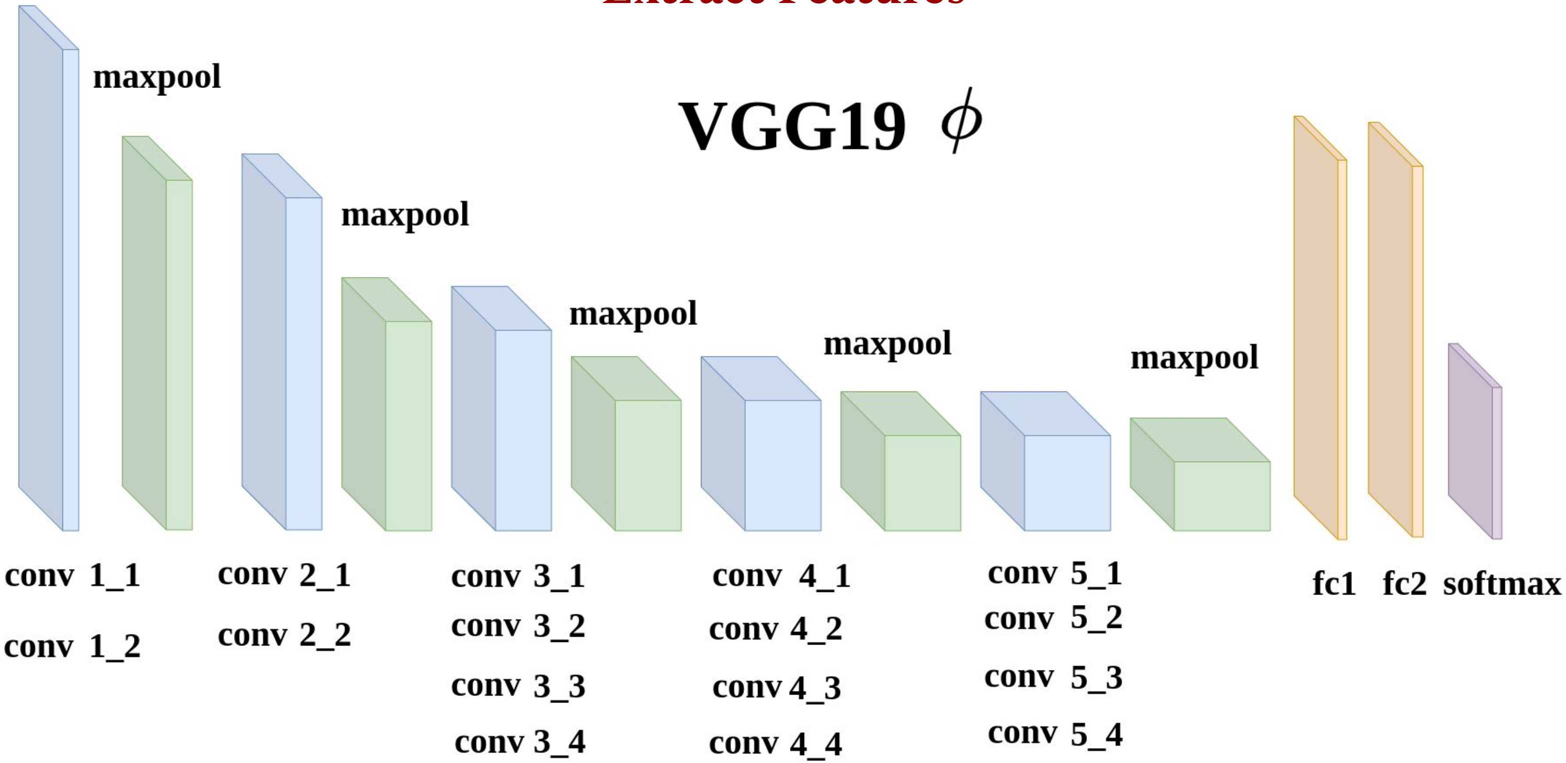
Neural Style Transfer

Base image

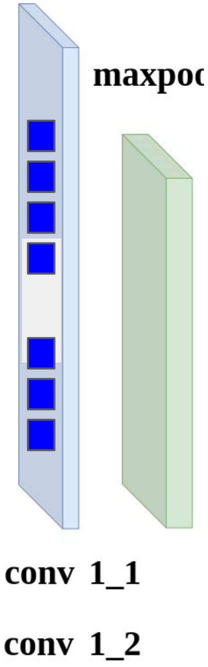
Combined image

Style image

Pre-trained Classifier to Extract Features



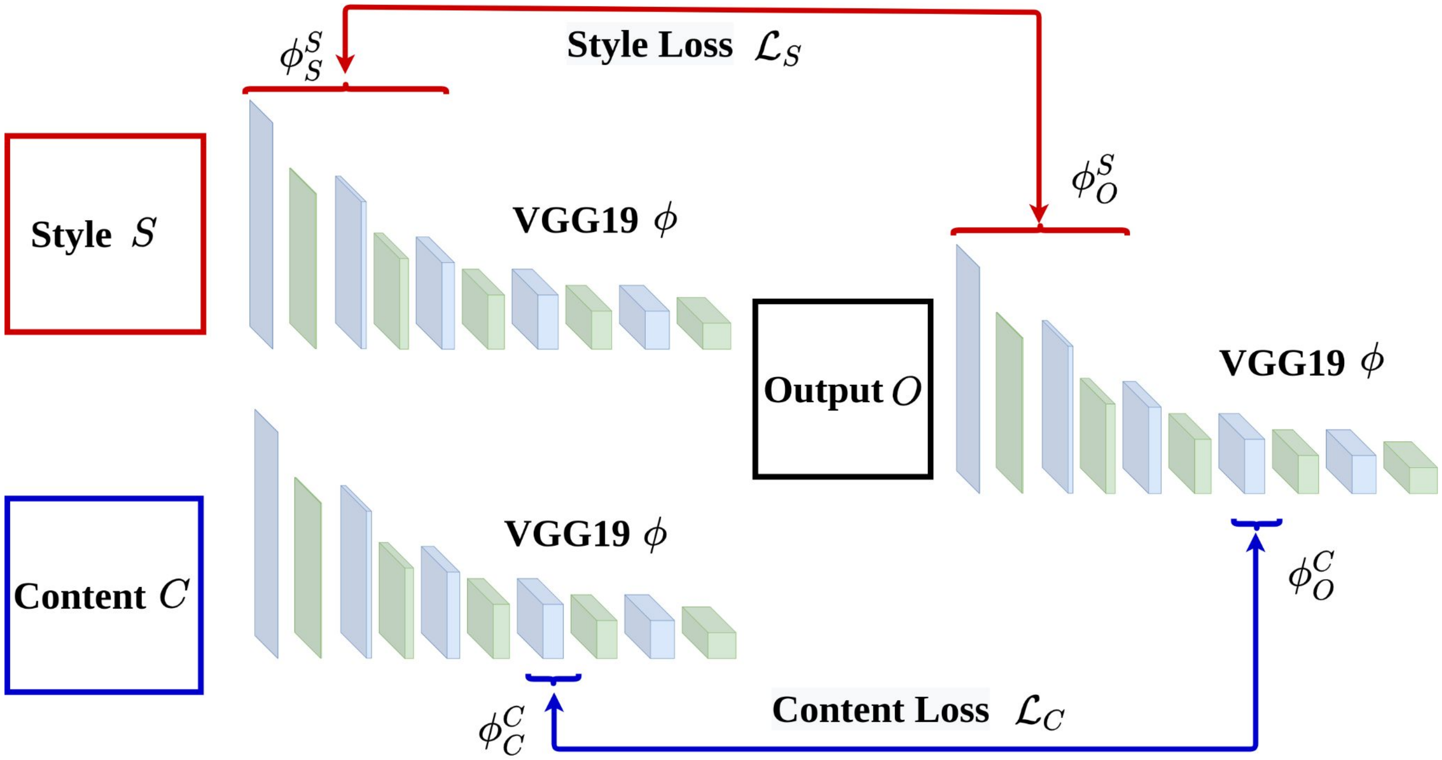
Features Extractor



Each layers contains N_l filters, where each filter is a vectorized feature map of size D_l.

Let the extracted feature matrix represented by F. The feature matrix for layer l is $F_l[\cdot] \in R^{N_l \times D_l}$.

Style Transfer Key Idea



Objective function for Style Transfer

$$\mathcal{L}_{total} = \sum_{l \in [L]} \alpha_l \mathcal{L}_C^l + \gamma \sum_{l \in [L]} \beta_l \mathcal{L}_S^l$$

Here, α_l and β_l are used to balance the extracted feature layers and γ is used to balance the style and the content features in the output image.

Content Loss

The content loss $\mathcal{L}_C^l$ is defined as below.

$$\mathcal{L}_C^l(C, O; \phi) = \frac{1}{2N_l D_l} \sum_{i,j} \|F_l^C[O] - F_l^C[\mathcal{C}]\|_{ij}$$

- Here, l denotes feature matching at the l-th convolutional layer of CNN.
- $\mathcal{L}_C^l(C, O; \phi)$ denotes the feature comparison for layer l.

Content Loss

The content loss $\mathcal{L}_C^l$ is defined as below.

$$\mathcal{L}_C^l(C, O; \phi) = \frac{1}{2N_l D_l} \sum_{i,j} \|F_l^C[O] - F_l^C[\mathcal{C}]\|_{ij}$$

- Each layers contains N_l filters, where each filter is a vectorized feature map of size D_l.
- Let the extracted feature matrix represented by F. The feature matrix for layer l is $F_l[\cdot] \in R^{N_l \times D_l}$.

Style Loss

The style loss $\mathcal{L}_S^l$ is defined as below.

$$\mathcal{L}_S^l(S, O; \phi) = \frac{1}{2N_l D_l} \sum_{i,j} \|G_l^S[O] - G_l^S[\mathcal{S}]\|_{ij}$$

- Here, l denotes feature matching at the l-th convolutional layer and (i, j) denotes features index.

Style Loss

The style loss $\mathcal{L}_S^l$ is defined as below.

$$\mathcal{L}_S^l(S, O; \phi) = \frac{1}{2N_l D_l} \sum_{i,j} \|G_l^S[O] - G_l^S[\mathcal{S}]\|_{ij}$$

- G denotes the Gram matrix. G is the inner product between the extracted features maps. We have $G_l[\cdot] = F_l F_l^T \in R^{N_l \times N_l}$.

Style Transfer Results

Content Image

Artistic Style Transfer

Content Image

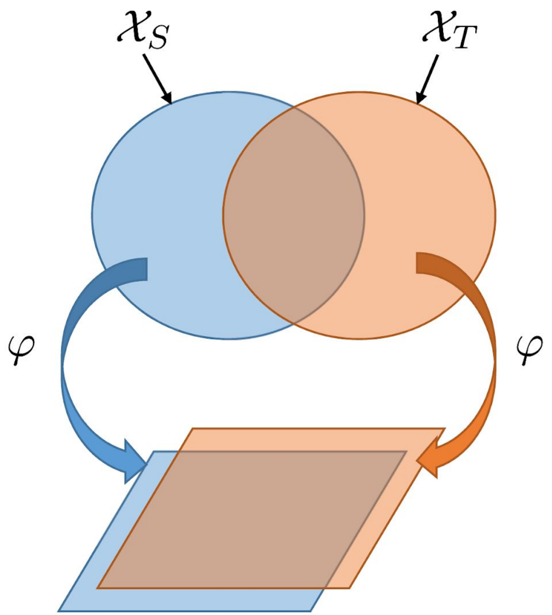
Style Image

Photorealistic Style Transfer

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Feature-based Transfer Learning

- When source and target domains only have some overlapping features
  - Lots of features only have support in either the source or the target domain
- Possible solutions
  - Encode application-specific knowledge
  - General approaches to learn the transformation φ

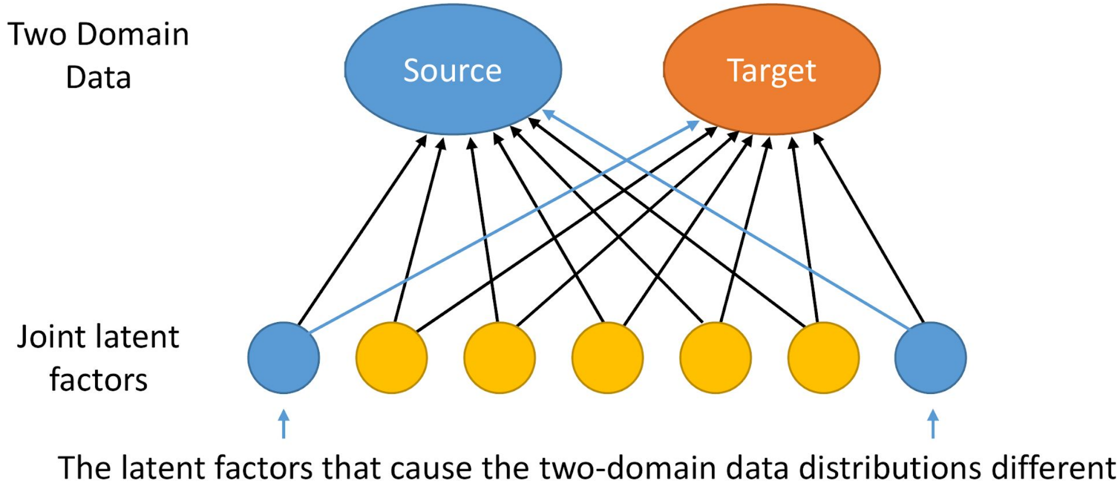


General Feature-Based TL Approach

- How to learn universal high-level features
  - Sparse Coding [Raina etal., 2007]
  - Autoencoder [Glorot *etal.*, 2011]
  - Other deep learning models, e.g., CNNs

Transfer Component Analysis

- Motivation
  - Minimize the distance between domain distributions by projecting data onto the learned transfer components



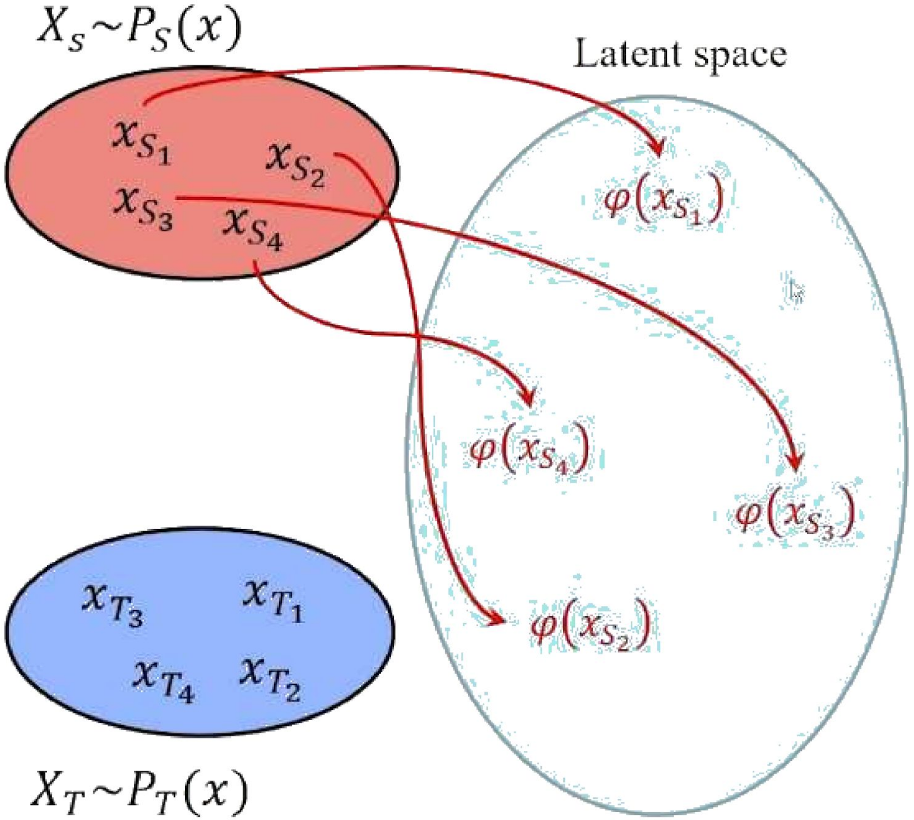
Pan, Sinno Jialin, et al. "Domain adaptation via transfer component analysis." *IEEE Transactions on Neural Networks* 22.2 (2011): 199-210.

Transfer Component Analysis

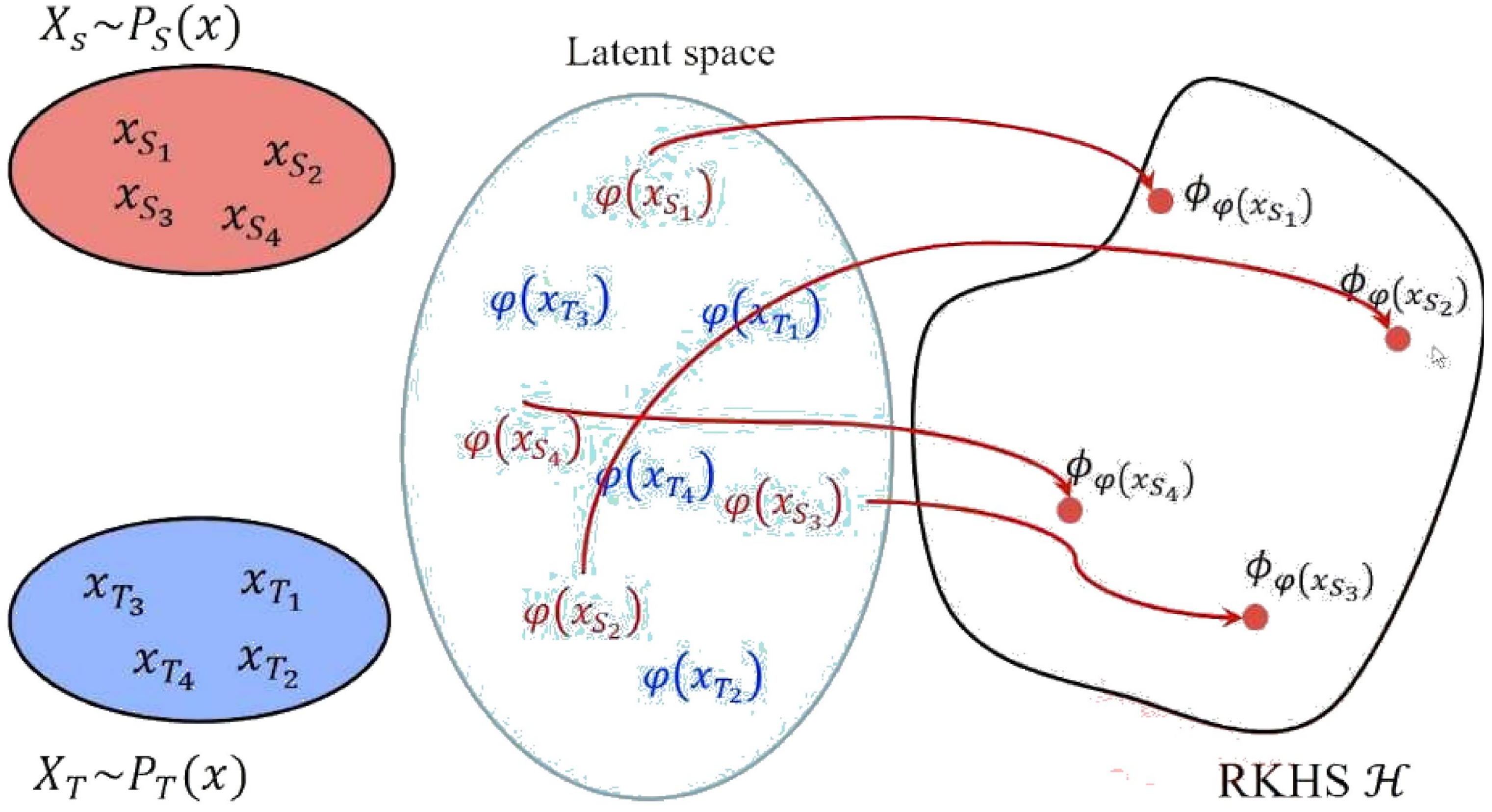
- Main idea
  - Learn φ to map the source and target domain data to the latent space spanned by the factors which can reduce domain difference and preserve original data structure

$$\begin{aligned} \min_{\varphi} \quad & \mathrm{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) + \lambda \Omega(\varphi) \\ \text{s.t.} \quad & \text{constraints on } \varphi(\mathbf{X}_S) \text{ and } \varphi(\mathbf{X}_T) \end{aligned}$$

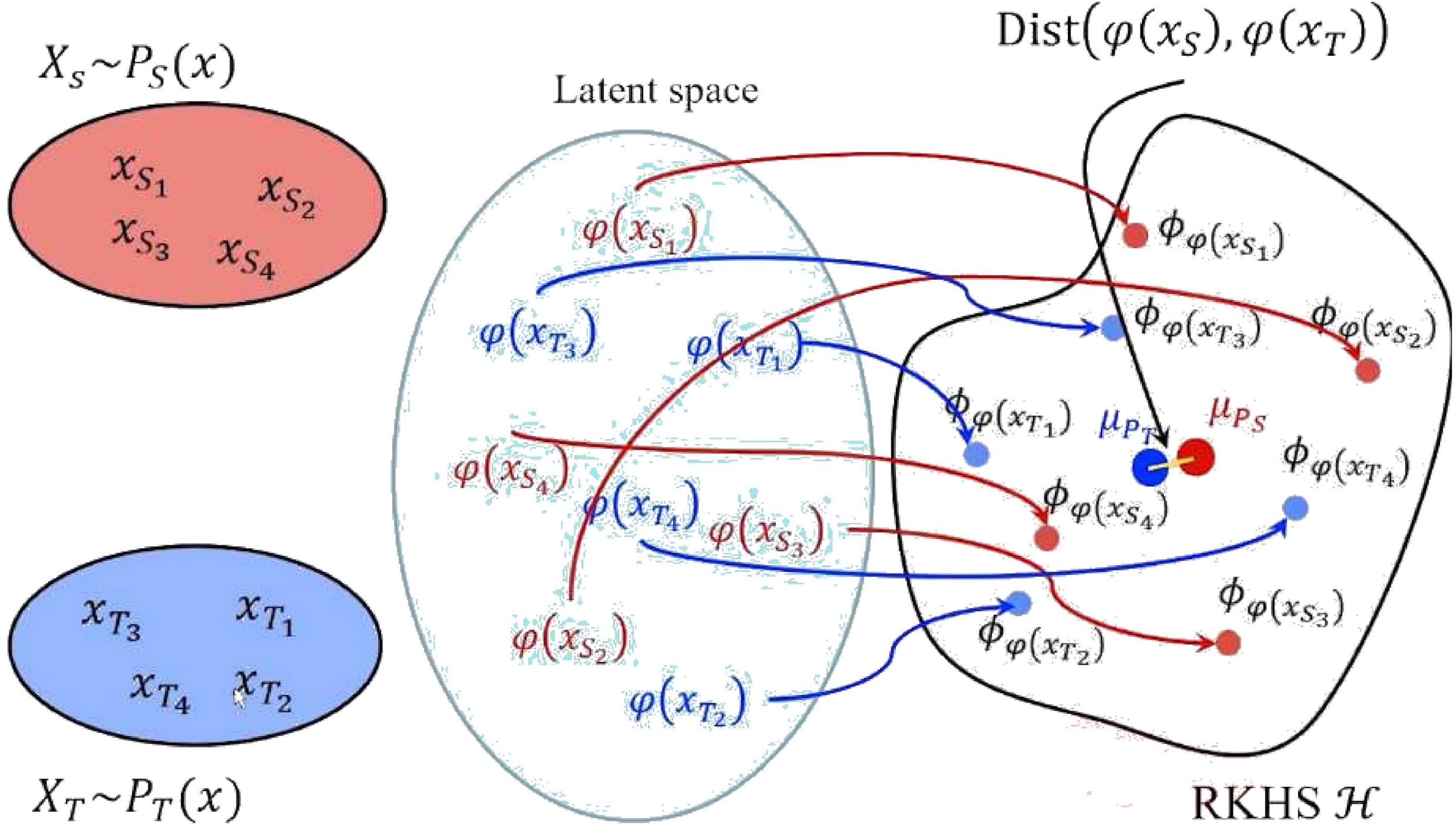
Transfer Component Analysis



Transfer Component Analysis



Transfer Component Analysis



Transfer Component Analysis

- Maximum Mean Discrepancy (MMD)
  - Given the source and target domain data

$$\mathbf{X}_S = \{x_{S_i}\}_{i=1}^{n_S} \qquad \mathbf{X}_T = \{x_{T_i}\}_{i=1}^{n_T}$$

drawn from $P_S(x)$ and $P_T(s)$ respectively

$$\mathrm{Dist}(\underset{\substack{\uparrow \\ \text{Mapping}}}{\varphi}(\mathbf{X}_S), \varphi(\mathbf{X}_T)) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \underset{\substack{\uparrow \\ \text{Kernel function}}}{\Phi}(\varphi(x_{S_i})) - \frac{1}{n_T} \sum_{i=1}^{n_T} \Phi(\varphi(x_{T_i})) \right\|_{\mathcal{H}}$$

Feature based approach in DL

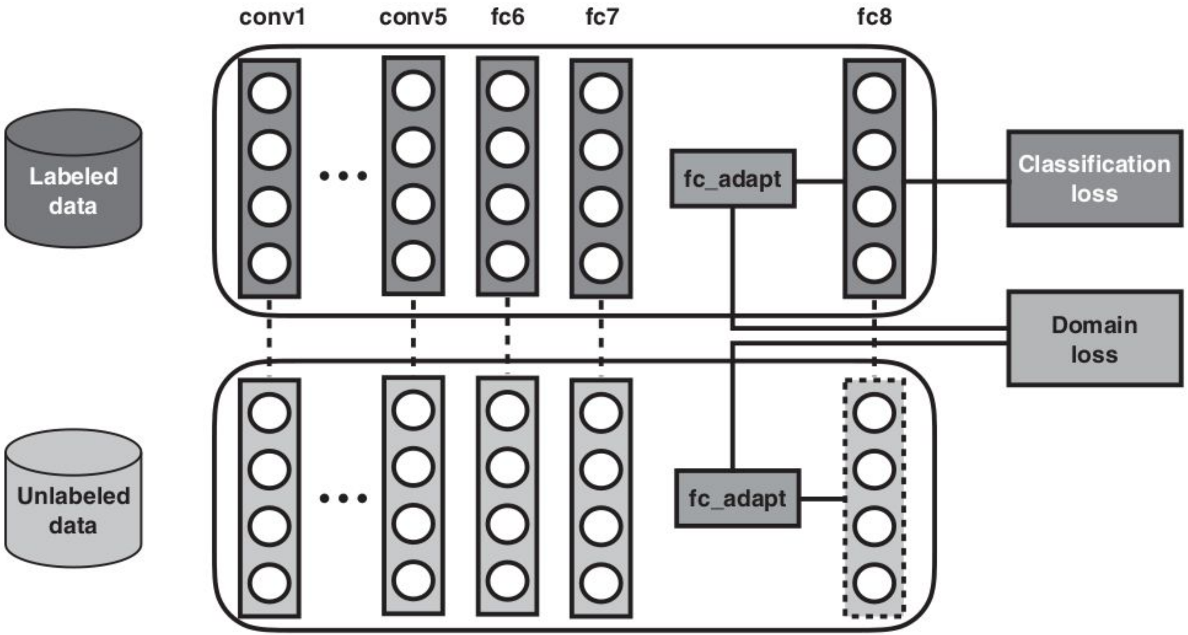
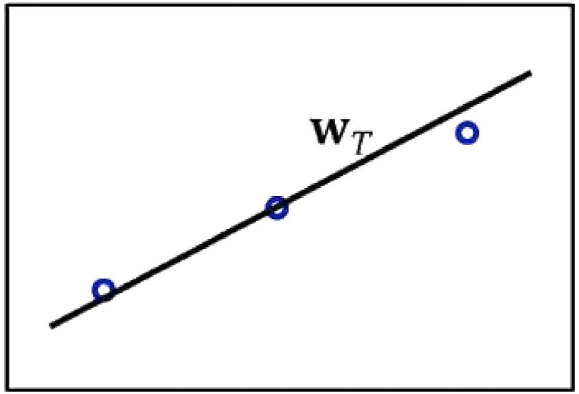
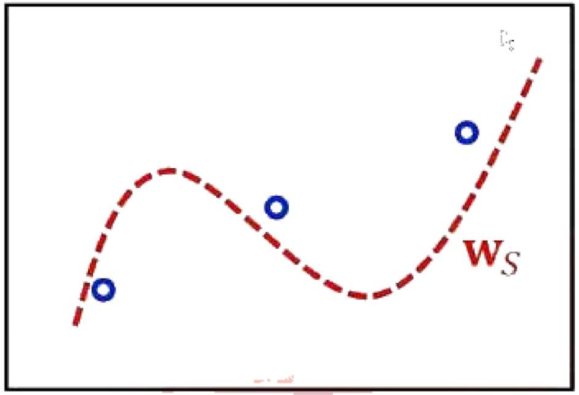


Figure 3.1 Deep CNN for both classification loss as well as domain invariance, where the dashed line means the weight sharing (adapted from Tzeng et al. [2014]).

Parameter based Transfer Learning

- **Motivation:** A well-trained source model $\mathbf{w}_S$ has captured a lot of structure from data. If two tasks are related, this structure can be transferred to learn a more precise target model $\mathbf{w}_T$ with a few labeled data in the target domain





Parameter based Transfer Learning

- The ϑ-parameterized function $f_\vartheta(x)$ learned on two domains

$$\theta_S^* = \arg\min_\theta \sum_{i=1}^{n_S} \mathcal{L}(y_{S_i}, f_\theta(x_{S_i})) + \lambda\Omega(\theta)$$

$$\theta_T^* = \arg\min_\theta \sum_{i=1}^{n_T} \mathcal{L}(y_{T_i}, f_\theta(x_{T_i})) + \lambda\Omega(\theta)$$

- Motivation
  - A well-trained model $f_{\theta_S^*}(x)$ has learned a lot of structure on the source domain.
  - If two tasks are related, this structure can be transferred to learn the model $f_{\theta_T^*}(x)$ on the target domain

Model Transfer via Deep Models

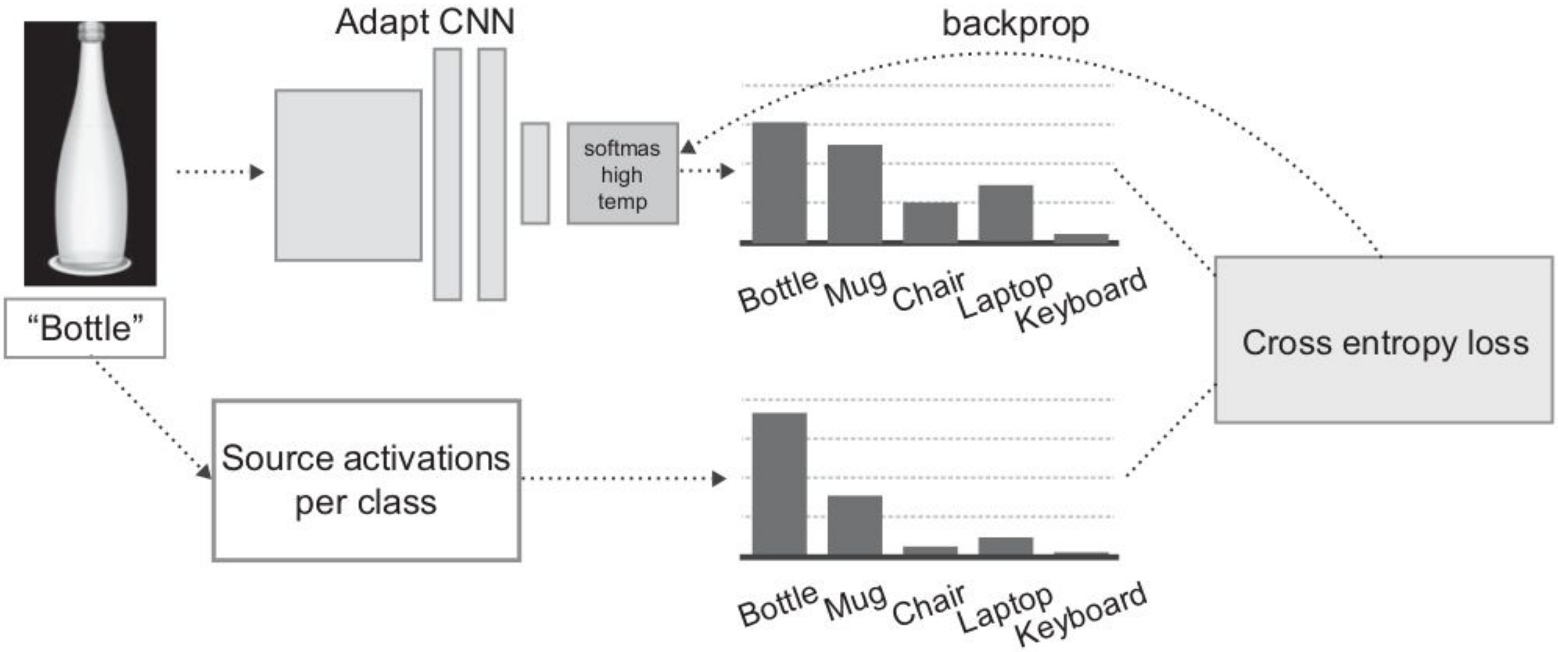


Figure 4.2 An illustration of the soft labeling method (adapted from Tzeng et al. [2014])

Relation-based transfer learning

- Relation-based transfer learning aims to build the mapping of the **relational knowledge** between the source relational domain and the target relational domain.
- Two mechanisms of relation-based transfer learning, including first-order relation-based and second-order relation-based transfer learning.

Relation-Based Transfer Learning (Structural Analogy)

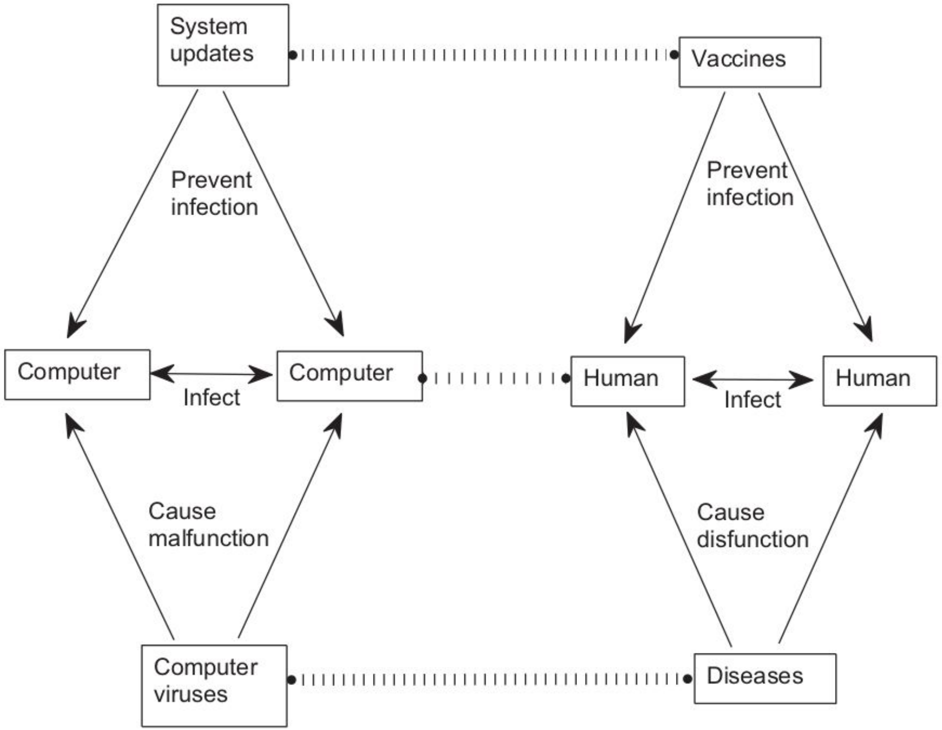


Figure 5.2 The analogy between debugging for computer viruses and diagnosing human diseases based on structural similarities. The dash lines bridge analogs across domains

Choosing TL Approaches with Caution

Benefits

- **Fine-Tuning**: Saves resources by adapting pre-trained models to new tasks.
- **Pretraining**: Most effective when source and target domains are alike.

Cautions

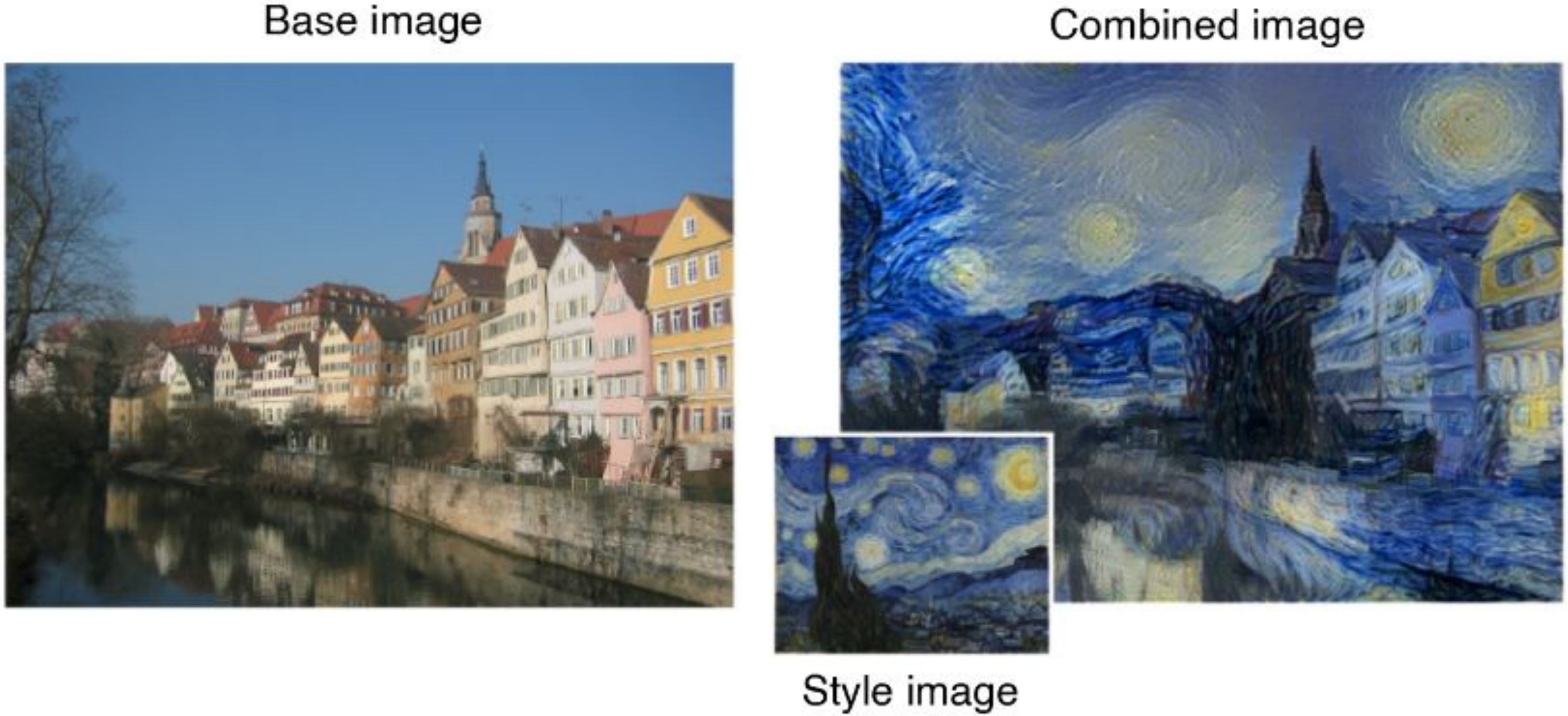
- **Domain Discrepancy**: Performance may drop if domains are too mismatched.
- **Task Complexity**: Highly complex tasks may not benefit as much from TL
- **Negative Transfer**: Incorrect application can worsen target task performance.

Semantic Style Transfer

Outline

- Content Mismatch Problem
- Semantic Style Transfer
- Semantic Style Transfer using Contextual loss
- Semantic Style Transfer using Segmentation Mask

Style Transfer



Content Mismatch Problem

Basic Style Transfer failure Case

CycleGAN failure Case

horse → zebra

Content Mismatch Problem

Basic Style Transfer failure Case

CycleGAN failure Case

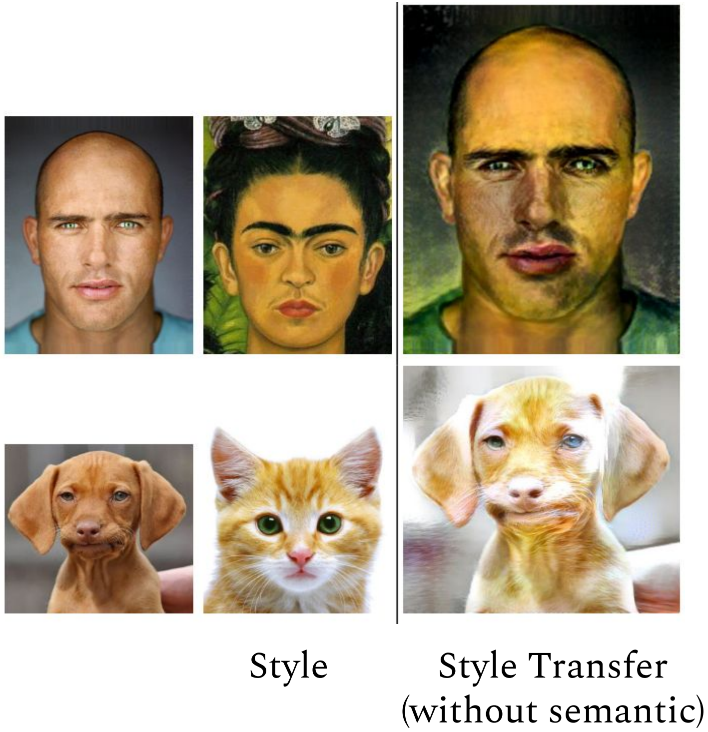
horse → zebra

Semantics of an image is related to the objects and their relative positions.

We need to learn how to preserve Semantics of input images in image transformation

Semantic Style Transfer

Neural Style Transfer using only content and style loss does not consider the image features supervision based on the semantics.

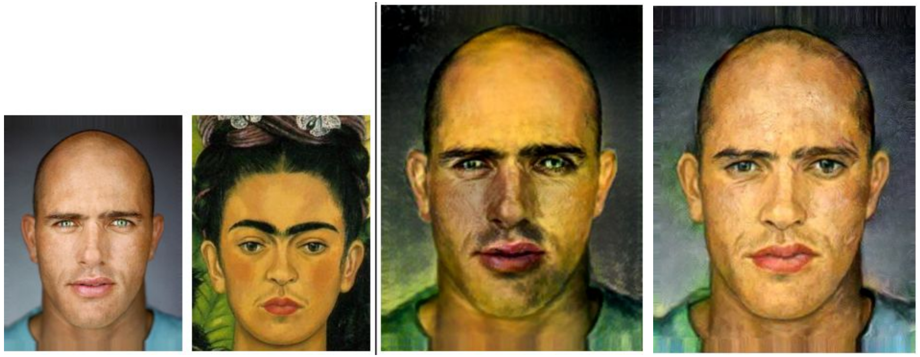


Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV).

Semantic Style Transfer

Neural Style Transfer using only content and style loss does not consider the image features supervision based on the semantics.

Semantic Style Transfer outputs images with better visual quality.



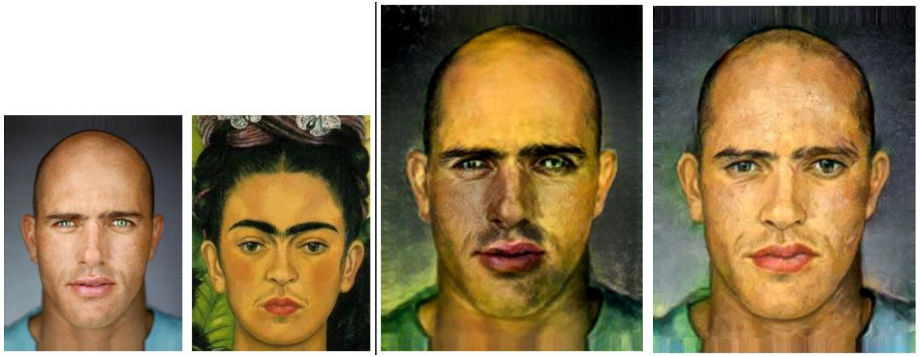
Content | Style | Style Transfer (without semantic) | Semantic Style Transfer

Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV).

Semantic Style Transfer

Image feature supervision based on the semantics could be performed using the following approaches:

> **(1) Using a specific loss term for the Style Feature Supervision.**
>
> (2) Using Segmentation Masks to Guide Style Feature Supervision

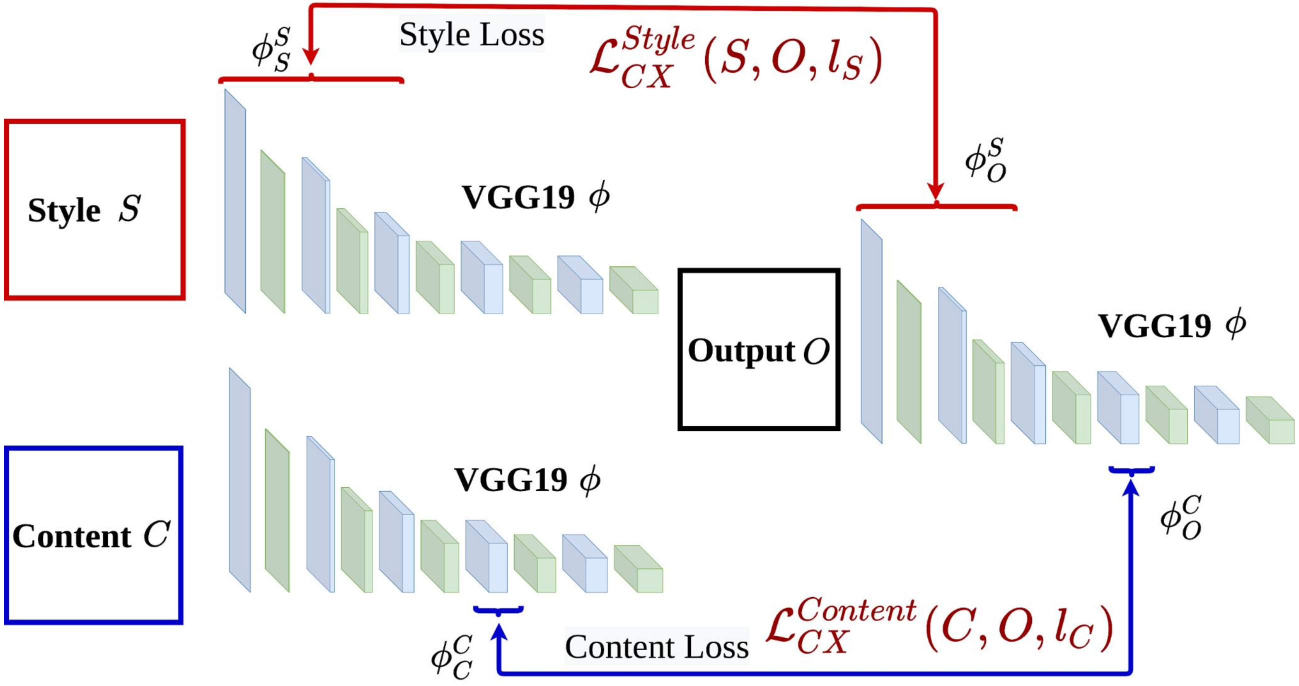


Content | Style | Style Transfer (without semantic) | Semantic Style Transfer

Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV).

(1) Using a specific loss term for the Style Feature Supervision.

Semantic Style Transfer using Contextual loss



Contextual loss for semantics of the images

The contextual comparison between two images would allow the feature matching between contextually similar object features.

A pretrained VGG19 network ϕ would allow us to capture semantics of an image represented by the high-level content representation.

Suppose two images x and y for which we want to preserve the semantics. The task is to extract the image representation using pre-trained feature extractor ϕ (e.g., VGG19 trained on Imagenet for classification) and use them to preserve the semantics.

Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV).

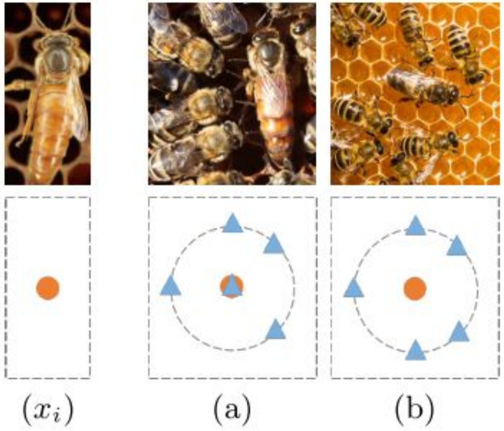
Contextual Similarity CX

CX is given as follows.

$$CX(\phi_l(x), \phi_l(y)) = \frac{1}{N}\sum_j \max_i CX_{ij}$$

Here, CX_{ij} is the similarity between the context vectors $\phi^l(x)_i$ and $\phi^l(y)_j$.

The contextual similarity CX_{ij} could be computed by using the cosine distance between feature vectors $\phi_l(x)_i$ and $\phi_l(y)_j$.



Contextual Loss L_{CX}

Let us denote extracted features from images x and y as $\phi(x)$ and $\phi(y)$.

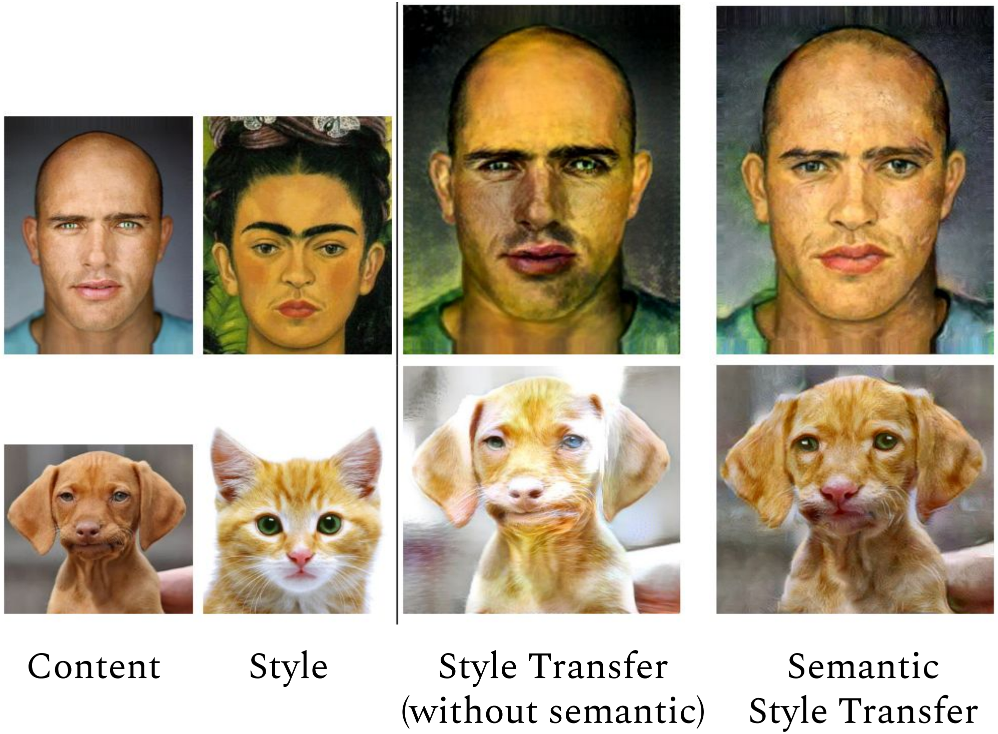
Suppose the high-level content information (**semantics**) is present at layer l of the feature extractor ϕ. The feature representations captured at layer l of ϕ is denoted by $\phi_l(\cdot)$.

The image context comparison using contextual loss defined as follows.

$$L_{CX}(x, y, l) = -\log CX(\phi_l(x), \phi_l(y))$$

Here, CX denotes the context similarity criterion, which allows us to compare the different objects in the image x and y based on their contextual similarity.

Semantic Style Transfer using Contextual loss



Semantic Style Transfer

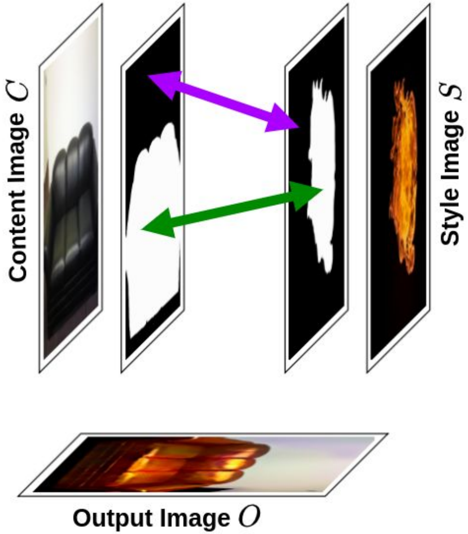
Neural Style Transfer using only content and style loss does not consider the image features supervision based on the semantics.

Image feature supervision based on the semantics could be performed using the following approaches:

(1) Using a specific loss term for the Style Feature Supervision.

(2) Using Segmentation Masks to Guide Style Feature Supervision

(2) Using Segmentation Masks to Guide Style Feature Supervision

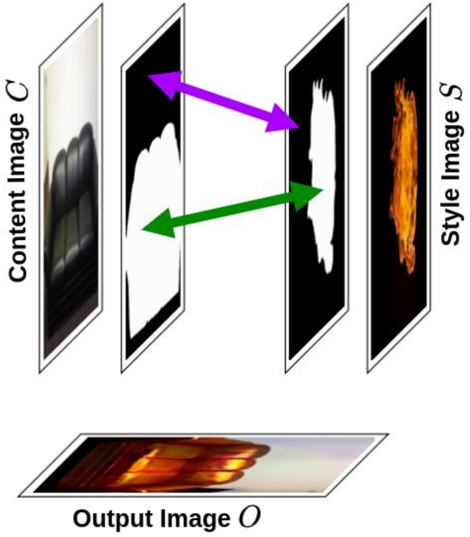


(2) Using Segmentation Masks to Guide Style Feature Supervision

$$\mathcal{L}_{s+}^{\ell} = \sum_{c=1}^{C} \frac{1}{2N_{\ell,c}^2} \sum_{ij} (G_{\ell,c}[O] - G_{\ell,c}[S])_{ij}^2 \quad \text{(3a)}$$

$$F_{\ell,c}[O] = F_{\ell}[O]M_{\ell,c}[I] \quad F_{\ell,c}[S] = F_{\ell}[S]M_{\ell,c}[S] \quad \text{(3b)}$$

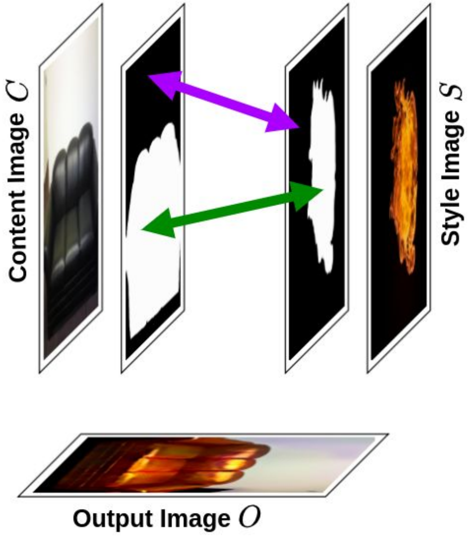
where C is the number of channels in the semantic segmentation mask, $M_{\ell,c}[\cdot]$ denotes the channel c of the segmentation mask in layer ℓ, and $G_{\ell,c}[\cdot]$ is the Gram matrix corresponding to $F_{\ell,c}[\cdot]$. We downsample the masks to match the feature map spatial size at each layer of the convolutional neural network.



Luan, Fujun, et al. "Deep photo style transfer." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

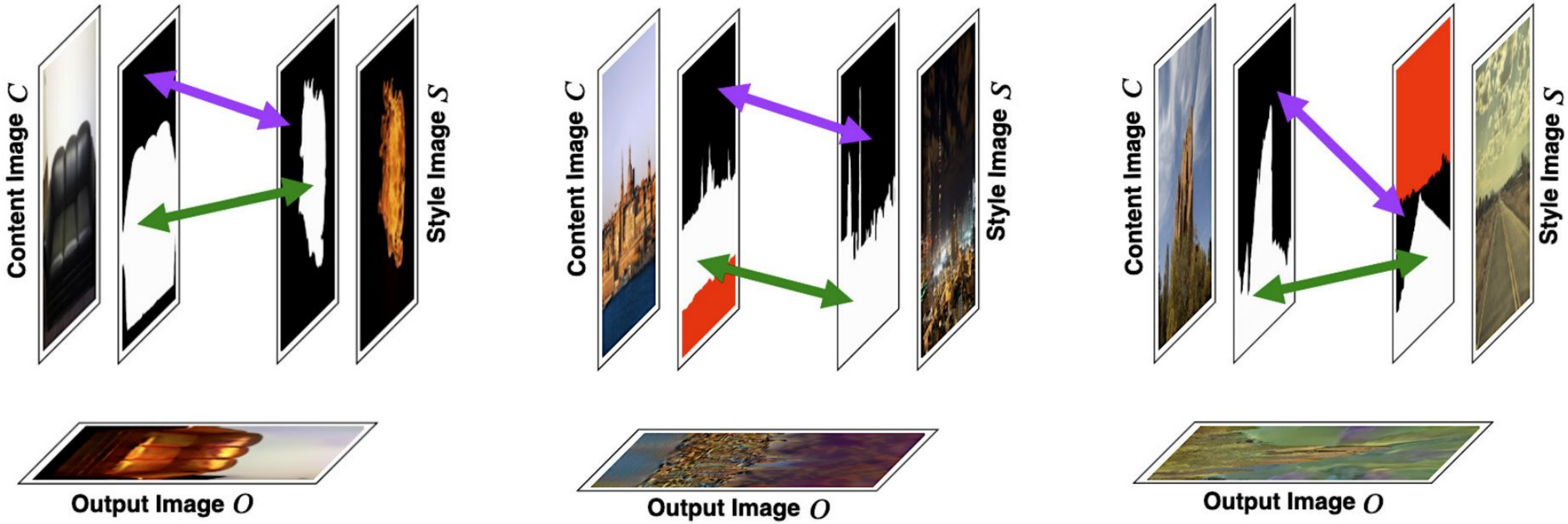
(2) Using Segmentation Masks to Guide Style Feature Supervision

Note: Segmentation based approaches is good when both style and content has equal number of objects.



Semantic Style Transfer using Segmentation Mask

<u>Note</u>: Segmentation based approaches may suffer from content mismatch when the input images contains different number of objects.



Mastan, I. D., & Raman, S. (2021). DeepObjStyle: deep object-based photo style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Further Reading

- Talebi, H., & Milanfar, P. (2018). NIMA: Neural image assessment. IEEE Transactions on Image Processing.
- Prashnani, E., Cai, H., Mostofi, Y., & Sen, P. (2018). Pieapp: Perceptual image-error assessment through pairwise preference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2019). Neural style transfer: A review. IEEE transactions on visualization and computer graphics.
- Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV).
- Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision.
- Luan, Fujun, et al. "Deep photo style transfer." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.