Department of Intelligent Systems
Faculty of Electrical Engineering, Mathematics and Computer Science

Domain adaptation networks for noisy image classification

Master Thesis

Chengqiu Zhang

Supervisors:	Committee:
Dr. Jan van Gemert	Prof. Martha Larson
Dr. Silvia-Laura Pintea	Dr. Jan van Gemert
Dr. Ildiko Suveg	Dr. Marco Loog
	Dr. Silvia-Laura Pintea
	Dr. Adriana Gonzalez

Eindhoven, Aug 2017

Abstract

In this thesis, we propose a novel semi-supervised clean-noisy datasets adaptation algorithm. We transfer the knowledge learned on clean images to unlabeled noise-distorted ones. This modification on standard deep networks produce stable classification performance on all distortion levels, which brings benefit to real-world cases. Specifically, we propose a strategy to jointly learn a shared feature encoder on the network, i.e., *i*) discrimination capability of network is learned by supervised training on labeled source (clean) dataset, *ii*) knowledge transferring is achieved by unsupervised domain adaptation to map features extracted from both domains (clean and noisy) to a common space. Our proposed network is optimized by a two-step backpropagation strategy, similar to that of Generative Adversarial Networks (GANs).

We evaluate our proposed network on two popular datasets, where both show clear improvement of classification performance compared to preprocessing noisy images using the state-of-the-art denoising algorithm BM3D (up to ~19% in average accuracy over all noise levels). Interestingly, we also observe that the proposed approach efficiently improves the feature transferability on very deep architectures, which is challenging for previous domain adaptation methods. In the future, we can also explore more challenging domain adversarial tasks like distorted image segmentation with the proposed algorithm.

Contents

Chapter 1

Introduction

Scene categorization or scene classification is a well defined topic [44], which is predicting the general semantic categories of a scene, such as *Highway* or *Casino*. Such semantic information of the observed scene provides an accurate description and benefits further applications, like high-level localization or system control. It is very useful in real-world applications like self-driving cars [7] and intelligent surveillance [27]. Recent progress in deep learning ensures the high performance of scene classification and understanding. But real-world cases are more complicated than that in the lab, i.e., the collected images suffer from noise distortion and other imperfections. In this chapter, we will briefly review the process of scene classification. Then we will discuss the problems for real-world cases. At last, the contributions of our work in this thesis will be introduced.

1.1 Scene Classification

Scene classification includes perceiving scenes and understanding their content, which is different from object classification. A scene contains background and objects, the concepts of which have no explicit definition in various tasks. Usually, background means the wide still surfaces or frameworks, such as mountain, sky, walls etc.. Objects are usually discontinuous and relatively small in a scene.

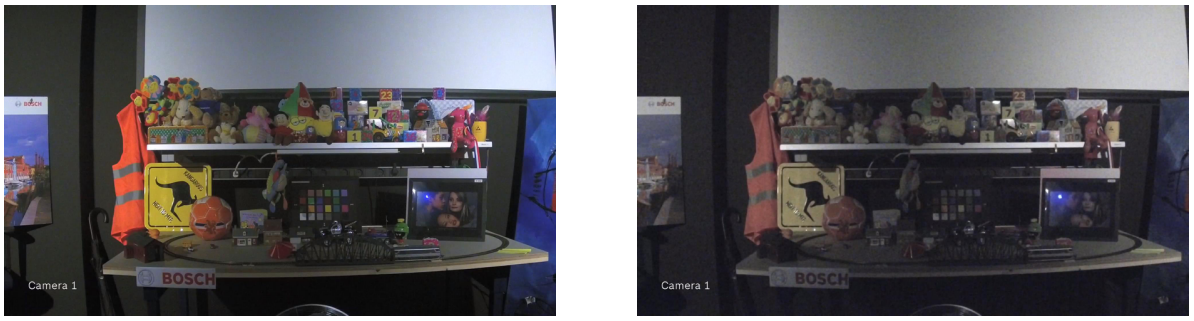
Saliency of a scene was widely leveraged before CNN for scene classification. Itti *et al.* [22] developed a computer model that simulates the human brain's mechanisms involved in the deployment of visual attention. When humans are facing scene images, only part of the scene captures most of the attention, even if it is unconsciously. A good saliency model can improve the performance of image analysis, as well as save computational resources, for only part of the data are being processed. Saliency feature extraction is widely utilized in many areas of computer vision, including segmentation [1] and object recognition [35].

Different from the aforementioned saliency model, spectral residual approach proposed by Hou *et al.* [21] analyzes the log-spectrum of an input image, and thus the spectral residual

of an image is extracted. It is a fast method to construct the saliency map in spatial domain. Based on this, the sparse coding method was a predominant approach in machine learning area until the renaissance of deep learning [53, 17, 16, 42].

After the incredible success of deep learning in computer vision domain, better performances are achieved by various standard networks, and furthermore, large scale scene classification dataset [57] ensures the probability of utilizing deep learning methods on various tasks. And through the comparison experiment done by Sunderhauf *et al.* [45] on state-of-the-art ConvNets, pre-trained deep learning networks have performance advantages for semantic place categorization.

1.2 Restrictions in Real-world Case



(a) good illumination (b) poor illumination

Figure 1.1: Scene collected by a Bosch camera (DINION IP 4000 HD indoor box IP camera) under different illumination conditions, we can see obvious distortion (noise) on images or the illumination conditions are poor.

Consider a company that specializes in manufacturing security cameras, integrating scene understanding functionalities can help to improve video quality and save installation cost. These cameras are used in many types of user scenarios, like *Indoor*, *Outdoor*, etc. To provide better results for specific user scenarios, cameras have several fixed modes with corresponding hardware settings. For example, surveillance of indoor versus outdoor scenes has corresponding models since many image pipeline settings and algorithms behave differently in each case. Parameters of automatic white balance algorithm, sharpness/contrast improvement and noise reduction would be adjusted based on the detected scene type. Therefore, supporting engineers need to set specific scene mode manually every time when installing cameras. To save expensive human cost and to get a robust surveillance quality, the ability to recognize and interpret the environment is essential.

Due to dynamic changing of weather and illumination, frames collected in real-world are usually of poor quality compared to standard dataset such as ImageNet [10] and Places [58].

Figure 1.1 shows the contrast of images collected by a Bosch security camera (DINION IP 4000 HD indoor box IP camer) under different illumination condition. We can see the obvious noise under poor illumination. To get a stable performance of scene classification, we need to make our network invariant of noise. Two strategies are available: *i*) first remove noise then operate the normal scene classification algorithm, or *ii*) modify existing scene classification architectures to make it capable of ignoring noise as well as categorization. Learning invariance to noisy data can be achieved by using domain adaptation techniques. And we will propose our modification with this methodology.

1.3 Contributions

The goal of this thesis is to explore the possibilities to build a noise invariant and computationally efficient network for scene classification. To be specific, this network is required to be accurate enough on classification task not only for clean image, but also noisy images at different levels without adding any complexity. Figure 1.2 is the proposed network. It consists of a shared encoder and a classifier. Both clean and noisy samples will pass through shared encoder and they are mapped to a noise-invariant feature space. Classifier will finally give a predicted label based on the feature extracted from encoder.

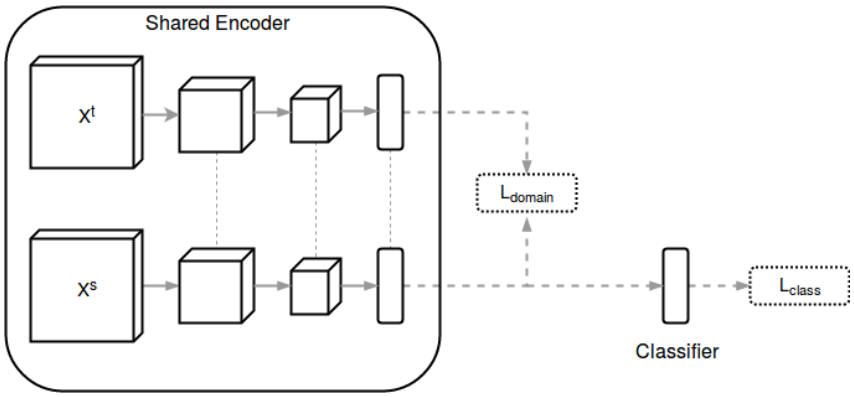


Figure 1.2: The proposed architecture for noisy-clean domain adaptation, includes a shared encoder, and a label classifier. Two-step training is processed: trying to minimize domain loss (L_{domian}) and classification loss (L_{class}), respectively. Shared encoder is constituted of the convolutional layers of pre-trained ResNet-50. It ensures the feature distribution over both domains are similar, resulting in domain-invariant features.

This thesis makes the following contributions:

- We use domain adaptation methods to solve the scene recognition in presence of noise. Based on the principle of noise-invariant feature mapping, we get an end-to-end network robust to noise.

- We integrate the MMD (Maximum Mean Discreapancy) distance in the network architecture, and the two-steps training performs better than previous domain adaptation networks. This can also be applied to other low-level semantic tasks in the future.
- Robustness of classification performance has been proved on two datasets with the presence of increasing noise. The proposed network substantially improves on the noisy images classification accuracy of the deep network without adding any complexity. It outperforms conventional denoiser.

Chapter 2

Related Work

Recent techniques based on deep neural networks (DNN) have achieved state-of-the-art results in various computer vision tasks [41]. It is of particular interest to deep learning networks is that deep networks trained on large scale datasets also work well in other tasks with certain modification.

Despite the impressive performance on semantic tasks from low level to high level, it is confirmed that deep networks are susceptible to adversarial samples [18]. Adversarial datasets are generated by adding worst case noise to original images. Even though these adversarial noise is imperceptible to human eyes, it can still confuse the network and give wrong predictions with a high confidence. On the other hand, the noise is carefully designed via adversarial methods like the optimization algorithm in [49]. Adversarial noise is a very interesting task, however, it is unlikely to see this kind of deliberately designed noise in real-world application. Many of the cases are i.i.d. Gaussian like noise distortion. In the following chapters we will further explore this case and give solution to make the network's performance invariant of noise influence.

In this chapter, we will give a brief overview of noise removal methods, especially the one that would be used in this thesis. Then we will introduce the common practices in deep networks, including the noise ignoring adversarial training methods and transfer learning. The latter based on fine tuning is also a key step implemented in this thesis. At last, we will overview the networks that are used in this thesis.

2.1 Removing noise directly from the image

Image restoration methods is a common way of dealing with noise distorted images. Numerous and diverse denoising approaches exist, which take a noisy image as input and produce a noise-reduced output.

One approach for denoising is to transfer the image from spatial domain to an alternative

domain where the noise signal is easier to be separated from the true information[52, 38, 30]. For example, Portilla *et al* [38] propose the Bayes Least Squares with a Gaussian Scale-Mixture (BLS-GSM) method, based on the wavelet transformation.

Another strategy is to capture image statistics directly in the image domain. Following this methodology, a number of models exploiting the (linear) sparse coding technique has gained much attention [36, 24, 14, 28, 34, 33]. Sparse dictionary learning methods reconstruct images from a sparse linear combination of an over-complete dictionary. In recent research, the dictionary is learned from data instead of hand crafted as before. This learning step improves the performance of sparse coding significantly. One example of these methods is the KSVD sparse coding algorithm proposed in [14].

Block-matching and 3D filtering (BM3D)[8] is different from the aforementioned wavelet shrinkage methods and non-local methods (NLM). It combines advantages both. That is, it not only utilizes inter-patch-correlation information in wavelet shrinkage, but also the intra-patch correlation used in NLM. BM3D algorithm first finds similarities on the image, and then transforms the generated image patches to spectral domain. This well engineered method represent the current state-of-the-art computer vision algorithm for natural image denoising. We will also use BM3D approach as our baseline to evaluate our proposed methods for noisy scene classification. Bm3D noise removal requires noise level as a precondition before operating. It is still very challenging when denoising the images from its noisy version without any knowledge of the noise. While our goal in this thesis is to solve this unsupervised task.

2.2 Learning Noise-invariant Features

Despite the fact that these decisioning methods perform well in practice, they all share a shallow linear structure. However, recent research suggests that deep, non-linear models have superior performance image restoration. Jain *et al* [23] proposed a method based on convolutional neural networks(CNN) to denoise images. Burger *et al.* [4] even show plain multi-layer perceptron (MLP) networks trained on large scale of images can compete with state-of-the-art denoising algorithms.

2.2.1 Restoration via deep networks

The common practice of restoration using deep networks is to extract noise-invariant features and then reconstruct the images. Dong *et al* [13] proposed a deep networks for image super-resolution and demonstrated a significant performance improvement compared with other traditional methods. To better extract image features from its noisy version, one strategy [12] is to design a multi-scale feature extraction layer, using regional correlation in the image to compensate the lack of information due to noise.

On the other hand, Xie *et al* [51] not only built an encode-decoder architecture for Image Denoising and Inpainting, but also examined their novel methods in the high-level tasks. They found that it is more effective and can improve the performance of unsupervised image classification. Similarly, in [50], they also found that denoising can also recover the classification accuracy.

2.2.2 High-level tasks

As mentioned above, the denoising processing can help improve the high-level tasks. However, as [54] claimed, most of those deep learning techniques aim at minimizing mean-squared-error (MSE) between a denoised image and the ground truth, which results in losing important structural details due to over-smoothing, although the PSNR based performance measure looks great. Therefore, they introduced a perceptual loss, intending to keep the critical structural information for diagnostic confidence. By comparing the performance of denoised results evaluated via peak signal-to-noise ratio (PSNR) and those really help much for classification task [11], a key fact is revealed. That is, low-level image processing like image denoising which intended to improve the high-level computer vision tasks like classification, is very different from producing visually pleasant images valued via PSNR. In this thesis, our task is to get the noise invariant classification network. Consequently we are not going to output visually satisfying images, but to adapt the extracted features to the source domain (extracted from clean images).

2.3 Domain Adaptation methods

Transfer learning is a commonly accepted unsupervised learning practice in real-world applications. The features of a deep neural network learned from source domain data is transferable [29, 41] to target domain data in a novel scenario. The ability of transfer learning depends on the correlation between multiple tasks, and the transferable knowledge is thus based on this correlation. Yosinki *et al.* [55] found that as the network goes deeper, features must eventually transition from general to specific, and the transferability of features drops significantly. If the new model works well on both tasks, we assume that the generalization of our model is better than the original network. Domain adaptation which pertains to transfer learning, is the process of adapting source domain for the means of transferring information to improve the performance of a target learner.

2.3.1 Problem Formulation

The following section lists the notation and definitions of domain adaptation that are also used in this thesis. The notation and definitions in this section match those from the survey paper by Pan and Yang [37].

The domain adaptation process tries to alter a source domain in an attempt to bring the distribution of the source closer to that of the target. A domain D is defined by two parts, a feature space X, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$, and a marginal probability distribution $P(X)$. For a given domain D, a task T is defined by two parts, a label space $\mathcal{Y}$, and a predictive function $f(\cdot)$, which is learned from the feature vector and label pairs $\{x_i, y_i\}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. From the definitions above, we have a domain $D = \{\mathcal{X}, P(X)\}$ and a task $T = \{\mathcal{Y}, f(\cdot)\}$. Therefore, D_S is defined as the source domain data where $D_S = \{(x_{S1}, y_{S1})..., (x_{Sn}, y_{Sn})\}$, where $x_{Si} \in \mathcal{X}_S$ is the ith data of D_S and $y_{Si} \in \mathcal{Y}_S$ is the corresponding class label for x_{Si}. In the same way, D_T is the target domain data where $D_T = \{(x_{T1}, y_{T1})..., (x_{Tn}, y_{Tn})\}$, $x_{Ti} \in \mathcal{X}_T$ is the ith data of D_T and $y_{Ti} \in \mathcal{Y}_T$ is the corresponding class label for x_{Ti}. Further, the source task is detonated as T_S, the target task as T_T , the source predictive function as $f_S(\cdot)$, and the target predictive function as $f_T(\cdot)$.

Given a source domain D_S with a corresponding task T_S and a target domain D_T with a corresponding task T_T, domain adaptation is the process of improving the target predictive function $f_T(\cdot)$ by using the related information from D_S and T_S, where $D_S \neq D_T$. Since $D_S = \{\mathcal{X}_S, P(X_S)\}$ and $D_T = \{\mathcal{X}_T, P(X_T)\}$, the condition where $D_S \neq D_T$ means that $\mathcal{X}_S \neq \mathcal{X}_T$ and/or $P(X_S) \neq P(X_T)$.

Daume *et al.* [9] and Chattopadhyay *et al.* [5] define supervised transfer learning as the case of having abundant labeled source data and limited labeled target data, and semi-supervised transfer learning as the case of abundant labeled source data and no labeled target data. In this thesis, abundant labeled source data and no labeled target data are available. The proposed method is thus semi-supervised transductive transfer learning.

2.3.2 Domain Adaptation Methods

In recent years, implementing domain adaptation via training deep networks has been explored. Ganin *et al.* [15] proposed a Domain-Adversarial Neural Networ (DANN), in which the mismatch of extracted feature distribution between source and target domains are reduced using reversing the gradient of the domain classification loss. By maximizing such "confusion", domain classifier cannot reliably predict the domain of the encoded representation and thus domain invariant features can be extracted. Similarly, Long *et al.* [29] leveraged the multiple kernel variant of Maximum Mean Discrepancy (MMD) objective as the similarity metric among feature spaces between source and target domain. By regularized training of deep networks, knowledge learned on the labeled source samples can be transferred to those unlabeled target samples prediction. These methods aim at finding a common feature space that is domain invariant, which is similar to the task in this thesis, that is to find a noise invariant feature space.

2.4 Relevant Architectures

Deep learning remained controversial until 2012, the proposed AlexNet [26] had a remarkable performance on ImageNet classification. They also won the well-known ImageNet Challenge in 2012. The CNN became widely accepted. Inspired by AlexNet, several variants have been proposed in the following years. The most popular ones are VGGNets [43], googLeNets [47], and ResNet [20]. The comparison of different architecture in detail is listed in Table 2.1. In the following we will explain why we select residual networks in the proposed architecture.

2.4.1 Residual Networks

	AlexNet	VGG	GoogLeNet	ResNet
First Released Year	2012	2014	2014	2015
Top-5 Error	16.4%	7.3%	6.7%	3.57%
Data Augmentation	+	+	+	+
Number of Conv-layer	5	16	21	151
Size of Conv-kernel	11,5,3	3	7,1,3,5	7,1,3,5
Number of FC-layer	3	3	1	1
Size of FC-layer	*4096* *4096* *1000*	*4096* *4096* *1000*	*1000*	*1000*
Batch Normalization	-	-	-	+

Table 2.1: Comparison of AlexNet, VGG, GoogLeNet, ResNet on ImageNet competition.

The latest residual networks proposed by He *et al.* [20] got a large success winning ImageNet and COCO 2015 competition. It also has achieved several state-of-the-art benchmarks, including object classification on ImageNet and CIFAR, object detection and segmentation on PASCAL VOC and MS COCO. Compared to inception networks [46], ResNets have better generalization, since its features can be utilized in transfer learning with higher efficiency [56]. In addition, ResNets can be scaled up to thousands of layers and still improve the performance. Residual block with identity mapping can be represented by the following formula:

$$X_{l+1} = X_l + F(X_l, W_l) \tag{2.1}$$

where X_l and X_{l+1} are input and output, respectively, of the l-th unit in the network, F is a residual function and W_l are parameters of the block. Residual function refers to two or three connected convolutional layers, and the parameters are the corresponding kernel weights. Residual network consists of sequentially stacked residual blocks. The block is shown in Fig 2.1. The order of the batch normalization, activation and convolution layers

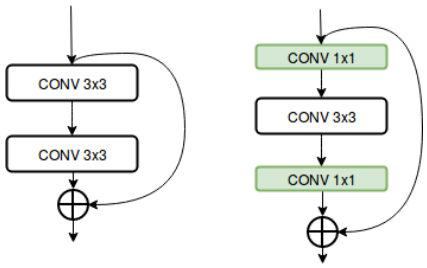


Figure 2.1: Residual blocks used in ResNet, left: basic, right: bottleneck. The simplified ResNets tested in this thesis utilized the basic blocks. In the proposed architecture, the encoder based on ResNet-50 leveraged the latter.

in a residual block is `BN-ReLU-conv`. When training CIFAR-10 dataset from scratch, we used the simplified ResNet, in which the first layer is 3×3 convolutions. Then we use a stack of $6n$ layers with 3×3 convolutions on the feature maps of sizes $\{32, 16, 8\}$ respectively, with $2n$ layers for each feature map size. The numbers of filters are $\{16, 32, 64\}$ respectively. We also build our proposed architecture using fine-tuned ResNet-50 as shared feature extractor.

We select ResNet as the base of our proposed domain adaptation network because of its outstanding performance as well as its scaling capabilities for different datasets. And ResNet with various depth helps us to understand how these domain-invariant features are influenced as the network goes deeper.

Chapter 3

Method

We are given a source domain $D_S = \{(x_{S1}, y_{S1})..., (x_{Sn}, y_{Sn})\}$, where $x_{Si} \in \mathcal{X}_S$ is the ith data of D_S and $y_{Si} \in \mathcal{Y}_S$ is the corresponding class label for x_{Si}. Target domain is $D_T = \{x_{T1}..., x_{Tm}\}$, where $x_{Ti} \in \mathcal{X}_T$ is the ith data of D_T, but corresponding labels y_T is unknown. Test samples are located in the target domain. We intend to achieve a deep adaptation network (DAN) to transfer classification capability learned on source data to target domain. A feature extractor $G(\cdot)$ which is capable to overcome dataset bias between two domains is the goal.

The methodology to solve the problem is based on the assumption of shared-encoder:
there exists a shared space for cross-domain feature representation and can be extracted by a shared encoder, then classifiers embedded with this encoder can work well both on source and target domain.

Our task is to extract noise invariant features, that is, we want to make the feature distribution $S(h_s) = \{G_f(x_s)\}$ and $T(h_t) = \{G_f(x_T)\}$ as similar as possible. Measuring the similarity between $S(h_s)$ and $T(h_t)$ is non-trivial, for the extracted feature (h_s and h_t) are high dimensional and constantly changing during the training process. In this chapter, we discuss about the two strategies that are based on different similarity representations for feature spaces, for the semi-supervised task on non-labeled noisy images.

3.1 Baseline domain adaptation model for noisy images

The deep network architecture utilized is shown in Figure 3.1, where a shared encoder maps input from source and target domains to feature space, $\mathcal{H}_s$ and $\mathcal{H}_t$ respectively, with parameter θ_g. $h_s \in \mathcal{H}_s$ and $h_t \in \mathcal{H}_t$ represents every single features encoded from each input sample. The classifier is a normal multi-layer perception network with parameter θ_y and only takes h_s with corresponding category label y_s. The adaptor has h_s, h_t as input which are relabeled as $\{0, 1\}$ respectively, to mark which domain they are from. The

adaptor works the same as the domain classifier except for an additional Gradient Reversal Layer (GRL). GRL reverses the domain loss gradient from the subsequent layer and passes it to the preceding layer, aiming at confusing the shared encoder of the origin each feature belongs to.

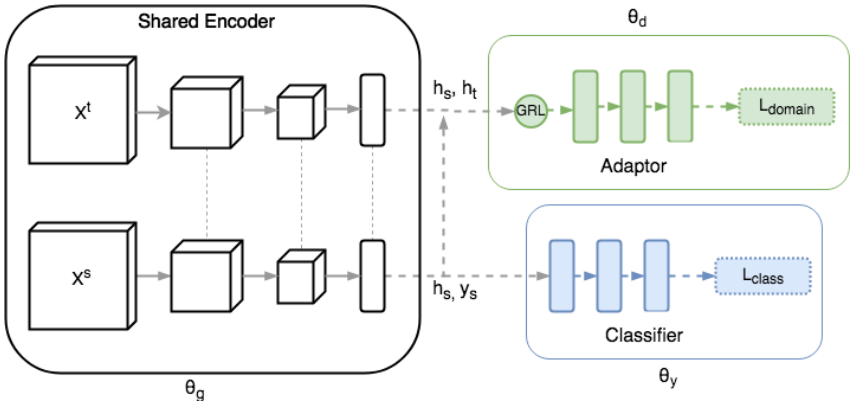


Figure 3.1: DANN architecture for noisy-clean domain adaptation,includes a feature extractor (the shared encoder), label classifier(blue), and domain adaptor (green). The domain adaptor is connected with the encoder via gradient reversal layer, which multiplies the gradient by a certain negative constant during back propagation process. Training process consists in minimizing the sum of label prediction loss and domain classification loss ($L_{class} + L_{domain}$). Gradient reversal ensures that the feature distribution over both domains are similar, resulting in domain-invariant features.

3.1.1 Model

Based on our idea, we are seeking parameters $\hat{\theta}_g$, $\hat{\theta}_y$, $\hat{\theta}_d$ by optimizing both of the functionals below:

$$(\hat{\theta}_g, \hat{\theta}_y) = \arg\min_{\theta_g, \theta_y} L_y(\theta_g, \theta_y, \hat{\theta}_d) \tag{3.1}$$

$$\hat{\theta}_d = \arg\max_{\theta_d} L_d(\hat{\theta}_g, \hat{\theta}_y, \theta_d) \tag{3.2}$$

where $L_y(\cdot)$, $L_d(\cdot)$ represent the loss functionals for the classifier and adaptor respectively.

Therefore, take the whole network with all parameters (θ_g, θ_d, θ_y) into account, we are seeking a saddle point. To make the gradient decent based training process feasible we define a new training loss:

$$L = L_{classifier} - \lambda L_{domain} \tag{3.3}$$

where

$$L_{classifier} = \sum_{\substack{i=1,\cdots,N \\ d_i=0}} L_y^i(\theta_g, \theta_y) \tag{3.4}$$

$$L_{domain} = \sum_{i=1,\cdots,N} L_d^i(\theta_g, \theta_d) \tag{3.5}$$

Where N is batch size. The hyper-parameter λ controls the trade-off between two tasks (label classification and feature domain adaption) that jointly work when encoding the features during training.

3.1.2 Optimization

Different from the common practice in Generative Adversarial Networks (GANs) [19], whose parameters in one of the trainings (on generative or discriminative part) must be fixed when training another one, we do the update for all parameters simultaneously at each iteration.

In order to make the network update in the forward propagation, a **gradient reversal layer**(GRL) is defined. During the forward propagation, GRL acts as an identity transform, which is $Q_\lambda(h) = h$, while during the back-propagation update, GRL takes the gradient from the subsequent level, multiplies it by $-\lambda$ and passes it to the preceding layer, that is $\frac{dQ_\lambda(h)}{dh} = -\lambda I$. where I is identity operator, λ is the hyper-parameter for the trade-off between tasks. The corresponding loss function is therefore represented as:

$$L(\theta_g, \theta_y, \theta_d) = \sum_{\substack{i=1,\cdots,N \\ d_i=0}} L_y(G_y(G_g(x_i;\theta_g);\theta_y), y_i) + \sum_{i=1,\cdots,N} L_d(G_d(Q_\lambda(G_g(x_i;\theta_g));\theta_d), d_i) \tag{3.6}$$

where d_i means the domain label; G_g means the feature generation processing of the shared encoder; and G_y and G_d are the category and domain label mapping functionals respectively. Training consists in minimizing the loss function using Stochastic Gradient Descent (SGD) update [2]. After the learning process, connected networks of shared-encoder and classifier can be used to categorize the test samples in the target domain.

3.1.3 Domain Adaptability in deeper networks

Inspired by the DANN network proposed by Ganin and Lempitsky [15], we build a relatively shallow network similar to theirs. To further test the domain adaptability in deeper networks, we compare the results on deeper networks with the same domain adaptation approach.

3.2 Domain adaptation with MMD loss

Like previous efforts, the proposed network is trained to find similar feature spaces of images from source and target domains. The resulting novel domain adaptation network model is

depicted in Figure 1.2. The shared encoder is a fine-tuned deep network with parameter θ_g without fully-connected (FC) layers whose , and the classifier is the remaining FC layers with parameter θ_y. No explicit domain adaptors are utilized. Task learning is based on the same idea of generating noise-invariant and discriminant features simultaneously by minimizing domain loss L_{domain} and the classifier loss $L_{classifier}$. And $L_{classifier}$ is defined as follows:

$$L_{classifier}(\theta_g, \theta_y) = \sum_{\substack{i=1,\cdots,N \\ d_i=0}} L_y(G_y(G_g(x_i;\theta_g);\theta_y), y_i) \tag{3.7}$$

We will introduce a different domain loss based on Mean Square Discrepancy (MMD), L_{domain}^{MMD}, which will be explained in next subsection.

The training consists in minimizing the loss functionn using SGD update. In contrast with all the shared-space component analysis available in the literature [3, 15, 29], we explicitly model the domain adaptation and classification tasks in separate steps, instead of jointing the two losses together to achieve both noise-invariant and discriminant network. Two-step training is utilized on the fine-tuned encoder. In each iteration, we train the shared encoder by backpropagating the domain loss gradient first, then we train the whole network by backpropagating the classifier loss gradient.

3.2.1 Optimization

Maximum Mean Discrepancy (MMD) loss [40] is another similarity metric to measure how close the feature space $\mathcal{H}_s$ and $\mathcal{H}_t$ are. It is computed with respect to a particular representation, $\phi(\cdot)$. In our case, we use a biased statistic, the linear combination of multiple kernel functions in the form of $\kappa(\cdot,\cdot)$ with various parameters, for the squared population MMD between extracted source and target features within the shared encoders:

$$L_{domain}^{MMD} = \frac{1}{N_s^2}\sum_{i,j=0}^{N_s}\kappa(h_{si}, h_{sj}) - \frac{2}{N_s N_t}\sum_{i,j=0}^{N_s,N_t}\kappa(h_{si}, h_{tj}) + \frac{1}{N_t^2}\sum_{i,j=0}^{N_t}\kappa(h_{ti}, h_{tj}) \tag{3.8}$$

where h_{si} and h_{ti} are the ith features of source samples and target samples respectively generated from the shared encoder, N_s and N_t are the source and target batch size. Classification loss $L_{classifier}$ is cross entropy loss calculated on the predicted result of source samples. $\kappa(\cdot,\cdot)$ is a PSD kernel function. $\kappa(x_i, x_j) = \sum_n exp\{-\frac{1}{2\sigma_n}\|x_i - x_j\|^2\}$, where σ_n is the standard deviation for our nth RBF kernel. Therefore the domain loss is

$$L_{domain}^{MMD}(\theta_g) = \frac{1}{N_s^2}\sum_{i,j=0}^{N_s}\kappa(G_g(x_{si};\theta_g), G_g(x_{sj};\theta_g)) - \frac{2}{N_s N_t}\sum_{i,j=0}^{N_s,N_t}\kappa(G_g(x_{si};\theta_g), G_g(x_{tj};\theta_g)) + \frac{1}{N_t^2}\sum_{i,j=0}^{N_t}\kappa(G_g(x_{ti};\theta_g), G_g(x_{tj};\theta_g)) \tag{3.9}$$

In summary, the training process of our proposed architecture is shown as follows, source dataset $D_S = \{(x_{S1}, y_{S1})..., (x_{Sn}, y_{Sn})\}$; target dataset $D_T = \{x_{T1}..., x_{Tm}\}$ without label information; and the classifier learning rate α_y, domain adapted rate α_d:

Algorithm 1 The training process of our proposed architecture

Input: $D_S = \{(x_{S1}, y_{S1})..., (x_{Sn}, y_{Sn})\}$, $D_T = \{x_{T1}..., x_{Tm}\}$, α_d, α_y

1: Initialize parameters θ_g, θ_y with fine-tuned weights
2: **while** not stop **do**
3: **for** each source-target mixed batch of size m_d **do**
4: Update θ_g by backpropageting gradient of L_{domain} :
$$\theta_g \leftarrow \theta_g - \alpha_d \nabla_{\theta_g} L^{MMD}_{domain}(\theta_g)$$
5: **end for**
6: **for** each source batch of size m_y **do**
7: Update $\theta = \theta_g \cup \theta_y$ by backpropageting gradient of $L_{classifier}$:
$$\theta \leftarrow \theta - \alpha_y \nabla_{\theta} L_{classifier}(\theta)$$
8: **end for**
9: **end while**

Output: Learnt prameters: $\hat{\theta} = \hat{\theta}_g \cup \hat{\theta}_y$

Training consists in alternately minimizing L_{domain} and $L_{classifer}$ using SGD update. After the learning, connected networks of shared-encoder and classifier can be used to categorize the test samples in target domain.

Chapter 4

Experiments and Results

In this chapter, we will show all the steps and settings of the experiments we have conducted. The results are also listed.

4.1 Datasets Preparation

We use two popular image classification datasets in the experiments, i.e., CIFAR-10 [25] and MIT Indoor Scene Recognition Dataset [39]. Not only the original datasets are used, but also the noisy version achieved by additive white Gaussian noise (AWGN) with different levels. To further evaluate the performance of our domain adaption networks, we also generate the correspondingly denoised version at each noise level using BM3D algorithm [8]. Noise level must be set as a pre-condition before implementing BM3D.

4.1.1 CIFAR-10 Dataset

The CIFAR-10 dataset consists of 60000 32×32 RGB images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test (validation) images. Test batch contains exactly 1000 randomly-selected images from each class. Training batches contain the remaining samples in random order.

The train-from-scratch accuracy for CIFAR-10 is 92.5% using ResNet-32 according tor [20]. For the input size requirement of our proposed network based on ResNet-50, we resize each image to size 224×224 by bilinear interpolation.

4.1.2 Indoor Scene Recognition Dataset

The database contains 67 Indoor categories and a total of 15620 images. The number of images varies across categories, but there are at least 100 images per category. We followed the official splitting to organize the training/validation set as [39]. All images are resized into 224×224. And for training dataset, there are around 80 images in each category.

Figure 4.1: Image samples of Indoor Scene Recognition Dataset

The test dataset has about 20 images in each class. The accuracy of Indoor-67 fine-tuning ResNet-50 pre-trained on ImageNet is 71.1% [32]. Samples of Indoor-67 is shown in Fig 4.1. We can see that these scene images are complicated with various fine objects.

Figure 4.2: Noisy ICIFAR-10 image samples, the images from 1st row to 5th row represent the images under the distortion of noise level 16, 25, 50, 90, 130 respectively.

4.1.3 Noisy Datasets

We add additive white Gaussian noise (AWGN) over 5 levels of distortion severity to both datasets. AWGN adds high frequency information to images and usually requires a smoothening filter to eliminate noise. We use a noise standard deviation $\sigma_n \in \{16, 25, 50, 90, 130\}$ to represent the distortion level. Comparison of the distorted images are shown in Fig 4.2. We can get a direct visual feeling that as the noise level increases, it becomes harder to recognize the details.

4.1.4 Denoised dataset via CBM3D

We also iapplied the denoised method CBM3D [8] to the noisy images. This algorithm get noisy image and the noise level as input and output the denoised version ofthe image. It can be summarized as four steps: *a*) find the image patches similar to a given image patch and group them in a 3D block *b*) operate a 3D linear transform of the 3D block. *c*) shrinkage of the transform spectrum coefficients. *d*) inverse 3D transformation. We can

Figure 4.3: The denoised image samples using CBM3D methods, from 1st row to 5th row represent the reconstructed images from the distortion of noise level 16, 25, 50, 90, 130 respectively.

see from Fig 4.3 that this algorithm reconstructs the distorted images very well. However, if we look into the results very carefully, we can see that when the noise level is very high, the denoised images suffer from blur and lose details. Take the last row for instance, the fine textures disappeared. We can hardly recognize the dog without the previous reference. This problem would be serious in scene classification task, where many of the objects in a scene are fine yet discriminant.

4.1.5 Data Split while Training

During training, images batches both from source and target domain are sent into the networks in fixed batch size. Half of the batch are the clean images with a random index set. The remaining half are consisted by those from noisy dataset. To mimic the real world case where noise level is unknown, we randomly selected noisy images among with random noise level and image index.

4.2 Experiments

We conducted three experiments on the investigation of Domain Adaption (DA) networks. First, we validated the domain adaptation can work in our case with a clean-noisy dataset adaptation. Second, we further explore this approach by comparing its performance on networks with different depth. At last, based on validation and restriction found in previous experiments, we design a DA network with high depth. Domain adversarial similarity loss achieved via Gradient Reversal Layer (GRL) is utilized in first two experiments, while Maximum Mean Discrepancy (MMD) loss is used in the last experiment.

4.2.1 Experiment 1: Validation with Shallow Network

We build a shallow domain adaptation network similar to Ganin and Lempitsky [15], intending to validate our hypothesis that domain adaptation networks can extract noise-invariant features and improve categorization performance on noisy images.

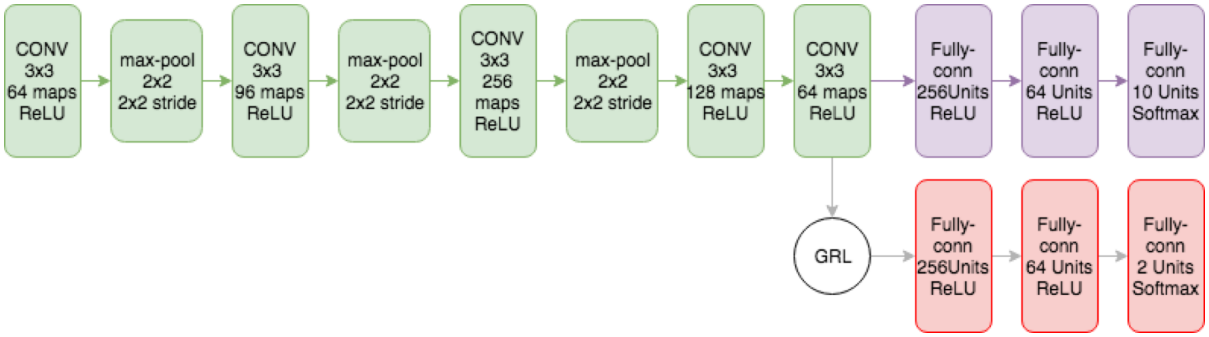


Figure 4.4: The DANN architecture for noisy-clean domain adaptation.

Training

A shallow domain adaptation network is composed of a 5-layer convolutional layers. Details of each layer are shown in Figure 4.4. As domain adaptor we add three fully connected layers ($x \rightarrow 256 \rightarrow 64 \rightarrow 2$) after the reversal layer. Classification loss L_y and domain loss L_d are both set as cross entropy loss. The source dataset is the original clean image samples and the target dataset is the noisy version at different levels.

Model is trained on 128-sized batches. All input samples are preprocessed by mean subtraction. Half of the batch is constituted by original CIFAR-10 dataset (with corresponding labels known) while the remaining are noisy CIFAR-10 (with random changed order and unknown labels). Only the feature maps of source dataset output from the fifth convolutional layer are sent to the classifier, while both feature maps (labeled as 0,1, respectively) are sent to the adaptor. Standard Generative Adversarial Networks (GANs) have multiple iterative steps in each main iteration, and follows the discrimination-generation order. Instead, we let our network focus more on learning discriminant features for classification task at early stage by gradually increasing the share domain loss out of the total loss. This is achieved by changing the adaptation factor λ from 0 to 1, as follows:

$$\lambda = \frac{2}{1 - exp(-\gamma \cdot 10)} - 1$$

Where γ is set to 10 in all and the number of iterations is 80,000 (around 205 epochs). In general, we observe a good correspondence between adaptation and errors. That is, adaptation is well learned when the source domain classification error is low as well as when the domain classifier error is around 50% (random prediction). The hyper parameter tuning is also non-trivial for domain adaptation network. If λ is set large at early stage, the learning of domain classifier will be dominant. While a very small λ could consequently suppress the learning of domain adaptation.

Results

The results are shown in Fig 4.5, the blue curve is the performance of this shallow network train from scratch on CIFAR-10 without domain adaptation. We can see that the classification accuracy drops rapidly for an increasing noise level. At noise level 130, it can almost predict nothing, with performance close to random prediction (10%). The red curve is the performance of the test dataset denoised by BM3D methods. Classification accuracy improves at all noise levels, which means that the denoising approach can to some extent reduce the noise influence. After implementing domain adaptation, the performance shown in black curve, has further improved compared to the BM3D predenoise method. Yet these results are not obvious when noise level is low, where denoising approach has nearly an equivalent performance.

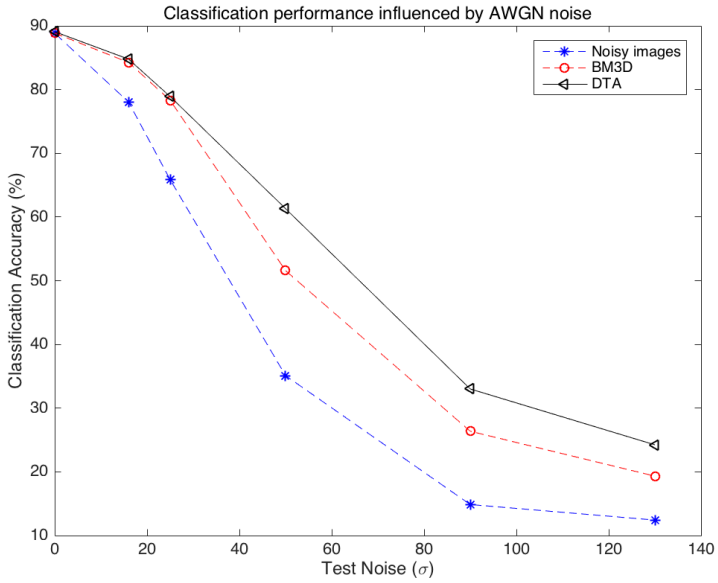


Figure 4.5: The Classification performance of Shallow DANN Network.

In a nutshell, domain adaptation on shallow CNN leads to an obvious improvement in classification performance in general when compared to preprocessing the noisy data using traditional denosing algorithms. However, there advantage is less clear at low noise level.

4.2.2 Experiment 2: Further explore DANN on deep networks

To compare the performance on various depth selection, we conducted DANN approach on ResNet-18 and ResNet-34. The number 18 or 34 refers to the actual number of layers in one network. In ResNet-18, each residual block consists of 2 convolutional layers, while ResNet-34 has 4 convolutional layers in every block.

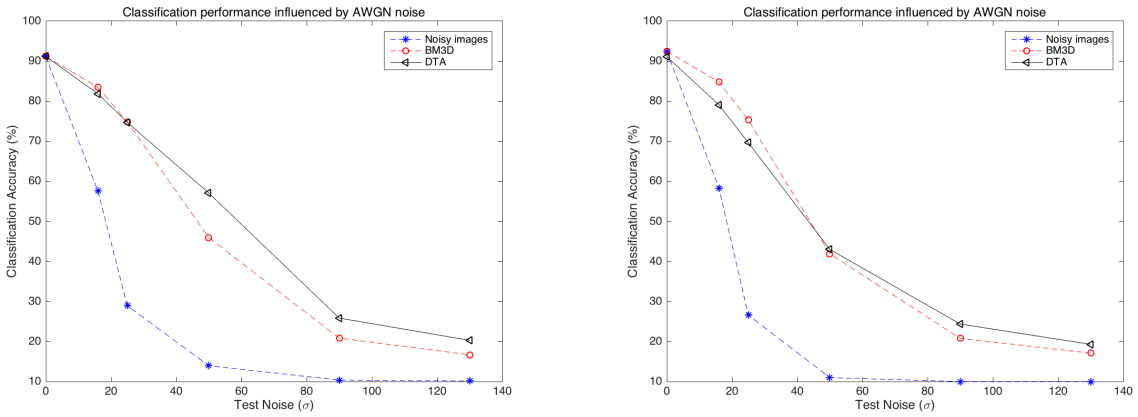
Training

We use fully-connected layers ($\mathfrak{x} \to 128 \to 10$) after the feature extractor. For domain classifier we add simpler fully-connect layers ($\mathfrak{x} \to 64 \to 2$). The number of filters of the first convolutional layer is 16. And for the residual blocks, number of filters are $\{16, 16, 32, 64\}$. Filter size is fixed 3×3.

The model is trained on 128-sized batches. Datasets are preprocessed by standardization. Data augmentation (random crop and flip) is also utilized during training. For test samples, only data standardization are processed. Half of the batch is constituted of the original CIFAR-10 dataset (with corresponding labels) while the remaining of the batch are noisy CIFAR-10 (randomly changed order and without labels). The learning rate for the label

classifier is 0.1 during the first 40k iterations, 0.01 between 40k to 60k iterations, and 0.001 after 60k iterations. Similarly, hyper parameter λ is set as $\lambda = \frac{0.4}{1-exp(-\gamma \cdot 10)} - 1$, where γ is set to 10 in all and the number of iterations is 80,000 (around 205 epochs).

Results



(a) Performance of ResNet-18 DANN

(b) Performance of ResNet-34 DANN

Figure 4.6: Classification accuracy of ResNet DANN. The influence of noise is similar to previous case. Prediction accuracy drops accordingly with noise level. BM3D methods can improve the performance. While as the network goes deeper, advantage of DANN vanishes.

Results are shown in Fig 4.6. Classification accuracy on deeper ResNet drops rapidly according to noise distortion level. When noise level is above 50, the classification network is not better than random guess. Our baseline, the denoised test dataset by BM3D methods, also contribute to performance improvements as expected at all noise levels. However, the DANN networks has vanishing advantages compared to BM3D, especially at low noise levels. In conclusion, as the network goes deeper, less strength left for DANN networks.

4.2.3 Experiment 3: Design deep DA network with MMD loss

In this section, we will train another domain adaptation architecture using Maximum Mean Discrepancy (MMD) similarity loss.

Fine-tuning

We started with the pre-trained ResNet-50 on ImageNet. The fully-connected layers are removed after trained weights have been loaded to the network. Then we add our task specific fully-connected layers on the top. We fine-tune this model on CIFAR-10 and Indoor-67, respectively. Similar to the procedure followed in [20], we resize the input

images into size 224×224. All input samples are standardized in the same as they have done for original network. Due to the limit of memory size, we set the batch size 24 and all the input samples are clean datasets with known labels to the network. Initial learning rate is set as 0.001, which is the minimum value of the original network. Stochastic Gradient Descent (SGD) optimizer is used with momentum. Learning rate is dropped by a factor of 10 every time the validation set accuracy is seen to plateau, until the training is finally terminated after 100 epochs. Data augmentation processing used during training is the same as that in [20]. We generated two variants of fine-tuned models for our two datasets, *a*) the fine-tuned model for CIFAR-10 with accuracy 94.5%, and *b*) the fine-tuned model for Indoor-67 dataset with accuracy 71.1%, same as the result in [32].

Training

The fine-tuned ResNet-50 without fully-connect layers is set as the shared encoder for samples from both domains. Due to the limit of memory size, we set batch size as 48. Datasets are preprocessed by standardization. Data augmentation (randomly cropped and flipped) is also utilized for training images. For test samples, only standardization are processed. Half of the input in each mini-batch are randomly chosen from source dataset and the remaining half is the noisy training samples distorted with a random distortion level (randomly selected from {16, 25, 50, 90, 130}). We choose the Adam optimizer [6], with learning rate $1e-4$ for CIFAR-10 and $1e-5$ for Indoor-67. Because for both CIFAR-10 and Indoor-67, the training samples are not enough compared with ImageNet, early stopping is utilized while training to avoid the over-fitting problem. If the average validation accuracy does not increase more than 0.5% within 4 epochs, the training will stop. Eventually the number of training epochs for CIFAR-10 is 20 and for Indoor-67 is 16.

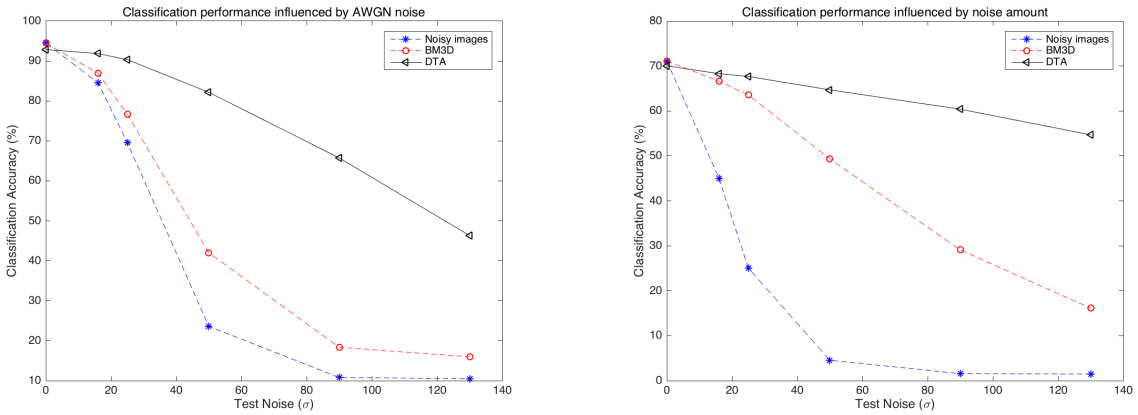
The training process for each iteration is described below:

1) activate the encoder for both source dataset and target dataset, calculate the MMD similarity loss $\mathtt{L}_{\mathtt{domain}}$ and update the shared encoder.

2) deactivate the target parts, calculate the classification loss $\mathtt{L}_{\mathtt{class}}$ and update the whole network.

Results

Fig 4.7 shows the results of evaluating the MMD approaches on CIFAR-10 and Indoor-67. The performance of a pre-trained network on distorted images is shown by the blue curves, the red curve show the prediction accuracy on BM3D denoised test samples. The noise influence is similar to previous networks. The preprocessing denoising approach improves the average classification rate of the network from 48.93% to 55.75% on CIFAR-10, and from 24.81% to 49.31% on Indoor-67. The proposed model further improves it to 78.22% on CIFAR-10 and 64.31% on Indoor-67. It has an obvious advantage over preprocessing



(a) Performance on CIFAR-10.

(b) Performance on Indoor-67.

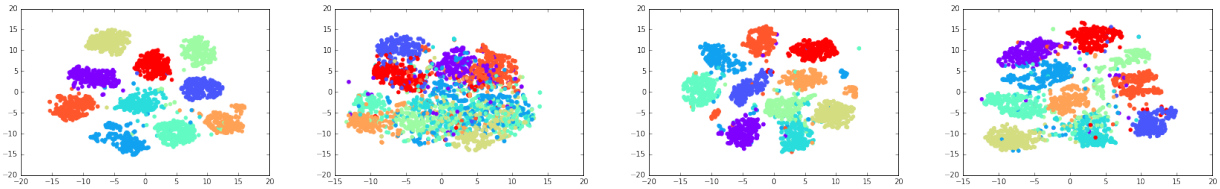
Figure 4.7: Classification accuracy on the proposed MMDA networks. The classification robustness is greatly increased compared to previous domain adaptation networks. The performance is acceptable even for high noise levels, i.e., both networks has more than 40% accuracy on noise level 130, which is rare in real practice. The classification error decrease around 2% on clean dataset because some discriminant features are domain-sensitive. Or in other words, trying to extract domain-invariant features would make these features ignored from learning.

using BM3D.

Moreover, the denoising processing with BM3D takes more than one week using a MacBook Pro with 8 GB 1867 MHz DDR3, and the noise-level of corresponding distorted images must be set as input before restoration. Thus, in real-world applications a good estimate of noise level is necessary. While the MMD similarity based domain adaptation network on pre-trained networks only takes around 2 hours and can achieve quite well performance on 12.0 GB TITAN X (Pascal) GPU without any knowledge about noise level. The domain-adapted network has a much more robust performance at various noise levels. This is our expectation for real engineering. However, one thing must be noticed that classification accuracy on clean images decreases about 2% during training in both cases. Some domain-sensitive features are ignored during domain-invariant features extracting.

Chapter 5

Discussion



(a) $\sigma = 0$ no domain adaptation

(b) $\sigma = 25$ no domain adaptation

(c) $\sigma = 0$ after MMD domain adaptation

(d) $\sigma = 25$ after MMD domain adaptation

Figure 5.1: Visualization of extracted feature distribution using t-SNE [31] of clean CIFAR-10 and noisy CIFAR-10 (with noise level 25) (a)The feature distribution of CIFAR-10 given by pre-trained ResNet-50 on ImageNet;(b)The feature distribution of noisy($\sigma = 25$) CIFAR-10 given by pre-trained ResNet-50 on ImageNet;(c)The feature distribution of CIFAR-10 after MMD loss based domain adapted ResNet-50;(d)The feature distribution of noisy($\sigma = 25$) CIFAR-10 after MMD loss based domain adapted ResNet-50

t-SNE [31] is a visualization technique that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. It is better than existing techniques at creating a single map that reveals structure at many different scales. We respectively visualize in Figure 5.1(a)-(d) the t-SNE embeddings of the features extracted by shared encoder of our proposed architecture. We can make the following observations: *1*) The feature distributions extracted on original ResNet-50 shown in Figure 5.1(a)-(b) show that the target feature are not well discriminated by the source encoder. Hence different feature distribution will be generated even when the source and target dataset are identically distributed. *2*) The encodings distribution shown in Figure 5.1(c)-(d) is trained by the proposed MMD loss basedb approach. It shows that the target features are discriminated better(larger class-to-class distances), which suggests that the proposed domain adaptation network is reasonable in noisy image classification task.

From last chapter, we see that as the network goes deeper, feature adaptability of DANN drops significantly. It is a commonly accepted truth that features extracted from deep neural networks have a transition from general to specific along the network. As confirmed in the work by Yosinski*et al.*[55], the feature transferability drops significantly in higher layers. However, the features in deeper layers are more discriminant. Task-specific features are hard to be transferred to new tasks on higher layers of a DANN network. Instead of training two tasks (domain adaptation, classification) simultaneously with single update at each iteration, we use a two-step strategy, each step for one task. Result comparison between DANN network and MMD loss based domain adaptation shows the advantage of the proposed approach in the clean-noisy adaptation task. Moreover, with a pre-trained network, it is very time efficient to do domain adaptation with the two steps training. In our experiment, training on both datasets takes no more than 20 epochs. Therefore, our proposed approach can efficiently be extended to deeper networks, solving more complicated tasks.

However, the classification performance on clean images slightly decrease after domain adaptation. This is some discriminant features are highly domain-sensitive and will be ignored during domain-invariant features extracting. Another interesting phenomenon is that our proposed algorithm works particularly well on very deep networks yet not good on simple networks. A stable transferability is highly determined by network learning capability intrinsically. We also conduct our adaptation strategy on Inception-v3 [48], Inception-v4 [46], both show improved classification performance at all distorted levels. While during the domain adaptation training on simple networks (like simplified ResNet-18 and ResNet-34 for CIFAR-10), the performance on clean images drops rapidly when the categorization accuracy on noisy samples increase, which is not acceptable.

Chapter 6

Conclusion

This thesis compared the performances of DANN networks on the clean-noisy dataset adaptation and proposed a novel approach to semi-supervised domain adaptation in deep networks. Through this work, a robust network is achieved without adding extra parameters. The two-step MMD similarity based domain adaptation can improve the prediction accuracy of noisy images even under serious distortion. It has a good robustness compared with previous domain adaptation approaches, especially when implemented in very deep networks. It reflects the generalization capability of convolutional neural networks. This proposed network greatly overcomes the difficulty of deeper feature transfer between clean and distorted datasets. In the future, more semantic tasks on distorted images can be explored, especially the low level tasks like image segmentation and colorization. Furthermore, more types of distortion can be taken into account. A good model to mimic realistic distortion will be more task specific, especially for the scene images collected by surveillance cameras with dynamic working environments.

Bibliography

[1] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. *Computer Vision Systems*, pages 66–75, 2008. 1

[2] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 13

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016. 14

[4] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2392–2399. IEEE, 2012. 6

[5] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):18, 2012. 8

[6] Trishul M Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *OSDI*, volume 14, pages 571–582, 2014. 23

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1

[8] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 6, 16, 18

[9] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009. 8

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2

[11] Steven Diamond, Vincent Sitzmann, Stephen Boyd, Gordon Wetzstein, and Felix Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487*, 2017. 7

[12] Nithish Divakar and R Venkatesh Babu. Image denoising via cnns: An adversarial approach. *arXiv preprint arXiv:1708.00159*, 2017. 6

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 6

[14] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 6

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 8, 13, 14, 19

[16] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010. 2

[17] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely–laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3555–3561. IEEE, 2010. 2

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 5

[19] Stephan Halbritter. Generative adversarial networks. 2017. 13

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9, 16, 22, 23

[21] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 1

[22] Laurent Itti, Geraint Rees, and John K Tsotsos. *Neurobiology of attention*. Academic Press, 2005. 1

[23] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009. 6

[24] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003. 6

[25] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 16

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 9

[27] Michael JV Leach, Ed P Sparks, and Neil M Robertson. Contextual anomaly detection in crowded surveillance scenes. *Pattern Recognition Letters*, 44:71–79, 2014. 1

[28] Huibin Li and Feng Liu. Image denoising via sparse and redundant representations over learned dictionaries in wavelet domain. In *Image and Graphics, 2009. ICIG'09. Fifth International Conference on*, pages 754–758. IEEE, 2009. 6

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 7, 8, 14

[30] Florian Luisier, Thierry Blu, and Michael Unser. A new sure approach to image denoising: Interscale orthonormal wavelet thresholding. *IEEE Transactions on image processing*, 16(3):593–606, 2007. 6

[31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 25

[32] Ammar Mahmood, Mohammed Bennamoun, Senjian An, and Ferdous Sohel. Resfeats: Residual network based features for image classification. *arXiv preprint arXiv:1611.06656*, 2016. 17, 23

[33] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009. 6

[34] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008. 6

[35] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 1

[36] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 6

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 7

[38] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image processing*, 12(11):1338–1351, 2003. 6

[39] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009. 16

[40] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, pages 2263–2291, 2013. 14

[41] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 5, 7

[42] Guofeng Sheng, Wen Yang, Tao Xu, and Hong Sun. High-resolution satellite scene classification using a sparse coding based multiple feature combination. *International journal of remote sensing*, 33(8):2395–2412, 2012. 2

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 9

[44] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Markus Eich, Ben Upcroft, and Michael Milford. Slam–quo vadis? in support of object oriented and semantic slam. 1

[45] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4297–4304. IEEE, 2015. 2

[46] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 9, 26

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 9

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 26

[49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 5

[50] Jiqing Wu, Radu Timofte, Zhiwu Huang, and Luc Van Gool. On the relation between color image denoising and classification. *arXiv preprint arXiv:1704.01372*, 2017. 7

[51] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012. 7

[52] Jiakun Xu, Kun Zhang, Mingyao Xu, and Zhigang Zhou. An adaptive threshold method for image denoising based on wavelet domain. In *Proc. of SPIE Vol*, volume 7495, pages 74954M–1, 2009. 6

[53] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009. 2

[54] Qingsong Yang, Pingkun Yan, Mannudeep K Kalra, and Ge Wang. Ct image denoising with perceptive deep neural networks. *arXiv preprint arXiv:1702.07019*, 2017. 7

[55] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 7, 26

[56] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 9

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[58] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 2