A Study on Localized Rotation for Cross Domain Learning

Abstract

In the last ten years, due to the huge increase of the computing power and the massive amount of available data, the statistic-based artificial intelligence has shown a comparable performance with respect to human being. The machine learning and deep learning are being known by more and more people, and the techniques based on machine learning and deep learning have achieved a great success in many fields, such as natural language processing and compute vision. This thesis will focus on the field of computer vision, in more detail, the image classification task. Since in 2012 AlexNet has won the ImageNet contest, the CNN-based model has become the main method to solve image-based task, and indeed it performs very well, but there are still some limitations and deficiencies, such as the lack of good labeled data, and the model lacks the ability of domain generalization. In this thesis, we try to introduce a model to solve these two problems, it is based on an existing model, in which it has used a multi-task learning structure that do both standard image classification task and the standard rotation task at the same time. Our contribution is that we propose a new task to substitute the standard rotation task, we call it the localized rotation task. We evaluate our method on the datasets of PACS with the standard multi-source domain generalization setting and show the training process and final result. We compare its performance with both the deep all baseline and the other methods, and we can draw the conclusion that our method is effective and better in this scenario.

Key words: Deep Learning, Image Classification, Multi-task Learning, Self-supervised Learning, Localized Rotation

装 订 线

Content

装

订

线

摘要

在过去的十年中，由于计算能力的巨大提高和大量可用数据的出现，基于统计学的人工智能在很多任务中已经表现出了可以和人类一较高下的性能。机器学习和深度学习已被越来越多的人所熟知，基于机器学习和深度学习的技术在自然语言处理和计算视觉等许多领域取得了巨大的成功。本文将聚焦于计算机视觉领域中的图像分类任务进行讨论。自从 2012 年 AlexNet 赢得 ImageNet 竞赛以来，基于深度卷积神经网络的模型已成为解决图像任务的主要模型，这些模型在图像任务上确实表现很好，但是仍然存在一些局限性和不足之处，其中一个不足是，这些模型的训练需要大量标记正确的训练数据，然而实际中却很难获取足够的标记正确的数据，另外一个不足是：这些模型往往只能工作在特定场景，解决特定问题，缺乏泛化迁移的能力。在本文中，我们将介绍一个模型来解决上述两个问题，该模型使用了多任务学习结构，该模型同时优化标准的图像分类任务和一个自监督学习任务。我们的创新点在于：（1）我们提出了一个新的自监督学习任务来代替现有模型中的自监督学习任务，我们称其为局部旋转任务。（2）我们提出了几种分组策略来优化我们的模型。为了验证我们的模型的有效性，我们设计了标准的多源域泛化实验，并在 PACS 数据集上评估我们的模型，我们展示了详细的训练过程和最终结果。最后的实验结果显示，我们的模型是有效的，且比当前的其它模型有更高的泛化能力。

关键词：深度学习，图像分类，多任务学习，自监督学习，局部旋转任务

装 订 线

1 综述

1.1 机器学习

学习是一个获取知识，信息和经验的过程。从出生到现在我们每天都在学习，当我们还是个孩子的时候，我们学习走路，说话，吃饭……，当我们在学校时，我们学习解决数学问题，学习写作文，学习世界的原理；当我们在公司工作时，我们学习如何处理与同事、上司的关系，我们学习如何控制自己的情绪，我们学习如何做重要的决定。对我们来说，学习是一件稀松平常的事。动物也可以学习，老虎学习狩猎，鸟儿学习飞行，鱼儿学习游泳。但我们或许没想到机器也可以像人类或动物一样学习。

在过去的几十年中，机器学习被越来越多的人所了解，我们可以把机器学习的过程理解成机器进化的过程，在这个过程中，机器通过分析大量的数据来获取知识和经验，然后自动改进和优化它的结构和程序，以便更好地完成任务。我们以一个例子来展示机器学习的过程，想象一下，现在机器是一个什么都不知道的孩子，现在我们要教这个孩子做图像分类任务，我们唯一需要做的就是：我们取出一个图像并显示给孩子，然后问他是什么，孩子给出答案，然后我们告诉他正确的答案，然后我们取出下一张图像并重复这个过程…，在此过程中，孩子将会学到一个规律：苹果是一个顶部有凹槽的圆形物体。然后当他再次看到具有这种形状的物体时，他就会知道这是一个苹果。

换句话说，我们不需要手动更改机器的代码结构以及各种变量的值，我们只需要给它输入数据，然后它就可以自行更新它的记忆。机器学习的优势在于，它可以利用大量数据并在找到其中的规律性。它的劣势在于：如果输入数据是错误的或者说我们无法提供足够的输入数据，那么训练出来的模型就无法提供令人满意的性能。在大多数情况下，训练数据的质量和数量仍然是机器学习性能的关键因素。

1.2 深度学习

深度学习是机器学习算法的一个分支，它是如今机器学习中最前沿的研究领域。传统机器学习算法与深度学习之间的区别在于对数据的利用能力，当训练数据量相对较小时，深度学习并没有显示出比传统机器学习算法更好的性能，但是当我们有更多数据时，传统机器学习算法的性能就出现了饱和，而深度学习则显示出了更强的利用数据的能力。

深度学习已经在许多领域展示出了与人类相当甚至更好的性能，例如计算机视觉，自然语言处理和自动语音识别领域。深度学习的核心实际上是人工神经网络，它就像一个人造大脑一样。

人的大脑皮层中大约有 140 亿个神经元，是最复杂的神经系统之一。神经元则是神经系统的核心，它最重要的功能组件是树突，细胞体和轴突。树突是一个感受器，接收来自其他神经元或环境的信号输入，细胞体收集并处理所有输入，然后决定是否激活输出并将信号发送给下一个神经元，一旦细胞体决定发送信号，轴突将负责信号的传送。这就是真实神经元最基本的机制，如果我们能模拟这个过程，那么我们就有可能建立一个人造的大脑。实际上科学家们确实在人工神

经网络中模拟了这个过程。在人工神经网络中，每个计算单元就像一个神经元，它接受一组输入，然后做矩阵计算，然后将结果输入激活函数，以决定激活这个计算单元。

但尽管人工神经网络和真正的神经网络结构类似，但它仍然存在一些不足。一方面，大脑中的神经元要多于人工网络中的神经元。虽然我们也可以通过增加人工神经网络的变量和层数时来增加神经元数量，但更多的神经元同时也意味着更高的计算功耗和更长的计算时间。

另一方面，实际的神经元要比人工神经元复杂得多，神经元里仍然有很多我们尚不清楚的机制和现象，例如信号在真实神经元之间传递时会受到许多原因的影响从而被随机地增强或削弱，但是，在人工神经元中，我们只有一个简单的静态激活函数。再例如一个真实的神经元可以与许多其他神经元连接，而在人工神经网络中，人工神经元之间的连接结构更多是一对一的。

1.3 图像分类和深度卷积神经网络

图像分类是计算机视觉领域中最基本的任务之一，我们通常使用它作为测试任务来评估和比较不同模型的性能。在本文中，我们只讨论物种级别的图像分类任务，换句话说，我们只需要根据图像所属的物种对图像进行分类，例如，给出一张图像，我们的任务仅仅在于它是否是一张猫，狗或是企鹅的图像。

2012 年，AlexNet 赢得了 ILSVRC 大赛冠军，从此 AlexNet 便被认为是深度卷积神经网络历史上的一个里程碑，其性能要比其它同时代的传统机器学习算法好得多。从那时起，深度卷积神经网络就成为了图像任务的最常用模型。

实际上，深度卷积神经网络模型可以分为两个部分，第一个部分是特征提取器，其功能是提取图像特征，降低输入的原始数据的维度。另一部分是完全连接的层组成的分类器。换句话说，完全连接层组成的分类器根据特征提取器提取到的图像特征给出分类结果。

1.4 图像领域泛化

图像领域表示该图像像素分布的特征。可以理解为该图像的样式或风格，不同的图像领域之间存在域偏移。让我们更详细地说明什么是图像领域。不同的图像风格是不同的领域，例如，照片样式的图像和卡通样式的图像就是不同的图像领域，因为照片样式的图像通常更逼真，而卡通样式的图像则相对更简单。图像拍摄条件也可以导致不同的图像领域，例如，在良好的光照条件下拍摄的图像和在较差的光照条件下拍摄的图像属于不同的图像领域，良好光线条件下拍摄的图像像素偏白，较差光线条件下拍摄的图像像素偏暗。图像拍摄时间也可以导致不同的图像领域，例如，在夏季拍摄的风景图像和冬季拍摄的风景图像存在可以认为是不同的两个领域，夏天图片可能会有更多的绿色，而冬天图片会有更多的白色。

图像领域泛化是我们非常关心的一个问题，在本文中它指的是：我们用来自多个领域的数据来训练模型，我们称训练数据所属的领域为源领域，我们将训练好的模型用于一个新的图像领域，我们将此领域称为目标领域。换句话说，我们用源领域的图像训练模型，但我们希望训练好的模型可以用在目标领域里。领域泛化问题如此重要的原因在于：有时我们无法获取目标领域的数据来训练模型，但我们却希望模型能工作在目标领域中。例如，现在我们仅有白天拍摄的图像可以

用于训练模型，如果该模型没有领域泛化能力，那么该模型将只知道如何对白天拍摄的图像进行分类，而不知道如何处理晚上拍摄的照片。

1.5 研究现状

如前所述，自 2012 年 AlexNet 赢得 ILSVRC 冠军以来，卷积神经网络已成为解决图像任务（例如图像分类，对象检测和语义分割）的最常用模型。在本文中我们只讨论图像分类任务，2012 年以后 ILSVRC 的冠军都使用了基于 CNN 的模型，例如 ZFNet[3]，VGGNet[4]，GoogLeNet[5]，ResNet[6]，和 SENet[7]，其中在 2017 年 SENet 将 top-5 测试错误率降低到了 2.25%。由于现有模型在解决普通图像分类任务方面已经超过了人类，提升空间很小，因此科学家现在针对的是更困难的问题，即图像领域泛化问题。当前一些现有的方法有：[8]尝试从多个源域中学习图像更本质的特征表示，它首先定义一个映射函数，然后使用这个映射函数来将不同源域的数据映射到一个新的领域，最后优化模型使得映射后的数据的差异性最小。[9]提出了一种多任务学习结构，其中模型同时优化标准的图像分类任务和拼图任务，拼图任务起到了正则化的作用，并帮助模型在不同领域间进行泛化。[10]提出了一种基于分辨不同类型的方法，该方法提出了一种新的自监督任务，并训练模型来对原始图像，旋转图像，扭曲图像，局部修复图像进行分类，以便模型可以学习更好的特征表示。[11]提出了一种学习原始图像和后图像的共同特征的方法，其核心思想是原始图像和后图像的应该具有相同的 BoW。

1.6 当前研究的不足存在的问题

基于 CNN 的深度模型在解决图像分类任务方面已经取得了巨大的成功，但是，仍然存在一些不足和局限性，这里我们关注两个问题，一个是缺乏好的标签数据，另一个是模型缺乏领域泛化能力。

1.6.1 缺乏好的标签数据

基于 CNN 的深度神经网络在处理 2D 数据方面具有非常强大的能力，它能完美地表达输入图像及其相应标签之间的非线性关系。但是现有的最优模型使用的算法主要是有监督学习。有监督学习是一种利用标记数据的机器学习算法，数据的数量和质量是影响使用有监督学习算法训练的模型的性能的最重要因素。确切地说，一方面，如果我们没有给模型提供足够的标签数据，则该模型将无法学习到数据的完整分布规律，通常情况下越大的网络所需的训练数据就越多。另一方面，标记的数据需要是正确的。这一点很容易理解，如果我们将错误的数据输入到模型中，那么它将学习到错误的数据分布，从而导致性能下降。但是，实际上标记数据非常昂贵且容易出错，通常情况是我们无法获得足够的好的标记数据来训练模型。

1.6.2 缺乏领域泛化能力

正如我们在背景部分中提到的那样，如果模型缺乏领域泛化能力，则该模型可能会在实际情况下无法正常工作，因为实际情况下的数据可能与训练数据属于不同的领域。没有域泛化能力的模型不能被认为是好的模型，因为它只能处理特定领域中的数据，当数据领域发生变化时，它就

没法工作。领域泛化能力是本文关注的另一个问题。

1.7 本文结构

在第一章中，我们介绍了一些有关机器学习，深度学习，图像分类，CNN 和数据领域的背景知识。然后说明了图像分类研究及提高模型的领域泛化能力的现状和意义。此外，我们提出了图像分类中存在的两个问题，即缺乏好的标签数据及模型缺乏领域泛化能力。

在第二章中，我们展示了模型中使用的主要算法和学习策略的原理，这一章是模型的理论基础。其中我们主要介绍了多任务学习，监督学习，无监督学习，自我监督学习和迁移学习。

在第三章中，我们详细介绍了我们的模型。模型的概览是：我们使用多任务[11]学习结构，其中我们同时优化两个任务，其中一个任务是使用监督学习算法的标准图像分类任务，并且另一个是使用自我监督学习算法的局部旋转任务。该模型的优点是可以通过自监督学习任务来利用大量可用的未标记图像，多任务学习结构可以帮助提高领域泛化能力。

在第四章中，我们展示了我们的实验设计细节及实验结果。我们使用 PACS 数据集来评估我们的模型，实验是标准的多源域泛化设置。最后我们展示出实验结果并进行讨论。

在五章中，我们介绍了我们的结论，疑问以及未来的工作。

装 订 线

2 算法与策略

本部分是最重要的理论基础，我们将介绍主要的学习算法和学习策略。首先我们将介绍多任务学习结构，这是我们模型的整体结构。其次我们将介绍监督学习，无监督学习和自监督学习算法。除此之外，我们将介绍在模型层面的迁移学习策略。

2.1 多任务学习

多任务学习的核心思想是该模型尝试同时优化多个任务。让我们先看一个在日常生活中进行多任务学习的示例，当我们学习一种新语言时，我们会同时练习听，说，读，写技能，因为这样可以全面提高我们的整体水平。这些任务实际上是不同的任务，但它们确实可以互相帮助。原因是这些任务基于相同的基础，也就是我们对单词的理解，所有这些任务都可以提高对单词的理解，因此它们可以互相帮助。

实际上，事实证明，多任务学习是提高模型的泛化能力的有效学习策略。该模型的具体实现是我们尝试同时优化一个主任务和一个副任务，每个任务都有一个损失函数。它要求模型算法要同时在主任务和副任务上表现良好，如果模型仅关注主任务，则将对模型进行惩罚，相比于标准正则化规则而言，这是一个更好的惩罚规则。

在特定情况下我们想对不同的任务给予不同的关注度，我们想在我们更关心的那些任务上花费更多的力量和力量。例如，写作测试恰好是在明天早上，那么我们通常希望在今晚进行更多的写作测试，这意味着在这种情况下，我们要更加关注写作任务。换句话说，我们需要在多任务学习算法中控制每个任务的权重，为此，我们只需要为各个损失函数分配不同的权重，因为任务和损失函数是一一对应的。

2.2 有监督学习

有监督学习的主要特征是它需要标签数据来训练模型。标签代表着数据的属性，例如，给出一张狗的图像，则狗是其标签。数据的标签就像模型的监督者一样，它教模型区分不同数据的差异。

监督学习通常用于解决分类和回归任务。分类任务是本文的重点，例如，对于标准图像分类任务来说，模型的任务便是预测给定图像的每个类别。更准确地说，给出一个图像，模型将处理该图像并告诉我们这样的信息：该图像有概率为 80％是狗，2％是猫，18％是人。

回归任务是另一种有监督的学习任务，它接受两组数据，可以将它们视为输入 x 和输出 y，任务便是是找到 x 和 y 之间的关系，模型需要根据输入 x 预测 y。

有监督学习在数据科学领域取得了巨大的成功，但是仍然存在一些缺陷和局限性：首先，它需要大量的数据，没有足够的训练数据它就无法有令人满意的表现。其次，训练数据的质量非常重要，如果数据标签错误，则模型会混乱并失败。最后，它丢弃了一些可能有用的信息，更确切地说，该模型仅保留了图像的类别信息，而不保留整个图像，因此损失了很多信息，例如格式，颜色，样式，模式等。

2.3 无监督学习

无监督学习是另一种常用的机器学习算法，它与有监督学习算法不同，它不需要注释数据，通常用于解决聚类任务和关联分析。 聚类任务是将各自没有标签的输入数据进行分组，算法必须自己探索数据结构，属于同一组的数据应该彼此相似，而与属于其他组的数据不相似。例如，其中一个聚类分析在市场分析上的应用是根据客户的点击行为，购物历史，浏览行为对他们进行分组，这样商家就可以根据客户购买行为预测其它同组客户想要购买的商品然后做出推荐。关联分析则是找到数据之间的关系，一个例子是商家将相关商品放在一起，这样消费者在购买了一样东西后便可以马上在附近购买到相关的东西，不仅给消费者提供了便利，同时可以提醒消费者可能需要购买的东西，商家便可以出售更多商品。

无监督学习算法具有许多优点。首先相比于有监督学习算法，它可以使用大量廉价的未注释数据，同时它可以避免潜在的错误标记。其次由于该模型直接从原始输入数据中探索信息，因此它可以为模型提供完整的信息而不会丢失任何信息。但是，它仍然存在一些缺点。首先，它不能给出准确而有意义的结果。例如，在聚类任务中，我们可以对数据进行分组，但是我们不知道分组的真正含义，只知道此数据属于该组，我们可以根据许多规则来解释该组的意义。其次，与无监督学习任务相比，无监督任务的训练相对更加困难和耗时。直观的解释是在监督学习中，有老师在教模型，而在无监督学习中，模型是在自己探索数据的结构，因此更加耗时且困难。

2.4 自监督学习

自监督学习既可以被视为一种无监督学习也可以被视为一种有监督学习方法，因为它使用无注释数据，训练过程中却存在标签，只不过标签是在训练过程中自己产生的。

例如标准旋转任务是一种自监督学习任务，其想法是，我们将图像随机旋转 0、90、180 或 270 度，然后训练模型来识别图像的旋转角度。在此过程中，我们需要的只是原始图像，没有任何人为注释的标签，并且该图像会从自身生成伪标签，即图像的旋转角度，由此我们便可以将此任务规范成一个为分类任务。

另一个自我监督的学习任务是解决拼图游戏，其中我们根据特定的序列对图片各部分进行混洗，该序列便是图像的伪标签，模型需要做的就是识别给定洗过的图像的序列。

自监督学习方法的优点是它可以利用未标记的数据并自己生成伪标签，从而利用未标记数据产生有监督学习的效果。

2.5 迁移学习

迁移学习是一种机器学习技术，旨在将从一项任务获得的知识转移到另一项相关任务。就像我们尝试重用现有知识，而不是从零开始学习新事物一样。例如，如果我们学会了一道数学问题，那么通常我们便可以快速解决类似的问题。

在机器学习中，变量的值就是知识，我们要做的就是将这些值存储在硬盘中，并在我们想重用它们时加载它们。如果我们想从头开始训练模型，那么我们将随机初始化模型的变量，但是带来的问题便是，训练模型将花费更多的时间。一个常用的范例是我们预先训练一个模型来解决

ImageNet [12]上的图像分类任务，ImageNet 是一个很重要的图像数据集，目前它已收集了14,197,122 张图像。然后我们创建一个新模型并使用预先训练的模型对其进行初始化，然后我们可以在其之上添加新的全连接层，并对其进行二次训练以解决我们实际的任务。在我们的论文中，我们使用了迁移学习技巧来缩短模型训练时间。

3 方法

在 3.1 节中，我们将概述模型的结构和工作流程。在 3.2 和 3.3 节中，我们将介绍模型试图完成的两个任务的细节。在 3.4 和 3.5 节中，我们将介绍模型超参数和整体损失函数。

3.1 模型概述

我们的模型旨在解决两个问题，即缺少良好的标记图像和缺乏领域泛化能力。对于第一个问题，我们的解决方案是使用自监督学习算法，该算法可以利用大量可用的未标记图像。关于第二个问题，我们的解决方案是使用多任务训练结构，该结构包含两个任务，这两个任务共享同一个图像特征提取器，其中一个任务是标准有监督图像分类任务，另一个是我们提出的局部旋转任务。使用多任务学习算法的好处在于它可以帮助模型在多个任务之间共享知识，提供正则化效果。

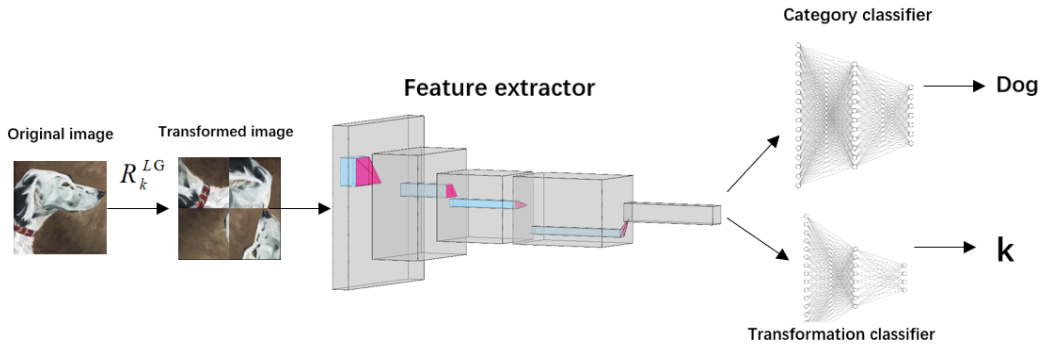

图 5.1 工作流程示意图

如图 5.1 所示，首先我们取出一张图像，并将其平均分为四个部分。我们定义了 25 种局部旋转，详细信息我们在 5.3 节中介绍。我们为每个变换分配了一个标签，此外，我们根据不同的分组策略对这些进行了分组，并为每个变换分配了一个组标签，属于同一组的变换将具有相同的组标签。

然后，我们从总共 25 个中选择一个变换并将其应用于图像以获取后的图像。此后的图像有两个标签，一个是类别标签，例如狗，猫，另一个是标签或所属组标签。在我们的模型中，我们有一个共用的特征提取器。在特征提取器的后面，我们有两个由完全连接层组成的分类器。我们将变换后的图像输入到模型中以解决两个任务：标准图像分类任务和局部旋转任务。更详细地，类别分类器解决标准图像分类任务，局部旋转分类器解决局部旋转任务。

3.2 标准的图像分类任务

在这一部分中，我们将介绍标准的图像分类任务。假设我们有 N^S 个源域，标准分类任务需要做的就是依次从各个源域中取出图片然后用特征提取器提取特征，然后用类别分类器进行处理，最后取预测概率最高得类别作为最终预测结果。

3.3 局部旋转任务

3.3.1 全局旋转和局部旋转

局部旋转任务的过程是：首先我们将原始图像分成 4 个部分，然后将整个图像旋转 0、90、180 或 270 度，我们将此步骤定义为全局旋转，下一步是我们再次对每个局部部分进行旋转，我们将此步骤定义为局部旋转。之所以将我们的方法称为局部旋转，是因为：它既包含全局旋转又包含局部旋转，而标准旋转方法仅包含全局旋转，我们想强调我们的方法与标准旋转方法之间的区别在于我们得方法中包含局部旋转。

3. 3. 2 减少变换种类

根据定义，我们有四种全局旋转。我们有四个局部部分，对于每个部分我们都有四个旋转角度可供选择。因此我们有 4 种全局旋转和 256 种局部旋转，因此总共可以有 4 * 256 = 1024 种。但是 1024 种意味着 1024 种类别，超维空间中类别的数量越多，标签之间的距离越短，因此模型很难区分它们。为了简化模型的任务，我们要减少数量。

首先我们做的第一个优化是：我们只选择图像的两个部分进行局部旋转，并且这两个选定的部分必须在对角线上，因此我们只有两个选择，分别是部分 1、4 或部分 2、3。如果我们选择对零件 1、4 进行局部旋转，则零件 2、3 将保持不变，反之亦然。因此，现在我们可以将总的数量减少到 4 * 2 * 4 * 4 = 128，其中第一个'4'表示全局旋转，第一个'2'表示在选择局部时有两个选择进行局部旋转，最后两个“ 4”表示，对于每个选定的局部，我们有四种旋转角度，分别为 0、90、180 或 270 度。

我们所做的第二个优化是：局部旋转可以旋转四种角度，分别为 0、90、180 或 270 度，但是，当旋转度为 0 时，实际上不发生任何局部。我们希望后的图像必须经过某种局部旋转。换句话说，在进行局部旋转时，我们不喜欢选择 0 度，因为这意味着不会发生任何局部。因此，我们强制局部旋转度不能为 0，这意味着我们只能选择将选定的 2 个局部旋转 90 度，180 度，270 度，而不是 0 度。因此，现在只有 4 * 2 * 3 * 3 = 128 种变换，其中'4'和'2'的含义不变，两个'3'表示对于每个选定的局部，现在我们只能将其旋转 90、180 或 270 度，不旋转 0 度。

我们还有第三次优化：如上所述，我们选择了两个零件进行局部旋转，现在我们添加了一个限制条件，那就是所选的两个零件必须旋转完全相同的角度。因此，现在总共只有 4 * 2 * 3 + 1 = 25 种。

3.3.3 损失函数

给出一张图片 $x^s_{i,j}$，首先我们从 $\{R^{LG}_w\}^{25}_{w=1}$ 种挑选一个变换 R^{LG}_w 用于处理图片：

$$u^s_{i,j,w} = R^{LG}_w(x^s_{i,j}) \tag{3.1}$$

然后我们用用特征提取器以及分类器处理可得：

$$\log it_R = h(u^s_{i,j,w} \mid \beta_f, \beta_r) \tag{3.2}$$

最后的损失函数为:

$$loss = L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), w) \tag{3.3}$$

其中 L_{ce} 是交叉损失熵函数。

3.3.4 分组策略

上面我们已经讨论了局部旋转任务的过程，在标准设置中我们有 25 种变换，因此有 25 种标签可以预测。但是实际上我们不需要预测所有 25 个标签，我们可以将这 25 种变换分为几组并分别给它们分组标签，深度模型需要做的只是预测分组标签，而不是预测具体变换标签。换句话说，我们可以根据不同的分组协议使用不同数量的标签进行预测。在这一部分中，我们将介绍几种协议来划分这 25 个变换。

（A）按全局变换分组

我们提出的第一种分组规则是将它们按全局旋转度进行分组。我们将所有具有相同全局旋转度的放在具有相同组标签的同一组中。在这种情况下，我们将有 5 个组标签。

（B）按局部变换分组

除了全局旋转度之外，每个还具有局部旋转度。因此，我们也可以将这些按其局部旋转度进行分组，可以将具有相同局部旋转度的变换视为同一组。在这种情况下，我们将有 4 个组标签。

（C）按局部旋转的部分分组

除了我们上面提出的两个分组协议之外，我们还可以根据选择应用局部旋转的两个部分对这些进行分组。由于就所选部分而言，我们只有两个组合，分别是部分 1、4 或 2、3，所以我们现在只有 3 个组。

3.4 超参数

如前所述，我们的模型使用多任务学习策略，并且同时执行标准图像分类任务和局部旋转任务。在这一部分中，我们将更详细地显示整个过程。

我们的模型中有两个超参数，我们定义的第一个超参数是原始图像的比例，它是指图像保持不变的概率，我们用 β 表示它。例如，如果我们设置 β= 0.6，那么我们有 60%的概率选择第一种变换，换句话说 β= 0.6 将近 60％的图像变换后任然是原始图像。

第二个超参数是局部旋转任务的权重，我们用 α 表示它。由于标准图像分类任务是我们的主要任务，而局部旋转任务只是一个附属任务，因此我们通常为局部旋转任务分配小于 1 的权重。

3.5 整体损失函数

标准图像分类任务损失函数为:

$$f(w) = \begin{cases} 1 & w = 1 \\ 0 & else \end{cases} \tag{3.4}$$

$$loss_{\text{sc}} = \sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} f(w)L_{ce}(h(R_w^{LG}(x_{i,j}^s)\,|\,\beta_f,\beta_c), y_{i,j}^s) \tag{3.5}$$

如果我们不使用分组策略则局部旋转任务损失函数为：

$$loss_{lr} = \sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} L_{ce}(h(R_w^{LG}(x_{i,j}^s)\,|\,\beta_f,\beta_r), w) \tag{3.6}$$

如果我们使用分组策略则局部旋转任务损失函数为：

$$loss_{lr} = \sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} L_{ce}(h(R_w^{LG}(x_{i,j}^s)\,|\,\beta_f,\beta_r), z) \tag{3.7}$$

整体损失函数为：

$$loss = loss_{sc} + \alpha loss_{lr} \tag{3.8}$$

装

3.6 测试分类准确率

在验证阶段，我们使用以下公式计算标准分类任务的分类准确性：

订

$$g(x,y) = \begin{cases} 1 & x = y \\ 0 & else \end{cases} \tag{3.9}$$

$$total = \sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} 1 \tag{3.10}$$

线

$$acc_{sc} = \frac{1}{total}\sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} g(\arg\max_{w}(h(x_{i,j}^s\,|\,\beta_f,\beta_r)), y_{i,j}^s) \tag{3.11}$$

在验证阶段，我们使用以下公式计算局部旋转分类任务的分类准确性：

$$g(x,y) = \begin{cases} 1 & x = y \\ 0 & else \end{cases} \tag{3.12}$$

$$total = \sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} 1 \tag{3.13}$$

$$acc_{sc} = \frac{1}{total}\sum_{i=1}^{N^s}\sum_{j=1}^{N_i^s} g(\arg\max_{w}(h(R_w^{LG}(x_{i,j}^s)\,|\,\beta_f,\beta_r)), w) \tag{3.14}$$

4 实验与结果分析

4.1 数据集

我们用来评估模型的数据集是 PACS，它是领域泛化实验中广泛使用的基准数据集，它总共构成了 9991 张图像，这些图像分为四个领域，分别是照片，美术绘画，卡通和素描，每个域包含 7 个类别，分别是狗，大象，长颈鹿，吉他，马，房子和人。

4.2 模型

我们在模型级别使用了迁移学习技巧，更详细地说，我们使用在 ImageNet 上经过预先训练的标准 AlexNet 模型来初始化我们的模型的特征提取器。

4.3 多源域泛化实验设置

我们使用标准的多源域泛化实验设置，这意味着我们将其中三个域用作源域，将其余一个域用作目标域。

我们将源域图像随机分为训练集和评估集，其中训练集中有 90％的图像，评估集中有剩余的 10％图像。测试集包括所有目标域图像。

在训练阶段，我们使用训练集中的图像来训练模型，然后在评估阶段，我们使用评估集图像来评估训练性能，在测试阶段，我们使用测试集图像来进行训练。测试模型的域泛化能力。

4.4 其他参数

我们使用 SGD 优化器，其权重衰减为 0.0005，动量为 0.0005，初始学习率为 0.001。我们使用 StepLR 调度程序将学习速率每 8 个周期衰减 0.1。批大小为 128。使用标准数据增强技巧（随机调整大小的裁剪，抖动，随机灰度）。

我们在 Pytorch 框架中实现我们的模型，Pytorch 的版本为 1.4.0，torchvision 的版本为 0.4.2。我们使用 GPU 加快训练过程，模型是 GTX 1070，CUDA 版本是 10.1。

4.5 结果分析

4.5.1 方法比较

我们通过比较模型在 PACS 数据集上的分类正确率来开始评估模型。如表 7.1 所示，第一列表示使用的方法，第四列至第七列的标题是用作目标域的域，例如，“Cartoon”列表示目标域是卡通，源域是“美术绘画”，“草图”和“照片”。每个实验重复 3 次并取平均值。 “平均”列是第 4 列到第 7 列的平均值。

Deep all 意味着使用所有源域数据微调在 ImageNet 上预先训练的标准 AlexNet 模型，并测试目标域数据的分类准确性。每种方法都有对应的 Deep all 的原因是每种方法的实验设置彼此之间略有不同，例如，学习率，批处理大小和数据增强参数可能不同。

其中我们的局部旋转方法种的 deep all 表示我们关闭解决自我监督任务的分支（即局部旋转任务），而只保留解决标准分类任务的分支。

我们将四种局部旋转方法与其他两种方法进行了比较，其中 TF 方法使用参数化的 CNN 模型来学习域之间的不变性，更详细地讲，它定义了一种协议，以结合从每个域中学习的模型参数，最后可以从中提取广义特征。 JiGen 方法也使用了多任务学习结构，该结构同时执行标准图像分类任务和拼图任务。

25-out-25, 25-out-5, 25-out-4 and 25-out-3 方法是我们的局部旋转方法。如前所述，我们在局部旋转方法中定义了 25 种变换。我们既可以预测转换的所有 25 个标签，也可以对其进行分组，然后仅预测相应的组标签。并且我们提出了三种群组协议，因此就像我们总共有四种基于局部旋转方法的不同的自我监督方法一样，我们将其称为 25-out-25、25-out-5 ，25-out-4 和 25-out-3。

更详细地说，25-out-25 表示我们可以预测所有 25 个转换标签。 25-out-5、25-out-4、25-out-3 表示我们分别根据全局旋转度，局部旋转度和局部旋转的部位的位置对 25 个变换进行分组。并且将分别对 5、4、3 个组标签进行预测

Table 4.1 Result of experiments on PACS

	PACS-DG	ART PAINTING	CARTOON	SKETCHES	PHOTO	AVG.
[13]	Deep all	63.30	63.13	54.07	87.70	67.05
	TF	62.86	66.97	57.51	89.50	69.21
[9]	Deep all	66.68	69.41	60.02	**89.98**	71.52
	JiGen	67.63	71.71	65.18	89.00	73.38
	Deep all	67.51	70.21	64.30	89.40	72.85
	25-out-25	68.90	70.22	68.63	89.74	74.37
	25-out-5	69.38	70.32	67.94	89.72	74.34
	25-out-4	68.57	**70.63**	**70.62**	89.58	74.85
	25-out-3	**70.23**	70.15	67.97	89.94	74.57

我们用粗体表示每列中的最高值，可以看到多任务结构加局部旋转任务确实提高了模型的泛化能力，且基于局部旋转方法的方法显示出比其他方法更好的性能。其中 25-out-4 方法给出了最佳平均结果，且当目标域为 Sketch 和 Cartoon 的情况下，25-out-4 也是最优的。当目标领域是 Art paiting 时，25-out-3 表现优于其他方法。结果表明，局部旋转任务可以帮助弥合不同域之间的域差。其中它对 Sketch 域的泛化能力提高最多很多，改善了 5%，这可能是因为草图比线条较为简单因此，模型更容易在训练和评估阶段中过拟合。而 Photo 域几乎没有改进，原因可能是标准 AlexNet 是使用照片域图像进行预训练的，它已经具有非常高的提取照片图像特征的能力。因此，自我监督的任务无济于事。

其它结果分析由于图片数量过多因此请见英文版

5 结论

在本文中，我们研究了一种旨在通过同时优化有监督的标准图像分类任务和自监督任务来解决领域泛化问题。我们复现了原有模型，并提出了一种新的自监督任务，它是基于标准旋转任务，我们称之为局部旋转任务，在此过程中，我们了解了多任务学习，有监督学习，无监督学习，自监督学习和迁移学习的知识。我们已经定义了模型损失函数，解释了标准图像分类任务和局部旋转任务的详细实现。

我们进行了实验以评估我们的模型，我们在 PACS 数据集上进行了标准的多源域泛化实验，并模型的性能与其它方法。结果表明在这种情况下，我们的局部旋转方法平均优于其他方法。但是实际上，我们不确定我们的方法是否也可以在其他数据集中良好工作，例如 DomainNet，VLCS，Office Home 等。我们不确定我们的方法是否可以在其他实验设置上很好地工作，例如领域自适应[17]和部分领域自适应[18]。

为了更深入地了解我们的模型，我们还观察了训练过程和分类准确性的细节，我们进行了剥离，最后尝试可视化模型并比较各个领域的的差异。我们可以得出结论，我们的方法是有效的，它可以帮助提高模型的域泛化能力。

总而言之，我们的主要贡献是我们提出了一种新型的自我监督任务，即局部旋转任务，并提出了几种分组协议以优化模型。

将来，我们将使用不同实验设置在更多数据集上进行更多实验，以验证我们的方法。除此之外，我们还想进一步研究这些方法为何可行，以及 25-out-25、25-out-5、25-out-4、25- out-3 方法之间的本质区别。此外，我们将尝试进行更多类型的转换并探索新的分组策略。我们将尝试寻找深度模型的基本规则和规律性，并尝试从高层次和低层次来解释我们的模型。

1 Introduction

1.1 Machine Learning

Learning is a process of gaining knowledge, information, and experience. We learn every day, and we learn from birth until now. When we are a little child, we learn to walk, speak, eat…, when we are in school, we learn to solve math question, learn to write an emotional essay, learn the principle of the world, when we are no longer a student, when we are working in a company, we learn how to deal with the relationship with our colleagues and supervisors, we learn how to control our emotion, we learn to make important decisions. It seems very normal to us, nothing special. The animals can also learn, tigers learn to hunt, birds learn to fly, fish learn to swim. There seems to be nothing new, we treat the learning process as a specific activity of living creatures, we never expect that a machine can also learn just like a human or animal does. However, the machine can learn indeed.

In the past decades, machine learning is known by more and more peoples, it can be considered as a process of machine evolution, in this process, the machine gets knowledge and experience by taking in and analyzing a bunch of data, then improves and optimizes its structure and program automatically in order to do tasks better. We take an example to show the process of machine learning, imagine that now the machine is a kid who knows nothing, now we want to teach this kid to do images classification task, the only thing we need to do is: we take out an image and show to the kid, then ask him what is that, the kid gives an answer then we tell him the correct answer, and then we take out next image and repeat the procedure…, during this process, the kid will gain knowledge like, an apple is more or less a round object with a groove in the top. the kid can remember the shape of the apple, then next time when he sees an object with this kind of shape, he knows it is an apple.

In other words, we don’t need to change manually the values of variables or code structure of the machine, we just feed it data, then it can update its memory, i.e., its values of variables, by itself. The advantage of machine learning is that it can leverage a huge amount of data and find regularity inside, which is relatively hard for the human brain. However, the disadvantage is that it will not give a satisfactory performance if the input data are wrong or we cannot provide enough input data. In most cases, the quality and amount of data are still the key factors and limitations of the performance of machine learning.

In summary, machine learning is a process in which the machine learns by itself from input data. The input data is very crucial to machine learning.

1.2 Deep Learning

Deep learning is a sub-branch of machine learning algorithms, now it is the most advanced and leading research field of machine learning. The difference between conventional machine learning algorithms and deep learning is the ability to utilize data, deep learning does not show a better performance than conventional machine learning algorithms when the amount of the training data is relatively small, but when we have more data, the conventional machine learning algorithms go into a saturated state earlier, deep learning shows a much better potential instead.

Actually, deep learning has shown comparable and even better performance with respect to human beings in many fields, e.g. computer vision, natural language processing, and automatic speech recognition. The core of deep learning is actually the artificial neural network, it just like an artificial brain. People could be confused about the relationship between this artificial brain and the human-being brain, how can this artificial brain memory and process the information that it gets, how can it analyses, thinks, and judges in the real situation. Actually, the scientist get inspiration from the real nervous system, let compare their structures and functionalities.

In the cerebral cortex, there are about 14 billion neurons, which compose one of the most sophisticated nervous systems in the world. The nerve cell is the core of the magic of our brain, it is the foundation of our intelligence. At a high level, the most important functional components of a neuron are Dendrite, cell body, and Axon. The Dendrite acts like an acceptor that receipts the input from other neuron or environment, the cell body collects all the input then decide should it activate the output and send a signal to the next neuron or not, once the cell body decides to send a signal, the Axon will take the responsibility of sending the signal.

This is the most basic and simplest mechanism of the real neuron. If we can mimic this process, then we could build an artificial brain. Actually, we do try to replicate this process in the artificial neural network. In an artificial neural network, each computational unit works like a neuron. it takes in a set of inputs, and then do matrix computation, and then feed the result into an activation layer to decide should it sends a value to the next unit.

However, although we can have a similar neural structure in the artificial neural network, there are still some deficiencies. On the one hand, usually there are many more neurons in our brain then those in an artificial network, the real brain is much stronger and robust. When we increase the number of variables and layers in artificial neural networks, we can have a more strong and powerful model, but the computation time and power consumption also increase.

On the other hand, the real neuron is much more complicated, there are many mechanisms and phenomena still unclear for us, one example is that the signal could be enhanced or weaken for many reasons during the transition between real neurons, however, in artificial neuron there is only a simple and static activate function. Another example is that one real neuron can connect with many other

neurons, which means a stronger ability for non-linear computation, instead, in the artificial neural network, a single connection structure between artificial neurons is more commonly used. In other words, the real neuron can cope with more difficult and complex tasks, and the connections between real neurons are much more complex.

In summary, deep learning gets inspiration from the real nervous system, in the past 20 years, we have a much stronger computing power, deep learning shines in many fields, but we don't know what is the end of this story, we are still wondering about whether it can fully win human beings or not in the future, we are still exploring in this field.

1.3 Images Classification and CNN

Image classification is one of the most basic tasks in the field of computer vision, we usually evaluate and compare the performance of different models using it as the test task. In this thesis, we only talk about the images classification task in species level, in other words, we only need to classify the image according to the species it belongs to, for example, given an image, the task is to find out if it contains a cat or a dog or something else.

In 2012, AlexNet[1] is the winner of the ILSVRC[2], which is one of the most famous image classification competitions. AlexNet is considered a milestone in the history of the convolution neural network (CNN), it is the first deep CNN that has a much higher performance than conventional machine learning algorithms in ILSVRC. From then on, CNN has become the most common model used to classify images.

Actually, a CNN model can be split into two components, one is features extractor, its mission is to convert the input raw data into a lower-dimensional space, which can reduce the amount of data needed to be processed, it works similarly with the algorithm of principal component analysis. The other one is fully-connected layers, which acts like a classifier on top of the features extracted. In other words, the fully-connected layers will give the classification result according to the features that come from the features extractor. In the later chapter, we will use these two concepts, i.e., features extractor and fully-connected layers to explain more detail about the main learning strategies used in this thesis.

1.4 Domain Generalization

Firstly, let's explain the concept of image domain, the domain of an image is a representation that describes the character of pixel distribution of this image. Intuitively, the domain of an image can be interpreted as the style, mode, pattern, or fashion of this image. For example, we can say that, this image belongs to the domain of sketch, that image belongs to the domain of cartoon. There is a domain shift between images that belong to different domains since their global pixel distributions are different.

Let's take more examples to explain what is the domain of an image. Different images styles can lead to different domains, e.g., images of photo style and images of cartoon style are in different

domains, because the images of photo style are usually more realistic, and images of cartoon style are relatively simpler and hide more detail information. The images shooting conditions can also lead to different domains, e.g., an image shot in a good light condition and an image shot in a poor light condition can be considered in different domains because they are shot in different light condition, and the pixels of the image shot in a good light condition is more white. The images shooting time can also lead to different domains, e.g., a scenery image shot in summer and that shot in winter can be considered in different domains, because there could be more green color in summer, and more white color in winter instead.

There are some benchmark datasets that compose images of different domains. For example, the datasets PACS[2] composes images from four domains, which are photo, art painting, cartoon, and sketch domains respectively, we show several examples in figure 1.1. The datasets of DomainNet compose images of six domains, which are clipart, infograph, painting, quickdraw, real, sketch respectively.

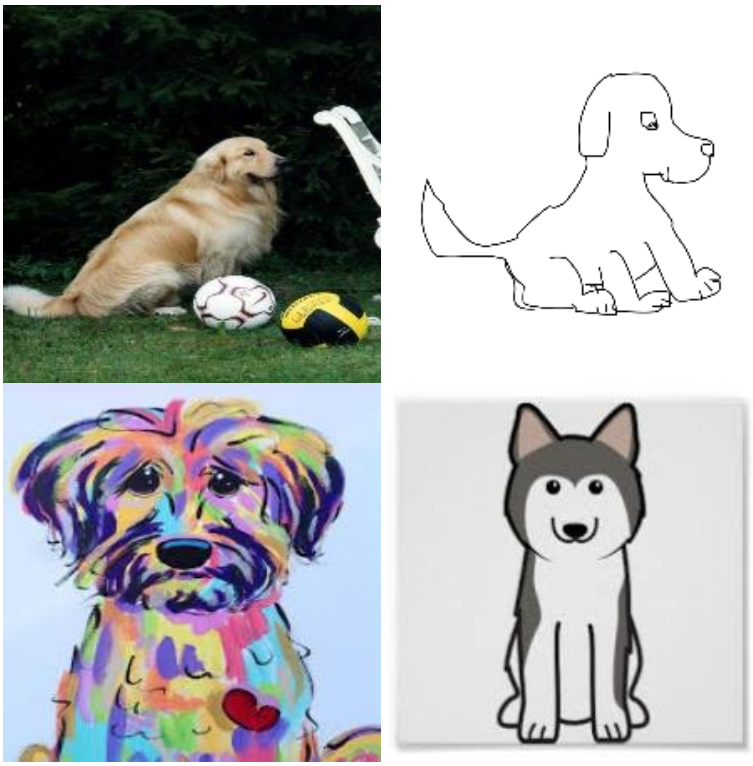

Figure 1.1 Dog images of different domains

Domain generalization is an issue that we care about very much, in this thesis, it refers to a scenario, in which a model is trained to do the task of image classification, the training data is from several domains, we call these domains as source domains, and we expect that the trained model can classify the images that belong to the domain that is not in the source domains, we call this domain as

the target domain. In other words, we have several source domains and one target domain, we train the model using images of source domains and test the trained model using the images of the target domain. Take the PACS datasets as an example, consider this scenario, we train the model using images of the art painting, cartoon, and photo domains, we call these three domains as source domains, and we will test the model performance using images of sketch domain, in this case, we call sketch domain as the target domain.

Let's show the reasons why the issue of domain generalization is so important. Now we have a scenario, in which we have used some images to train a model to classify images and we want to deploy this model in a real situation. We will have a problem, which is that although the model has already had a very good performance during the training process, it cannot work well in the real situation, because the images used to train the model and the images in the real situation could belong to different domains, the model is suffering a domain shift and it cannot generalize the information to the real situation. For example, now we have used only the images shoot in the daytime to train the model, and if the model has no ability to generalize the information across domains, then the model will only know how to classify images shoot in the daytime, and it fails at night time. To solve the problem, we have to train the model to get the ability of generalizing across domains.

1.5 Situation and Significance of Study

Firstly, we want to recall the main topic that we want to focus on, this thesis will focus on solving the image classification task and increasing the model's ability in terms of domain generalization.

As we mentioned before, since in 2012 AlexNet has won the ILSVRC, convolutional neural network has become the most popular and powerful model solving image-based tasks, e.g., image classification, object detection, and semantic segmentation. In this thesis, we focus on the task of image classification. Indeed, as for the task of image classification, CNN-based models have an absolute advantage over all the other kinds of models. Since 2012, all of the winners of ILSVRC have used CNN-based models, e.g., ZFNet[3], VGGNet[4], GoogLeNet[5], and ResNet[6], SENet[7], particularly, the SENet has achieved a test top-5 error rate of 2.25% in 2017. There is little space for improvement in term of the most usual image classification task, so the scientists now aim at a more difficult problem, which is the domain generalization issue. There are some existing methods, for example, [8] try to learn the invariant feature representation from several source domains, in which, firstly they define a transformation function, and then they use it to transform the data of different domains, then they try to reduce the dissimilarity of the transformed data. [9] proposes a multi-task learning structure, in which it solve the standard image classification task and the jigsaw puzzle task at the same time, the jigsaw puzzle task performs a regularization effect and help the model to generalize across different domains. [10]proposed a method based on classifying the different kinds of transformations, in which it has presented a new kind of self-supervised task, and train a model to

classify the original image, rotated, warped, local inpainting images, so that the model can learn a better generalized feature representation. [11] presented a method that try to learn the invariant BoW of the original image and the transformed image, in which the main point is that the BoW of the original image and the transformed image should have the same BoW.

The image classification task is the most basic image-based task, we are surprised by the ability of the robotics shown in the film, the robotics can process and identify the image they capture, and then do the correct action. Image classification is only a basic functionality of the super eyes of the robotics. And we will never buy a robotic that can only identify the images of specific domain, what we want is a universal robotic that can work on different situations, both at the daytime and night time. Thus, we need to increase the model's domain generalization ability.

1.6 Existing Problems

Indeed CNN-based deep models have achieved great success in solving the image classification task, however, there are still some deficiencies and limitations, here we focus on two problems, one is the lack of good data, the other one is the lack of ability to generalize information across domain. Our method to solve these two problems will be introduced in later section.

1.6.1 The Lack of Good Labeled Data

The CNN-based deep network has a very powerful ability in processing 2-D data, particularly, it works well on expressing the non-linear relationship between the input images and its corresponding labels. However, the model can get this great performance mainly through supervised learning.

Supervised learning is a kind of machine learning algorithm that makes use of labeled data. The amount and the quality of the data are the most important factor that affects the performance of the model trained with supervised learning algorithm.

To be more precise, on the one hand, if we don't feed the model enough labeled data, the model cannot learn the overall data distribution, thus it cannot work well. And usually the larger the network, the more data it needs to get a good enough performance.

On the other hand, the labeled data should be correct. This point is obvious, if we feed the wrong data to the model, of course it will learn a wrong data distribution and give a bad performance.

However, the labeled data is very expensive and error-prone, the common situation is that we usually cannot get enough good labeled data to train the model, which is a restriction and limitation in supervised learning.

1.6.2 The Lack of Domain Generalization Ability

As we mentioned in the background section, the model can fail in the real situation if it doesn't have the ability to generalize the information across domain, because the data in the real situation can differ from the training data. A model without the domain generalization ability cannot be considered as

a good model, because it can only cope with the data that in a specific domain, when the domain change, it fail. What we need is a powerful, stable and robust model that can work in different situation.

In summary, we have a strong demand for the model with domain generalization ability, the ability of domain generalization is important. In this thesis, we will propose the idea and method to solve this problem.

1.7 Thesis Structure

In chapter 1, we have introduced some background information about machine learning, deep learning, image classification, CNN, and domain generalization. And then we show the current situation and significance of the study of image classification and improving the model's ability of domain generalization. What's more, we have presented two existing problems in image classification, which is the lack of good labeled data since the most commonly used and powerful learning algorithm is supervised learning and the lack of domain generalization ability.

In chapter 2, we show the common knowledge and principle of the main algorithms and learning strategies that we have used in our model, this chapter is the theory foundation of our model. We have introduced the multi-task learning, supervised learning, unsupervised learning, self-supervised learning, and transfer learning.

In chapter 3, we introduce our model to solve these two problems, our model is based on a previous model that is proposed in this paper[9], we keep most of the structure of this previous model, which has been proven as an effective model, and do some modification to it. More specifically, in our model, we propose a new kind of self-supervised task, which is based on the standard rotation task[14]. And we replace the self-supervised task used in the previous model with this new self-supervised task. In brief, in our model, we have used a different self-supervised method with respect to the previous model.

the overview of our model is that we use a multi-task[15] learning structure to train the model, in which we do two tasks at the same time, one of the tasks is the standard image classification task using supervised learning algorithm, and the other one is a modified rotation task using the self-supervised learning algorithm, we call it the localized rotation task. The advantage of our model is that we can make use of the massive available unlabeled images through the self-supervised task, in the meanwhile, the multi-task learning structure can helps to increase the ability of domain generalization.

In chapter 4, we demonstrate how we design the experiments the evaluate our model. We use the datasets of PACS to evaluate our model, the experiments setting is the standard domain generalization. And then we present the experiments result and discuss them. We show the comparison of different methods, the training process, and something else.

In chapter 5, we post our conclusion in brief, the doubt we have, and the future work we can do.

2 Algorithms and Strategies

This section is the most important theory foundation, we will introduce all the learning algorithm and the strategies that we have used. Firstly, we will introduce the multi-task learning structure, which is the overall structure of our model. And we will introduce supervised learning, unsupervised learning, and self-supervised learning algorithm, particularly, we have used both supervised learning and self-supervised learning algorithm in our model. Apart from these, we will introduce a transfer learning strategy that we have used in our evaluate experiments, in more detail, we use a model pre-trained on ImageNet to initialize our model.

2.1 Multi-task Learning

The core idea of multi-task learning is that the model tries to do multiple tasks at the same time.

First of all, let's see an example of multi-task learning in daily life, which is that when we are learning a new language, we would like to practice listening, speaking, reading, and writing skills at the same time, since it is a comprehensive way to improve our overall level. These tasks actually are different tasks, but they help each other indeed. The reason is that these tasks share the same foundation, which can be considered as the understanding of the words, all of these tasks can improve the understanding of the words, so all of them can help each other.

Actually, multi-task learning has been proven to be an effective learning strategy to increase the model's ability of generalization. The concrete implementation of our model is that we tries to do a main task and a relative task at the same time, each task will generate a loss component, so the model has to minimize multiple loss components at the same time. Thus, it requires the algorithm perform well not only on the original task but also the relative task, it punishes the model if the model only focuses on the original task, which is a better penalty rule with respect to the standard regularization rule that punishes all the variable according to their magnitude. We can explain this regularity effect in a more intuitive way, during training, these multiple tasks share the same backbone, in other words, these tasks share representations and knowledge with each other, which means that the knowledge learned by task 1 can be used by task 2, just like the example of learning language proposed above, they help each other to learn better in an overall way.

In brief, the model can have a better generalization ability in terms of expressing the whole data space distribution, instead of overfit on a specific data domain.

We have introduced the character of multi-task learning strategy in a general overview above, and we want to talk about controlling the weight of each task. Sometimes, we want to pay different attention to different tasks, we want to spend more power and strength on those tasks that we care about more.

For example, the writing test is exactly on tomorrow morning, then we usually would like to do more writing tests at tonight, which means that in this situation we want to pay more attention to the task of writing task, we care about more the result of writing tests at this moment, instead of the overall language level. In other words, we need to control the weight of each task in multi-task learning algorithm, to do this thing, we can just assign different weights to the loss component since the task and the loss component are one-to-one corresponding.

2.2 Supervised Learning

The main character of supervised learning is that it needs annotated data to train the model. The annotated data is the data that has a label, the label represent the data's property and attribute, for example, given a dog image, then the dog is its species category, in other words, its label.

We call this kind of learning algorithm as supervised learning, because the label of the data is just like a supervisor, it teach the model to distinguish the difference of different data.

Supervised learning are commonly used to solve classification and regression task. The classification task is the main topic of our thesis, for example, the image classification task is a standard classification task, the mission of the model is to predict the probability of each category given an image. To be more precise, give an image, the model will process this image and tell us something like that: this image has a probability of 80% to be a dog image, and 2% to be a cat, and 18% to be a person.

The regression task is another kind of supervised learning task, it take in two group of data, which can be treat as input x and output y, the mission is to find the relationship between x and y, the model need to predict y according the input x.

The difference between classification and regression is that, in most cases, we need to know the total number of the classes in the classification task, which is a problem, the common solution is that we increase the number of the total classes, so we can cover more kinds of category, for example, the ImageNet compose more than 20000 categories, it is a very important image dataset in the field of computer vision. And as for the regression task, the input x can be continuous, the output y can also be continuous.

In summary, supervised learning has achieved a great success in the field of data science, but there still some drawback and limitation, firstly, it needs a bunch of data, it cannot show satisfactory performance without enough training data. Secondly, the quality of the training data matter a lot, if the labels of the data is wrong, then the model gets confused and fails. Thirdly, it has discarded some information that maybe useful, to be more precise, the model only keep the category information of a given image, instead of the whole image, it losses a lot information, such as the format, color, style, mode of the image.

2.3 Unsupervised Learning

The unsupervised learning is another kind of commonly used machine learning algorithm, it is different from the supervised learning algorithm, it does not need annotated data and it is usually used to solve the clustering task and association analysis.

The clustering task is to group a set of input data without any label, the algorithm has to explore the underlying structure by itself, the data that belong to the same group should be similar to each other, and unsimilar to the data that belong to other group. There are many kinds of applications of clustering, for example, market segmentation is to group the customer according to their click action, shopping history, browsing behavior, so that the shopping website can predict what the customer want to buy and then make recommendation.

The association analysis is to find the relationship between the data, one application example is the market analysis. The store would put the relevant goods together, so that the consumer can remind himself of something he might need to buy, then the store can sell more goods.

There are many advantages in unsupervised learning algorithm, first of all, it can make use of massive cheap unannotated data, so that it can avoid the potential error labels with respect to the supervised learning algorithm. What's more, it gives the model the whole information without dropping any piece because the model explore information directly from the raw input data. However, there are also some disadvantages, firstly, it doesn't give an exact meaningful result, for example, in the clustering task, we can group the data, but we don't know what is the real meaning of the group, we only know this data belongs to this group, but we can interpret this group according to many rules. Secondly, the training of unsupervised task is relatively more difficult and time-consuming then the supervised learning task, an intuitive explanation could be that, in supervised learning, there seems to be a teacher to teach the model, but in unsupervised learning, the model explore the structure of the data by itself.

2.4 Self-supervised Learning

In my point of view, self-supervised learning can be considered as either a kind of unsupervised learning or a kind of supervised learning method, in the sense that it utilize unannotated data, which is similar to the unsupervised learning method, and it supervises the training process by itself, which is similar to the supervised learning method.

The standard rotation task is a kind of self-supervised learning task, the idea is that, we rotate an image randomly by 0, 90, 180, or 270 degrees, and we train a model to recognize the degree it has rotated, during this process, what we need is only the raw image without any label annotated by human being, and the image generate a pseudo label from itself, which is the degrees it has rotate. We form this process as a classification process, in which the degrees is the pseudo label of the image.

Another self-supervised learning tasks is solving the jigsaw puzzle in which we shuffle the jigsaw puzzle according to a specific sequence, then the sequence is the pseudo label of the image, what the

model needs to do is to recognize the sequence given a shuffled image.

The advantage of the self-supervised learning method is that it can make use of unlabeled data and generate a pseudo label by itself, so we can form the self-supervised learning task as a similar supervised learning task.

2.5 Transfer Learning

Transfer learning is a kind of machine learning technique that tries to transfer the knowledge gained from one task to another relative task. It just like that we try to reuse the existing knowledge instead of learning a new thing from zero, for example, if we have learned to solve a math question, then usually we can solve the similar questions quickly.

In machine learning, the value of the variables is the knowledge, what we do is to store these values in hard drive and load them when we want to reuse them. If we want to train the model from scratch, then we will initialize the variables of the model randomly, but the problem is that, it will take much more time to train the model.

A commonly used paradigm is that we pretrain a model to solve image classification task on ImageNet[12], which is a useful image dataset that composes 14,197,122 images by now, and then we create a new model and initialize it with the pretrained model, then we can add new fully-connected layers on top of it and train it to solve our required task. In our thesis, we have also used this trick to shorten the training time.

In summary, transfer learning aim at reusing the gained knowledge and shorten the time needed to train the model.

3 Method

In section 3.1, we will introduce the structure and the workflow of our model in a general overview. In section 3.2 and 3.3 we will introduce the detail of the two tasks that the model try to do. In section of 3.4 and 3.5 we will show the hyperparameters and overall loss equation.

3.1 Model Overview

Our model aims at solving the two problems, which are the lack of good labeled images and the lack of domain generalization ability. As for the first problem, our solution is to use the self-supervised learning algorithm, which can make use of massive available unlabeled images. As for the second problem, our solution is that we use a multi-task training structure, which composes two tasks, these two tasks share the same backbone that extracts image features, one of the tasks is the standard supervised image classification task, the other one is the self-supervised localized rotation task. The reason why we use the multi-task learning algorithm is that it can help to share the knowledge across multiple tasks and gives a regularization effect as we introduced before.

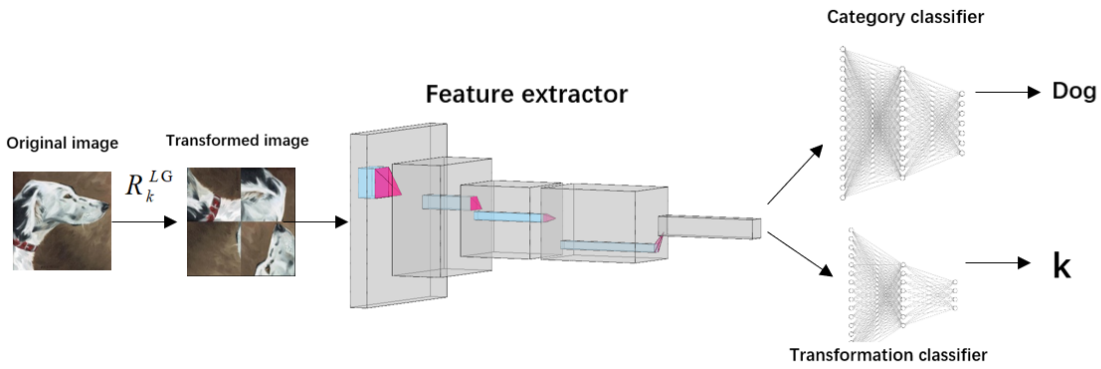

Figure 3.1 Illustration of workflow

As shown in figure 5.1, firstly we take out an image, and split it into four parts equally. We have defined 25 kinds of transformations, of which we have shown the detail in section 5.3, particularly the first transformation is actually keeping the image unchanged. We have assigned a label for each transformation, furthermore, we group these transformations according to different protocol, and assign a group label for each transformation, the transformations that belong to the same group will have the same group label.

Then we select one transformation out of the total 25 ones and apply it to the image to get the transformed image. This transformed image has two labels, one is the category label, e.g., dog, cat, or

something else, the other one is the transformation label or the group label.

In our model, we have a feature extractor, which is the common backbone of the two tasks. On top of the feature extractor, we have two classifiers, which are composed by several fully-connected layers respectively.

Then we feed this transformed image into the model to solve two tasks: the standard image classification task and the localized rotation task. In more detail, the category classifier solve the standard image classification task, the localized rotation classifier solve the localized rotation task.

3.2 Standard Image Classification Task

In this part we show how we do the standard image classification task. Assume we have N^S source domains, the $i-th$ source domain contains N_i^S images. And these images belong to N^C categories. We use the symbol shown as below to represent all images that belong to the $i-th$ source domain:

$$D_i^S = \{x_{i,j}^S, y_{i,j}^S\}_{j=1}^{N_i^S} \tag{3.1}$$

in which, $x_{i,j}^S$ represent the $j-th$ image in the $i\text{-}th$ source domain, and $y_{i,j}^S$ is the category label that belongs to $\{1,2,...N_C\}$.

So given an image, we can use the loss equation below to measure the distance or error between the predicted category and the ground truth:

$$loss_{sc} = L_{ce}(h(x_{i,j}^s \mid \beta_f, \beta_c), y_{i,j}^s) \tag{3.2}$$

in which β_f represents the parameters of the feature extractor, and β_C represents the parameters of the classifier of image categories, thus h represents the deep model that is parametrized by β_f and β_C, and $h(x_{i,j}^S \mid \beta_f, \beta_c)$ means the predicted result of the given image. And L_{ce} is the standard cross-entropy loss equation, which is suitable for classification tasks.

3.3 Localized Rotation Task

3.3.1 Definition of Local and Global Rotation

At the beginning, we would recall the standard rotation transformation method, in which, we rotate the whole image by 0, 90, 180, or 270 degrees, then we can have four kinds of different transformations and four corresponding labels. In image 2, we show an example of these four transformations.

The procedure of the localized rotation transformation is that: Firstly, we split the original image into 4 parts as shown in figure 5.2, then we rotate the whole image by 0, 90, 180, or 270 degrees, we define this step as global rotation, one example is shown in figure 5.3, the next step is that, we apply rotation for each part again, we define this step as local rotation, one example is shown in figure 5.4.

The reason why we call our transformation method as localized rotation transformation is that: It

composes both global and local rotation, and the standard rotation transformation method only composes global rotation, we want to highlight the difference between our transformation method and the standard rotation transformation method.

According to the definition, we can have four kinds of global rotation, and we use $\{R_k^G\}_{k=1}^4$ to represent the global rotation transformations. We have four local parts, for each part, we can rotate it by 0, 90, 180, or 270 degrees, so we have $4^4 = 256$ kinds of local rotation transformations, and we use $\{R_k^L\}_{k=1}^{256}$ to represent the local rotation transformations.

Figure 3.2 We split the image into four part

Figure 3.3 Global rotation with degree of 90°

Figure 3.4 Local rotation of part 2 of 90°

3.3.2 Reduce the Total Number of Transformations

As shown above, we have 4 kinds of global rotation transformations and 256 kinds of local rotation transformations, thus in total we can have 4*256=1024kinds of transformations since we can apply global rotation and local rotation to an image at the same time.

However, 1024 kinds of transformations mean 1024 kinds of categories, a larger number of categories in hyper-dimensional space means less distance between labels, then it is harder for the model to distinguish them. In order to make the task easier for the model, we have reduced a lot of kinds of transformations.

The first reduction we do is that: we select only two parts of the image to go through the local rotation transformation, and these two selected parts must on the diagonal, thus we have only two choices which are parts 1, 4 or parts 2, 3. for example, if we choose to do local rotation transformations to parts 1, 4, then parts 2, 3 will be left unchanged, and vice versa. So now we can reduce the number of total transformations to 4*2*4*4=128, in which the first '4' refers to the global rotation, the first '2' means we have two choices when selecting the local parts to undergo local rotation, the last two '4' means, for each selected local part, we have four kinds of degrees to rotate, which are 0, 90, 180, or 270 degrees respectively.

The second reduction we have done is that: We can have four kinds of degrees to rotate, which are 0, 90, 180, or 270 degrees, however, when the rotation degree is 0, no transformation happens actually. And we want the transformed image must go through some sort of local rotation. In other words, when doing local rotation, we don't like the choice of 0 degree, because it means no transformation happen. So, we force the local rotation degree cannot be 0, which means that we can only choose to rotate the selected 2 local parts by 90, 180, 270 degrees, no 0 degree. So now there are only 4*2*3*3=128kinds of transformations, in which the meanings of '4' and '2' are unchanged, the two '3' represent that, for each selected local part, now we can only rotate it by 90, 180, or 270 degrees, no 0 degree.

Well we have even the third reduction: As we mentioned above, we select two parts to undergo local rotation, now we add a restricted condition, which is that the selected two parts must rotate by exactly the same degrees. So now we have only 4*2*3+1=25 kinds of transformations in total, in which we have only one '3' now, and particularly the '1' means the original image. We use $\{R_w^{LG}\}_{w=1}^{25}$ to represent these reduced transformations, one example is shown in figure 5.5, particularly, R_1^{LG} actually means the image undergoes no transformation.

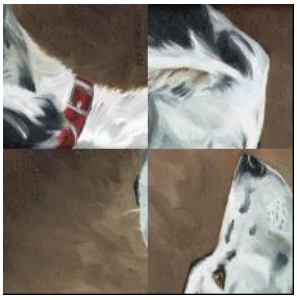

Figure 3.5 Global rotation of 90° plus local rotation of parts 2, 3 of 180°

3.3.3 Loss Equation

Given an image $x_{i,j}^{s}$, firstly we select an R_w^{LG} from $\{R_w^{LG}\}_{w=1}^{25}$ and apply it to this image, thus we get a transformed image:

$$u_{i,j,w}^{s} = R_w^{LG}(x_{i,j}^{s}) \tag{3.3}$$

then we feed it into the feature extractor and the localized rotation classifier to get the predict logit value:

$$\log it_R = h(u_{i,j,w}^{s} \mid \beta_f, \beta_r) \tag{3.4}$$

in which, the deep model h has the same structure with respect to the standard image category classification task, and there share the same parameter βf that parametrizes the feature extractor, and βc parametrizes the classifier of localized rotation transformations, $h(R_w^{LG}(x_{i,j}^{s}) \mid \beta_f, \beta_r)$ means the predicted index of localized rotation of the given image.

Thus, the loss equation is:

$$loss = L_{ce}(h(R_w^{LG}(x_{i,j}^{s}) \mid \beta_f, \beta_r), w) \tag{3.5}$$

in which, L_{ce} is the standard cross-entropy loss equation.

3.3.4 Group the Transformations

We have talked about the procedure of the localized rotation task above, in the standard-setting we have 25 kinds of transformation thus we have 25 labels to predict. However, actually we don't need to predict all 25 labels, we can divide these 25 kinds of transformation into several groups and give them group labels respectively, what the deep model needs to do is to predict only the group label, instead of the explicit transformation label. In other words, we can have a different number of labels to predict according to different group protocols. In this part, we will introduce several protocols to divide these 25 transformations.

(A) Group by global rotation

The first group rule we propose is to group them by the global rotation degree. We put all of the transformations that have the same global rotation degree in the same group with the same group label.

In this situation, we have 5 groups labels, we use $\{G_z^G\}_{z=1}^5$ to represent them, in which,

$$G_1^G = \{R_1^{LG}\} \tag{3.6}$$

$$G_2^G = \{R_w^{LG}\}_{w=2}^7 \tag{3.7}$$

$$G_3^G = \{R_w^{LG}\}_{w=8}^{13} \tag{3.8}$$

$$G_4^G = \{R_w^{LG}\}_{w=14}^{19} \tag{3.9}$$

$$G_5^G = \{R_w^{LG}\}_{w=20}^{25} \tag{3.10}$$

the loss equation can be redefined as:

$$loss = L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), z) \tag{3.11}$$

in which the R_w^{LG} belongs to G_z^G.

(B) Group by local rotation

Apart from the global rotation degree, each transformation has also a local rotation degree. So we can also group these transformations by their local rotation degree, the transformations that have the same local rotation degree can be treated as in the same group, so we have 4 groups, we use the symbol $\{G_z^L\}_{z=1}^4$ to represent them, in which,

$$G_1^L = \{R_1^{LG}\} \tag{3.12}$$

$$G_2^L = \{R_{2+3w}^{LG}\}_{w=0}^7 \tag{3.13}$$

$$G_3^L = \{R_{3+3w}^{LG}\}_{w=0}^7 \tag{3.14}$$

$$G_4^L = \{R_{4+3w}^{LG}\}_{w=0}^7 \tag{3.15}$$

Thus, the loss equation becomes:

$$loss = L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), z) \tag{3.16}$$

in which the R_w^{LG} belongs to G_z^L.

(C) Group by the parts that undergo local rotation

Apart from the two group protocols that we proposed above, we can also group these transformations according to which two parts we select to apply local rotation. Since we have only two combinations in terms of the parts selected, which are parts 1, 4, or 2, 3, so we have only 3 groups now, we use $\{G_z^P\}_{z=1}^3$ to symbolize them, in which,

$$G_1^P = \{R_1^{LG}\} \tag{3.17}$$

$$G_2^P = \sum_{i=2}^{4} \{R_{i+6w}^{LG}\}_{w=0}^{3} \tag{3.18}$$

$$G_3^P = \sum_{i=5}^{7} \{R_{i+6w}^{LG}\}_{w=0}^{3} \tag{3.19}$$

now the loss equation is:

$$loss = L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), z) \tag{3.20}$$

in which the R_w^{LG} belongs to G_z^P.

3.4 Hyperparameters

As we mentioned before, our model uses a multi-task learning strategy and it does the standard image classification task and the localized rotation task at the same time. In this part we will show the overall procedure in more detail.

There are two hyperparameters in our model, the first one that we define is the proportion of the original images, it refers to the probability that images remain unchanged, we use β to represent it. For example, if we set β=0.6, then we will have the probability of 60% to select R_1^{LG} from, and the probability of $(1-60\%) \div 24 = 1.6\%$ to select one from the others transformations to apply to the image. In other words, after the localized transformation, nearly 60% of images are the original images since R1LG actually means no transformation.

The second hyperparameter is the weight of the localized rotation task, we use α to represent it. Since the standard image classification task is our main task, and the localized rotation task is only a relative task, we usually assign a weight that is less than one to the localized rotation task.

3.5 Overall Loss Equation

The loss equation of the standard image classification is:

$$f(w) = \begin{cases} 1 & w = 1 \\ 0 & else \end{cases} \tag{3.21}$$

$$loss_{sc} = \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} f(w) L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_c), y_{i,j}^s) \tag{3.22}$$

if we don't use any group protocol to group the localized rotation task then the loss equation of the localized rotation task is:

$$loss_{lr} = \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), w) \tag{3.23}$$

if we group the transformation, and take the protocol that groups the transformations by global rotation degree as an example, the $loss_{lr}$ become:

$$loss_{lr} = \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} L_{ce}(h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r), z) \tag{3.24}$$

in which, R_w^{LG} belongs to G_z^G. And now we can have the overall loss equation:

$$loss = loss_{sc} + \alpha loss_{lr} \tag{3.25}$$

3.6 Validate the Accuracy

During the validation phase, we calculate the classification accuracy of the standard classification task using the equation below:

$$g(x, y) = \begin{cases} 1 & x = y \\ 0 & else \end{cases} \tag{3.26}$$

$$total = \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} 1 \tag{3.27}$$

$$acc_{sc} = \frac{1}{total} \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} g(\arg\max_w (h(x_{i,j}^s \mid \beta_f, \beta_r)), y_{i,j}^s) \tag{3.28}$$

And we calculate the classification accuracy of the localized rotation task using the equation below:

$$g(x, y) = \begin{cases} 1 & x = y \\ 0 & else \end{cases} \tag{3.29}$$

$$total = \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} 1 \tag{3.30}$$

$$acc_{sc} = \frac{1}{total} \sum_{i=1}^{N^s} \sum_{j=1}^{N_i^s} g(\arg\max_w (h(R_w^{LG}(x_{i,j}^s) \mid \beta_f, \beta_r)), w) \tag{3.31}$$

4 Experiments and Result Analysis

4.1 Datasets

The datasets that we use to evaluate our model is PACS, which is a widely used benchmark datasets in the domain generalization experiments, it composes 9991 images totally, these images are of four domains, which are photo, art painting, cartoon, and sketch, and each domain contains 7 categories, which are dog, elephant, giraffe, guitar, horse, house, and person.

4.2 Model Backbone

We have used a transfer learning trick in the model level, in more detail, we use a standard AlexNet model that is pretrained on ImageNet to solve the standard image classification task to initialize the feature extractor of our model.

4.3 Multi-source Domain Generalization

We use the setting of standard multi-source domains generalization, which means that, we use three domains as the source domains and the remaining one as the target domain.

We split the source domains images into a training set and evaluation set randomly, in particular, 90%images are in the training set, the remaining 10% images are in the evaluation set. All the target domain images are in the test set.

During the training phase, we use the images that are in the training set to train the model, then during the evaluation phase, we use the evaluation set images to evaluate the training performance, during the test phase, we use the test set images to test the domain generalization ability of the model.

4.4 Miscellaneous Parameters

We use the SGD optimizer with the weight decay of 0.0005, the momentum of 0.0005, and the initial learning rate of 0.001. We use a StepLR scheduler to decay the learning rate by 0.1 every 8 epochs. The batch size is 128. The standard data augmentation trick (random resized cropping, jitter, random grayscale) is used.

We implement our model in Pytorch framework, the version of Pytorch is 1.4.0 , the version of torchvision is 0.4.2 . We use GPU to speed-up our training process, the model is GTX 1070, the CUDA version is 10.1.

装 订 线

4.5 Result Analysis

4.5.1 Methods Comparison

We start our evaluation experiments by comparing the average performance on the datasets of PACS. As shown in table 7.1, the first column indicates the method used, the title of the 4th to 7th column is the domain that is used as the target domain, for example, the column of 'Cartoon' means that the target domain is Cartoon and the source domains are Art Painting, Sketches, and Photo. Each experiment is repeated for 3 times and take the average value. The column of 'Avg.' is the average value of the 4th to the 7th column.

Table 4.1 Result of experiments on PACS

	PACS-DG	ART PAINTING	CARTOON	SKETCHES	PHOTO	AVG.
[13]	Deep all	63.30	63.13	54.07	87.70	67.05
	TF	62.86	66.97	57.51	89.50	69.21
[9]	Deep all	66.68	69.41	60.02	**89.98**	71.52
	JiGen	67.63	71.71	65.18	89.00	73.38
	Deep all	67.51	70.21	64.30	89.40	72.85
	25-out-25	68.90	70.22	68.63	89.74	74.37
	25-out-5	69.38	70.32	67.94	89.72	74.34
	25-out-4	68.57	**70.63**	**70.62**	89.58	74.85
	25-out-3	**70.23**	70.15	67.97	89.94	74.57

The Deep all means finetuning the standard AlexNet model pre-trained on ImageNet using all the source domains data, and test the classification accuracy on the target domain data. And the reason why each method has its own deep all is that the experiment setting of each method is slightly different from each other, for example, the learning rate, batch size, and data augmentation parameters can be different.

Particularly, the Deep all of our methods means we shut down the branch that solves the self-supervised task, i.e., the localized rotation task, and leave only the branch that solves the standard classification task. We use deep all as our baseline to evaluate the contribution of the self-supervised task.

We compare our four localized rotation methods with other two methods, among which the TF method uses a parametrized CNN model to learn the invariant feature among domains, in more detail, it defines a protocol to combine the parameters of the model learned from each domain, and finally they

can extract the generalized features from them. The JiGen method also uses a multi-task learning structure, which do the standard image classification task and the jigsaw puzzle task at the same time.

And the 25-out-25, 25-out-5, 25-out-4 and 25-out-3 methods are our localized rotation methods. As we introduced before, we have defined 25 kinds of transformations in the localized rotation method. We can either predict all of the 25 labels of the transformations or group them and predict only the corresponding group labels. And we have proposed three kinds of group protocol, so it just like that we have totally four kinds of different self-supervised methods, which are based on the localized rotation method, we call them 25-out-25, 25-out-5, 25-out-4, 25-out-3 respectively.

In more detail, the 25-out-25 means we predict all of the 25 transformations labels. The 25-out-5, 25-out-4, 25-out-3 means we group the 25 transformations according to the global rotation degrees, the local rotation degrees, the position of the parts that undergo local rotation respectively. And there will be 5, 4, 3 group labels to predict respectively.

We bold the value if it is the highest one in that column, we can see that the multi-task structure plus localized rotation task indeed improve the model's generalization ability since our method outperform the deep all. And our methods that are based on localized rotation methods show a better performance than the other methods in this situation, particularly the 25-out-4 gives the best average result and performs best when the target domain is Sketches and Cartoon. 25-out-3 outperform the others when the target domain is Art Painting. This result shows that the localized rotation task can help bridge the domain gap between different domains, particularly, it helps generalize the sketch domain a lot, it shows an improvement of 5 percent, may be the reason is that the sketch images are simpler than the other domain, thus it is easier for the model to overfit in the training and evaluation phased, which means that there is a bigger space for improvement in terms of ability of domain generalization. And there is little improvement in the photo domain, the reason could be that, the standard AlexNet is pretrained using photo domain images, it has already a very high ability of extracting the feature of photo images. Thus, the self-supervised task cannot help a lot.

We have also compared the training process, which is shown in figure 7.1, 7.2, 7.3. In more detail, we use an image loader that output 25% original images and transformed image with an equal probability to test the classification accuracy of each self-supervised method in different phases, for example, if the self-supervised method is 25-out-24, then the image loader will output 25% original images and 75% transformed images. Among the transformed images, the probability of each transformation is the same, which is 3.125%. And note that, we use different images sets in different phase, to be more precise, we use training, evaluation, and test set to test the accuracy in training, evaluation, test phase respectively.

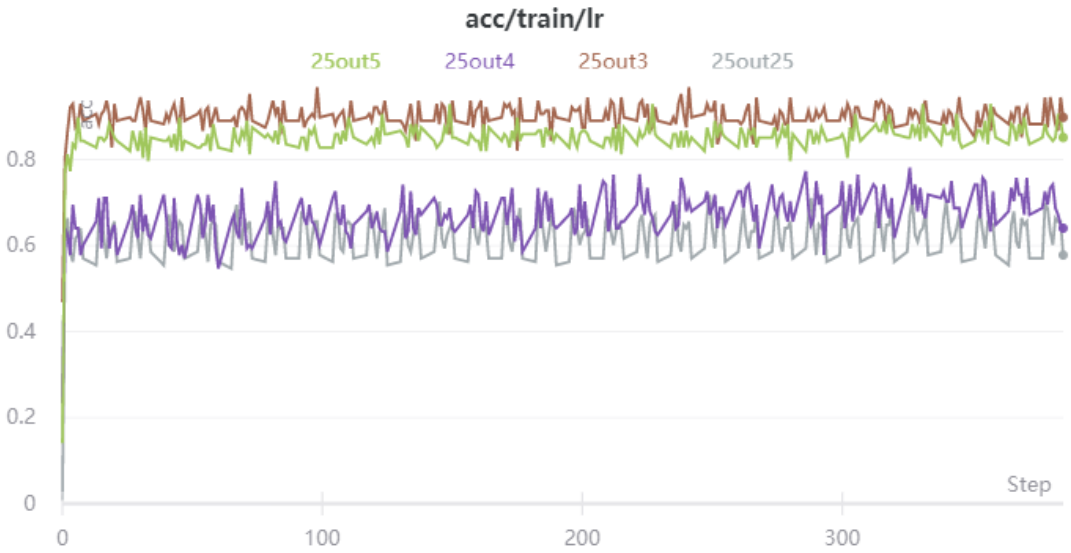

Figure 4.1 Accuracy of different self-supervised tasks in training phase

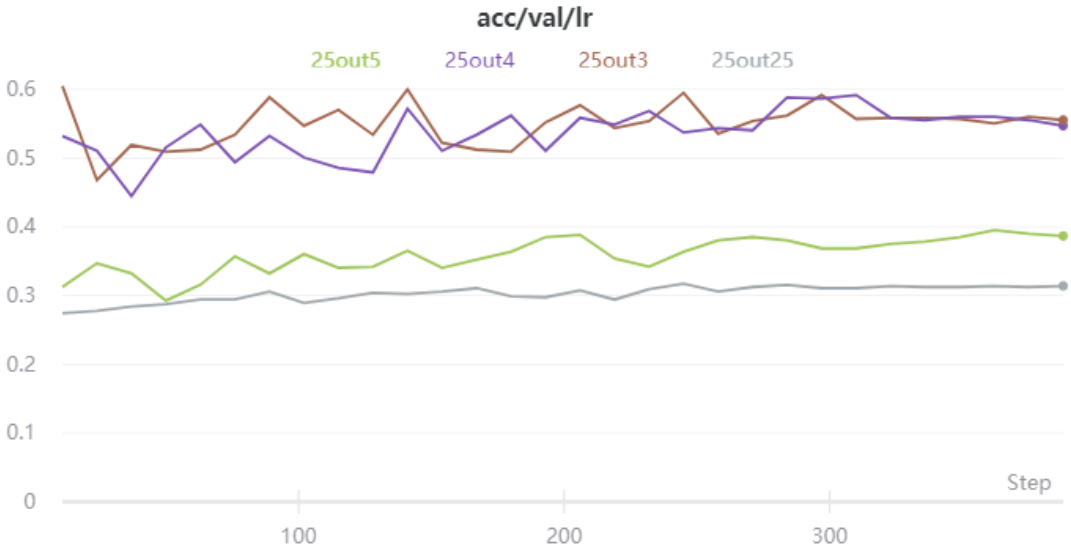

Figure 4.2 Accuracy of different self-supervised tasks in evaluation phase

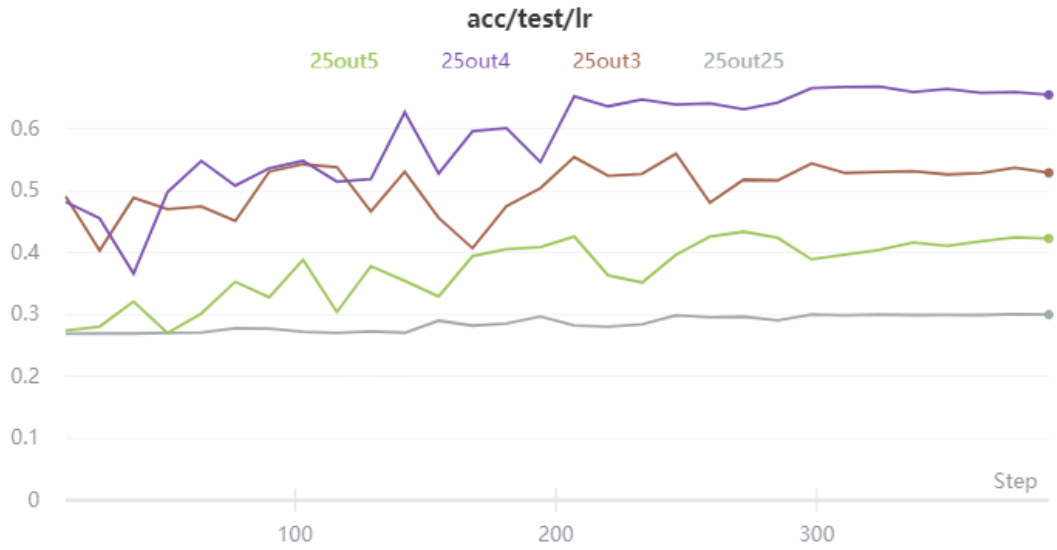

Figure 4.3 Accuracy of different self-supervised tasks in test phase

We can find the most interesting thing is that, the rank of the 25-out-4 among these four methods keep increasing if we observe the pictures according to the sequence of training phase, validation phase, and test phase. The method of the 25-out-4 shows only a middle performance in the training phase with respect to the other methods, and perform much better in validation phase, but still a little bit lower than 25-out-3, and it shows the best performance in the test phase. Which means that it is suitable to use the 25-out-4 method to help generalize cross domain.

As for the method of 25-out-3, although it is better in training phase and almost the same in evaluation phase with respect to 25-out-4, it show a worse performance in the test phase, which means that it could be easier to overfit in training phase and has a lower domain generalization ability when comparing with the method of the 25-out-4.

The 25-out-5 shows a relatively high performance in training phase, but shows a lower performance in the evaluation and test phase with respect to the other methods, which means that it can overfit in the training phase.

The 25-out-25 and doesn't give any surprise, it is always the last one at all time, maybe the reason could be that it has the most labels to predict, the model cannot cope with this problem.

We has also shown the change graph of the loss of the self-supervised task in the training phase in figure 7.4, the 25-out-25 has the highest loss, and then the second one is the 25-out-4, and then the third one is the 25-out-5, the last one is the 25-out-3, which is corresponding to the rank of the classification accuracy in training phase shown in figure 7.1.

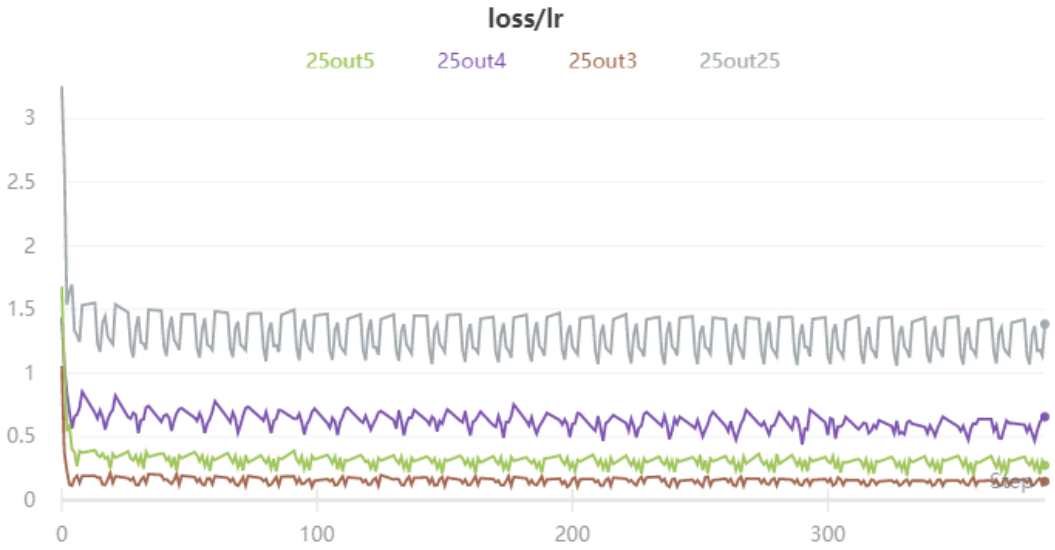

Figure 4.4 Accuracy of different self-supervised tasks in test phase

In summary, in this part we have compared the different self-supervised method, we show their training process and discuss the potential reason. The method of 25-out-4 shows a better performance in terms of preventing overfitting and domain generalization ability in this scenario.

4.5.2 Task Accuracy Comparison

We also want to know how the classification accuracy of the standard classification task improves with the self-supervised task, here we show an example experiment to demonstrate, the experiment setting is that: the self-supervised method is 25-out-4, the target domain is sketch, the self-supervised task weight is 0.3, the proportion of the original image is 0.4. We show the accuracy change of both the standard classification task and the self-supervised task in training, evaluation and test phase in figure 7.5, 7.6, 7.7, in which the solid and dash lines refer to the standard classification and the 25-out-4 method respectively. The point we want to indicate is that the accuracy of the standard classification task is indeed improving with the accuracy of 25-out-4 task. And especially, the accuracy of the localized rotation task enter into a saturate state quickly in the training and evaluation phase, and show a relatively more obvious increasing trend in the test phase, the reason could be that the standard AlexNet is so powerful that it can cope with the source domain data easily, but it is not good enough to deal with the target domain data, which is unseen during the training phase, which means a bigger space for improvement, and now the localized rotation task can show its regularity effect.

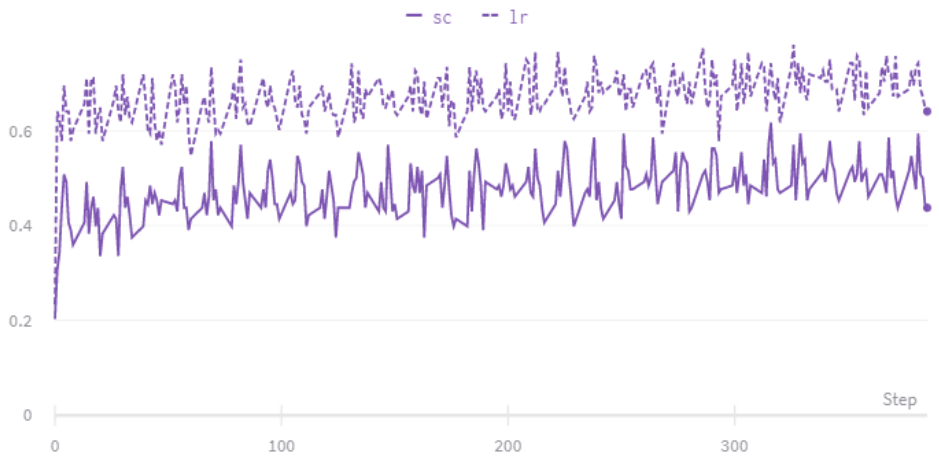

Figure 4.5 The accuracy of the standard classification task and the self-supervised task in training phase

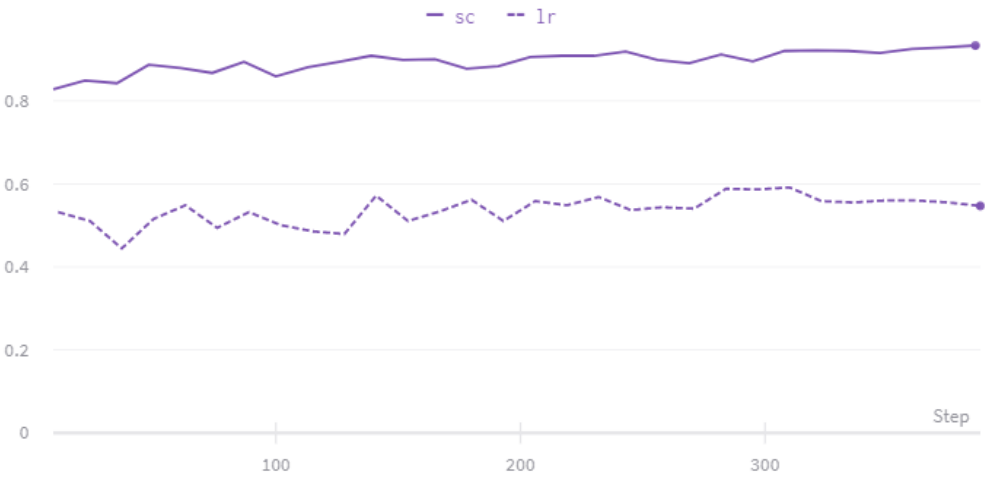

Figure 4.6 The accuracy of the standard classification task and the self-supervised task in evaluation phase

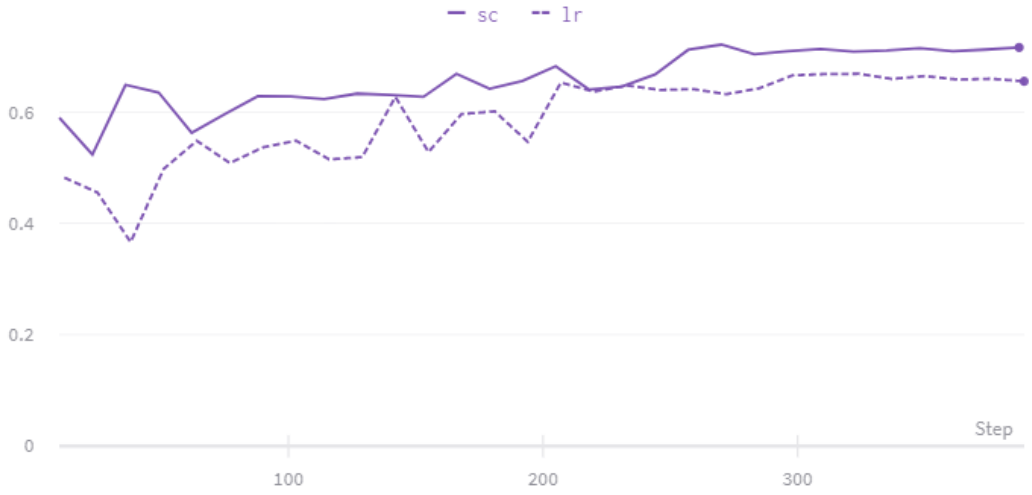

Figure 4.7 The accuracy of the standard classification task and the self-supervised task in test phase

4.5.3 Ablation Analysis

We have done an ablation experiment to analyze the contribution and importance of each hyperparameter. The experiment setting is that: The self-supervised method is 25-out-4, the target domain is sketch, each run is repeated for three times. In the figure, the orange band refers to the deep all baseline, the red line is the means value, and the value of the upper boundary is the means plus standard deviation, the value of the lower boundary is the means minus standard deviation. The blue line refers to the method of 25-out-5, it also shows the means value and the standard deviation of experiments with different hyperparameters. In more detail, the α is the localized rotation task weight, and the β is the proportion of the original images.

In the figure 7.8, we fix β=0.4 and change the value of α from 0.1 to 0.9. In figure 7.9, we fix α=0.3 and change the value of β from 0.1 to 0.9. We can observe that all the results outperform the base line. And as shown in figure 7.8 we can see that a high localized rotation task weight could lead to a high standard deviation, maybe it is because if we use less original images to train the model, the model cannot see enough original images then it cannot learn enough knowledge and then shows a non-stable performance. From figure 7.9 we can find that increase the localized rotation weight can somehow increase the performance of the model, but when the weight is bigger than 0.4, it shows a decreasing trend, we can say that a bigger localized rotation weight can let the model put more attention on the localized rotation task, which can help increase the model's ability of generalization, but a too big localized rotation task can let the model lose the main target, which is the standard classification task.

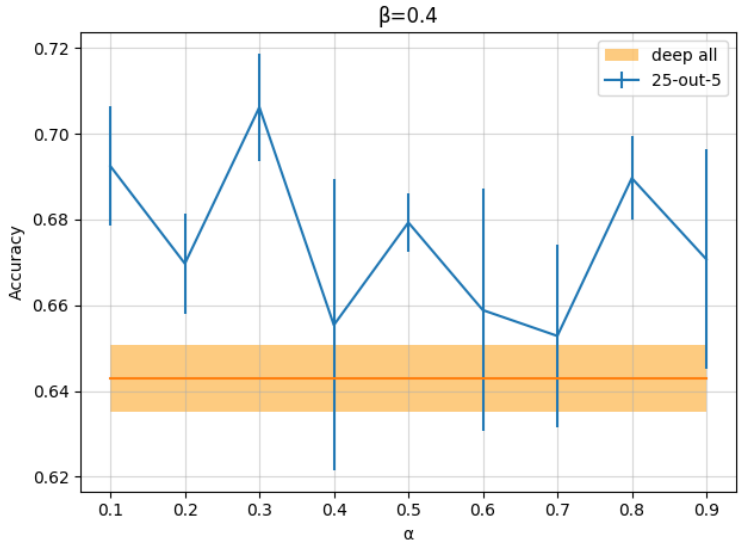

Figure 4.8

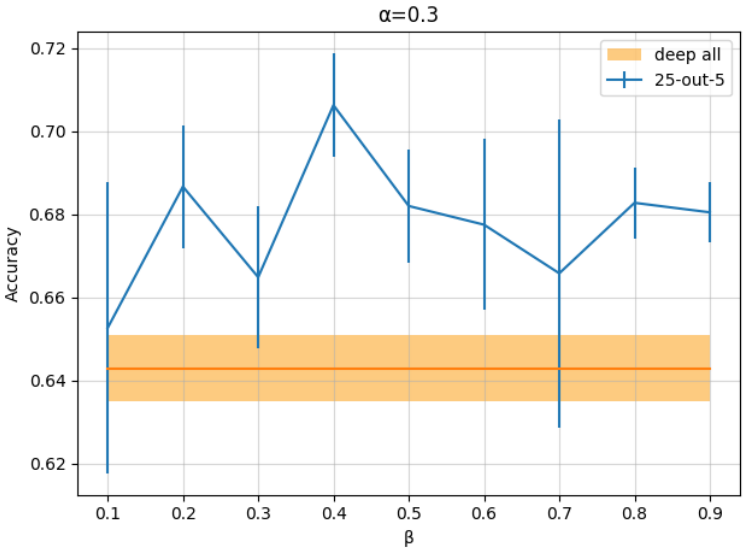

Figure 4.9

4.5.4 Confusion Matrix

In this part we focus on the performance of the model in the class level, we show the confusion matrix of the standard classification task of each kind of localized rotation method in figure 7.10 to 7.14. In more detail, the title of the figure is the method being used, the x axis is the real category, and the y axis is the predict category, the values in the cells indicate the probability of prediction of each category, for example, in the figure 7.10, the value of 17.6, which is in the third row and the first column, means that, 17.6 % of the dog images were predicted as giraffe. The confusion matrix of an ideal model should be a diagonal matrix, and the values on the diagonal line are all 100.

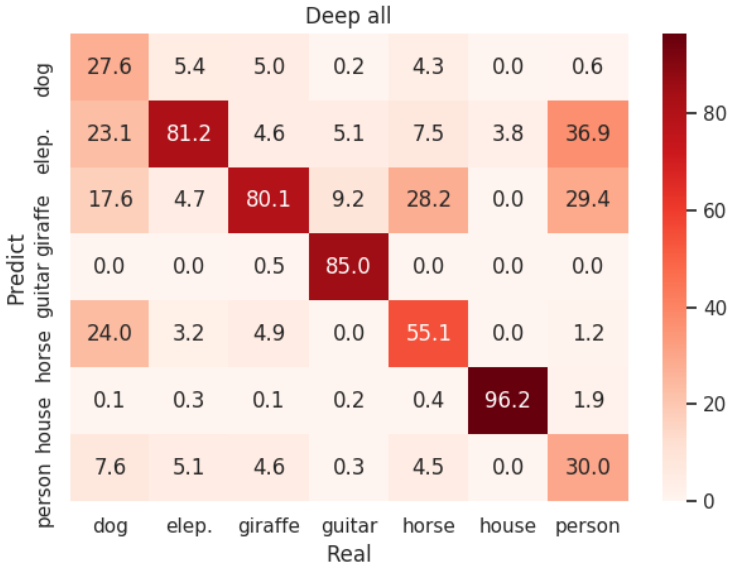

Figure 4.10 Classification accuracy of each category using deep all

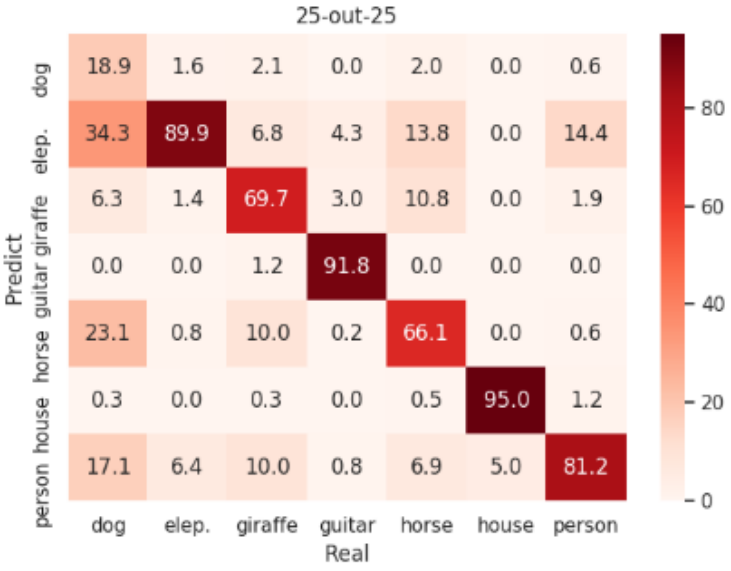

Figure 4.11 Classification accuracy of each category using 25-out-25 method

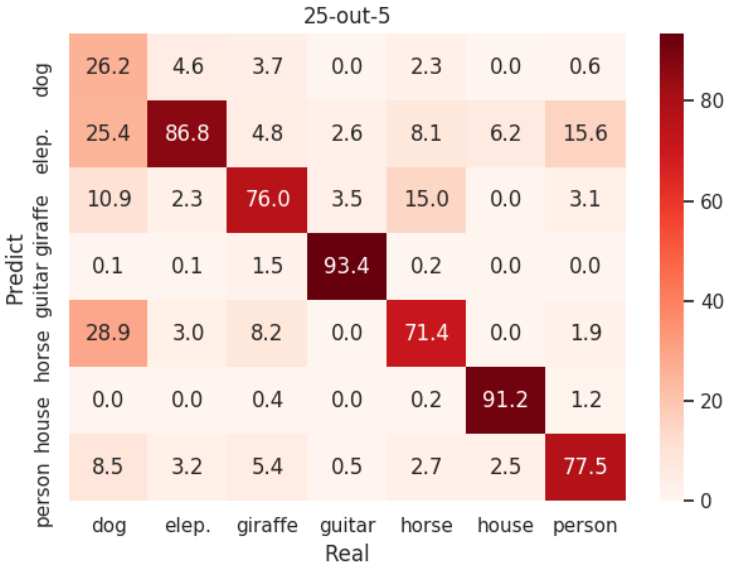

Figure 4.12 Classification accuracy of each category using 25-out-5 method

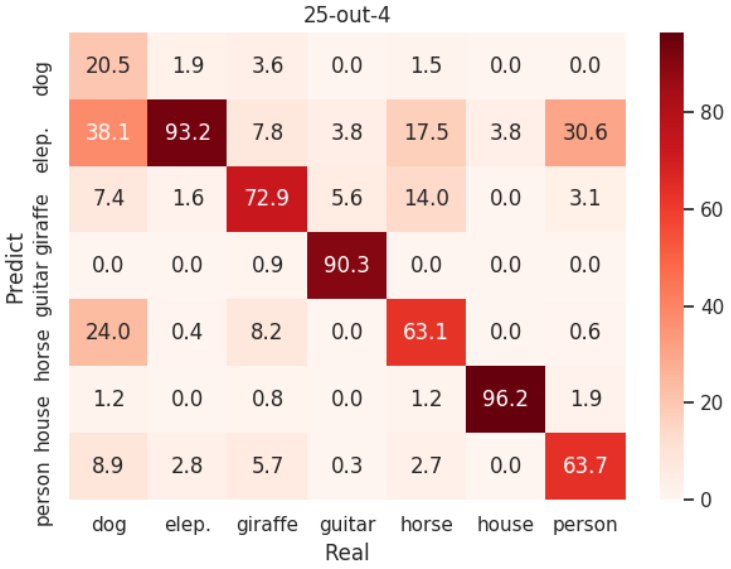

Figure 4.13 Classification accuracy of each category using 25-out-4 method

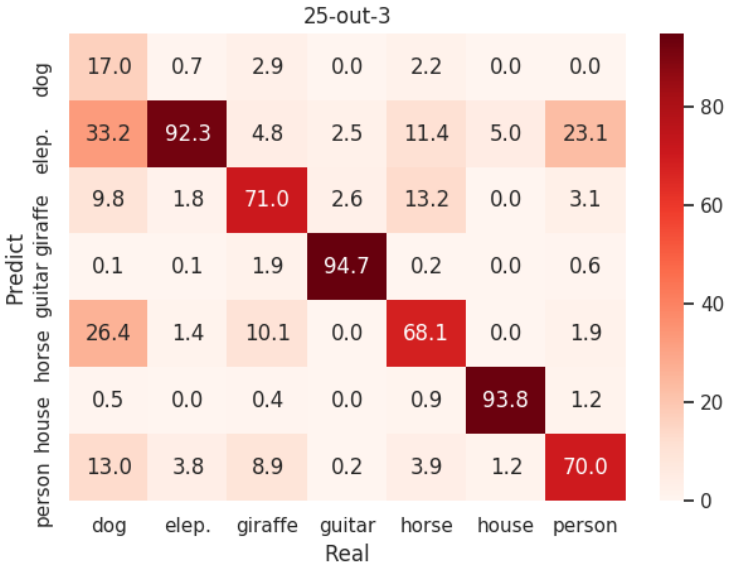

Figure 4.14 Classification accuracy of each category using 25-out-3 method

We can find that, the dog is the hardest category to generalize, and the elephant, giraffe, guitar, and house are relatively easier to generalize on average. And our self-supervised methods make different contributions to different categories, particularly, the classification accuracy of person images has the biggest improvement, which is approximately 40 percent improvement. However, the giraffe suffer a negative effect, the classification accuracy decrease approximately by 10 percent.

4.5.5 Visualization

The technique of visualizing a convolutional neural network is proposed for the first time in this paper[3]. We try to replicate some of those visualization process to get a deeper insight of our model. The model that we use to do these visualizations is trained with this experiment setting: The target domain is sketch, the self-supervised method is 25-out-4, the self-supervised task weight is 0.3, and the proportion of the original images is 0.4.

In the figure 7.15, we try to visualize the kernel of our model, what we do is: For each kernel in each layer, we split its values of weight into several 2-dimentional data, and show it as gray image.

For example, we show the kernel visualization of the first layer of our model in the left part of the figure 7.15, each row refers to a kernel, which composes three grey images, and there are 96 rows. We would like to explain more detail why we have these numbers, the backbone of our model is AlexNet, and the first layer of AlexNet composes 96 kernels with the size of 11*11*3, so for each kernel, we split the raw data into 3 pieces with the size of 11*11, each piece is a grey image, so for the first layer we have in total 96*3 pieces of image.

The second layer of AlexNet compose 256 kernels with the size of 5*5*48, so we will have 256

rows and each row compose 48 grey images with the size of 5*5. Here we only show a small part of the whole second layer in the middle of figure 7.15. It is the same story for the third layer, we show only a part of the kernels of the third layer in the right part of figure 7.15.

We can find that, in upper layer, the structure of the kernel is more complex and delicate, the kernel in the first layer would detect the simple direction, orientation and color of the image, the upper layer would detect more detailed structures, such as the eye, nose, tail or leg in the images.

Apart from the kernel, we also visualize the feature maps of the input image, we select 4 person images of sketch domain from the datasets of PACS and see each feature map. What we do is that: we unwrap the model into several layers, and apply them sequentially to the input image, we select the response data of each layer, and show them in image form. In figure 7.16, the left part is the 4 original images, and the right part is the feature maps of the first layer, since we have 96 kernels in the first layer, so we would have 96 features for each image. In the figure 7.17 we show the feature maps of the second, third, fourth, and fifth layer.

We can find that in the lower layer, the feature maps are clearer, and we can easily recognize the corresponding original image, in the deeper layer, the features are more abstract and fuzzier.

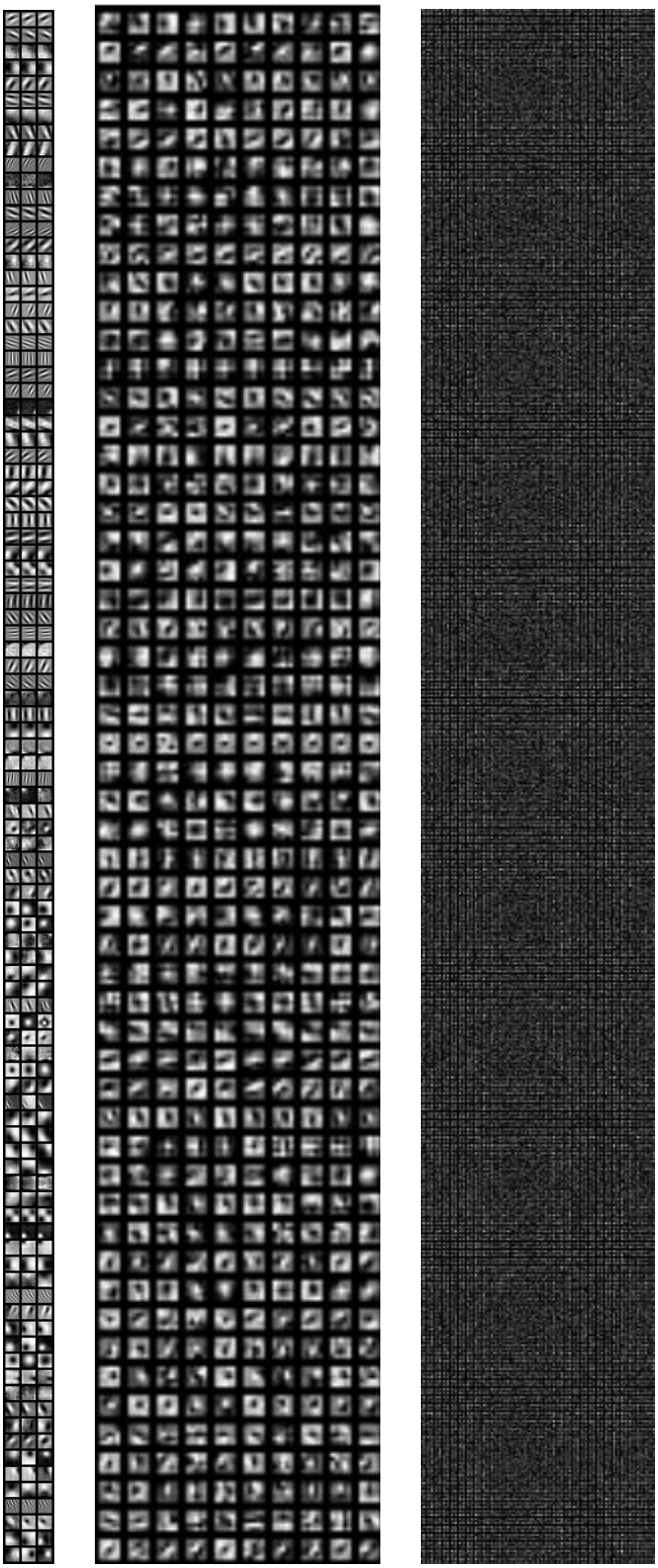

Figure 4.15 The visualization of the kernels of the first, second, third convolutional layer.

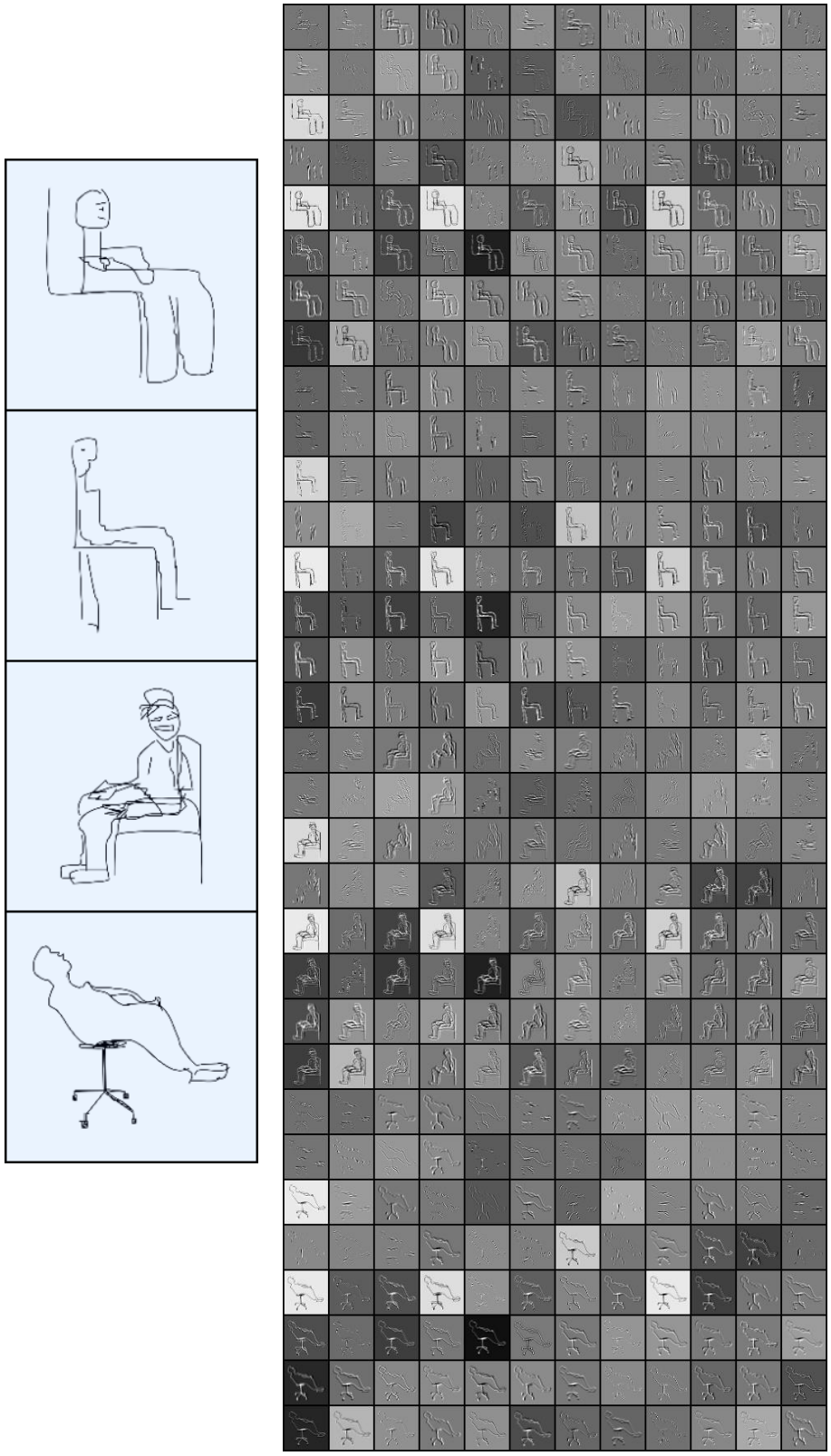

Figure 4.16 The original images and the visualization of the feature maps of the first convolutional layer.

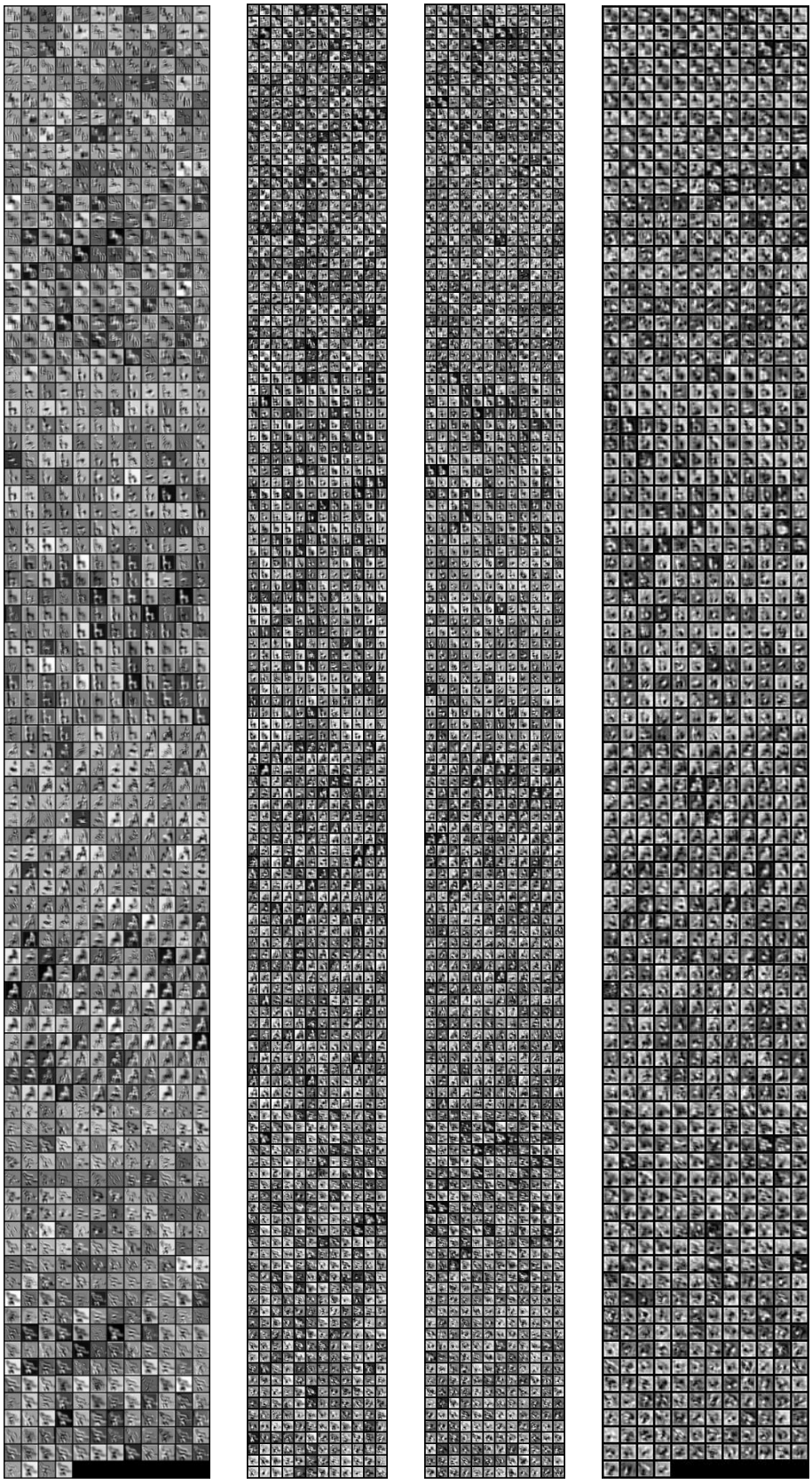

Figure 4.17 The visualization of the feature maps of the second, third, fourth, and fifth convolutional layer

4.5.6 Domains Comparison

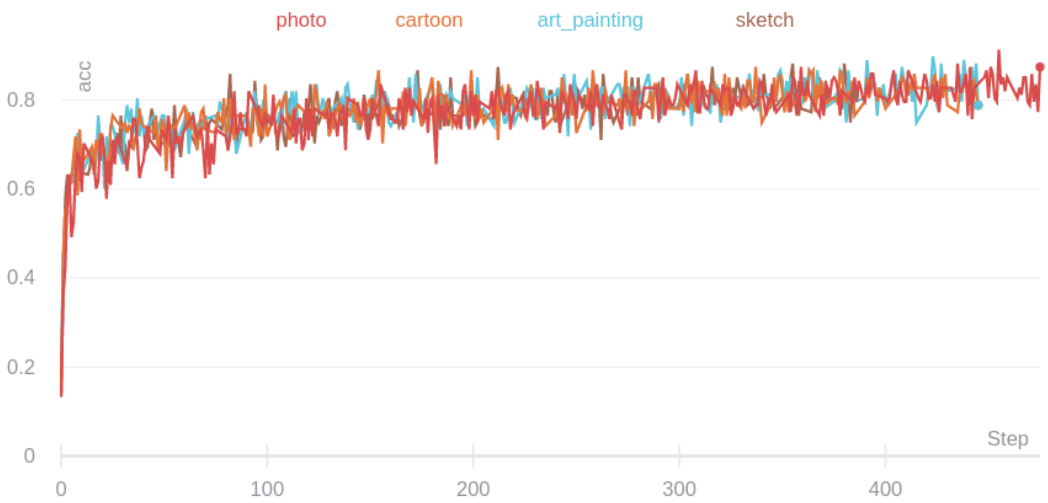

Figure 4.18 Accuracy of the standard classification task in training phase

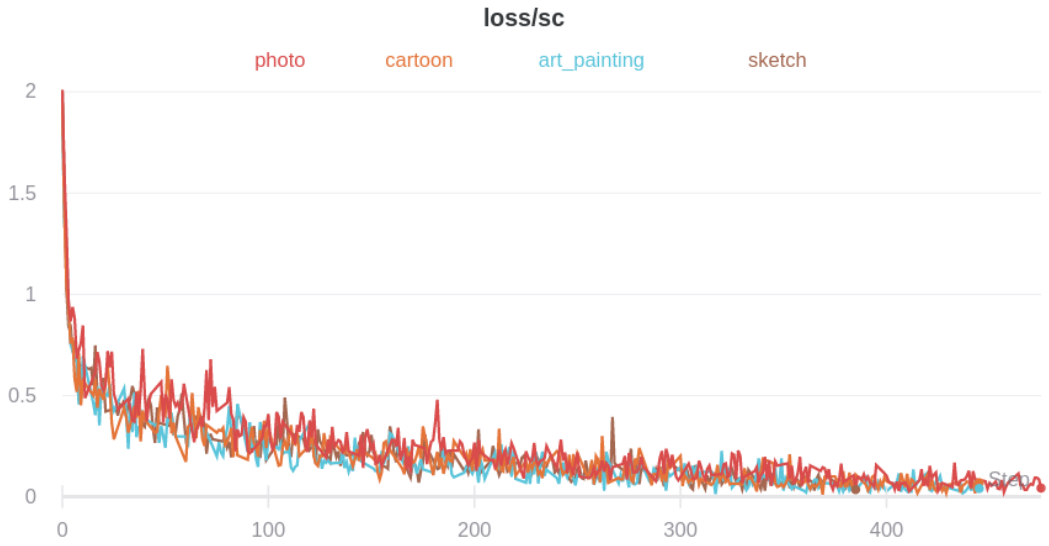

Figure 4.19 Loss of the standard classification task in training phase

In this part we compare the performance of the standard classification task when different domains is being used as the target domain. The experiment setting is that: the self-supervised method is 25-out-3, the self-supervised task weight is 0.5, the proportion of the original image is 0.8, we record the value every 5 mini-patch.

In figure 7.18, we show the change of the accuracy of the standard classification task during the

training phase. And in figure 7.19, we present the change of the loss.

We can observe that there is no big difference concerning the accuracy or loss of the standard classification task during training phase, which means that, the model work well on simulating the map relationship between the training image and the label, no matter which target domain we use.

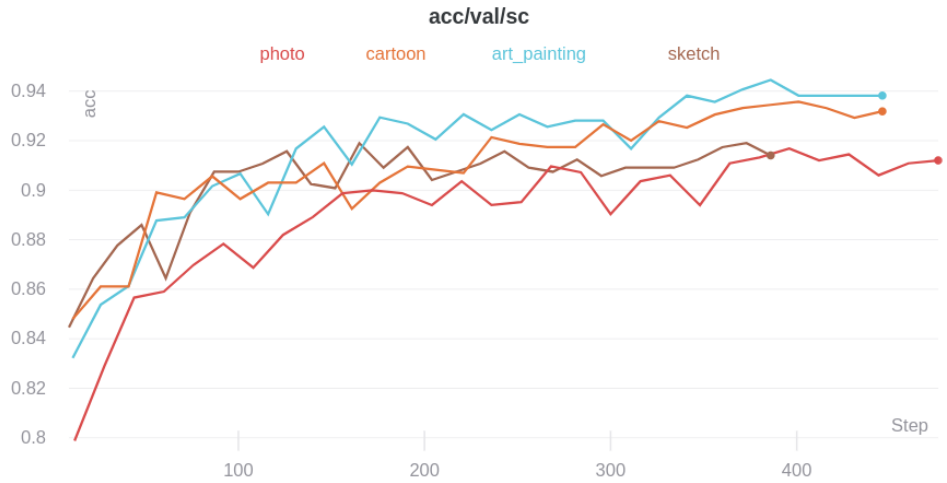

Figure 4.20 Accuracy of the standard classification task in validation phase

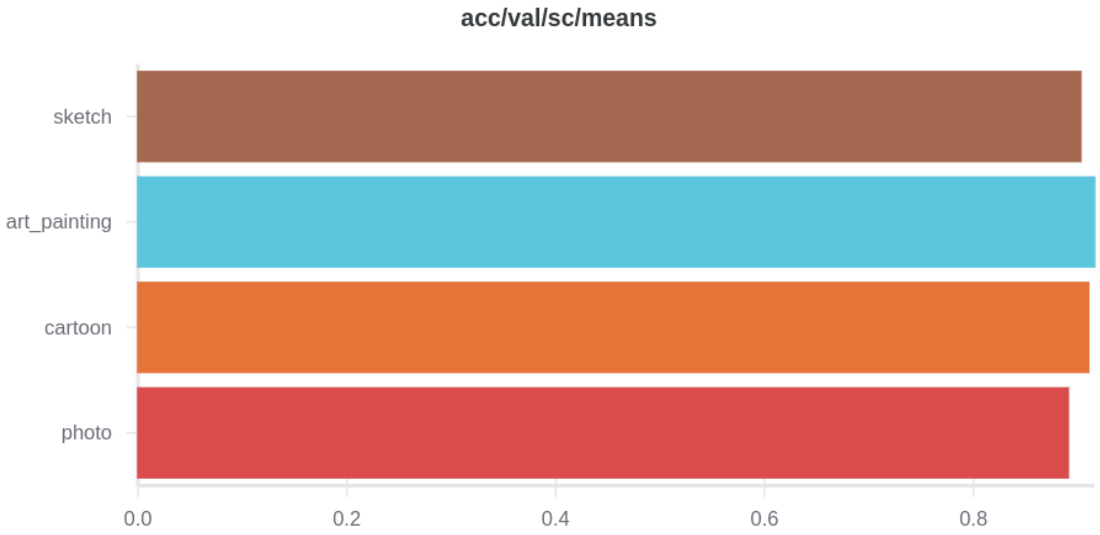

Figure 4.21 The means of accuracy of the standard classification task in validation phase

We demonstrate the accuracy change of the standard classification task during evaluation phase in figure 7.20, in order to have a more intuitive sense, we show their average accuracy in figure 7.21.

These two charts show us that, during the validation phase, the model presents only a slightly different performance when we are using different target domains. In more detail, the model shows the

best result when the target domain is Art painting, and shows the worst result when the target domain is Photo.

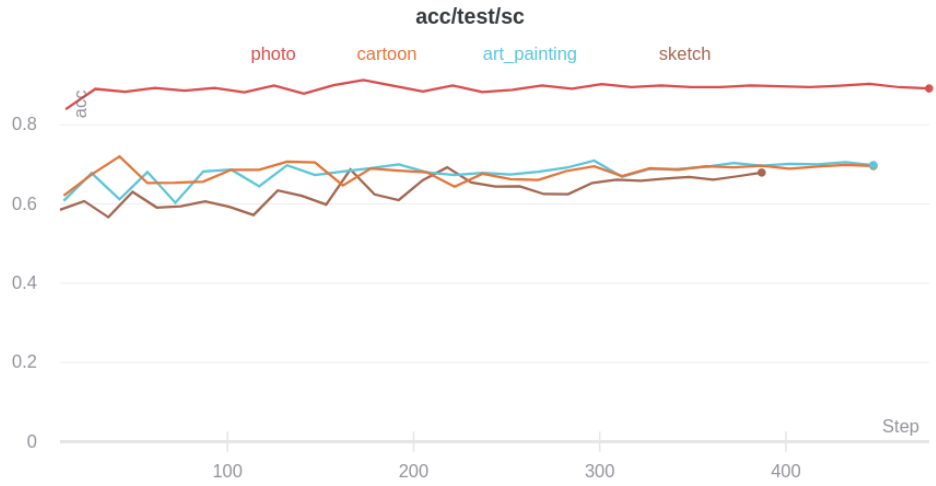

Figure 4.22 Accuracy of the standard classification task in test phase

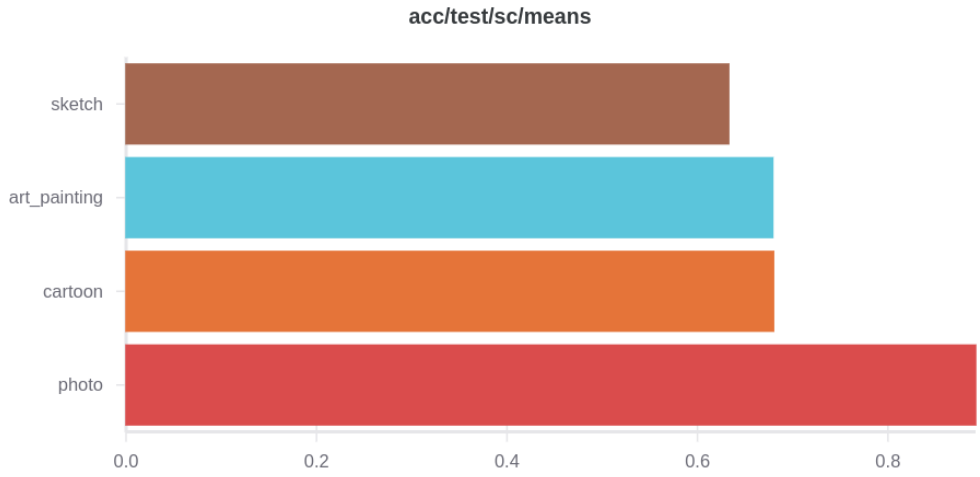

Figure 4.23 The means of accuracy of the standard classification task in test phase

We show also the change of the accuracy of the standard classification task during test phase in figure 7.22 and we show the means in figure 7.23. We can observe that the classification accuracy suffers a big drop when the target domain is Art painting, cartoon, or sketch. So, we can say that the photo domain is relatively easier to generalize with respect to the others.

5 Conclusion

In this thesis, we have studied an existing model that aims at solving the domain generalization problem by solving both supervised task and self-supervised task at the same time. We manage to replicate this model, and propose a new self-supervised method, which is a variant of the standard rotation method, we call it localized rotation method, during this process, we have learned the knowledge of multi-task learning, supervised learning, unsupervised learning, self-supervised learning, and transfer learning. We have defined our loss equation explain the detail implementation of both the standard image classification task and the localized rotation task.

We have conducted experiments to evaluate our model, particularly, we conduct the standard multi-source domain generalization experiments with our model on the datasets of PACS, which is a commonly used benchmark images datasets of domain generalization.

We have compared the performance of our model with both the deep all baseline and the other methods in an overall general view. The result show that our localized rotation method is better than the other methods in this scenario on average. But actually, we are not sure whether our method can also work well on other datasets or not, such as DomainNet, VLCS, Office Home and so on. We are not sure whether our method can work well on other setting, such as domain adaptation[17] and partial domain adaptation[18].

In order to have a deeper insight of our model, we have also observed the training process and the classification accuracy detail, we have done an ablation analysis, at last we try to visualize the model and compare the performance across domain. We can draw the conclusion that our method is effective, it can help increase the domain generalization ability of the model.

In summary, our main contribution is that we have proposed a new kind of self-supervised task, which is the localized rotation task, and we have proposed several group protocols to improve the model’s performance.

In the future, we will conduct more experiments on more datasets with different settings to validate our method. Apart from this, we would like to do more research on the reason why these kinds of methods can work, and the essential difference between the methods of 25-out-25, 25-out-5, 25-out-4, 25-out-3, and the standard rotation task. What’s more we will try more kinds of transformations and explore new group protocols. We will try to find the underlying rule and regularity in the deep model, and try to explain our model in both high and low level.

6 References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (n.d.). *ImageNet Classification with Deep Convolutional Neural Networks*. Retrieved from http://code.google.com/p/cuda-convnet/

[2] Li, F.--F., Russakovsky, O., Deng, J., Huang, Z., Berg, A., & Fei, L. (n.d.). *Analysis of Large Scale Visual Recognition*.

[3] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8689 LNCS(PART 1), 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

[4] Simonyan, K., & Zisserman, A. (2015). *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. Retrieved from http://www.robots.ox.ac.uk/

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 07-12-June-2015, pp. 1–9). IEEE Computer Society. https://doi.org/10.1109/CVPR.2015.7298594

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 2016-December, pp. 770–778). IEEE Computer Society. https://doi.org/10.1109/CVPR.2016.90

[7] Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2017). Squeeze-and-Excitation Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 7132–7141. Retrieved from http://arxiv.org/abs/1709.01507

[8] Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). *Domain Generalization via Invariant Feature Representation*.

[9] Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., & Tommasi, T. (2019). Domain generalization by solving jigsaw puzzles. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 2224–2233. https://doi.org/10.1109/CVPR.2019.00233

[10] Jenni, S., Jin, H., & Favaro, P. (2020). *Steering Self-Supervised Feature Learning Beyond Local Pixel Statistics*. 1–13.

[11] Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., & Cord, M. (2020). *Learning Representations by Predicting Bags of Visual Words*. http://arxiv.org/abs/2002.12247

[12] Huh, M., Agrawal, P., & Efros, A. A. (n.d.). *What makes ImageNet good for* transfer learning?

[13] Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (n.d.). *Deeper, Broader and Artier Domain Generalization*.

[14] Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised Representation Learning by Predicting Image Rotations. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. Retrieved from http://arxiv.org/abs/1803.07728

[15] Doersch, C., Zisserman, A., & Deepmind, †. (n.d.). *Multi-task Self-Supervised Visual Learning*.

[16] Huh, M., Agrawal, P., & Efros, A. A. (n.d.). *What makes ImageNet good for transfer learning?*

[17] Wilson, G., & Cook, D. J. (n.d.). *A Survey of Unsupervised Deep Domain Adaptation*

[18] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (n.d.). *ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from http://code.google.com/p/cuda-convnet/*

装

订

线

7 Acknowledgement

The bachelor thesis is just like the symbol of the ending of my university period, during which, I have learned a lot, experienced a lot. I show the knowledge that I have learned in this thesis, what I have experienced will shine the road to the future. Here, firstly I would thank my parents, who support my decision always, they are my solid shield. Then I would thank professor Tatiana, who gives me the chance of internship, leads me to the field of deep learning, she is kind to me. And I would thank Silvia, who is always willing to tell me everything about deep learning. Then I would thank professor Kangli and Yan, who give me suggestions on writing the thesis, who answer my questions, who help me with the study and everything else, who give me warm. At last I would thank everybody who has ever given me suggestion, who has ever comforted me, who has ever shared feeling or information with me, who has ever listened to my words. It is all of you that give me power, strength, courage, motivation and passion to continue to explore the life.