上海大学

SHANGHAIUNIVERSITY

大数据：从理论到实践A（论文）

Big Data: From theory to practice (A)(THESIS)

题目：基于自然语言监督信号训练视觉模型

学院	计算机工程与科学学院
专业	智能科学与技术
学号	**20123101**
学生姓名	李昀哲
指导教师	王健嘉
起讫日期	**2022.05.24– 2022.05.31**

目录

摘要.....II
ABSTRACT..... III
第 1 章 论文解读..... 1
§ 1.1 研究背景.....1
§ 1.2 研究方法.....2
§ 1.2.1 自然语言监督信号.....2
§ 1.2.2 创建数据集.....2
§ 1.2.3 选择高效的预训练方法.....3
§ 1.2.4 模型.....3
§ 1.3 实验.....5
§ 1.3.1 零次学习（Zero-Shot)迁移.....5
§ 1.3.2 和 N-GRAMS 的对比.....5
§ 1.3.3 Prompt Engineering.....6
§ 1.3.4 27 个数据集的实验结果.....7
§ 1.4 局限性.....9
第 2 章 论文实验结果验证.....11
§ 2.1 不同模型同一图片、给定类，分类精度验证.....11
§ 2.2 同一模型、图片不同给定类，分类准确度验证.....12
第 3 章 总结与展望..... 13
§ 3.1 本文总结.....13
§ 3.2 展望.....13
致谢..... 14
参考文献..... 15
附录：部分源程序清单..... 16

基于自然语言监督信号训练视觉模型

摘要

本文灵感来源于：近期对 NLP 和 CV 领域创新项目的学习中阅读到的一篇论文 Learning Transferable Visual Models From Natural Language Supervision[1]，发表于 2021 年 2 月 26 日。

由于在过去几年里，NLP 领域取得了巨大的成果，BERT、GPT 相继问世，引发了研究人员的一种想法——将 NLP 领域中“训练模式和下游任务分开”的方式应用到其他领域，这篇论文就开创性地将这种方法复制到了 CV 领域：用自然语言的监督信号，采用对比学习训练视觉模型。作者将这种模型命名为 CLIP（Contrastive Language-Image Pre-Training）。CLIP 具有良好的泛化性，同时能在零次学习（Zero-Shot）的情况下，对视觉的图像数据有较好的分类效果。在 ImageNet 的数据集上，CLIP 可以在不用一张有标签数据的情况下，和有监督学习的 ResNet-50 达到相同的效果。采用的方法出奇的简单，但效果却出奇的好。

本文就将对这篇具有开创性的论文用自己读后的观点进行解读：将从论文发表背景、研究采用的方法、研究过程中的实验、以及模型的局限等多个角度进行；同时将对论文中的实验用多种模型进行验证，最后进行总结和对这种开创性方法的展望。

关键词：自然语言监督信号，对比学习，视觉模型，CLIP，数据分类

Training Visual Models From Natural Language Supervision

ABSTRACT

Inspiration of this article comes from a paper (Learning Transferable Visual Models From Natural Language Supervision), published on February 26, 2021.

Due to the great achievements in the NLP field in the past few years, BERT and GPT came out one after another, which gave rise to the researchers' idea of applying the "training mode and downstream task separation" approach in the NLP field to other fields, and this paper pioneered the replication of this approach in the CV field: The visual model was trained by contrastive learning using natural language supervisions. The authors named this model CLIP(Contrastive Language-Image Pre-Training). CLIP has good generalization and can classify visual image data in Zero-Shot mode. On ImageNet data sets, CLIP can achieve the same effect as supervised learning ResNet-50 without using a single piece of labeled data. The method is surprisingly simple, but the results are surprisingly fabulous.

In this article, I will interpret this groundbreaking paper from my own point of view and from many aspects like the publication background, research methods, experiments in the research process, as well as the limitations of the model, etc. After that, the experiments in this paper are verified by a variety of models. Finally, a summary and the prospect of this pioneering method are given.

Keywords: Natural Language Supervisions,Contrastive Learning,Visual Models, CLIP, Data Classification

第 1 章 论文解读

本章主要对 Learning Transferable Visual Models From Natural Language Supervisions 进行解读，将结合阅读后的观点介绍这项研究的背景、方法、实验以及局限性，进而对这项研究的内容和目的有整体的了解。

§1.1 研究背景

直接从原始的文本中预训练模型已经在过去几年的 NLP 领域中，取得了革命性的成果，比如：BERT[2]，GPT 等。无论是自回归（auto-regressive）预测还是用掩码的完形填空的方式，都是自监督的训练模式，即：目标函数和下游任务是无关的，只是通过预训练得到一个泛化特征特别好的模型。随着计算能力的增强以及数据量的增多，模型的性能也在稳步提升，但这一套系统只是“文字进，文字出”（text-to-text)，并不是一个特殊的分类任务，模型架构也是和下游任务无关的。因此在研究下游任务时，不需要针对某个数据集进行特殊处理。比如 OpenAI 自己的 GPT[3]模型就是这样的效果，能完成分类、翻译、写邮件等任务，且不需要或只需要很少特定领域的数据做微调。

这些结果表明了在文本领域，利用自监督的信号训练的模型框架下，这种大规模没有标注的数据集甚至能比高质量标注过的数据集达到更好的效果。但在视觉领域，一般的做法还是在 ImageNet[4]数据集上预训练一个模型。这就会让训练好的模型有诸多的限制那能否将 NLP 领域里的框架用在视觉领域里呢？过去 20 年中，相关工作有大跨度的进展，论文作者团队开发的 CLIP（Contrastive Language-Image Pre-Training）模型的工作和 Li et al(2017)[5]特别相似，都用了零次学习（Zero-Shot），但在 2017 年 Transformer 并未问世，且并没有大规模的数据集，因此这项研究的效果并不好。而 VirTex[6]，ICMLM 和 ConVIRT 等在有了强大的对比学习工具后，基于 Transformer 尝试过这种迁移学习的方法，和 CLIP 相似，不过在具体做法上还是存在区别的。VirTex 采用自回归的预测方式做预训练，ICMLM 用完型填空方式做预训练，ConVIRT 和 CLIP 最为接近，但只在医疗图像上做了测试。因此，由于没有大量的数据集、没有好的自监督模型，导致精度很低，并没有引发人们关注。

这些研究用文本的弱监督信号来训练有监督的模型进行分类，避免了使用有限的高质量的标注数据且提升了一定的精度，但分类的类别仍然有限，识别到的类别是固定的，一旦有新的类别，就无能为力了。因此没有灵活的、做零次学习（Zero-Shot）的能力。

这些研究的方法和 CLIP 其实相差不大，但最大的区别是在数据集的规模上，这也是 OpenAI 团队在这项研究中有意去克服和优化的方向，他们构建了一个四亿数据量的数据集，模型的尝试上，选择了 8 种模型，从 ResNet 到 Vision Transformer。测试发现，精度和模型大小基本是正相关的，在第二章也会对这项结果进行验证。因此使用 CLIP 可以根据自己采用的模型大小，大概估算出迁移学习的效果，这是一个很实用的性质。在大数据集和大模型的加持下，CLIP 迁移学习的效果相当出色，且泛化性很好。

§1.2 研究方法

§1.2.1 自然语言监督信号

该团队方法的核心思路是利用自然语言的监督信号，根据前人的工作不难发现，这种思路其实并不是第一次提出，但之前的方法对于用词和用语有些不妥。例如 Zhang et al(2020)[7], Gomez et al(2017)[8]以及 Joulin et al(2016)[9]等人都是用到了“文本-图片”配对的方式，但他们却将这种方法描述为“无监督的”、“自监督的”和“弱监督的”。本论文作者的工作无非就是总结了前人的经验和方法，并扩大了模型和数据集的规模。核心仍旧是将文本作为训练的信号。

在 Transformer 出现之前，NLP 的模型其实并不好学，随着上下文具有语义环境的学习方式的发展，比如 BERT，使得在自监督的模式之下，文本方面的监督信号拥有很多的信息资源，所以 NLP 训练出来的模型变得又大又好，简单、泛化性强，很适合多模态的学习。

那为什么使用 NLP 的方法来做视觉的任务呢？首先，不用标注，只需要下载或爬取网络上“文本-图片”的配对，数据集是很容易得到的；其次，文本和视觉特征绑定在一起，学习到的特征也得到了多个维度、多模态，方便做零次学习（Zero-Shot）的迁移。但毫无疑问，这样的方式需要足够大的数据集。

§1.2.2 创建数据集

在 1.2.1 节末尾提到，对于已有的数据集来说，大规模的数据集存在“文不对图”的情况，例如一些图片的文本信息为相机的各类参数等，显然不是在描述图片内容，而进行清洗过后，原本数据集的规模又将大幅下降，最终的大小都和 ImageNet 数据集大小类似，但这样的数据集规模是不够的。

因此，作者团队创建了一个足够大的数据集，收集了四亿个“文本-图片”的配对，相比于视觉领域 Google 的 GFT 还多一个亿的数据，和 NLP 领域的 GPT-2 差不多大。

§1.2.3 选择高效的预训练方法

视觉领域的模型都非常大，训练的代价是很高的，对于 ImageNet 来说，还只是预测 1000 个类，就已经很耗费资源了。对于想要达到开放视觉的分类，训练这样的系统即使是拥有大量计算资源的团队，例如本论文的 OpenAI，都认为是非常惊人的训练量，是不切实际的。

因此，该团队做了几个尝试，首先对于图像使用 CNN，对于文本使用 Transformer，通过图片来预测文本。但这种方法的问题在于，对于一个场景会有很多不同的解释，比如图片是一个人在打字，可能会解释为“这个人在写论文”，也可能解释为“这个人穿着蓝色的衣服”等等，会产生太多的可能性，训练就会非常慢。因此，该团队尝试了对比学习的方法，只需要判断图片和文本是不是一个配对，不需要去逐字逐句预测文本了，预测型的目标函数替换为了对比型的目标函数，效率直接提升了四倍，使用到了对比学习，因此命名为 Contrastive Image-Language Pre-Training，对比图片和文本的预训练模型。如图 1 所示为各种尝试的性能对比。

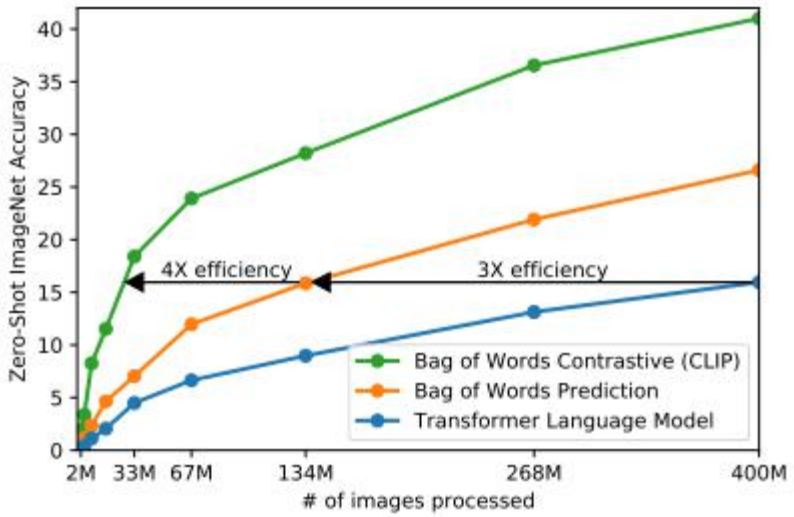


图 1 零次学习（Zero-Shot）下 CLIP 和其他方式的性能对比

§1.2.4 模型

模型的伪代码如下代码块所示，思路为：需要两个编码器，图像可以是 ResNet 或 Vision Transformer，文本可以是 Text Transformer。输入图片和文本，通过两个编码器，得到相应特征；再进行多模态合并，归一化处理；再对特征计算 cosine 相似度，将用于分类，其中的 t 是学习参数，是需要调整的；最后进行交叉熵计算损失函数。

```
# image_encoder              - 图片编码器
# text_encoder               - 文本编码器
# ImageInput[n, h, w, c]     - 一个批量的图片，[数量,高，宽，通道数]
# TextInput[n, l]            - 一个批量的文本，[数量，文本长度]
# W_image[d_image, d_e]      - 图像嵌入投射层
# W_text[d_text, d_e]        - 文本嵌入投射层
# t                          - 可学习的参数

# 提取特征
Image_feature = image_encoder(ImageInput) # [n, d_image]
Text_feature  = text_encoder(TextInput)   # [n, d_text]
# 多模态合并,归一化
Image_e = l2_normalize(np.dot(Image_feature, W_image), axis=1)
Text_e  = l2_normalize(np.dot(Text_feature , W_text),  axis=1)
# 计算特征相似度，用于分类
Logits = np.dot(Image_e, Text_e.T) * np.exp(t)

# 对称损失函数
labels = np.arange(n)
Loss_image = cross_entropy_loss(Logits, labels, axis=0)
Loss_text  = cross_entropy_loss(Logits, labels, axis=1)
Loss       = (Loss_image + Loss_text) / 2
```

由于本研究创建的数据集很大，所以训练本身不会有过拟合的问题。对于投射层的选择，并没有使用非线性的投射层。在对比学习中，非线性投射往往会比线性投射效果高 10 个百分点；而对于多模态而言，线性和非线性投射没太大影响，非线性投射只是用来适配图片单模态学习的。

模型训练方面，作者在视觉模型上训练了 8 个模型，每个模型基于 Adam 优化器（Kingma & Ba, 2014)[10]训练了 32 个 epochs，同时，由于数据集很大，训练十分耗时，不好调参，超参搜索过程都是在最小的 ResNet-50 上进行。最小的批量设置为 32,768，非常大，因此使用了很多优化手段，如混精度训练（Micikevicius et al., 2017)[11]、梯度检查（Griewank & Walther,2000;Chen et al., 2016)[12]、半精度 Adam 统计(Dhariwal et al,2020)[13]等等。

§1.3 实验

§1.3.1 零次学习（Zero-Shot)迁移

只训练一个模型，之后就不再训练、不再微调，就是作者研究零次学习（Zero-Shot）迁移的动机。之前各种自监督或者无监督的方法，主要研究的是特征学习的能力，他们的目标是去学一种泛化性比较好的特征。但即使学到了很好的特征，应用到下一个任务时，仍然需要有标签的数据集去做微调，就会牵扯各种各样的问题：下游任务不好去收集数据等。

然而，一旦借助文本训练了一个又大又好的模型，就可以用这个文本作为引导，去灵活地做这种零次学习（Zero-Shot）的迁移。至少在分类上效果都非常好。

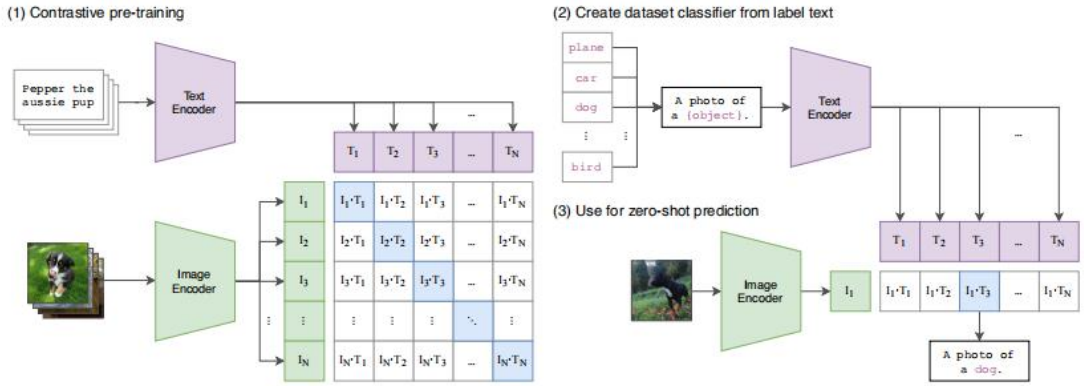


图 2 CLIP 基本流程

CLIP 实现零次学习（Zero-Shot）的方式，可以通过图 2 的(2), (3)两步来解释。图片通过图片编码器，得到一个图片的特征；文本的输入，则是用户希望识别到的类别（不同于 ImageNet 固定的 1000 个类）。通过 Prompt Engineering（后文会详细介绍，这里仅需知道是将单词变成句子），这些文本会变为句子，比如“汽车”变为“这是一张汽车的照片”。输入几个单词就会变成几个句子，这些句子通过文本编码器，得到相应数量的文本特征，文本特征和图像特征计算相似度，相似度再通过一层 softmax，得到一个概率分布，哪一个的概率最大，相似度就最高，对应的句子大概率就是在描述这种图片。以 ImageNet 为例，有 1000 个类，就会生成 1000 个句子，相当于每输入一个图片，都会用这 1000 个句子去问它，看和哪个文本最接近就是哪类。同时这个过程并不是顺次进行，而是批次进行的，所以推理是很高效的。

§1.3.2 和 N-GRAMS 的对比

N-GRAMS 模型就是前文提到和 CLIP 最相似的研究，通过表 1 可以看出 N-GRAMS 在 ImageNet 上仅有 11.5%的准确率，而 CLIP 提升到了 76.2%，这一

结果和原版的是用 128 万数据样本进行训练的 ResNet-50 达到了同样的效果。

表 1 CLIP 和先前零次学习（Zero-Shot）迁移模型的对比

	aYahoo	ImageNet	SUN
Visual N-Grams	72.4	11.5	23.0
CLIP	**98.4**	**76.2**	**58.5**

但这种对比并不完全公平，作者使用的数据集比 Visual N-Grams 大了 10 倍，视觉方面的模型也比它大了 100 倍的计算能力，所有相当于训练上用了超过 1000 倍的资源去训练，架构上也用了 2017 年 Visual N-Grams 发表是没有提出的 Transformer。因此作者也表达了对此前这项工作的尊重。

§1.3.3 Prompt Engineering

本节将介绍在 1.3.1 中提到 Prompt Engineering，这是在推理和微调时采用的一种方法，而不是在预训练阶段，所以并不需要很多的计算资源。这项技术顾名思义，起到了文本的提示和引导作用。那为什么需要文本引导的工作呢?

首先是文本的多义性，只用一个单词对应就会歧义。比如 ImageNets 中 construction cranes 是建筑中的起重机，而单单一个 crane 是鹤；再比如 remote，数据集中是遥控器的意思，但也有“遥远的”之意。如不加引导的输入，这样计算出的相似度就会有问题；其次，在预训练时，匹配的文本都是一个句子，很少是一个单词，所以通过 prompt 就可以使输入仅为一个单词时，转换为句子，使输入的分布差（distribution gap）得到匹配，使抽取到的特征更好。

```
imagenet_templates = [
    'a bad photo of a {}.',
    'a photo of many {}.',
    'a sculpture of a {}.',
    'a photo of the hard to see {}.',
    'a low resolution photo of the {}.',
    'a rendering of a {}.',
    'graffiti of a {}.',
    'a bad photo of the {}.',
    'a cropped photo of the {}.',
    'a tattoo of a {}.',
    'the embroidered {}.',
    'a photo of a hard to see {}.',
    'a bright photo of a {}.',
    'a photo of a clean {}.',
    'a photo of a dirty {}.',
    'a dark photo of the {}.',
    'a drawing of a {}.',
    'a photo of my {}.',
    'the plastic {}.',
    'a photo of the cool {}.',
    'a close-up photo of a {}.',
    'a black and white photo of the {}.',
    'a painting of the {}.',
    'a painting of a {}.',
    'a pixelated photo of the {}.',
    'a sculpture of the {}.',
    'a bright photo of the {}.',
    'a cropped photo of a {}.',
    'a plastic {}.',
    'a photo of the dirty {}.',
    'a jpeg corrupted photo of a {}.',
    'a blurry photo of the {}.',
    'a photo of the {}.',
    'a good photo of the {}.',
    'a rendering of the {}.',
    'a {} in a video game.',
```

图 3 Prompt Engineering 模板句子

作者使用的处理方法是：将单词 prompt 为“a photo of a {label}.”，使这个 label 一定使名词，虽然很简单粗暴，但是很好用，做出这个修改，准确度就提升了 1.3%，比如上述的“remote”变为“a photo of a remote”，这就消除了多义词的问题。同时，还能根据不同的数据集修改提示的句子，达到缩小选择空间的目的，比如：在“宠物”的数据集上测试，那就可以改为“a photo of a {label}, a type of pet”。作者还采用了多个（开源代码中是 80 个）提示模板联合分类，如图 3 所示，列出了部分模板。作者的意图是尽可能的包含所有的可能以提升准确率。

§1.3.4 27 个数据集的实验结果

图 4 展示了将有监督学习的 ResNet-50 作为基线，同零次学习（Zero-Shot）的 CLIP 进行 27 个数据集上的对比。可以看出，大多数结果超过了有监督学习的 ResNet-50。通过对数据集的调研发现：对于普通的对物体分类的数据集，效果比较好。但对于更难的数据集（如对纹理、对图中物体计数等）效果并不好。我认为，对特别难的任务，零次学习（Zero-Shot）的迁移就太过苛刻了，对于人来说，没有先验知识都很难做。

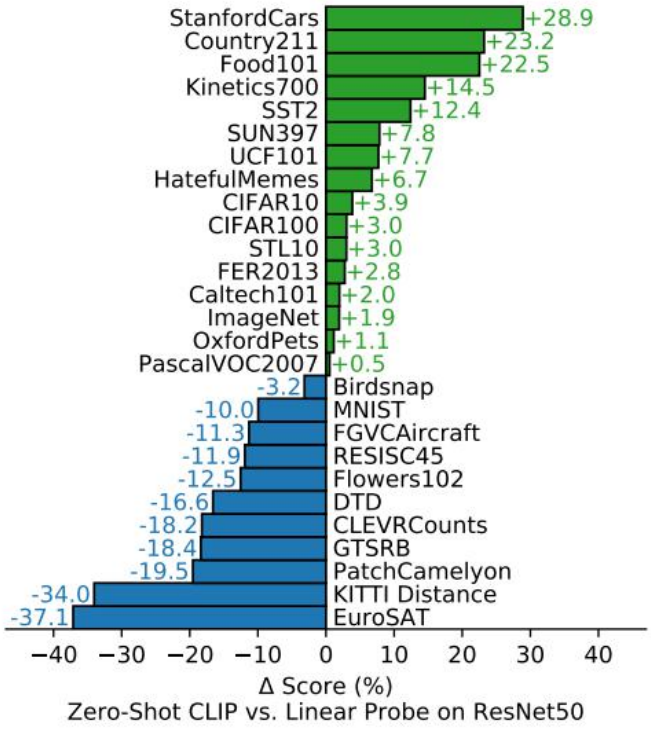
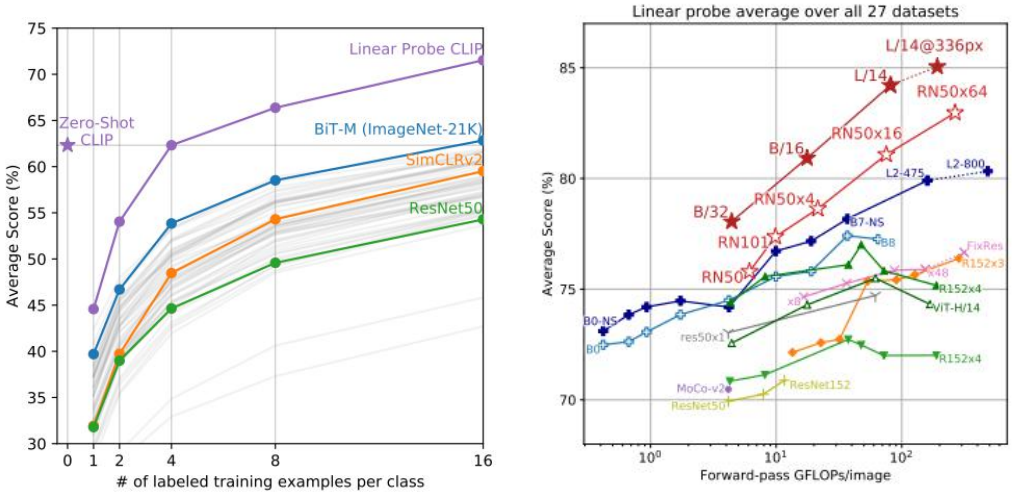


图 4 零次学习（Zero-Shot）的 CLIP 模型和全监督训练的基线模型对比

由于作者认为没有先验知识处理困难的问题，对 CLIP 有点强人所难，因此还进行了少量学习（Few-Shots）的测试，20 个数据集上合并的测试结果如图 5(a) 所示，对比了在各个 shot 下的表现。可以看出对于一些比较难的数据集，先验知

识，即 Few-Shots 是很有必要的。同时，作者还测试了用下游任务的所有数据集进行训练的效果，结果如图 5 (b)所示。由此得出：不仅在 Zero-Shot 下，Few-Shots 和所有数据下 CLIP 都完胜其他模型。



(A) Few-Shots 下对比　　(B) 全数据下对比

图 5 CLIP 和其他模型在各数据集上性能对比

在 Zero-Shot, Few-Shotss 和所有数据集上的测试对比工作完成后，基本衡量了模型的好坏，作者还通过实验，衡量了模型的泛化性。如图 6 可以看出，在数据有偏移时，CLIP 仍旧十分稳健，而普通模型的掉点就非常严重。

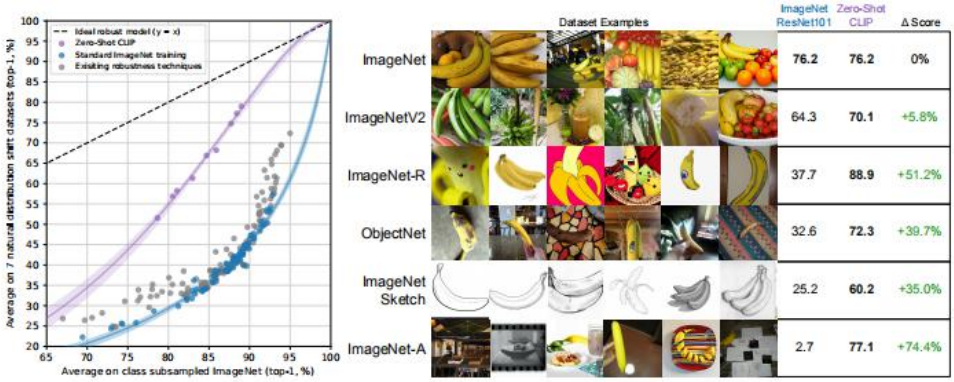


图 6 数据偏移情况下，CLIP 和其他模型性能对比

最后，由于几个测试中 CLIP 性能都很好，因此作者还将 CLIP 和人进行对比，

针对 Zero-Shot 的情况，人以不能上网搜索来限定；Few-Shots 的情况给人赛前看一张或两张图片。测试数据集采用 Oxford IIT，为宠物类别的数据集，测试结果如表 2 所示。但代表性并不够，只是做了个测试，体现 CLIP 的强大。对于人和 CLIP 而言，分类准确度低的类二者都低，准确度高的都高。因此对于分类而言，人和计算机的分类方式还是有内在联系的。

表 2 CLIP 和人类在各情况下的分类对比

	Accuracy	Accuracy on Guesses
Zero-Shot human	53.7	69.7
Zero-Shot CLIP	**93.5**	**93.5**
One-Shot human	75.7	78.5
Two-Shot human	75.7	79.2

§1.4 局限性

我认为，这一章节才是本文最重要的一部分，虽然 CLIP 已经足够强大了，但一定还是有很多做不了的事情，而这些局限性，才是对于我们、对于广大的研究而言能做出提升的点，给后续研究留下更多的发展空间。通过作者对于自身局限性的看待，能获得很多启发。

1. CLIP 性能强，但并没有达到不可一世的地步，本文对比的只是一个基线的模型，即 ResNet-50，才和它在 ImageNet 上打成平手。但相较于最新最大的 Vision Transformer[14]和 Noisy Student[15]，还是有十几个点的差距。继续加大模型和数据集，CLIP 模型的性能还会上升，但对比能达到 88%的模型，想要弥补十几个点的差距，要再扩大 1000 倍的计算资源，这显然需要新的方法。

2. 在有些数据集上（如细分类、抽象的数据集），Zero-Shot 效果也并不好，低于基线的 ResNet-50。因此，在很多数据集上，CLIP 的性能和瞎猜是一样的。

3. CLIP 的泛化做得很好，对自然图像的分布偏移，模型还是相对稳健的，但如果在推理时，数据偏移的非常远，泛化效果同样也很差。比如在 MNIST 数据集上，Zero-Shot 的 CLIP 仅有 88%，一般的模型都能轻易达到 99%。排查原因发现，预训练 CLIP 时，尽管有四亿个样本，由于 MNIST 是合成数据集，并没有一些自然的特征，所以没有一个样本和 MNIST 类似。因此，对于 CLIP 而言，这样的数据就在分布外，识别精度极低了。

4. 虽然 CLIP 可以做 Zero-Shot 的分类任务，但还是在用户期望的输出类别中选择，但更为直接输出是：直接生成文本输出。有个自然的解决想法：将生成式的目标函数和对比学习的目标函数合在一起，就有可能结合两种方法的优势。

5. 对数据的利用并不是很高效。测试一共用了 32 个 epochs，每个 epoch

有四亿个样本，数据的利用率太低。

6. 测试时不断在已有的数据上测试，调超参数，其实已经偏离了 Zero-Shot 的初衷。因此希望未来能创建一个专门的数据集用来做 Zero-Shot 迁移能力的测试，就会帮助解决很多问题。

7. CLIP 所使用的文本、图像对是从网上爬取的，并没有经过严格的清洗和审查。因此训练出的模型很可能带有社会上的偏见，如性别、肤色、宗教等。

8. 很复杂的概念有些即使连语言都没法描述的情况，就无法处理了。同时在测试中还看到在 Few-Shots 如 One-Shot 和 Two-Shots 时，性能还不如 Zero-Shot，这是很耐人寻味且不符合一般人类规律的。因此后期研究还将聚焦于如何使 CLIP 能在 Zero-Shot 和 Few-Shots 上都取得高效的表现。

第 2 章 论文实验结果验证

本章将选取第 1 章中作者训练的 8 个视觉模型中一部分进行精度验证；并对论文中测试结果较好的模型，网络上任意选择图片进行分类准确度的测试。

§2.1 不同模型同一图片、给定类，分类精度验证

代码如下所示，其中红色字体部分为加载模型的替换部分；划线部分为可修改的期望识别到的类。

```
import numpy as np
import torch
import clip
from PIL import Image
# 选择加载模型，可选项为：ViT-B/16, RN50, RN50x16 等
device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/16", device=device)
# 创建编码器和希望得到的图片类别
image = preprocess(Image.open("plane.jpeg")).unsqueeze(0).to(device)
text = clip.tokenize([ "Wonton","Shanghai Dumpling", "purple", "steamed bun"]).to(device)
with torch.no_grad():
    # 提取特征
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)
    # 计算相似度
    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()
# 输出每个类别的概率，概率高的相似度高
print("Label probs:", probs)
```

对于不同模型，同一图片、给定类，用图 7 (a)进行测试。这里的加载模型，选择 RN50, RN50x16, ViT-B/16，模型规模依次增大。测试图为小笼包（Shanghai Dumpling），给定类为馄饨、小笼包、馒头、面粉和上海。测试结果如表 所示，在小规模的模型 RN50 上，概率最高的是馒头（Steamed Bun)，显然分类出错；随着模型规模的增大，概率为小笼包的占比逐渐增大，在 ViT-B/16 上精度最高，为 92.68%，基本满足论文中提到的精度和模型规模正相关。

(a) 不同模型同一图片分类精度测试图　(b) 同一模型、图片，不同类别精度测试图

图 7 测试用图

表 3 不同模型，同一图片、给定类的测试精度对比

	Wonton	Shanghai Dumlping	Steamed Bun	Flour	Shanghai
RN50	0.1399	0.09393	**0.4958**	0.1388	0.1315
RN50x16	0.1011	**0.3544**	0.3458	0.0823	0.1301
ViT-B/16	0.0204	**0.9268**	0.05228	0.000037	0.000392

§2.2 同一模型、图片不同给定类，分类准确度验证

第二个验证实验为对同一图片、模型，给定有细微区别的不同的两个类，测试分类准确度。测试用图为图 7 (b) 紫馒头（Purple Bun），给定类 (a) 中包含“紫馒头”类，测试结果如表 4 (a) 所示，分类准确；给定类 (b) 中，不包含“紫馒头”，改为“馒头”和“紫色”，测试结果如表 4 (b) 所示，这次分类的最大值给到了“紫色”(Purple)，虽然并没有错，但是否将它分为“馒头”会更合理些呢？

对于“红包”图片也做了类似的测试，其中干扰项为“Red”和“Envelope”，最后的结果也将最大值给到了“红色”(Red)。因此引发思考：是否模型在无法确定准确类的情况下，优先根据颜色分类呢？这也值得更深入的研究。

表 4 同一模型、图片，不同给定类的测试精度对比

(a)

	Wonton	Shanghai Dumlping	Purple	**Purple Bun**
ViT-B/16	0.001049	0.0198	0.1632	**0.816**

(b)

	Wonton	Shanghai Dumlping	Purple	**Steamed Bun**
ViT-B/16	0.00463	0.08734	**0.72**	0.1879

第 3 章 总结与展望

§3.1 本文总结

本文通过将“NLP 领域预训练模型使之与下游任务无关”的思想，开创性地复制到了视觉领域，目的在于方法的迁移和使模型获得良好的泛化性。虽然也有很多的局限性，但总体而言，在多个数据集上的效果确实不错。预训练采用了对比学习的方式，在大规模数据集和大模型的加持下性能较好且有不错的泛化性，提升空间也很大。核心在于打破了固定种类标签的范式，在收集数据集和训练模型时，不用再预定义各种固定的类。

简而言之，在以下三个角度，这篇论文研究都很出色：

1. 新颖度：打破了视觉领域固定标签的做法，彻底放飞了视觉模型的训练过程；
2. 有效性：创建的数据集规模大，模型分类效果好、泛化性可观，在某种数据集下比人类分类的性能还好；
3. 解决问题大小：一个模型解决了大部分的分类任务，光是图像数据的分类，就是一个宏观上的大问题了。

因此，这篇论文的学习、研究价值是很高的，也为我创新项目中涉及的图像分类任务提供了一个新思路。也引发了我对如何进一步优化模型局限性的思考。

§3.2 展望

该论文为计算机领域的研究提供了一个新思路，不仅打通了 NLP 领域和视觉领域的方法迁移，更打通了计算机各技术领域间的壁垒，鼓励技术间的触类旁通。计算机不仅可以和其他学科开展交叉研究，用其他学科领域知识（如脑科学等）引导计算机领域的工作，计算机专业本身的各个技术领域也能交叉引导其他技术领域的发展。就我个人而言，想要实时了解最前沿的其他各个技术领域的知识并不容易，不仅涉及很多技术上的难点，还涉及获取上的困难。在本科阶段，我认为课程中安排这种精读论文、辅以自己根据论文进行代码应用和实验的方式非常好，虽然精读的量并不大，但提供了沉下心读论文的机会，也对技术细节有了大致的了解。今后课程结束，会将其作为习惯，定期精读前沿论文。

计算机的工作往往是模仿人类、帮助人类完成各项枯燥、危险、高难度的工作，因此对计算机技术的研究往往需要结合对人类活动特点的研究，比如对图像分类而言，如果人类得到一句文本的提示，很自然地会在图像分类时会加大准确率。这种想法通常是自然的，但如何将这种自然的想法应用到技术领域中，就是更高阶的思想和处理方式了。相信今后会有更多技术得到这种人类活动的启发。

致谢

在本次课程中，虽然由于疫情不能在教室里和同学老师相见，但课程的形式保障了教学质量，线上课堂中我不仅学习、了解到了大数据的理论知识和各种方法，更得到了沉下心阅读前沿论文的机会，真正实现了从理论到实践。作为计算机专业的同学，学习是常伴吾身的，时刻不能松懈，但惭愧的是，直到这次课程才真正研读第一篇论文。不过我相信这会是一个好的开始，非常感谢王健嘉老师和计算机学院开设此课程，虽有不能面对面相见的遗憾，却充满收获知识的喜悦。

本论文的完成不仅标志着本课程的结束，也标志着我得到启迪的新学习之路的开始，生活中离不开大数据，虽然课程中仅对相关概念、技术简要的进行了介绍，但师傅领进门，修行靠自身，相信会在未来的学习生活中不断提升对于“大数据”的见解。疫情正在不断好转，也相信在不远的未来，就能在校园里相见。

参考文献

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Leaning Transferable Visual Models From Natual Language Supervision. Proceedings of the 38th International Conference on Machine Learning, PMLR 139:8748-8763, 2021.

[2] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[3] Tom B. Brown, Benjamin Mann, et al. GPT-3: Language Models are Few-Shots Learners[J]. 2020.

[4] Deng, Jia et al. “ImageNet: A large-scale hierarchical image database.” 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009): 248-255.

[5] Li, A., Jabri, A., Joulin, A., and van der Maaten, L. Learning visual n-grams from web data. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4183–4192, 2017.

[6] Desai, K. and Johnson, J. Virtex: Learning visual representations from textual annotations. arXiv preprint arXiv:2006.06666, 2020

[7] Chen, T., Xu, B., Zhang, C., and Guestrin, C. Training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174, 2016.

[8] Gomez, L., Patel, Y., Rusinol, M., Karatzas, D., and Jawahar,C. Self-supervised learning of visual features through embedding images into text topic spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4230–4239, 2017.

[9] Joulin, A., Van Der Maaten, L., Jabri, A., and Vasilache, N. Learning visual features from large weakly supervised data. In European Conference on Computer Vision, pp. 67–84. Springer, 2016.

[10] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[11] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. Mixed precision training. arXiv preprint arXiv:1710.03740, 2017.

[12] Griewank, A. and Walther, A. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Transactions on Mathematical Software (TOMS), 26(1):19–45, 2000.

[13] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341, 2020.

[14] Xie, Qizhe, Luong, Minh-Thang et al. Self-training with Noisy Student improves ImageNet classification. 10.48550/ARXIV.1911.04252, 2020

[15] Dosovitskiy, Alexey and Beyer et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 10.48550/ARXIV.2010.11929, 2021

[16] Yann LeCun, Leon Bottou, et al. Gradient-Based Learning Applied to Document Recognition[J]. 1998

附录：部分源程序清单

```
# 导入所需库
import numpy as np
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"

# 加载模型：选择 RN50, RNx16 或 ViT-B/16
model, preprocess = clip.load("ViT-B/16", device=device)
# model, preprocess = clip.load("RN50", device=device)

# 第 2 章验证实验 1：不同模型同一图片、给定类精度测试
image = preprocess(Image.open("shanghai dumpling.jpeg")).unsqueeze(0).to(device)
text = clip.tokenize([ "Wonton","Shanghai Dumpling", "steam bun", "flour",
"Shanghai"]).to(device)

# 第 2 章验证实验 2：同一模型、图片不同给定类准确度测试
image = preprocess(Image.open("purple bun.jpg")).unsqueeze(0).to(device)
# text = clip.tokenize([ "Wonton","Shanghai Dumpling", "purple", "purple
bun"]).to(device)
text = clip.tokenize([ "Wonton","Shanghai Dumpling", "purple", "steamed
bun"]).to(device)

# 计算相似度
with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

# 输出概率
print("Label probs:", probs)
```