VirTex: Learning Visual Representation from Text annotation

Authors: Karan Desai, Justin Johnson
Date of Publication: 2021

Abstract: Authors carried out a set of experiments to find the best strategy to train Convolutional layers (feature extractor layers) of a CNN architecture that could be used for downstream tasks like classification, object segmentation or object detection, through transfer learning. Experiment showed that "learning visual representation from text annotation" out performed supervised RestNet baseline, and unsupervised baseline performance.
Their main contribution was to show that natural language can provide supervision for learning transferable visual representations with better data-efficiency than other approaches.

Link to the original paper:
https://openaccess.thecvf.com/content/CVPR2021/papers/Desai_VirTex_Learning_Visual_Representations_From_Textual_Annotations_CVPR_2021_paper.pdf

Source code: https://github.com/kdexd/virtex

Terminology

If you are new in the deep learning world then you have to learn a few terminologies that are used in the paper.

- ***ImageNet:***
  - It is a large-scale dataset that is used in computer vision research
  - ImageNet originally consisted of **over 14 million images** in more than **20,000 categories**.
  - The most well-known subset of ImageNet, known as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), contains around **1.2 million images** across **1,000 categories.**
- ***Pretrain:***
  - This is the term used to denote "training" of ML models in the context of transfer learning.
  - More formally, it refers to the process of training ML models with a large dataset before fine-tuning it on a specific task.
- ***Downstream transfer:***
  - It is a term that is used in the context of transfer learning, it basically means the use of a pre pre-trained model to perform specific tasks.
  - "downstream" refers to the subsequent task or tasks that the pretrained model is applied to.
  - The process of downstream transfer involves
    - **Pretraining**: learn generac features/patterns from large dataset

        - **Downstream task**: fine-tune the mode on task specific dataset related to the specific task.

- ***Self-supervised learning***
    - A Machine Learning paradigm that involves training models to learn from data itself without the need of extensive human label annotations.
    - Example, a word sequence prediction model trained to predict the next word based on the previous input sentence.

- ***Visual Representation***
    - Visual representation refers to the process of converting raw visual data into numerical forms that can be understood and analyzed by computers.
    - It is interchangeable with “feature map”.

- ***Semantic Density***
    - It refers to how much meaningful information is packed into a given piece of text.
    - For instance, Captions describe multiple objects, their attributes, relationships, and actions, giving a semantically dense learning signal.

- ***Linear projection***
    - A "linear projection" is a simple linear/dense layer without activation function, that is, **a matrix multiplication and a bias vector addition**
    - When doing linear projection, we can project a vector x of dimension n to a vector y of dimension size m by multiplying a projection matrix W of shape n×m.
    - Mathematically,
        - Output=Input×Weights+Bias
- ***BPE algorithm***
    - Byte-Pair Encoding: Subword-based tokenization algorithm
    - Link: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0

- **PASCAL VOC (PASCAL Visual Object Classes Challenge)**
    - The PASCAL VOC dataset contains images with annotations for various object classes. It covers a wide range of everyday objects, and each image is annotated with bounding boxes and labels indicating the presence of specific objects. The dataset was collected from various sources and covers different scenarios and settings.

Introduction

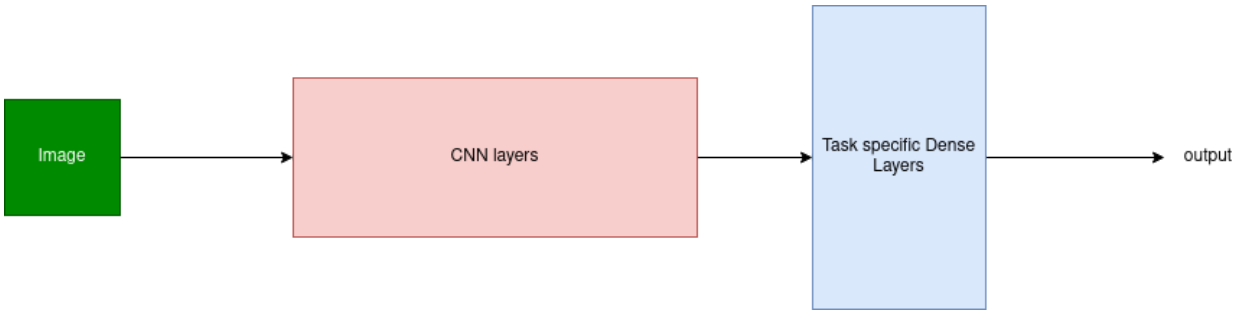

Fig: A generic block diagram of CNN

The prevailing paradigm for learning visual representations is first to pretrain a convolutional network to perform image classification on ImageNet, then transfer the learned features to downstream tasks.

Above diagram depicts the generic block diagram of a CNN, where

- **CNN block:**
  - Consists of convolution, batch normalization, pooling and activation layers
  - It is responsible for capturing various features like edges, textures, and patterns.
  - The quality of the model performance directly relies on this block.
  - In the context of transfer learning, we transfer knowledge that this pretrained CNN block has learned from a large generic dataset to perform a specific task.
- **Dense layers:**
  - It consists of neurons that use the extracted features to carry out specific tasks like classification, detections, segmentation and many more.

Various approach for pretraining

1. **Supervised learning for labeled dataset**
   - Use ImageNet to pretrain a machine learning model
   - Supervised learning is a machine learning paradigm where a model learns from labeled training data to make predictions or decisions.
   - **Dataset:** (image, label)
   - **Limitations:**
     - expensive to scale since the pretraining step relies on images annotated by human workers
     - Moderate semantic signal, since few words (one or two) are used to define image label

2. **Self supervised learning or unsupervised learning**

- It overcomes the scaling limitation of supervised learning, since it does not require annotated images.
- Unsupervised learning is a machine learning paradigm where a model learns patterns and relationships from unlabeled data without explicit guidance in the form of labeled outputs.
- **Dataset:** (image)
- **Limitations:**
    - Required large amount of unlabeled dataset
    - Semantically sparse signal, since no labels or textual information is provided.

3. **Purposed VirTex**
    - Learn high quality visual representation with fewer images
    - It uses supervised learning but instead of using image labels for pre-training, image caption or text annotation is used.
    - **Dataset:** (image, caption)
    - **Advantages:**
        - Captions provide a *semantically denser* learning signal than unsupervised contrastive methods and supervised classification.
        - Easy to collect, since image captioning can be done by non-experts in contrast with the image labeling.

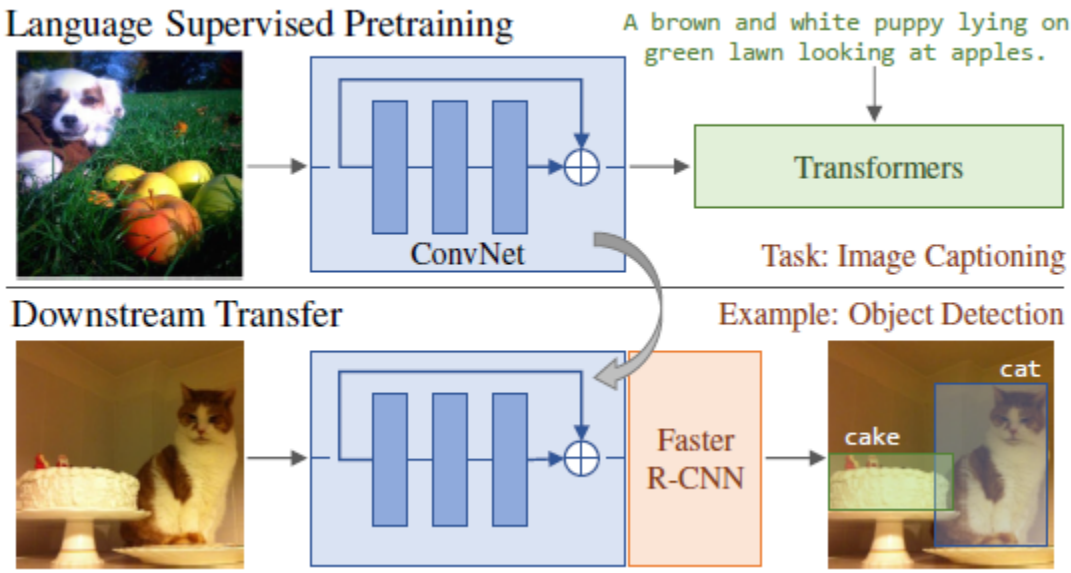

Fig: learning visual features from language

First, the author jointly trained a ConvNet and Transformers using image-caption pairs, for the task of image captioning (top). Then, they transfer the learned ConvNet to several downstream vision tasks, for example object detection (bottom).

Method

Model architecture

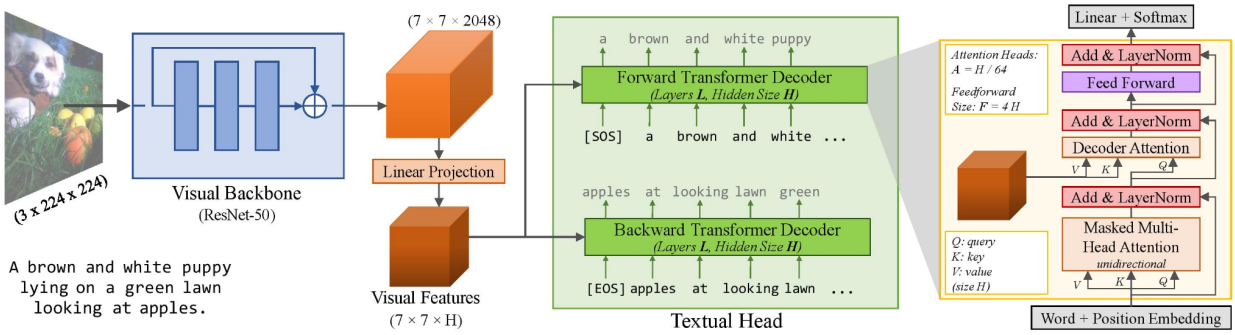

Fig: VirTex pre-training setup

They trained image captioning generator models as an upstream model to predict image caption. The model has two components

- **Visual Backbone**
  - This visual backbone is a convolutional network which computes image features
  - During pre training these features are used to predict captions
  - They used **ResNet-50** as a visual backbone
  - Input :
    - Image of size (3, 224,224)
  - Output:
    - Feature size (7, 7, 2048)

  - ***Projection layer***
    - During pretraining, we apply a linear projection layer to the visual features before passing them to the textual head to facilitate decoder attention over visual features.
    - This projection layer is not used in downstream tasks
    - Output: (7, 7, H)

- **Textual Head**
  - Receive features from visual backbone and predict captions $C = (c_0, c_1, \ldots, c_T, c_{T+1})$ token by token where C_0 = [SOS] and C_T = [EOS]
  - It comprise of two identical language model which predict captions in forward and backward direction respectively
  - For the language model they used **Transformer decoder architecture with GELU activation function.**
  - They used casual language modeling.
  - **Forward model:**
    - Inputs:
      - image feature: (7, 7, H)

        - image caption: (T+2) tokens
    - **Backward model:**
        - Similar to forward model, except it operates right-to-left, train to predict $C_{T:0}$, given c_{T+1}
    - The forward and backward model consist of independent transformer layers.
    - However, they share the same token embedding matrix which is also used at output layer of each model

Loss function

All model components are randomly initialized, and jointly trained to maximize the log-likelihood of the correct caption tokens.

$$\mathcal{L}(\theta, \phi) = \sum_{t=1}^{T+1} \log \Big(p(c_t \mid c_{0:t-1}, I; \phi_f, \theta) \Big) + \sum_{t=0}^{T} \log \Big(p(c_t \mid c_{t+1:T+1}, I; \phi_b, \theta) \Big) \quad (1)$$

where θ, ϕ_f, and ϕ_b are the parameters of the visual backbone, forward, and backward models respectively. After

Model hyper parameters:

Several architectural hyperparameters control the size of the textual head.

- **Hidden size of transformer:** *H*
- **Number of attention head:** *A*
    - *A = H/64*
- **feed forward size:** *F*
    - *F = 4H*
- **Number of transformer layer:** *L*

Tokenization:

- Used BPE algorithm; compared to basic tokenization that split on whitespace, BPE

makes fewer linguistic assumptions, exploits subword information, and results in fewer out-of-vocab tokens.

- **Vocab size:** 10K including [SOS], [EOS], and [UNK] tokens
- **Preprocessing:** lowercase and strip accent

Training Details

- **Dataset:** *train2016* split of COCO captions dataset, which provides 118K images with 5 captions each.
- **Data augmentation:**
    - ***Random crop:*** 20-100% of the original size
    - ***Color jitter:*** brightness, contrast, saturation and hue
    - ***Normalization:*** using ImageNet mean color
    - ***Horizontal flip:*** *random, (also interchanged the word "left" and "right" in the caption)*
- **Optimizer:**
    - SGD
        - momentum: 0.9
        - Weight decay: 10^{-4} wrapped in LookAhead
        - Lr: 0.5
        - Steps: 5
- **Distributed training**
    - **8 GPUs** with batch normalization per GPU
- **Batch size:**
    - **256** (32 per GPU)
- **No of epochs:** 1080

Experiments Results:

Method	Annotations	Cost (hours)†	VOC07	IN-1k
MoCo-COCO	self-sup.	–	63.3	41.1
Multi-label Clf.	labels	11.1K [30]	86.2	46.2
Instance Segmentation	masks	30.0K [30]	82.3	51.0
VirTex (1 caption)	captions	1.3K [100]	84.2	50.2
VirTex (5 caption)	captions	6.5K [100]	**88.7**	**53.8**

Table 1: comparison of downstream performance of various pretraining methods on COCO. VirTex outperforms all other methods trained on the same set of images with best performance vs. cost tradeoff.

Where,

- **IN-1k** is the subset of original ImageNet dataset
- **MoCo-COCO:** Momentum Contrast for Unsupervised Visual Representation Learning. MoCo, or Momentum Contrast, is a self-supervised learning algorithm with a contrastive loss.

Note that, Upstream modes are trained on COCO whereas downstream model are tested on VOC07 and IN-1k

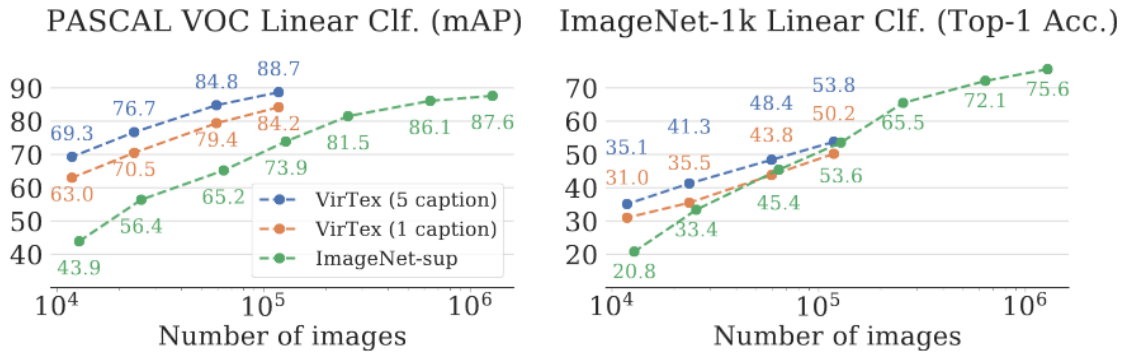

Figure: Data Efficiency: comparison of VirTex and IN-sup (ImageNet-Supervised) models trained using varying amounts of images. VirTex closely matches or significantly outperforms IN-sup on downstream tasks despite using 10× fewer images.

Please checkout origin paper to get more insight into various experiments.

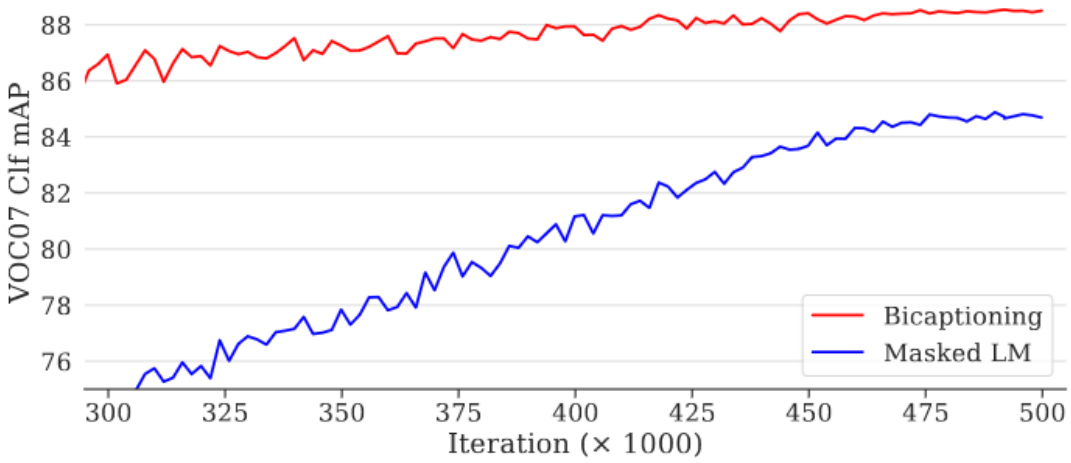

Figure 7: **Bicaptioning vs. Masked Language Modeling:** We compare VOC07 mAP of Bicaptioning and Masked LM pretraining tasks. We observe that Masked LM converges slower than Bicaptioning, indicating poor sample efficiency.

Backbone	VOC07	IN-1k	PASCAL VOC Detection		
	mAP	Top-1	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}
ResNet-50	88.3	53.2	55.2	81.2	60.8
ResNet-50 w2×	$88.5_{+0.2}$	$52.9_{-0.3}$	$56.6_{+1.4}$	$82.0_{+0.8}$	$62.8_{+2.0}$
ResNet-101	$88.7_{+0.4}$	$52.0_{-1.2}$	$57.9_{+2.7}$	$82.0_{+0.8}$	$63.6_{+2.8}$

Table 4: **Additional Evaluations for Backbone Ablations.** We compare VirTex models ($L = 1, H = 1024$) with different visual backbones. We observe that larger backbones generally improve downstream performance.

Method	Pretrain Images	LVIS v0.5 Instance Segmentation		
		AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}
1) Random Init		22.5	34.8	23.8
2) IN-sup	1.28M	24.5	38.0	26.1
3) IN-sup-50%	640K	23.7 –0.8	36.7 –1.3	25.1 –1.0
4) IN-sup-10%	128K	20.5 –4.0	32.8 –6.2	21.7 –5.2
5) MoCo-IN	1.28M	24.1 –0.4	37.4 –0.6	25.5 –0.6
6) MoCo-COCO	118K	23.1 –1.4	35.3 –2.7	24.9 –1.2
7) VirTex	118K	25.4 +0.9	39.0 +1.0	26.9 +0.8

Table 8: **Downstream Evaluation: LVIS v0.5 Instance Segmentation.** We compare VirTex with different pretraining methods for LVIS `v0.5` Instance Segmentation. All methods use Mask R-CNN with ResNet-50-FPN backbone. Performance gaps with IN-sup are shown on the side. The trends are similar to LVIS `v1.0` Table 3 – VirTex significantly outperforms all baseline methods.

Conclusion

It shows that learning visual representation using textual annotations can be competitive to methods based on supervised classification and self-supervised learning on ImageNet.