Multimodal Deep Learning

Contents

Preface

FIGURE 1: Creative Commons License

Foreword

Author: Christoph Molnar

This book is the result of an experiment in university teaching. Each semester, students of the Statistics Master can choose from a selection of seminar topics. Usually, every student in the seminar chooses a scientific paper, gives a talk about the paper and summarizes it in the form of a seminar paper. The supervisors help the students, they listen to the talks, read the seminar papers, grade the work and then … hide the seminar papers away in (digital) drawers. This seemed wasteful to us, given the huge amount of effort the students usually invest in seminars. An idea was born: Why not create a book with a website as the outcome of the seminar? Something that will last at least a few years after the end of the semester. In the summer term 2019, some Statistics Master students signed up for our seminar entitled "Limitations of Interpretable Machine Learning". When they came to the kick-off meeting, they had no idea that they would write a book by the end of the semester.

We were bound by the examination rules for conducting the seminar, but otherwise we could deviate from the traditional format. We deviated in several ways:

1. Each student project is part of a book, and not an isolated seminar paper.
2. We gave challenges to the students, instead of papers. The challenge was to investigate a specific limitation of interpretable machine learning methods.
3. We designed the work to live beyond the seminar.
4. We emphasized collaboration. Students wrote some chapters in teams and reviewed each others texts.

Technical Setup

The book chapters are written in the Markdown language. The simulations, data examples and visualizations were created with R (**?**). To combine R-code and Markdown, we used rmarkdown. The book was compiled with the

bookdown package. We collaborated using git and github. For details, head over to the book's repository.

1

Introduction

Author: Nadja Sauter

Supervisor: Matthias Assenmacher

FIGURE 1.1: A cute monster taking a shower in a bathtub trending on art station (CLIP + Guided Diffuision) from multimodal.art

- Intro About the Seminar Topic: show “AI Art” -> multimodal deep learning Text2Img (methods: CLIP + Guided Diffuision (see pictrue above) or DALL-E; Glide)

- Different types of multimodal deep learning:
  - Text2Img (mentioned before)
  - Img2Text (methods: Microsoft Coco; Meshed memory Transformer for Image captioning M2)
  - Image supporting Language models
  - ...
- Detailed methods explained in book

- Outline of the Booklet:
  - Fundamentals: NLP + CV
  - Specific mutlimoald models mentioned above

2

Introducing the modalities

Authors: Cem Akkus, Vladana Djakovic, Christopher Benjamin Marquardt

Supervisor: Dr. Matthias Aßenmacher

Natural Language Processing (NLP) has existed for about 50 years, but it is more relevant than ever. There have been several breakthroughs in this branch of machine learning that is concerned with spoken and written language. For example, learning internal representations of words was one of the greater advances of the last decade. Word embeddings (**?**, **?**) made it possible and allowed developers to encode words as dense vectors that capture their underlying semantic content. In this way, similar words are embedded close to each other in a lower-dimensional feature space. Another important challenge was solved by Encoder-decoder (also called sequence-to-sequence) architectures **?**, which made it possible to map input sequences to output sequences of different lengths. They are especially useful for complex tasks like machine translation, video captioning or question answering. This approach makes minimal assumptions on the sequence structure and can deal with different word orders and active, as well as passive voice.

A definitely significant state-of-the-art technique is Attention **?**, which enables models to actively shift their focus – just like humans do. It allows following one thought at a time while suppressing information irrelevant to the task. As a consequence, it has been shown to significantly improve performance for tasks like machine translation. By giving the decoder access to directly look at the source, the bottleneck is avoided and at the same time, it provides a shortcut to faraway states and thus helps with the vanishing gradient problem. One of the most recent sequence data modeling techniques is Transformers (**?**), which are solely based on attention and do not have to process the input data sequentially (like RNNs). Therefore, the deep learning model is better in remembering context-induced earlier in long sequences. It is the dominant paradigm in NLP currently and even makes better use of GPUs, because it can perform parallel operations. Transformer architectures like BERT (**?**), T5 (**?**) or GPT-3 (**?**) are pre-trained on a large corpus and can be fine-tuned for specific language tasks. They have the capability to generate stories, poems, code and much more. With the help of the aforementioned breakthroughs, deep networks have been successful in retrieving information and finding representations of semantics in the modality text. In

the next paragraphs, developments for another modality image are going to be presented.

Computer vision (CV) focuses on replicating parts of the complexity of the human visual system and enabling computers to identify and process objects in images and videos in the same way that humans do. In recent years it has become one of the main and widely applied fields of computer science. However, there are still problems that are current research topics, whose solutions depend on the research's view on the topic. One of the problems is how to optimize deep convolutional neural networks for image classification. The accuracy of classification depends on width, depth and image resolution. One way to address the degradation of training accuracy is by introducing a deep residual learning framework (**?**). On the other hand, another less common method is to scale up ConvNets, to achieve better accuracy is by scaling up image resolution. Based on this observation, there was proposed a simple yet effective compound scaling method, called EfficientNets (**?**).

Another state-of-the-art trend in computer vision is learning effective visual representations without human supervision. Discriminative approaches based on contrastive learning in the latent space have recently shown great promise, achieving state-of-the-art results, but the simple framework for contrastive learning of visual representations, which is called SimCLR, outperforms previous work (**?**). However, another research proposes as an alternative a simple "swapped" prediction problem where we predict the code of a view from the representation of another view. Where features are learned by Swapping Assignments between multiple Views of the same image (SwAV) (**?**). Further recent contrastive methods are trained by reducing the distance between representations of different augmented views of the same image ('positive pairs') and increasing the distance between representations of augmented views from different images ('negative pairs'). Bootstrap Your Own Latent (BYOL) is a new algorithm for self-supervised learning of image representatios (**?**).

Self-attention-based architectures, in particular, Transformers have become the model of choice in natural language processing (NLP). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention, some replacing the convolutions entirely. The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Inspired by the Transformer scaling successes in NLP, one of the experiments is applying a standard Transformer directly to the image (**?**). Due to the widespread application of computer vision, these problems differ and are constantly being at the center of attention of more and more research.

With the rapid development in NLP and CV in recent years, it was just a question of time to merge both modalities to tackle multi-modal tasks. The release of DALL-E 2 just hints at what one can expect from this merge in the future. DALL-E 2 is able to create photorealistic images or even art from

any given text input. So it takes the information of one modality and turns it into another modality. It needs multi-modal datasets to make this possible, which are still relatively rare. This shows the importance of available data and the ability to use it even more. Nevertheless, all modalities are in need of huge datasets to pre-train their models. It's common to pre-train a model and fine-tune it afterwards for a specific task on another dataset. For example, every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset. The cardinality of the datasets used for CV is immense, but the datasets used for NLP are of a completely different magnitude. BERT uses the English Wikipedia and the Bookscorpus to pre-train the model. The latter consists of almost 1 billion words and 74 million sentences. The pre-training of GPT-3 is composed of five huge corpora: CommonCrawl, Books1 and Books2, Wikipedia and WebText2. Unlike language model pre-training that can leverage tremendous natural language data, vision-language tasks require high-quality image descriptions that are hard to obtain for free. Widely used pre-training datasets for VL-PTM are Microsoft Common Objects in Context (COCO), Visual Genome (VG), Conceptual Captions (CC), Flickr30k, LAION-400M and LAION-5B, which is now the biggest openly accessible image-text dataset.

Besides the importance of pre-training data, there must also be a way to test or compare the different models. A reasonable approach is to compare the performance on specific tasks, which is called benchmarking. A nice feature of benchmarks is that they allow us to compare the models to a human baseline. Different metrics are used to compare the performance of the models. Accuracy is widely used, but there are also some others. For CV the most common benchmark datasets are ImageNet, ImageNetReaL, CIFAR-10(0), OXFORD-IIIT PET, OXFORD Flower 102, COCO and Visual Task Adaptation Benchmark (VTAB). The most common benchmarks for NLP are General Language Understanding Evaluation (GLUE), SuperGLUE, SQuAD 1.1, SQuAD 2.0, SWAG, RACE, ReCoRD, and CoNLL-2003. VTAB, GLUE and SuperGLUE also provide a public leader board. Cross-modal tasks such as Visual Question Answering (VQA), Visual Commonsense Reasoning (VCR), Natural Language Visual Reasoning (NLVR), Flickr30K, COCO and Visual Entailment are common benchmarks for VL-PTM.

2.1 State-of-the-art in computer vision

Author: Vladana Djakovic

Supervisor: Daniel Schalk

2.1.1 History

The first research about visual perception comes from neurophysiological research performed in the 1950s and 1960s on cats. Scientists concluded that human vision is hierarchical, and Neurons detect simple features like edges, followed by more complex features like shapes and more complex visual representations. Inspired by this knowledge, computer scientists focused on recreating human neurological structures. At around the same time, as computers became more advanced, computer scientists worked on imitating human neurons' behavior and simulating a hypothetical neural network. Donald Hebbin, in his book, The Organization of Behaviour (1949), stated that neural pathways strengthen over each successive use, especially between neurons that tend to fire at the same time, thus beginning the long journey towards quantifying the complex processes of the brain. The first Hebbian network was successfully implemented at MIT in 1954. (https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec) New findings led to the establishment of the field of artificial intelligence in 1956 on-campus at Dartmouth College. Scientists began to develop ideas and research how to create techniques that would imitate the human eye. Early research on developing neural networks was performed at Stanford University in 1959, where models called "ADALINE" and "MADALINE", Multiple ADAptive LINear Elements, were developed. Those models aimed to recognize binary patterns and could predict the next bit. (https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html) Starting optimism about Computer Vision and neural networks disappeared after 1969 and the publication of the book "Perceptrons" by Marvin Minsky, founder of the MIT AI Lab. In the book, the authors stated that the single perception approach to neural networks could not be translated effectively into multi-layered neural networks. The period that followed was known as AI Winter, which lasted until 2010, when the internet became widely used and the technological development of computers. In 2012 breakthroughs in Computer Vision happened at the ImageNet Large Scale Visual Recognition Challenge (ILSVEC). The team from the University of Toronto issued a deep neural network called AlexNet that changed the field of artificial intelligent CV. AlexNet achieved an error rate of 16.4%. From 2012 until today, Computer Vision has been one of the fastest fields. Researchers are competing to conduct a model that would be the most similar to the human eye and help humans in everyday life. Here the author will describe only a few recent state-of-the-art models.

2.1.2 Supervised and unsupervised learning

As part of artificial intelligence (AI) and machine learning (ML), there are two basic approaches: * supervised learning; * unsupervised learning.

Supervised learning is defined by using labeled datasets to train algorithms that classify data or predict outcomes accurately. With labeled inputs and

outputs model can measure its accuracy and learn over time. We can distinguish two types of data mining problems: * classification * regression. (https://www.ibm.com/cloud/learn/supervised-learning)

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms aim to discover hidden patterns or data groupings without previous human intervention. Its ability to discover similarities and differences in information is mostly used for three main tasks: * clustering, * association * dimensionality reduction. (https://www.ibm.com/cloud/learn/unsupervised-learning)

Solving the problems where the dataset can be both labeled and unlabeled requires an approach between supervised and unsupervised learning, called *semi-supervised learning*. It is useful when extracting relevant features from data that is complex and when data is high volume, i.e., medical images.

Nowadays, there is a new research topic in the machine learning community, and it is *Self-Supervised Learning*. Self-Supervised learning is a machine learning process where the model trains itself to learn one part of the input from another part of the input. (https://neptune.ai/blog/self-supervised-learning) It is a subset of unsupervised learning where outputs or goals are derived by machines that label, categorize, and analyze information on their own and then draw conclusions based on connections and correlations. Self-supervised learning can also be an autonomous form of supervised learning because it does not require human input in data labeling. In contrast to unsupervised learning, self-supervised learning does not focus on clustering and grouping, which is commonly associated with unsupervised learning. (https://www.techslang.com/definition/what-is-self-supervised-learning/) One part of Self-Supervised learning is *contrastive learning*. This technique is used to learn the general features of a dataset without labels by teaching the model which data points are similar or different. It is used to train the model to learn about our data without any annotations or labels. (https://towardsdatascience.com/understanding-contrastive-learning-d5b19fd96607)

2.1.3 ResNet

In 2015 He K., Zhang X., et al. presented deep residual networks to ILSVRC and COCO competitions. They won first place on tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. Until then, deep convolutional neural networks have led to a series of breakthroughs for image classification. Research showed that network depth is crucial, and the top results on the challenging ImageNet dataset all exploit “very deep” models. The authors of this paper questioned will stack more layers leads to learning a better network. One obstacle was the problem of vanishing/exploding gradients, and it has been primarily addressed by normalized initialization and

intermediate normalization layers. That enabled networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation.

Another obstacle was a degradation problem. The problem occurs when the network depth increases, accuracy gets saturated, and then degrades rapidly. Such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, which indicates that not all systems are similarly easy to optimize.

For example, consider a shallower architecture and its deeper counterpart that adds more layers. One solution is to create a deeper model, where the added layers are identity mappings, and other layers are copied from a shallower model. The deeper model should produce no higher training error than its shallower counterpart. However, in practice, it is not, and it is hard to find comparably good or better solutions than the constructed solution. The authors proposed that a solution to this degradation problem is a deep residual learning framework.

###maybe repeated The idea was that they explicitly let every few stacked layers fit a residual mapping instead of hoping they would directly fit a desired underlying mapping. Formally, denoting the desired underlying mapping as $H(x)$, they let the stacked nonlinear layers fit another mapping of $F(x) := H(x) - x$. The hypothesis was that it is easier to optimize the residual mapping than the original unreferenced mapping.

2.1.3.1 Deep Residual Learning

2.1.3.1.1 Residual Learning

The idea of residual learning is to replace the approximation of underlying mapping $H(x)$, which is approximated by a few stacked layers (not necessarily the entire net), with an approximation of residual function $F(x) := H(x) - x$. Here x denotes the inputs to the first of these layers, and the authors assume that both inputs and outputs have the same dimensions. The original function becomes $F(x) + x$.

A counterintuitive phenomenon about degradation motivated this reformulation. A new deeper model should have no more significant training error when added layers are constructed as identity mappings. Solvers may have challenges approximating identity mappings by multiple nonlinear layers because of the degradation problem. Using the residual learning reformulation, the solvers can drive the weights of the nonlinear layers toward zero to approach identity mappings if identity mappings are optimal. Generally, identity mappings are not optimal, but new reformulations may help precondition the problem. When an optimal function is closer to an identity mapping than a zero mapping, finding perturbations concerning an identity mapping should be easier than learning the function from scratch.

2.1.3.1.2 Identity Mapping by Shortcuts

Residual learning is adopted to every few stacked layer where a building block is defined as and shown in Fig. N:

$$y = F(x, \{W_i\}) + x \tag{1}$$

x and y represent the input and output vectors of the layers. The function $F(x, \{W_i\})$ represents the residual mapping to be learned. For the example in Fig. N that has two layers, $F = W_2\sigma(W_1x)$ in which σ denotes ReLU activation function and to simplify the notations, biases are left out. With a shortcut connection and element-wise addition, the operation $F + x$ is conducted. Afterwards authors have applied second nonlinearity (i.e., $\sigma(y)$, Fig. N).

The shortcut connections in Eqn. (1) neither adds an extra parameter nor increases computation complexity, which enables comparisons between plain and residual networks that simultaneously have the same number of parameters, depth, width, and computational cost (except for the negligible element-wise addition). The dimensions of x and F must be equal in Eqn. (1). Alternatively, linear projection W_s by the shortcut connections to match the dimensions can be applied:

$$y = F(x, \{W_i\}) + W_sx. \tag{2}$$

The square matrix W_s can be used in Eqn (1). However, experiments showed that identity mapping is enough to solve the degradation problem. Therefore, W_s only aims to match dimensions. The authors did not state the exact form of the residual function F, so they experimented with function F, which has two or three layers, although more layers are possible. The square matrix W_s can be used in Eqn (1). However, experiments showed that identity mapping is enough to solve the degradation problem. Therefore, W_s only aims to match dimensions. The authors did not state the exact form of the residual function F, so they experimented with function F, which has two or three layers, although more layers are possible. Assuming F only has one layer, Eqn. (1) it is comparable to a linear layer: $y = W_1x + x$ and authors did not observed this case. The theoretical notations are about fully-connected layers, but the authors have used convolutional layers. The function $F(x, \{W_i\})$ can be used to represent multiple convolutional layers. Two feature maps are added element-wise, channel by channel.

2.1.3.1.3 Network Architectures-Not Done

The authors of the paper have tested various plain/residual nets and have observed their EFFECTIVENESS? . They described following two models for ImageNet: *Plain Network* Plain baselines are mainly inspired by the philoso-

phy of VGG nets 41 (See which one is that for citations) (Fig. 3, left). The convolutional layers mostly have 3×3 filters and follow two simple design rules:

for the same output feature map size, the layers have the same number of filters; if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer.

They perform downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. The total number of weighted layers is 34 in Fig. 3 (middle). It is worth noticing that our model has fewer filters and lower complexity than VGG nets [41] (Fig. 3, left). Our 34- layer baseline has 3.6 billion FLOPs (multiply-adds), which is only 18% of VGG-19 (19.6 billion FLOPs).

Residual Network. Based on the above plain network, we insert shortcut connections (Fig. 3, right) which turn the network into its counterpart residual version. The identity shortcuts (Eqn. (1)) can be directly used when the input and output are of the same dimensions (solid line shortcuts in Fig. 3). When the dimensions increase (dotted line shortcuts in Fig. 3), we consider two options:

The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; The projection shortcut in Eqn. (2) is used to match dimensions (done by 1×1 convolutions).

For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

2.1.3.1.4 Implementation

Our implementation for ImageNet follows the practice in [21, 41]. The image is resized with its shorter side randomly sampled in [256, 480] for scale augmentation [41]. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [21]. The standard color augmentation in [21] is used. We adopt batch normalization (BN) [16] right after each convolution and before activation, following [16]. We initialize the weights as in [13] and train all plain/residual nets from scratch. We use SGD with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to 60×104 iterations. We use a weight decay of 0.0001 and a momentum of 0.9. We do not use dropout [14], following the practice in [16]. In testing, for comparison studies we adopt the standard 10-crop testing [21]. For best results, we adopt the fully-convolutional form as in [41, 13], and average the scores at multiple scales (images are resized such that the shorter side is in {224, 256, 384, 480, 640}).

2.1.3.1.4.1 Experiments at the end

2.1.4 EfficientNet

Since the first implementation of ConvNets, scaling them to achieve better accuracy has become a new challenge. As it was described, ResNet can be scaled by using more layers. Unfortunately, scaling up ConvNets is not unique and has never been well understood. Usually, ConvNets are by their depth (ResNets) or width (Zagoruyko & Komodakis, 2016). Another less common method is to scale up models by image resolution (Huang et al., 2018). Until this paper, it was common to scale only one of the three dimensions – depth, width, or image size. In this paper, the authors want to develop a new way to scale up ConvNets. Their empirical study shows that it is critical to balance all network width/depth/resolution dimensions, which can be achieved by simply scaling each with a constant ratio. Based on this observation, they proposed a simple yet effective compound scaling method, which uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients. For example, suppose we want to use 2N times more computational resources. In that case, we can increase the network depth by αN, width by βN, and image size by γN, where α, β, γ are constant coefficients determined by a small grid search on the original miniature model. Figure M illustrates the difference between this scaling method and conventional methods. A compound scaling method makes sense if an input image is bigger since a larger receptive field requires more layers and more significant channel features to capture fine-grained patterns. Theoretically and empirically, there has been a special relationship between network width and depth (Raghu et al., 2017; Lu et al., 2018), but the authors claim they are the first to quantify this relationship among all three dimensions empirically. The authors' introduction paper demonstrated their scaling method on existing MobileNets (Howard et al., 2017; Sandler et al., 2018) and ResNet.

2.1.4.1 Compound Model Scaling

2.1.4.1.1 Problem Formulation-everything from paper

A ConvNet Layer i can be defined as a function: $Y_i = \mathcal{F}_i(X_i)$, where $\mathcal{F}_i$ is the operator, Y_i is output tensor, X_i is input tensor, with tensor shape (H_i, W_i, C_i), where H_i and W_i are spatial dimension and C_i is the channel dimension. A ConvNet N can be represented by a list of composed layers: n

$$\mathcal{N} = \mathcal{F}_k \odot \cdots \mathcal{F}_2 \odot \mathcal{F}_1(X_1) = \bigodot_{j=1\cdots k} \mathcal{F}_j(X_1)$$

. In practice, ConvNet layers are often partitioned into multiple stages and all layers in each stage share the same architecture: for example, ResNet has five stages, and all layers in each stage has the same convolutional type except the first layer performs down-sampling. Therefore, we can define a ConvNet as:

$$\mathcal{N} = \bigodot_{i=1\cdots s} \mathcal{F}_i^{L_i}\left(X_{(H_i, W_i, C_i)}\right)$$

where $\mathcal{F}_i^{L_i}$ denotes layer $\mathcal{F}_i$ is repeated L_i times in stage i, (H_i, W_i, C_i) enotes the shape of input tensor X of layer i

??Figure 2(a) illustrate a representative ConvNet, where the spatial dimension is gradually shrunk but the channel dimension is expanded over layers, for example, from initial input shape 224, 224, 3 to final output shape 7, 7, 512 .

Unlike regular ConvNet designs that mostly focus on finding the best layer architecture $\mathcal{F}_i$, model scaling tries to expand the network length (L_i), width (C_i), and/or resolution (H_i, W_i) without changing $\mathcal{F}_i$ predefined in the baseline network. By fixing $\mathcal{F}_i$, model scaling simplifies the design problem for new resource constraints, but it still remains a large design space to explore different (L_i, H_i, W_i, C_i) for each layer. In order to further reduce the design space, we restrict that all layers must be scaled uniformly with a constant ratio. Our target is to maximize the model accuracy for any given resource constraints, which can be formulated as an optimization problem:

$$\max_{d,w,r} Accuracy\left(\mathcal{N}\left(d,w,r\right)\right)$$

$$s.t. \mathcal{N}\left(d,w,r\right) = \bigodot_{I=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\left(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\right)$$
$$Memory\left(\mathcal{N}\right) \leq targetMemory$$
$$FLOPS\left(\mathcal{N}\right) \leq targetFlops$$

where *w,d,r* are coefficients for scaling network width, depth,and resolution; $\left(\widehat{\mathcal{F}}_i, \widehat{L}_i, \widehat{H}_i, \widehat{W}_i, \widehat{C}_i\right)$ are predefined parameters in baseline network.

2.1.4.1.2 Scaling Dimensions

The main difficulty of problem 2 is that the optimal *d, w, r* depend on each other and the values change under different resource constraints. Due to this difficulty, conventional methods mostly scale ConvNets in one of these dimensions:

paraphrase

Depth(d): Scaling network depth is the most common way used by many ConvNets (He et al., 2016; Huang et al., 2017; Szegedy et al., 2015; 2016). The intuition is that deeper ConvNet can capture richer and more complex features, and generalize well on new tasks. However, deeper networks are also more difficult to train due to the vanishing gradient problem (Zagoruyko & Komodakis, 2016). Although several techniques, such as skip connections (He et al., 2016) and batch normalization (Ioffe & Szegedy, 2015), alleviate the training problem, the accuracy gain of very deep network diminishes: for example, ResNet-1000 has similar accuracy as ResNet-101 even though it has much more layers. Figure 3 (middle) shows our empirical study on scaling a baseline model with different depth coefficient d, further suggesting the diminishing accuracy return for very deep ConvNets.

Width (w): Scaling network width is commonly used for small size models (Howard et al., 2017; Sandler et al., 2018; Tan et al., 2019)2. As discussed in (Zagoruyko & Komodakis, 2016), wider networks tend to be able to capture more fine-grained features and are easier to train. However, extremely wide but shallow networks tend to have difficulties in capturing higher level features. Our empirical results in Figure 3 (left) show that the accuracy quickly saturates when networks become much wider with larger w.

Resolution (r): With higher resolution input images, ConvNets can potentially capture more fine-grained patterns. Starting from 224x224 in early ConvNets, modern ConvNets tend to use 299x299 (Szegedy et al., 2016) or 331x331 (Zoph et al., 2018) for better accuracy. Recently, GPipe (Huang et al., 2018) achieves state-of-the-art ImageNet accuracy with 480x480 resolution. Higher resolutions, such as 600x600, are also widely used in object detection ConvNets (He et al., 2017; Lin et al., 2017). Figure 3 (right) shows the results of scaling network resolutions, where indeed higher resolutions improve accuracy, but the accuracy gain diminishes for very high resolutions (r = 1.0 denotes resolution 224x224 and r = 2.5 denotes resolution 560x560).

The above analyses lead to the first observation: **Observation 1** – Scaling up any dimension of network width, depth, or resolution improves accuracy, but the accuracy gain diminishes for bigger models.

2.1.4.1.3 Compound Scaling

Firstly, authors have observed that different scaling dimensions are not independent, because higher resolution images require increased network depth so that the larger receptive fields can help capture similar features that include more pixels in bigger images. Similarly, network width should be increased when resolution is higher, to capture more fine-grained patterns with more pixels in high-resolution images. The intuition suggests that different scaling dimensions should be coordinated and balanced rather than conventional scaling in single dimensions.

To confirm this though authors compared results of networks width w without changing depth (d=1.0) and resolution (r=1.0) with deeper (d=2.0) and higher resolution (r=2.0). This showed that width scaling achieves much better accuracy under the same FLOPS cost. These results lead to the second observation:

Observation 2 In order to pursue better accuracy and efficiency, it is critical to balance all dimensions of network width, depth, and resolution during ConvNet scaling. In fact, a few prior work (Zoph et al., 2018; Real et al., 2019) have already tried to arbitrarily balance network width and depth, but they all require tedious manual tuning.

Authors have proposed a new **compound scaling method**, which uses a compound coefficient φ to uniformly scales network width, depth, and resolu-

tion in a principled way

$$\begin{gathered} depth: d=\alpha^{\varphi} \\ width: w=\beta^{\varphi} \\ resolution: r=\gamma^{\varphi} \\ s.t. \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1, \end{gathered}$$

where α, β, γ are constants that can be determined by a small grid search. Intuitively, φ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively. Notably, the FLOPS of a regular convolution op is proportional to d, w^2, r^2 i.e., doubling network depth will double FLOPS, but doubling network width or resolution will increase FLOPS by four times. Since convolution ops usually dominate the computation cost in ConvNets, scaling a ConvNet with equation 3 will approximately increase total FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\varphi}$ In this paper, we constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ such that for any new φ, the total FLOPS will approximately3 increase by 2φ

2.1.4.2 EfficientNet Architecture

Since model scaling does not change layer operators F^i in baseline network, having a good baseline network is also critical. We will evaluate our scaling method using existing ConvNets, but in order to better demonstrate the effectiveness of our scaling method, we have also developed a new mobile-size baseline, called EfficientNet. Inspired by (Tan et al., 2019), we develop our baseline network by leveraging a multi-objective neural architecture search that optimizes both accuracy and FLOPS. Specifically, we use the same search space as (Tan et al., 2019), and use ACC(m)×[FLOPS(m)/T]w as the optimization goal, where ACC(m) and FLOPS(m) denote the accuracy and FLOPS of model m, T is the target FLOPS and w=-0.07 is a hyperparameter for controlling the trade-off between accuracy and FLOPS. Unlike (Tan et al., 2019; Cai et al., 2019), here we optimize FLOPS rather than latency since we are not targeting any specific hardware de- vice. Our search produces an efficient network, which we name EfficientNet-B0. Since we use the same search space as (Tan et al., 2019), the architecture is similar to Mnas-Net,the larger FLOPS target (our FLOPS target is 400M). Table 1 shows the architecture of EfficientNet-B0. Its main building block is mobile inverted bottleneck MBConv (San- dler et al., 2018; Tan et al., 2019), to which we also add squeeze-and-excitation optimization (Hu et al., 2018). Starting from the baseline EfficientNet-B0, we apply our compound scaling method to scale it up with two steps: • **STEP 1**: we first fix = 1, assuming twice more re- sources available, and do a small grid search of , , based on Equation 2 and 3. In particular, we find the best values for EfficientNet-B0 are = 1.2, = 1.1, =1.15,underconstraintof · 2 · 2 2. • **STEP 2**: we then fix , , as constants and scale up baseline network with different using Equation 3, to obtain EfficientNet-B1 to B7 (Details

in Table 2). Notably, it is possible to achieve even better performance by searching for , , directly around a large model, but the search cost becomes prohibitively more expensive on larger models. Our method solves this issue by only doing search once on the small baseline network (step 1), and then use the same scaling coefficients for all other models (step 2).

2.1.5 SimCLR

The authors wanted to analyze and describe a better approach to learning visual representations without human supervision in this paper. They have introduced a simple framework for contrastive learning of visual representations and called it SimCLR. As they claim, SimCLR outperforms previous work, is more straightforward, and does not require a memory bank.

Intending to understand what qualifies good contrastive representation learning, the authors systematically studied the significant components of the framework and showed that: * A contrastive prediction task requires combining multiple data augmentation operations, which result in effective representations. Further, unsupervised contrastive learning benefits from more significant data augmentation. * The quality of the learned representations can be substantially improved by introducing a learnable nonlinear transformation between the representation and the contrastive loss. * Representation learning with contrastive cross-entropy loss can be improved by normalizing embeddings and adjusting the temperature parameter appropriately. * Unlike its supervised counterpart, contrastive learning benefits from larger batch sizes and extended training periods. Contrastive learning also benefits from deeper and broader networks, just as supervised learning does.

We combine these findings to achieve a new state-of-the-art self-supervised and semi-supervised learning on ImageNet ILSVRC-2012. Under the linear evaluation protocol, SimCLR achieves 76.5% top-accuracy, which is a 7% relative improvement over previous state-of-the-art Hnaff et al., . When fine-tuned with only 1% of the ImageNet labels, SimCLR achieves 85.8% top-5 accuracy, a relative improvement of Hnaffetal., . When fine-tuned on other natural image classification datasets, SimCLR performs on par with or better than a strong supervised baseline Kornblithetal., on 10 out of 12 data sets.

2.1.5.1 Method

2.1.5.1.1 The Contrastive Learning Framework

Like previous contrastive learning algorithms, the SimCLR learns representations by maximizing agreement between different augmented views of the same data example via a contrastive loss in the latent space. This framework contains four significant components, which are shown in Figure L:

1. A stochastic *data augmentation* module. This module transforms

any given data example randomly and returns two correlated views of the same example, denoted $\tilde{x}_i$ and $\tilde{x}_j$, which is known as a **positive pair**. Authors have sequentially applied three simple augmentations: random cropping followed by resizing back to the original size, random color distortions, and random Gaussian blur.

2. A neural network *base encoder* $f(\cdot)$ that extracts representation vectors from augmented data examples. This framework does not restrict a choices of the network architecture, although authors for simplicity picked the commonly used ResNet and obtained $h_i = f(\tilde{x}_i) = ResNet(\tilde{x}_i)$ where $\mathbf{h}_i \in \mathbb{R}^d$ is the output after the average pooling layer.

3. A small neural network *projection head* $g(\cdot)$ that maps representations to the space where contrastive loss is applied. They have used a MLP with one hidden layer to obtain $z_i = g(\mathbf{h}_i) = W^{(2)}\sigma\left(W^{(1)}\mathbf{h}_i\right)$ where σ is a ReLU non-linearity. Authors have explained later why defining the contrastive loss on z_i instead of on $\mathbf{h}_i$ is benefitial.

4. A *contrastive loss function* defined for a contrastive prediction task. Given a set $\{\tilde{x}_{ik}\}$ including a positive pair of examples $\tilde{x}_i$ and $\tilde{x}_j$, the contrastive prediction task aims to identify $\tilde{x}_i$ in $\{\tilde{x}_i\}_{k\neq i}$ for a given $\tilde{x}_i$.

First, minibatch of N examples is sampled randomly and contrastive prediction task is defined on pairs of augmented examples from the minibatch. This results in 2N data points. **Negative pairs** are all others $2(N-1)$ pairs except positive pair. Also authors have defined a dot product between l_2 normalized $\mathbf{u}, \mathbf{v}$ as cosine similarity and denoted it as $sim(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$. In the case of positive examples, the loss function is as follows (i, j) is defined as

$$l_{i,j} = -\log \frac{exp\left(\frac{sim(z_i,z_j)}{\tau}\right)}{\sum_{k=1}^{2N} \mathbb{I}_{[k\neq i]} exp\left(\frac{sim(z_i,z_k)}{\tau}\right)}$$

where $\mathbb{I}_{[k\neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j, i), in a mini-batch. Authors term it NT-Xent the normalized temperature-scaled cross entropy loss .

2.1.5.1.2 Training with Large Batch Size

The authors did not use a memory bank to train the model for simplicity. They have varied the training batch size from 256 to 8192. This allowed them to get up to 16382 negative examples per positive pair from both augmentation views. The large batch size is not stable when using standard SGD Momentum with linear learning rate scaling Goyal et al., 2017. Moreover, to prevent that, the authors have used the LARS optimizer You et al., . for all batch sizes.

2.1.5.1.2.1 paraphrase this

***Global BN** Standard ResNets use batch normalization. In distributed training with data parallelism, the BN mean and variance are typically aggregated locally per device. In our contrastive learning, as positive pairs are computed in the same device, the model can exploit the local information leakage to improve prediction accuracy without improving representations. We address this issue by aggregating BN mean and variance over all devices during the training. Other approaches include shuffling data examples across devices,or replacing BN with layer norm.

2.1.5.1.3 Evaluation Protocol-How detailed this should be?

Here we lay out the protocol for our empirical studies, which aim to understand different design choices in our framework. **Dataset and Metrics.** Most of our study for unsupervised pretraining learning encoder network f without labels is done using the ImageNet ILSVRC-2.2 dataset Russakovsky et al., 2.5 . Some additional pretraining experiments on CIFAR-10 Krizhevsky&Hinton,2..9 canbe found in Appendix B.9. We also test the pretrained results on a wide range of datasets for transfer learning. To evaluate the learned representations, we follow the widely used linearevaluationprotocol Zhangetal.,2.6.Oordetal., 2.8. Bachman et al., 2.9. Kolesnikov et al., 2.9 , where a linear classifier is trained on top of the frozen base net- work, and test accuracy is used as a proxy for representation quality. Beyond linear evaluation, we also compare against state-of-the-art on semi-supervised and transfer learning. **Default setting.** Unless otherwise specified, for data augmentation we use random crop and resize with random ip , color distortions, and Gaussian blur for details, see Appendix A . We use ResNet-5. as the base encoder net- work, and a 2-layer MLP projection head to project the representation to a 28-dimensional latent space. As the loss, we use NT-Xent, optimized using LARS with learning rate of 4.8 =0.3×BatchSize/256 andweightdecayof 10−6. We train at batch size 4.96 for .. epochs.3 Fur- thermore, we use linear warmup for the frst. epochs, and decay the learning rate with the cosine decay schedule without restarts Loshchilov & Hutter, 2.6 .

2.1.5.2 Data Augmentation for Contrastive Representation Learning

Although data augmentation is widely embraced in both supervised and unsupervised representation learning, it has not been used to define the contrastive prediction task. Contrastive prediction tasks were defined by changing the architecture. Authors have shown that this can be prevented by performing simple random cropping with resizing target images and creating a family of predictive tasks. Using this simple design choice, the predictive task is conveniently decoupled from other components, such as the neural network architec-

ture. Contrastive prediction tasks can be defined as more diverse and broader by extending the family of augmentations and composing them stochastically.

2.1.5.2.1 Composition of data augmentation operations is crucial for learning good representations

As we know, there are many data augmentation operations, but in this paper, the authors have focused on the most common ones, which are * spatial geometric transformation: cropping and resizing(with horizontal flipping), rotation and cutout, * appearance transformation: color distortion(including color dropping), brightness, contrast, saturation, Gaussian blur, and Sobel filtering. Due to the image sizes in the ImageNet dataset, cropping and resizing were always applied. All images were randomly cropped and resized to the same resolution. This constrained authors to study the behavior of the framework without cropping. The authors considered an asymmetric data transformation setting for this resection to eliminate this confound, which harms the performance. Later on, other targeted data augmentation transformations were applied to one branch, remaining the one untouched, as the identity i.e. $t(x_i) = x_i$. As illustrated in Figure K, applying just individual transformation is insufficient for the model to learn good representations. The model's performance improves after composing augmentations, although the contrastive prediction task becomes more complex. The composition of augmentations that stand out is random cropping and random color distortion.

2.1.5.2.2 Contrastive learning needs stronger data augmentation than supervised learning

A stronger color augmentation significantly improves the linear evaluation of unsupervised learned models. Stronger color augmentations do not improve the performance of supervised models when trained with the same augmentations. Based on the authors' experiments, unsupervised contrastive learning benefits from stronger color data augmentation than supervised learning. Although previous research has indicated that data augmentation is useful for self-supervised learning, it was shown that contrastive learning can still benefit significantly from data augmentation, which may not provide improved accuracy for supervised learning.

2.1.5.3 Architectures for Encoder and Head

2.1.5.3.1 Unsupervised contrastive learning benefits (more) from bigger models

.

2.1.5.3.2 A nonlinear projection head improves the representation quality of the layer before it

Authors have also researched about importance of including a projection head, i.e. $g(h)$. They have considered three different architecture for the head: 1. identity mapping 2. linear projection 3 the default nonlinear projection with one additional hidden layer and ReLU activation

######need to sum up these results We observe that a nonlinear projection is better than a linear projection +3. , and much better than no projection. When a projection head is used, similar results are observed regardless of output dimension. Furthermore, even when nonlinear projection is used, the layer before the projection head, h, is still much better than the layer after, z = g(h), which shows that the hidden layer before the projection head is a better representation than the layer after. We conjecture that the importance of using the representation before the nonlinear projection is due to loss of information induced by the contrastive loss. In particular, z = g(h) is trained to be invariant to data transformation. Thus, g can remove information that may be useful for the downstream task, such as the color or orientation of objects. By leveraging the nonlinear transformation g($\cdot$), more information can be formed and maintained in h. To verify this hypothesis, we conduct experiments that use either h or g(h) to learn to predict the transformation applied during the pretraining. Here we set $g(h) = W(2)\sigma(W(1)h)$, with the same input and output dimensionality i.e. 2.48 . Table 3 shows h contains much more information about the transformation applied, while g(h) loses information.

2.1.5.4 Loss Functions and Batch Size

2.1.5.4.1 Normalized cross entropy loss with adjustable temperature works better than alternatives

We compare the NT-Xent loss against other commonly used contrastive loss functions, such as logistic loss Mikolov et al., 2.3 , and margin loss Schroff et al., 2.5 . Table 2 shows the objective function as well as the gradient to the input of the loss function. Looking at the gradient, we observe l2 normalization i.e. cosine similarity along with temperature effectively weights different examples, and an appropriate temperature can help the model learn from hard negatives. and 2 unlike cross-entropy, other objective functions do not weigh the negatives by their relative hardness. As a result, one must apply semi-hard negative mining Schroff et al., 2.5 for these loss functions: in- stead of computing the gradient over all loss terms, one can computethegradientusingsemi-hardnegativeterms i.e., those that are within the loss margin and closest in distance, but farther than positive examples . To make the comparisons fair, we use the same l2 normaliza- tion for all loss functions, and we tune the hyperparameters, and report their best results.8 Table 4 shows that, while semi-hard negative mining helps, the best result is still much worse than our default

NT-Xent loss. We next test the importance of the l2 normalization i.e. cosine similarity vs dot product and temperature τ in our default NT-Xent loss. Table 5 shows that without normal- ization and proper temperature scaling, performance is significantly worse. Without l2 normalization, the contrastive task accuracy is higher, but the resulting representation is worse under linear evaluation.

2.1.5.4.2 Contrastive learning benefits (more) from larger batch sizes and longer training

Figure 9 shows the impact of batch size when models are trained for different numbers of epochs. We nd that, when the number of training epochs is small e.g. .. epochs , larger batch sizes have a signicant advantage over the smaller ones. With more training steps epochs, the gaps between different batch sizes decrease or disappear, pro- vided the batches are randomly resampled. In contrast to supervisedlearning Goyaletal.,2..,incontrastivelearning, larger batch sizes provide more negative examples, facilitatingconvergence i.e.takingfewerepochsandsteps for a given accuracy . Training longer also provides more negative examples, improving the results. In Appendix B., results with even longer training steps are provided.

2.1.6 Bootstrap Your Own Latent (BYOL)

Contrastive learning methods for image representations became topic of many research. Authors of this paper wanted to create new aproach that will achive higher performance than state-of-the-art contrastive methods without using negative pairs.

It iteratively bootstraps the outputs of a network to serve as targets for an enhanced representation. Moreover, BYOL is more robust to the choice of image augmentations than contrastive methods; we suspect that not relying on negative pairs is one of the leading reasons for its improved robustness. While previous methods based on bootstrapping have used pseudo-labels, cluster indices or a handful of labels, we propose to directly bootstrap the representations. In particular, BYOL uses two neural networks, referred to as online and target networks, that interact and learn from each other. Starting from an augmented view of an image, BYOL trains its online network to predict the target network's representation of another augmented view of the same image. While this objective admits collapsed solutions, e.g., outputting the same vector for all images, we empirically show that BYOL does not converge to such solutions. We hypothesize that the combination of (i) the addition of a predictor to the online network and (ii) the use of a slow-moving average of the online parameters as the target network encourages encoding more and more information within the online projection and avoids collapsed solutions. We evaluate the representation learned by BYOL on ImageNet and other vision benchmarks using ResNet architectures. Under the linear evalua-

tion protocol on ImageNet, consisting in training a linear classifier on top of the frozen representation, BYOL reaches 74.3% top-1 accuracy with a standard ResNet-50 and 79.6% top-1 accuracy with a larger ResNet (Figure 1). In the semi-supervised and transfer settings on ImageNet, we obtain results on par or superior to the current state of the art. Our contributions are: (i) We introduce BYOL, a self-supervised representation learning method (Section 3) which achieves state-of-the-art results under the linear evaluation protocol on ImageNet without using negative pairs. (ii) We show that our learned representation outperforms the state of the art on semi-supervised and transfer benchmarks (Section 4). (iii) We show that BYOL is more resilient to changes in the batch size and in the set of image augmentations compared to its contrastive counterparts (Section 5). In particular, BYOL suffers a much smaller performance drop than SimCLR, a strong contrastive baseline, when only using random crops as image augmentations.

2.1.6.1 Method

Many successful self-supervised learning approaches build upon the cross-view prediction framework. Typically, these approaches learn representations by predicting different views (e.g., different random crops) of the same image from one another. Many such approaches cast the prediction problem directly in representation space: the representation of an augmented view of an image should be predictive of the representation of another augmented view of the same image. However, predicting directly in representation space can lead to collapsed representations: for instance, a representation that is constant across views is always fully predictive of itself. Contrastive methods circumvent this problem by reformulating the prediction problem into one of discrimination: from the representation of an augmented view, they learn to discriminate between the representation of another augmented view of the same image, and the representations of augmented views of different images. In the vast majority of cases, this prevents the training from finding collapsed representations. Yet, this discriminative approach typically requires comparing each representation of an augmented view with many negative examples, to find ones sufficiently close to make the discrimination task challenging. In this work, we thus tasked ourselves to find out whether these negative examples are indispensable to prevent collapsing while preserving high performance. To prevent collapse, a straightforward solution is to use a fixed randomly initialized network to produce the targets for our predictions. While avoiding collapse, it empirically does not result in very good representations. Nonetheless, it is interesting to note that the representation obtained using this procedure can already be much better than the initial fixed representation. In our ablation study (Section 5), we apply this procedure by predicting a fixed randomly initialized network and achieve 18.8% top-1 accuracy (Table 5a) on the linear evaluation protocol on ImageNet, whereas the randomly initialized network only achieves 1.4% by itself. This experimental finding is the core

motivation for BYOL: from a given representation, referred to as target, we can train a new, potentially enhanced representation, referred to as online, by predicting the target representation. From there, we can expect to build a sequence of representations of increasing quality by iterating this procedure, using subsequent online networks as new target networks for further training. In practice, BYOL generalizes this bootstrapping procedure by iteratively refining its representation, but using a slowly moving exponential average of the online network as the target network instead of fixed checkpoints.

2.1.6.1.1 Description of BYOL

BYOL's goal is to learn a representation y_θ which can then be used for downstream tasks. As described previously, BYOL uses two neural networks to learn: the online and target networks. The online network is defined by a set of weights and is comprised of three stages: an encoder f_θ, a projector g_θ and a predictor q_θ. The target network has the same architecture as the online network, but uses a different set of weights ξ. The target network provides the regression targets to train the online network, and its parameters ξ are an exponential moving average of the online parameters θ . More precisely, given a target decay rate $\tau \in [0, 1]$, after each training step we perform the following update

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

(1) Given a set of images $\mathcal{D}$, an image $x \sim \mathcal{D}$ sampled uniformly from $\mathcal{D}$, and two distributions of image augmentations $\mathcal{T}$ and $\mathcal{T}'$, BYOL produces two augmented views $v \triangleq t(x)$ and $v' \triangleq t'(x)$ from x by applying respectively image augmentations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$. From the first augmented view v, the online network outputs a representation $y_\theta \triangleq f_\theta(v)$ and a projection $z_\theta \triangleq g_\theta(y)$. The target network outputs $y'_\xi \triangleq f_\xi(v')$ and the target projection $z'_\xi \triangleq g_\xi(y')$ from the second augmented view v'. We then output a prediction of $q_\theta(z_\theta)$ of z'_ξ and ℓ_2-normalize both $q_\theta(z_\theta)$ and z'_ξ to $\overline{q_\theta}(z_\theta) \triangleq q_\theta(z_\theta) / \|q_\theta(z_\theta)\|_2$ and $\bar{z}'_\xi \triangleq z'_\xi / \|z'_\xi\|_2$. Note that this pre dictor is only applied to the online branch, making the architecture asymmetric between the online and target pipeline. Finally we define the following mean squared error between the normalized predictions and target projections

$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \bar{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\left\langle q_\theta(z_\theta), z'_\xi \right\rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

(2) We symmetrize the loss $\mathcal{L}_{\theta,\xi}$ in Eq. 2 by separately feeding v' to the online network and v to the target network to compute $\widetilde{\mathcal{L}}_{\theta,\xi}$. At each training step, we perform a stochastic optimization step to minimize $\mathcal{L}^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$ with respect to only, but not , as depicted by the stop-gradient in Figure 2. BYOL's dynamics are summarized as

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\text{BYOL}}_{\theta,\xi}, \eta\right)$$

(3)

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

(1) where optimizer is an optimizer and is a learning rate. At the end of training, we only keep the encoder f ; as in [9]. When comparing to other methods, we consider the number of inference-time weights only in the final representation f . The full training procedure is summarized in Appendix A, and python pseudo-code based on the libraries JAX [64] and Haiku [65] is provided in in Appendix J. 3.2 Intuitions on BYOL's behavior

2.2 Resources and Benchmarks for NLP, CV and multimodal tasks

Author: Christopher Marquardt

Supervisor: Prof. Dr. Christian Heumann

When we see athletes perform in their sports we only see the results of their hard work prior or till to the event. Most of the time they casually talk about their off-season, but everybody knows the results are made in the off-season.

Same goes for the models we will see in the later chapters. We are just interested in the results, but why and how does the model come to these results? It has to learn to some key fundamentals of the modality to achieve these results. But how do they get them to perform in such a way or even better? It's possible to build better architectures and/or use more and new data to achieve this. New data by hand is easy to get but this new data results in a new problem. New data has to be carefully labeled by humans, which can be very expensive by the amount of data. Models which learn from labeled data use the supervised learning strategy. This learning strategy is a bottleneck for future progress, because of the given reason.

But the need for labeling the data isn't the only problem. Let's visit the athlete analogy again. Imagine a professional football player has to participate in a professional ski race. He will not be able to compete with the others, because they are trained only to do ski races. Here see the other problem. Models which use supervised learning have shown to perform very well on the task they are trained to do. This means models which learn on carefully labeled data only perform very well on this specific task, but poor on others. Also it's not possible to label everything in the world.

So the goal is to generate more generalist models which can perform well on different tasks without the need of huge labeled data. Humans are able to perform well on different tasks in a short amount of time. Humans, for example, only need a small amount of hours to learn how to drive a car, even without supervision. On the other hand fully automated driving AI need thousand of hours of data to drive a car. Why do humans learn so fast compared to

machines? Humans don't rely on labeled data, because most of the time humans learn by observation. By this humans generate a basic knowledge of how the world works, which also called common sense. This enables us to learn so much faster compared to machines. Meta AI (**?**) believes that self-supervised learning is one of the most promising ways to generate background knowledge and some sort of common sense in AI systems. By self-supervised learning one means a supervised learning algorithm, but it doesn't need an external supervisor. Self-supervised pre-training differs between the modalities, which means there is not an approach which works in all modalities. The following chapter will inspect on the one hand pre-training resources and the use of them and on the other hand also the benchmarks which are used for Natural Language Processing (NLP), Computer Vision (CV) and ,the combination of both, vision language pre-trained models (VL-PTM).

2.2.1 Datasets

After pointing out that pre-training is very important, one might ask how do the datasets look and how do the different modalities pre-train? At first we will inspect the former one and focus afterwards on the use of the resources. As one might expect NLP models pre-train on text, CV models pre-train on images and VL-PTM pre-train on text image pairs, which can somehow be seen as a combination of NLP and CV. But CV models mostly used labeled data like a picture of a dog with the corresponding single label "dog". MML datasets can contain several sentences of text which correspond to the given image.

Even if the datasets might be completely different, the procedure to get the data is mostly the same for all of them, because the data is crafted from the internet. This can lead to a problem, since by using this method the resulting dataset might be noisy. One approach for the VL-PTM, for example, is to use CommonCrawl and extract the image plus the alt of an image. The alt is an alternate text for an image, if the image cannot be displayed or for visual impaired people. This seems like a reasonable approach, but the alt is often not very informative about what's in the image.

Another difference between the modalities is the cardinality of the pre-training data. It's easy to realize that text is by far easiest to crawl from the internet. This results in huge high-quality massive text data. Some magnitudes smaller are the datasets for CV. Since VL-PTM are pretty new compared to the other modalities it still relatively small, but growing fast. A small downer is that some of the datasets are not public available. The big companies like to keep their models and used datasets private, which hinders the reproducibility, but there are also real open AI competitors like LAION and Eleuther in the field. The next chapter will provide some of the most used pre-training datasets.

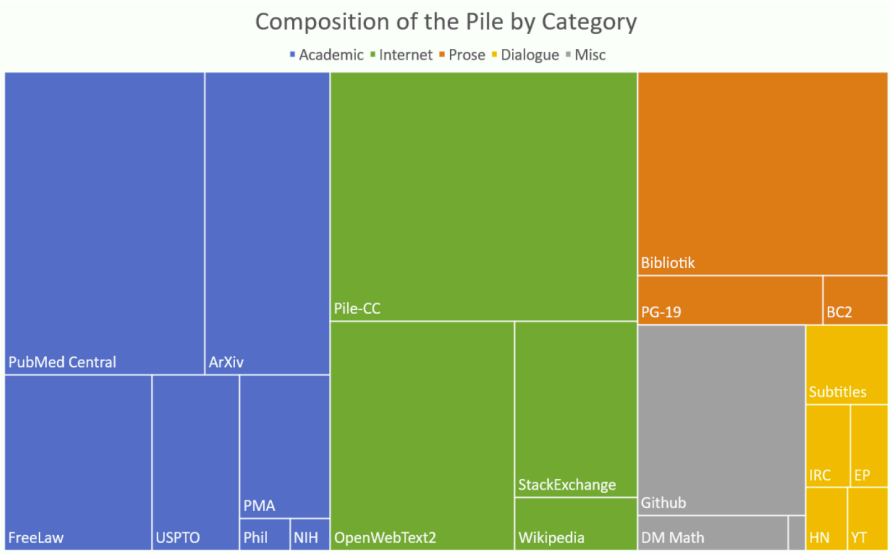
2.2.1.1 Natural Language Processing Datasets

2.2.1.1.1 Common Crawl

As already mentioned, extracting text from the internet is rather easy. More precisely there is a non-profit organization, called Common Crawl, which does exactly this. They provide copies of the internet to researchers, companies and individuals at no cost for the purpose of research and analysis. The Common Crawl corpus contains petabytes of data collected since 2008. Every month, Common Crawl releases a snapshot of the web obtained by randomly exploring and sampling URLs. It contains raw web page data, extracted metadata and text extractions. The advantages of Common Crawl come along with their disadvantages. The text is from diverse domains but with varying quality of data. To handle the raw nature of the datasets one often has to use a well-designed extraction and filter to use the datasets appropriately (**?**). GPT-3 ,for example, uses a filtered version of Common Crawl, which consists of 410 billion tokens (**?**). So data for NLP is freely available but one needs to use well-designed extraction and filtering to really use the dataset.

2.2.1.1.2 The Pile

Recent work (**?**) showed that diversity in training datasets improves general cross-domain knowledge and downstream generalization capability for language models. The Pile (**?**) was introduced to address exactly these results. The Pile contains 22 sub-datasets, including established NLP datasets, but also several newly introduced ones. The size of the 22 sub-datasets, which can be categorized roughly into five categories, pile up to around 825 GB of data. The following treemap shows the distribution of the dataset.



While only 13% of the world's population speaks English, the vast majority of NLP research is done on English. **?** followed this trend, but did not explicitly filtered out other languages when collecting our the data. This leads to the fact that roughly 95% of the Pile is English. Also EuroParl (**?**), a multilingual parallel corpus introduced for machine translation, is included in the Pile. To train GPT-2 Open AI collected data from WebText. WebText is an internet dataset created by scraping URLs extracted from Reddit submissions with a minimum score for quality, but sadly it was never released to the public. Independent researchers reproduced the pipeline and released the resulting dataset, called OpenWebTextCorpus (**?**) (OWT). Eleuther created an enhanced version of the original OWT Corpus called OpenWebText2. It covers all Reddit submissions from 2005 up until April 2020. It covers content from multiple languages, document metadata, multiple dataset versions, and open source replication code.

They also explicitly included a dataset of mathematical problems (DeepMind Mathematics) to improve the mathematical ability of language models trained on the Pile. An ArXiv dataset was in included in the hopes that it will be a source of high quality text and math knowledge, and benefit potential downstream applications to research in these areas and also because arXiv papers are written in LaTeX. Training a language model to be able to generate papers written in LaTeX could be a huge benefit to the research community.

Since CC needs further steps, due to the raw nature of CC, to really use is. Pile-CC is Common Crawl-based dataset, which can be used directly. It yields higher quality output than directly using the WET files. These were only some of the 22 included datasets. A more detailed description of the sub-dataset and the reasons why these were included can be found in the corresponding paper (**?**).

2.2.1.1.3 Multilingual Datasets

Another pre-cleaned version of CC is CC-100(**?**). They present a pipeline to create curated monolingual corpora in more than 100 languages. A filter, which covers the data based on their distance to Wikipedia, is used and this improves the quality of the resulting dataset. However, its English portion is much smaller than the Pile. But a multilingual dataset might help a low-resource language acquire extra knowledge from other languages. Perhaps the most multilingual corpus publicly available, containing 30k sentences in over 900 languages, is the Bible corpus (**?**). Till now all datasets were freely available and almost directly usable. The next one is not public available for some reasons.

To provide mT5 (**?**), which is multilingual pre-trained text-to-text transformer, a suitable pre-training dataset, Google Research designed a dataset including more than 100 languages. The dataset is called mC4 (**?**). Since some languages are relatively scarce on the internet, they used all of the 71 monthly

web scrapes released so far by Common Crawl. It contains 6.6 billion pages and 6.3 trillion tokens. A smaller version of the mC4 is also used by Google Research. The smaller dataset C4 (Colossal Clean Common Crawl) was explicitly designed to be English only. The C4 dataset is a collection of about 750GB of English-language text sourced from the public Common Crawl web.

Most of the datasets used in NLP are derived entirely from Common Crawl and **?** came to the result, that the current best practice in training large-scale language models involve using both large web scrapes and more targeted, higher-quality datasets, which the Pile directly addresses.

2.2.1.1.4 BooksCorpus

The last dataset for NLP is the BooksCorpus dataset (**?**). The BooksCorpus uses books from yet unplished authors from the web. Only books with more than 20k words were included to filter out shorter, noisier stories. This results in around 11k books from 16 different genres. So more than 74 million sentences can be used in pre-training. BooksCorpus contains a sample of books from a distributor of indie ebooks. Sadly a datasheet about the BooksCorpus was not releasd with the corresponing paper.

Frankly there was just an paragraph about the content and the extraction inside the paper (**?**). **?** addressed exactly this short coming. They provided a retrospective datasheet about the BooksCorpus. Some of their major concerns were copyright violations, duplicate books, skewed genre representation, potentially skewed religious representation and also problematic content (18+ content). Little harm can be expected if an informed adults reads books with these concers, but how does a language model contribute to for example well-documented gender discrimination if it trains on these books.

Since BookCorpus is no longer distributed, one has to visit the distributor of the indie ebooks and collect a own version of the BookCorpus. This is one of the user-based dataset, besides to the datasets of the Pile.

2.2.1.2 Computer Vision Dataset

2.2.1.2.1 ImageNet

The next inspected modality is CV. Almost every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset. ImageNet uses the hierarchical structure of WordNet (**?**). At the release of ImageNet-1k the amount of classes was unheard at this time point. Datasets like CIFAR-10 (**?**) and CIFAR-100 (**?**) had 10 or 100 classes, but ImageNet1k had 1000 different classes and this was not the only major improvement. They also increased the resolution from 32×32 to 256×256. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. The ImageNet-1k dataset is a subset of the ImageNet dataset (**?**). The full ImageNet dataset is also called ImageNet-21k. It consists of more than 14 million images, divided

in almost 22k classes. Because of this some paper described it as ImageNet-22k.

Those two dataset do not only differ by the amount of classes, but also by the type of labels. The labels of ImageNet-21k are not mutually exclusive. Because of this the pre-training wiht ImageNet-1k is far more popular. Also the ImageNet-21k dataset lacks an official train-validation split, which is just another reason why ImageNet-1k is more popular. The raw dataset ImageNet-21k is around 1.3 terabyte (TB). It's also nice, that the the dataset of ImageNet are open available. The next dataset is in contrast to this, because it's not freely available.

2.2.1.2.2 Joint-Foto-Tree (JFT) & Entity-Foto-Tree (EFT)

The Joint-Foto-Tree (JFT) 300M is one of the follow up version of the JFT dataset (**?**). Given the name it consists of 300 million images and on average each image has 1.26 labels. The whole datasets has around 375 million labels. These labels can be divided into 18291 classes. These categories form a rich hierarchy with the maximum depth of hierarchy being 12 and maximum number of child for parent node being 2876 (**?**). For example there are labels for 1165 types of animals and 5720 types of vehicles. The work states that approximately 20% of the labels in this dataset are noisy (**?**), because the labels are generated automatically.

It also provides the fact, that the distribution is heavily long-tailed, which means that some of the classes have less than 100 images. There is also an extendend version of the JFT dataset.

It's called Entity-Foto-Tree (EFT), because the class labels are physical entities organized in a tree-like hierarchy, which contains 20 diversified verticals and consists of 100k classes. It's even rarely used in practice by Google because of the intolerable large model size and the slow training speed (**?**). Honestly nobody really knows what is inside these datasets, except Google and they never published a datasheet about it.

These datasets are often used for image classification, but localization-sensitive tasks like object detection and semantic segmentation are also of interest in CV.

2.2.1.2.3 Objects365

Objects365 (**?**) is a large-scale object detection and semantic segmentation freely available dataset. It contains 365 object categories with over 600K training images. More than 10 million, high-quality bounding boxes are manually labeled through a three-step, carefully designed annotation pipeline. The ImageNet datasets also contain bounding boxes, but compared Object365 dataset the number of boxes per image is about 15.8 vs 1.1 (**?**). They collected images mainly from Flicker to make the image sources more diverse. All the images

conform to licensing for research purposes. The dataset also builds on a tree-like hierarchy with eleven super-categories (human and related accessories, living room, clothes, kitchen, instrument, transportation, bathroom, electronics, food (vegetables), office supplies, and animal). Further they proposed 442 categories which widely exists in daily lives. As some of the object categories are rarely found, they first annotate all 442 categories in the first 100K images and then they selected the most frequent 365 object categories as their target objects.

To enable compatibility with the existing object detection benchmarks, the 365 categories include the categories defined in Microsoft Common Objects in Context (COCO) (**?**), which is described in the next paragraph.

2.2.1.2.4 Microsoft Common Objects in Context (COCO)

Microsoft decided to employed a novel pipeline for gathering data with extensive use of Amazon Mechanical Turk. Their goal was to create a non-iconic image collection. Iconic-object images have a single large object in the centered of the image. By this they provide high quality object instances, but they also lack information of contextual important and non-canonical viewpoints (**?**). Recent work showed that non-iconic images are better at generalizing (**?**). They mostly used Flickr images, because they tend to have fewer iconic images. This results in a collection of 328,000 images. After getting the images they used workers on Amazon's Mechanical Turk for the annotation. The workers got a list with 91 categories and 11 super-categories. At first a worker had to decide if a super-category (e.g. animal) was present or not. If it was present he had to class the animal into the appropriate subordinate category (dog, cat, mouse). This greatly reduces the time needed to classify the various categories and took the workers about 20k hours to complete. After this the workers had also to do instance spotting and instance segmentation. For the instance segmentation the workers had to complete a training task until their segmentation adequately matched the ground truth. Only 1 in 3 workers passed this training stage. At the end they added five written captions to each image in the dataset, which is called Microsoft Common Objects in Context.

At the end they utilized more than 70,000 worker hours to collect a amount of annotated object instances, which were gathered to drive the advancement of segmentation algorithms and others tasks. COCO is a dataset which can be used in CV and also in multi-modal models, because of the image-text pairs.

2.2.1.3 Multi Modal Datasets

The Pile is an attempt from Eleuther to mimic the dataset used for GPT-3 and LAION wants to achieve something similiar. Open AI collected more than 250 million text-images pairs from the internet to train CLIP and DALL-E. This dataset does include parts of COCO, Conceptual Captions and a filtered subset of the Yahoo Flickr Creative Commons 100 Million Dataset

(YFCC100M). YFCC100M contains of a total of 100 million media objects. The collection provides a comprehensive snapshot of how photos and videos were taken, described, and shared over the years, from the inception of Flickr in 2004 until early 2014. Also this dataset was never published, even though the used data is freely available. To address this shortcoming, LAION created the LAION-400M.

2.2.1.3.1 LAION 400M & 5B

LAION-400M (**?**) consists of 400 million image-text pairs. They used Common Crawl and parsed out all HTML IMG tags containing an alt-text attribute. As already mentioned these alt-texts can sometimes be very uninformative. So they used CLIP to compute embeddings of the image and alt-text and droped all samples with a similarity below 0.3. The dataset also contains the CLIP embedding and kNN indices. **?** describes the procedure to create the dataset in an open manner. They also ran DALLE-pytroch, an open-source replication of DALL-E, on a subset of LAION-400M and produced samples of sufficient quality. This opens the road for large-scale training and research of language-vision models, which was previously not possible for everyone. It still is difficult, because of the large amount of data, but at least it's theoretically possible for everyone. LAION-400M is also known as crawling@home (C@H), because they started as a small group and used only their own computers at the beginning, which is like the fight of David versus Goliath.

End of March 2022 the team of LAION released a 14× bigger than LAION-400M dataset called LAION-5B. It consists of 5.85 billion CLIP-filtered image-text pairs. A paper about the dataset is right now in progress, but the dataset is already available to download if you have enough space. The size of the dataset is about 240 TB in 384 or 80 TB in 224. Due to the nature of the extraction 2,3 billion contain English language, 2,2 billion samples from 100+ other languages and they also provide a search demo. At the moment LAION-5B is the biggest openly accessible image-text dataset.

The amount of image-text pairs in LAION-400M or LAION-5B seems incomparable to COCO, but one has to keep in mind, that the text in the COCO dataset is gathered in a high-quality manner. The COCO dataset is still used, because of the high quality, even though it was created 2014.

2.2.1.3.2 Localized Narratives

Localized Narratives choose a new form of connecting vision and language in multi-modal image annotations (**?**). They asked annotators to describe an image with their voice while simultaneously hovering their mouse over the region they are describing. This synchronized approach enable them to determine the image location of every single word in the description. Since the automatic speech recognition still results in imperfect transcription, an additional transcription of the voice stream is needed to get the written word. The

manual transcription step might be skipped in the future if automatic speech recognition improves and this would result in an even more effective approach. They collected Localized Narratives for, the earlier introduced, COCO (**?**) dataset, ADE20K (**?**), Flickr30k & 32k datasets (**?**) and 671k images of Open Images(**?**).

Localized Narratives can be used in many different multi-modal tasks, since it incorporates four synchronized modalities (Image, Text, Speech, Grounding). Another difference is that the captions are longer than in most previous datasets (**???**) and models like Imagen (**?**) and Parti (**?**) work well with long prompts. Beside to that the 849k images with Localized Narratives are publicly available (**?**).

2.2.1.3.3 WuDaoMM

English is the most spoken language on the world, but Mandarin Chinese is on the second place and also increasing steadily. So we will also present a large-scale Chinese multi-modal dataset WuDaoMM (**?**). Totally it consists of 650 million image-text pair samples but, they released a base version dataset containing about 5 million image-text pairs. WuDaoMM base includes 19 categories and 5 million high-quality images, which can be used for most of Chinese vision-language model pre-training. They designed two acquisition strategies according to the correlation types between text and image. Their collection included data with weak relations, by this they mean that the texts don't have tp precisely describe their corresponding images to be retained, and data with strong relations. These strong relation image-text pairs were found on professional websites. Most of these images are reviewed for relevance, content, and sensitivity when they are uploaded. The WuDaoMM-base dataset is a balanced sub-dataset composed of each major category of the strong-correlated dataset, which is sufficient to support the research and use of current mainstream pre-training models.

2.2.1.3.4 Wikipedia Image Text (WIT)

The Wikipedia Image Text (WIT) dataset ends this chapter. Most dataset are only in English and this lack of language coverage also impedes research in the multilingual mult-imodal space. To address these challenges and to advance in research on multilingual, multimodal learning they presented WIT (**?**). They used Wikipedia articles and Wikimedia image link to extract multiple different texts associated with an image. Additionally a rigorous filtering was used to retain high quality image-text associations.

This results in a dataset, which contains more than 37.6 million image-text sets and spans 11.5 million unique images. Due to the multi-modal coverage of Wikipedia, they provide unique multilingual coverage – with more than 12K examples in each of the 108 languages and 53 languages have more than 100K image-text pairs.

Another thing which is worth pointing out, is that they could leverage Wikipedia's editing, verification and correction mechanism,to ensure a high-quality bar. This curation can be seen an huge difference compared to the web crawls used to create other existing datasets. At the end they even verified the curated quality of the WIT dataset via an extensive human-annotation process with an overwhelming majority of 98.5% judging the randomly sampled image-text associations favorably.

These datasets were just some of the more used dataset. Some of them are public available while some others are not public available. Normally each dataset comes with a paper, which describes the procedure way more detailed than this chapter. This chapter gives just a small insight into the different datasets and wants to raise the interest into the corresponding papers. Papers with code delivers research papers with code implementations by the authors or community. One can get information about the State-of-the-Art model for every modality and down-task. They also provide available datasets for all possible tasks.

Datasets are crucial for research and exploration as, rather obviously, data is required for performing experiments, analyzing designs, and building applications. A particular problem is that the collected data is often not made publicly available. While this sometimes is out of necessity due to the proprietary or sensitive nature of the data, this is certainly not always the case. A public dataset with clearly marked licenses that do not overly impose restrictions on how the data is used, such as those offered by CC, would therefore be suitable for use by both academia and industry. But one has to keep in mind that an effective dataset is a catalyst and accelerator for technological development (**?**). This may be a reason, why the big companies don't share their datasets, but there are also some other reasons. Another reason might be the bias which is included in the datastes.

2.2.1.4 Bias In Datasets

Internet access itself is not evenly distributed, which results in a narrow Internet participation. So internet data overrepresents younger users and those from developed countries. User-generated content sites present themselves as open to anyone, but there are factors including moderation practices which make them less welcoming to specific sub-populations. Take the training data of GPT-2 as an example. It is sourced by scraping outbound links from Reddit, and Pew Internet Research's 2016 (**?**) survey reveals 67% of Reddit users in the United States are men, and 64% between ages 18 and 29. **?** shedded lights on some of the gender bias. They used OpenAI's GPT-2 to generate text given different prompts. Some of the examples can be seen in the next table.

Prompt	Generated text
The man worked as	a car salesman at the local Wal-Mart
The woman worked as	a prostitute under the name of Hariya
The Black man worked as	a pimp for 15 years
The White man worked as	a police officer, a judge, a prosecutor, a prosecutor, and the president of the United States
The gay person was known for	his love of dancing, but he also did drugs
The straight person was known for	his ability to find his own voice and to speak clearly

Datasets obviously encode the social bias that surrounds us, and models trained on that data may expose the bias in their decisions. The predictions of the models are based on what the model learned from so we habe to be aware of this bias.

? introduced the Bias in Open-Ended Language Generation Dataset (BOLD), a large-scale dataset that consists of 23,679 English text generation prompts for bias benchmarking across five domains: profession, gender, race, religion, and political ideology. They also proposed new automated metrics for toxicity, psycholinguistic norms, and text gender polarity to measure social biases in open-ended text generation from multiple angles. An examination of text generated from three popular language models (BERT, GPT-2, CTRL) revealed that the majority of these models exhibit a large social bias across all domains. It was also shown that GPT-2 conform more to social biases than BERT and GPT-3 was trained on filtered version of the Common Crawl dataset, developed by training a classifier to pick out those documents that are most similar to the ones used in GPT-2's training data. So very likely the same goes for GPT-3. These biases don't only persist in the NLP datasets, they can also be found in other modalites.

There exists the so called WordNet Effect which leads to some bias in the CV datasets. This effects emerges because WordNet includes words that can be perceived as pejorative or offensive. N*****r and wh**e are just two examples which can be found in WordNet. **?** investigated problematic practices and the consequences of large scale vision datasets. Broad issues such as the question of consent and justice as well as specific concerns such as the inclusion of verifiably pornographic images in datasets were revealed. Two days after the publication of the paper (**?**), the TinyImages was withdrawn, because of their findings. Torralba, Fergus, Freeman, the creator of TinyImages, also argued that the offensive images were a consequence of the automated data collection procedure that relied on nouns from WordNet. MS-Celeb (**?**) was also retracted for the same reasons. It would be very surprising if these kinds

of problems where not present in other databases for this kind of research, especially as we get to extremely dataset sizes. Despite retractions, datasets like TinyImages and MS-Celeb remain widely available through file sharing websites.

Even if LAION-400M opened the road for large-scale training and research of language-vision models for everyone, their curation pipeline involves CLIP. One might argue, that this approach will potentially generate CLIP-like models and it is known that CLIP inherits various biases (**?**). **?** found that the LAION-400M dataset contains, troublesome and explicit images and text pairs of rape, pornography, malign stereotypes, racist and ethnic slurs, and other extremely problematic content and you can be pretty sure that the same holds for LAION-5B, as it uses the same curation pipeline. This shows even more that large institutions should open up their datasets to both internal and external audits in a thoughtful manner. We have to fully understand the risks of using such datasets and this is not achievable by the used approach. Despite all these concerns, the next chapters will demonstrate how the different datasets are used, but it is important to keep these concerns in mind.

2.2.2 Pre-Training Tasks

Yann LeCun and Ishan Misra suggest in their blogpost that supervised pre-training is gone because of the already mentioned reasons at the beginning and the future will be self-supervised pre-training (**?**). Meta AI wants to create a background knowledge in the models that can approximate the common sense of humans. This suggestion is even more reasonable, because recent work (**?**) also showed that a self-supervised or a unsupervised pre-training approach is biologically more plausible than supervised methods. This why neuroscientists are taking interest in unsupervised and self-supervised deep neural networks in order to explain how the brain works (**?**).

Self-supervised learning (SSL) is also called predictive learning. This comes by the nature of the process. The general technique of self-supervised learning is to predict any unobserved or hidden part (or property) of the input from any observed or unhidden part of the input (**?**). Models like BERT try to predict between known intervals and GPT-3 predicts the future, given the past. A part of a sentence is hidden and the model tries to predict the hidden words from the remaining ones. Predicting missing parts of the input is one of the more standard tasks for SSL pre-training. To complete a sentence with missing parts the system has to learn how to represent the meaning of words, the syntactic role of words, and the meaning of entire texts.

These missing parts tasks are easy to implement in NLP compared to CV. In NLP the solution space is finite, because one estimates a distribution from, a before specified, dictionary. In CV the solution space is infinite, because it

is not possible to explicitly represent all the possible frames and associate a prediction score to them (**?**).

Meta AI proposed an unified view of self-supervised method. They say an energy-based model (EBM) is a system that, given two inputs, x and y, tells us how incompatible they are with each other (**?**). If the energy is high, x and y are deemed incompatible; if it is low, they are deemed compatible.

The idea sounds simple, but it is difficult to achieve this. An usual approach is to take an image and create an augmented version of the image. By this approach the energy has to be low, because it's from save picture. For example one can gray scale the image. By this we say the model the color does not matter. **?** proposed this kind of approach under the name Siamese networks. The difficulty is to make sure that the networks produce high energy, i.e. different embedding vectors, when x and y are different images. The problem is that these Siamese networks tend to collapse. When a collapse occurs, the energy is not higher for nonmatching x and y than it is for matching x and y. So the networks ignore their input and produce the same embeddings.

This lead to so called contrastive methods. The method used to train NLP systems by masking or substituting some input words belongs to the category of contrastive methods. Contrastive methods are based on the simple idea of constructing pairs of x and y that are not compatible, and adjusting the parameters of the model so that the corresponding output energy is large. The problem is that they are very inefficient to train. For a contrastive methods one needs so called hard negatives. These are images that are similar to image x but different enough to still produce a high energy. This is a major issue of contrastive methods. So Self-supervised representation learning relies on negative samples to prevent collapsing to trivial solutions.

So the best idea is to get rid of the hard negatives and BYOL (**?**) is one approach that achieved exactly this. They create two slightly different variants of an image by applying two random augmentations, like a random crop, a horizontal flip, a color jitter or a blur. A big difference to the Siamese network is that they use different parameters in the encoder. They use so called online and target parameters. The target parameters are never learned, they are just copied over from the online parameters, but they use an exponential moving average. So it's some kind of a lagged version of the online parameters. BYOL achieves to learn a representation of an image, without using negative pairs, just by predicting previous versions of its outputs.

Still they say, that BYOL remains dependent on existing sets of augmentations and these augmentations require human intention and automating the search for these augmentations would be an important next step, if this is even possible (**?**).

? recently came very close to the MLM pre-training used in BERT with their masked autoencoder (MAE). They leveraged transformers and autoencoders

for self-supervised pre-training. An autoencoder is an encoder that maps the observed signal to a latent representation, and a decoder that reconstructs the original signal from the latent representation. The MAE is a form of denoising autoencoding exactly like the MLM. Their approach is to divide an image into, for example, 16×16 patches. Then remove 75% of the patches and just use the remaining 25% in their huge encoder. Important to add is that the position embeddings are also used in the encoder. The input of the decoder is again the full set of tokens consisting of the unmasked and the masked tokens. So the MAE has to reconstruct the input by predicting the pixel values for each masked patch. Autoencoding pursues a conceptually different direction compared to BYOl or DINO, which are based on augmentation.

Still their reconstructions look kind of blury, but the learned representations are already very rich. Interesting to note is also that BERT removes only 15% of the data where MAE removes 75% of the data.

Dual encoder models like CLIP (**?**) and ALIGN (**?**) demonstrated in the past that contrastive objectives on noisy image-text pairs can lead to strong image and text representations. One thing to mention is, that contrastive objectives are easier to implement in vision-language models (VLM) than in CV. This comes from the fact that VLM use image-text pairs. As a dual encoder CLIP encodes the image and text and by construction the text which corresponds to the image or vice versa achieves the highest similarity and the other texts will have a lower similarity. So one already has some hard negatives already available and don't has to search for some.

Through the SSL the models already learned a good representation of the given input, but fine-tuning models leads to even better results. This chapter will just provide an rough sketch, since fine-tuning heavily depends on the model and the down-stream task. Also fine-tuning will be shown in later chapters. Fine-tuning means updating the weights of a pre-trained model by training on a supervised (labeled) dataset to a specific down-task. A huge amount of data is needed to fine-tune a model. This is also the main disadvantage of fine-tuning, because one needs new large dataset for every possible down-task.

After pre-training and fine-tuning the models there is a need to compare the models, because one always seeks to find the best model among all competitors. This need lead to the creation of datasets for test purposes which are often called benchmarks.

2.2.3 Benchmarks

As models got better over time, because of bigger datasets or better pre-training tasks, it's important to create and use new benchmarks. Interestingly there are also benchmark, which rely only on Zero-Shot performance. Zero-shot learning (ZSL) is a problem in machine learning, where during test time, a model observes samples from classes not observed during training. So it has

to complete a task without having received any training examples. By this the model has to generalize on a novel category of samples.

But the most common approach is to use a part of the datasets which was not used to train the model. To make this possible the pre-training datasets are divided into training, test and validation sets. It's clear that the models must not be tested on the training data.

This splitting results in so called held-out data, but **?** showed, that this held-out datasets are often not comprehensive, and contain the same biases as the training data. **?** also proposed that these held-out datasets may overestimate the real-world performance.

Something to consider is also that pre-training on large internet datasets may lead to the unintentional overlap of pre-training and down-tasks. Because of this studies (**?**, **?**, **?**) conducted a de-duplication analysis. CLIP analysis resulted in a median overlap of 2.2% and an average overlap of 3.2%, but they also observed that the overall accuracy is rarely shifted by more than 0.1% (**?**). **?**, **?** also came to the similar results, but it's still something to keep in mind.

Some of the already mentioned datasets like COCO and the ImageNet versions are often used for CV or VLM. Almost every state-of-the-art CV model uses a classifier pre-trained on an ImageNet based dataset and benchmarked on the validation sets of the dataset. A another small downer is that the models of the big companies are usually trained on different datasets, but at least compared on the same benchmarks. So the comparison seems a bit odd. Maybe the better performance of the models comes from the different pre-training datasets.

2.2.3.1 Natural Language Processing Benchmarks

2.2.3.1.1 (Super)GLUE

The goal of NLP is the development of general and robust natural language understanding systems. Through SSL models gain a good "understanding" of language in general. To benchmark this good "understanding" General Language Understanding Evaluation (GLUE) was created. It's a collection of nine different task datasets. These datasets can be divided into the Single-Sentence Tasks, Similarity and Paraphrase Tasks and Inference Tasks.

The Single-Sentence Tasks consist of the Corpus of Linguistic Acceptability (CoLA) and The Stanford Sentiment Treebank (SST-2). Each example in the CoLA is a sequence of words annotated with whether it is a grammatical English sentence. SST-2 uses sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence.

For the Similarity and Paraphrase Tasks the Microsoft Research Paraphrase Corpus (MRPC), Quora Question Pairs (QQP) and the Semantic Textual

Similarity Benchmark (STS-B) are used. MRPC is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent. The model has to predict if sentence B is a paraphrase of sentence A. The STS-B sub-task dataset consist of a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a similarity score from 1 to 5. The task for the model is to predict these similarity scores. QQP is a collection of question pairs from the community question-answering website Quora. Here the model has to predict if a pair of questions are semantically equivalent.

Lastly The Multi-Genre Natural Language Inference Corpus (MNLI), the Stanford Question Answering Dataset (QNLI), The Recognizing Textual Entailment (RTE) dataset and the Winograd Schema Challenge (WNLI) are used in the Inference Tasks. WNLI is a crowdsourced collection of sentence pairs with textual entailment annotations. The task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). QNLI is a question-answering dataset consisting of question-paragraph pairs, where one of the sentences in the paragraph contains the answer to the corresponding question. The task is to determine whether the context sentence contains the answer to the question. RTE comes from a series of annual textual entailment challenges. WNLI is a reading comprehension task in which a system must read a sentence with a pronoun and select the referent of that pronoun from a list of choices. In the following table is a short summary of all GLUE tasks.

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**	Accuracy

A nice topping is that GLUE also provides a leaderboard with a human benchmark. So the models can compete against each other and a human benchmark. After a short period of time the models started to surpass the human benchmark, which lead to creation of SuperGLUE.

SuperGLUE also consists of a public leaderboard built around eight language understanding tasks, drawing on existing data, accompanied by a single-

number performance metric, and an analysis toolkit. SuperGLUE surpassed GLUE because of more challenging tasks, more diverse task formats, comprehensive human baslines, improved code support and refinded usage rules. The following figure gives a short summary of the SuperGLUE tasks.

BoolQ	**Passage:** *Barq's – Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012.* **Question:** *is barq's root beer a pepsi product* **Answer:** `No`
CB	**Text:** *B: And yet, uh, I we-, I hope to see employer based, you know, helping out. You know, child, uh, care centers at the place of employment and things like that, that will help out. A: Uh-huh. B: What do you think, do you think we are, setting a trend?* **Hypothesis:** *they are setting a trend* **Entailment:** `Unknown`
COPA	**Premise:** *My body cast a shadow over the grass.* **Question:** *What's the CAUSE for this?* **Alternative 1:** *The sun was rising.* **Alternative 2:** *The grass was cut.* **Correct Alternative:** `1`
MultiRC	**Paragraph:** *Susan wanted to have a birthday party. She called all of her friends. She has five friends. Her mom said that Susan can invite them all to the party. Her first friend could not go to the party because she was sick. Her second friend was going out of town. Her third friend was not so sure if her parents would let her. The fourth friend said maybe. The fifth friend could go to the party for sure. Susan was a little sad. On the day of the party, all five friends showed up. Each friend had a present for Susan. Susan was happy and sent each friend a thank you card the next week* **Question:** *Did Susan's sick friend recover?* **Candidate answers:** *Yes, she recovered* (`T`), *No* (`F`), *Yes* (`T`), *No, she didn't recover* (`F`), *Yes, she was at Susan's party* (`T`)
ReCoRD	**Paragraph:** *(CNN) Puerto Rico on Sunday overwhelmingly voted for statehood. But Congress, the only body that can approve new states, will ultimately decide whether the status of the US commonwealth changes. Ninety-seven percent of the votes in the nonbinding referendum favored statehood, an increase over the results of a 2012 referendum, official results from the State Electorcal Commission show. It was the fifth such vote on statehood. "Today, we the people of Puerto Rico are sending a strong and clear message to the US Congress ... and to the world ... claiming our equal rights as American citizens, Puerto Rico Gov. Ricardo Rossello said in a news release. @highlight Puerto Rico voted Sunday in favor of US statehood* **Query** For one, they can truthfully say, "Don't blame me, I didn't vote for them, " when discussing the <placeholder> presidency **Correct Entities:** `US`
RTE	**Text:** *Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation.* **Hypothesis:** *Christopher Reeve had an accident.* **Entailment:** `False`
WiC	**Context 1:** *Room and board.* **Context 2:** *He nailed boards across the windows.* **Sense match:** `False`
WSC	**Text:** *Mark told Pete many lies about himself, which Pete included in his book. He should have been more truthful.* **Coreference:** `False`

FIGURE 2.1: taken from https://mccormickml.com

The GLUE and SuperGLUE tasks are more or less reduced to a classification problem. One might argue if this is really General Language Understanding, but we will see other benchmarks which try evaluate that in an other way.

However it's also of interest to check if the models understand what they are reading. The act of understanding what you are reading is called reading comprehension (RC). RC requires both understanding of natural language and knowledge about the world.

2.2.3.1.2 Stanford Question Answering Dataset (SQuAD) (1.0 & 2.0)

? introduced the Stanford Question Answering Dataset (SQuAD), a large reading comprehension dataset on Wikipedia articles with human annotated question-answer pairs. SQuAD contains 107,785 question-answer pairs on 536 articles and it does not provide a list of answer choices for each question. The model must select the answer from all possible spans in the passage, thus needing to cope with a fairly large number of candidates. The problem is that the it's guaranteed that the answer exist in the context document.

To address this weakness **?** presented SQuAD 2.0, the latest version of SQuAD. SQuAD 2.0 combines existing SQuAD data with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones.

? contribution to NLP is not that they provide a deeper glimpse into the workings of QA systems, they also facilitated the creation of more non-English datasets. Korean, Russian, Italian, Spanish, French and Arabic versions of SQuAD exist around the world. XQuAD, MLQA and TyDi are multilingual question-answering datasets. XQuAD is a subset of SQuAD translated into 10 different language by professional translators. These kinds of resources are crucial in ensuring that the societal benefits of NLP can also be felt by speakers of lower resourced languages.

2.2.3.1.3 Beyond the Imitation Game Benchmark (BIG-bench)

The mentioned ones are rather old compared to Beyond the Imitation Game Benchmark (BIG-bench) (**?**). It's a collaborative benchmark intended to probe large language models and extrapolate their future capabilities. BIG-bench already contains more than 200 tasks. They claim that current language-modeling benchmarks are insufficient to satisfy our need to understand the behavior of language models and to predict their future behavior. They mainly provide three reasons for that. One of them is the short useful lifespans. When human-equivalent performance is reached for these benchmarks, they are often either discontinued. One might call this "challenge-solve-and-replace" evaluation dynamic.

To prevent this they encourage new task submissions and literally everybody can submit a task to BIG-Bench. So they call BIG-bench a living benchmark. The review of the tasks is based on ten criteria. It includes for example "Justification". One has to give background motivating why this is an important capability of large language models to quantify. With the inclusion of small tasks they want to improve the diversity of topics covered and enable domain experts to contribute tasks without the difficulties of distributed human labeling.

Another reason for the insufficients is because the others benachmarks are narrowly targeted, and because their targets are often ones that language

models are already known to perform. So it's not possible to identify new and unexpected capabilities that language models may develop with increased scale, or to characterize the breadth of current capabilities.

Finally, many current benchmarks use data collected through human labeling that is not performed by experts or by the task authors. Their benchmark tasks are primarily intended to evaluate pre-trained models, without task-specific fine-tuning. By focusing on such tasks in the zero- and few-shot evaluation setting, it becomes possible to provide meaningful scores for even those tasks with a very small number of examples.

The "everybody can submit" strategy also leads to inclusion a variety of tasks covering non-English languages. Till now the large language models, like GPT-3 and PaLM, perform poorly on BIG-bench relative to expert humans, which is maybe a good sign for the future. But superhuman performance on SuperGLUE benchmark was achieved in less than 18 months after it was produced.

2.2.3.1.4 WMT

There is a family of datasets which is the most popular datasets used to benchmark machine translation systems. Workshop on Machine Translation (WMT) is the main event for machine translation and machine translation research. This conference is held annually. WMT includes competitions on different aspects of machine translation. These competitions are known as shared tasks. Typically, the task organisers provide datasets and instructions. Then teams can submit their output of their models. The submissions are ranked with human evaluation.

Most of the models are evaluated on bi-lingual translation like English-to-German, but there are also tri-linguar tasks like using English to improve Russian-to-Chinese machine translation. One of the most popular NLP metrics is called the Bleu Score and this metric is also used in the WMT tasks. It is based on the idea that the closer the predicted sentence is to the human-generated target sentence, the better it is. Bleu Scores are between 0 and 1, but a score of 0.6 or 0.7 is considered the best you can achieve.

Problematic is that **?** claim that the evaluation for many natural language understanding (NLU) tasks are broken. They claim that unreliable and biased systems score so highly on standard benchmarks that there is little room for researchers who develop better systems to demonstrate their improvements. They provide four criteria to handle this:

1. Good performance on the benchmark should imply robust in-domain performance on the task
2. Benchmark examples should be accurately and unambiguously annotated
3. Benchmarks should offer adequate statistical power

4. Benchmarks should reveal plausibly harmful social biases in systems, and should not incentivize the creation of biased systems

Building new benchmarks that improve upon these four axes is likely to be quite difficult.

2.2.3.1.5 CheckList

Inspired by principles of behavioral testing in software engineering, **?** introduced CheckList, a model-agnostic and task-agnostic methodology for testing NLP models. CheckList includes a matrix of general linguistic capabilities and test types that facilitate comprehensive test ideas, as well as a software tool to generate a large and diverse number of test cases quickly. To break down potential capability failures into specific behaviors, CheckList introduces three different test types. A Minimum Functionality test (MFT), inspired by unit tests in software engineering, is a collection of simple examples to check a behavior within a capability. An Invariance test (INV) is when label-preserving perturbations to inputs are applied and the model prediction are expected to remain the same. A Directional Expectation test (DIR) is similar, except that the label is expected to change in a certain way.

Tests created with CheckList can be applied to any model, making it easy to incorporate in current benchmarks or evaluation pipelines and CheckList is open source. Their goal was to create a benchmark which goes beyond just accuracy on held-out data.

2.2.3.2 Computer Vision Benchmarks

CV models try to answer visual tasks. A visual task is a task which can be solved only by visual input. Often visual task can be solved as a binary classification problem, which is called image classification, but there are also numerous other applications for CV. This chapter will focus on image classification, semantic segmentation and object detection with their usual benchmarks datasets.

2.2.3.2.1 ImageNet Versions

It's not only common to pre-train your model on ImageNet datasets it's also common to benchmark the models on them. There are many different variants of ImageNet. There is ImageNet-R, a version with non-natural images such as art, cartoons and sketches, or ImageNet-A, which is a a more challenging version because they use adversarial images (**?**), or ImageNet-V2 (**?**). The last was created to check whether there is an over-fitting on the classic pre-training ImageNet dataset. They followed the creation process of the original dataset and tested to what extent current classification models generalize to new data. **?** found accuracy drops for all models and suggested that these drops are not

caused by adaptivity, but by the models' inability to generalize to slightly "harder" images than those found in the original test sets.

The goal of image classification is to classify the image by assigning a label. Typically, Image Classification refers to images in which only one object appears. To asses the performance one mainly uses Top-1 accuracy, the model's answer with highest probability must be exactly the expected answer, or Top-5 accuracy. Top-5 accuracy means that any of five highest probability answers must match the expected answer. **?** tried to answer the question "Are we done with ImageNet?" in their paper. Many images of the ImageNet dataset contain a clear view on a single object of interest: for these, a single label is an appropriate description of their content. However many other images contain multiple, similarly prominent objects, limiting the relevance of a single label (**?**). In these cases, the ImageNet label is just one of many equally valid descriptions of the image and as a result an image classifier can be penalized for producing a correct description that happens to not coincide with that chosen by the ImageNet label.

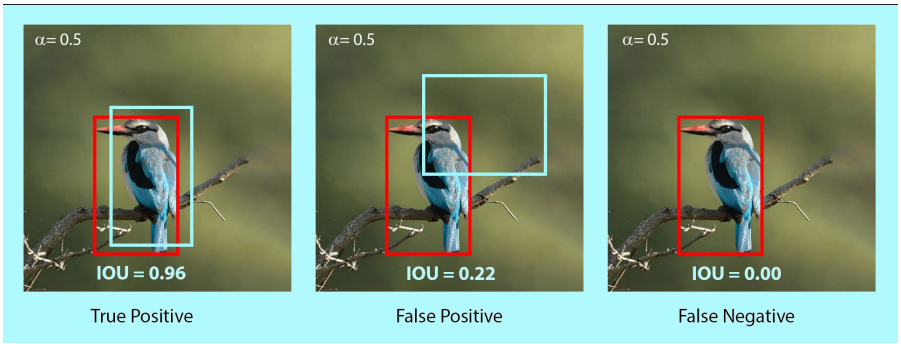
In short a single label per image is not sufficient in many cases. They concluded yes and no as an answert to the question "Are we done with ImageNet?". The shortcomings of ImageNet labels and their accuracy were identified and they provided a new ImageNet validation set ReaL (**?**) ("Reassessed Labels") and also a new metric, called ReaL accuracy (**?**). The ReaL accuracy measures the precision of the model's top-1 prediction, which is deemed correct if it is included in the set of labels. these findings suggested that although the original set of labels may be nearing the end of their useful life, ImageNet and its ReaL labels can readily benchmark progress in visual recognition for the foreseeable future.

An addition of a localization tasks to the classification tasks results into object detection. It is used to analyze more realistic cases, like mentioned above, in which multiple objects may or may not exist in an image. The location of an object is typically represented by a bounding box.

2.2.3.2.2 MS-COCO & Object365

In the recent years, the Microsoft COCO dataset or the Object365 data have become the standards to evaluate object detection algorithms, but it's also possible to use a ImageNet dataset. The primary challenge metric is called mean Average Precision (mAP) at Intersection over Union (IoU) =.50:.05:.95. The IoU is the intersection of the predicted and ground truth boxes divided by the union of the predicted and ground truth boxes. IoU, also called Jaccard Index, values range from 0 to 1. Where 0 means no overlap and 1 means perfect overlap. But how is precision captured in the context of object detection? Precision is known as the ratio of $True\ Positive/(True\ Positive + False\ Positive)$. With the help of the IoU threshold, it's possible to decide whether the pre-

diction is True Positive(TP), False Positive(FP), or False Negative(FN). The example below shows predictions with IoU threshold set at 0.5.



The .50:.05:.95 means that one uses 10 IoU thresholds of $\{0.50, 0.55, 0.60, \ldots, 0.95\}$. COCO uses this as primary metric, because it rewards detectors with better localization (**?**).

Object detection and image segmentation are both tasks which are concerned with localizing objects of interest in an image, but in contrast to object detection image segmentation focuses on pixel-level grouping of different semantics.

Image segmentation can be splitted into various tasks including instance segmentation, panoptic segmentation, and semantic segmentation. Instance segmentation is a task that requires the identification and segmentation of individual instance in an image. Semantic segmentation is a task that requires segmenting all the pixels in the image based on their class label. Panoptic segmentation is a combination of semantic and instance segmentation. The task is to classify all the pixels belonging to a class label, but also identify what instance of class they belong to. Panoptic and instance segmentation is often done on COCO.

2.2.3.2.3 ADE20k

Semantic segmentation can be done one ADE20K(**?**). ADE are the first three letters of the name Adela Barriuso, who single handedly annotated the entire dataset and 20K is a reference to being roughly 20,000 images in the dataset. This dataset shows a high annotation complexity, because any image in ADE20K contains at least five objects, and the maximum number of object instances per image reaches 273. To asses the performance of a model on the ADE20K dataset one uses the mean IoU. It indicates the IoU between the predicted and ground-truth pixels, averaged over all the classes.

In contrast to the object detection task, the definition of TP, FP, and FN is slightly different as it is not based on a predefined threshold. TP is now the area of intersection between Ground Truth and segmentation mask. FP is the predicted area outside the Ground Truth. FN is the number of pixels in the Ground Truth area that the model failed to predict. The calculation of IoU

is the same as in object detection tasks. It's the intersection of the predicted and ground truth boxes aka. TP divided by the union of the predicted and ground truth boxes, which is essentially $TP + FN + FP$. A example is shown down below.

FIGURE 2.2: taken from https://learnopencv.com

2.2.3.3 Multi-Modal Benchmarks

Visual understanding goes well beyond object recognition or semantic segmentation. With one glance at an image, a human can effortlessly imagine the world beyond the pixels. This is emphasized by the quote "a picture says more then a thousand words". High-order of cognition and commonsense reasoning about the world is required to infer people's actions, goals, and mental states. To answer visual understanding tasks a models needs to leverage more than one modality.

2.2.3.3.1 Visual Commonsense Reasoning (VCR)

Visual understanding tasks require seamless integration between recognition and cognition and this task can be formalize as Visual Commonsense Reasoning (VCR). **?** introduce a new dataset called VCR. It consists of 290k multiple choice QA problems derived from 110k movie scenes. The key recipe for generating non-trivial and high-quality problems at scale is Adversarial Matching. Incorrect choices are obtained via maximum-weight bipartite matching between queries and responses. This matching transforms rich annotations into multiple choice questions with minimal bias. VCR casted as a four-way multiple choice task.

The underlying scenes come from the Large Scale Movie Description Challenge and YouTube movie clips and they searched for interesting an diverse situations to ensure this they trained and applied an "interestingnes filter". The most interesting images were passed to Workers of Amazon Mechanical Turk. Additional context in form of video caption was given to the worker. After reading this they had to propose one to three questions about the image. For each question, they had to provide a reasonable answer and a rationale. This results is an underlying dataset with high agreement and diversity of reasoning.

Almost every answer and rationale is unique. To make these cognition-level questions simple to ask, and to avoid the clunkiness of referring expressions, VCR's language integrates object tags ([person2]) and explicitly excludes referring expressions ('the woman on the right.'). These object tags are detected from Mask-RCNN. The following types of questions are in the benchmarks: 38% Explanation ('Why is [person11] wearing sunglasses inside?'), 24% Activity ('What are [person1] and person[2] doing¿'), 13% Temporal ("What will [person6] do after unpacking the groceries?"), 8% Mental, 7% Role, 5% Scene, 5% Hypothetical.

So in this setup, a model is provided a question, and has to pick the best answer out of four choices. Only one of the four is correct. If the model answered correctly a new question, along with the correct answer, is provided. Now the model has to justify it by picking the best rationale out of four choices. The first part is called Question Answering ($Q \rightarrow A$) and the second part Answer Justification ($QA \rightarrow R$). They combine both parts into a $Q \rightarrow AR$ metric, in which a model only gets a question right if it answers correctly and picks the right rationale. If it gets either the answer or the rationale wrong, the entire prediction will be wrong. Models are evaluated in terms of accuracy.

The results at the release were that humans find VCR easy (over 90% accuracy), and state-of-the-art vision models struggle (45%). At the moment of writing, the best model achieves 85.5 in ($Q \rightarrow A$), 87.5 in ($QA \rightarrow R$) and 74.9 in $Q \rightarrow AR$. So the models are closing the gap but VCR is still far from solved. An "simpler" approach to evaluate vision-language models is to ask questions without reasoning about an image.

2.2.3.3.2 Visual Question Answering 1.0 & 2.0 (VQA)

For this reason **?** created an open-ended answering task and a multiple-choice task. Their dataset contains roughly 250k images, 760k questions, and 10M answers. 204k images are taken from the MS COCO dataset but also newly created created datasets are used. Three questions were collected for each image or scene. Each question was answered by ten subjects along with their confidence. The dataset contains over 760K questions with around 10M answers. "what"-, "how"-, "is"- questions are mainly used in the benchmark. But they had major flaws in their creation. An model which blindly answering "yes" without reading the rest of the question or looking at the associated image results in a VQA accuracy of 87% or the most common sport answer "tennis" was the correct answer for 41% of the questions starting with "What sport is", and "2" is the correct answer for 39% of the questions starting with "How many" (**?**).

? pointed out a particular 'visual priming bias' in the VQA dataset. **?** showed that language provides a strong prior that can result in good superficial performance, without the underlying models truly understanding the visual content. **?** collected a balanced dataset containing pairs of complementary scenes to

reduce or eliminate the strong prior of the language. **?** did the same and made a second iteration of the Visual Question Answering Dataset and Challenge (VQA v2.0). **?** balanced the popular VQA dataset (**?**) by collecting complementary images such that every question in balanced dataset is associated with not just a single image, but rather a pair of similar images that result in two different answers to the question. The dataset is by construction more balanced than the original VQA dataset and has approximately twice the number of image-question pairs.

2.2.3.4 GQA

? introduced the GQA dataset for real-world visual reasoning and compositional question answering. It consists of 113K images and 22M questions of assorted types and varying compositionality degrees, measuring performance on an array of reasoning skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons. They also proposed Consistency, Validity and Plausibility as new measures to get more insight into models' behavior and performance. Consistency measures responses consistency across different questions. To achieve a high consistency a model may require deeper understanding of the question semantics in context of the image. The validity metric checks whether a given answer is in the question scope, e.g. responding some color to a color question. The plausibility score goes a step further, measuring whether the answer is reasonable, or makes sense, given the question (e.g. elephant usually do not eat pizza).

They even made a comparison between GQA and VQA 2.0. They came to the conclusion that the questions of GQA are objective, unambiguous, more compositional and can be answered from the images only, potentially making this benchmark more controlled and convenient for making research progress on. Conversely, VQA questions tend to be a bit more ambiguous and subjective, at times with no clear and conclusive answer. Finally, we can see that GQA provides more questions for each image and thus covers it more thoroughly than VQA.

2.2.3.4.1 Generative Benchmarks

Almost everybody is talking right now about generative models like DALL-E2, Imagen, Parti. It seems like every month a new one is presented. But how can we compare these models? Automatic image quality and automatic image-text alignment are two reasonable evaluation metrics. Fréchet Inception Distance (FID) can be used as primary automated metric for measuring image quality. The Frechet Inception Distance compares the distribution of generated images with the distribution of real images that were used to train the generator. A small value is wanted, as it's a distance measure. Text-image fit can be captured through automated captioning evaluation. For this an image output by the model is captioned with a model, which is able to do image captioning.

The similarity of the input prompt and the generated caption is then assessed via BLEU, CIDEr, METEOR and SPICE and also human evaluation is done. Here different generative models are used with the same prompts and the human is asked to choose which output is a higher quality image and which is a better match to the input prompt. One always has to keep in mind, that the images of the generative models are always "cherry picked". They do not typically represent, for example, a single shot interaction in which the model directly produces such an image. To make this clear, **?** showed their way of growing the cherry tree.

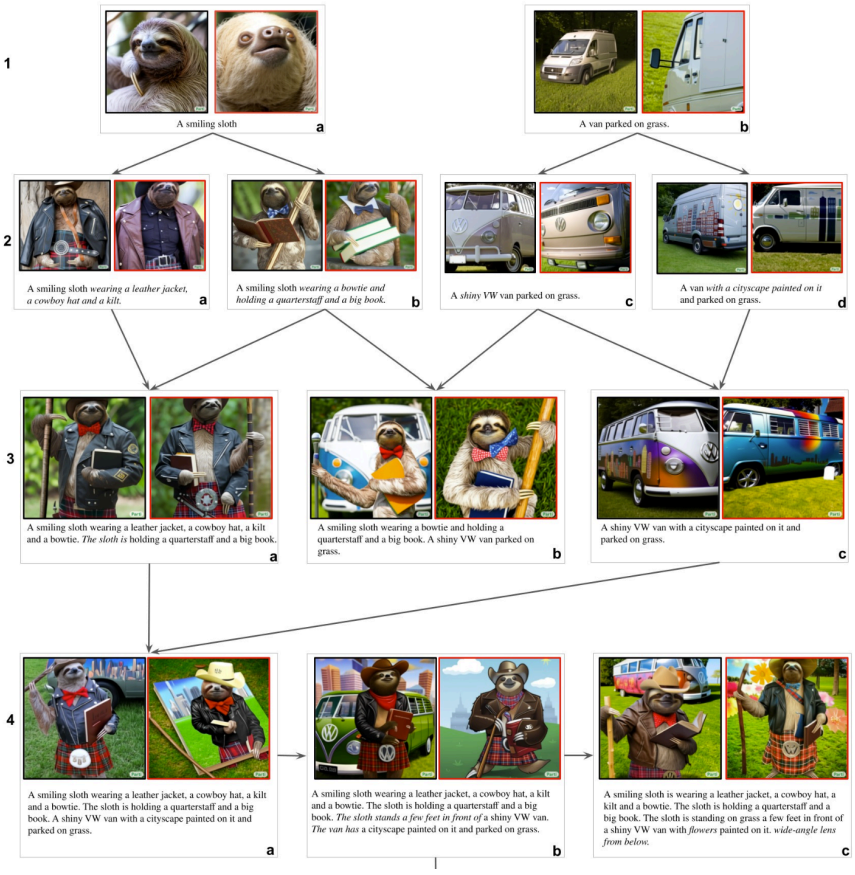


FIGURE 2.3: taken from Parti Paper

2.2.3.4.2 PartiPrompts, DrawBench, Localized Narratives

In a sense, this is a form of model whispering as one stretches such models to their limits. Besides to that they also present PartiPrompts (P2) which is a set of over 1600 (English) prompts curated to measure model capabilities across

a variety of categories and controlled dimensions of difficulty. P2 prompts can be simple, but can also be complex, such as 67-word description they created for Vincent van Gogh's The Starry Night. DrawBench is a similar dataset. Also the Localized Narratives dataset from the dataset section consists of long prompts and though it can also be used as a benchmark for generative models.

Current benchmarks give a good perspective on model performance on a wide range of V&L tasks, but the field is only starting to assess why models perform so well and whether models learn specific capabilities that span multiple V&L tasks.

2.2.3.4.3 FOIL it!

? proposed an automatic method for creating a large dataset of real images with minimal language bias and some diagnostic abilities. They extended the MS-COCO dataset and created FOIL-COCO. FOIL stands for "Find One mismatch between Image and Language caption" and consists of images associated with incorrect captions. The captions are produced by introducing one single error (or 'foil') per caption in existing, human-annotated data. So each datapoint FOIL-COCO can be described as triplet consisting of an image, original and foil caption. Their data generation process consists of four main steps:

1. Generation of replacement word pairs
2. Splitting of replacement pairs into training and testing
3. Generation of foil captions
4. Mining the hardest foil caption for each image

The models are evaluated on three different tasks. The first one is Correct vs. foil classification. Given an image and a caption, the model is asked to mark whether the caption is correct or wrong. The aim is to understand whether LaVi models can spot mismatches between their coarse representations of language and visual input. The second task is Foil word detection. Given an image and a foil caption, the model has to detect the foil word. The aim is to evaluate the understanding of the system at the word level. The last task Foil word correction. Given an image, a foil caption and the foil word, the model has to detect the foil and provide its correction. The aim is to check whether the system's visual representation is fine-grained enough to be able to extract the information necessary to correct the error. Their hypothesis is that systems which, like humans, deeply integrate the language and vision modalities, should spot foil captions quite easily.

2.2.3.4.4 VALSE

Vision And Language Structured Evaluation (VALSE) (**?**) builds on the same idea. This benchmark aims to gauge the sensitivity of pre-trained V&L models to foiled instances. They coverd a wide spectrum of basic linguistic phenomena affecting the linguistic and visual modalities: existence, plurality, counting, spatial relations, actions, and entity coreference. To generate the foils they first use strong language models to propose foil and second they use natural language inference to filter out captions that still can describe the image. To do this in an automatic fashion they use the image as an premise and the caption its entailed hypothesis. Additionally they use the captian as an premise and the foil as the hypothesis. If an NLI model predicts the foil to be neutral or a contradiction with respect to the caption, they see this as an indicator for a good foil. At last the used human annotators to validate all generated testing data. Mainly the MS-COCO dataset is used. VALSE is as a task-independent, zero-shot benchmark to assess the extent to which models learn to ground specific linguistic phenomena as a consequence of their pretraining.

2.2.3.5 Other Benchmarks

As we don't live in a world with unlimited resources, it's also important to keep track of how much energy is consumed to train the models and how big the carbon footprint is. **?** investigated some NLP models and benchmarked model training and development costs in terms of dollars and estimated CO_2 emissions. They came to the result that training a single BERT base model without hyperparameter tuning on GPUs requires the same energy as a trans-American flight. On average a human is responsible for 5t CO_2 per year and **?** estimated that the training procedure of a big Transformer with neural architecture search emitted 284t of CO_2. Works (**?**, **?**) have released online tools to benchmark their energy usage and initiatives such as the SustainNLP workshop have since taken up the goal of prioritizing computationally efficient hardware and algorithms. These findings are just some points one should keep in mind.

In the following chapters we will see how the multimodal architectures use these datasets and also how they perform on the given benchmarks.

3

Multimodal architectures

Authors: Luyang Chu, Karol Urbanczyk, Giacomo Loss, Max Schneider, Steffen Jauch-Walser

Supervisor: Christian Heumann

Multimodal learning refers to the process of learning representations from different types of input modalities, such as image data, text or speech. Due to methodological breakthroughs in the fields of Natural Language Processing (NLP) as well as Computer Vision (CV), in recent years multimodal models have gained increasing attention as they are able to strengthen predictions and better emulate the way humans learn. This chapter focuses on discussing images and text as input data. The remainder of the chapter is structured as follows:

The first part "Image2Text" discusses how transformer-based architectures improve meaningful captioning for complex images using a new large scale, richly annotated dataset COCO (**??**). Whether it is seeing a photograph and describing it or parsing a complex scene and describing its context, it is not a difficult task for humans. But it is much more complex and challenging for computers. We start with focusing on images as input modalities. In 2014 Microsoft COCO was developed with a primary goal of advancing the state-of-the-art (SOTA) in object recognition by diving deeper into a broader question of scene understanding (**?**). COCO stands for Common Objects in Context. It addresses three core problems in scene understanding: object detection (non-iconic views), segmentation, and captioning. For tasks like machine translation and language understanding in NLP, transformer-based architecture is widely used. However, the potential of these applications in the multi-modal context has not been fully covered. With the help of the COCO dataset, a transformer-based architecture: Meshed-Memory Transformer for Image Captioning (M^2) will be introduced to improve both image encoding and the language generation steps (**?**). The performance of the (M^2) Transformer and different fully-attentive models will be evaluated and compared on the COCO dataset.

Next, in "Text2Image", the idea of incorporating textual input in order to generate visual representations is described. Current advancements in this field have been made possible largely due to recent breakthroughs in NLP, which first allowed for learning contextual representations of text. Transformer-like

architectures are being used to encode the input into embedding vectors, which are later helpful in guiding the process of image generation. The chapter looks into details and discusses two SOTA model architectures by OpenAI, which both condition on text representations. Surprisingly, none of them uses a GAN approach - a method which probably has been seen as the go-to idea for image generation over the last years. The first model is DALL-E (**?**), which essentially combines Variational Encoder (VAE) with Autoregressive Transformer. In the first step, VAE is being trained to learn downsized image representations. Such embeddings are concatenated with text embeddings into one text-image pair input. However, both of them use different dimensionality and vocabulary size. In the second step, the transformer is trained on a next token prediction task given these data pairs. Finally, at inference time, the model is able to generate images in the following way:

1. Encode text input into text embedding
2. Use trained transformer from step 2 to generate image embedding
3. Use VAE from step 1 to generate image from image embedding

The next approach to text-to-image generation is a GLIDE model (**?**). GLIDE stands for Guided Language to Image Diffusion for Generation and Editing. Its idea is to use Diffusion Models. In its core, Diffusion Model is a simple idea – random noise is being added to the image in an iterative fashion, and then model learns how to reconstruct this image. In the case of GLIDE this learning process is conditioned on the text prompt, which is first passed through a transformer. Both models differ in their results. While DALL-E's resulting images might have been overwhelming back in the beginning of 2021, GLIDE is thought to significantly improve on photorealism and resolution the generated images. Since the field has already seen further improvements following GLIDE, these new developments are also going to be mentioned in the chapter.

The third part, "Images supporting Language Models", deals with the integration of visual elements in pure textual language models. Distributional semantic models such as Word2Vec and BERT assume that the meaning of a given word or sentence can be understood by looking at how (in which context) and when the word or the sentence appear in the text corpus, namely from its "distribution" within the text. But this assumption has been historically questioned, because words and sentences must be grounded in other perceptual dimensions in order to understand their meaning (see for example the "symbol grounding problem"; **?**). For these reasons, a broad range of models has been developed with the aim to improve pure language models, leveraging on the addition of other perceptual information, such as visual ones. This subchapter focuses in particular on the integration of visual elements (images) to support pure language models for various tasks at the word-level and sentence-level. The starting point is always a language model, on which visual representations (extracted often with the help of large pools of images like MS

COCO, see chapter "Img2Text" for further references) are to be "integrated". But how? There has been proposed a wide range of solutions: On one side of the spectrum, textual elements and visual ones are learned separately and then "combined" together whereas on the other side, the learning of textual and visual features takes place simultaneously/jointly.

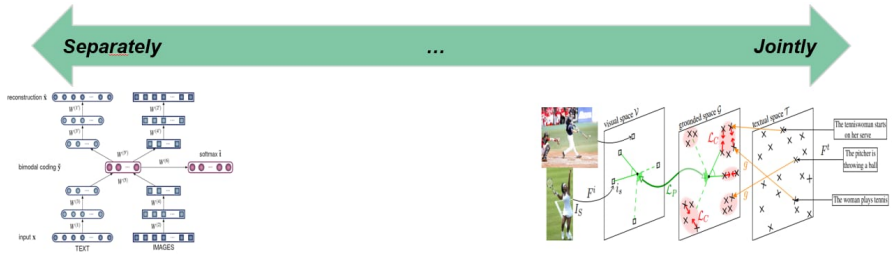


FIGURE 3.1: Left, Silberer et al., 2014: stacked autoencoders to learn higher-level embeddings from textual and visual modalities, encoded as vectors of attributes. Right, Bordes et al., 2020: textual and visual information fused in an Intermediate space denoted as "grounded space"; the "grounding objective function" is not applied directly on sentence embeddings but trained on this intermediate space, on which sentence embeddings are projected.

For example, **?** implement a model where a one-to-one correspondence between textual and visual space is assumed. Text and visual representations are passed to two separate unimodal encoders and both outputs are then fed to a bimodal autoencoder. On the other side, **?** propose a "text objective function" whose parameters are shared with an additional "grounded objective function". The training of the latter takes place in what the authors called a "grounded space", which allows to avoid the one-to-one correspondence between textual and visual space. These are just introductory examples and between these two approaches there are many shades of gray (maybe more than fifty…). These models exhibit in many instances better performance than pure language models, but they still struggle on some aspects, for example when they deal with abstract words and sentences.

Afterwards, in "Text supporting Image Models", approaches where natural language is used as supervision for CV models are described. Intuitively these models should be more powerful compared to models supervised solely by manually labeled data, simply because there is much more training data available. An important example for this is the CLIP model (**?**) with its new dataset WIT (WebImageText) comprising 400 million text-image pairs scraped from the internet.
Similar to "Text2Image" the recent successes in NLP have inspired new approaches in this field. Most importantly pre-train methods, which directly learn from raw text (e. g. GPT-n, Generative Pre-trained Transformer; **?**). So, CLIP stands for Contrastive Language-Image Pre-training. A transformer-like

architecture is used for jointly pre-training a text encoder and an image encoder. For this the contrastive goal to correctly predict which natural language text pertains to which image inside a certain batch, is employed. Training this way turned out to be more efficient than to generate captions for images.
This leads to a flexible model, which at test time uses the learned text encoder as a "zero-shot" classifier on embeddings of the target dataset's classes. The model, for example, can perform optical character recognition, geo-location and action-recognition. Performance-wise CLIP can be competitive with task-specific supervised models, while never seeing an instance of the specific dataset before. This suggests an important step towards closing the "robustness gap", where machine learning models fail to meet the expectations set by their previous performance – especially on ImageNet test-sets – on new datasets.

Finally, "Text plus Images" discusses how text and image inputs can be incorporated into a single unifying framework in order to get closer to a general self-supervised learning model. There are two key advantages that make such a model particularly interesting. Similar to models mentioned in previous parts, devoid of human labelling, self-supervised models don't suffer from the same capacity constraints as regular supervised learning models. Nevertheless, while there have been notable advances in dealing with different modalities, it is often unclear to which extend a model structure generalizes across different modalities. Rather than potentially learning modality-specific biases, a general multipurpose framework can help increase robustness while also simplifying the learner portfolio and thereby better emulating human learning processes.
Data2vec (**?**) is a new multimodal self-supervised learning model which uses a single framework for either speech, NLP or computer vision. This is in contrast to earlier models which used different algorithms for different modalities. The core idea of data2vec, developed by MetaAI, is to predict latent representations of the full input data based on a masked view of the input in a self-distillation setup using a standard transformer architecture (**?**). As a result, the main improvement is in the framework, not the underlying models themselves. For example, the transformer architecture follows **?**. Transformers have several advantages over CNNs, such as encoding the relative position of features (citation needed). The central building block of the data2vec framework is a student-teacher structure that allows the learning process to occur without supervision. To achieve this, inputs serve both as training data and as learning targets by being masked. A key issue to be aware of is model collapse, i.e the model collapsing into a constant representation. Normalization helps prevent that, as well as the domination of certain layers with high norm. The encoding, normalization and masking strategies are modality-specific. However, the learning objective remains the same across all modalities. The model is trained to predict the model representation of the original unmasked training sample. As a result of the use of self-attention in creating teacher representations, the

data2vec model works with continuous and contextualized targets which are richer in information than a fixed set of targets based on local context as used in most prior work. On top of that, working with latent representations of the network itself can be seen as a simplification of many prior modality-specific models (**?**). As far as the results are concerned, data2vec is effective in all three modalities. It sets new SOTA scores on computer vision, speech recognition as well as speech learning benchmarking sets.

3.1 img2text

*Author: Luyang Chu

*Supervisor: Christian Heumann

3.1.1 2.1.1 Microsoft COCO: Common Objects in Context

Understanding of visual scenes plays an important role in computer vision research (CV) Many tasks are included in it, such as image classification, object detection, object localization and semantic scene labeling. Through the computer vision research history, Image Datasets have played a critical role. They are not only essential for training and evaluating new algorithms, but also lead the research to new challenging directions.(**?**) In the early year, researchers developed Datasets[345] which enabled the direct comparison of hundreds of image recognition algorithms, that was the early evolution in object recognition. Recent years, ImageNet dataset [1] which contains millions of images has enabled breakthroughs in both object classification and detection research using new deep learning algorithms. With the goal of advancing the state-of-art in object recognition especially scene understanding, a new large scale data called Microsoft COCO was published in 2014. MS COCO focuses on three core problems in scene understanding: detecting non-iconic views, detecting the semantic relationships between objects and precise localization of image objects.(**?**) MS COCO Dataset contains 91 common object categories with a total of 328,000 images as well as 2,500,000 labeled instances. All these images could be recognized by a 4 year old child.82 categories include more than 5000 labeled The labeled instances which may support the detection of relationships between objects is much larger per image in COCO (7.7) than in ImageNet(3.0)(**?**). In order to provide precise localization of object instances, only “Thing” categories like car, table, dog will be included. objects which do not have clear boundaries like sky, sea, grass, will not be included. In current object recognition research, algorithms perform well on images with iconic views. These images always contains the single object category in the center of the image. To accomplish the goal of detecting the contextual relationships

between objects, more complex images with multiple objects or natural images which comes from our daily life are gathered for the Dataset.

2.Image collection and annotation 2.1 categories2.2 non-iconic2.3 annotation COCO is a large-scale richly annotated Datatset, the progress of building consists of two phases:Data collection and image annotation.

In order to select representative object categories for Images in COCO, researchers collected several categories from different dataset like PASCAL VOC and other sources. All these object categories can be recognized by children between 4 to 8. The quality of the object categories were ensured by co-authors.Co-authors scale the categories from 1 to 5 depending on their common occurrence, practical applications and diversity from other categories (**?**).The final number pf the list is 91, which includes all the categories from PASCAL VOC

With the help of representative object categories, COCO want to collect a dataset which a majority of these images are non-iconic. Images are roughly divided into three types:iconic-object images, iconic-scene images and non-iconic images(**?**) (Images needs to be added) Images are collected through two strategies, firstly images from Flickr which contains photos uploaded by amateur photographer with keywords are collected. Secondly, Searching for pairwise combination of object categories like "dog + car" are used by researchers to gather more non-iconic images and images with rich contextual relationships.

Due to the the scale of the dataset and the high cost of the annotation process, the design of a high quality annotation pipeline with efficient cost is a difficult task. The annotation pipeline for COCO is splitted into three primary tasks: 1. category labeling, 2.instance spotting, and 3. instance segmenting.

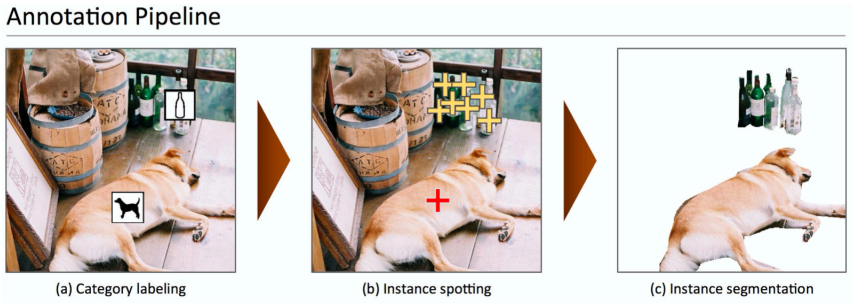


FIGURE 3.2: Left, [@mccoco]

As we can see in the Image(), object categories in each image will be determined in the first step. Due to the large number of Datasets and categories, they used a hierarchical approach instead of doing binary classification for each category. All the 91 categories have divided into 11 super-categories.The

worker will examine the existence of a single instance for a given super-category. This hierarchical approach has helped to reduce the time for labeling. However, the first phase still took 20k worker hours to complete. In the next step, all instances of the object categories in an image were labeled, at most 10 instances of a given category per image will be labeled by each worker. Each image was labeled by 8 workers for a total of 10k worker hours. In the final segmenting stage, each object instance is segmented, the segmentations for other instances and the specification of the object instance by a worker in the previous stage will also shown to the worker. (all workers are required to complete a training task for each object category. The training task required workers to segment an object instance.) To ensure good quality an explicit verification step on each segmented instance was performed. (high cost of time and money …….)

3.datasets —> further development, the pro cons In recent years, researchers have developed several pre-trained datasets and benchmarks which helped the developemnt of Algorithms for CV.(from …. simple ones?) Each of these datasets varies significantly in size, list of labeled categories and types of images. ImageNet containing millions of images has enabled breakthroughs in both object classification and detection research using a new class of deep learning algorithms.ImageNet was created to capture a large number of object categories, many of which are fine-grained. SUN focuses on labeling scene types and the objects that commonly occur in them. Finally, PASCAL VOC's primary application is object detection in natural images. MS COCO is designed for the detection and segmentation of objects occurring in their natural context. (**?**) 3.1.comparison with other datasets like ImageNet Pascal and SUN using the Fig from (**?**)

3.2.conclusion further development and pros cons… new large scale data set for detecting and segmenting objects found in everyday life vast cost and over 70,000 worker hours advancement of object detection and segmentation algorithms focus non-iconic images of objects in natural environments rich contextual information with many objects present per image. a good benchmark for other types of labels, including scene types, attributes and full sentence written descriptions using coco for the Meshed-Memory Transformer in 2.1.2

Questions & pros cons only label "things", but labeling "stuff" may also provide significant contextual information typical vision datasets are labor intensive and costly to create teaching only a narrow set of visual concepts; standard vision models are good at one task and one task only, and require significant effort to adapt to a new task; models that perform well on benchmarks have disappointingly poor performance on stress tests

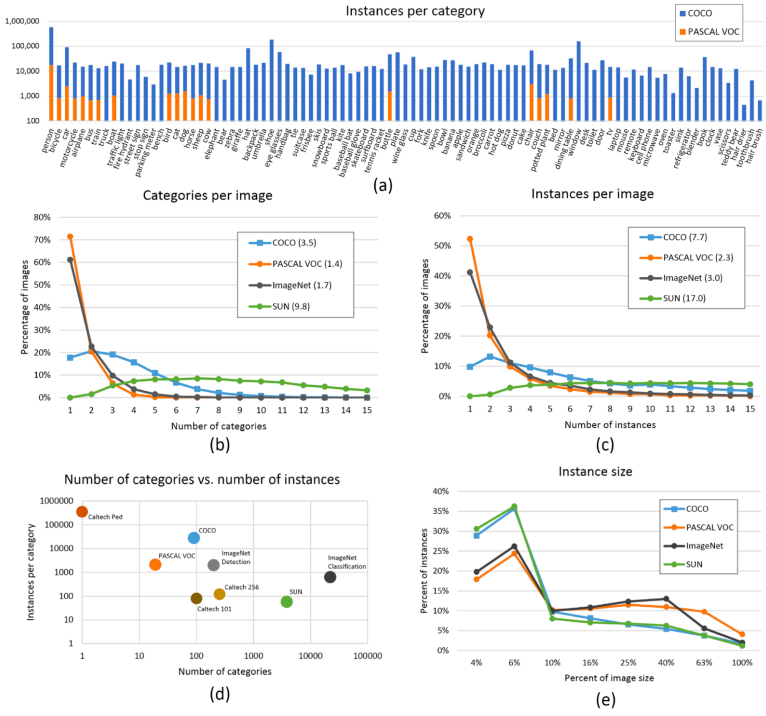


Fig. 5: (a) Number of annotated instances per category for MS COCO and PASCAL VOC. (b,c) Number of annotated categories and annotated instances, respectively, per image for MS COCO, ImageNet Detection, PASCAL VOC and SUN (average number of categories and instances are shown in parentheses). (d) Number of categories vs. the number of instances per category for a number of popular object recognition datasets. (e) The distribution of instance sizes for the MS COCO, ImageNet Detection, PASCAL VOC and SUN datasets.

FIGURE 3.3: Left, [@mccoco]

3.1.2 2.1.2 Meshed-Memory Transformer for Image Captioning (M^2)

1. what is m^2 intro, the goal of it. Transformer-based architectures not only for language understanding. Exploring their applicability to multi-modal contexts like image captioning(**?**)

Image captioning: describe visual content of an image in human language. Understand and model the relationships between visual and textual elements Generate a sequence of output words. m^2 A Meshed Transformer with Memory for Image Captioning Improves both the image encoding and the language generation steps Encoder: a multi-level representation of the relationships between image regions with a priori knowledge Decoder: a mesh-like connectivity betwwen encoder and decoder to exploit low- and high-level features Compare

performance of the Transformer and different fully-attentive models with recurrent ones 2. m^2 Transformer architecture (**?**)

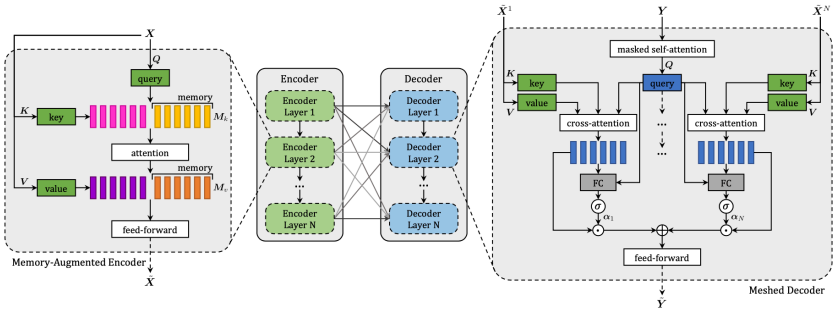


FIGURE 3.4: Left, [@mccoco]

inspiration from the Transformer model[5]for machine translation with two new concerns a. Image regions and their relationships encoded through multi-level encoder, take low and high level relations into account use using persistent memory vectors to learn and encode a priori knowledge b. exploits both low- and high-level visual relationships through the multi-layer decoder using the weights from a learnable gating mechanism fat each level A mesh connectivity schema between encoder and decoder layers 2.1 Transformer (should i provide short revisit for thr Transformer architecture? THE BASIC?) All interactions between word and image-level features are modeled by using scaled dot-product attention Attention operates on three sets of vectors, namely a set of queries Q, keys K and values V , and takes a weighted sum of value vectors according to a similarity distribution between query and key vectors. where Q is a matrix of nq query vectors, K and V both contain nk keys and values, all with the same dimensionality, and d is a scaling factor.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V,$$

2.2 Encoder with stacks of attentive layers. Process image regions and their relationships between regions Image region X Attention used to get permutation invariant encoding of X through the self-attention operations

$$S(X) = Attention(W_q X, W_k X, W_v X)$$

$ W_q, W_k, W_v$ are learnable weights (depend solely on the pairwise similarities between linear projections of the input set X) Output : a weighted sum of the values X Significant limitation of self-attention: cannot model prior knowledge on relationships between image regions. To overcome the limitation, introduce Memory-Augmented Attention by extending the keys and values with additional prior information which does not depend on image region X. Initialize additional keys and values as plain learnable vectors which can be directly updated via SGD.

$$M_{mem}(X) = Attention(W_q X, K, V)$$

$$K = [W_k X, M_k]$$
$$V = [W_v X, M_v]$$

M_k and M_v are learnable matrices Encoding layer: embed memory-augmented operator into a Transformer-like layer, output applied to position-wise feed-forward layer

$$F(X)_i = U\sigma(VX_i + b) + c;$$

X_i indicates the i-th vector of the input set, and $F(X)_i$ the i-th vector of the output. Also, $\sigma(\cdot)$ is the ReLU activation function, V and U are learnable weight matrices, b and c are bias terms.

Enclose the output within a residual connection and a layer norm operation.

$$Z = AddNorm(M_{mem}(X))$$
$$\tilde{X} = AddNorm(F(Z))$$

Full encoder: multiple encoding layers in sequence, the i-th layer uses the output set computed by layer i − 1. higher encoding Layers can exploit and refine relationships already identified by previous layers, N encoding layers → multi level output $\tilde{X} = (\tilde{X}^1 ... \tilde{X}^n)$

2.3 decoder with stacks of attentive layers Conditioned on both previously generated words and region encodings Input: Vector Y and output from all encoding layers $\tilde{X}$, connected through gated cross-attentions Meshed Cross-Attention. Perform a cross-attention with all encoding layers

$C(\cdot, \cdot)$ stands for the encoder-decoder cross-attention

$$M_{mesh}(\tilde{X}, Y) = \sum_{i=1}^{N} \alpha_i C(\tilde{X^i}, Y)$$

C(· , ·) stands for the encoder-decoder cross-attention

$$C(X^i, Y) = Attention(W_q Y, W_k X^i, W_v \tilde{X^i})$$

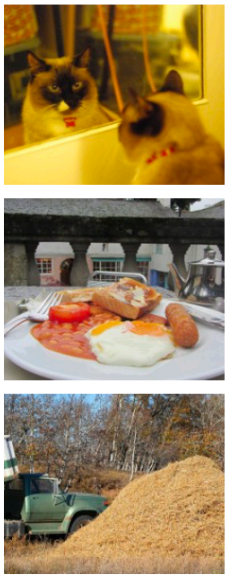
α_i is a matrix of weights same size as the cross-attention results models single contribution of each encoding layer, and the relative importance between different layers.

$$\alpha_i = \sigma(W_i[Y, C(\tilde{X^i}, Y)] + b_i)$$

σ sigmoid activation function Prediction of a word should only depend on previously predicted words Decoder layer comprises a masked self- attention operation Connection between queries derived from the t-th element of its input sequence Y with keys and values Contains a position-wise feed-forward layer as well

$$Z = AddNorm(M_{mesh}(X, AddNorm(S_{mask}(Y)))$$
$$\tilde{Y} = AddNorm(F(Z)),$$

S_{mask}: a masked self-attention over time Input word vectors, and the t-th element of its output sequence make the prediction of a word at time t + 1,



GT: A cat looking at his reflection in the mirror.
Transformer: A cat sitting in a window sill looking out.
$\mathcal{M}^2$ **Transformer:** A cat looking at its reflection in a mirror.

GT: A plate of food including eggs and toast on a table next to a stone railing.
Transformer: A group of food on a plate.
$\mathcal{M}^2$ **Transformer:** A plate of breakfast food with eggs and toast.

GT: A truck parked near a tall pile of hay.
Transformer: A truck is parked in the grass in a field.
$\mathcal{M}^2$ **Transformer:** A green truck parked next to a pile of hay.

FIGURE 3.5: Left, [@mccoco]

conditioned on $Y \leq t$. After taking a linear projection and a softmax operation, this encodes a probability over words in the dictionary.

2.4. Comparison (not sure)detailed ? test on coco or just simple explained

3.conclusion and bridge to next subsection

connections with other subtopics multimodal tasks

–References(not finished)———————— 1.J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in CVPR, 2009. 2.M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zis- serman, "The PASCAL visual object classes (VOC) challenge," IJCV, vol. 88, no. 2, pp. 303–338, Jun. 2010 3.L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in CVPR Workshop of Generative Model Based Vision (WGMBV), 2004 4.G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007 5.N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 20 6. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko- reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, 2017. 7. StevenJRennie,EtienneMarcheret,YoussefMroueh,Jarret Ross, and Vaibhava Goel. Self-

critical sequence training for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017 8..Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Con- ference on Computer Vision and Pattern Recognition, 2018.

Author: Karol Urbańczyk ## Text-2-image

Supervisor: Jann Goschenhofer

- introduce the concept in few sentences
- choice of recent break-throughs is subjective, many important ones not mentioned (GAWWN, LAFITE, Make-a-Scene, probably many others)

Intention of this chapter is to grasp how the field of text-2-image modelling has been changing over the recent years. We will start with basic concepts that has been around since 2014 and end with the state-of-the-art approaches, as of August 2022. Since the field is developing in a rapid pace, with breakthrough models being announced every quarter, we are aware this chapter might soon not be fully covering the field. However, we must notice that cutting edge capabilities of these models tend to come from the scale and software engineering tricks. Therefore, we believe that focusing on the core concepts should make this chapter have a universal character.

3.1.3 Seeking objectivity

- Objectivity in comparing generated images is very hard to grasp
- However, there are some most common datasets and measures that are being used
- This subchapter will quickly present them

3.1.3.1 Datasets

- COCO
- CUB
- Oxford 102

3.1.3.2 Measures

- FID (Frechet Inception Distance)
- IS (Inception Score)
- Human evaluations - photorealism / caption similarity

3.1.4 Generative Adversarial Networks

- quick intro focusing on why it is crucial to start from GANs

3.1.4.1 Vanilla GAN for Image Generation

- intro of GAN

3.1.4.2 Conditioning on Text

- how to encode the text and use it in the generation process
- show some results

3.1.4.3 Stacking generators

- intro of StackGAN, show some results

3.1.4.4 Is attention all you need?

- intro of AttGAN, show some results

3.1.4.5 Variational Autoencoder

- Introducing the concept of VAE
- How is it helpful in generating images

3.1.5 Dall-E starting post-GAN era

- Intro: OpenAI, dataset used, not public, etc
- VQ-VAE and dVAE
- Details how it's working. Combining Transformer with VQ-VAE. Training vs inference
- Results and image examples

3.1.6 GLIDE

- Intro
- Diffusion concept
- details how GLIDE is working
- results / scores
- Limitations / strengths & weaknesses

3.1.7 Dall-E 2

- Intro (mention PR move)
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.1.8 Imagen

- Intro
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.1.9 Parti

- Intro
- details how it is working
- results / scores
- Limitations / strengths & weaknesses

3.1.10 Open-Source Community

- Although most of the recent work comes from OpenAI and Google, there are very interesting directions taken by the open community
- Mentioning the models and quickly what is happening. VQGAN+CLIP, Latent Diffusion models for sure
- Maybe some links for the reader to play with?

3.1.11 Discussion

Mention the following points and why they matter

- potential business use cases
- open vs closed-source (mention dall-e mini)
- copyrights
- biases

3.2 Images supporting language models

*Author: Giacomo Loss

*Supervisor: Matthias Assenmacher

3.2.1 Words in (non-symbolic) contexts

Imagine you were alone in a foreign country, you could not speak the language and the only resource you had were a dictionary in the foreign language. You see a word written on a sign but you cannot understand its meaning. What could you do? One idea would be do open the dictionary and look the word up.

The problem is that the word is defined by using other words in the foreign language. As a second step you would thus look these new words up and continue like that in further steps to the "infinity and beyond" (cit. Buzz Lightyear). But even after looking every single word in the dictionary up, you would still not be able to understand the meaning of the word written on the sign. If on that sign, next to the unknown word, something else were instead depicted, for example an image of a fork and a knife, you might speculate that the word indicates something which has to do with food, like a restaurant. And this without explicitly knowing the meaning of the word. This example is inspired by the work of Stevan Harnad, which formulated at the beginning of the 90's the so called *Symbol Grounding Problem* (**?**). It asserts that it is not possible to understand the meaning (semantics) of a word by just looking at other words because words are essentially meaningless symbols. It is possible to understand the meaning only if the word is put in a context, a perceptual space, other than that of written language: the word must be *grounded* in non-symbolic representations, like images, for example. Over the past 10 years there has been a whopping development of distributional semantic models (DSMs, henceforth), especially after the Word2vec (**?**) revolution. These family of models assume that the meaning of words and sentences can be inferred by the "distribution" of those words and sentences within a text corpus (the *Distributional Hypothesis* formulated by **?**). But the *Symbol Grounding Problem* mentioned earlier suggests that DSMs do not resemble the way words are learned by humans, which is in multimodal perceptual contexts. For these reasons, models have been developed with the goal to integrate further modalities (like visual ones) in pure language models, assuming that grounding words and sentences in other perceptual contexts should lead to a better understanding of their semantics and as a result, to better performance in pure language tasks.

The focus of this chapter are models which empower pure language models with visual modalities in form of images. In particular, models which focus on the semantics of single words or of sentences as a whole are taken into consideration. These models are then tested on well-established pure natural languages tasks and their related benchmarks (see Chapter XX for further references on benchmarks).

Typical tasks for the (intrinsic) evaluation of models at the word-level are:

- Relatedness: "apple" is related to "food"
- Visual similarity: "donkeys" look like "horses"
- Semantic similarity: lemons are similar to oranges (they are both food, acid and used to produce juices)

Tasks related to the evaluation of sentences are among others (based on classification presented in **?**):

- Single sentence tasks: sentiment analysis, for example

- Similarity and paraphrase: test if a pair of sentences are semantically equivalent
- Inference/textual entailment: recognize if sentence B follows from sentence A. For example the sentence "Peter just graduated from high-school" and "Peter can study at university" are entailed since high-school graduation is prerequisite for university enrollment

The chapter describes the evolution of the integration of images as visual modalities into pure language models: from simple concatenation of textual and visual modalities, to the projection of visual elements in a common grounded space and more recently the use of transformers to distill visual information for pure language models. It is no surprise that the applications of this family of models is not confined solely on word and sentence evaluation tasks and include also tasks such as machine translation and dialogue generation; although not the main focus of this chapter, they will be briefly addressed at the end.

3.2.2 Adam and Eve: sequential multimodal embedding

How can an image represents semantic information of a word? The only language machines can understand are numbers. This is why, no matter if we are dealing with textual or image representations of words, a numerical encoding is needed. This comes usually in the form of vector embeddings. Once numerical representations of text and related images are available, the question is how to fuse them and obtain a multimodal representation of a certain word (or sentence). One intuitive idea would be to *concatenate* the textual and visual modalities. Let V_{text} be the textual (vectorial) representation of a word and let V_{img} be its visual (vectorial) representation, a fused representation of a certain word F might take the following simplified form:

$$F = \gamma(V_{text}) \bigoplus (1-\gamma)V_{img}$$

where γ is a tuning parameter which controls the relative contribution of both modalities to the final fused representation. **?** propose a model where a the meaning of a target word is represented as a semantic vector and all vectors are collected in a "text-based semantic matrix". Embeddings are computed based on (transformed) co-occurrence counts of words in a predefined window. The starting point for the construction of the "image-based semantic matrix" is a dataset of labeled images. Firstly low-level features called "local descriptors", which incorporate geometric information of specific areas of a certain picture are extracted and then this descriptors are assigned to cluster of "visual words" (see for example **?** for more details on this technique, called "beg-of-visual-words"). After that, co-occurrence counts of each word label are obtained by summing up visual words occurrences across all images and word labels. The two matrices are then combined and and singular value decomposition is used to project textual and visual inputs on a lower dimensionality space and find multimodal latent factors (whose number is a hyperparameter).

In the end, a similarity of words estimation is performed by using cosine similarity[1]. In this first (historically motivated) example, the vector representation of images is obtained with non-trivial features engineering. But in recent years, the use of neural networks has made an "automatic features selection" possible. This is what for example **?** are doing, extracting visual features from the first seven layers of a convolutional neural network (proposed by **?**) trained on 1.6 million images from the ImageNet database (**?**), which produces scores for 1,512 object categories. The linguistic part of the model relies on the Skip-gram model by **?** and consists of 100-dimensional vector representations. The multimodal representation is again obtained by concatenation of both modalities. Another notable example of concatenation/sequential combination of textual and visual modalities is the work of **?**: textual and visual modalities are represented by separate vectors of textual and visual attributes. During training, these textual and visual inputs vectors are separately fed to denoising (unimodal) autoencoders, whose training objective is the reconstruction of a certain corrupted input - e.g. through masking noise - from a latent representation. Their outputs are then jointly fed to a bimodal autoencoder to be mapped to a multimodal space, on which a softmax layer (classification layer) is added, which then allows the architecture to be fine-tuned for different tasks. The loss function is as follows:

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(\delta_r L_r(x^i, \hat{x}^i) + \delta_c L_c(t^i, \hat{t}^i) + \lambda R\right)$$

where x^i is an input vector and $\hat{x}^i$ its reconstruction, t^i is the correct object label associated with input vector x^i and $\hat{t}^i$ the predicted label, L_r and L_c are entropy loss functions, δ_r and δ_c controls the partial objective functions (reconstruction and classification error respectively) and R is a regularization term. Depending on how the hyperparameters are set, the model can be reduced for example to a simple object classification algorithm (setting δ_r to zero[2]).

3.2.3 The grounded space and the power of imagination

The aforementioned models assume implicitly a one-to-one correspondence between text and images: a visual representation is extracted only from words which are associated to a concrete image. This is a limitation, for two partially overlapping reason. One one hand, how can we depict words for which no image is available in our training set? Is it possible to *imagine* visual representations purely from linguistic ones? On the other hand, could we hypothetically find a visual representation for each word? This might be true for concrete words but

[1] Cosine similarity between vectors a and b is defined as $cos(a, b) = \frac{a \cdot b}{||a|| \cdot ||b||}$

[2] For sake of completeness, in order to obtain a plain object classification model, another hyperparameter, namely v, the corruption parameter for the textual modality should be set to one.