Leipzig University
Institute of Computer Science
Degree Programme Data Science, M.Sc.

# Extracting Large-Scale Multimodal Datasets From Web Archives

## Master's Thesis

Thilo Brummerloh

1. Referee: Prof. Dr. Martin Potthast
2. Referee: Dr. Harrisen Scells

Submission date: September 23, 2024

# Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, September 23, 2024

.....
Thilo Brummerloh

**Abstract**

Large-scale datasets have become a necessity to train machine learning models. Generative text-to-image models require these datasets to be able to depict as many images from the large text space that can be input. Finding enough images with corresponding text that is descriptive of the image is a challenge. This thesis modifies an existing method used by popular text-image datasets, such as LAION, to generate a more comprehensive dataset by going beyond reliance on alt tags as image descriptions. The proposed approach involves extracting image-text pairs from web archives by leveraging span prediction models, allowing for a richer description of images from surrounding HTML content. The extracted dataset is used to fine-tune a pre-trained Stable Diffusion model, and its performance is evaluated through both manual and automated comparisons with the native model. Overall, this work aims to contribute to advancing the quality and quantity of text-image pairs for training next-generation text-to-image models.

# Contents

# Chapter 1

# Introduction

The rapid expansion of digital technology has resulted in an overwhelming increase in multimedia content available on the web. This diverse array of data, encompassing text, images, videos, and other forms of media, offers enormous potential for research and various applications in fields such as natural language processing (NLP), computer vision, and data analysis. However, harnessing the value of multimodal data from web archives presents significant challenges. This thesis aims to address these challenges by proposing effective methods for extracting and utilizing multimodal datasets efficiently, especially focusing on text and image pairs from web archives.

**Motivation** Large-scale multimodal datasets have become essential for training advanced machine learning models, particularly generative models. Web archives, such as the Common Crawl, the Internet Archive, and various national digital libraries, are treasure troves of historical and contemporary data that provide a rich resource for training models. However, the vast volume and heterogeneity of the data pose substantial difficulties in terms of extraction, organization, and usability. Without robust methods for extracting this data, much of its potential remains untapped, hindering progress in generative modeling, Artificial Intelligence (AI), and other areas. The motivation behind this research is to unlock this potential by developing scalable solutions for retrieving, organizing, and curating multimodal datasets from web archives.

**Generative Models** Generative models, such as Generative Adversarial Networks (GAN)s and Variational Autoencoders (VAE)s, have revolutionized content creation and understanding in multimedia research [Goodfellow et al., 2014] [Kingma and Welling, 2014]. These models require vast amounts of diverse, high-quality training data to produce accurate and realistic outputs. In particular, models such as Stable Diffusion and Contrastive Language-Image

Pre-Training (CLIP) rely heavily on multimodal datasets to generate contextually meaningful images based on text [Podell et al., 2023][Radford et al., 2021]. Thus, access to large, high-quality multimodal datasets, such as those stored in web archives, is critical for advancing the performance of these models and enabling better AI applications.

**Challenges with Multimodal Datasets** Extracting coherent multimodal datasets from web archives is far from straightforward. These archives contain a wide array of structured and unstructured data, often in varying formats, resolutions, and languages. Identifying meaningful image-text pairs is particularly challenging due to inconsistent metadata, scattered descriptions, and noisy or irrelevant data. Furthermore, the vast size of web archives complicates the extraction process, making it necessary to design efficient and scalable methods for data filtering, validation, and organization.

**Connecting Text and Image Data** Effective integration of text and image data is critical for building meaningful multimodal datasets. Text provides context and semantic information that enhances the interpretation of images, while images offer visual evidence that supports the understanding of the accompanying text. This relationship is especially relevant for tasks like image captioning, visual question answering, and content-based image retrieval. The challenge lies in automatically linking text and image data in web archives to create datasets that are both contextually rich and relevant, supporting a wide range of machine learning applications.

**Proposed Approach** To address these challenges, this thesis proposes a systematic pipeline for extracting and curating large-scale multimodal datasets from web archives. The approach involves several key steps: data collection from web archives, training machine learning models, extracting text-image pairs, and validating the results. The method employs a combination of web scraping, Natural Language Processing (NLP), and image processing to automate the extraction and ensure high-quality datasets. By automating these processes, the pipeline is designed to handle large volumes of data, making the method scalable and adaptable to different datasets and web archives.

**Web Archives as a Data Source** Web archives, such as the Common Crawl (CC), Internet Archive, and national digital libraries, preserve digital snapshots of web pages over time, capturing the evolution of the internet. These archives provide invaluable historical data for research, offering a longitudinal view of digital trends, online media, and social discourse. However,

extracting relevant multimodal data from these archives is a complex task that requires specialized tools and techniques to efficiently access, retrieve, and process data in diverse formats.

**Methodology Overview** The proposed method for extracting multimodal datasets from web archives follows a multi-phase process. Initially, raw data is collected from CC snapshots using lightweight methods optimized for speed and efficiency. The collected data is then preprocessed to standardize formats and remove irrelevant content. Advanced machine learning models, such as BERT, are used to extract relevant text data, while image processing techniques are employed to extract images. Finally, the extracted text-image pairs are validated using a combination of automated and manual checks, ensuring the dataset's quality and suitability for training models like Stable Diffusion.

**Overview of the Thesis** This thesis is organized as follows: Chapter 1 introduces the problem and its significance, providing an overview of the research objectives. Chapter 2 delves into the technical background, including a discussion of transformers, BERT, and their relevance to multimodal dataset extraction. Chapter 3 reviews related work on generative models, dataset extraction techniques, and the current state of the art. Chapter 4 presents the datasets and models used in this thesis, with an emphasis on data preparation and preprocessing techniques. Chapter 5 describes the implementation of the proposed pipeline, detailing the training, extraction, and validation processes. Finally, Chapter 6 presents the results, evaluating the effectiveness of the proposed method, and offers directions for future research.

# Chapter 2

# Background

## 2.1 Transformers and their Impact on Natural Language Processing

**Transformer** The Transformer model, introduced by Vaswani et al. [2017], marked a paradigm shift in NLP and sequence modeling. Unlike traditional Recurrent Neural Networkss (RNNs) and their variants like Long Short-Term Memory (LSTM) networks, Transformers eschew recurrence in favor of self-attention mechanisms. This architecture allows for the parallelization of computations, which significantly accelerates training and inference times.

At the core of Transformer architecture is the self-attention mechanism, which enables the model to weigh the importance of different words in a sentence relative to each other. This is accomplished through the calculation of attention scores, which are then used to create a weighted sum of the input representations. This mechanism is repeated in multiple layers, allowing the model to capture complex dependencies and contextual information.

The Transformer architecture consists of an encoder and a decoder, each composed of multiple layers of self-attention and feed-forward neural networks. The encoder processes the input sequence, creating a set of contextualized embeddings. The decoder generates the output sequence, one element at a time, using the embeddings produced by the encoder. This design is particularly effective for tasks such as translation, where the model must understand and generate sequences of text.

**Bidirectional Encoder Representations from Transformers (BERT)** Building upon the Transformer architecture, BERT was introduced by Devlin et al. [2018]. BERT's key innovation is its bidirectional approach to pre-training, which allows it to consider the context from both the left and right

sides of a word simultaneously. This is in contrast to traditional unidirectional models, which only process text in one direction.

As shown in Figure 2.1, BERT's architecture consists of multiple layers of transformer encoders that operate over a sequence of input tokens. The tokens are first passed through an embedding layer (denoted as $E_1, E_2, \ldots E_N$), and then processed by the transformer layers (labeled as $T_1, T_2, \ldots T_N$) where each layer applies self-attention and feed-forward networks. The outputs of the final transformer layers are contextually-rich token representations that can be used for a variety of NLP tasks.

BERT is pre-trained on a large corpus of text using two objectives: masked language modeling (MLM) and next sentence prediction (NSP). In MLM, random words in a sentence are masked, and the model is trained to predict these masked words based on the surrounding context. NSP, on the other hand, trains the model to predict whether two given sentences are consecutive in the original text. These pre-training tasks enable BERT to learn rich representations of text that capture syntactic and semantic nuances.

Once pre-trained, BERT and BERT-like architectures can be fine-tuned on specific downstream tasks, such as text classification, named entity recognition, or question answering. This fine-tuning process involves training the model on task-specific data with minimal adjustments to the model architecture. The versatility and effectiveness of BERT have made it a cornerstone in modern NLP applications.

Further work by Sanh et al. [2019] leveraged this architecture to produce a relatively smaller model, DistilBERT, that retained most of the language understanding capabilities with fewer parameters.

SpanBERT, introduced by Joshi et al. [2019], represents a significant advancement over BERT for tasks involving span prediction, such as question answering and coreference resolution [Joshi et al., 2019]. Unlike BERT, which masks individual tokens during pre-training, SpanBERT masks contiguous spans of text and trains the model to predict the entire span from the context of its boundary tokens. This span-based pre-training improves the model's ability to capture relationships between spans, yielding superior performance on extractive tasks. The introduction of the span boundary objective (SBO) allows the model to focus on the span-level context, which is critical for tasks like question answering where understanding the connection between entities is essential. SpanBERT has been shown to outperform BERT on benchmarks such as SQuAD 1.1, achieving state-of-the-art results in span-based tasks.

**Transfer Learning** Transfer learning is a machine learning paradigm where a model trained on one task is repurposed for another related task. This approach leverages the knowledge gained during the initial training phase,

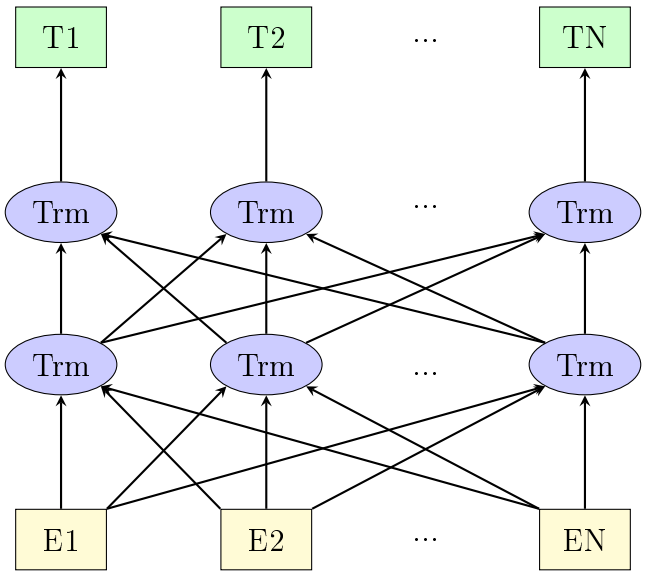



**Figure 2.1:** Simplified BERT Architecture, E for Embedding, Trm for Transformer Nodes, T for Output Transformer Nodes [Devlin et al., 2018]

allowing for more efficient learning and improved performance on the target task [Yosinski et al., 2014]. In the context of NLP, transfer learning has proven to be particularly powerful, as pre-trained models can be adapted to a wide range of language-related tasks [Howard and Ruder, 2018].

Transfer learning typically involves two main steps: pre-training and fine-tuning. During pre-training, the model is exposed to a large and diverse corpus of text, enabling it to learn general language patterns and representations. This stage is computationally intensive and requires significant resources. However, once pre-trained, the model can be fine-tuned on smaller, task-specific datasets. Fine-tuning involves adjusting the pre-trained model weights to optimize performance on the target task, often requiring far less data and computational power than training from scratch.

The success of BERT exemplifies the effectiveness of transfer learning in NLP. By leveraging a pre-trained language model, researchers and practitioners can achieve state-of-the-art results with minimal task-specific training. This approach has democratized access to advanced NLP capabilities, enabling applications across various domains, from sentiment analysis to automated summarization [Devlin et al., 2018].

**Interconnections and Implications for Multimodal Data Extraction**
The synergy between transformers, BERT, and transfer learning has profound implications for extracting large-scale multimodal datasets from web archives.

Web archives, which encompass vast amounts of textual, visual, and auditory data, present unique challenges and opportunities for data extraction and analysis.

Transformers' ability to handle sequential data efficiently and capture long-range dependencies makes them ideal for processing the complex and heterogeneous data found in web archives. BERT's bidirectional contextual understanding enhances the accuracy of text-based tasks, such as entity extraction and sentiment analysis, which are critical for interpreting the vast textual content in web archives.

Transfer learning, on the other hand, facilitates the adaptation of pre-trained models to specific multimodal tasks. For instance, a model pre-trained on text can be fine-tuned for tasks that require integrating text with images or audio, such as extracting relevant information from web pages or transcribing and analyzing multimedia content. This adaptability is crucial for handling the diverse and unstructured nature of web archives.

The integration of these technologies enables the development of sophisticated tools for automatic data extraction and analysis. For example, a system leveraging them could automatically identify and extract key information from web pages, such as headlines, authors, publication dates, and relevant text snippets. Such a system could also integrate visual data, recognizing and extracting relevant images or transcribing text from images.

## 2.2 Known Problem: Text Segmentation and Reference to Images

A known challenge in the realm of web data extraction is accurately identifying and linking textual content with corresponding images. In many web documents, particularly in HTML formats, images and their related descriptions are often not clearly connected. Image captions, alt-text tags, and descriptive sentences around an image may exist, but they are frequently scattered or ambiguous, making it difficult for automated systems to determine which pieces of text directly refer to a particular image.

This problem becomes even more complex when you consider the variety of formats and layouts in web archives. Some websites may place descriptions far from their images or embed images with minimal textual context. The lack of explicit metadata or structured formats further complicates extraction. This issue is not limited to the direct referencing of images but extends more broadly to text segmentation—understanding which parts of a document are relevant to which visual or multimedia components.

If not addressed, this problem can significantly hinder the process of build-

ing large, high-quality datasets of multimodal data like text-image pairs, which are essential for training modern models like those used for text-to-image generation. The challenge, therefore, lies in designing an automated method that can not only recognize images within an HTML structure but also intelligently extract the surrounding descriptive text referencing it.

## 2.3 Proposed Solution: Building a Pipeline for Text-Image Extraction

A central goal of this thesis is to develop an efficient pipeline for extracting text-image pairs from web archives, focusing on image descriptions that are not solely dependent on 'alt' tags. While alt text is useful, it is often missing or lacks descriptive richness, which limits its utility in constructing robust multimodal datasets. This thesis, therefore, proposes a solution that moves beyond alt-text extraction, leveraging a more generalized approach to identifying image descriptions in web data.

A key consideration throughout the design of this pipeline is the need for it to be lightweight in terms of hardware resources. Most steps of the extraction process, such as text tokenization, segmentation, and matching, are computationally efficient and can be quickly applied to large numbers of potential descriptions. The pipeline is designed to run on modest systems. This makes the approach more accessible and allows for scalable deployment across a wide range of web data.

However, one step of the process does require more computational power: the final stage, where extracted image-description pairs are validated through CLIP-based comparison. Contrastive Language-Image Pre-Training (CLIP) is a powerful model that can evaluate the semantic alignment between text and images. While this comparison ensures the accuracy and relevance of the extracted pairs, it is resource-intensive, particularly when applied on a large scale. Therefore, the CLIP comparison is reserved for the final stage, after the initial lightweight filtering has weeded out irrelevant examples. This strategy minimizes the overall computational burden, ensuring that only high-potential image-description pairs are processed with CLIP.

To address this problem, this thesis proposes a multi-step pipeline aimed at efficiently extracting and matching textual descriptions with their corresponding images in web archives.

**Step 1: Dataset Creation** The first step is to extract a manageable subset of tens of thousands of examples from a vast array of web pages. These examples consist of image and text pairs, where the text directly or indirectly

describes the image. This relatively small dataset will serve as a foundation for training a model that can generalize well to the broader task of text-image extraction.

**Step 2: Fine-Tuning BERT for Image Description** Next, a BERT model is fine-tuned using these examples. The model will be trained specifically to understand the textual patterns that typically describe images in web pages. This fine-tuning step focuses on learning the general building blocks of image descriptions, particularly in the unstructured and often noisy context of HTML documents. By doing so, the model becomes capable of identifying descriptive sentences, captions, or tags that reference images in a more accurate and context-aware manner.

**Step 3: Large-Scale Text-Image Extraction** Once the BERT model has been fine-tuned, it will be deployed across a larger corpus of web archives to extract text-image pairs on a larger scale. This process involves feeding web pages to the model, which will then identify the relevant descriptions and link them to the corresponding images. By leveraging this model, the accuracy and recall of the text-image extraction process should remain high.

**Step 4: Fine-Tuning Stable Diffusion with Extracted Pairs** The final step in this pipeline is to utilize the extracted image-description pairs to fine-tune a generative model like Stable Diffusion. Stable Diffusion is a state-of-the-art model for text-to-image generation, and fine-tuning it with these newly extracted pairs will allow the model to produce higher-quality, more relevant images based on natural language input. The results of this fine-tuning will serve as a benchmark for evaluating the effectiveness of the entire pipeline and the quality of the extracted multimodal dataset.

**Conclusion** In summary, the problem of associating text with images in web archives can be addressed through a carefully designed extraction pipeline. By first fine-tuning a BERT model and then deploying it for large-scale extraction, a robust dataset of image-description pairs can be built. This dataset can subsequently be used to fine-tune advanced generative models like Stable Diffusion, ultimately improving the quality of text-to-image generation tasks. This approach not only offers a solution to the specific problem of text-image pairing but also provides a broader framework for extracting multimodal datasets from unstructured web content.

# Chapter 3

# Related Work

## 3.1 Extracting Information from HTML Documents

Extracting information from HTML documents is a crucial task for building large-scale multimodal datasets from web archives. HTML provides a structured representation of web content, consisting of tags that define text, images, and other elements. To extract and link images with their corresponding descriptions, it is necessary to understand the structure of the document and employ efficient tools for parsing and processing the content. In the context of this thesis, resiliparse and fastwarc were used to handle Web ARChive (WARC) files, which store web pages in a compressed format, and to extract text and images from these HTML documents. The process of associating images with their textual descriptions, however, requires sophisticated methods and has been the subject of research in the field of web data mining.

**Extracting Images and Text from HTML** Images in HTML are typically represented using the `<img>` tag, which includes attributes such as `src`, which provides the image URL, and `alt`, a textual description meant to serve as an alternative for the image. `alt` text is especially useful for creating text-image pairs, as it often offers concise descriptions of images. For example, an HTML snippet might look like:

```
<img src="image.jpg" alt="A scenic mountain view">
```

Here, the `alt` attribute provides a brief description of the image, which can be extracted for use in multimodal datasets. Programmatically, this extraction can be efficiently handled using tools like resiliparse, a fast and memory-efficient parser for HTML documents [Bevendorff et al., 2023]. This tool allows

the extraction of text, images, and metadata from large numbers of HTML files, directly from WARC archives. Combined with fastwarc, it enables processing large-scale web crawls quickly, ensuring the data is usable for downstream tasks like machine learning. fastwarc is particularly optimized for large-scale web archives like Common Crawl, which has been used for similar dataset creation tasks, emphasizing the importance of scalable web content extraction [Bevendorff et al., 2021].

**Linking Images and Text with Regular Expressions** While `alt` text provides a direct link between images and descriptions, additional context is often found in the surrounding text, captions, or other elements near the image. Captions, for example, are frequently enclosed in `<figcaption>` or nearby `<p>` tags. The challenge is to associate images not only with their `alt` descriptions but also with any surrounding text that further elaborates on the content of the image. To achieve this, regular expressions (regex) are often used to split text and link it with images. regex patterns can identify sequences of text that appear before and after `<img>` tags, helping to extract context from surrounding paragraphs. For instance, given the HTML snippet:

```
<p>A stunning image of the Alps at sunset.</p>
<img src="sunset.jpg" alt="Sunset">
<p>The sun dips below the horizon, casting a warm glow.</p>
```

A regex pattern could be used to extract the text surrounding the `<img>` tag and link it with the image. This approach is particularly useful when `alt` text is missing or insufficiently descriptive, as it allows for the capture of additional context. This method is similar to the one proposed in Schuhmann et al. [2022], where Large-scale Artificial Intelligence Open Network 5 Billion Image Data Set (LAION-5B) creators linked images with contextual text using a combination of CLIP filtering and regular expression-based extraction techniques.

**Challenges in Text-Image Extraction** Despite the usefulness of `alt` attributes and contextual text, several challenges arise in the extraction process. First, not all images have `alt` text, and some descriptions may be incomplete or irrelevant. This problem has been noted in large-scale web datasets, where missing or poor-quality metadata can introduce noise. Additionally, dynamic content generated by JavaScript can be difficult to access using traditional HTML parsers like resiliparse. In such cases, tools like Selenium or Puppeteer can be employed to render web pages and extract the dynamically loaded content. JavaScript-heavy websites often generate captions and descriptions on

the client side, making it difficult to extract them without fully rendering the page. This dynamic content issue is exacerbated by modern web practices and has been a recurring challenge in web scraping research [Schuhmann et al., 2021].

Another challenge is the prevalence of noise in web pages, such as advertisements, irrelevant stock images, or decorative elements. These elements can be mistaken for meaningful content during text-image extraction, leading to incorrect pairings. To mitigate these issues, machine learning models can be trained to assess the relevance of text-image pairs by comparing the semantic similarity between text and images. For instance, CLIP-based models have been used to filter out noisy or irrelevant image-text pairs in datasets like LAION-5B [Schuhmann et al., 2022] [Schuhmann et al., 2021]. These models compute the cosine similarity between image and text embeddings, allowing only the more relevant pairs to be included in the final dataset. This approach significantly improves the quality of the dataset, reducing noise and ensuring that only meaningful image-text associations are retained.

**Summary** The extraction and linking of images and text from HTML documents are foundational for constructing large-scale multimodal datasets from web archives. By leveraging tools like resiliparse, fastwarc, and regex, it is possible to efficiently process web data and extract meaningful text-image pairs. However, challenges such as incomplete descriptions, dynamic content, and noisy data must be addressed to ensure high-quality dataset construction. Recent research efforts, including the use of semantic filtering models like CLIP and advancements in visual context modeling, offer promising solutions to these issues. These techniques play a crucial role in projects like LAION-5B, which aim to build massive, open-access datasets for training multimodal machine learning models.

## 3.2 LAION-5B: Dataset Creation and Alt Tag Extraction

The **LAION-5B** dataset is a large-scale open dataset designed for training next-generation image-text models like CLIP, DALL-E, and other multimodal AI systems. A critical component of its creation involved the use of *alt tags* found in HTML documents as a means of pairing textual descriptions with images. This section provides a detailed explanation of the methodology behind finding and extracting image descriptions through alt tags, as well as the subsequent filtering process used to build LAION-5B [Schuhmann et al., 2022].

**Using Alt Tags as Image Descriptions** HTML **alt tags** (alternative text) are attributes assigned to images on web pages that provide a text description of the image when it cannot be displayed or when screen readers are used. These descriptions serve a dual purpose: they improve accessibility for visually impaired users and assist search engines in indexing web content. Since alt text is explicitly linked to images, it was a natural candidate for creating image-text pairs for the LAION-5B dataset.

Alt tags were chosen over other forms of text, like captions or surrounding paragraphs, due to their direct association with specific images. While alt text can be noisy, it is typically concise and relevant to the image, especially in cases where it has been optimized for search engine indexing or accessibility.

**Data Collection from Common Crawl** The LAION-5B dataset was built using data obtained from **Common Crawl**, an organization that regularly crawls and archives web pages across the internet. Common Crawl archives snapshots of web pages, including their HTML structure, which contains both images and text.

To create image-text pairs, the LAION-5B pipeline began by parsing the HTML `<img>` tags found in Common Crawl's web archives. These `<img>` tags typically include an *alt* attribute that holds the description of the image.

For example, a typical HTML image tag looks like this:

```
<img src="image.jpg" alt="A photo of a sunset over the ocean">
```

Here, the `alt` text "A photo of a sunset over the ocean" serves as the textual description of the image located at `image.jpg`. The dataset collection script specifically targeted such `<img>` tags with non-empty *alt* attributes.

The team used metadata files from Common Crawl known as Web Archive Transformation (WAT), which provide structured data on URLs, images, and associated metadata. The WAT files were parsed to identify all occurrences of `<img>` tags that contained an *alt* attribute.

After extracting image-text pairs from the HTML documents, a critical step was to ensure the quality of the data by filtering the extracted content. This involved two major processes:

The extracted *alt* text was subjected to a language detection process. The creators of LAION-5B used the CLD3 (Compact Language Detector 3) algorithm to classify the *alt* text based on its language. This was done to segment the dataset into different linguistic subsets, including English and non-English categories.

Each image-text pair was categorized into three possible outcomes:

- **English**: Text confidently classified as English.

- **Other Language**: Text classified as non-English, covering over 100 languages.
- **No Detected Language**: Text that did not meet the confidence threshold for classification, often containing short labels, product names, or other minimal descriptions.

Approximately 2.32 billion image-text pairs were categorized as English, and 2.26 billion pairs were categorized as non-English, while 1.27 billion pairs had minimal or ambiguous text descriptions that could not be clearly assigned to a specific language.

**CLIP Filtering for Image-Text Alignment** While alt tags provide a direct link between text and images, not all image-text pairs have meaningful semantic connections. Therefore, the dataset creators implemented an additional filtering step using CLIP. CLIP (Contrastive Language–Image Pre-training) is designed to compute embeddings for both images and text, placing them in a shared vector space where semantically related pairs are closer together.

The team calculated the cosine similarity between the text and image embeddings generated by CLIP's ViT-B/32 model. Pairs with a cosine similarity above a certain threshold were retained, while pairs below the threshold were discarded.

The chosen thresholds for filtering were:

- **0.28 for English image-text pairs**: Ensuring a reasonable correlation between the image and its English description.
- **0.26 for non-English pairs**: Accounting for the additional noise in non-English web content.

This filtering process removed about 90% of the raw data, reducing the initial pool from 50 billion potential pairs to approximately 5.85 billion high-quality image-text pairs.

**Distributed Processing and Dataset Assembly** Given the large scale of the dataset, processing was distributed across hundreds of worker nodes. Each node was responsible for parsing the WAT files, downloading the images, and applying the CLIP filtering. To maximize efficiency, Python-based frameworks like `Trio` and `Asks` were used for asynchronous downloading and request handling.

The processed data was stored in PostgreSQL databases, where the image URLs, textual descriptions, and relevant metadata (including image dimensions and CLIP embeddings) were kept for further analysis and curation.

**Post-Processing: Addressing Noise and Bias** The dataset creation team took additional steps to ensure the quality of the data by addressing several issues commonly encountered in web-scraped datasets:

- **Noise Reduction**: Image-text pairs with fewer than five characters in the text or images smaller than 5KB were discarded. Similarly, redundant, corrupt, or malicious content was identified and removed.
- **NSFW and Watermark Filtering**: A classifier was trained to identify Not Safe For Work (NSFW) (Not Safe for Work) images, which comprised around 3% of the dataset. These images were tagged, allowing users to easily filter out explicit content. Additionally, watermark detection was applied to reduce the number of commercial or tampered images in the dataset.

**Ethical Considerations and Bias in Alt Tags** While alt tags provide a convenient and direct means of linking text with images, they are not without limitations. Many alt texts are optimized for search engines rather than human readability, and in some cases, they may not accurately reflect the content of the image. There are also potential biases encoded in alt text due to cultural or linguistic differences, as well as the contexts in which the images are used.

The creators of LAION-5B acknowledged these challenges and highlighted the need for responsible usage of the dataset, particularly when developing models for real-world applications. They emphasized that LAION-5B should be considered a research artifact and recommended its use in academic research rather than deployed systems until further bias analysis and mitigation strategies are developed[Schuhmann et al., 2022] [Schuhmann et al., 2021].

To summarize, the LAION-5B dataset was created by:

- Extracting image-text pairs from Common Crawl's HTML data using `<img>` tags with *alt* attributes.
- Performing language detection and segmentation of the dataset into English and non-English subsets.
- Applying CLIP-based filtering to ensure semantic alignment between images and their textual descriptions.
- Conducting distributed processing and post-processing to reduce noise, address explicit content, and curate the final dataset.

Through this approach, LAION-5B offers the research community a large-scale, high-quality dataset for training and fine-tuning multimodal models. Its reliance on alt tags, while imperfect, provides a scalable and efficient way to collect diverse image-text pairs from across the web.

## 3.3 Training Text-to-Image Models

Text-to-image models such as Stable Diffusion XL (SDXL) represent a significant advance in generative modeling. These models leverage the powerful framework of Diffusion Probabilistic Models (DPMs) to produce high-quality images conditioned on natural language text prompts. This section discusses how these models are trained, the role of training data in the process, and how further improvements may be realized through careful data curation and conditioning techniques.

**Overview of Training Process** Training text-to-image models such as SDXL follows the general paradigm of DPMs, where the model is trained to reverse a gradual noise-adding process applied to data. More specifically, the model learns to denoise an image step by step, starting from pure noise and eventually generating a meaningful image that aligns with the text prompt. This denoising process is controlled by a U-Net architecture, which serves as the backbone for the model's operations. The diffusion model, as described in Ho et al. [2020], involves adding Gaussian noise to images and then learning the reverse of this process.

In the case of SDXL, the model architecture builds on this framework by incorporating a significantly larger U-Net with additional attention blocks and a second text encoder to handle complex multimodal conditioning [Podell et al., 2023]. This allows SDXL to produce high-resolution images with fine-grained details that are closely aligned with the given text input. A notable feature of SDXL is the integration of multiple transformers, each specialized to process different levels of image features and text embeddings [Esser et al., 2024].

**Utilization of Training Data** The quality of the model's output is heavily dependent on the training data, which consists of paired text-image examples. These pairs are derived from large, publicly available datasets such as LAION-5B, which provide extensive collections of images along with associated textual descriptions [Schuhmann et al., 2022]. The text component, often processed through CLIP-based encoders, helps condition the image generation process, ensuring that the output image aligns semantically with the input text [Radford et al., 2021].

During training, the model is exposed to a diverse set of images and captions that vary in resolution, aspect ratio, and complexity. This diversity is crucial for improving the generalization capabilities of the model, allowing it to handle a wide range of prompts during inference. To maintain consistency and prevent the introduction of artifacts, the images are often resized to standard resolutions, although techniques such as *size-conditioning* have been proposed

to allow models to adapt to the original resolution of the input images [Podell et al., 2023].

In addition to resizing, data augmentation techniques such as random cropping and color jittering are applied to ensure that the model learns to generate images that are robust to variations in the input data. However, these augmentations can sometimes introduce undesirable biases into the model, such as cropped or incomplete objects in the generated images. To mitigate these issues, SDXL employs crop-conditioning, where the cropping parameters are passed into the model as additional conditioning signals, ensuring that the model learns to account for these transformations during training. These can also be found in further improvements on the SDXL architecture [Esser et al., 2024].

**Improving Model Performance Through Training Data** The performance of text-to-image models can be significantly improved through better utilization and augmentation of the training data. In the case of SDXL, several innovations were introduced to enhance the model's ability to handle diverse data.

One such improvement is multi-aspect training, which allows the model to be trained on images of various aspect ratios rather than just standard square images (e.g., 512x512 pixels). By dividing the dataset into buckets based on aspect ratios, SDXL can learn to generate images of different formats (e.g., landscape or portrait) without sacrificing quality. This approach not only enhances the model's flexibility but also reduces the need for upscaling artifacts that could degrade image quality[Podell et al., 2023].

Another key improvement is the use of a **refinement model**, which is applied as a post-processing step to improve the fidelity of generated images. The refinement model is a smaller diffusion model trained specifically to enhance local details, such as textures and sharp edges, in the output images. By denoising the latent space generated by the base SDXL model, the refinement stage helps correct minor imperfections, especially in high-resolution images [Podell et al., 2023].

Furthermore, the introduction of **conditioning schemes** based on image size and cropping parameters has proven to be an effective strategy for ensuring that the model adapts to the varying scales and compositions found in real-world datasets. By embedding these additional conditioning signals into the model's training process, SDXL can generate more coherent and contextually accurate images [Esser et al., 2024]. These improvements show that the careful handling of training data—whether through selective augmentation, conditioning, or post-processing—can lead to substantial gains in the model's overall performance.

**Future Developments** While SDXL represents a significant leap in text-to-image generation, there are still areas where further improvements can be made, particularly with regard to the training data. One potential avenue for improvement is the curation of more diverse and balanced datasets. Ensuring that the training data covers a wider range of subjects, styles, and languages can help reduce biases in the model's output and improve its ability to handle less common prompts [Ho and Salimans, 2022].

Additionally, increasing the size and diversity of textual descriptions and moving beyond short captions to more detailed narratives, could enhance the model's understanding of complex scenes and improve prompt adherence. As suggested by [Esser et al., 2024], incorporating more structured text sources, such as articles or stories, may provide richer semantic information for training and enable the generation of more intricate images.

Finally, the exploration of alternative architectures, such as transformer-based diffusion models, offers possibilities for improving both the speed and accuracy of text-to-image generation. These models could allow for better integration of image and text modalities, as demonstrated by recent research on bidirectional transformer architectures [Esser et al., 2024]. These innovations have the potential to further push the boundaries of what is achievable with generative text-to-image models.

The training of models like SDXL relies on a sophisticated interplay between large-scale datasets, innovative conditioning techniques, and powerful model architectures. By leveraging advances in data augmentation, multi-aspect training, and post-hoc refinement, SDXL has set a new standard for text-to-image synthesis. However, the careful selection and augmentation of training data remain crucial for continued improvements in this field, as models evolve to generate more realistic, diverse, and contextually accurate images.

## 3.4 Quantifying Quality of Text-to-Image Networks

The quality of text-to-image models, such as those used in generative models like Stable Diffusion, is traditionally measured using both objective and subjective metrics. Quantifying quality is essential for comparing model outputs, understanding their behavior, and improving their performance. In this section, we review different approaches for evaluating the quality of text-to-image networks, with a focus on automated evaluation metrics, particularly CLIP conformity and image complexity.

**CLIP Conformity** One of the primary methods for quantifying the quality of a generated image in relation to the input text prompt is through CLIP (Contrastive Language–Image Pretraining) [Radford et al., 2021]. CLIP is a neural network that can map both text and images to a shared embedding space, allowing for the computation of semantic similarity between an image and a text prompt. This similarity, or *conformity*, provides an automated way to evaluate how well the generated image aligns with the input text.

In recent work, CLIP conformity has emerged as a powerful tool for assessing text-to-image models [Podell et al., 2023]. The method involves calculating the cosine similarity between the text and image embeddings, both generated by CLIP. A higher cosine similarity indicates a stronger semantic alignment between the input prompt and the generated image, which correlates with the model's ability to accurately interpret and render the input text.

The advantage of using CLIP for conformity is its ability to generalize across diverse text and image modalities, making it an ideal metric for evaluating models trained on large-scale multimodal datasets [Ramesh et al., 2021]. Moreover, because CLIP has been trained on a wide variety of internet-sourced image-text pairs, it is less likely to be biased towards specific content, providing a fair assessment of alignment between text and image.

**Quantifying Complexity of Generated Images** Beyond semantic alignment, another critical metric for evaluating the quality of generated images is their *complexity*. Complexity refers to how detailed and intricate an image is, often measured by the number of distinct features, textures, or objects present in the generated image. Models like SDXL are expected to generate images with high complexity, particularly when conditioned on prompts that contain detailed descriptions.

Quantifying image complexity can be done in several ways. One approach is to use edge detection algorithms, such as Canny Edge Detection, to quantify the number of edges or boundaries within an image. The more edges detected, the more complex the image is likely to be [Esser et al., 2024]. Another method is to calculate the entropy of the image. Entropy measures the amount of information contained in an image; higher entropy typically corresponds to a more detailed and complex image. This method is particularly useful when comparing models that generate images with varying levels of detail [Ho and Salimans, 2022]. Additionally, object detection models like YOLO (You Only Look Once) can be used to detect the number of distinct objects in an image. A higher count of objects or elements suggests that the model has generated an image with more intricate details [Podell et al., 2023].

In the analysis, a combination of edge detection and object detection to measure the complexity of images generated by different text-to-image models

was used. The results showed that SDXL consistently outperformed earlier models by generating images with more distinct features and higher object counts, particularly for prompts that described complex scenes or environments. This suggests that the enhanced architecture of SDXL, with its multi-aspect training and refinement techniques, allows for better handling of intricate prompts, resulting in more detailed and contextually rich images [Podell et al., 2023].

**Potential for Improvement** While both CLIP conformity and image complexity offer valuable insights into the performance of text-to-image models, there are still opportunities for improvement. One area that could enhance evaluation metrics is the combination of these approaches with more advanced multi-modal metrics. For example, incorporating Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al., 2018] alongside CLIP conformity may provide a more holistic view of both visual fidelity and semantic alignment. Additionally, the curation and augmentation of training data can play a significant role in improving these metrics. By increasing the diversity and richness of the training dataset, models can be better trained to handle both simple and complex prompts, leading to improvements in both CLIP conformity and image complexity [Esser et al., 2024]. The careful handling of edge cases, such as ambiguous or abstract prompts, can also reduce the likelihood of generating low-complexity images that fail to meet the expectations of the input text.

Another important aspect of evaluating text-to-image models involves using perceptual metrics that assess the visual quality of the generated images independent of the text, as well as metrics that gauge the aesthetic and compositional harmony between image and text. One such metric is the Fréchet Inception Distance (FID), which measures the similarity between the distribution of real images and the generated images [Heusel et al., 2017]. FID operates by comparing feature vectors obtained from a pre-trained network (usually InceptionV3) and calculates how closely the generated images resemble real-world images. While FID does not directly assess the alignment with text, it remains one of the most reliable metrics for evaluating the overall visual fidelity of generated images.

Another related metric is the Inception Score (IS), which evaluates the quality and diversity of the images [Salimans et al., 2016]. High IS values indicate that the generated images belong to well-defined classes, making it particularly useful when assessing the diversity of a model's outputs. Beyond these metrics, aesthetic scoring has also emerged as a practical measure for image quality, especially in models like SDXL, where visual beauty and adherence to artistic styles are often key considerations. Aesthetic evaluators, which use neural networks trained on human-annotated image rankings, can

provide a score based on how visually appealing an image is perceived to be [Schuhmann et al., 2022].

Combining these methods, FID, IS, and aesthetic scoring, allows for a more comprehensive assessment of image quality, capturing both technical aspects (such as sharpness and clarity) and subjective factors (such as beauty and harmony). These additional metrics complement CLIP conformity and complexity measurements, offering a holistic evaluation of the generated images in terms of both their standalone quality and their correspondence to text prompts.

In summary, evaluating the quality of text-to-image networks requires a multi-faceted approach. CLIP conformity provides a robust measure of semantic alignment between input prompts and generated images, while metrics such as image complexity allow for the assessment of the level of detail present in the output. By combining these methods, as well as exploring potential improvements through data augmentation and more sophisticated evaluation techniques, a deeper understanding of the strengths and limitations of modern text-to-image models like SDXL can be gained.

# Chapter 4

# Datasets and Models

## 4.1 Common Crawl Web Archive

### 4.1.1 The Common Crawl Web Archive: An Overview

The Common Crawl web archive is a widely used and comprehensive dataset that captures snapshots of publicly accessible web pages. As an open-access repository, it has been used for various research purposes including natural language processing, information retrieval, and machine learning. The dataset is constructed through monthly crawls of the web, focusing on diverse domains and web pages in multiple languages. Founded in 2008, Common Crawl continues to build its repository by systematically traversing the web, downloading webpages, images, and other embedded content such as videos and metadata.

The archive is organized into Web ARChive (WARC) files, which are standardized for storing web page data in a structured format. Each WARC file includes raw HTML content, images, and metadata like HTTP headers, timestamps, and URLs. This multimodal dataset, consisting of text, images, and metadata, is a valuable resource for research that involves large-scale web data analysis and dataset creation for machine learning [Common Crawl, 2024].

**How It Can Be Accessed** Accessing Common Crawl data is straightforward, facilitated by Amazon Web Services (AWS) where WARC files are hosted. Users can directly download the WARC files using tools such as `wget` or `curl`. Common Crawl provides an index for each monthly crawl, allowing users to navigate and identify relevant WARC files for download.

For more efficient access, Common Crawl is part of the AWS Open Data program, enabling users to query the archive via Amazon Athena without needing to download the full dataset. Users can also interact with the WARC files programmatically using libraries like fastwarc in Python, which allows

the opening, parsing, and filtering of WARC records based on specific criteria, such as metadata fields or content type.

**Contents of WARC Files** Each WARC file encapsulates a variety of web resources. The most common types of WARC records include:

- WARC-Info: This record contains metadata about the WARC file itself, such as crawl date and parameters.
- WARC-Request: Stores HTTP request information sent by the crawler, including the URL and HTTP method.
- WARC-Response: Contains the actual content of the web resource (e.g., HTML pages, images) along with HTTP headers.
- WARC-Metadata: Stores additional metadata about web resources.
- WARC-Conversion: Preserves both original and transformed versions of content, useful when processing or converting data (e.g., text extraction).

The archived resources cover a range of content, including HTML pages, embedded images, multimedia, and rich metadata. This multimodal structure allows for a variety of analyses and processing tasks, from natural language processing to image recognition.

### 4.1.2 How It Was Used

In this research, Common Crawl's WARC files are utilized to create a large-scale multimodal dataset of text-image pairs. Two Python scripts, formed the core pipeline for extracting data that could be used for training a model to recognize alt tags and further down the line predict another set of texts to extract image descriptions.

**Training Data Extraction** The initial step in the pipeline was implemented using a python script, which processes the WARC files and extracts text that contains alt tags linked to images. This script systematically iterates through WARC records, detecting HTML pages and parsing their content. The following process is applied:

1. HTML Parsing: Using the `resiliparse` library, HTML content is parsed and processed. The script focuses on identifying image tags (‘<img>‘) and extracting their alt attributes.

2. Alt Tag Identification: If an image tag contains a non-empty alt attribute, the alt text is extracted and saved.

3. Text Segmentation: Text before and after the image tag is segmented, ensuring a minimum word count for both segments. These segments provide context around the image, which is essential for downstream tasks in multimodal dataset creation.

4. Language Filtering: Only English-language content is kept, as detected by the `resiliparse` language detection tool.

5. Alt Tag Matching: The script identifies and matches alt tags in the surrounding text, incrementing counters to track the number of alt tags found and linked to surrounding text.

The result of this script is a dataset of approximately 200,000 text segments that contain an alt tag linked to an image. This output serves as the foundation for the next stage of the pipeline, where these text-alt tag pairs are used to train a span detection model.

**Prediction Data Extraction** The second script extracts relevant text surrounding images to prepare the data for prediction by a span prediction model. The main objective of this script is to tokenize the text around the image and prepare it for further prediction and eventual extraction of the images.

1. HTML Parsing and Image Extraction: The script iterates through WARC files and parses the containing HTML content to identify images. It segments the text into two parts: the text before the image and the text after.

2. Tokenization: Using a pre-trained DistilBERT tokenizer, the text is tokenized with a maximum sequence length of 511 tokens. Additionally, a special token '[IMG]' is added to mark the position of the image within the text, helping the model distinguish between regular text and image-linked text.

3. Text-Image Pair Creation: For each image, the surrounding text is tokenized, and the image URL is linked to the tokenized text. This forms the core of the prediction dataset.

4. Counter and Progress Tracking: The script includes mechanisms to track the number of web pages processed, the number of images found, and the number of alt tags linked to surrounding text. This helps monitor progress and ensures that a sufficient number of examples are extracted.

The output of this script consists of JSONL files, where each entry represents a processed text-image pair. This dataset is subsequently used to train a span prediction model that learns to link images to their corresponding descriptions.

This dataset was then fed into a span prediction model, which was fine-tuned using the extracted examples to predict text spans that describe images. By leveraging the rich multimodal content from Common Crawl and the pipeline described above, the dataset serves as a valuable resource for training text-to-image and image-to-text models.

## 4.2 Created Datasets

### 4.2.1 Training Dataset

The training dataset was designed to extract text-image pairs from Common Crawl WARC files. Its construction was guided by the need to collect representative text snippets for images on a large scale, enabling the training of models capable of understanding multimodal data.

**What should the dataset contain?** The training dataset should include diverse text-image-description pairs that capture a wide range of domains, styles, and content. Each pair must provide context-rich text that describes or relates to the accompanying image. The text should be tokenized using the DistilBERT tokenizer to ensure consistency and compatibility with the model pipeline. Images should be linked directly to their source URLs.

Each training dataset entry should consist of:

- **Contextual Text**: At least 200 tokens of text preceding the image, capturing relevant context for the image.
- **Alt Tag**: A special description given by website creators that describes an image.
- **Additional Text**: At least 250 tokens of text succeeding the image, capturing relevant context for the image.
- **Image URL**: A direct link to the image for retrieval and further processing.
- **Source URL**: The URL of the webpage where the image and text were originally found, ensuring traceability.

**What does the dataset contain?** The constructed dataset contains over 200,000 text-image pairs extracted from Common Crawl. Each entry consists of the image URL and the surrounding text, which is preprocessed and tokenized for further use in model training. The text includes spans before and after the image, and care has been taken to preserve the contextual relevance of the image description. All examples in the set include the alt tag in the text which might limit the ability to generalize to images that are described in text but where the description does not conform to the form of alt tags.

Post-extraction cleaning ensured that invalid or irrelevant examples were removed. This process included filtering out images without meaningful descriptive text and discarding broken links. Furthermore examples with NSFW text and duplicates were removed. The final dataset is designed to be used in various machine learning tasks, especially those related to text-image pairing and multimodal learning. The cleaning steps should ensure no train-test-leakage is taking place.

### 4.2.2 Multimodal Dataset

The multimodal dataset is a refined version of the training dataset. It was created by applying CLIP-based filtering to retain only those text-image pairs with a high semantic similarity. The goal was to ensure the high quality of the data, enabling better performance in downstream model fine-tuning and evaluation tasks.

**What should the dataset contain?** The multimodal dataset should contain pairs that exhibit high alignment between the text and the corresponding image. The text should clearly describe the content of the image, and the image must be relevant to the surrounding text. This alignment can be ensured through cosine similarity scoring using the CLIP model. Each entry should include:

- **CLIP Score**: A similarity score between the image embedding and the corresponding text embedding.
- **High-Quality Images**: Images that are semantically aligned with their descriptions and are free from distortion or irrelevant content.
- **Text Descriptions**: Descriptions that are accurate and contextually informative, clearly reflecting the visual content.

**What does the dataset contain?** After the refinement and filtering process, the final multimodal dataset consists of 5,000 text-image pairs. Each pair

was evaluated using the CLIP model, which calculated the cosine similarity between the image and the accompanying text. Only pairs with scores above a predefined threshold were retained, ensuring a high level of semantic alignment and quality.

The text descriptions have been carefully processed to maintain their contextual relevance and accuracy, reflecting the content of the associated images. The dataset includes image metadata such as source URLs, alt-text (where available), and image dimensions, providing comprehensive information for each pair. This multimodal dataset forms the basis for training and evaluating models that require precise alignment between textual descriptions and visual content, and it plays a critical role in tasks such as image generation and semantic understanding.

## 4.3 Pretrained Models

In this section, the application of three pretrained models, BERT, CLIP, and Stable Diffusion XL, are presented. These models, having been previously detailed, were central to extracting, refining, and generating multimodal data.

### 4.3.1 BERT

The DistilBERT model was fine-tuned using a span prediction approach. The standard tokenizer was modified to incorporate a special token, [IMG], to mark the position of images within the extracted text. The fine-tuning process involved training on 100000 examples, with the goal of identifying the start and end spans of descriptions around the [IMG] token. As shown in the training method, IoU loss was employed to improve performance on span predictions. This step was crucial to predicting the text that describes images from web archives. The trained model was then used in the next step to predict text spans for subsequent processing. An evaluation of the performance of this model is done in Chapter 6.1.

### 4.3.2 CLIP

To ensure the extracted descriptions were appropriate for the corresponding images, the CLIP model was used, which computes cosine similarity between the image embeddings and the extracted descriptions. For each image-description pair, a CLIP score was calculated, allowing to filter out low-scoring pairs and keep only those that exhibited high semantic similarity. This refinement was essential for improving the quality of the dataset, ensuring that the text-image pairs were suitable for training further models.

### 4.3.3 Stable Diffusion XL

The final stage of the process involved fine-tuning Stable Diffusion XL using the filtered image-description pairs. Exactly 5000 images, paired with their predicted descriptions, were used for this fine-tuning. The hypothesis was that, since the descriptions were more accurate and detailed than simple alt-text, the fine-tuned model would generate images with greater fidelity to complex prompts. This step demonstrated that the pretrained Stable Diffusion model could be further adapted to produce more detailed images when exposed to better descriptive data. Results of this fine-tuning can be found in Chapters 6.2 and 6.3.

# Chapter 5

# Methods

The following sections will describe in detail the process of creating training data, training a model and using that model to extract text-image-pairs. Lastly there is a description of how those pairs can be used to fine-tune an existing text2image network.

## 5.1 Overview of the Pipeline

For a better this section gives an overview of the pipeline. This is to give a better overview of the work, but also determines the structure of the implementation.

### 5.1.1 Training Data Extraction

The first step in the pipeline mainly consists of handling html documents from WARC files. The proposed method uses the `fastwarc` library to access the content of these files. Each WARC file contains multiple records, each corresponding to a web page or resource. The `resiliparse` library is used to process the HTML and extract `<img>`-tags, which represent the images embedded in the webpage. Each `<img>`-tag contains attributes such as the `src` (source of the image) and `alt` (alternative text description of the image) which on some web pages are displayed to users if the image fails to load. Since the general idea consists of extracting image descriptions from the text surrounding images, websites are split along `<img>`-tags with the text before and after the image is linked to that image. In a first approach to build a training dataset, a human annotator was supposed to review a small number of examples. Due to problems that are discussed in a later section, this was dismissed in favor of an automatic approach.

### 5.1.2 Training Span Detection Model

Once the HTML content is extracted into a training dataset, the next step is to build a model that can learn from that and generalize to a larger number of examples. In this naive approach, .

The `src` attribute is critical for identifying the image's URL, while the `alt` attribute often provides valuable textual descriptions of the image. This basic information is gathered from the HTML DOM tree and stored as part of the extracted dataset. The use of the `alt` attribute is particularly useful for cases where the image does not have an explicit relationship with surrounding text, as it provides direct linguistic context for the image.

### 5.1.3 Fine Tuning Stable Diffusion

After identifying the text segments and associated images, the pipeline proceeds to download and store the image data. The URLs extracted from the `src` attributes of the `<img>` tags are used to download the images, and these are stored in a local directory. The corresponding text is then saved alongside the images in a JSONL (JSON Lines) format, where each entry includes the image's file path, the associated text before and after the image, and the `alt` tag if available.

The JSONL format was chosen for its simplicity and ease of deployment. Each line in the file represents a single image-text pair, making it easy to append new entries as more records are processed. This format also allows for easy parallelization of processing steps, enabling the pipeline quickly populate it with image data.

In Figure 5.1 an overview of the pipeline can be gained. Here, the four steps Training Data Extraction, Model Training, Data Extraction and Fine Tuning can be seen with their dependencies on previous steps and the data resources that are taken from CC and the internet or depend on the execution of previous steps.

## 5.2 Training Data Creation

To find and extract a large number of suitable examples, an automated approach to creating a training dataset creation was chosen.

**Manual Dataset Creation** A first attempt to manually build a set of examples was dismissed due to the low number of good matches between images and a corresponding text describing that image. To find matches a Doccano project was created. Doccano can be used as a platform to host manual annotation of datasets. Annotators could access a web page where pairs of images

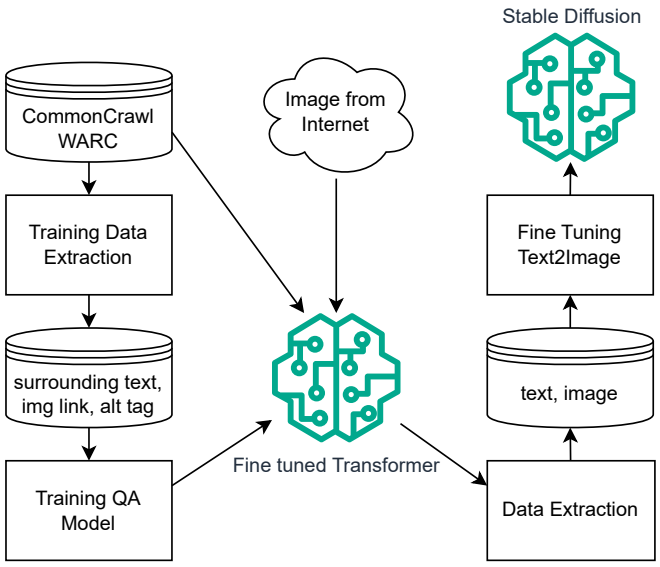



**Figure 5.1:** Overview of the data sources, steps in the pipeline as well as the resulting datasets.

and the text that surrounds that image on a webpage were shown. Due to most images not being referenced in their surrounding text, this attempt proved to be too time consuming and ultimately failed. In 400 examples only 2 were found to be suitable to be used as training examples.

**Automated Dataset Creation** To find a larger number of examples, a simple method for finding text that describes an image was necessary. An approach similar to the one described by LAION in their approach to building the LAION-5B dataset was chosen. Any image that had an alt tag, and whose alt tag was also present in the webpage's text was included in the training dataset with some filtering applied. The filter excludes alt tags that are not English, too short, or too long, as well as those that could be considered NSFW.

**Data Source** The web archive was used as input for this. Multiple WARC-files are read and the containing web pages split along images, in a way that produced a string that contains the preceding 2000 character as well as the succeeding 2500 characters. An Overview of the number of extracted examples can be seen in Figure 5.2. Only about 6.5% of surveyed webpages did not have an image embedded. Although this does not mean that any of them could be

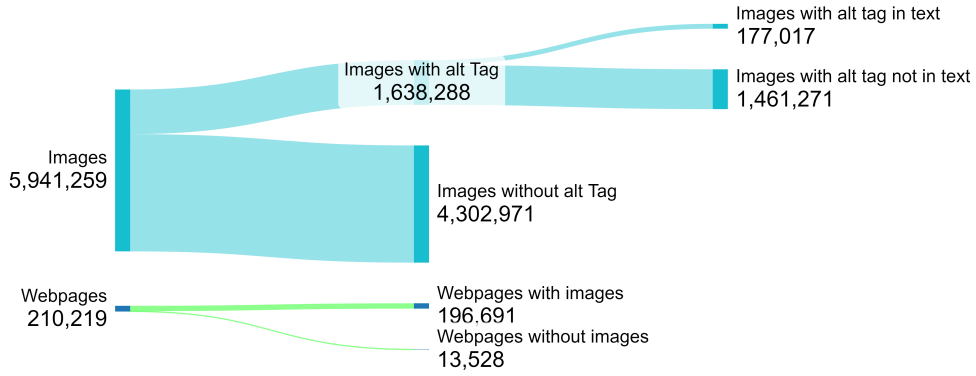



**Figure 5.2:** Overview of the number of webpages surveyed and the images and alt tags found.

useful for either the training data or as a potential image for the multimodal dataset. In a later step a filter is applied to only allow images of certain sizes. In the same figure it also becomes apparent that only 2.98% of images contain an alt tag that can be found in the immediate surrounding text, which is a requirement for this first step of creating a training dataset.

Specific care has been taken to ensure no train test leakage is taking place. For this examples had to be deduplicated so no single example was too similar to another. Additionally web pages, meaning the second-level domain name, were allowed either in the training or test set of examples to ensure neither step could unduly influence the other by leaking information.

## 5.3 Model Architecture and Training

**Tokenization and Model Training for Span Prediction** This step of the pipeline uses a pre-trained version of DistilBERT by Sanh et al. [2019] for extracting meaningful representations from text around images. This step involves several key processes, including text normalization, tokenization, and model training. The objective of this component is to predict the span of text corresponding to the image's description by identifying the start and end positions within the tokenized text.

**Text Normalization** Text extracted from web archives often contains inconsistencies, such as irregular punctuation and case-sensitive variations. To ensure consistency in the data, a text normalization function is applied. This function converts all characters to lowercase and removes any punctuation characters. Additionally, common articles such as "a," "an," and "the" are

removed to reduce noise. This preprocessing step ensures that the text is prepared for tokenization in a format compatible with BERT-based models.

**Tokenization** Once the text is normalized, the next step involves tokenizing the text into subword units. The DistilBERT tokenizer is used for this purpose, which splits the input text into tokens that can be processed by the model. A special token, denoted as `[IMG]`, is inserted into the text to mark the position of the image. This token allows the model to learn the relationship between text segments surrounding the image.

The tokenizer applies subword tokenization with a fixed maximum length of 512 tokens, as DistilBERT models require a constant input size. To avoid truncation of important content, the pipeline extracts a specified number of tokens before and after the image's location, with the `[IMG]` token indicating where the image appears. The tokenized representation of the text is then passed to the model along with attention masks, which help the model focus on relevant parts of the input. This is necessary in case the input had to padded due to a length smaller than 512.

**Model Structure** The model is designed to predict the start and end positions of the text span corresponding to the image's description. It uses the DistilBERT encoder as the backbone, which provides contextualized embeddings for each token in the input sequence. These embeddings are then passed through two dense layers to predict the start and end positions. Each of these layers outputs logits, which are subsequently converted into probabilities using a softmax activation function.

**Loss Functions** Depending on the configuration, the model is trained using either a soft Intersection-over-Union (soft-IoU) loss function or a sparse categorical cross-entropy loss function. The soft-IoU loss is designed specifically for the span prediction task, where the model must predict both the start and end positions. This loss function is applied separately to each predicted span, ensuring that the predicted span closely matches the ground truth span [Huang et al., 2020].

If cross-entropy is used, the model computes the loss for both the start and end position predictions independently, treating them as classification tasks. In this case, the true start and end positions are passed as separate labels to the model, and the loss is computed based on the accuracy of the predicted token indices.

**Metrics and Callbacks** To evaluate the performance of the model during training, several metrics are tracked. The exact match score calculates the

proportion of examples for which both the start and end positions are predicted correctly. Additionally, the Intersection-over-Union (IoU) is computed as a measure of overlap between the predicted span and the true span, providing a more nuanced evaluation metric. The IoU metric is logged along with the exact match score, and the model checkpoints are saved based on the best IoU score during training. These metrics are critical for ensuring that the model is effectively learning to localize the relevant text spans for image descriptions.

## 5.4 Text and Image Pairing Pipeline for Data Extraction

In this stage of the pipeline, the pre-trained and fine-tuned BERT model from the previous step is employed to extract image descriptions from text passages surrounding image references in Hypertext Markup Language (HTML) data. The process begins by passing tokenized sequences of text into the fine-tuned BERT model to predict the span of text that is most likely to describe the associated image. This model is trained to identify relevant descriptions by examining the context before and after the image reference.

**Input for Prediction** To generate the input for the model, the sequence consists of tokens from the text preceding the image reference, followed by a special token representing the image, the tokenized `alt` tag, and tokens from the text following the image. The maximum sequence length is limited to 512 tokens to ensure compatibility with the BERT model. The model produces logits for the start and end positions of the predicted span, from which the most probable indices are selected using the softmax function. These indices correspond to the predicted start and end positions of the image description within the sequence. Confidence scores for the span prediction are derived from the softmax probabilities for both the start and end positions, which are used to evaluate the reliability of the description.

**Downloading Images** Once the description is extracted, the corresponding image is downloaded. The system validates the image by checking that its size exceeds 5KB and that its resolution meets the minimum requirement of 224x224 pixels. If the image does not meet these criteria, it is discarded to ensure that only high-quality images are processed further.

After successfully downloading a valid image, the image and its corresponding text description are enriched with similarity metrics derived from the CLIP model. The CLIP model, which is pre-trained on large-scale multimodal datasets, calculates a cosine similarity score between the image and the

extracted description. Both the image and the text are first processed through the CLIP model to generate embeddings, which are then normalized. The similarity score is computed by calculating the cosine similarity between the normalized embeddings of the image and the text, producing a score ranging from -1 to 1 that quantifies how well the description matches the image. This score, along with the span prediction confidence metrics from BERT, is stored for later analysis.

Through this multi-step process, the pipeline extracts meaningful text-image pairs from large-scale web archives, validating the quality of both the image and the description using metrics from the BERT and CLIP models. This enriched data is critical for subsequent tasks, such as training models for tasks involving image generation or multimodal understanding.

## 5.5 Fine-tuning Stable Diffusion XL

The multimodal dataset, whose extraction was described in the previous section, was used to fine-tune a pre-trained SDXL model. The dataset consists of image-caption pairs, where each image is associated with a descriptive caption, and was leveraged to enhance the model's ability to generate high-fidelity images from textual descriptions.

**Dataset and Pre-processing** The dataset used for fine-tuning was stored in a JSON format, with each entry containing the file path to an image and its corresponding caption. The images were resized to a fixed resolution of 1024x1024 pixels. The captions were tokenized using the CLIP tokenizer from the `transformers` library, ensuring that each tokenized sequence had a uniform maximum length.

For training purposes, the text descriptions were encoded into embeddings using the pre-trained CLIP text encoder [Radford et al., 2021]. A custom PyTorch Dataset class was implemented to manage the pre-processing of images and captions, converting both into the required tensor format suitable for model input.

**Fine-tuning Procedure** The fine-tuning procedure was conducted using the Stable Diffusion Pipeline provided by the `diffusers` library. The pre-trained model was initialized, and the weights were adapted based on the new multimodal dataset. The `StableDiffusionTrainer` class was utilized to streamline the fine-tuning process. This allowed the integration of the multimodal dataset while adjusting various hyperparameters for efficient model convergence.

The training was conducted over fifty epochs with a batch size of 2. The AdamW optimizer was employed, with a learning rate of $5 \times 10^{-6}$, to minimize the loss between the generated images and the original image-text pairs. Additionally, gradient checkpointing was enabled to reduce memory usage during training. Logs were generated using TensorBoard for monitoring performance metrics. After completing the fine-tuning process, the model was saved to disk. This allowed future use of the fine-tuned model for generating images from new textual prompts.

By using the multimodal dataset for fine-tuning, the Stable Diffusion XL model was adapted to generate high-quality images that are closely aligned with the textual descriptions in the dataset. The fine-tuned model is expected to exhibit enhanced performance when generating images from new, unseen prompts, benefiting from the specialized training on the collected dataset.

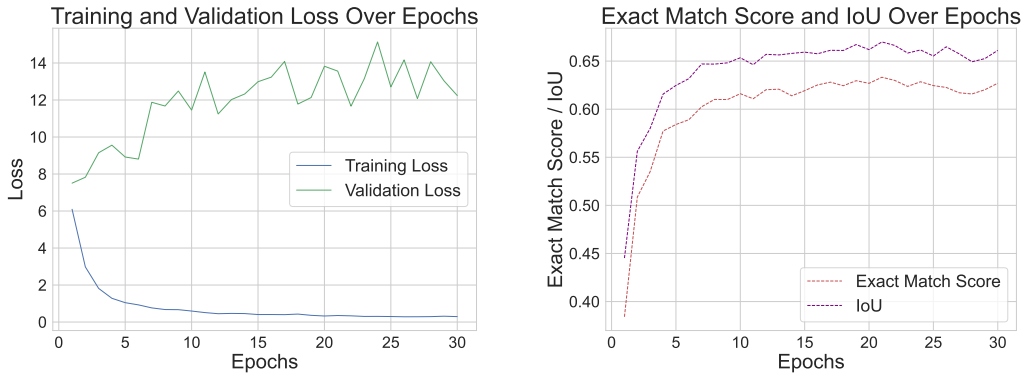
# Chapter 6

# Experimental Results

## 6.1 Quality of Model and text-image Dataset

### 6.1.1 Model Evaluation

**Training and Validation Loss Over Epochs** The model was trained for 30 epochs, and the training loss decreased significantly over the first few epochs, reaching a minimum of approximately 0.29 by epoch 30, as shown in Figure 6.1a. However, the validation loss demonstrated instability, fluctuating at higher values, peaking at 15.13 in epoch 23. This increasing validation loss suggests potential overfitting, with the model performing better on training data compared to the unseen validation set. The divergence between training and validation losses points to the need for improvements in generalization.




**(a)** Training and Validation Loss Over 30 Epochs

**(b)** Exact Match Score and IoU Over 30 Epochs

**Figure 6.1:** Comparison of Model Performance Metrics: (a) Training and Validation Loss and (b) Exact Match Score and IoU over epochs.

**Model Accuracy: Exact Match and Intersection-over-Union Scores** Both the Exact Match Score and IoU were tracked over the 30 epochs, as depicted in Figure 6.1b. The Exact Match Score increased from 0.45 to a maximum of 0.627, while IoU improved from 0.46 to 0.661. This gradual increase in the two metrics indicates that the model progressively learned to predict text-image matches with greater accuracy. The rising IoU scores demonstrate that the model increasingly recognized larger portions of the tokens corresponding to the descriptions.

**IoU Distribution Analysis** The distribution of IoU scores over the 30 epochs is visualized in Figure 6.2, highlighting a pattern of convergence. In early epochs, a more substantial percentage of samples had IoU values of 5% or less, indicating poor recognition. This percentage decreased from 46.5% in the first epoch to 29.3% by the end of training. In contrast, the number of examples where the model recognized 95% or more of the tokens rose from 38.5% to 62.7%. This suggests that the model moved from a state of limited recognition to a state of almost full recognition of tokens within the descriptions for a majority of examples.

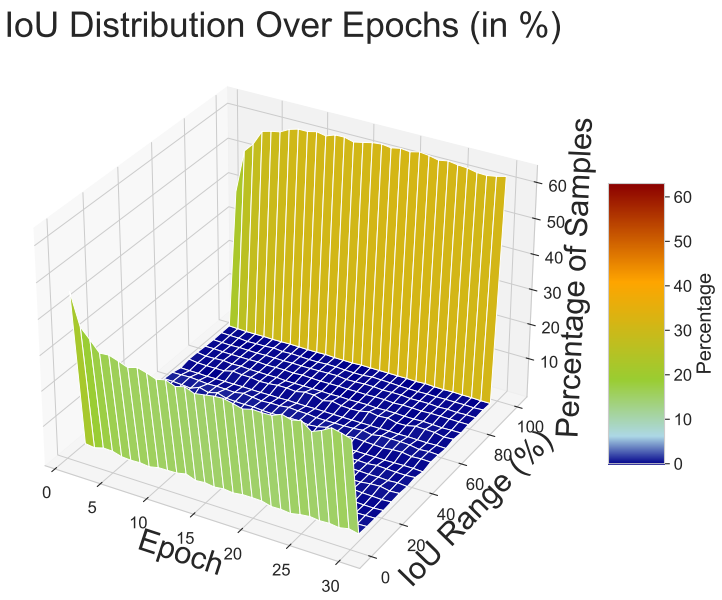



**Figure 6.2:** IoU Distribution Over 30 Epochs (in %)

**Manual Evaluation of Examples** A manual evaluation was conducted to further assess the model's performance, focusing on four different sets of exam-

ples: a complete review of 50, from unseen data randomly selected examples, 25 names, and 25 examples located either directly before or after an image. The evaluation results are displayed in Table 6.1, and a visual representation is provided in Figure 6.3.

For the 50 random examples, 27 instances were fully recognized, meaning the model identified all relevant tokens. One example had partial recognition, and the remaining 22 examples were not recognized at all. In the evaluation of 25 names, the model performed exceptionally well, with 23 examples fully recognized and only one case of partial and non-recognition, respectively. However, the model struggled more with text placement directly around the images. For the examples placed immediately after the image, 18 were fully recognized, 2 had partial recognition, and 5 were unrecognized. Similarly, for examples positioned immediately before the image, 13 examples were fully recognized, 2 examples had partial recognition, and 10 were unrecognized.

This manual evaluation reveals that the model excels in recognizing proper names but faces challenges when processing text directly surrounding images. It performed even worse for examples that could be considered to be chosen randomly from an unseen set of data.

| Example Category | All | >1 and <All | None |
|---|---|---|---|
| 50 Examples | 27 | 1 | 22 |
| 25 Names | 23 | 1 | 1 |
| 25 After Image | 18 | 2 | 5 |
| 25 Before Image | 13 | 2 | 10 |

**Table 6.1:** Manual Evaluation of Examples: Number of Correctly Recognized Descriptions in Various Categories

### 6.1.2 Dataset Evaluation

The evaluation was conducted on the dataset that was constructed by extracting images based on whether a description can be found in the text surrounding an image. Immediately available are is the confidence of the model for the start and end indices of the description. Additionally a CLIP score was calculated that allows a direct comparison of image and text description of that image.

**Clip Score Analysis** The clip score serves as an overall metric to evaluate the similarity between the text prompt and the extracted image. Figure 6.4 displays a box plot of the clip scores along with a histogram of their distribution. The summary statistics for clip scores reveal that they span from a minimum value of 0.1076 to a maximum of 0.3782, with a median score of

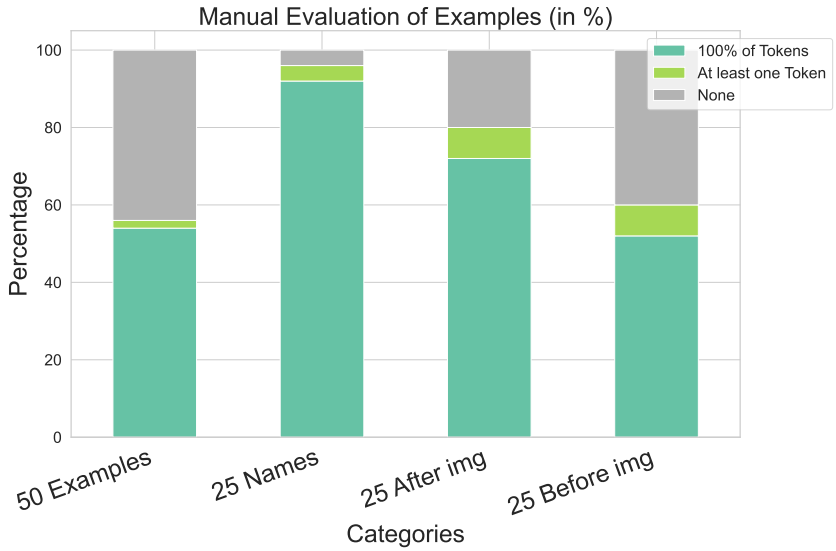



**Figure 6.3:** Manual Evaluation of Examples (in %)

0.2294 and an interquartile range (IQR) between 0.2009 and 0.2711 (Q1 and Q3, respectively).

The clip score distribution shown in Figure 6.4 reveals a positively skewed distribution, with the majority of clip scores clustering around the median value. This indicates that while some generated images exhibit high similarity to their corresponding prompts, a significant portion exhibits only moderate alignment. The box plot also reveals the presence of outliers on the higher end, which correspond to particularly well-matched image-prompt pairs.

**Confidence Metrics of Extracted Descriptions** The confidence metrics for the span prediction model, used to extract image descriptions, are summarized in Table 6.2 and visualized in Figure 6.5. Four confidence metrics were evaluated: start confidence, end confidence, average confidence, and span confidence.

| Metric | Min | Q1 | Median | Q3 | Max |
|---|---|---|---|---|---|
| Start Confidence | 0.0089 | 0.1200 | 0.2512 | 0.5308 | 0.9892 |
| End Confidence | 0.0081 | 0.1154 | 0.2482 | 0.4580 | 0.9602 |
| Average Confidence | 0.0111 | 0.1436 | 0.2938 | 0.4778 | 0.8531 |
| Span Confidence | 0.0001 | 0.0163 | 0.0695 | 0.1793 | 0.7203 |

**Table 6.2:** Descriptive statistics of confidence metrics

The start confidence, as depicted in the first box plot of Figure 6.5, has a

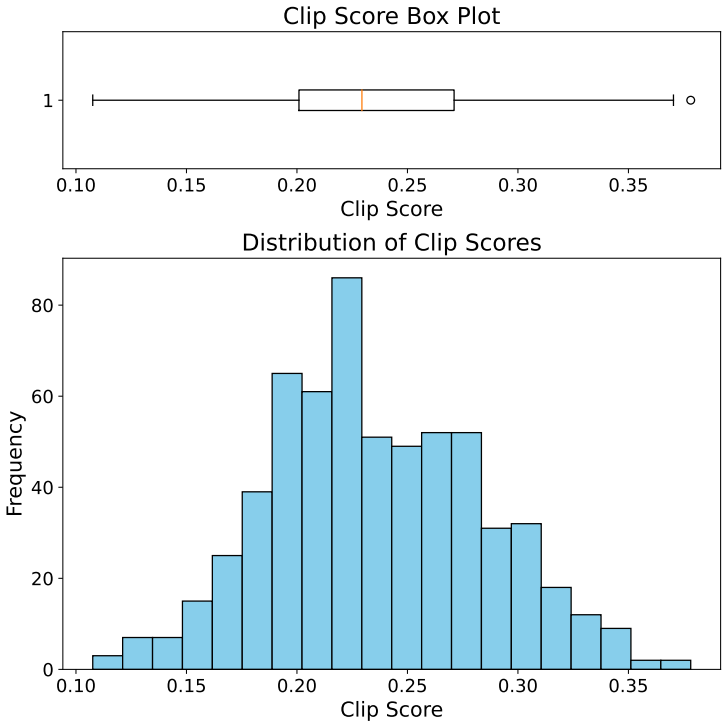



**Figure 6.4:** Box plot and distribution of CLIP scores

wide range, with values spanning from 0.0089 to 0.9892 and a median value of 0.2512. The interquartile range indicates moderate variation, with half of the values falling between 0.1200 (Q1) and 0.5308 (Q3). The end confidence follows a similar pattern, with slightly lower upper bounds and a slightly lower median (0.2482).

The average confidence values are notably more centralized compared to start and end confidences, with a median of 0.2938 and a narrower IQR (0.1436 to 0.4778). The span confidence exhibits the most substantial variability, as seen by its wide range from 0.0001 to 0.7203 and the presence of numerous outliers. The median span confidence is 0.0695, indicating that most of the predictions generated low confidence in capturing the entire span of the description.

The performance of the span prediction model in extracting image descriptions can be inferred from the median and IQR of the confidence metrics. The relatively low span confidence, combined with its substantial variability, suggests that the model struggles with predicting the full span of the image description. However, the higher confidence in individual token predictions (start and end) reflects that the model is better at identifying the specific start and end tokens in isolation rather than the entire sequence.

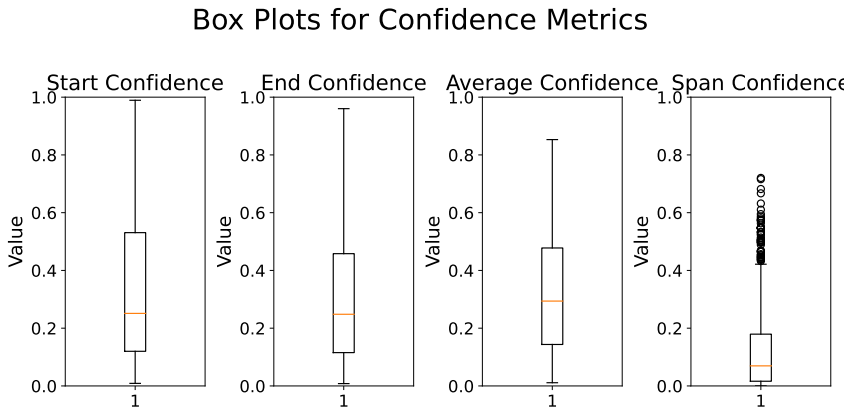



**Figure 6.5:** Box plots for confidence metrics of the span prediction model

**Correlation Between Clip Score and Span Confidence** To further investigate the relationship between the confidence of the extracted descriptions and the similarity of the generated images to their prompts, a correlation analysis was conducted between the span confidence and the CLIP score. The scatter plot in Figure 6.6 visualizes this relationship along with a fitted regression line. The correlation coefficient between the two variables was calculated to be -0.1090, with a p-value of 0.0067, indicating a statistically significant, albeit weak, negative correlation.

This result highlights an inverse relationship between span confidence and CLIP score, suggesting that higher confidence in span prediction does not correspond to better alignment between image and text. In fact, as span confidence increases, the CLIP score tends to decrease slightly, which is counterintuitive to what would be expected for an effective model. This behavior could suggest that while the model might be overconfident in certain span predictions, the descriptions it generates may not align well with the images, as evidenced by the lower CLIP scores.

The fitted regression line in Figure 6.6 further demonstrates this inverse relationship, with the trend line showing a slight downward slope. This finding suggests that improvements to the span prediction model should focus not only on increasing confidence but also on ensuring that the predicted spans contribute meaningfully to the alignment between the image and its description.

The overall performance evaluation reveals moderate alignment between generated images and their prompts, as reflected in the clip scores. The confidence metrics of the description extraction model indicate room for improvement in capturing entire descriptions with higher accuracy. These findings could inform subsequent model refinements and improvements in prompt-

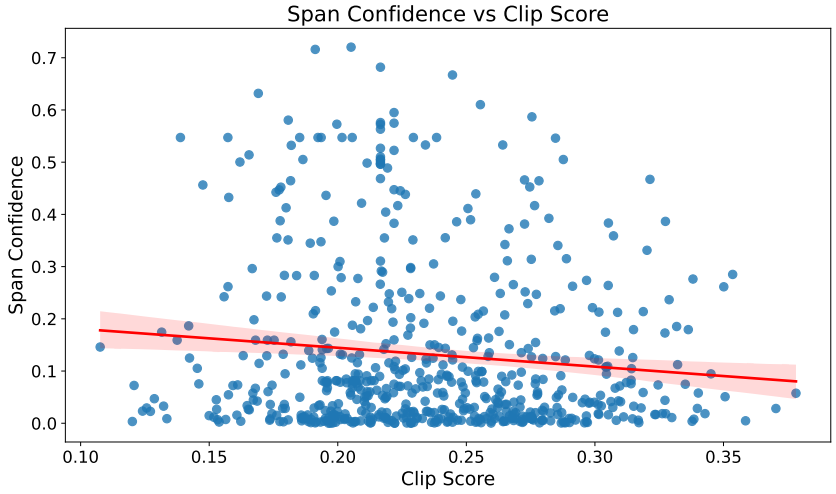



**Figure 6.6:** Span confidence vs. CLIP score with regression line

image alignment.

## 6.2 Manual Comparison of Images

### 6.2.1 Prompt Generation and Image Creation

To evaluate the performance of both the native Stable Diffusion model (SD) and the fine-tuned version (SDFT), 100 groups of prompts were generated. Each group consisted of five prompts, starting with a simple object and progressively becoming more complex. For instance, the first prompt might simply be "cat", while subsequent prompts in the group added more details, culminating in the fifth prompt, which could be something like "gray cat on sofa wearing hat playing with ball of yarn." This method resulted in 500 unique prompts.

However, initial testing revealed that some of these prompts were nonsensical or difficult to interpret, which led to the creation of a second set of 500 prompts that were manually refined to ensure they described more plausible scenarios. Thus, the total set consisted of 1000 prompts, split evenly between questionable and sensible categories.

For each of the 1000 prompts, images were generated using both the SD and SDFT models, resulting in 1000 generated images for each model, for a total of 2000 images.

Figure 6.7 shows two sets if images generated by the models for the progressively complex prompt "gray cat on sofa wearing hat playing with ball of

yarn". Here you can see that both models already performed well on this task and were able to represent all aspect of the prompt. In Appendix A are more images that depict two prompts from each set.

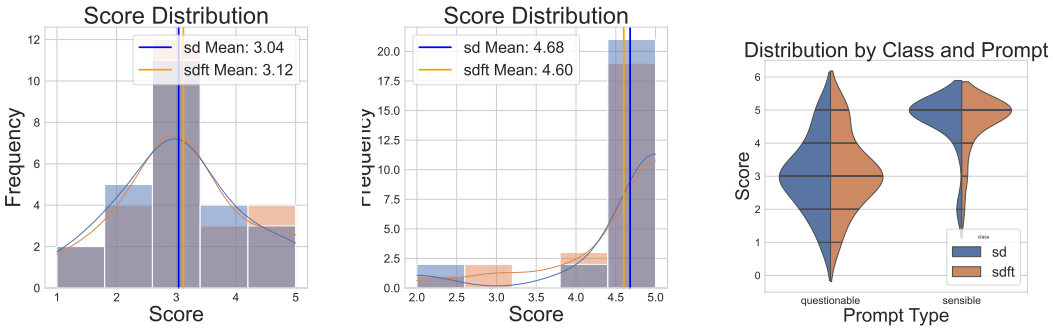

**Figure 6.7:** Stable Diffusion XL generated images that shows a cat and was iteratively prompted with additional text to make it more complex. Starting with "cat" and ending with "gray cat on sofa wearing hat playing with ball of yarn"

### 6.2.2 Manual Evaluation

A human annotator manually evaluated 25 images for both Stable Diffusion models and both sets of prompts, based on how well it conformed to the given prompt. This gave an evaluation for a total of 100 images. Each image was rated on a scale from 0 to 5, with 5 indicating that all aspects of the prompt were represented in the image, and 0 indicating that no aspect of the prompt was captured. Importantly, all evaluated images contained at least one aspect of the prompt, even for some of the more nonsensical inputs.

The evaluation results were separated into two categories: "questionable" and "sensible" prompts, with both the SD and SDFT models evaluated under these categories. The distribution of scores and a Violin plot showing the distribution of each group are presented in Figure 6.8.

**Analysis of Evaluation Results** The results show that the SDFT model slightly outperformed the native SD model in both categories of prompts. For questionable prompts, both models struggled, with many images scoring between 2 and 3 points. However, SDFT managed to generate slightly more images that scored a 5, with a slightly better balance in conforming to the aspects of the prompts.




**(a)** Questionable Prompts **(b)** Sensible Prompts **(c)** By Class and Prompt

**Figure 6.8:** Score Distribution for Questionable and Sensible Prompts, and Overall Summary of Scores

For sensible prompts, both models performed well, but the native SD model produced more images that scored a perfect 5. However, SDFT was more consistent in generating images that captured nearly all aspects of the prompts, with fewer images scoring low (1 or 2 points).

Overall, the fine-tuned SDFT model displayed a better ability to generate coherent images for nonsensical prompts, while the native SD model performed slightly better with sensible prompts, especially in generating images that matched all prompt aspects.

To better visualize the distribution of scores across different evaluations, violin plots were used instead of boxplots (). Violin plots provide a detailed representation of the density of evaluations at each score level, helping to illustrate where the majority of the evaluations were concentrated.

In the case of questionable prompts, the distribution, visible in Figure 6.8c, shows a wider spread across different score values for both SD and SDFT models, with the peak density around the median score of 3. For sensible prompts, however, the majority of the evaluations are clustered around the score of 5, especially for the SD model, where the distribution is heavily skewed toward the highest score.

In addition to the plots, Table 6.3 provides a summary of the evaluation results for both SD and SDFT models across the two types of prompts. The statistics include the count of evaluations (n), the mean score, standard deviation (std), and key percentiles (min, 25%, median (50%), 75%, max).

The results in Table 6.3 reinforce the findings from the plots. For the sensible prompts, both models show much higher mean scores, 4.68 for the SD model and 4.60 for the SDFT model, with a smaller standard deviation, indicating more consistent evaluations. The median for both models in this case is 5, with the 25% and 75% percentiles showing that nearly all evaluations

| Class | Prompt | n | Mean | Std | Min | 25% | 50% | 75% | Max |
|---|---|---|---|---|---|---|---|---|---|
| SD | Questionable | 25 | 3.04 | 1.10 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| SDFT | Questionable | 25 | 3.12 | 1.13 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| SD | Sensible | 25 | 4.68 | 0.85 | 2.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| SDFT | Sensible | 25 | 4.60 | 0.82 | 2.0 | 5.0 | 5.0 | 5.0 | 5.0 |

**Table 6.3:** Summary statistics for the SD and SDFT models across questionable and sensible prompts.

gave a score of 5. The slight difference in mean scores and standard deviations reflects the slightly higher concentration of perfect scores for the SD model.

## 6.3 Automatic Analysis of Fine-Tuned Stable Diffusion vs. Native Stable Diffusion

To evaluate the effectiveness of fine-tuning Stable Diffusion for generating images that better align with specific prompts, a comparative analysis between the images generated by the native Stable Diffusion model ('sd') and the fine-tuned version ('sdft') was conducted. The analysis focused on two main metrics:

- **Compliance Score**: Measures how well the generated images match the textual prompts used to generate them. This was evaluated using a CLIP-based cosine similarity metric.
- **Complexity Score**: Quantifies the level of detail and intricacy within each image, using an edge-detection-based metric that counts the number of edges (details) visible in the image.

The following subsections present a detailed summary of the results obtained from this analysis.

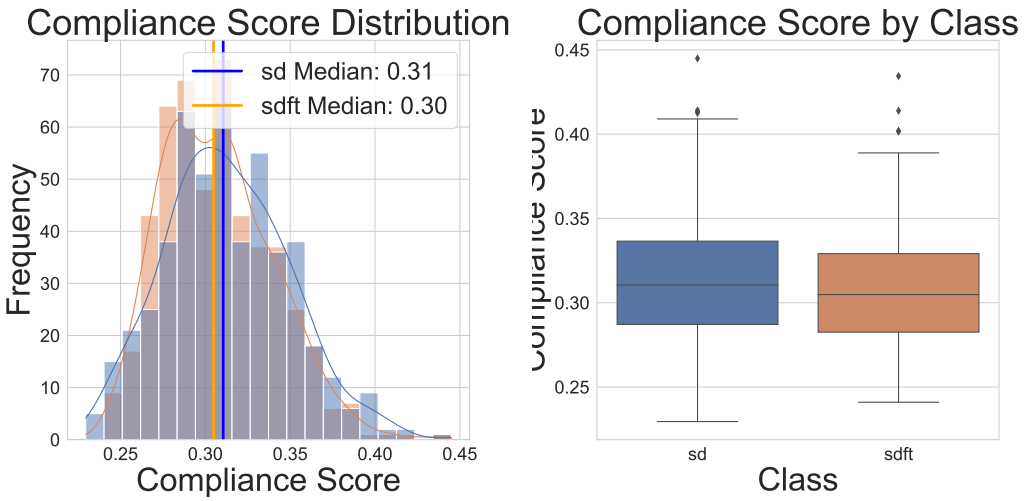
**Compliance Score** The compliance score measures the alignment between the image and its corresponding textual prompt. A higher compliance score indicates a better match between the visual content of the image and the semantics of the prompt.

For the 'sd' model, the compliance score ranged from **0.2295** to **0.4449**, with a median of **0.3105**. The interquartile range (IQR), represented by the 25th percentile and 75th percentile, was **0.2872** and **0.3366**, respectively.

For the fine-tuned 'sdft' model, the compliance score ranged from **0.2410** to **0.4345**, with a slightly lower median of **0.3048**. The IQR for this class was

slightly narrower, with a 25th percentile of **0.2826** and a 75th percentile of **0.3291**.

Both models performed comparable in terms of compliance, with the fine-tuned model showing a slight reduction in the median compliance score. This may suggest that, while the fine-tuned model was specialized to the dataset, it did not significantly increase the overall alignment with the prompts beyond the performance of the native model.




**(a)** Compliance Score Histogram **(b)** Compliance Score Boxplot

**Figure 6.9:** Comparison of Compliance Score Distributions of generated prompts

**Complexity Score** The complexity score assesses the level of detail present in the images, with higher scores reflecting a higher number of edges detected in the image. These edges often correspond to intricate textures or more detailed visual elements.

The ‘sd‘ model produced images with complexity scores ranging from **0.0039** to **0.1637**, with a median complexity score of **0.0521**. The 25th percentile was **0.0343**, and the 75th percentile was **0.0757**.

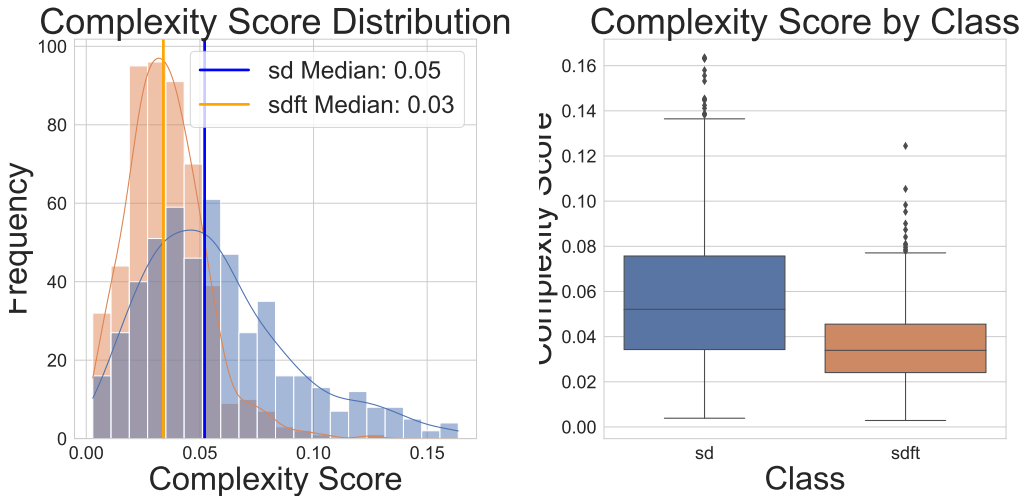
In contrast, the ‘sdft‘ model generated images with lower complexity, ranging from **0.0029** to **0.1245**, with a median complexity score of **0.0339**. The IQR for ‘sdft‘ was significantly lower, with the 25th percentile at **0.0241** and the 75th percentile at **0.0455**.

The results indicate that the fine-tuned model produced less visually intricate images compared to the native model. This reduction in complexity could imply that the fine-tuning process favored simpler images that aligned better

with prompt requirements but may have lost some of the visual richness found in the native Stable Diffusion outputs.

To further illustrate the difference in image quality and alignment, histograms of the complexity scores for both models are presented (Figures 6.10). Here the difference in the distribution becomes clear with the native SDXL being represented more in higher score regions.

- **Figure 1** shows the complexity score distribution, highlighting that the native Stable Diffusion model ('sd') generates images with more intricate visual details (as indicated by the higher median and longer tail on the histogram) compared to the fine-tuned model ('sdft').
- **Figure 2** presents the compliance score distribution, where the difference between the models is less pronounced, though the native model exhibits slightly higher overall compliance with the prompts.




**(a)** Complexity Score Histogram **(b)** Complexity Score Boxplot

**Figure 6.10:** Comparison of Complexity Score Distributions of generated prompts

The fine-tuning of the Stable Diffusion model resulted in images that were slightly less detailed but comparable in terms of compliance with the textual prompts. The results suggest that fine-tuning may have a trade-off between producing visually simpler images and achieving better prompt alignment. Further experimentation could explore techniques to maintain or even enhance image complexity while improving compliance.

## 6.4 Outlook

The overall performance of the current system, particularly in the span prediction model, did not meet initial expectations, which subsequently impacted the quality of the downstream dataset and the fine-tuning of the Stable Diffusion model. Future iterations should investigate the difference in prediction quality that is currently performing well on names, and descriptions close to the image yet fails to reliably predict descriptions when they are buried deeper in the text. Higher prioritization should also be set to get a more comprehensive evaluation of key performance metrics. For instance, logging inference time during span prediction would provide valuable insights into the model's efficiency. If this is combined with a comparison to a state of the art method like the one used for LAION-5B, it could be decided whether the method proposed in this thesis is worth developing further.

Future work could explore replacing DistilBERT with SpanBERT for span prediction tasks. SpanBERT's span-based pre-training and focus on capturing relationships between spans could improve the model's ability to link textual descriptions with images more effectively, particularly in complex multimodal datasets where understanding the broader context of spans is crucial.

A shortcoming of the current approach is the inability to scale the dataset as originally envisioned. While the goal was to extract large-scale multimodal data, ineffectiveness in the extraction pipeline, particularly with span prediction, limited the dataset's size and quality. Addressing these limitations will be key to realizing the full potential of the pipeline and enabling the training of more robust models.

Improvements could be achieved by rethinking the data management system. The current reliance on `jsonl` files, while functional, has limitations in terms of ease of maintenance, query efficiency, and scalability. An SQLite database could offer substantial benefits in this regard, providing structured, indexed data storage that is easier to access, write, and maintain. This shift could also simplify the overall pipeline, potentially merging steps such as training data extraction and model training into a more cohesive process. By reducing complexity, this could enhance both development speed and maintainability.

Moreover, integrating SQLite into the workflow might support a more robust data management framework that could handle larger datasets more effectively, especially as models like Stable Diffusion and CLIP continue to scale in terms of input data and computational resources. Proper database management would also allow for better versioning and tracking of extracted data, which is crucial for long-term reproducibility and model fine-tuning.

Looking forward, advancements in multimodal datasets extraction, espe-

cially from web archives, will require increased attention to both system performance and the efficient handling of large-scale datasets. Further research into optimizing span prediction models for text-image pair extraction could provide significant gains.

# Appendix A

# Examples of Generated Images


**Figure A.1:** Images generated with SD and SDFT using the sensible prompts: "computer" to "A shiny computer in an office displaying code running a simulation"


**Figure A.2:** Images generated with SD and SDFT using the sensible prompts: "bicycle" to "A fast bicycle on the street being repaired being ridden"


**Figure A.3:** Images generated with SD and SDFT using the partly nonsensical prompts: "Mountain" to "A fast mountain under the table with sunglasses reading a book"

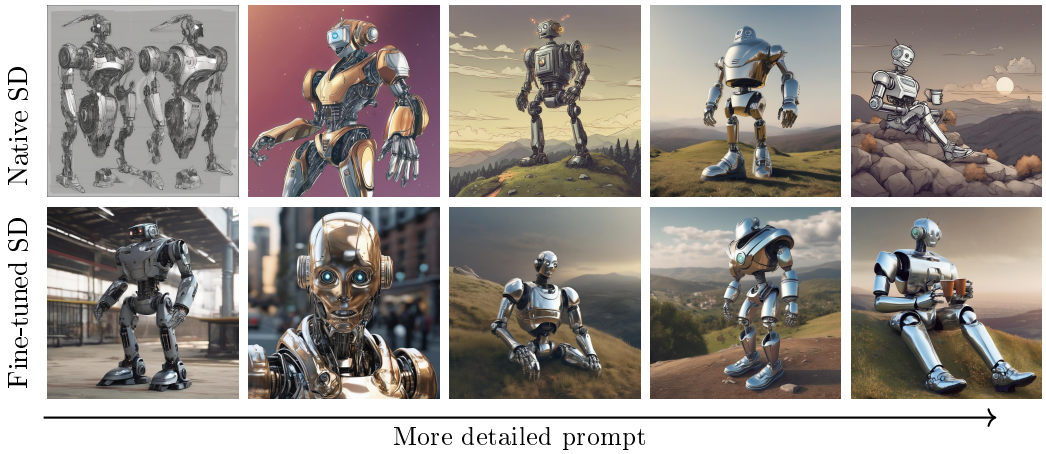



**Figure A.4:** Images generated with SD and SDFT using the partly nonsensical prompts: "robot" to "A shiny robot on a hill wearing shoes drinking coffee"

# Bibliography


Bevendorff, J., Potthast, M., and Stein, B. (2021). FastWARC: Optimizing Large-Scale Web Archive Analytics. In Wagner, A., Guetl, C., Granitzer, M., and Voigt, S., editors, *3rd International Symposium on Open Search Technology (OSSYM 2021)*. International Open Search Symposium.

Bevendorff, J., Stein, B., Hagen, M., and Potthast, M. (2023). Chatnoir resiliparse (version 1.0.0). `https://doi.org/10.5281/zenodo.8262470`.

Common Crawl (2024). Common crawl. Accessed: 2024-09-21.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., and Rombach, R. (2024). Scaling rectified flow transformers for high-resolution image synthesis.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *CoRR*, abs/2207.12598.

Howard, J. and Ruder, S. (2018). Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Huang, Y., Tang, Z., Chen, D., Su, K., and Chen, C. (2020). Batching soft iou for training semantic segmentation networks. *IEEE Signal Processing Letters*, 27:66–70.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2019). Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2023). SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *CoRR*, abs/2102.12092.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *CoRR*, abs/1606.03498.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., and Jitsev, J. (2022). LAION-5B: an open large-scale dataset for training next generation image-text models. *CoRR*, abs/2210.08402.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (2021). LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *CoRR*, abs/2111.02114.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *CoRR*, abs/1411.1792.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924.

# Table of Acronyms

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **DPMs** | Diffusion Probabilistic Models |
| **CC** | Common Crawl |
| **CLIP** | Contrastive Language-Image Pre-Training |
| **DOM** | Document Object Model |
| **FID** | Fréchet Inception Distance |
| **GAN** | Generative Adversarial Networks |
| **HTML** | Hypertext Markup Language |
| **IS** | Inception Score |
| **soft-IoU** | soft Intersection-over-Union |
| **IoU** | Intersection-over-Union |
| **LAION-5B** | Large-scale Artificial Intelligence Open Network 5 Billion Image Data Set |
| **LPIPS** | Learned Perceptual Image Patch Similarity |
| **LSTM** | Long Short-Term Memory |
| **NLP** | Natural Language Processing |
| **NSFW** | Not Safe For Work |
| **RNN** | Recurrent Neural Networks |
| **regex** | regular expressions |
| **SD** | Stable Diffusion |
| **SDXL** | Stable Diffusion XL |
| **VAE** | Variational Autoencoders |
| **WARC** | Web ARChive |
| **WAT** | Web Archive Transformation |