



Introduction to Multimodal Models

SIMON BETHDAVID, HELENA BJÖRNESJÖ,
TOVE GUSTAVI, HANNA LILJA, MAGNUS ROSELL,
JOHAN SABEL, EDWARD TJÖRNHAMMAR,
SEBASTIAN ÖBERG



Simon Bethdavid, Helena Björnesjö,
Tove Gustavi, Hanna Lilja, Magnus Rosell,
Johan Sabel, Edward Tjörnhammar,
Sebastian Öberg

Introduction to Multimodal Models

Cover image: “Multimodal librarian” by Edward Tjörnhammar (License CC0-1.0)

Titel	Introduktion till multimodala modeller
Title	Introduction to Multimodal Models
Report no	FOI-R--5505--SE
Month	April
Year	2024
Pages	52
ISSN	1650-1942
Client	Swedish Armed Forces
FOI Research area	C3 and Human Factors
Armed Forces R&T area	Command and Control
Project no	E38538
Approved by	Emil Hjalmarson
Division	Cyber defence and C2 technology

This work is protected by the Swedish Act on Copyright in Literary and Artistic Works (1960:729). Citation is permitted in accordance with article 22 in said act. Any form of use that goes beyond what is permitted by Swedish copyright law, requires the written permission of FOI.

Summary

During the last decade, there has been an extraordinary development of innovative artificial neural network models. Most of these models were constructed to handle only one *modality*, where a modality can be thought of as a channel for communication or a type of data, such as text or images. However, in the last few years, machine learning models based on the novel *transformer* architecture have been able to produce impressive results on tasks that require the ability to process two or more modalities jointly. The multimodal capabilities make these models better suited to handle a variety of problems that arise in our multimodal world.

The most well-known multimodal models so far are those that combine text and images, for instance by generating images from text prompts or by answering questions about images. Similarly, advances are being made for models combining text and video. Other models combine text and sound, for music generation, or for text-to-speech and speech-to-text conversion. In addition, there are models that are able to combine more than two modalities, laying the foundation for new solutions to complicated problems in fields such as data fusion and robotics. For example, while industrial robots work well in controlled environments, a multi-purpose robot in an uncontrolled environment needs to be able to perform agile task and motion planning, based on input from a variety of sensors. This ability can be seen in early work on multimodal models for robotics.

This report provides an overview of recent developments in the field of multimodal neural network models. A selection of multimodal models, developed in recent years, is presented. The focus is on models that process *media data*, where media data is to be understood as data primarily intended for human communication, such as text, images, sound, and video.

Although the multimodal models used today have limitations, their ability for automatic multimodal reasoning is, in parts, so impressive that we have to ask ourselves in what ways multimodal machine learning models may come to impact our lives in the years to come.

Keywords

artificial intelligence, machine learning, deep learning, deep neural networks, multimodal models, language models

Sammanfattning

Under det senaste årtiondet har det gjorts betydande framsteg inom utvecklingen av maskininlärningsmodeller, baserade på artificiella neurala nätverk. De flesta av dessa modeller är konstruerade för att hantera en enda så kallad *modalitet*. En modalitet kan betraktas som en kommunikationskanal eller datatyp, där text och bild är exempel på två olika modaliteter. De senaste åren har emellertid maskininlärningsmodeller, baserade på den så kallade *transformer*-arkitekturen, uppvisat imponerande resultat på uppgifter som kräver att två eller flera modaliteter behandlas gemensamt. De multimodala egenskaperna gör dessa modeller väl lämpade att hantera en mängd olika problem som uppstår i vår multimodala värld.

De multimodala modeller som hittills rönt störst uppmärksamhet är de som kombinerar text och bild för att exempelvis generera bilder baserade på textbeskrivningar eller svara på frågor om bilder. Parallellt utvecklas modeller som kombinerar text och video. Det finns även modeller som kombinerar text och ljud för att generera musik, eller för att konvertera text till tal och tal till text.

Vissa modeller har förmåga att kombinera fler än två modaliteter och lägger grunden för nya lösningar på komplicerade problem inom bland annat datafusion och robotik. Industriell robotik fungerar väl i kontrollerade miljöer, men mer generell robotik i okontrollerade miljöer kräver möjligheten till flexibel uppgifts- och rörelseplanering baserad på data från olika typer av sensorer; något som multimodala modeller redan nu visar lovande förmåga att kunna åstadkomma.

Denna rapport ger en översikt över den senaste utvecklingen inom forskningsfältet Multimodala Neuronnätmodeller och beskriver ett urval av de modeller som utvecklats de senaste åren. Rapporten fokuserar på modeller som behandlar *mediedata*, det vill säga data som primärt är avsedd för mänsklig kommunikation. Text, bild, ljud och video är exempel på mediedata.

Även om dagens multimodala modeller har begränsningar så är deras förmåga till automatiskt multimodalt resonande, åtminstone i vissa avseenden, så imponerande att vi måste ställa oss frågan: hur kommer multimodala maskininlärningsmodeller att påverka våra liv under kommande år?

Nyckelord

artificiell intelligens, maskininlärning, djupinlärning, djupa neuronnät, multimodala modeller, språkmodeller

Contents

1	Introduction	7
1.1	Multimodal Models for Media	7
1.2	Scope of Report	10
1.3	Content and Reading Instructions	10
2	Background	13
2.1	Machine Learning in the Age of Deep Neural Networks	13
2.2	Representations and Embeddings	14
2.3	Encoders and Decoders	14
2.4	Convolutional Neural Networks	15
2.5	Diffusion Models	15
2.6	The Transformer	16
2.7	Vision Transformer	18
3	Image – Text	21
3.1	CLIP	21
3.2	Stable Diffusion	23
3.3	PaLI and PaLI-X	23
3.4	Detic	24
3.5	SAM	25
3.6	SEEM	26
4	Sound – Text	29
4.1	Whisper	29
4.2	SeamlessM4T	31
5	Video – Text	33
5.1	X-CLIP	33
5.2	Make-A-Video	34
5.3	Sora	35
6	Many Modalities	37
6.1	Gato	37
6.2	ImageBind	37
6.3	Robotics at Google	39
7	Conclusions	45
	References	47

1 Introduction

In the last decade, deep artificial neural networks have had a significant impact on computer science. The rapid development of neural networks and associated technologies is likely to continue for many years, and the effects on society are hard to predict. Until recently, most advancements in deep learning¹ – including powerful image recognition methods, chess-playing algorithms with super-human performance, and automatic translation between numerous languages – had been accomplished by processing only one type of data at a time. In recent years, however, the deep learning field has seen rapid advancements in *multimodal* neural network models, i.e., models that combine information from multiple *modalities*. In this context, a modality can be thought of as a channel of communication,² or more broadly as a specific type of data.

It is interesting to note that humans typically experience and process the world in a “multimodal manner” through our different senses. One possible requirement for declaring an automatic system “intelligent” could be that it demonstrates a capacity, similar to that of a human, to combine information from different modalities to “reason” about the world. The study of automatic multimodal systems is not new,³ but developments in artificial neural networks provide new possibilities, and impressive results are continuously presented.

This report will provide an introduction to multimodal neural network models. We use the term *model* as an abbreviation for *neural network model*, and occasionally also to denote a (multimodal) *system* that consists of two or more neural network models.

1.1 Multimodal Models for Media

This report focuses on multimodal models that are applied to *media*, by which we mean data that is used for human communication, i.e., text, sound, image, and video. As media data is available in large quantities, it is comparatively easy to find suitable training data for these models. Well-known unimodal models, such as the model backing the first version of ChatGPT, as well as recent multimodal models, are trained on media data.

Even when considering only text, sound, image and video as modalities, the number of possible applications for multimodal models is huge. Examples include the generation and alteration of media, detection and categorization of various phenomena in media, and transformations, translations, and comparisons between media types. These applications can handle pairs of modalities (e.g., an image and a text snippet) or a combination of several modalities. Considering that human activities are increasingly supported by computers, and that many human-computer interfaces are based on media data such as text, sound and images, the potential impact of multimodal models should be apparent.

Textual data has a special significance at the interface between human and machine. Of all data available on digital format, textual data is likely the modality that contains the most information about how humans think, feel and, reason about the surrounding world. However, text is made up by abstract symbols, which up until the development of large language models (LLMs) were challenging to interpret and process using computers. In recent years we have seen how language models, which are trained on large amounts of textual data, are able to learn patterns and structures in the texts which enable them to – to some extent – mimic human reasoning. This ability is demonstrated in applications like ChatGPT.

One of the most remarkable properties of unimodal deep learning models is that they can be pre-trained on large amounts of data and then fine-tuned for specific tasks. This means that the general capabilities of a model can be reused to solve specific tasks. This is true specifically for language models, which can be pre-trained on vast amounts of text. When language

¹ Machine learning based on deep artificial neural networks.

² See for instance [https://en.wikipedia.org/wiki/Modality_\(human-computer_interaction\)](https://en.wikipedia.org/wiki/Modality_(human-computer_interaction)) (Visited 2023-10-18).

³ See for instance https://en.wikipedia.org/wiki/Multimodal_learning (Visited 2023-10-07).

models are combined with models for other modalities, new possibilities for “reasoning” over these modalities emerge. Examples of this can be seen at the end of this chapter (pages 11-12), where Examples 1 and 2 demonstrate *visual question answering* using the multimodal (text and image based) system ChatGPT 4. See also the robot models described in Chapter 6, that are capable of planning series of actions based on textual commands and visual input.

Table 1: Some of the multimodal models for media described in this report. The models at the top are primarily used to generate an output from an input, as explained in Section 1.1. The arrows represent the input and output of the models. The models at the bottom are primarily used to create representations that connect several modalities listed within curly brackets.

Modalities		Example Model	Section
Text	→ Image	Stable Diffusion	3.2
Text	↔ Speech	Whisper	4.1
Text	→ Video	Make-A-Video	5.2
[Text, Image(s)]	→ Text	PaLI	3.3
[Text, Image(s)]	→ Action	RT-2	6.3.4
{Text, Image}		CLIP	3.1
{Text, Video}		X-CLIP	5.1
{Image, Text, Audio, ... Depth, Thermal, IMU readings}		ImageBind	6.2

Superficially, neural network models work similarly to other computer programs; they take an input and produce an output. For instance, the input to a unimodal language model, such as the first version of ChatGPT, is text, and the output is text. This can be abbreviated by writing: Text → Text.

Table 1 gives an overview of some of the possible modality combinations, as well as a subset of the models presented in this report. In the top part of the table, the left column specifies input and output modality types for the models. The lower part of Table 1 contains models that take data of several modalities as input (listed within curly brackets) but do not output data of a specific modality. These models are instead primarily used to create computer representations of the data in the input modalities. The model CLIP [55], for instance, learns how to relate texts and images by looking at many images and their corresponding caption. When the model has been trained, it can generate representations for any image and any text that can then be compared with one another, opening up many potential use cases, such as searching for images similar to a text.

This report is more or less restricted to multimodal models for media, but it is likely that many of the lessons learned from the study of these models could be useful when developing models that handle other types of data as well, for example different types of sensor data.

1.1.1 Applications and Challenges

Multimodal media models can be used in many application domains to solve various problems. Consider, for instance, artificial intelligence assistants such as Microsoft’s Copilot⁴ and Google’s Duet AI⁵, which can help with information extraction, and with generation and alteration of media. Advanced AI assistants can even perform creative tasks, such as creating images and composing music from text instructions, and complex tasks such as multimodal information assessment. In essence, multimodal systems and applications push the boundary from what used to be creative work into the realm of dreary production.

⁴<https://blogs.microsoft.com/blog/2023/09/21/announcing-microsoft-copilot-your-everyday-ai-companion/> (Visited 2023-10-14.)

⁵<https://workspace.google.com/blog/product-announcements/duet-ai> (Visited 2023-10-14.)

Recent multimodal media models are technically very advanced. Still, many technical challenges remain. Many of them are associated with the models' neural-network based architecture. Since neural network models are trained on datasets, the models will reflect the shortcomings of the datasets used to train them [74–76]. For instance, training data may not contain the information needed in a particular application, or training data can contain unwanted bias, like prejudices [40, 41, 73]. As a consequence, multimodal media models can produce both biased and incorrect results due to the quality of training data. Today, big corporations control the largest datasets and computational resources. They also either support or participate in much of the research within the deep learning field. It is a problematic fact that many neural network models, including multimodal media models, are directly or indirectly depending on fundamental models which are trained on privately owned data that is not available for inspection.

Furthermore, neural network models are complex and may not pick up on information in the training data or the user's input to the model in the way a human would expect. This can lead to unexpected and seemingly strange output results, often called *hallucinations*. As many users have experienced, it typically takes a substantial amount of trial and error to formulate a successful text prompt for an application like image generation.

Like most technologies, multimodal media models can be used for both well-intended and malicious purposes. From a security perspective, it is concerning that multimodal media models are excellent tools for generating highly realistic faked media content. Such content can be used in influence operations and for other types of deception, and the high quality of the generated content makes it more challenging to distinguish between faked and genuine material. Also, multimodal media models could harness information about individual persons or critical infrastructure from available information sources, including text, images, sound and video. Depending on the intent, this type of monitoring could also be problematic.

Over time, we must assume that the quality and accessibility of generative multimodal media models—regardless of the motive for using them—will give rise to enormous volumes of information, including fake and erroneous data. This means that society has to find ways of safeguarding the trustworthiness of certain types of public information, e.g., government information, university research, and news forecasts. If not, the loss of trust is likely to have devastating effects on the society as we know it.

1.1.2 Development and Availability

Most multimodal models consist of two or more unimodal neural network models, one per modality. Sometimes, the models are trained jointly on all the modalities from the beginning, but often they are trained separately first. In the latter case, the full multimodal model can leverage the information learned from the single modalities from one modality to another; a form of transfer learning

Training a useful unimodal model is both complicated and costly. This is also true for multimodal models. Novel model architectures have been essential to developing deep neural networks, as have the increasing amount of training data and computational resources. The Transformer architecture is fundamental to language models, and increasing the size of the model and the amount of training data has led to increasingly impressive *emergent* capabilities. Similar capabilities are already present in multimodal models, at least those consisting of transformer based language models. Further developments in this direction may happen as more modalities and data are used.

This report focuses on some models which have been described in research papers. Some of the models can easily be used through services, others are available in code repositories, and some are not released by the corporations that developed them. Some services are subscription-based, and others are free.

1.2 Scope of Report

This report introduces multimodal neural network models. It focuses on models applied to media, such as text, sound, images, and videos, although we include some models that incorporate other modalities as well. Furthermore, we have chosen not to include multimodal systems comprising many neural network models.

Even with these restrictions, the landscape of multimodal neural network models is vast. In each chapter, we have chosen to describe a few models we find interesting and believe to be important or useful. There are many other models that could have been included, and that may be equally important and/or useful.

1.3 Content and Reading Instructions

This is a technical report in which we strive to explain the technology in a more approachable manner. Multimodal models build on many unimodal models that have been studied previously. Chapter 2 describes some of the model architectures, techniques, and concepts referenced in the rest of the report. Thus, it is a suitable read for anyone who wants to learn about the technology behind multimodal models. It may be hard to understand the more technical parts of the rest of the report without similar knowledge.

Chapters 3 through 5 describe models that combine text with one other modality, while Chapter 6 describes models that handle several modalities. Each chapter begins with an introduction. The following sections of the chapters are dedicated to one specific model each, and start with a general description of what can be achieved with the model, followed by more technical details.

Finally, Chapter 7 contains a summary of the report and some conclusions.

Example 1: An image and a question is given to ChatGPT 4, which returns a text response. The result is almost perfect. Notice the inference in the last sentence.



Question: Image + What does this tell me?

ChatGPT: The image displays a fuel station under the brand “preem”. Here’s a breakdown of what’s visible:

1. Brand Logo: The “preem” logo at the top, depicting what appears to be a green face.
2. Amenities: Below the logo, there are mentions of what the station offers:
 - café: Indicates there’s a café on the premises.
 - nybakat: This is Swedish for “freshly baked”, suggesting they have freshly baked goods.
 - freshfood: They offer fresh food items.
 - biltvätt: This is Swedish for “car wash”, indicating car washing services are available.
3. Fuel Prices: There’s a board displaying prices for different types of fuel:
 - D: Priced at 24.96 (likely the currency per liter, though the currency isn’t specified).
 - 95: Priced at 20.39.
 - HVO: Priced at 28.29. HVO is a type of renewable diesel.
4. Banner: At the bottom, there’s a banner advertising:
 - “viktigt prisvärd lunch”: This is Swedish for “important value-for-money lunch”.
 - “kaffe ingår”: This means “coffee included” in Swedish.
5. General Ambience: The sky is overcast, and the overall setting suggests an urban environment with multi-storied buildings in the background.

From the information and the use of Swedish, it can be inferred that this is a gas station in Sweden offering various amenities beyond just fuel.

Image by the authors.

Example 2: In this experiment, ChatGPT 4 is provided the same question twice, but associated with two slightly different images. In the first case (left), ChatGPT does not succeed in identifying the building in the picture. In the second case (right), the image shows more of the building and ChatGPT succeeds. Images by the authors.



Question: Image + What building is that?

ChatGPT: Sorry, I cannot help with that.



Question: Image + What building is that?

ChatGPT: The building in the photo appears to be Stockholm City Hall (Stadshuset) located in Stockholm, Sweden. It's an iconic building known for its tower with a golden Three Crowns symbol on top and is the venue for the Nobel Prize banquet held annually on December 10th.

Images by the authors.

2 Background

In this chapter, we describe the gist of the technological background necessary to grasp the multimodal (neural network) models presented in this report. For more in-depth descriptions, there are available resources, e.g., Foster [26]. FOI has also published several reports on neural network models [31, 32, 35, 59, 61, 69].

Multimodal models build upon and use unimodal neural network models, i.e., models that process data from one modality, such as language models. Unimodal models have been studied extensively for around a decade. In the following chapters, we describe some key concepts relating to this field and specific unimodal models used to construct the multimodal models discussed.

2.1 Machine Learning in the Age of Deep Neural Networks

Machine learning has been used in many applications for decades. The introduction of deep learning and deep neural networks radically improved the possibilities. Perhaps the most crucial difference between “traditional” machine learning and deep learning is how the latter can leverage unstructured data, ostensibly letting the dataset perform the feature selection. As such, deep neural networks can be trained from unstructured, unlabeled data. This can be desirable as manual data labeling can be a costly process.

A trained network can be further trained on a specific and labeled task and then benefit from its previous training on unstructured data. This reuse of learned information by first training for general capability and then fine-tuning for a specific capability is called *transfer learning*. For example, a language model can be *pre-trained* on unlabeled text to predict the next word by showing many examples of word sequences with the next word in the sequence removed. This requires no manual labeling. The pre-trained model can then be *fine-tuned* (further trained) by showing examples of a specific task, e.g., estimating the sentiment of a text. In the fine-tuning phase, fewer examples are needed than if a machine learning model was to be trained from the beginning for a specific task, such as sentiment classification. This is because the language model starts learning the new task from a position of already “knowing” a lot about texts from having been trained extensively on generating the next word. Transfer learning has been used successfully for many types of data. In many cases, a pre-trained model can be fine-tuned for several different tasks.

Another property of large language models⁶ arising from training to predict the next word is the possibility for users to *prompt* the model. A prompt is the user text input to a language model. When interacting with ChatGPT and similar systems, the prompt can be a question or a request. By providing more information within the prompt, the user can often acquire more appropriate answers from the model. This steers the model towards utilizing contextual information related to the prompt. The skill of writing prompts to obtain relevant answers is often called *prompt engineering*. This ability to utilize *prompts* to provide online training examples is often called *in-context learning*.

In essence, *in-context learning* steers the model towards associations that are helpful to give a correct or satisfying answer to the question being posed, thus yielding better results than when given no examples [23]. However, *in-context learning* is not a learning strategy, as the model being used for question-answering does not update its weights. Due to this, the model does not retain any new information from the input-output association. When performing *in-context learning* on larger models, the emergent capabilities can convince the user that the model contains “knowledge” or can perform “reasoning”, but there is currently sparse evidence for such conclusions[43].

⁶A common term is *large language models*, but what is large today will likely not be that in a few years.

As models have increased in size, such as the Generative pre-trained transformers (GPT) models [10, 46, 53, 54], new capabilities, often referred to as *emergent capabilities* have been observed. Here *emergent* refers to capabilities that a model was not expected to exhibit, based on its implementation. Rather, these capabilities were uncovered with the increased amount of training data and model size [72].

Transfer learning and emergent capabilities can be observed in multimodal models. For instance, what is learned by a language model can be leveraged when combined with an image model. Transfer learning and emergent capabilities make it difficult to predict the capability horizon for deep neural networks.

2.2 Representations and Embeddings

In computer science, mathematics is used to represent objects. For instance, letters can be represented as natural numbers, and matrices of real numbers can be used to represent images. Therefore, a matrix of real numbers can be used as an image's *representation*. *Embeddings* can be considered representations with additional properties. In machine learning for text, word embeddings (usually realized as real numbers vectors) represent words. If learned successfully, the embedding vectors reflect similarities between words so that embeddings of similar words appear close to each other in a vector space. Embeddings are often vectors of real numbers but can also be arrays or matrices of any dimension.

The objective for some neural network models is to produce embeddings for the objects of interest, e.g., words or tokens, parts of an image, a part of a sound wave, or an image frame in a video. These embeddings can be used in other models, often for several purposes. Other models produce embeddings as a consequence of achieving something else primarily. However, such embeddings can also frequently be used elsewhere.

In many multimodal models, embeddings of several different modalities are used. In some of the models, the training of the embeddings is done in a fashion that connects them over modalities. Such embeddings allow for comparisons between modalities and can be said to occupy a *joint embedding space*. A joint embedding space combines the textual and visual embeddings into a higher dimensional space. If we allow textual embedding vectors of dimensionality m and visual embedding vectors of dimensionality n , the resulting joint embedding space will have a dimensionality of either mn or $m+n$ depending on the joint strategy. The joint embedding space is also often referred to as a joint or latent representation space. Two examples of models that produce joint embedding spaces are CLIP [55] (Section 3.1) and ImageBind [28] (Section 6.2).

2.3 Encoders and Decoders

The *encoder-decoder* architecture has many uses, and it can be implemented in different ways depending on the application. The encoder and the decoder are usually implemented as separate networks in neural network models. See Figure 1 for an overview. The encoder takes a representation (R) as input and produces an encoding (E , an embedding), which is usually smaller, i.e., has a lower dimensionality. The decoder decodes the encoding, creating a new representation (R'). A typical use case is translation from a source language to a target language, where the representation of the source language text is converted to a representation of the target language text. The encoder and decoder are trained in tandem.

In an *autoencoder* the goal is for the new representation (R') to be as similar to the original representation (R) as possible while requiring the embedding (E) to have fewer dimensions than the representation (R), and thereby requiring less memory. The embedding preserves much of the information from the original representation while occupying a smaller space, often called a *latent space*, which allows for faster computations.

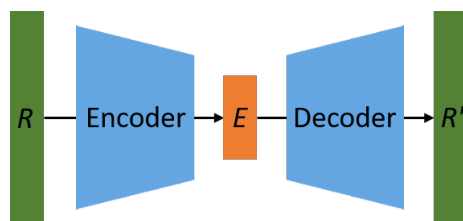


Figure 1: Encoder-decoder. A representation R is encoded into a lower-dimensional embedding E . Starting from the embedding, the decoder can create a new representation R' . In the case of an autoencoder, the goal is for R' to be as similar to R as possible.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) were developed for the processing of images. They are based on the mathematical operation *convolution*. Images are usually represented as matrices of real numbers upon which linear algebra operations are performed. CNNs reduce the complexity and size of an image by calculating a weighted average of a small “window” of the image, i.e., an image patch. The window is slid over the image in overlapping steps, and a new weighted average is calculated for each step. When the window has moved over the entire image, the collected averages comprise a smaller image that summarizes the original image. This process is repeated several times on smaller and smaller images until an image of a specific size is reached. That image is used as a representation. Multiple tasks can be achieved by using such representations, such as the categorization of the content of the image. CNNs can be applied to any data that can be visualized as an image.

CNNs have been developed in many directions and have been used for, among other things, image recognition and classification, denoising of images, and image segmentation. They have also been used as parts of more complicated architectures. CNNs can be used as both the encoder and the decoder. One of the most well-known networks of this kind is the U-Net [58], which has been used for, among other things, image segmentation, which is the task of detecting which parts of an image constitute separate objects.

CNNs are also often used in the implementation of both of the two parts of a Generative Adversarial Neural network – *GAN* [29]: the generator and the discriminator. GANs are most well-known for being used to generate realistic images of people that do not exist,⁷ but they can be used to generate other types of images as well. The two parts of the GAN learn by “competing” with each other. The generator tries to generate images that look as realistic as possible, while the discriminator tries to decide whether the generated images are authentic or not. Some multimodal models build upon GANs, but we do not discuss these further in this report.

CNNs are used in many multimodal models as they are one of the architectures suitable for image processing. Diffusion Models, discussed in the next section (2.5), are usually implemented using CNNs and are also used in multimodal models.

2.5 Diffusion Models

Diffusion models [66] are based on the physics phenomenon *diffusion*. An example of this is how water-soluble red paint diffuses in a glass of water – it is slowly mixed with the water, resulting in a glass of water with a faint red tinge. The process can be described mathematically as a series of applications of a probability distribution to the water particles, for instance, a Gaussian distribution. Each application of the probability distribution corresponds to small random movements of the particles in the water.

Diffusion models learn to reverse a diffusion process [66]. Figure 2 gives an overview of how the model is trained and used. A set of images are diffused one by one by iteratively

⁷<https://thispersondoesnotexist.com/> (Visited 2023-12-06)

applying noise according to a probability function q until each image is reduced to noise.⁸ A deep neural network learns a generation function p from the reversal of each step of the diffusion of each image.⁹ Each application of q gives a training example for the network p , which learns how to estimate and remove the noise applied by q . For instance, the deep neural network p can be a U-Net (Section 2.4).

After training, the generation function can be applied to noise of the same type as the diffusion result and generates an image after iterative applications. The generated image belongs to the same distribution of images as the original training images. Indeed, if we give the exact noise after diffusion for one of the original images, the generation process returns an image very close to the original image.

Albeit slow, diffusion models can be used for tasks such as generating or denoising images. In Section 3.2, we describe the multimodal model Stable Diffusion [57], which is more efficient than the diffusion model of [66]. Stable Diffusion achieves this by using an encoder to create embeddings of lower dimensionality that are diffused using less computational resources. The result of the generation function is an embedding passed through a decoder to generate the final image.

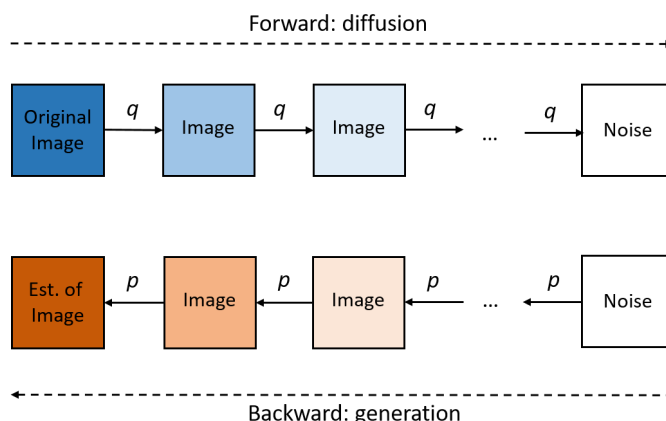


Figure 2: Diffusion Model [66]. During the application of diffusion (q) to a set of images, the generation function (p) is learned. Given the noise of the proper distribution as input, p can be used to estimate an image in a distribution containing the original set of images.

2.6 The Transformer

The transformer network architecture [71], often just referred to as “transformer”, was developed initially for translation between natural languages. Transformers have since been used for a wide range of language technology tasks [21], and generative text models such as ChatGPT¹⁰ and Gemini¹¹. Transformers for text processing have previously been described in [35]. The transformer network can be used for any data represented as a sequence of tokens if meaningful embeddings can be created for them. An example is the vision transformer for images that is described in Section 2.7. Transformers are used in many multimodal models.

Figure 3 shows an overview of translation between natural languages using the transformer, as described by Vaswani et al. [71]. All tokens are represented by embedding vectors.¹² The embeddings of the words in the source language are input into an encoder that

⁸The noise applied is Gaussian noise, and the final noisy image also follows this distribution.

⁹If the diffusion step is small enough (adding a tiny amount of noise in each step), the generation function p has the same probability distribution (Gaussian distribution) as the diffusion function.

¹⁰<https://chat.openai.com/> (Visited 2024-02-09)

¹¹<https://gemini.google.com/> (Visited 2024-02-09)

¹²Here, a token can be a part of a word, but it is easier to think of them as words while trying to understand the mechanisms of the transformer.

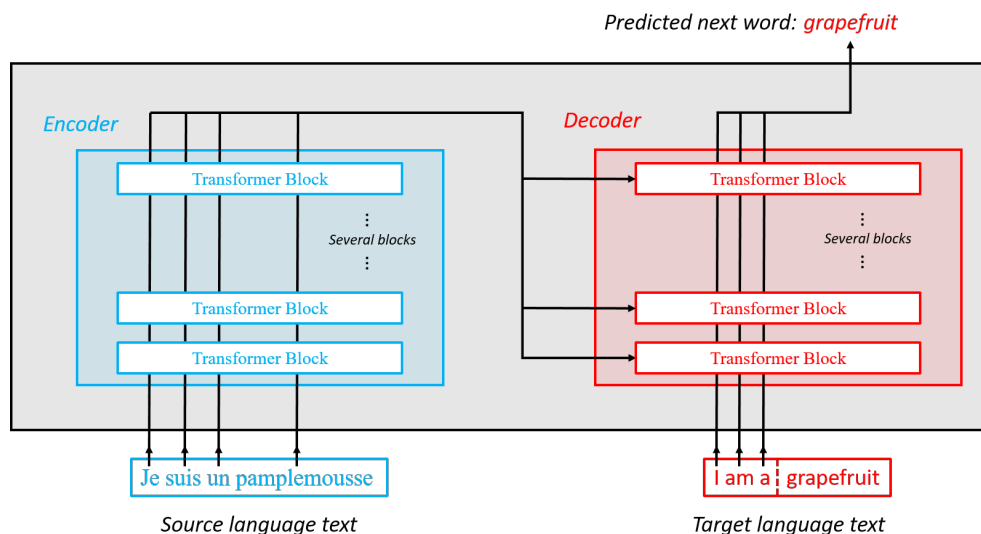


Figure 3: An overview of translation using two transformer networks: an encoder and a decoder.

recalculates them through several blocks. The encoder outputs new embeddings, one for each word. There is no reduction in the number of embeddings or sizes, as with the encoders in Section 2.3.

The translation is achieved by producing one token in the target language at a time. If tokens in the translation have already been produced, these embeddings are inputted into the decoder. Each decoder block recalculates the embeddings for the target language tokens based on the tokens on the previous transformer block and the encoder's final token embeddings of the source language. The final target language embeddings are used to predict the next token in the translation.

An *attention* mechanism is used to recalculate the embeddings in each transformer block. The attention mechanism is not described in detail in this report. However, in essence, the attention mechanism is based on the idea that to calculate an embedding for a token, the embeddings of the tokens around it must be considered, or *paid attention* to. During training, the attention mechanism learns how much weight to give to each of the tokens from the previous transformer block while considering which those tokens are. Figure 4 shows the encoder and decoder as separate networks. In the encoder, attention is paid to tokens before and after the current token, while in the decoder, attention is only paid to the tokens before.

The encoder and decoder networks can be used jointly as in language translation or can be used separately for a range of various tasks. Different variations are used in different multimodal models presented in the following chapters. The encoder can be trained to create embeddings of high quality for each token and can, in turn, be used for many applications. The most well-known encoder network is Bidirectional Encoder Representations from Transformers (BERT) [22], which adds an extra token embedding specifically intended for classification (called CLS).

The decoder helps predict the next token in a sequence. The most well-known decoder networks are the GPT models [10, 46, 53, 54], which form the basis for ChatGPT, and Pathways Language Model 2 (PaLM 2) [16].

In an encoder-decoder network for each transformer block, the decoder pays attention to the token embeddings from the encoder (Figure 3). This is called *cross-attention*. Cross-attention is used in some multimodal models to combine two different modalities by informing the recalculation of the embeddings for one modality by the embeddings of another.

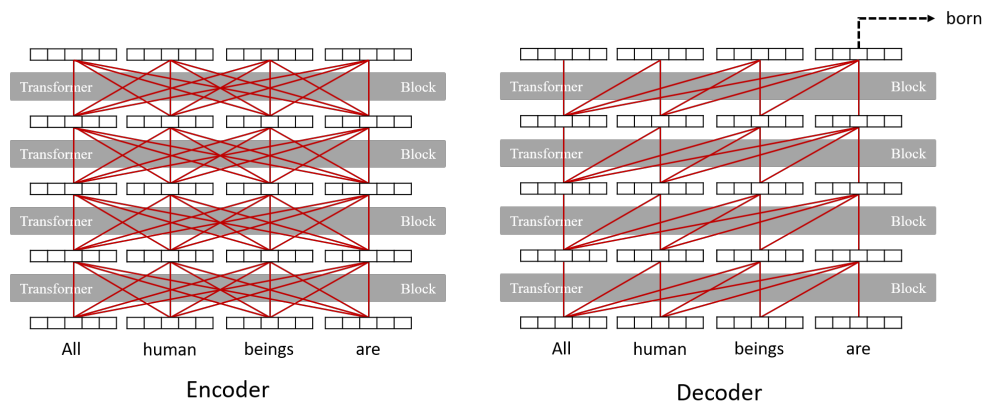


Figure 4: *Left: the transformer encoder.* Through several layers of blocks (bottom to top), embeddings for each word are calculated by taking the embeddings from the previous block into account through the attention mechanism. The top embeddings can be used for many applications. *Right: the transformer decoder.* Similarly to the encoder, new embeddings for each block are calculated using the attention mechanism based on the previous block. Here, however, for each word, attention is only paid to the words preceding that word. The top right embedding can be used to predict the next word.

2.7 Vision Transformer

The Vision Transformer (ViT) [24] resembles a transformer [71] used in the text domain but takes image tokens as input. Image tokens are created by dividing an image into fixed-size patches that do not overlap.¹³

These patches are projected onto one-dimensional vector embeddings by a neural network that learns a linear projection during the training of the full ViT. Learnable position embeddings are then added to the patch embeddings, and the resulting series of vectors are given as input to the ViT encoder. The transformer architecture provides good scalability and allows training high-performing ViT models on large datasets containing hundreds of millions of images [24].

Originally, ViT was used for image classification by means of a classification token, like in BERT [21]. The transformer architecture provides good scalability and makes it possible to train high-performing ViT models on large datasets containing hundreds of millions of images [24]. ViT has also been extended to videos [3]. Many multimodal models use a ViT or related networks as an essential component.

Comparison to Convolutional Models

The ViT encoder is a standard Transformer encoder that does not rely on the type of convolutions known from traditional CNN architectures, which are commonly used when analyzing images. CNNs' convolution model captures "local" relationships between pixels since they rely on kernel filters. ViT is based on the attention mechanism, which models relationships between embeddings representing different image patches, which can be far from each other. This can be said to model "global" relationships between the patches.

Both perspectives are useful: images have both local properties, such as how the color of pixels close to each other affects one another, and global properties that can only be modeled by taking larger portions of an image into account, such as how a person's head and leg may both be part of a person's body.

¹³A learnable classification token (CLS) that represents the entire image is also added to the end of the sequence of patch embeddings.

A repeated application of convolution can make a CNN able to model global properties, but there is no guarantee that all such properties can be modeled. A ViT with small enough image patches can model local properties, but the computational cost easily becomes high. A compromise can be to create embeddings of image patches for a transformer-based model using a CNN that takes the local properties into account.

3 Image – Text

This chapter intends to inspect and elaborate on different text and image multimodal models since the success and broad impact of multimodal models in recent years have primarily been driven, at least initially, by the development of models that can handle both images and texts. Much effort is being put into this rapidly moving field of research, and new image-text models with impressive capabilities are constantly being released. Many of the models that combine image and text utilize transformer-based architectures. Transformers were previously mainly associated with language modeling in the text domain alone but can, in fact, be applied to images as well (Section 2.7), which makes it possible to build models that, for instance, generate text based on a combination of image and text input.

The Contrastive Language-Image Pre-training (CLIP) model (Section 3.1) attracted much attention in 2021 due to its ability to learn meaningful representations (also referred to as embeddings) of images and texts. Since CLIP can produce vector representations for any image and text in a joint embedding space, it is possible to decide how well an arbitrary image matches an arbitrary text by comparing their representations. CLIP embeddings enable many downstream (unseen) tasks, such as searching through large image databases to find images that match a text description or detecting specific objects in images based on a text prompt. The open-vocabulary object detector Detic (Section 3.4) utilizes CLIP embeddings and can be used to detect objects given a text prompt.

The image-text models that have probably had the greatest impact among the general public so far are those that are able to generate images based on text descriptions. Text-to-image functionality is easily accessible through popular services such as Midjourney,¹⁴ Microsoft Bing,¹⁵ and ChatGPT.¹⁶ Most of these services are built using methods similar to and/or inspired by diffusion models (Section 2.5) and the freely available model Stable Diffusion (Section 3.2) that can be used to generate and modify images in different ways based on text input (and sometimes also image input).

As mentioned, some models can be used to process images and texts in order to generate new text. PaLI (Section 3.3) is one example of such a model. The model can perform image captioning, image classification, and visual question answering (VQA), amongst other things.

Further we present two multimodal segmentation models, i.e., models that “highlight” objects and other interesting image regions with *masks*. The “Segment Anything Model” (SAM) described in Section 3.5 is able to produce masks based on visual prompts, i.e., starting points or bounding box coordinates, but also simple text prompts. “Segment Everything Everywhere with Multi-modal prompts all at once” (SEEM) that is described in Section 3.6 encodes all input prompts into a joint representation space. SEEM can segment images based on visual prompts, but also on text and audio prompts.

3.1 CLIP

The CLIP (Contrastive Language-Image Pre-training) model was released at the beginning of 2021 and constitutes a breakthrough in multimodal machine learning [55]. CLIP is capable of associating images and their corresponding textual descriptions. Hence, in its basic form, the model can be applied to a wide range of visual classification tasks since it is able to determine how well an arbitrary image matches a text prompt describing a specific visual concept. The fact that CLIP is trained to handle arbitrary text descriptions makes it possible to introduce new visual concepts at a later stage and generalize to unseen data. There are no clear rules regarding how a text prompt should be formulated, although prompts that help specify that

¹⁴<https://www.midjourney.com/> (Visited 2023-11-03)

¹⁵<https://www.bing.com/create> (Visited 2023-11-03)

¹⁶<https://chat.openai.com/> (Visited 2023-11-03)

the text is about the *content* of the image (e.g., “a photo of a military tank”) tend to work well [55].

In practice, CLIP utilizes an *image encoder* and a *text encoder* to extract feature representations of each modality (i.e., features for each input image and features for each text prompt) [55]. The text encoder is a slightly modified version of the well-known transformer-based language model GPT-2 [54], while the image encoder is based on the Vision Transformer (ViT) [24] (Section 2.7).

The image and text features obtained from the CLIP encoders are projected into a joint multimodal embedding space [55]. This operation produces embedding vectors of fixed size, which makes it possible to utilize the cosine similarity metric to measure the distance (i.e., the similarity) between each image vector and text vector. The metric ranges from 0 to 1 (inclusive), where a larger value indicates a smaller distance and, therefore, a better matching image-text pair. A perfect image-text score of 1 is nearly impossible to get. Figure 5 shows an example where CLIP is used to classify images of tanks and fighter jets.

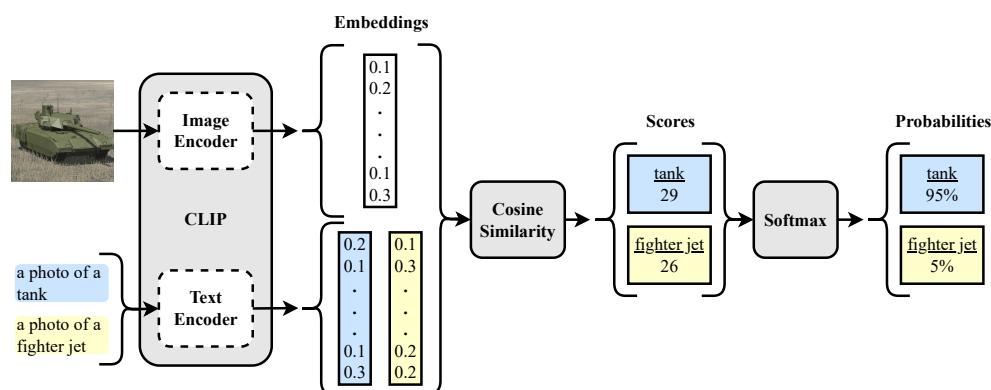


Figure 5: Classification of tanks and fighter jets using CLIP embeddings. In this binary classification example, the distance (cosine similarity) is measured between the tank image (white vector) and text prompt “a photo of a tank” (blue vector), and between the tank image (white vector) and text prompt “a photo of a fighter jet” (yellow vector). The resulting cosine similarity values are multiplied by a factor 100 to obtain scores that are fed to a Softmax function computing probabilities.

For CLIP to successfully embed matching image-text pairs close to each other, the model has been trained using a symmetric cross-entropy loss function that encourages the model to maximize the cosine similarity of matching image-text pairs and minimize the cosine similarity of non-matching image-text pairs [55]. Hence, by contrasting positive (matching) and negative (non-matching) pairs, CLIP has learned meaningful representations (embeddings) such that matching pairs will be close to each other in the representation space, while non-matching pairs will be far apart. This type of representation learning is known as *contrastive learning*.

The official CLIP model has been trained on 400 million image-text pairs collected *directly* from the internet, thus eliminating the need to label training data manually. For comparison, the manually labeled computer visions dataset ImageNet only contains approximately 14 million images [19]. The large amount of training data is one of the reasons why CLIP generalizes well to unseen test data. In numerous benchmarks, CLIP achieves similar performance as state-of-the-art models trained in a supervised manner for the specific task being evaluated, unlike CLIP, which is evaluated in a *zero-shot setting* [55]. In addition to image classification, CLIP embeddings can be used in many other applications such as human identity verification [70], action recognition [55], and object detection [77].

3.2 Stable Diffusion

Stable Diffusion [57] is a flexible model that is best known for its capability for quality image generation from text. It is based on diffusion models (Section 2.5). Aside from text, its image generation can be conditioned on different types of data. This makes it possible to generate images from any combination of text, images, and parts of images. The latter allows the method to fill in parts of an image that have been removed, so-called *inpainting*, and extend an image beyond its original size. In both cases, what should be added can be described in text. Further, Stable Diffusion can be used to increase the resolution of an image.

Stable Diffusion is more efficient than the diffusion model because it does the calculations in a latent space of fewer dimensions than the original image space [66]. This is achieved by separately training an encoder-decoder pair used before and after the diffusion and generation (Figure 6). The encoder reduces the size of the original image x to a smaller representation, z , that retains enough information about the image.

The forward diffusion and backward generation are done incrementally as in the original diffusion model, but in Figure 6, the U-Net responsible for the backward steps is only depicted once. The generation is conditioned on added data, such as text and images, encoded using encoders specific to each data type. To achieve the flexible conditioning of Stable Diffusion, cross-attention (Section 2.6) is used in the layers of the U-Net. For some tasks, such as increasing the resolution of an image, the encoded data can be concatenated to the image's latent space representation z .

Stable Diffusion partially owes its success to LAION-5B [63], which is a dataset containing approximately 5.85 billion CLIP 3.1 filtered image-text pairs, which Stable Diffusion has been trained on. The main contributions of Stable Diffusion [57] (latent diffusion and conditioning) form the foundation of most successful image-to-text systems available today. One example is Stable Diffusion XL [51], which increases the model size by leveraging a larger cross-attention context and a secondary text decoder. It exploits a post-process sampling image-to-image refinement technique, increasing the model's quality and performance. They are also used in several models for video generation from text (Chapter 5).

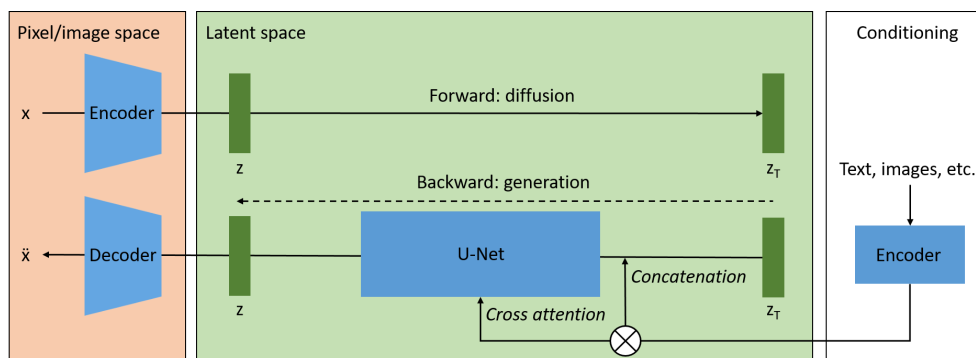


Figure 6: Stable Diffusion. Image after [57] (simplified).

3.3 PaLI and PaLI-X

The input to the Pathways Language and Image model (PaLI) [14] is an image and a text. In response, it generates a text. This makes it suitable for performing many image-language tasks, such as image captioning, visual question answering, i.e., answering questions about an image, and zero-shot image classification.¹⁷ PaLI achieves state-of-the-art performance on these tasks and can perform them in many languages as the model is pre-trained in over 100 languages.

¹⁷The transformer decoder is used to score texts describing the classes of interest.

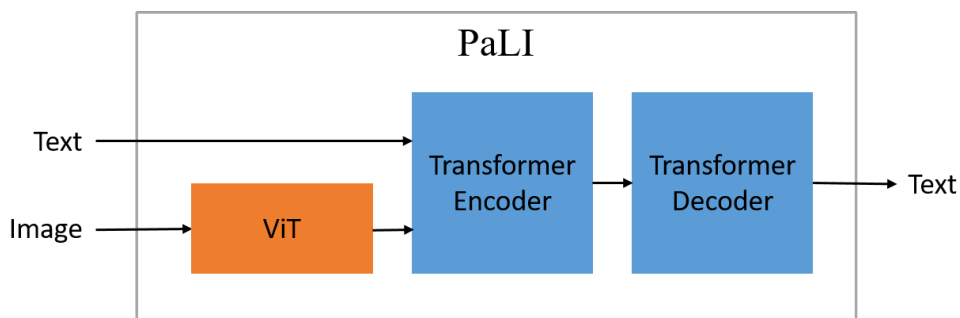


Figure 7: PaLI. Images are encoded into embeddings by a ViT model and input into the transformer beside the text tokens. Image after [14].

The PaLI architecture consists of a visual and a language component (Figure 7). The language component is a Transformer model with both an encoder and a decoder, like in [71]. A vision transformer, ViT (Section 2.7), is used as the visual component. Its output of image token embeddings is added to the text tokens and used as input to the transformer encoder.

The language and vision components are pre-trained separately and can thus be scaled independently. The combined model is then pre-trained on a large multilingual image-language dataset constructed by the authors, using several different pre-training tasks with the goal of capturing many capabilities. During pre-training, only the language component is updated. The model is fine-tuned on specific data for the different tasks (zero shot tasks obviously omitted).

PaLI-X [13] has a similar architecture as PaLI. The ViT component is scaled up to $22B$ parameters and handles video by individually processing a series of frames. The encoder-decoder part is scaled up to $32B$ parameters. This larger model outperforms PaLI and other models on a wide range of tasks.

3.4 Detic

Detic (Detector with Image Classes) [77] is a model that can be used to detect objects in images. Object detection involves two problems: *localization* and *classification*. Localization aims to predict bounding boxes that indicate where objects appear in an image (i.e., finding objects), while classification seeks to determine the object types (i.e., naming the objects). This is exemplified in Figure 8.

Object detectors are typically trained on images paired with ground truth bounding box labels and class labels and, therefore, cannot detect objects belonging to novel classes not present in the training dataset. For instance, a detector trained to detect cars and persons cannot be used to detect assault rifles, at least not without being fine-tuned on data representing the class. Although it is possible to fine-tune detectors to handle new classes, it can only sometimes be done correctly since the amount of object detection training data is often limited. Detic, however, can detect many objects belonging to novel classes without being fine-tuned. This has been achieved by relying on embeddings from the multimodal CLIP model (Section 3.1) and incorporating additional data from an image classification dataset during training [77].

Specifically, the name of an arbitrary class (e.g., “assault rifle”) can be encoded using CLIP to obtain an embedding. Detic uses CLIP embeddings of class names as classification weights, making it possible to detect novel objects outside the training vocabulary [77]. However, CLIP embeddings alone are not sufficient to achieve satisfying performance. The authors behind Detic realized that the *localization* and *classification* problems can be decoupled. Many object detectors are able to *localize* objects belonging to novel classes. The authors, therefore, focused on improving *classification* performance, and thereby detection



Figure 8: Object detection using a model trained to detect cars and persons. Each detected object has been marked with a rectangular bounding box and assigned a class label (class name) and a confidence score. The original image comes from the HERIDAL dataset [7].

performance, by expanding the training dataset with images from the ImageNet [19] classification dataset that only provides a single class label per image and no bounding box labels.

The benefit of using classification datasets, such as ImageNet, is that they are easier to label and, therefore, significantly larger than object detection datasets that often require multiple bounding box labels and corresponding class labels for each image. ImageNet contains 14 million images and provides labels for 21 thousand different classes. During the training of Detic, the class label of each ImageNet image was assigned to the largest-sized object proposal (i.e., the proposal with the largest “bounding box”) obtained when feeding Detic with the image [77]. A traditional object detection dataset was used to train Detic’s localization and classification abilities. In contrast, the ImageNet classification dataset improved the classification ability by training the object proposal region features to better match the CLIP embeddings [77].

3.5 SAM

Segment Anything Model (SAM) [36] aims to be a foundational model that does not require fine-tuning for image segmentation using masks. Masks are the segments of an image corresponding to a specific and separate object of interest.

SAM solves the generalized segmentation problem, which means it can take a downstream task image mask, such as a mask of a cat, and then be able to segment cats in all future images. Masks can be indicated in images manually by clicks or as drawn shapes together with textual information relating to the object. SAM can output multiple plausible masks for the desired visual and textual prompts.

The model architecture is presented in Figure 9. The image encoder is based on a Masked AutoEncoder (MAE), which, like in CLIP (Section 3.1), is a Vision Transformer (ViT) (Section 2.7). Input, or dense masks, are calculated using a standard convolutional internal channel (the rhombi in Figure 9) and layer mapping (the circle-cross in the same figure). This matches the image embedding (Section 2.2), which means that the mask summed element-wise with the image embedding. As such, masks are not “prompts” but are fed back into the model through this mapping. The mask decoder is a transformer decoder that predicts the final mask and its inputs are the various sparse mask indicators, such as text input prompts and point cloud prompts, but also the dense prompts: mask convolutions, points, lines, and boxes.

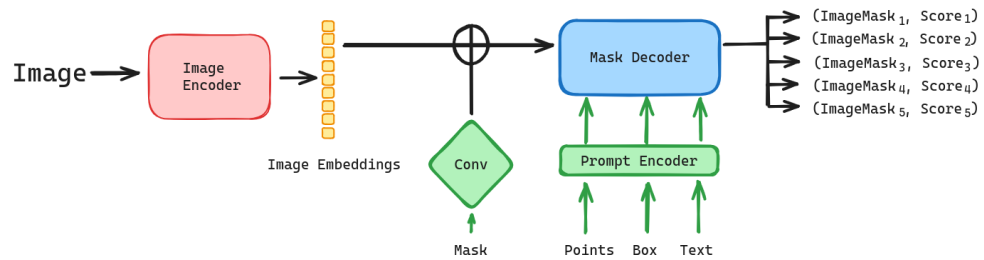


Figure 9: The image encoder produces embeddings which can be efficiently queried using input prompts, downstream tasks are specialized via convoluted masks.

The authors of [36] also present a dataset (SA-1B) of 1 billion masks together with 1 million corresponding images, making it two magnitude orders larger than Google OpenImages v5 and four magnitude orders larger than COCO with regard to labeled data. SA-1B is analyzed from a diversification and fairness perspective and determined to be diverse.

3.6 SEEM

Segment Everything Everywhere with Multi-modal prompts all at once (SEEM) [78] is a segmentation model that predates but is similar to ImageBind (Section 6.2) since it relies on a joint embedding space (Section 2.2) for images and texts to support *semantic segmentation*, i.e., regular segmentation coupled with classification. Since SEEM utilizes a joint embedding space, the prompt encoder architecture can be easily extended to custom modalities. In contrast to SAM (Section 3.5), SEEM can operate on not only simple text prompts or point clouds but also video and audio¹⁸ prompts and reference point clouds. The efficacy of SEEM is demonstrated in [39] through a comprehensive segmentation study over different modalities, number of passes, and detected masked classes. It should be noted that masks are ostensibly point clouds when input iteratively since these gets encoded to a visual prompt.

SEEM can also process multiple rounds of interactions using its prompt decoder since it encodes so-called *memory prompts*, which can be seen in Figure 10. Since reference point clouds are parts of one image where the sought mask captures a class, which should also be

¹⁸SEEM utilizes Whisper (Section 4.1) to convert audio into text prompts.

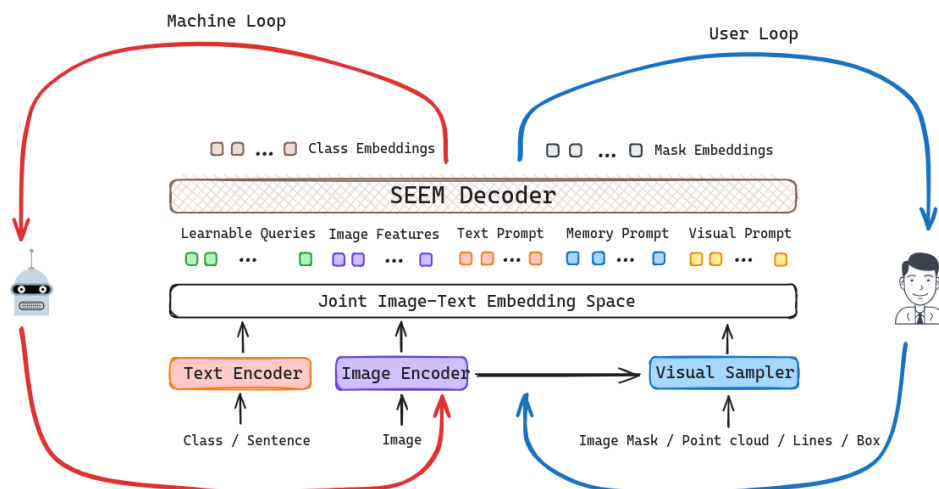


Figure 10: SEEM model where the machine loop “remembers” previous masks and the user loop “refines” upon kept information through memory prompts.

masked in other images, this can be used for downstream task specialization. An example would be the cat segmenter from Section 3.5 where the mask is refined iteratively to be specialized on a specific type of cat, or perhaps just some part of the cat, to produce the final semantic segmenter. SEEM is also “semantically aware”, meaning that it produces labels to predicted masks. The Semantic-SAM model [39], essentially SEEM trained on the SA-1B images with masks from SAM [36], can perform segmentation at an even finer granularity but requires more inference time and powerful hardware.

4 Sound – Text

Sound as a modality is comparable to images, since we can convert audio to an image by looking at its spectrogram (Section 4.1). However, sound has a temporal dimension, and this complicates audio processing. This complexity is partly because converting sound to a spectrogram is an imprecise procedure, but also due to the intimate connection between how humans perceive sound and how they perceive time. To achieve “harmony” in music one needs to consider the fact that specific mathematical ratios sound pleasant to the human brain while others sound unpleasant, a phenomenon also closely tied to sound propagation in time. This phenomenon has historically been difficult for machines to grasp, so music generation has mostly been a human endeavor. However, with the latest wave of machine learning models, some interesting examples of music-generating machine models, such as MusicLM [1] and Stable Audio¹⁹, have emerged.

Although advancements have been made in many sound-related areas, this chapter is limited to models that work with speech and text data. Speech, in particular, is a very intricate thing to model. Compared to text, speech as a modality is less effective regarding record keeping and data storage. As such, the amount of available speech data is smaller than that of text. This has also led to speech being the least researched of the two. But comparing text and speech in this way is somewhat unfair since speech is more than spoken text. Humans can recognize complex meanings in utterances, such as sarcasm or emotion, that the corresponding text would not convey. Speech can further be used to signal social context, such as formality, in a way that is significantly more difficult to achieve in written text. Even grammar and morphology can differ between the modalities [50]. These differences tell us that there is a lot to gain from building systems that can handle both modalities jointly, especially when it comes to understanding language and how humans communicate. These differences also tell us that building such a system will be expensive.

A model capable of *automatic speech recognition* (ASR) is a model that can automatically transcribe speech into its corresponding text. One noteworthy ASR model is Whisper (Section 4.1). Whisper explicitly tackles the problem with little available speech data paired with transcripts, resulting in state-of-the-art performance in robust speech processing.

The amount of high-quality speech data becomes even smaller when considering languages other than English. Many speech models thus focus predominantly on English speech. The translation model SeamlessM4T (Section 4.2) departs from this trend, with its capability to translate between multiple modalities in up to 100 different languages.

4.1 Whisper

Whisper [52] is an automatic speech recognition (ASR) system developed by OpenAI. The model is trained on 680k hours of multilingual data collected from the web and tested in various experiments. In most cases, Whisper exceeded or matched the performance of state-of-the-art models in experiments such as, e.g., zero-shot robustness over a wide set of speech processing datasets, zero-shot $X \rightarrow$ English translation means translation from an unspecified language into English.²⁰ and robustness to additive noise. Experiments also showed that Whisper’s performance was close to that of professional human transcribers.

The data used to train Whisper was not perfect, i.e., some of the audio, e.g., contained noise and some of the transcriptions were misaligned.²¹ However, the amount of data available for training was greatly increased by relaxing the requirement of a high-quality dataset. This trade-off, which is understudied in speech recognition, significantly improves the robust-

¹⁹<https://www.stableaudio.com/> (Visited 2023-12-15)

²⁰ $X \rightarrow$ English translation Similarly, $English \rightarrow X$ translation indicates translation from English into an unspecified language.

²¹The audio data and transcription is misaligned when a part of the audio is matched to the wrong part of the text.

ness and generalization of models in other domains [37, 45]. The dataset used by Whisper consisted of 680k hours of audio and their corresponding transcriptions. It is a multilingual dataset, and the distribution of the dataset is shown in Table 2. The non-English data enables Whisper to transcribe speech in a non-English language and translate speech from another language to English. Whisper, thereby, is both a multimodal and a multitasking model while also being multilingual.

Table 2: Multilingual dataset distribution. 438k hours of English speech with English transcriptions, 125k hours of non-English speech but with English transcriptions, 117k hours of data in 98 languages other than English (99 total).

Distribution	Size (hours)
English Speech & Transcription	438 000
Non-Eng. Speech, Eng. Transcription	125 000
Non-Eng. Speech & Transcription	117 000

Whisper, an encoder-decoder Transformer (Section 2.6), feeds audio to the encoder while the decoder outputs text. An illustration of the model architecture is shown in Figure 11. Audio data in Whisper is represented using log Mel-spectrograms. To get the log Mel-spectrogram of sound, one starts with standard raw audio, which is amplitude plotted against time, and then one takes the Fourier transform of a small-time window. This yields frequency plotted against amplitude. Combining multiple such frequency plots, one gets a spectrogram, a picture where the pixels represent the audio data. The x -position of the spectrogram gives you the time, the y -position gives you the frequency, and the color intensity of the pixel gives you the amplitude. By rescaling the frequencies to the mel scale,²² and the amplitude to a logarithmic scale (also known as the decibel scale), you get a log Mel-spectrograms (Figure 11). Since the resulting log Mel-spectrogram is an image, employing CNNs comes naturally.

The goal for developing Whisper was to create a robust speech recognition system that performs well in zero-shot scenarios, reducing the need for dataset-specific fine-tuning. This implies that there can be no overlap between the dataset Whisper is trained on and the datasets it is tested on.

A standard metric used to measure ASR performance is the *word error rate* (WER). It measures the distance between two text-strings. An issue with WER is that transcriptions that could be considered correct by humans can still have a large WER due to minor formatting differences. All text-strings are normalized before WER calculations to mitigate this issue and minimize the penalization of non-semantic differences WER calculations (Table 3). To measure translation quality, the widespread performance metric bilingual evaluation understudy (BLEU) is used, which has been claimed to have high correlation with human analysis [18, 47]²³.

Table 3: Examples of text normalization.

Original text	Normalized
THIS_1 IS A NORMALIZATION EXAMPLE	this is a normalization example
How's the terrain looking today?	how is the terrain looking today

²²The Mel scale is adapted to represent better how humans perceive sound frequencies.

²³There are many problems with BLEU, see for instance [11], but for large-scale automatic evaluations it is convenient.

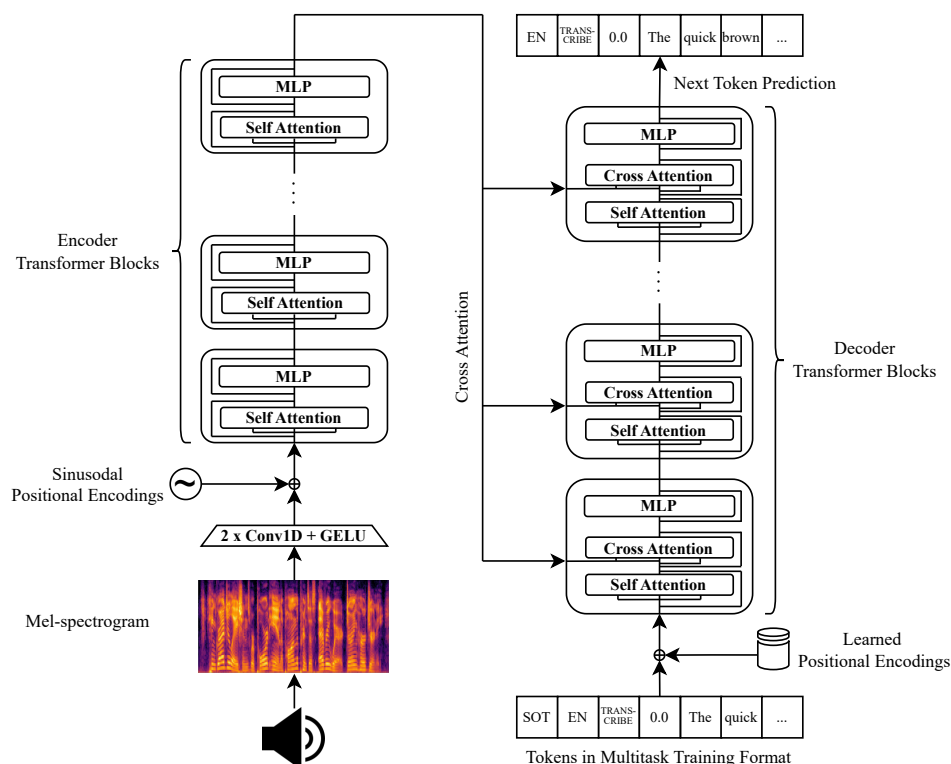


Figure 11: Illustration of Whisper model architecture. The encoder encodes the log Mel-spectrograms by a transformer network. The encoded Mel-spectrogram is decoded to tokens via the decoder transformer network. Figure inspired by Radford et al. [52].

4.1.1 Distill-Whisper

Distill-Whisper [27], uses the information distillation technique,²⁴ first described by Hinton [30], to halve the Whisper model parameter size, yet still performs within 1% of Whisper’s word-error-rate (WER), while being 5.8 times faster and robust towards noisy environment. The Distill-Whisper architecture is essentially the same as Whisper; the encoder is copied directly from Whisper’s and frozen during training. It is only in the decoder that of the “student model” (a name for the model that is under the influence of the distillation technique) we see a difference in terms of a Kullback-Liebler (KL), in essence, measuring the loss between “student” and “teacher” (regular Whisper) and a Pseudo Labelling (PL) loss, measuring the against the sound-text ground-truth loss. The final loss is calculated the sum of KL and PL given some hyperparameters.

4.2 SeamlessM4T

SeamlessM4T [64] – massively multilingual & multimodal machine translation – is a machine translation system from Meta AI that supports speech-to-speech translation (S2ST), speech-to-text translation (S2TT), text-to-speech translation (T2ST), text-to-text translation (T2TT) and automatic speech recognition (ASR) for up to 100 languages. To achieve this, SeamlessM4T is trained on 1 million hours of open speech audio data and creates a dataset, called SeamlessAlign, with automatically aligned speech translations of more than 470 000 hours.

²⁴Although incorrectly named knowledge distillation.

Before SeamlessM4T, most of the state-of-the-art models that handled speech translation tasks were *cascaded systems*, meaning systems composed of multiple subsystems progressively. A cascaded S2ST system would, for example, consist of an ASR subsystem, such as Whisper (Section 4.1), followed by a T2TT subsystem, and lastly, finalized by a text-to-speech synthesizer. The cascaded approach comes with challenges since errors from different subsystems propagate and accumulate in the final translation. Moreover, problems may arise from domain mismatches between systems,²⁵ which would negatively impact the final result. With SeamlessM4T the team at Meta AI wanted to avoid these problems by building a unified model instead while having a large language coverage.

To describe the architecture of SeamlessM4T, it is helpful to talk about different subsystems. However, as opposed to a cascaded system, these subsystems are all trained simultaneously in a unified manner. Hence, we still view them combined as a unified model. There are four different subsystems present in SeamlessM4T: w2v-BERT 2.0 [17], SeamlessM4T-NLLB, a text-to-unit (T2U) subsystem, and HiFi-GAN [38]. w2v-BERT 2.0 [17] is a conformer-based ASR-system inspired by BERT [22], SeamlessM4T-NLLB is a transformer-based T2TT system based on NLLB [68], the T2U subsystem is based on XLS-R [4]; and HiFi-GAN [38] is a unit-to-speech synthesizer (i.e., a vocoder). Each subsystem is first pre-trained individually, then are fine-tuned jointly (Figure 12).²⁶

Pre-trained Models



SeamlessM4T architecture

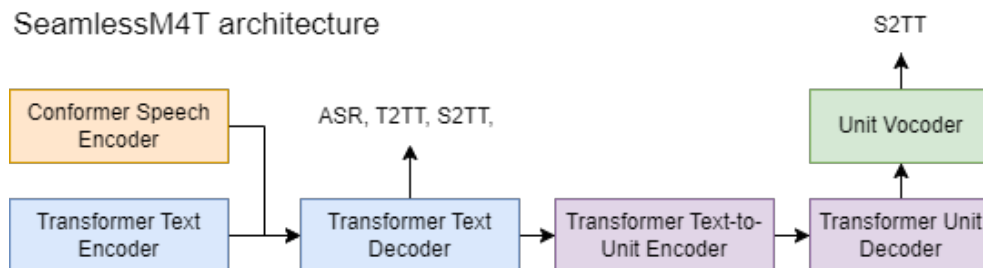


Figure 12: Illustration of the SeamlessM4T model architecture. The different parts are colored based on what pre-trained subsystem they belong to. Figure inspired by *Seamless: Multilingual Expressive and Streaming Speech Translation | Research - AI at Meta* [64].

SeamlessM4T outperforms many of the previous state-of-the-art models in S2TT and S2ST tasks. It does so while also being the first system that performs S2ST from 100 languages to English (100-eng) and from English to 35 languages (eng-35), along with high language coverage on the other modalities (ASR, S2TT, T2ST, T2TT). Combining high-quality translations and diverse language coverage makes it an interesting model.

²⁵An example of a domain mismatch would be if a subsystem trained on research articles is combined with a subsystem trained on chat logs.

²⁶The fine-tuning is done in three stages: first, the model is trained on supervised English ASR and into-English S2TT; then the model is trained on supervised non-English ASR and from-English S2TT; lastly the parameters corresponding to into-text tasks (ASR, T2TT, S2TT) are frozen, and the model is trained on supervised S2ST data.

5 Video – Text

Similar to how sentences consist of sequences of words, a video consists of a series of frames where each frame effectively is an image. One challenge in text-image modeling is inferring actions (series of events) from static images. Unlike individual images, video has a temporal dimension, enabling actions to be observed. Analogously to sound, this introduces challenges due to increased dimensionality and long-term dependencies across video frames. A model must correctly predict image pixels as a step in text-image processing. In text-video modeling, however, pixel changes in the temporal dimension also need to be predicted. Text-video modeling is a research area currently undergoing remarkable advancements, primarily leveraging breakthroughs in AI systems for text-image modeling. For instance, the model Stable Diffusion, previously covered in Section 3.2, has recently given rise to a video-based model called Stable Video.²⁷

Although we might typically expect video to be accompanied by sound, this chapter is restricted to text-video models for silent videos. X-CLIP (Section 5.1) is a framework for video-text retrieval, similar to CLIP (Section 3.1). Make-A-Video (Section 5.2) is an example of a video generation model and is in part based on diffusion models (Section 2.5). Sora (Section 5.3) is the most recent of the models reviewed in this chapter, and is distinguished by its capacity to generate comparatively long (up to one minute) video sequences.

5.1 X-CLIP

X-CLIP [44] is a multimodal information retrieval architecture. Similar to other multimodal endeavors, overall challenges in video-text retrieval lie in comparing data from different modalities and identifying similarities and dissimilarities between them. As in the case of multimodal image-text processing discussed in Chapter 3, recent video-text tasks efforts have explored contrastive learning methods. These attempts have primarily focused on comparing coarse similarities between complete videos and complete sentences or fine-grained similarities between separate video frames and individual words. Unlike previous video-text models, X-CLIP enables similarity comparisons between complete sentences and individual video frames, complete videos and separate word tokens, and video-sentence and frame-word comparisons (Figure 13).

Ma et al. [44] observe that these solutions often lack abilities to distinguish between relevant and irrelevant video-text similarities for a given retrieval query. The work of [44] aims to improve retrieval by excluding task-irrelevant information originating from any of the two modalities. To this end, [44] presents X-CLIP, a multi-grained contrastive framework for semantic video-text alignment learning, where similarity comparisons are made between video and text at different levels of granularity. As the name suggests, X-CLIP is an extension of the CLIP model (Section 3.1) and encodes word sequences and video frame sequences as CLIP word embeddings and temporally encoded ViT video frame (i.e., image) embeddings, respectively. Significant experimental improvements on video-to-text and text-to-video retrieval tasks are reported for several video-caption datasets, compared to state-of-the-art models [44].

X-CLIP has capacities for multiple contrastive video-text similarity comparisons of varying granularity, and is further capable of aggregating similarities across these granular levels. In the aggregation part, an attention scheme (Section 2.6) yields instance-level similarity scores for video-word similarity vectors, sentence-frame similarity vectors, and for fine-grained frame-word matrices. An ultimate instance-level video-text similarity score is obtained by averaging over the different calculated similarities. The score thereby holds information about multi-grained video-text relationships, as well as about the relevance of those

²⁷<https://stability.ai/stable-video> (Visited 2023-12-04)

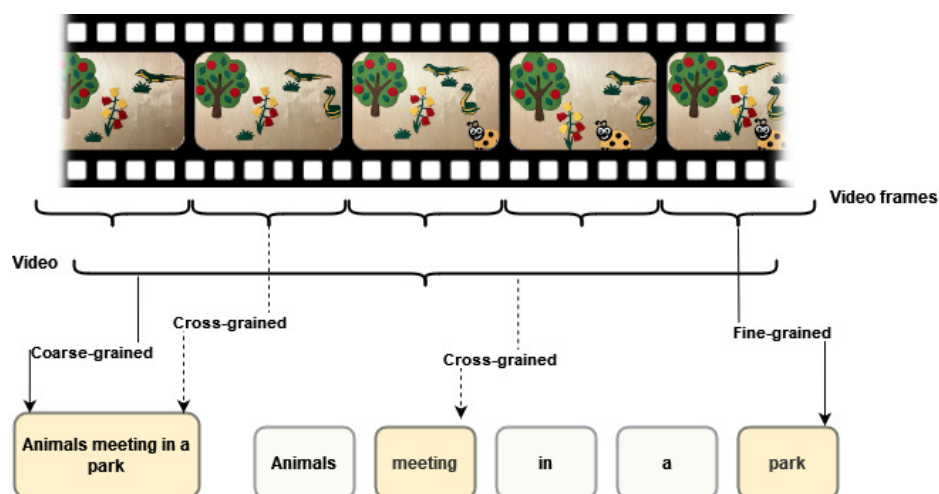


Figure 13: X-CLIP with the cross-granular similarity comparisons, varying from fine-grained to coarse-grained.

relationships relative to a retrieval query. Informally, in the X-CLIP framework, coarse features function as a relevancy filtering mechanism on fine-grained features, a strategy which apparently improves both retrieval accuracy and efficiency.

5.2 Make-A-Video

Make-A-Video [65], introduced by Meta AI in a 2022 publication, is a text-to-video diffusion based model architecture. The model is capable of generating short, high frame rate videos based on text prompts by learning semantic similarities between text and images, and learning motion patterns from viewing unlabeled video. In addition to video generation based solely on text prompts, the model also has capabilities for generating video based on static images, and for altering existing videos. Singer et al. [65] report substantial experimental improvements on benchmark datasets, compared to prior state-of-the-art methods.

The Make-A-Video model architecture is partitioned into a sequence of separate components. Prior to training of the generative model, a text-image decoder takes text-image pairs, infers intermediate CLIP image embeddings, and returns high resolution images. The model is then trained on these images, followed by modifications to the convolutional and attention layers of the model to handle extended spatiotemporal information (Figure 14). These modifications enable the model to learn image-to-image motion patterns, thereby improving frame sequence prediction. The spatiotemporal network layers are then fine-tuned on video data in an unsupervised manner to make use of video content during training without a need for labelled text-video data. Finally, a separate network for video frame-interpolation is added to increase video frame-rate, resulting in smooth video with high spatiotemporal resolution.

Despite a lack of capability for modelling actions and events that can only be inferred from text-video mappings, Make-A-Video's ability to learn from existing video data without requiring large datasets of video-text pairs is also a strength. The video content generated by Make-A-Video is short, and exhibits blurry entities and video anomalies such as depictions of legs moving in ways that actual legs do not. Still, selected model outputs are convincing and span multiple domains, from realism to fantasy. Prompting the system to generate a video of a "teddy bear painting a portrait",²⁸ apparently yields a short video sequence which unmistakably displays what was textually requested.

²⁸Quote and example from: Make-A-Video. <https://makeavideo.studio/> (Visited 2023-11-21).

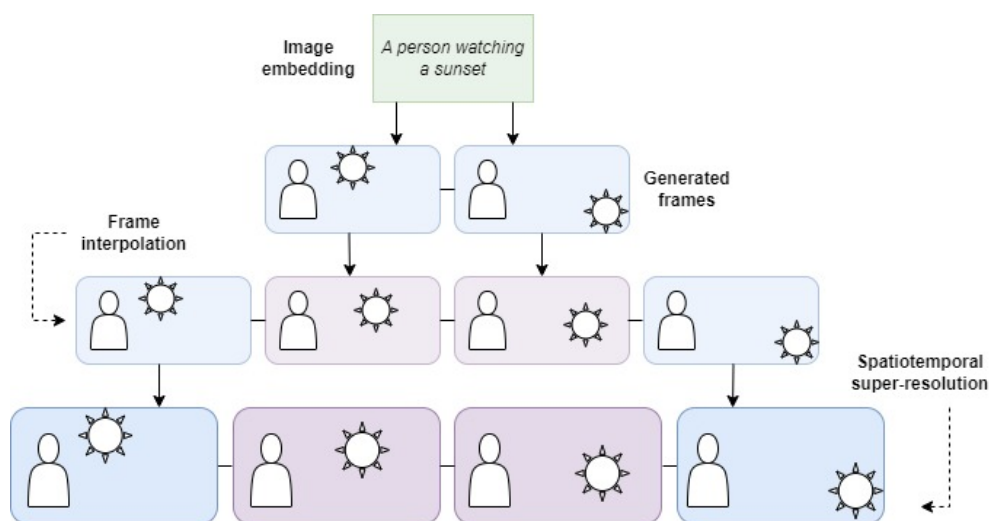


Figure 14: A high-level representation of the Make-A-Video model architecture, from text input to video output.

5.3 Sora

In February 2024, OpenAI released Sora, a generative text-to-video state-of-the-art multi-modal and multi-model system that can depict and animate video of up to one minute in length based on complex, detailed written instructions. Sora builds on recent breakthroughs in text-to-image generation and presented a leap in the state-of-the-art for text-to-video generation. Similarly to Make-A-Video, Sora is capable of generating video from text, as well as from static images, and can be used for editing existing video²⁹. In addition to the possibility of generating relatively long video sequences, Sora can sample and generate videos of different resolutions, unlike previous text-to-video models where the resolution is fixed.

Sora is a diffusion-transformer model [48]. As such, the model iteratively learns to predict high-resolution images (i.e., video frames) from observing noisy images. In a technical report by OpenAI²⁹, the architecture of Sora is outlined. The report states that enriched GPT user prompts guide video generation and that Sora is trained on video-caption pairs, where captions are conditioned by a separately trained video caption model [5]. Raw video input is first compressed onto a latent representation space, which is used during training and generation. The compressed video is then decomposed into image patches, analogous to breaking text into separate token elements. Sequences of image patches with added noise function as visual representations of space-time data for the model. This patching approach allows for training on visual data of varying image resolutions, temporal interval lengths, and aspect ratios. Finally, a decoder maps the generated latent representation onto a higher dimensional pixel space.

OpenAI attributes the relatively high quality of select demonstration videos partly to the scaling properties of the underlying diffusion transformer [48]. Despite the progress, as for Make-A-Video, Sora shows some areas of improvement; much like many deep model systems, it hallucinates concerning (i) physical realism, i.e., misinterpreting cause and effect, and (ii) spatiotemporal misplacements or object insertions. This can be inferred by ocular inspection from OpenAI's demonstration video landing page. Examples include paper planes merging unnaturally as they swarm above some trees, a forklift curiously sliding across beams on the ground, or people clapping their hands in seemingly strained ways³⁰. Further review and speculative analysis has been performed by Liu et al. [42].

²⁹<https://openai.com/research/video-generation-models-as-world-simulators> (visited 2024-03-05)

³⁰<https://openai.com/sora> (visited 2024-03-05)

6 Many Modalities

Models that handle many modalities are more complex than the models of the previous chapters but open up for more interesting holistic applications and other possibilities. A few recently presented models are described in the following subsections. The first two are multi-purpose models that can be used for several tasks, spanning different sets of modalities. The model Gato trains one transformer model to handle inputs from many modalities, while the model ImageBind builds a joint embedding space similar to CLIP (Section 3.1), but handles many modalities.

Among the applications in which a joint embedding space could be helpful is the difficult task of data fusion. In data fusion, data of several modalities are combined to reduce uncertainty in data, resolve conflicts between sources, and correlate additional types of information. Models specifically developed for data fusion are not included in this report, but there is promising research being done, for instance [15] that creates embeddings using older methods, and [20] that uses transformers to fuse data of different modalities. Additional exploration of this field will be left for the future if and when the area has developed further.

The last section in this chapter describes a few robotics models. In contrast to more specific robotics, i.e., robotics for industrial production, general robotics is a genuinely multi-modal problem. To navigate complex environments, a robot must be equipped with multiple sensors that provide many types of data and process this data collectively to decide which actions to take (a form of data fusion).

6.1 Gato

Gato is a general-purpose model that can perform a variety of tasks such as captioning images, engaging in chat conversations, playing board games, stacking blocks with a robot arm, and navigating in simulated 3D environments (i.e., actions can be taken in the physical world) [56]. Gato, a *single* transformer model, handles data from any task and modality provided that the data is serialized into a sequence of tokens [56]. This makes it possible to train or fine-tune Gato on a large amount of varied data to build a general-purpose model. The trained model can be prompted to perform different tasks, i.e., a demonstrating example is tokenized to form an initial input sequence that makes Gato output task-specific step-by-step actions, where each output action is influenced by the prior suggested output actions (so-called autoregressive planning) [56].

Gato is inspired by large language models, but tokenization and sequencing are done differently depending on the input modality. For instance, input images are split into non-overlapping fixed-size patch tokens as in ViT (Section 2.7), while continuous values representing senses or forces in the physical world are flattened into sequences of floating point values that are encoded, discretized and shifted [56]. The final model input is obtained by applying an embedding function to each token (different embedding operations are performed depending on the modality) [56]. Gato then uses a decoder-only transformer to output a distribution over the next token to be predicted (the output is predicted autoregressively). The model can predict text tokens, discrete and continuous values, and actions, but not image tokens [56]. This means, for instance, that Gato can be used to caption images but not generate images based on text descriptions. A sketch of Gato is shown in Figure 15.

6.2 ImageBind

Some models can be trained to represent the semantic content of data across multiple modalities in a joint embedding space (Section 2.2). An example is the CLIP model (Section 3.1), which learns a mutual representation for texts and images. The trained CLIP model can then compare a text with an image by measuring how close (i.e., similar) their representations

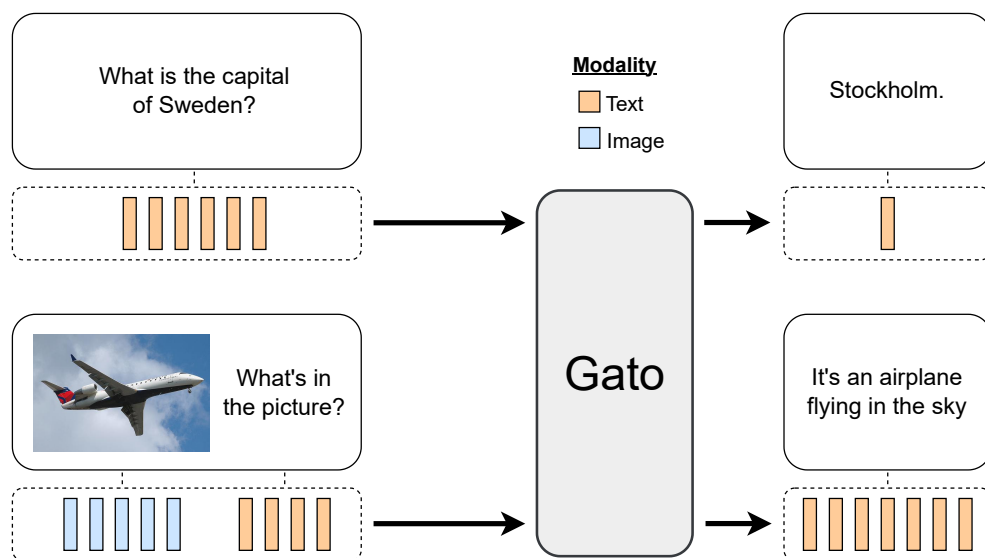


Figure 15: Here, Gato processes input data of two modalities (text and image) to answer questions. In each instance, the input is first serialized into a sequence of tokens (the orange bars represent text tokens, while the blue bars represent image tokens). The tokens are fed to Gato, which autoregressively predicts the subsequent tokens (i.e., the output). The task of answering questions about an image can be performed by other models such as PaLI (Section 3.3). However, Gato can handle additional modalities if the input data is serialized into a sequence of tokens [56]. The airplane image originates from the ImageNet [19] dataset.

are in the joint embedding space. ImageBind [28] is an approach to extend this concept to cover six different modalities (images, text, audio, depth, thermal, and inertial measurement unit (IMU) readings) at once. Acquiring paired data of all possible combinations of these modalities would pose a significant challenge. A key feature of ImageBind is its use of the image modality as a collective reference point, reducing the data requirement to the naturally occurring pairing of each modality to images (or videos). Figure 16 illustrates which modality combinations the model is trained on explicitly and which are learned implicitly.

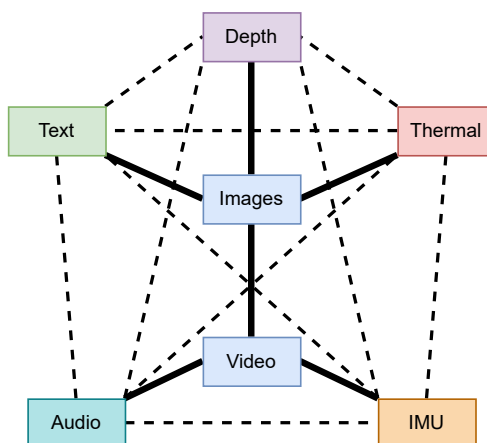


Figure 16: ImageBind is trained on six modalities: images/video, text, audio, depth, thermal, and inertial measurement unit (IMU) readings. The modality combinations are learned either explicitly or implicitly. Solid lines indicate that a pair of modalities occurs in the training data. Dashed lines indicate that the model learns a modality combination implicitly by using the image modality as a reference.

This results in a model capable of tasks such as cross-modal retrieval, detection of audio sources in images, and image generation given audio input. The model matches or outperforms specialist-supervised models on several benchmark tasks despite not seeing all pairs of modalities during training.

ImageBind creates embeddings using one dedicated transformer encoder for each modality in the training data. Embeddings for all modalities are projected to the same size using an additional small neural network, to allow for comparisons between them all. The encoders and embeddings are optimized through a symmetric contrastive loss function. The image and text encoders are initialized from pre-trained CLIP encoders, and their weights are kept frozen while the encoders for the other modalities are updated. The training is done without a specific downstream task, resulting in general-purpose embeddings. In theory, this makes it possible to retrieve data of any modality by posing a question in text or even a question in the form of any of the other modalities.

6.3 Robotics at Google

Robotics is a broad field faced with many challenges and problems. Steering a robot is indeed a multimodal problem – a robot control system usually needs to take many kinds of sensory data into consideration to steer the robot effectively. Robot control can be structured hierarchically into control levels, using several policies, which can be thought of as strategies for achieving goals. Low-level policies can control the basic movements of a robot, while higher-level policies can handle more complex tasks and break them down into actions or even singular movements. Higher-level control is typically more challenging, although even lower levels of control can also present difficulties. While industry robots in controlled environments have been essential to many types of production for a long time, deploying effective robotics in uncontrolled environments is well beyond current capabilities, unlike humans, who easily navigate, traverse, and interact with and within such environments. Examples include walking around a supermarket shopping for groceries or moving through crowds and security in an airport. Deep learning opens up new possibilities in robot control as in many other fields, and multimodal models, especially, may contribute to several solutions as robotics is inherently multimodal.

Since the introduction of deep learning in robotics, the field has seen extensive work. In particular, recent years have witnessed a growing number of publications on multimodal models, e.g., Gato (Section 6.1). Here, we describe a few recent papers from Google, but the reader is cautioned that many other organizations are also working on these problems. Moreover, the selection of papers covered in this report excludes several Google publications. The papers are quite complex as the subject and the experiments are complex and challenging to describe in text. It is also rather difficult to assess these methods' effectiveness without looking at video examples of robot actions executing various tasks, such as moving things in a kitchen environment. Further, it is also challenging to grasp how far these robotic systems are from actual real-life use outside the laboratories. There is no doubt, however, that impressive advancements are being made.

One of the biggest challenges in robotics based on machine learning is how difficult and costly it is to gather large amounts of data for many different tasks. The papers described here show that it is possible to improve robotics learning by leveraging the large amounts of data ingested by language- and vision-language models. For instance, these large “base models” facilitate transfer learning; the models can learn from one set of robotics data describing one task and become more successful at other tasks [6].

The following subsections each describe one model/paper. They all build upon each other to some extent, and reading in the corresponding order is recommended. Several interesting themes that are revisited in the selected papers, such as *grounding* and how to achieve long-horizon planning, i.e., high-level control involving creating a plan to reach a goal. Here, grounding refers to connecting a language- or vision-language model to the “reality” of a robot to control it. An interesting observation is that the two most recent papers use sim-

pler but larger models trained on more data to achieve better results. This is a theme in deep neural network research; a comparably simple network architecture often outperforms other solutions when trained on more data.

This set of papers uses a robot from Everyday Robots,³¹ a Google subsidiary that recently has merged into Google. The robot is by no means as impressive as for instance robots from Boston Dynamics^{32,33} but is instead meant to be a helper to humans in many environments. In these papers, the focus is kitchen environments. The robot has a base that can travel by wheels, a central stalk with a camera attached at the top that can swivel and turn up and down, and an arm with a gripper with many degrees of freedom. In the experiments in the following papers, using large neural network models, the robots are not fast. The control frequency of the fully integrated system in Section 6.3.4, where the robot is controlled by a model (RT-2) in a local cloud, is 1-3 Hz. Although they may seem slow now, this will be fine with time, improvements in computing power, and research into the efficiency of neural networks.

6.3.1 SayCan

SayCan [2] is a method for grounding a language model in the actions available to the robot. Put differently, it is a human language interface to a robot. SayCan is constructed to do planning, so when it receives a request, such as “Get me a soda can”, it repeatedly breaks it down into instructions by considering which of its set of possible actions is most likely to lead to achieving the request. A sketch of the system can be seen in Figure 17.

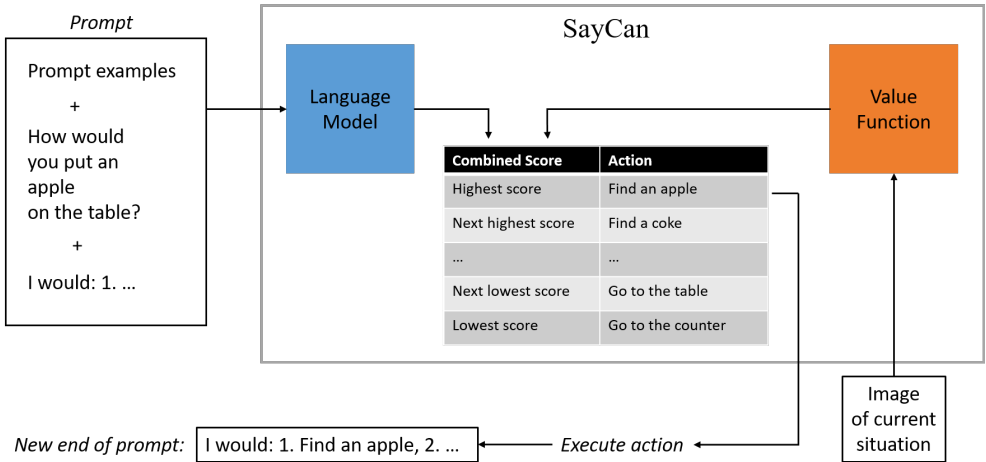


Figure 17: SayCan. A value function gives a score by estimating the success of each action a robot can do. Given the prompt, the language model scores the same robot actions. The robot executes the action with the highest combined score. If it succeeds, the action is added at the end of the prompt to the language model. The process is repeated until the model produces a termination token, which the language model produces.

The robot that SayCan instructs has been trained to perform actions. During this training, it has also learned a value function that estimates how likely it is that a particular action will be successful given the current situation. The situation is represented by the sensory data available to the robot, such as images from its camera. Each action also has an associated short text description, e.g., “find an apple,” “find a soda can,” and “pick up the soda can.”

³¹<https://everydayrobots.com/> (Visited 2023-09-13)
³²<https://bostondynamics.com/> (Visited 2023-09-13)
³³Notably these robots do not use machine learning for their control systems.

The language model (PaLM [16]) is given a prompt, starting with examples of how a higher-level task should be broken down into suitable actions, followed by a question describing the current task. For each action available to the robot, the probability of its success is estimated by the value function. As each action can also be considered a text string, its likelihood can be estimated by the main language model (PaLM [16]) given the prompt. The action with the highest combined probability (value function probability and language model probability) is chosen as the next step. For the “Get me a soda can” request, it could be the action to “find a soda can”. When and if the robot has implemented this action, i.e., found the can, the action text is added at the end of the prompt, and the process is repeated until the language model produces a termination token. The termination token (“done.”) is included in the examples at the beginning of the prompt.

In this work, a mobile manipulator from Everyday Robots was used,³⁴ and the evaluation was made over “101 instructions from 7 instruction families”. The method made successful plans in 81% of test cases (84% in the training environment) and executed correctly in 60% (74%).

6.3.2 RT-1

Robotics Transformer 1 (RT-1) [8] is trained on datasets of demonstrations of performed actions, the largest consisting of 130 000 episodes over 700 actions, and collected from 13 Everyday Robots over 17 months. It is constructed to receive text instructions and images representing the recent history as input and output of a robot action (Figure 18). The actions are encoded as a sequence of tokens representing seven arm movement dimensions (x, y, z, roll, pitch, yaw, opening of the gripper), three base movement dimensions (x, y, yaw), and a mode dimension signifying whether to control the arm, control the base, or terminating the action.

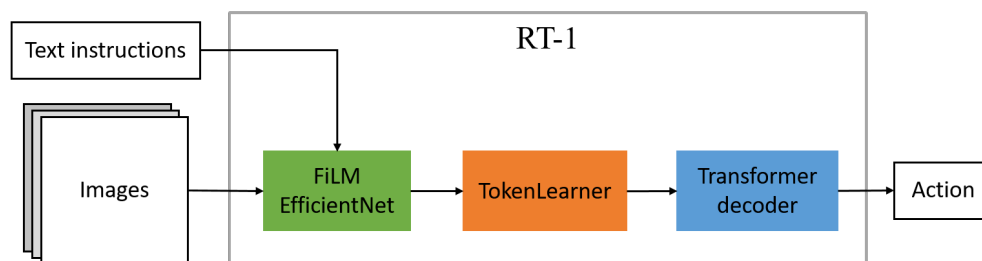


Figure 18: Robotics Transformer 1 (RT-1) [8] takes text instructions and a sequence of images representing history as input and output actions. The instructions are encoded using the Universal Sentence Encoder [12] and influence the EfficientNet-B3 [67] via FiLM [49] layers while it processes the image, resulting in tokens combining the text instructions and the image history. The combined tokens are passed through a TokenLearner [60] that adaptively learns how to reduce the set of tokens during training. These tokens are passed through a Transformer decoder to produce robot actions. Image after [8].

RT-1 is an imitation learner, meaning it learns by observing actions in the dataset. However, in several evaluations, of which we describe a few here, it exhibits both transfer learning capabilities and some generalization, which is likely due to the influence of the text instructions via the FiLM layers [49] (Figure 18). In the first evaluation, the model gets a success rate of 97% on tasks withheld from the primary dataset, performing significantly better than previous comparable models, such as Gato [56]. On a set of tasks it has not been exposed to before, the success rate is 76%. To test its robustness, distracting objects are introduced into the scene, and the background of the environment is changed in several ways. The performance drops to 83% with distracting objects and 59% with altered backgrounds.

³⁴<https://everydayrobots.com/> (Visited 23-11-15)

In the second evaluation, the model is retrained with two additional robot datasets, one at a time. In both cases, the performance of the original data is not impacted much, but the model improves on the actions in the new datasets:

- Simulated data with previously unseen objects improves performance on these objects in real-world evaluation.
- Data from a different robot arm³⁵ doing “bin-picking” [34] improves performance on this task over training only on the original data.

In the third evaluation, RT-1 in combination with SayCan (Section 6.3.1) is evaluated on so-called *long horizon tasks*, tasks with up to 50 steps (average 9.6 steps). SayCan iteratively breaks down a long horizon instruction into actions that RT-1 performs. A value function for RT-1 used in the SayCan method is implemented using MT-OPT [33].³⁶ The planning success rate on a test set in a similar kitchen as used for training is 87%, and the execution success rate is 67%. Evaluated in a kitchen different to the training kitchen, with drawers never seen before, the planning and execution success rate is still the same.

6.3.3 PaLM-E

PaLM-E [25] is based on the language model PaLM [16] and can be used for a wide range of tasks, including high-level robot control, language-only tasks, tasks on images and text such as image captioning, visual question answering of many varieties. The E in PaLM-E stands for *embodied*, which can be thought of as the language model in contact with real-world sensory data. It takes combinations of text, images, and other sensory data for input and generates text. The generated text is interpreted differently depending on the task. It can be an ordinary text, such as an answer to a question, or a text that should be interpreted as a set of instructions that a robot could carry out. Although the model has all these capacities, it has primarily been developed to do the latter. In the authors’ of [25] words; “...we investigate a generalist, multi-embodiment model, across multiple modalities.” the implied goal is that PaLM-E should be able to learn to control multiple different robots using inputs from several distinct modalities.

In Figure 19, the input is an image of the current state of some colored blocks on a table and a question on how to manipulate the blocks. The image is encoded into a number of embeddings in the same space as the language tokens using the ViT model (Section 2.7). The “mod Encoder” (Figure 19) symbolizes that data of other modalities also can be encoded and added to the “multimodal sentence”. The output is a text describing the manipulations the model suggests, which is then used to control the robot arm. When used to control a specific robot, PaLM-E is integrated into a control loop with lower-level policies for the robot. During training, it adapts to the lower-level policies and the multimodal data it receives. PaLM-E is grounded through this control loop and not via a value function as in SayCan (Section 6.3.1).

PaLM-E is trained and tested on a mixture of visual-language tasks and robotics tasks, where the latter constitutes less than 10% of the data. Two sets of experiments are conducted on this data. In the first set of experiments, a few ways to encode inputs, most importantly images, are evaluated. The ViT model (Section 2.7) uses the same static grid over every image. Although a static grid has proven effective in many circumstances, the authors of PaLM-E argue that combining with language models that operate on symbols representing objects is not ideal. It is also not ideal for “embodied reasoning,” which considers interactions with physical objects. The object scene representation transformer (OSRT) [62] decomposes scenes into objects in an unsupervised fashion and improves results when used in PaLM-E compared to ViT.

The second set of experiments investigates several things using one specific architecture, including a fixed way of encoding inputs, namely pre-trained ViT and PaLM models.

³⁵Kuka LBR iiwa, <https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa> (Visited 2023-08-24).

³⁶This is not explained further in the paper, and we leave it here.

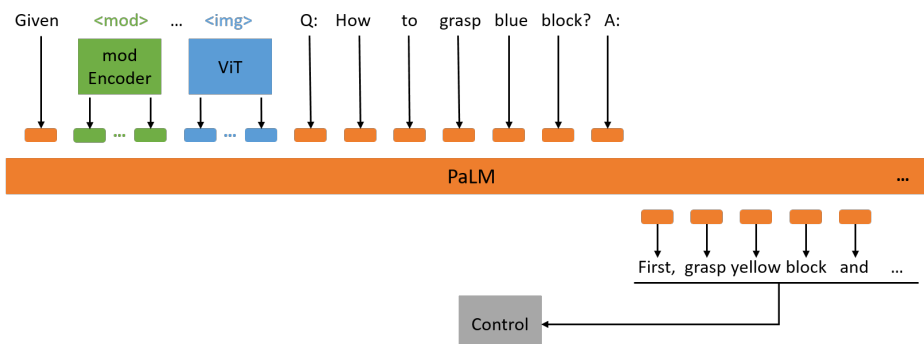


Figure 19: PaLM-E. The retrained PaLM receives embeddings of text and other modalities as input and output text. Images are encoded to embeddings using a visual transformer (ViT), and other modalities are treated similarly with other encoders. In this example, the model receives an image () depicting blocks of different colors placed on a table, potential data of other modalities (<mod>), and a textual question (Q) on how to manipulate the blocks. The model generates a text that answers (A) the question, i.e., describes how to grasp the blue block. PaLM-E adapts this output text to suit the lower level control policies of the particular robot and instruct the robot to manipulate the blocks. Example and image (with some minor changes) after [25].

Transfer learning is demonstrated; results on a few robotics tasks are improved when the model is trained on all of them, compared to being trained on each one separately. Long horizon planning similar to what is investigated in SayCan is evaluated qualitatively and found to be successful even under adversarial disturbances.

Catastrophic forgetting is always a risk when training with more and new data³⁷. Two methods for preventing this have been tested. First, the language model is frozen, and only the encoders are trained. This works to some extent but leads to some problems in robotics tasks. Using larger models can prevent catastrophic forgetting. The largest version, called PaLM-E-562B,³⁸ is shown to retain its general language capabilities and shows state-of-the-art performance in visual question answering although it is also trained on robotics tasks.

6.3.4 RT-2

The RT-2 (Robotics Transformer 2) [9] serves the same function as RT-1. It also has the same structure, but the three internal models of RT-1 are replaced with one vision language model, either PaLM-E or PaLI-X as shown in Figure 20. The largest versions used are a PaLI-X model of 55B parameters and a PaLM-E model of 12B, which can be compared to the full PaLM-E model of 562B parameters. The reason for using these smaller models is to make them practically usable. RT-2 is run in a cloud, with which the robot continuously communicates, and results in the entire system's roundtrip control frequency being only 1-3Hz.

The vision language model (VLM), i.e., PaLM-E or PaLI-X, is retrained to a vision language action model (VLA) by fine-tuning on both robot action data and text-image web data. This leads to better generalization than fine-tuning on just robot action data. RT-2 uses the same tokenization of actions as RT-1, and when RT-2 is used for robot control, the sampling of the next token is restricted to valid action tokens.

RT-2 is evaluated in several ways, and we describe a few of them here. RT-1 and RT-2 perform on par for tasks in the training data. RT-2 outperforms RT-1 and other previous models on objects, not in the training data. It is also much more robust to background changes. RT-2 even achieves a success rate of almost 50% in environments it has not been trained on.

³⁷Forgetting what was previously learned when trained further on more data.

³⁸PaLM-E-562B has 562 billion parameters, divided between 540B in the PaLM model and a ViT of 22B.

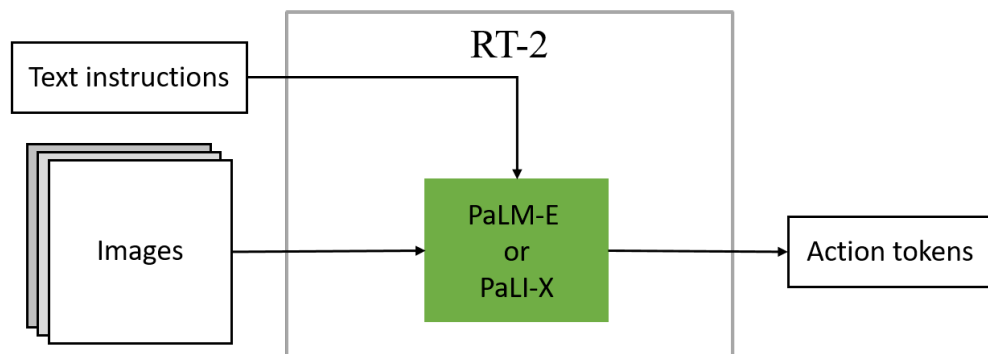


Figure 20: Robotics Transformer 2 (RT-2) [9] is, in essence, a PaLM-E or a PaLI-X model retrained to (also) handle robotics tasks. The input and output are the same as in RT-1 (Section 6.3.2).

The PaLM-E version of RT-2 performs better than the PaLI-X version on the more difficult generalization tasks while underperforming on easier tasks.

The vision-language capabilities in the vision-language models (PaLM-E and PaLI-X) result in *emergent* capabilities in RT-2 that were not present in the robot training data. This is tested on a separate list of instructions that is divided into three categories: symbol understanding (e.g., “push coke can on top of *heart*”), reasoning (e.g., “move the apple to *cup with same color*”), and human recognition (e.g., “move the coke can to *the person with glasses*”). On these instructions, RT-2 outperforms previous models significantly.

Due to its vision language model, RT-2 can also use the actions available in the robot training data in new ways. However, it does not learn new actions. The paper suggests that future work could investigate the possibility of learning from human videos.

7 Conclusions

Multimodal neural network models are currently attracting a lot of interest in the data science community. New or updated models are continuously released and gain considerable attention in both scientific forums and regular media.

In this report, we first provide an overview of some core technologies behind the current developments of multimodal models, after which we describe and discuss a set of models. The selected models process combinations of different modalities, focusing on text, sound, image, and video. As the models described are trained on data primarily intended for human communication or entertainment, we refer to them as multimodal media models.

Multimodal models in general have many potential applications. Their capability to connect and process data from several modalities enables them to solve complex tasks that were previously not possible to automate. Examples of applications in which multimodal models have already proved to be highly useful are artificial intelligence-based office assistants and home robotics. In addition, the model architecture – where all input regardless of modality is mapped to a common numerical embedding space – alleviates some of the need for so-called model *grounding*. Grounding in this context can be briefly explained as the process of “calibrating” the model against the real world which it represents.

Unfortunately, multimodal models, like much other technology, can both be used for malicious purposes and cause unintentional harm if used carelessly. For instance, multimodal media models can easily generate fake but highly realistic-looking media content, which can be used to affect the public opinion. Given the accessibility of the technology, we have to anticipate a dramatic and problematic increase in false or altered media content of all kinds in the coming years.

Like all neural network-based models, multimodal media models come with some technical challenges. For example, they reflect biases present in the data they are trained on, and their transparency and explainability are generally poor.

To conclude, much suggests that multimodal models have the potential to profoundly affect our society. For this reason, it is important that authorities and actors in the security and defense domain understand the technology so that they are able to identify possible implications – both good and bad – of the technology on their respective organizations, and adapt accordingly. On a national level it may even be necessary to discuss the potential need for control and regulation of the use and accessibility of future developments of multimodal technology.

References

- [1] Andrea Agostinelli et al. *MusicLM: Generating Music From Text*. Jan. 26, 2023. DOI: 10.48550/arXiv.2301.11325. preprint (cit. on p. 29).
- [2] Michael Ahn et al. *Do As I Can, Not As I Say: Grounding Language in Robotic Affordances*. Aug. 16, 2022. DOI: 10.48550/arXiv.2204.01691. preprint (cit. on p. 40).
- [3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. *ViViT: A Video Vision Transformer*. Nov. 1, 2021. DOI: 10.48550/arXiv.2103.15691. preprint (cit. on p. 18).
- [4] Arun Babu et al. *XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale*. Dec. 16, 2021. DOI: 10.48550/arXiv.2111.09296. preprint (cit. on p. 32).
- [5] James Betker et al. “Improving Image Generation with Better Captions”. In: 2023. URL: <https://www.semanticscholar.org/paper/Improving-Image-Generation-with-Better-Captions-Betker-Goh/cfee1826dd4743eab44c6e27a0cc5970effa4d80> [03/04/2024] (cit. on p. 35).
- [6] Rishi Bommasani et al. “On the Opportunities and Risks of Foundation Models”. Aug. 18, 2021. DOI: 10.48550/arxiv.2108.07258 (cit. on p. 39).
- [7] Dunja Božić-Štulić, Željko Marušić, and Sven Gotovac. “Deep Learning Approach in Aerial Imagery for Supporting Land Search and Rescue Missions”. In: *International Journal of Computer Vision* 127.9 (Sept. 1, 2019), pp. 1256–1278. DOI: 10.1007/s11263-019-01177-1 (cit. on p. 25).
- [8] Anthony Brohan et al. *RT-1: Robotics Transformer for Real-World Control at Scale*. Aug. 11, 2023. DOI: 10.48550/arXiv.2212.06817. preprint (cit. on p. 41).
- [9] Anthony Brohan et al. *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control*. July 28, 2023. DOI: 10.48550/arXiv.2307.15818. preprint (cit. on pp. 43, 44).
- [10] Tom Brown et al. “Language Models Are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: <https://papers.nips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html> [12/06/2022] (cit. on p. 14, 17).
- [11] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. “Re-Evaluating the Role of Bleu in Machine Translation Research”. In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. EACL 2006. Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 249–256. URL: <https://aclanthology.org/E06-1032> [10/02/2023] (cit. on p. 30).
- [12] Daniel Cer et al. *Universal Sentence Encoder*. Apr. 12, 2018. DOI: 10.48550/arXiv.1803.11175. preprint (cit. on p. 41).
- [13] Xi Chen et al. *PaLI-X: On Scaling up a Multilingual Vision and Language Model*. May 29, 2023. DOI: 10.48550/arXiv.2305.18565. preprint (cit. on p. 24).
- [14] Xi Chen et al. *PaLI: A Jointly-Scaled Multilingual Language-Image Model*. June 5, 2023. DOI: 10.48550/arXiv.2209.06794. preprint (cit. on pp. 23, 24).
- [15] Dongjin Choi, Barry Drake, and Haesun Park. “Co-Embedding Multi-type Data for Information Fusion and Visual Analytics”. In: *2023 26th International Conference on Information Fusion (FUSION)*. 2023 26th International Conference on Information Fusion (FUSION). June 2023, pp. 1–8. DOI: 10.23919/FUSION52260.2023.10224157 (cit. on p. 37).

- [16] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. Oct. 5, 2022. DOI: 10.48550/arXiv.2204.02311. preprint (cit. on pp. 17, 41, 42).
- [17] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. “W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training”. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). Dec. 2021, pp. 244–250. DOI: 10.1109/ASRU51503.2021.9688253 (cit. on p. 32).
- [18] Deborah Coughlin. “Correlating Automated and Human Assessments of Machine Translation Quality”. In: *Proceedings of Machine Translation Summit IX: Papers*. MTSummit 2003. New Orleans, USA, Sept. 23, 2003. URL: <https://aclanthology.org/2003.mtsummit-papers.9> [08/25/2023] (cit. on p. 30).
- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 22, 25, 38).
- [20] Alexander C. DeRieux, Walid Saad, Wangda Zuo, Rachmawan Budiarto, Mochamad Donny Koerniawan, and Dwi Novitasari. *A Transformer Framework for Data Fusion and Multi-Task Learning in Smart Cities*. Nov. 18, 2022. DOI: 10.48550/arXiv.2211.10506. preprint (cit. on p. 37).
- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 (cit. on pp. 16, 18).
- [22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. May 24, 2019. DOI: 10.48550/arXiv.1810.04805. preprint (cit. on pp. 17, 32).
- [23] Qingxiu Dong et al. *A Survey on In-context Learning*. June 1, 2023. DOI: 10.48550/arXiv.2301.00234. preprint (cit. on p. 13).
- [24] Alexey Dosovitskiy et al. *An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale*. June 3, 2021. DOI: 10.48550/arXiv.2010.11929. preprint (cit. on pp. 18, 22).
- [25] Danny Driess et al. *PaLM-E: An Embodied Multimodal Language Model*. Mar. 6, 2023. DOI: 10.48550/arXiv.2303.03378. preprint (cit. on pp. 42, 43).
- [26] David Foster. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*. O’Reilly Media, Incorporated, 2023. ISBN: 978-1-09-813418-1 (cit. on p. 13).
- [27] Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. *Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling*. Nov. 1, 2023. DOI: 10.48550/arXiv.2311.00430. preprint (cit. on p. 31).
- [28] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. *ImageBind: One Embedding Space To Bind Them All*. May 31, 2023. DOI: 10.48550/arXiv.2305.05665. preprint (cit. on pp. 14, 38).
- [29] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. June 10, 2014. DOI: 10.48550/arXiv.1406.2661. preprint (cit. on p. 15).

- [30] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. Mar. 9, 2015. DOI: 10.48550/arXiv.1503.02531. preprint (cit. on p. 31).
- [31] Fredrik Johansson, Andreas Horndahl, Hanna Lilja, Marianela Garcia Lozano, Lukas Lundmark, Magnus Rosell, and Harald Stiff. *Detection of Fabricated Media*. Tech. rep. FOI-R--5132--SE. Stockholm, Sweden: Swedish Defence Research Agency, Apr. 16, 2021 (cit. on p. 13).
- [32] Ronnie Johansson, Magnus Rosell, Tove Gustavi, Edward Tjörnhammar, and Peter Hammar. *Knowledge, Graphs and Language Models*. Tech. rep. FOI-R--5504--SE. Stockholm, Sweden: Swedish Defence Research Agency, Feb. 15, 2024, p. 57 (cit. on p. 13).
- [33] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. *MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale*. Apr. 27, 2021. DOI: 10.48550/arXiv.2104.08212. preprint (cit. on p. 42).
- [34] Dmitry Kalashnikov et al. *QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation*. Nov. 27, 2018. DOI: 10.48550/arXiv.1806.10293. preprint (cit. on p. 42).
- [35] Maja Karasalo, Fredrik Johansson, Magnus Rosell, Andreas Horndahl, Lukas Lundmark, Hanna Lilja, Ulrika Wickenberg Bolin, and Harald Stiff. *Natural Language Processing Using Deep Neural Networks*. Tech. rep. FOI-R--4948--SE. Stockholm, Sweden: Swedish Defence Research Agency, May 8, 2020, p. 46 (cit. on pp. 13, 16).
- [36] Alexander Kirillov et al. *Segment Anything*. Apr. 5, 2023. DOI: 10.48550/arXiv.2304.02643. preprint (cit. on pp. 25–27).
- [37] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. *Big Transfer (BiT): General Visual Representation Learning*. May 5, 2020. DOI: 10.48550/arXiv.1912.11370. preprint (cit. on p. 30).
- [38] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 17022–17033. URL: <https://proceedings.neurips.cc/paper/2020/hash/c5d736809766d46260d816d8dbc9eb44-Abstract.html> [12/06/2023] (cit. on p. 32).
- [39] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. *Semantic-SAM: Segment and Recognize Anything at Any Granularity*. July 10, 2023. DOI: 10.48550/arXiv.2307.04767. preprint (cit. on pp. 26, 27).
- [40] Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. *Visually Grounded Reasoning across Languages and Cultures*. Oct. 21, 2021. DOI: 10.48550/arXiv.2109.13238. preprint (cit. on p. 9).
- [41] Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. *Mitigating Political Bias in Language Models Through Reinforced Calibration*. Apr. 30, 2021. DOI: 10.48550/arXiv.2104.14795. preprint (cit. on p. 9).
- [42] Yixin Liu et al. *Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models*. Feb. 26, 2024. preprint (cit. on p. 35).
- [43] Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. *Are Emergent Abilities in Large Language Models Just In-Context Learning?* Sept. 4, 2023. DOI: 10.48550/arXiv.2309.01809. preprint (cit. on p. 13).
- [44] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. *X-CLIP: End-to-End Multi-grained Contrastive Learning for Video-Text Retrieval*. Sept. 22, 2022. DOI: 10.48550/arXiv.2207.07285. preprint (cit. on p. 33).

- [45] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. *Exploring the Limits of Weakly Supervised Pretraining*. May 2, 2018. DOI: 10.48550/arXiv.1805.00932. preprint (cit. on p. 30).
- [46] OpenAI. *GPT-4 Technical Report*. Mar. 27, 2023. DOI: 10.48550/arXiv.2303.08774. preprint (cit. on pp. 14, 17).
- [47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “Bleu: A Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL 2002. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135 (cit. on p. 30).
- [48] William Peebles and Saining Xie. *Scalable Diffusion Models with Transformers*. Mar. 2, 2023. DOI: 10.48550/arXiv.2212.09748. preprint (cit. on p. 35).
- [49] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. “FiLM: Visual Reasoning with a General Conditioning Layer”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI’18/IAAI’18/EAAI’18. New Orleans, Louisiana, USA: AAAI Press, Feb. 2, 2018, pp. 3942–3951. ISBN: 978-1-57735-800-8 (cit. on p. 41).
- [50] Ingo Plag, Christiane Dalton-Puffer, and Harald Baayen. “Morphological Productivity across Speech and Writing”. In: *English Language & Linguistics* 3.2 (Nov. 1999), pp. 209–228. DOI: 10.1017/S1360674399000222 (cit. on p. 29).
- [51] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. *SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis*. July 4, 2023. DOI: 10.48550/arXiv.2307.01952. preprint (cit. on p. 23).
- [52] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. *Robust Speech Recognition via Large-Scale Weak Supervision*. Dec. 6, 2022. DOI: 10.48550/arXiv.2212.04356. preprint (cit. on pp. 29, 31).
- [53] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving Language Understanding by Generative Pre-Training”. In: (2018). URL: <https://openai.com/blog/language-unsupervised/> (cit. on pp. 14, 17).
- [54] Alec Radford, Jeffry Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. “Language Models Are Unsupervised Multitask Learners”. In: (2019). URL: <https://openai.com/blog/better-language-models/> (cit. on pp. 14, 17, 22).
- [55] Alec Radford et al. “Learning Transferable Visual Models from Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 18–24, 2021, pp. 8748–8763. URL: <https://proceedings.mlr.press/v139/radford21a.html> (cit. on pp. 8, 14, 21, 22).
- [56] Scott Reed et al. *A Generalist Agent*. Nov. 11, 2022. DOI: 10.48550/arXiv.2205.06175. preprint (cit. on pp. 37, 38, 41).
- [57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 13, 2022. DOI: 10.48550/arXiv.2112.10752. preprint (cit. on pp. 16, 23).
- [58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. May 18, 2015. DOI: 10.48550/arXiv.1505.04597. preprint (cit. on p. 15).

- [59] Magnus Rosell et al. *Semi-automatisk datadriven webbanalys: detektion av fabricerad media, trovärdighetsbedömning och cyberhotsbevakning*. Tech. rep. FOI-R--5262--SE. Stockholm, Sweden: Swedish Defence Research Agency, Apr. 5, 2022, p. 101 (cit. on p. 13).
- [60] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. “TokenLearner: Adaptive Space-Time Tokenization for Videos”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 12786–12797. URL: <https://proceedings.neurips.cc/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html> [08/22/2023] (cit. on p. 41).
- [61] Johan Sabel, Hanna Lilja, and David Gustafsson. *Automatisk bildanalys med multimodala modeller*. FOI Memo 8012. Stockholm, Sweden: Swedish Defence Research Agency, Dec. 19, 2022, p. 14 (cit. on p. 13).
- [62] Mehdi S. M. Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J. Guibas, Klaus Greff, and Thomas Kipf. “Object Scene Representation Transformer”. In: *Advances in Neural Information Processing Systems* 35 (Dec. 6, 2022), pp. 9512–9524. [09/01/2023] (cit. on p. 42).
- [63] Christoph Schuhmann et al. *LAION-5B: An Open Large-Scale Dataset for Training next Generation Image-Text Models*. Oct. 15, 2022. DOI: 10.48550/arXiv.2210.08402. preprint (cit. on p. 23).
- [64] *Seamless: Multilingual Expressive and Streaming Speech Translation | Research - AI at Meta*. URL: <https://ai.meta.com/research/publications/seamless-multilingual-expressive-and-streaming-speech-translation/> [12/06/2023] (cit. on pp. 31, 32).
- [65] Uriel Singer et al. *Make-A-Video: Text-to-Video Generation without Text-Video Data*. Sept. 29, 2022. preprint (cit. on p. 34).
- [66] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning Using Nonequilibrium Thermodynamics*. Nov. 18, 2015. DOI: 10.48550/arXiv.1503.03585. preprint (cit. on pp. 15, 16, 23).
- [67] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Sept. 11, 2020. DOI: 10.48550/arXiv.1905.11946. preprint (cit. on p. 41).
- [68] NLLB Team et al. *No Language Left Behind: Scaling Human-Centered Machine Translation*. Aug. 25, 2022. DOI: 10.48550/arXiv.2207.04672. preprint (cit. on p. 32).
- [69] Edward Tjörnhammar, Ronnie Johansson, Johan Schubert, and Gunnar Eriksson. *Forskningsöversikt: datakomprimering, informationsfusion, dataanalys*. Tech. rep. FOI-R--5237--SE. Stockholm, Sweden: Swedish Defence Research Agency, May 2, 2022, p. 72 (cit. on p. 13).
- [70] Erik Valldor. *CLIP För Återigenkänning Av Personer i Bevakningskameror*. Tech. rep. FOI-D--1175--SE. Stockholm: Swedish Defence Research Agency, 2022 (cit. on p. 22).
- [71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need”. Dec. 5, 2017. DOI: 10.48550/arXiv.1706.03762. preprint (cit. on pp. 16, 18, 24).
- [72] Jason Wei et al. “Emergent Abilities of Large Language Models”. In: *Transactions on Machine Learning Research* (2022). ISSN: 2835-8856. URL: <https://openreview.net/forum?id=yzkSU5zdwD> (cit. on p. 14).
- [73] Johannes Welbl et al. *Challenges in Detoxifying Language Models*. Sept. 15, 2021. DOI: 10.48550/arXiv.2109.07445. preprint (cit. on p. 9).
- [74] David H. Wolpert. “Stacked Generalization”. In: *Neural Networks* 5.2 (Jan. 1992), pp. 241–259. DOI: 10.1016/s0893-6080(05)80023-1 (cit. on p. 9).

- [75] David H. Wolpert. “The Lack of A Priori Distinctions Between Learning Algorithms”. In: *Neural Computation* 8.7 (Oct. 1996). DOI: 10.1162/neco.1996.8.7.1341 (cit. on p. 9).
- [76] David H. Wolpert. “The Supervised Learning No-Free-Lunch Theorems”. In: *Soft Computing and Industry*. Ed. by Rajkumar Roy, Mario Köppen, Seppo Ovaska, Takeshi Furuhashi, and Frank Hoffmann. London: Springer London, 2002, pp. 25–42. DOI: 10.1007/978-1-4471-0123-9_3 (cit. on p. 9).
- [77] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. *Detecting Twenty-thousand Classes Using Image-level Supervision*. July 29, 2022. DOI: 10.48550/arXiv.2201.02605. preprint (cit. on pp. 22, 24, 25).
- [78] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. *Segment Everything Everywhere All at Once*. Version 4. July 11, 2023. DOI: 10.48550/arXiv.2304.06718. preprint (cit. on p. 26).

