

June 2023

Image-To-Text Mock-Up

[Open: Research Question](#)

Done: Research Questions

General & First Questions

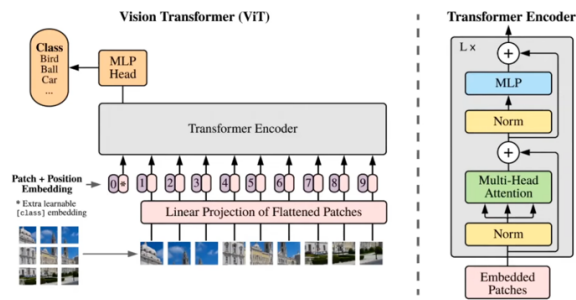
- / Question:
If I want to have multiple labels / annotations derived from one image - what is the correct search term?
- / Answer:
Multi-label image classification: Multi-label image classification is the task of predicting a set of labels corresponding to objects, attributes or other entities present in an image.
- / Question:
Also, here we do not have pretrained data, means zero-shot image classification?
- / Answer:
Yes. Zero-shot image classification is a task that involves classifying images into different categories using a model that was not explicitly trained on data containing labeled examples from those specific categories. Traditionally, image classification requires training a model on a specific set of labeled images, and this model learns to “map” certain image features to labels. When there’s a need to use such model for a classification task that introduces a new set of labels, fine-tuning is required to “recalibrate” the model. In contrast, zero-shot or open vocabulary image classification models are typically multi-modal models that have been trained on a large dataset of images and associated descriptions. These models learn aligned vision-language representations that can be used for many downstream tasks including zero-shot image classification. This is a more flexible approach to image classification that allows models to generalize to new and unseen categories without the need for additional training data and enables users to query images with free-form text descriptions of their target objects .
- / Resource:
[Hugging Face / Zero Shot Image Classification \(great article\)](#)

Architecture

- / Question:
For Multi-label image classification, which library am I supposed to use? Generally, are CNNs still a thing or are only transformers used?
- / Answer:
You can still find all relevant *CNN* Models (ResNet, RegNet, EfficientNet) on Hugging Face (the library is called Timm).
Vision transformer is a recent breakthrough in the area of computer vision. While transformer-based models have dominated the field of natural language processing since 2017, CNN-based models are still demonstrating state-of-the-art performances in vision problems. In 2021, a group of researchers from Google figured out how to make a transformer work on recognition. They called it "vision transformer". The follow-up works by the community demonstrated superior performance of vision transformers not only in recognition but also in other downstream tasks such as detection, segmentation, multi-modal learning and scene text recognition to mention a few.

Let's examine the vision transformer architecture step by step.

1. Split an image into patches
2. Flatten the patches
3. Produce lower-dimensional linear embeddings from the flattened patches
4. Add positional embeddings
5. Feed the sequence as an input to a standard transformer encoder
6. Pretrain the model with image labels (fully supervised on a huge dataset)
7. Finetune on the downstream dataset for image classification



/ Resources:

[Hugging Face / Timm Library for CNNs](#)

[YouTube / Lecture on Vision Transformer and its Applications](#)

[Paper / An image is worth 16x16 words: transformers for image recognition at scale](#)

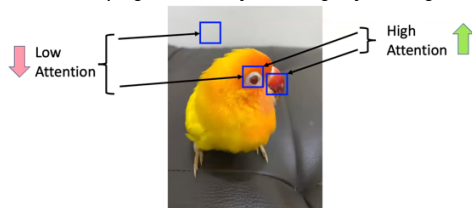
[Blog Post / Visual Transformer Guide \(really really good!\)](#)

/ Question:

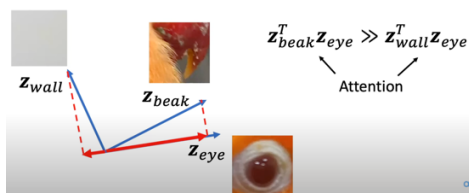
How is *visual attention* translated to images?

/ Answer:

Patches having a high relevance/relationship/dependence are scored with a high attention (e.g. bird's eye and bird's mouth) and patches with low relevance/relationship/dependence are scored with a low attention (e.g. bird's eye and grey background).



Also, from a mathematical point, attention is the dot-product between two features:



Of course, also true for natural language processing where attention is the dot-product between two words describing their dependence or relationship.

High Attention
The quick brown fox jumps over the lazy dog
Low Attention

/ Resources:

[YouTube / Lecture on Vision Transformer and its Applications](#)

/ Question:

What is the SOTA model when it comes down to vision transformers for multi-model image classification?

/ Answer:

/ CLIP (Contrastive Language–Image Pre-training) by OpenAI builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning. The idea of zero-data learning dates

back over a decade but until recently was mostly studied in computer vision as a way of generalizing to unseen object categories. The model was trained on 400 million image-text-pairs, also very diverse set of data (in comparison, ImageNet was trained on 1,2 million)

- + CLIP outperforms other models on zero-shot performance on unseen data
- + CLIP does not only concentrate on one class (e.g. a picture of a dog with a house in the background, CLIP does not only classify “dog” but could also classify “house”)
- + CLIP can solve unknown tasks like numerical character recognition, satellite image recognition ...
- CLIP is very data heavy (is computational more efficient due to parallel processing however, needs tons of data to be effectively trained, CNNs are more data efficient)



In the training phase, CLIP calculated square similarities between a given image and the corresponding text:

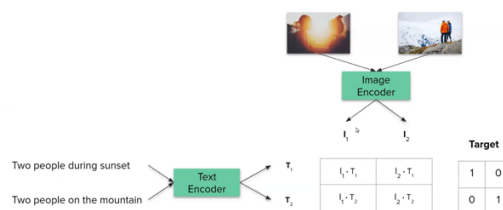


To do so, CLIP needs both: an image encoder and a text encoder:



During training:

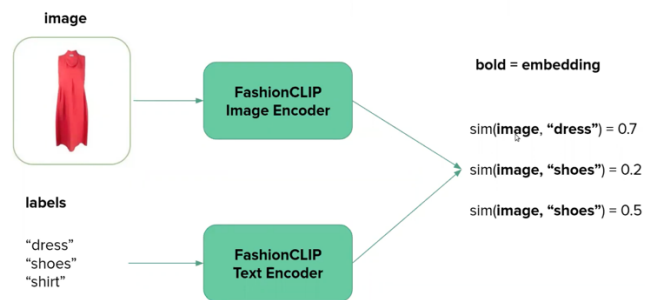
		
Two people during sunset	i_1	0
Two people on the mountain	0	i_2



- / Resources:
 - [Hugging Face / OpenAI CLIP Model](#)
 - [OpenAI Article / CLIP: Connecting text and images](#)
 - [YouTube / OpenAI's CLIP explained](#)

- / Question:
 - Is there anything specific for Fashion?

- / Answer:
 - There is a FashionCLIP, based on CLIP and fine-tuned with 800k fashion image-text pairs. Based on own data set as well as Fashion MNIST, Kaggle Dataset and Deep Fashion.



Result of F1 is that it Fashion CLIP performs better

/ Resource:

[YouTube / Domain-Specific Multi-Modal ML with CLIP](#)

[Arxiv Article / Contrastive language and vision learning of general fashion concepts](#)

Implementation (Image Classification)

/ Question:

How does the implementation work?

/ Answer:

A mixture between this implementation by Hugging Face and the FashionCLIP implementation could be an option

/ Resources:

[Hugging Face / Zero Shot Image Classification \(great article\)](#)

[Hugging Face / FashionCLIP Model Card](#)

Implementation (Image Description)

/ Question:

How does the implementation work for detailed image descriptions?

/ Answer:

There are two dominant models here: <https://huggingface.co/nlpconnect/vit-gpt2-image-captioning> und <https://huggingface.co/Salesforce/blip-image-captioning-base>

However, the results are very limited – only short descriptions

Thoughts

For article classification (use case e.g. automatic tagging of images on the homepage), the fashionCLIP model works very well and could be implemented. For product description, it is not possible yet (at least according to my research) to retrieve a detailed image description in a certain style (e.g. Witt Weiden style) from an image. Also as many key facts like material and so on are missing and the models are not able yet to describe the details (like this "cotton-style" ...). Also: the generated text has to be in German and has to fit to the style of the other descriptions (impossible (?) from only image-to-text).

In the end, you need to have

- 1) Example description: Eleganter Slingpumps für viele Anlässe von Caprice. Das Besondere ist die Trittdämpfung, die jeden Schritt weich abfedert. Obermaterial und Innenausstattung aus Leder. Flexible TR-Laufsohle mit ca. 45 mm Absatz. Weite G.
- 2) Information on the new product:
 - Artikelnummer: 536.514.002
 - Besonderheit: Antirutschsohle, Wechselfußbett, weiche Polsterung
 - Marke: Gemini
 - Obermaterial: Glattleder
 - Innenfutter: Ungefüttert
 - Farbe: weiß
 - Verschluss: Schlupf

Innensohle: Leder

Innenfutter: Leder

Außensohle: TR

Futter: 100% Leder

Material: 100% Glattleder

Schuhe: Slipper

- 3) Prompt: Kannst du einen Text (max 120 Wörter) generieren basierend auf den obigen Daten?

Testing

Appendix: Recap on Transformer Models

Background

RNNs and LSTMs are slow to train due to sequential processing (one word after another). Transformer models however can process input in a simultaneous manner (huge chunks of text at once), and are therefore significantly faster to train on GPUs.

Three different groupings of Transformer Models:

- GPT-like (also called auto-regressive Transformer models)
- BERT-like (also called auto-encoding Transformer models)
- BART/T5-like (also called sequence-to-sequence Transformer models)

Learning

All the models use self-supervised learning, a type of training in which the objective is automatically computed from the inputs of the model eliminating the need for humans to label the data. After the initial learning, there is a need for transfer learning (initializing a model with another model's weights) for the model to be specified on the given task. Here, the model is fine-tuned in a supervised way — that is, using human-annotated labels — on a given task.

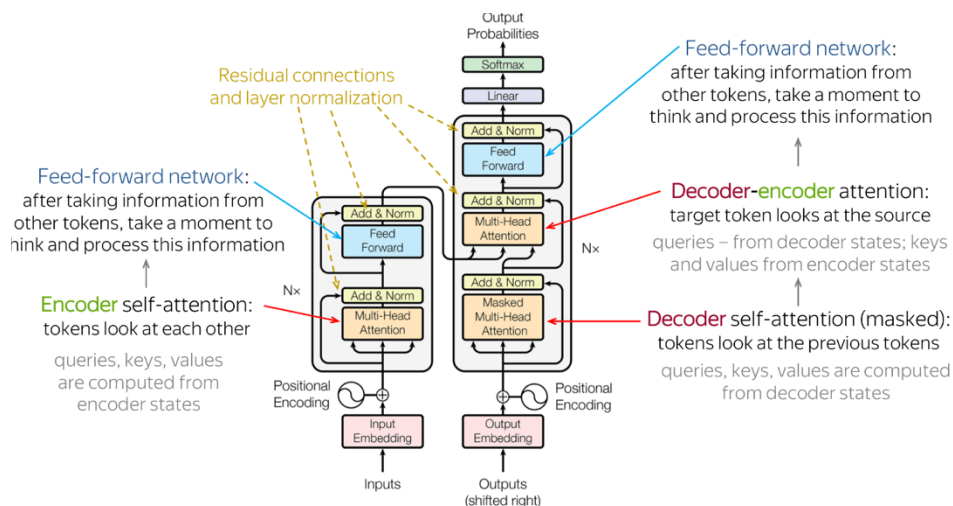
Transformer Architecture

/ Encoder

The encoder receives an input and builds a representation of it (its features). After “encoding” e.g. text into numerical representations, the encoder feeds the encoded features forward to the decoder.

/ Decoder

The decoder uses the encoder's representation (features) along with other inputs to generate a target sequence. After “decoding” the representations from the encoder, the output probabilities for e.g. a predicted word are given.



Resources

[Website / Hugging Face Intro Transformer Architecture](#)

[Paper / Attention Is All You Need](#)

[YouTube / Transformer Neural Networks Explained](#)