Neural Networks - II

Supported by AI Tennessee initiative
IAMM, UTK

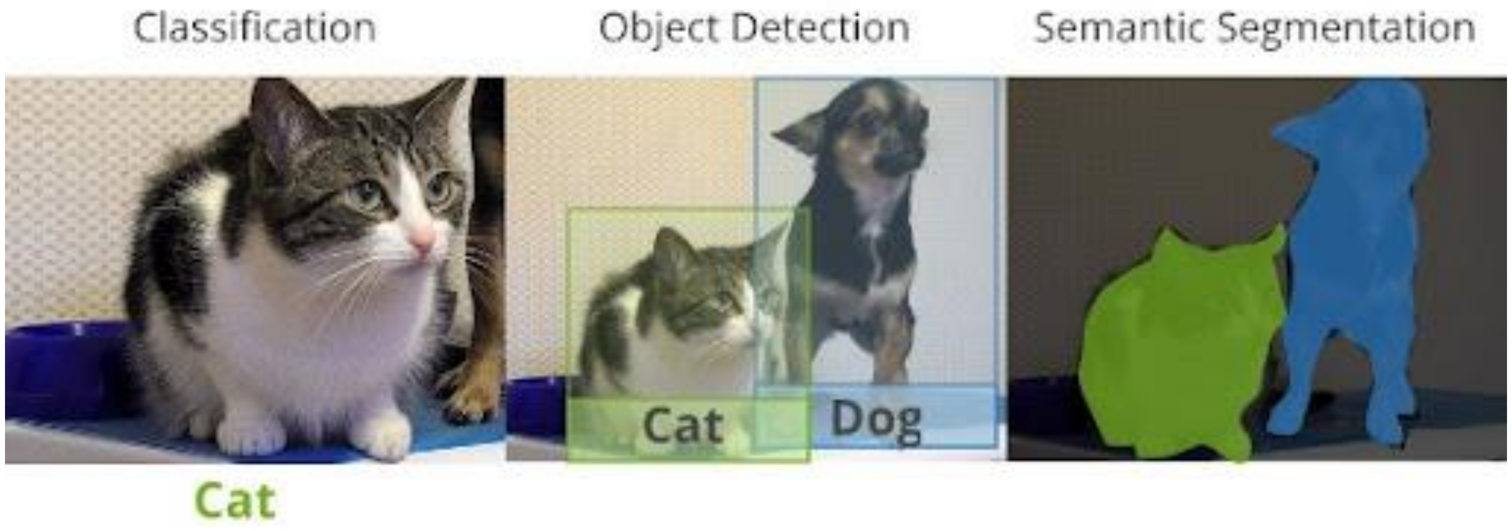
THE UNIVERSITY OF TENNESSEE KNOXVILLE

- Presented by Utkarsh Pratiush

Agenda

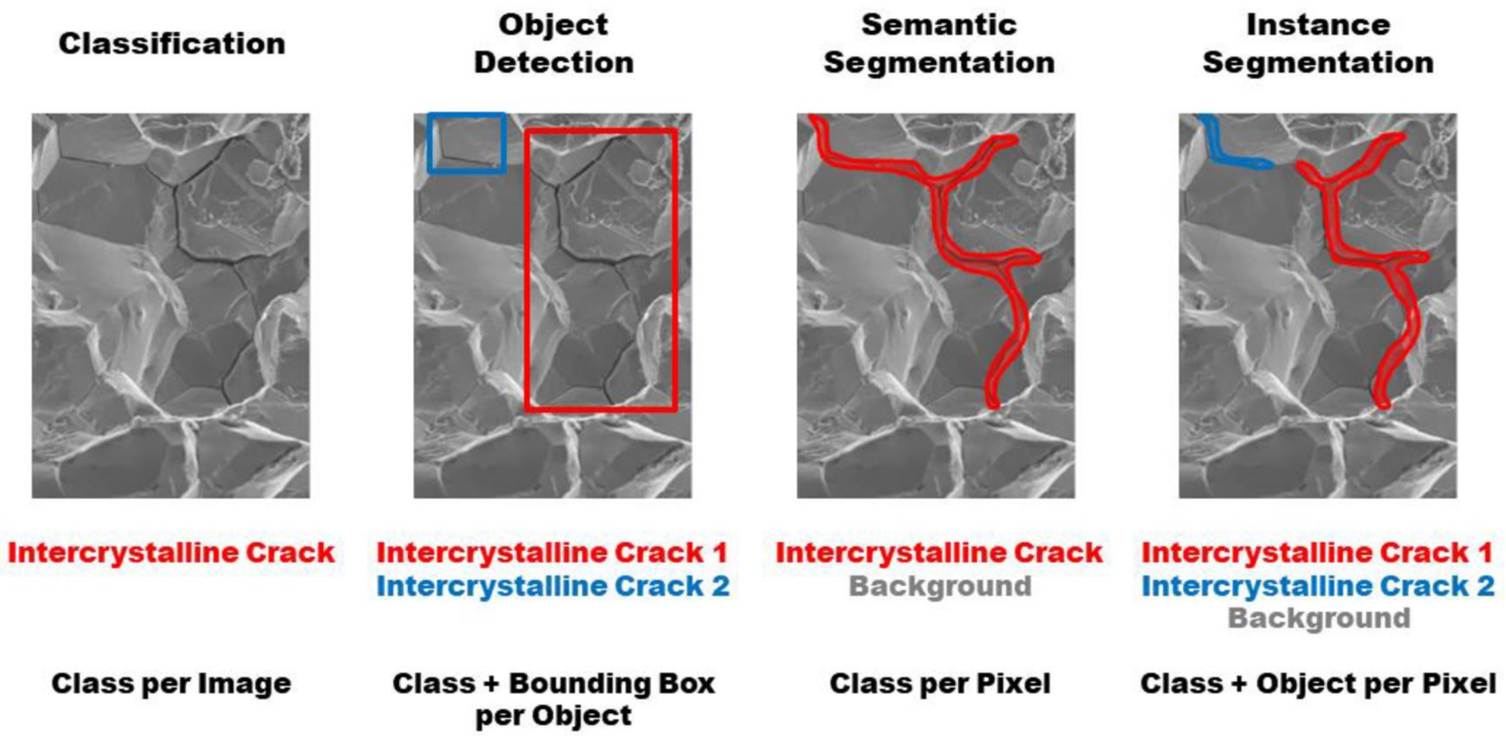
1. Problems in Image analysis
2. Popular architectures
   - U-net
     - Architecture
     - Loss function
     - Results
     - Hands on
     - U-net for Moiré lattices - Gomb-net
   - **Foundational model's**
     - CLIP
     - SAM
       - CELL-SAM
3. Explainability in Neural networks
   - Saliency maps, SHAP, Attention..

1. Problems in Image analysis



Q How is Semantic segmentation and classification problem related?

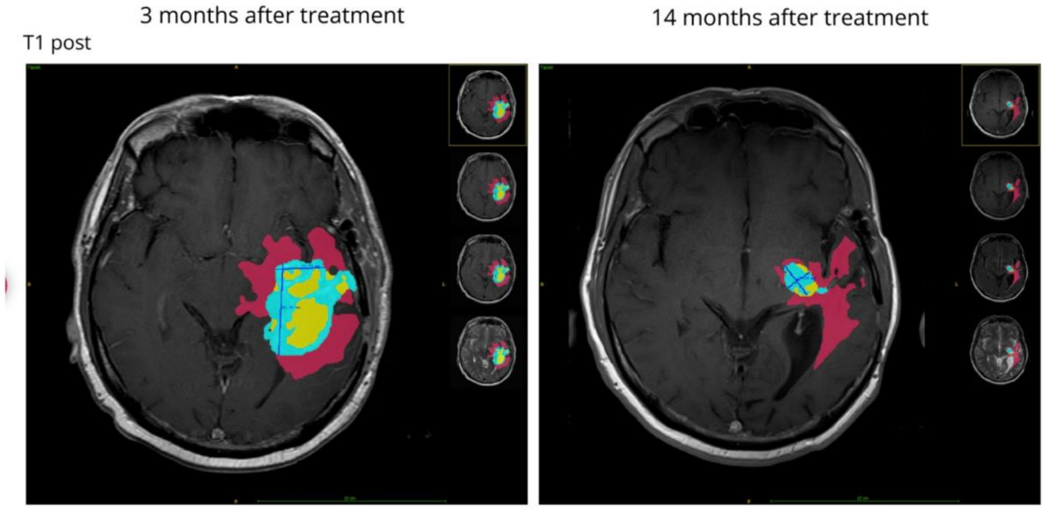
1. Problems in Image analysis



Müller, M., Stiefel, M., Bachmann, B.-I., Britz, D., & Mücklich, F. (2024). Overview: Machine Learning for Segmentation and Classification of Complex Steel Microstructures. *Metals*, 14(5), 553.

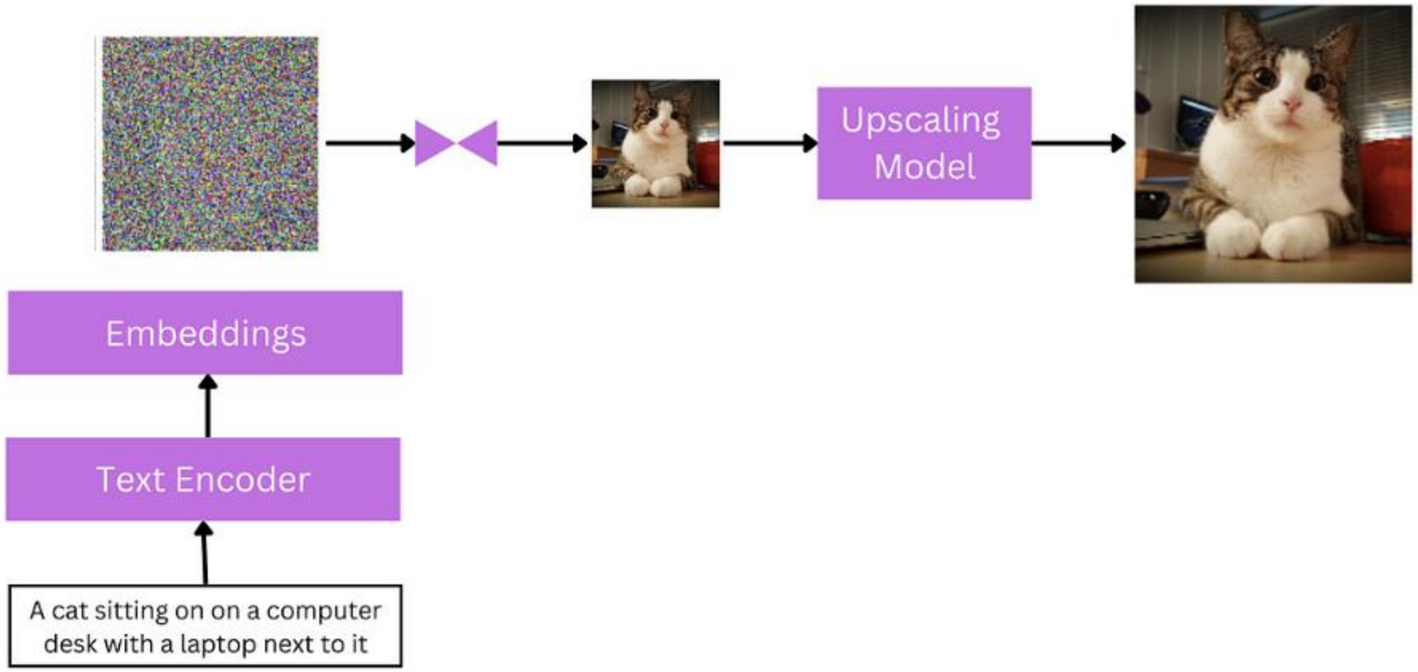
1. Problems in Image analysis

Why segmentation important?



1. Problems in Image analysis

- **Going beyond segmentation**



Chatgpt demo!

Not very useful for Microscopy at this point!

2. Popular architectures : U-net

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

Introduced in 2015
Cited ~ 111077 till today

2. Popular architectures : U-net

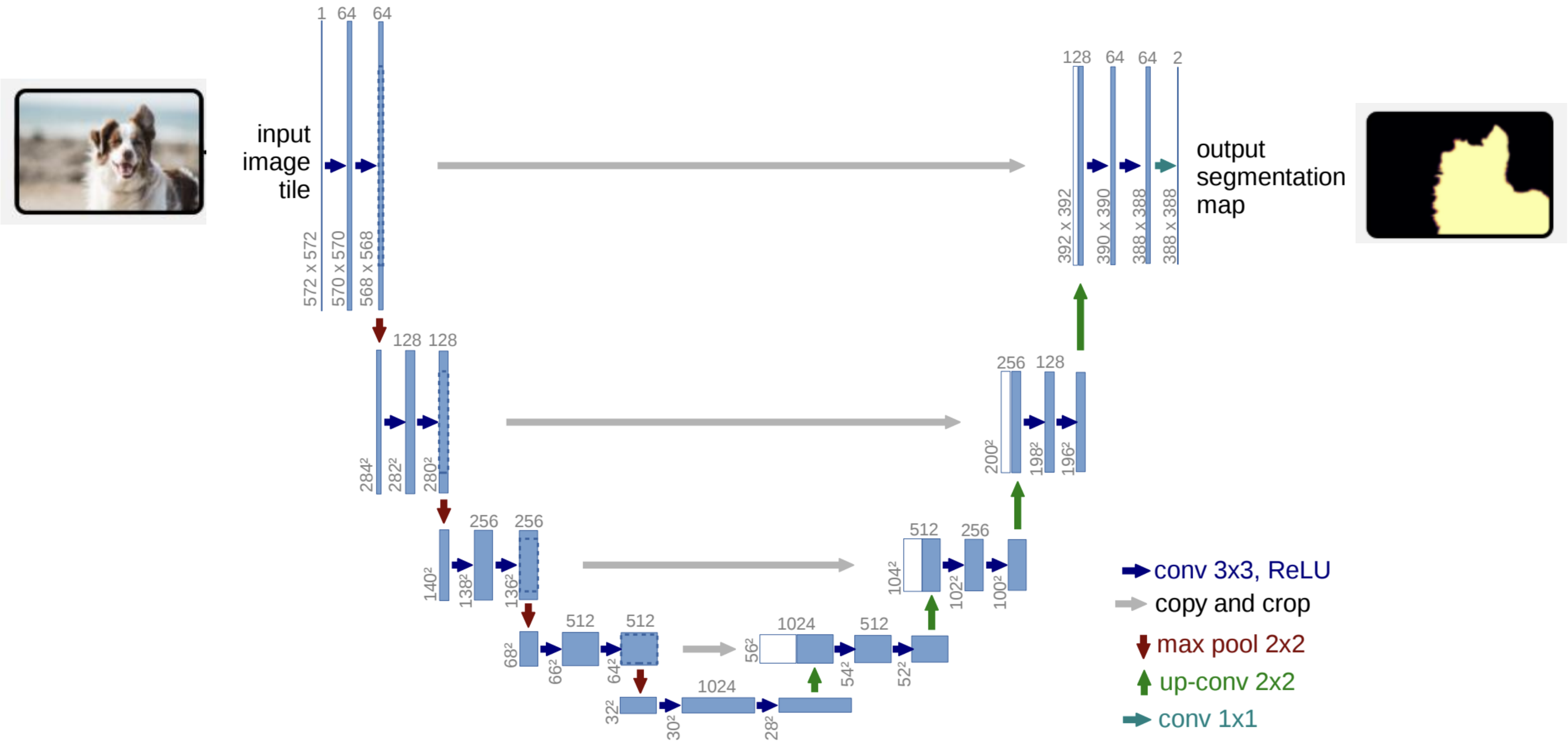


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Closer look:

- U shape
  - Down sampling
    - Increasing depth
    - Reducing resolution
  - Up-sampling
    - Reducing depth
    - Increasing resolution

- Q. Is there padding? Why
- Q. Starting with 572 and ending in 388?
- Cropping and adding – what the entire feature map mean?
- Q. What are last two channels at end

2. Popular architectures : U-net

The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function. The soft-max is defined as $p_k(\mathbf{x}) = \exp(a_k(\mathbf{x}))/\left(\sum_{k'=1}^{K} \exp(a_{k'}(\mathbf{x}))\right)$ where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(\mathbf{x})$ is the approximated maximum-function. I.e. $p_k(\mathbf{x}) \approx 1$ for the k that has the maximum activation $a_k(\mathbf{x})$ and $p_k(\mathbf{x}) \approx 0$ for all other k. The cross entropy then penalizes at each position the deviation of $p_{\ell(\mathbf{x})}(\mathbf{x})$ from 1 using

$$E = \sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x})) \tag{1}$$

We pre-compute the weight map for each ground truth segmentation to compensate the different frequency of pixels from a certain class in the training data set, and to force the network to learn the small separation borders that we introduce between touching cells (See Figure 3c and d).

The separation border is computed using morphological operations. The weight map is then computed as

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right) \tag{2}$$

2. Popular architectures : U-net

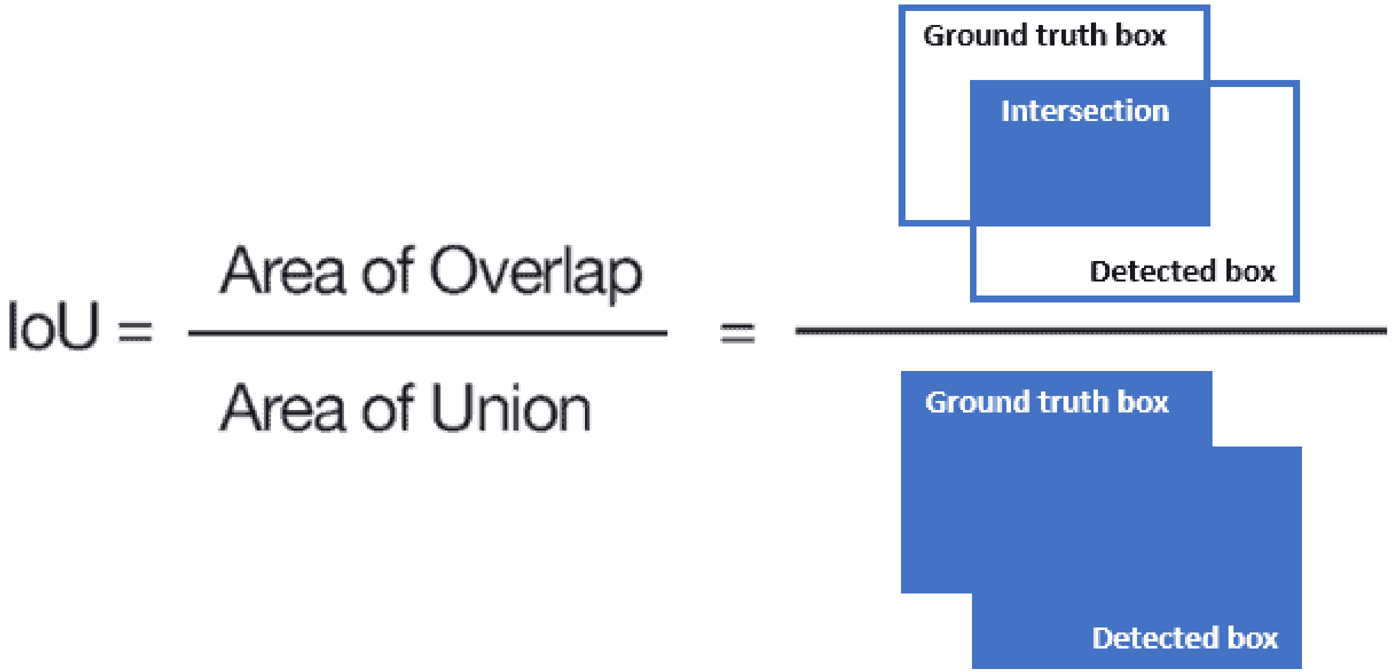
Table 2. Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

Crushing all the previous models!

Q. What is IOU?

2. Popular architectures : U-net



Higher IOU is better!

2. Popular architectures : U-net

- Hands on!
- Link to notebook to train and inference on atoms
- https://github.com/gduscher/MLSTEM2025/blob/main/Day4/AtomicSemanticSegmention.ipynb

2. Popular architectures : U-net

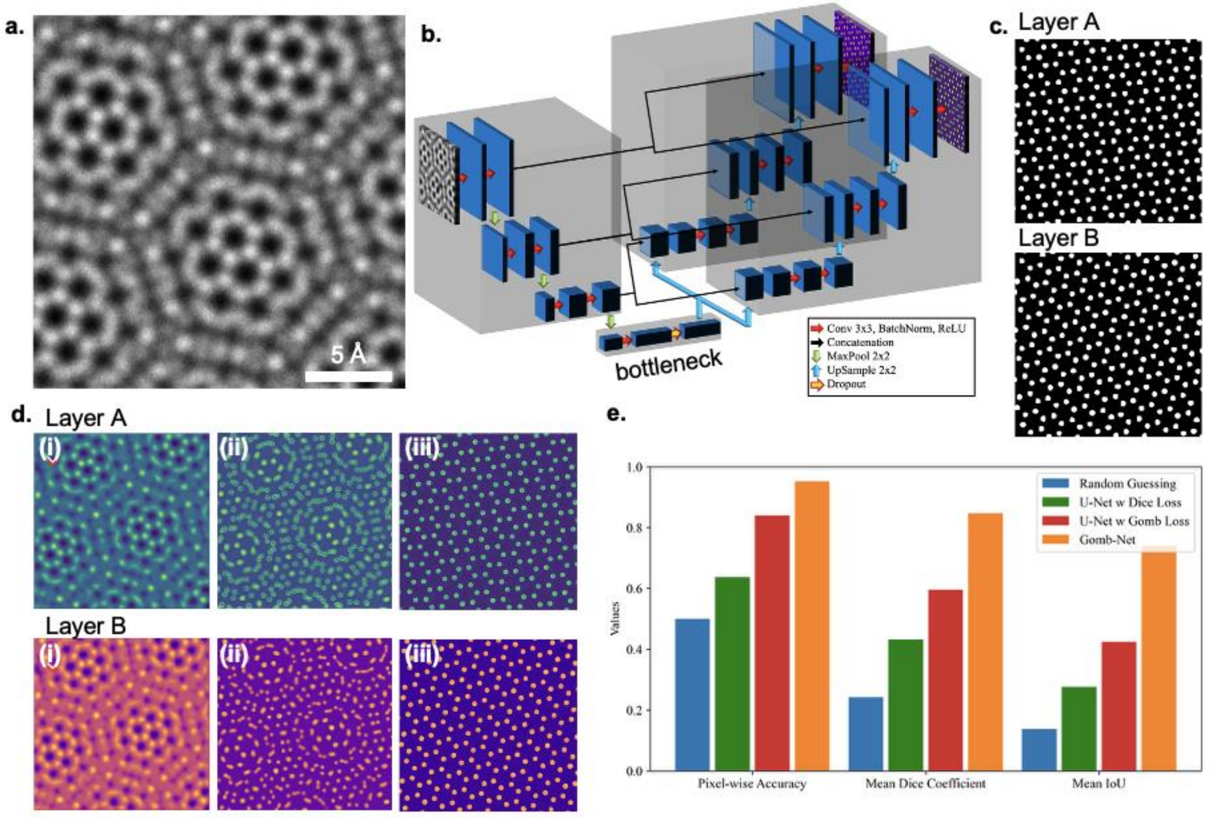


Figure 1. Atom-finding in Moiré lattices is enabled by Gomb-Net. a, Simulated STEM Z-contrast test image of atomically resolved twisted bilayer graphene, used as network input (not seen during training). **b,** Schematic of the network architecture with a single encoder before the bottleneck layer, and two separate decoders after the bottleneck layer. Skip connections (black) help preserve fine detail. **c,** Binary images showing the predicted carbon atom positions for each layer derived from the input. **d,** Raw network outputs from the test image for each layer using (i) U-Net, (ii) U-Net with the groupwise combinatorial loss function, and (iii) Gomb-Net. **e,** Performance comparison of the different networks on a test dataset of 800 images.

GOMB-Net : Modified U-net for Moiré Lattices

Austin C. Houston, Sumner B. Harris, Hao Wang, Yu-Chuan Lin, David B. Geohegan, Kai Xiao, and Gerd Duscher. *Atom identification in bilayer moiré materials with Gomb-Net*. arXiv:2502.09791 (2024).

2. Popular models: Foundational models

Definition: Foundation Models Foundation models are large Artificial Intelligence (ML) models trained on broad data that can:

- ➢ produce/generate wide variety of outputs.
- ➢ adapt to a wide range of downstream tasks.
- ➢ **generalize beyond** training data distributions

Future will be about using these models as major players like **Meta**, **Google**, **OpenAI** opensource them. After training them on hand labelled data for millions of dollars worth of **GPU time**

“The steam” engine era for **Neural Networks**

2. Popular models: Foundational models

- Bert (2018) – Google
- GPT-4, CLIP and all the OpenAI models
- Gemini – Google
- Segment Anything model – Meta
- And more!

Ideally we would want Foundational models to be also Opensource

2. Popular models: Foundational models - CLIP

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1] Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

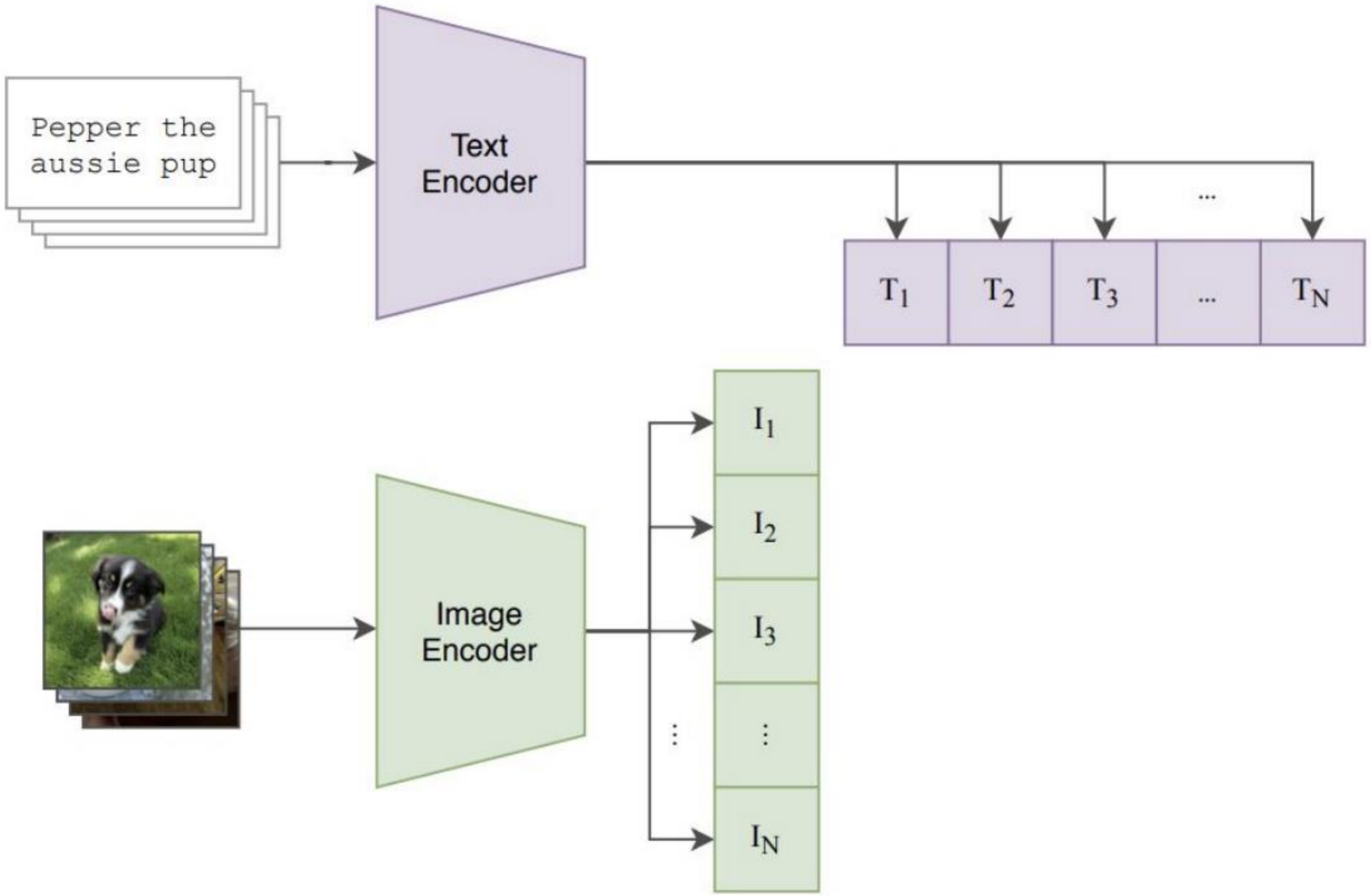
Introduced in 2021
Cited ~ 34405 till today

AKA
CLIP (Contrastive Language-Image Pretraining), Predict the most relevant text snippet given an image

Chatgpt demo!

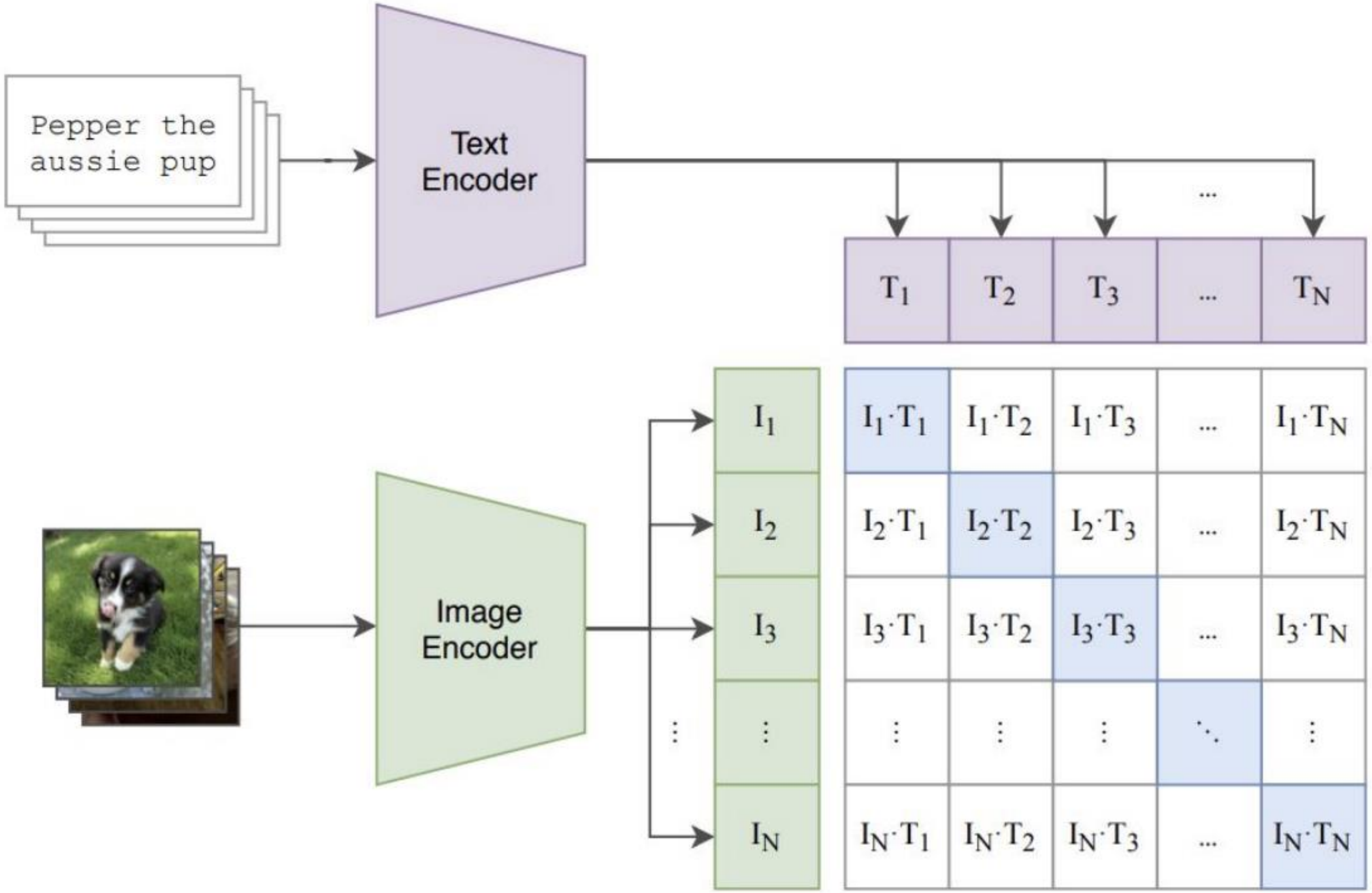
2. Popular models: Foundational models - CLIP

Pre-training



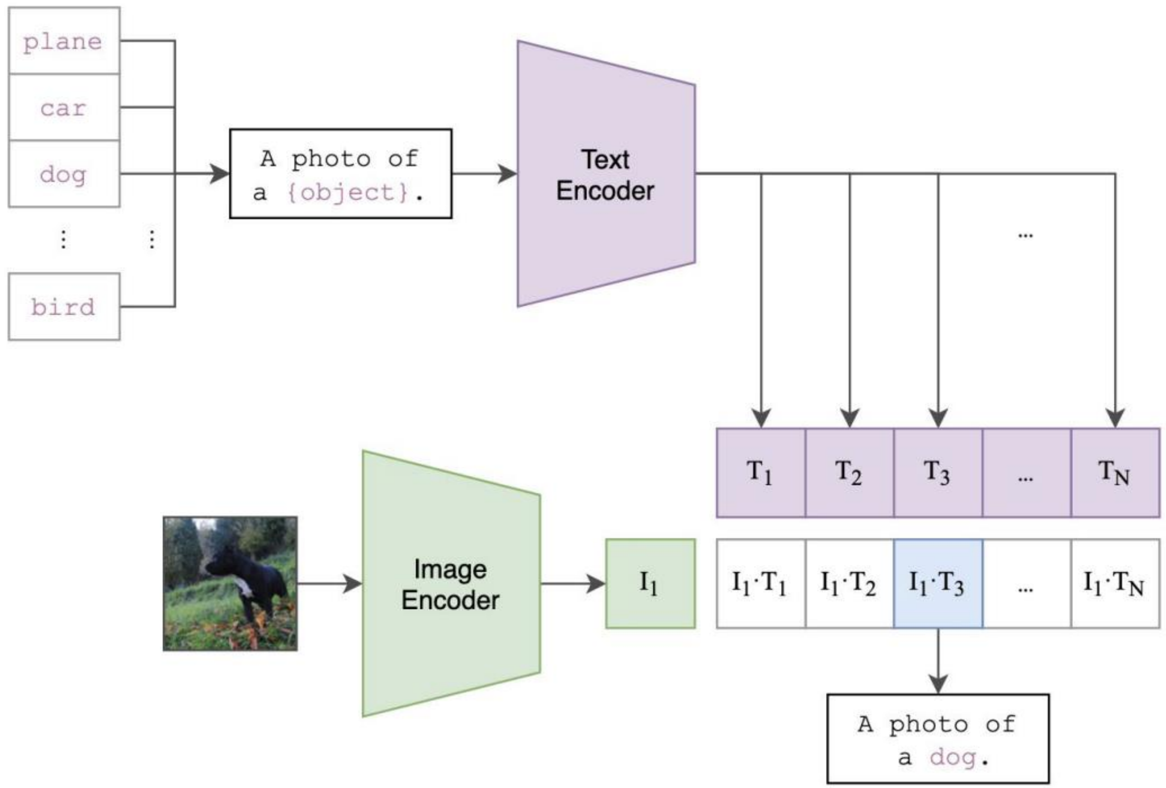
2. Popular models: Foundational models - CLIP

Pre-training



2. Popular models: Foundational models - CLIP

Zero-shot classification



2. Popular models: Foundational models - CLIP

Some CLIP details

Training

- Trained on 400M image-text pairs from the internet
- Batch size of 32,768
- 32 epochs over the dataset
- Cosine learning rate decay

Architecture

- ResNet-based or ViT-based image encoder
- Transformer-based text encoder

2. Popular models: Foundational models - CLIP

Q. How to get started?

Full code: https://github.com/openai/CLIP

Colab notebook - https://github.com/openai/CLIP/blob/main/notebooks/Interacting_with_CLIP.ipynb

2. Popular models: Foundational models - SAM

Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead
Meta AI Research, FAIR

Introduced in 2022
Cited ~ 10761 till today

Meta AI

2. Popular models: Foundational models - SAM

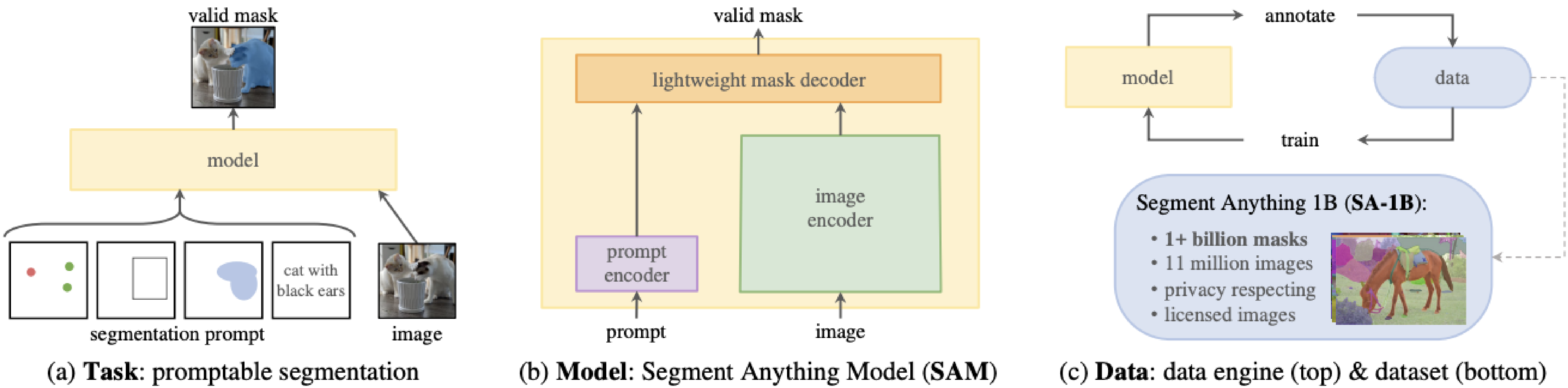


Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

2. Popular models: Foundational models - SAM

Q. How to get started?

Full code: https://github.com/facebookresearch/segment-anything

Examples: https://github.com/facebookresearch/segment-anything/tree/main/notebooks

2. Popular models: Foundational models - cellSAM

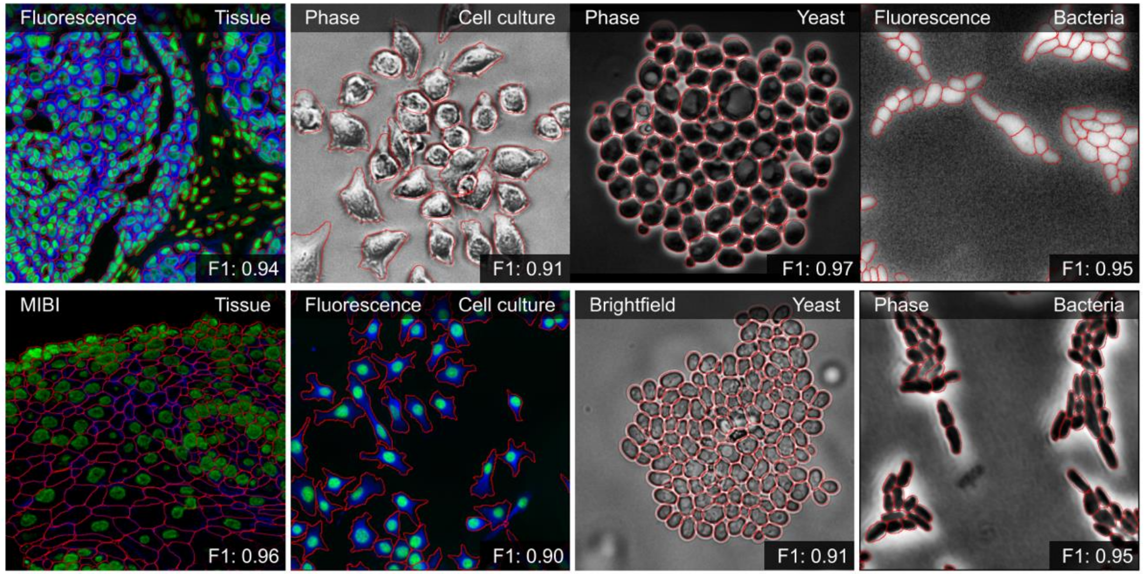
This is how Opensource helps for other domains – Eg: Biology, Material science

A Foundation Model for Cell Segmentation

Uriah Israel[1,3†], Markus Marks[2,3†], Rohit Dilip[3†], Qilin Li[2], Morgan Schwartz[1], Elora Pradhan[1], Edward Pao[1], Shenyi Li[1], Alexander Pearson-Goulart[1], Pietro Perona[2,3], Georgia Gkioxari[3], Ross Barnowski[1], Yisong Yue[3], David Van Valen[1,4*]

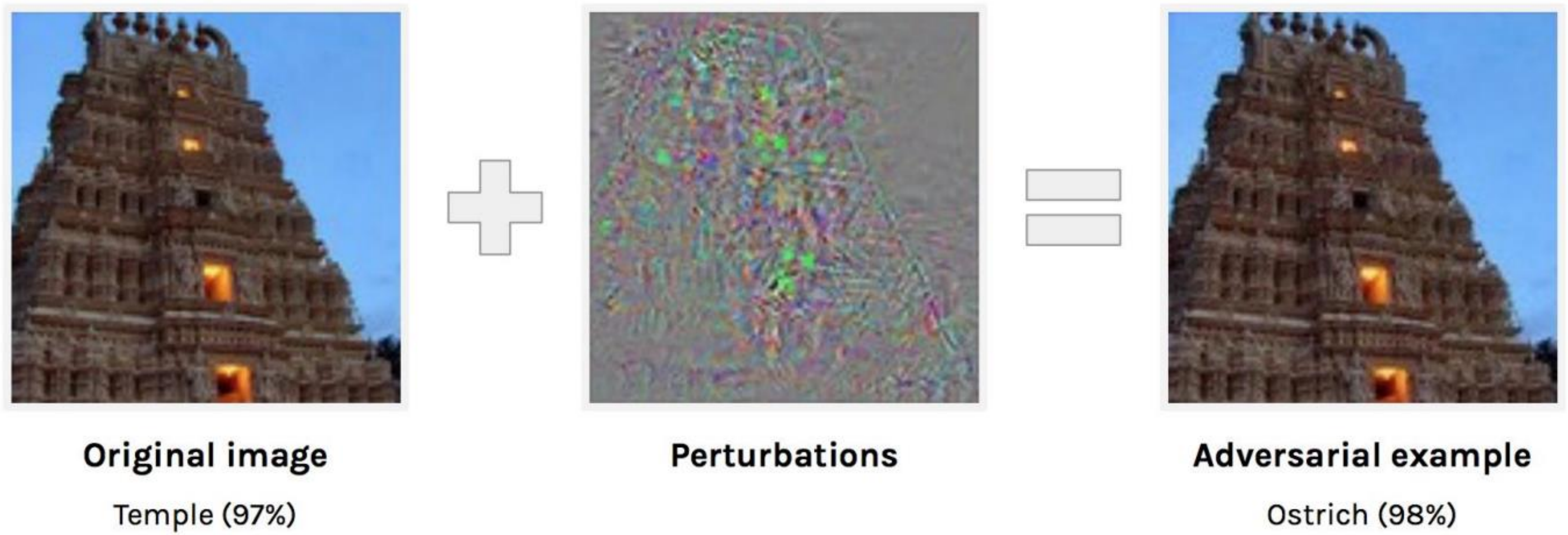
[1*]Division of Biology and Biological Engineering, Caltech.
[2]Division of Engineering and Applied Science, Caltech.
[3]Division of Computing and Mathematical Science, Caltech.
[4]Howard Hughes Medical Institute.

*Corresponding author(s). E-mail(s): vanvalen@caltech.edu;
Contributing authors: ulisrael@caltech.edu; marks@caltech.edu; rdilip@caltech.edu; qli2@caltech.edu; msschwartz@caltech.edu; epradhan@caltech.edu; epao@caltech.edu; sli5@caltech; pearsongoulart@gmail.com; perona@caltech.edu; georgia@caltech.edu; rossbar@caltech.edu; yyue@caltech.edu;
†These authors contributed equally to this work.



3. Explainability in Neural Networks - Motivation

Adversarial attack



https://course.ccs.neu.edu/ds4420sp20/slides/18-ds4420-interpretability.pdf

3. Explainability in Neural Networks - Problems

- We don’t trust the models
- We don’t know what happens in extreme cases
- Mistakes can be expensive / harmful
- Does the model make similar mistakes as humans ?
- How to change model when things go wrong ?

What do we want to get?

- Interactive feedback - can model learn from human actions in online setting ? (Can you tell a model to not repeat a specific mistake ?)
- Recourse – Can a model tell us what actions we can take to change its output ? (For example, what can you do to improve your credit score?)

3. Explainability in Neural Networks - Methods

- Attention maps – if using transformers
- Saliency maps
- SHAP: Shapely Additive explanations
- LIME

3. Explainability in Neural Networks - Attention

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Introduced in 2015
Cited ~ 39137 till today

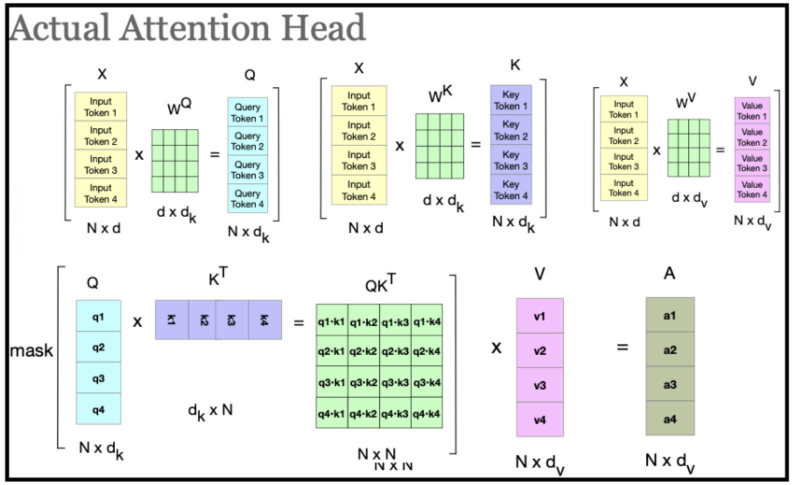
Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Introduced in 2017
Cited ~ 180700 till today

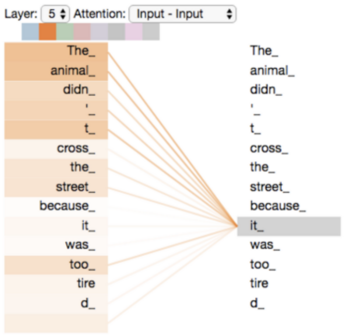
Not introduced as an explainability method early on

3. Explainability in Neural Networks - Attention



$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

"The animal didn't cross the street because it was too tired"



3. Explainability in Neural Networks - Attention

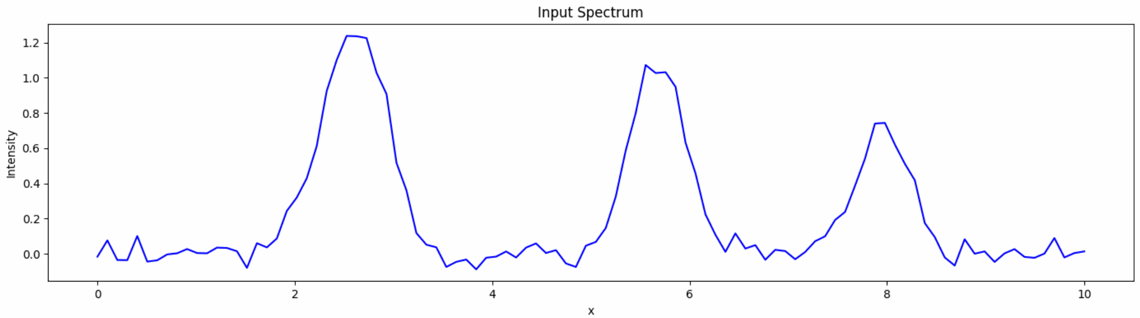
Data generation process: $f(x) = G_1(x) + G_2(x) + G_3(x)$

Where: $G_i(x) = A_i \cdot \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$

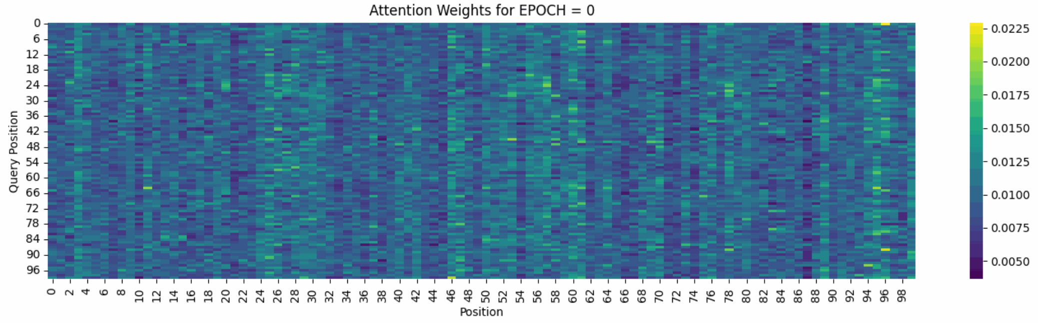
We input into the attention neural network: f(x)
Predict – Peak **A1**

Explainability methods - Attention

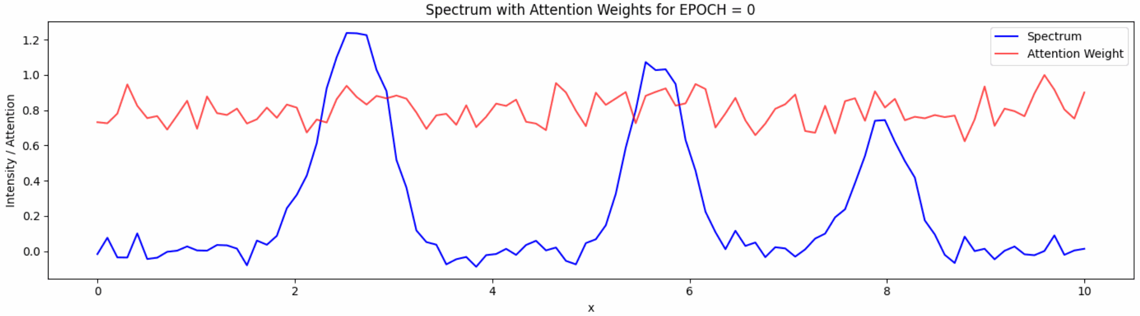
CG(x) – Cumulative signal



Attention map - learnable



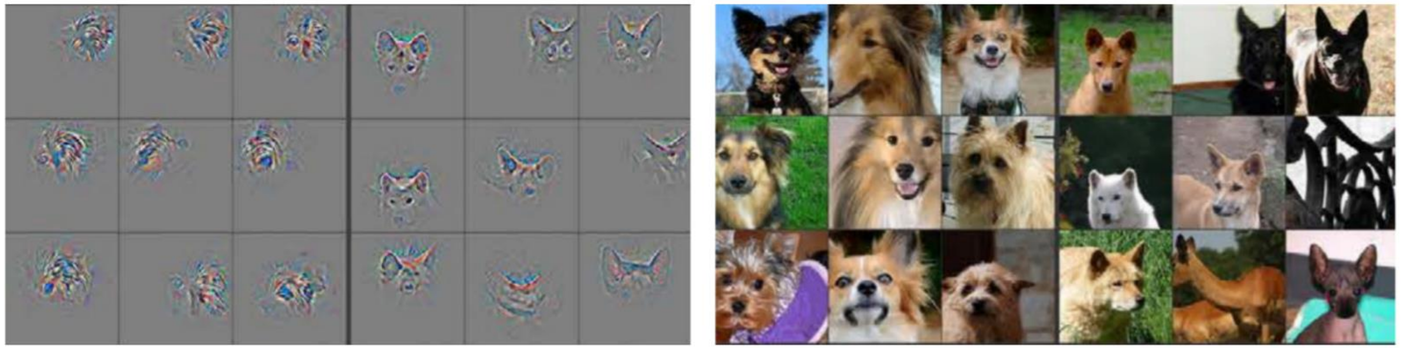
Attention network prediction



3. Explainability in Neural Networks - Saliency Maps

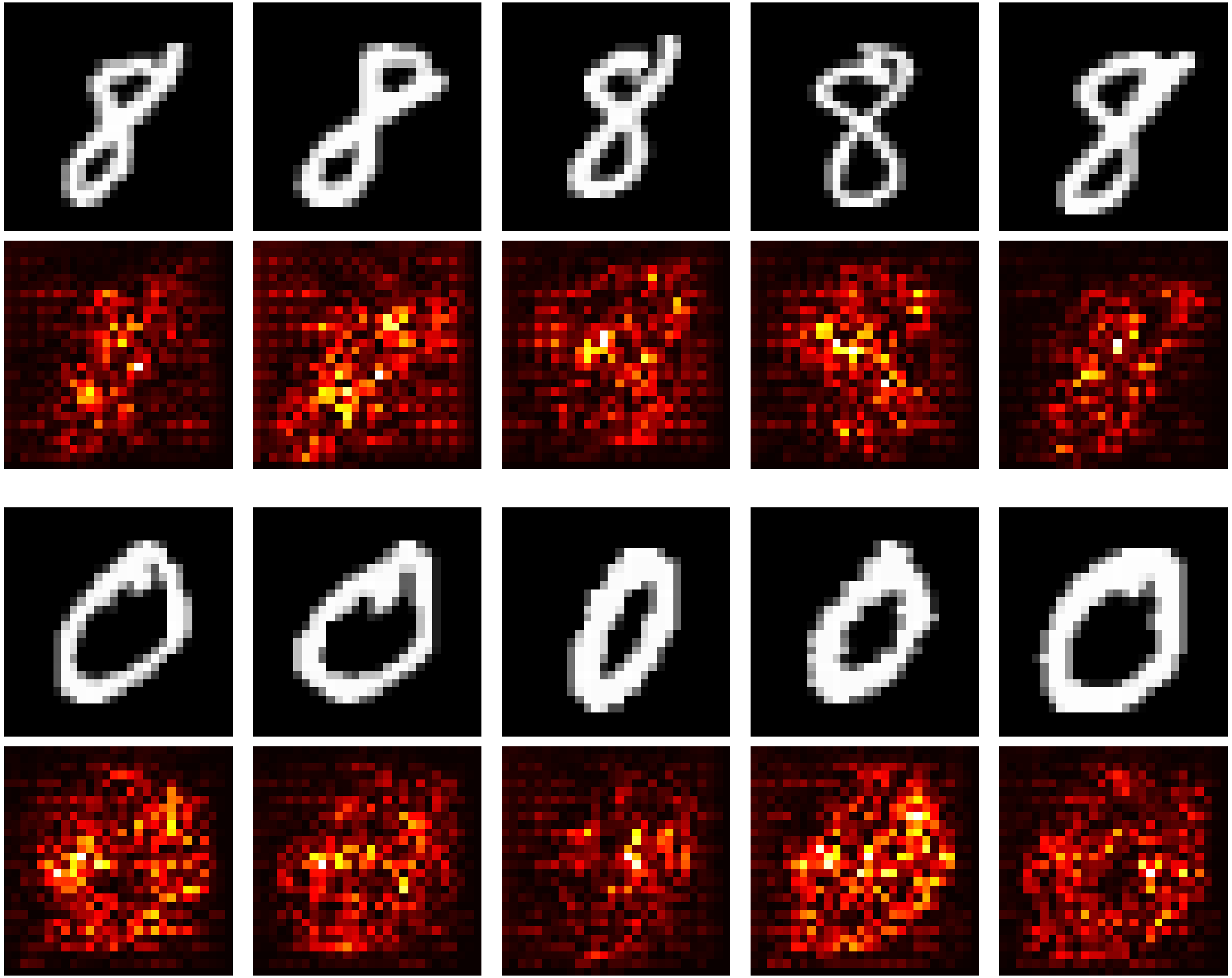
- Mean or aggregating Activation maps
- What is activation map?
  - feature maps of the image at different depths

Here are some of the results of the reconstruction of the input image from the activations of the fifth layer.

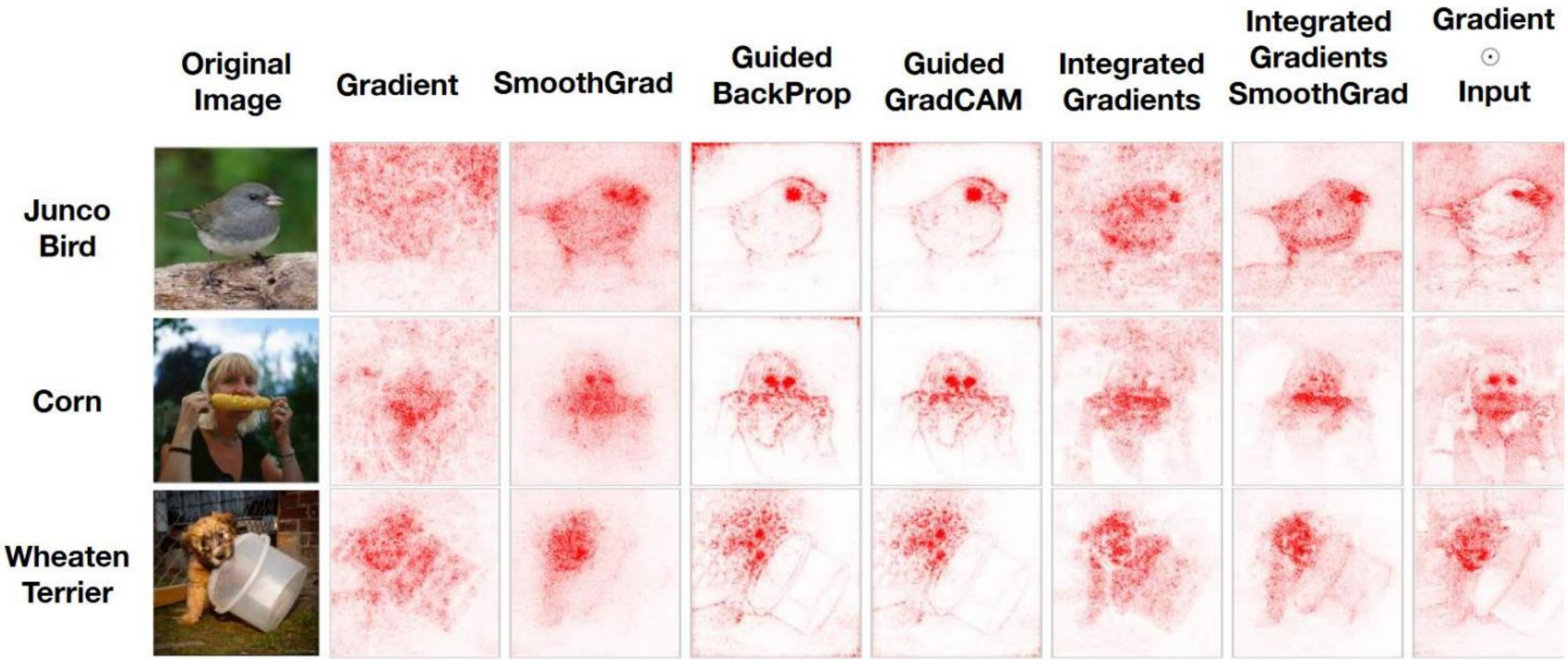


Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, 2013

Example of saliency maps for MNIST



There are many ways to get salience maps

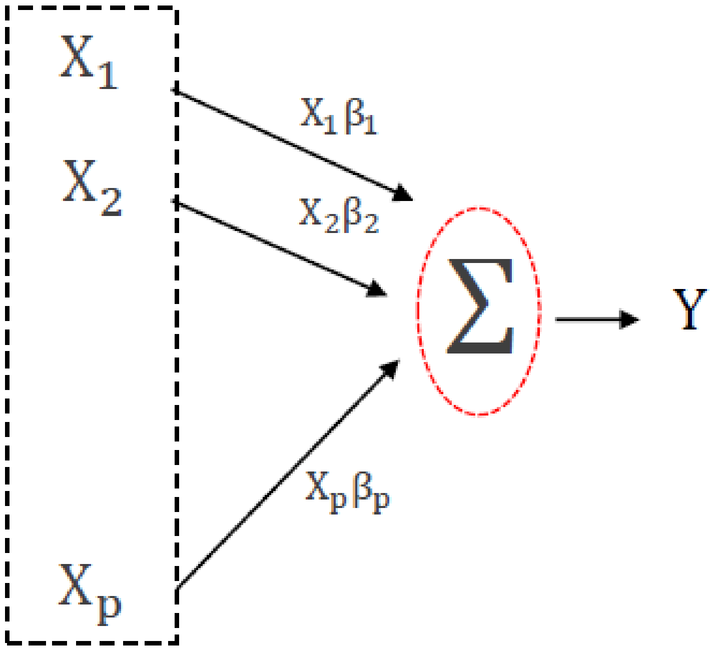


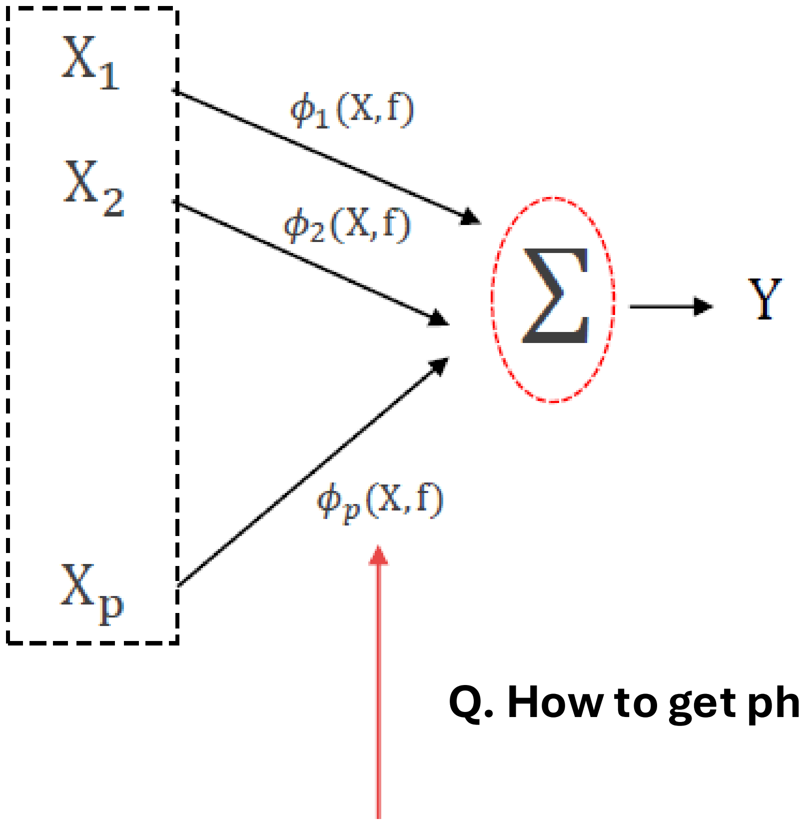
[Adebayo et al 2018]

- Only capture first order information
- Not very reliable

https://course.ccs.neu.edu/ds4420sp20/slides/18-ds4420-interpretability.pdf

3. Explainability in Neural Networks - SHAP





Q. How to get phi's?

function to attribute credit to the prediction

https://towardsdatascience.com/interpretable-machine-learning-using-shap-theory-and-applications-26c12f7a7f1a

3. Explainability in Neural Networks - SHAP

SHAP Value Interpretation:

1. Positive SHAP Value (> 0)

- **Feature increases the prediction.**
- In classification, this often means the feature pushes the prediction **towards the positive class**.
- In regression, this means the feature **increases the predicted output** compared to the baseline.

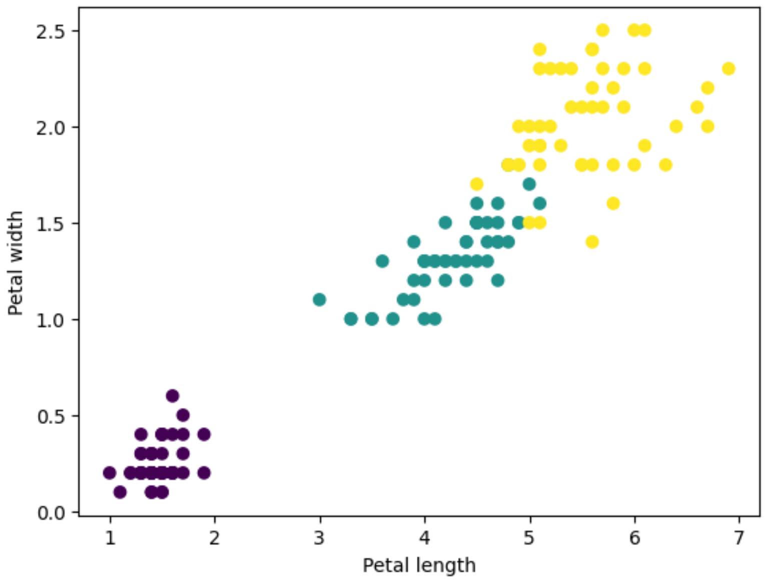
2. Negative SHAP Value (< 0)

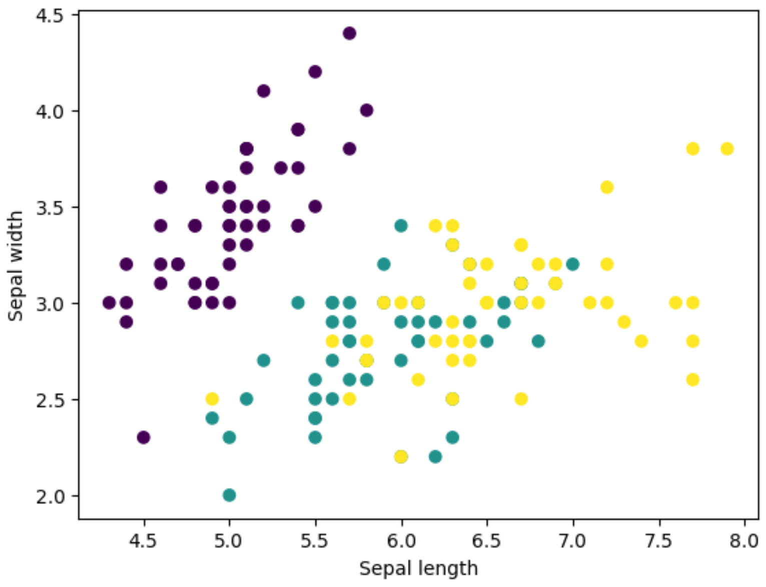
- **Feature decreases the prediction.**
- In classification, this often means the feature pushes the prediction **towards the negative class**.
- In regression, this means the feature **reduces the predicted output** compared to the baseline.

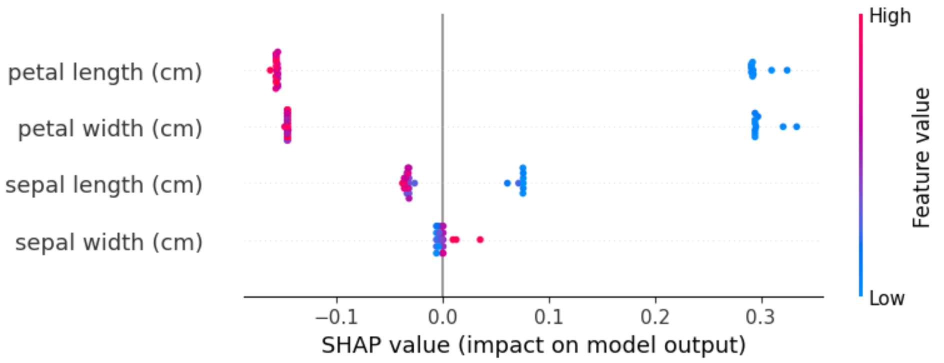
3. SHAP Value Close to Zero (≈ 0)

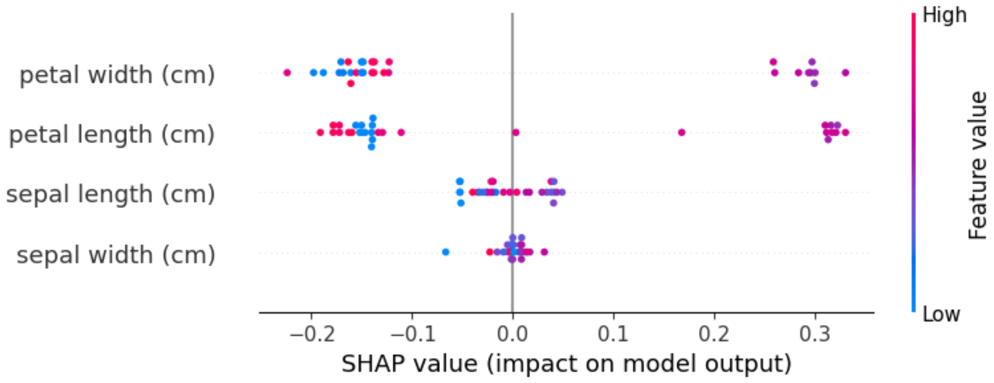
- **Feature has little to no effect** on the model's prediction for this instance.

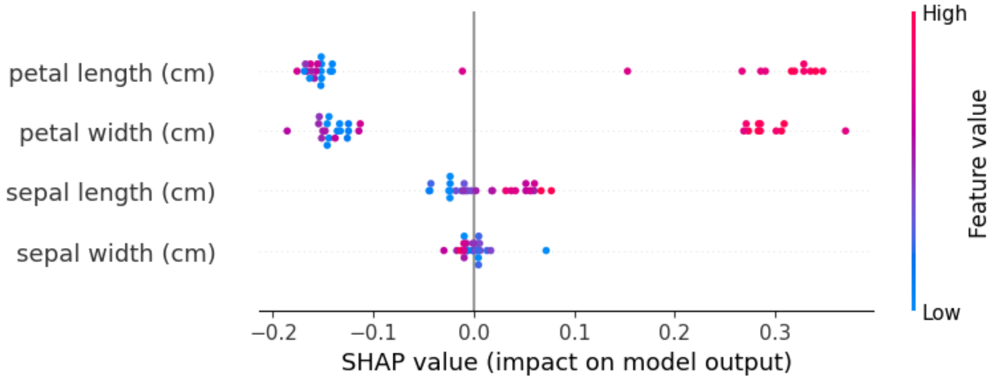
3. Explainability in Neural Networks - SHAP



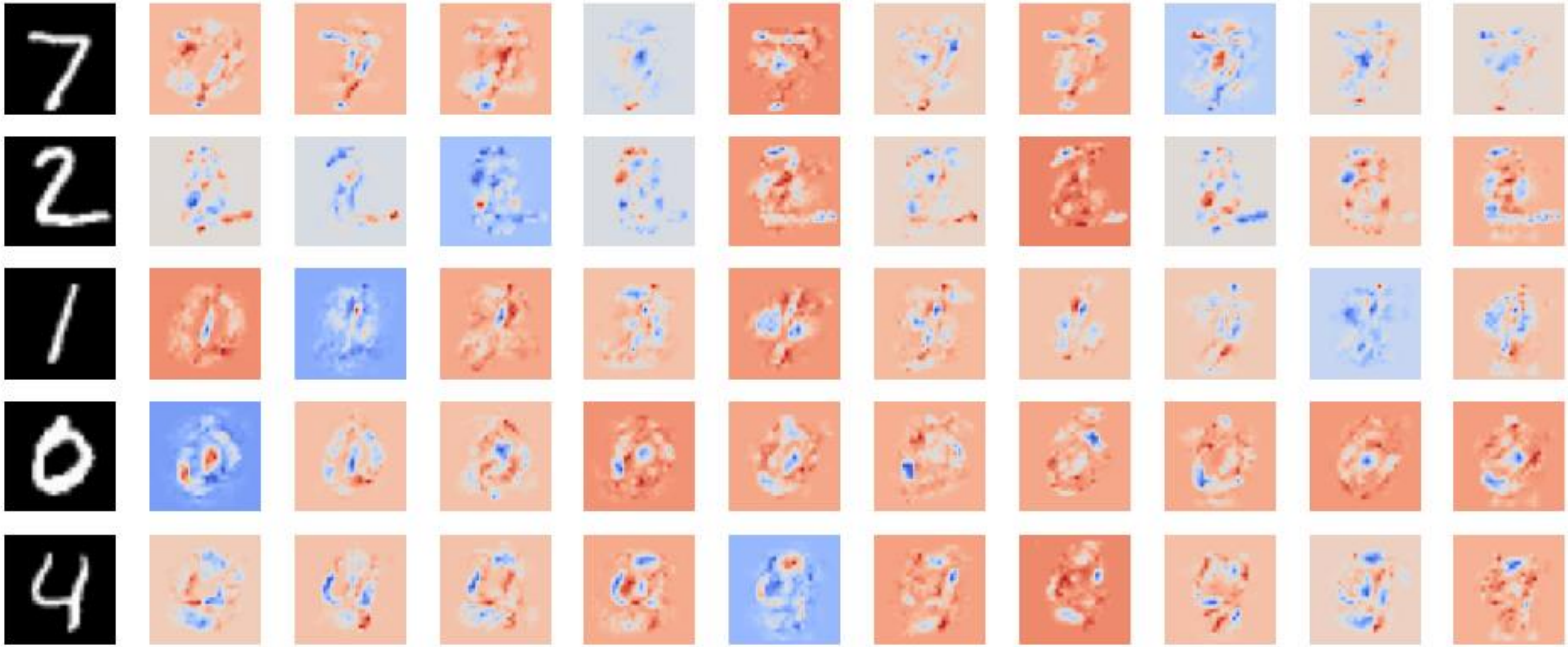








3. Explainability in Neural Networks - SHAP



Thank you for your attention!

2. Foundational model : Vision transformer

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

Introduced in 2021
Cited ~ 62345 till today

Q. Why was transformer important?

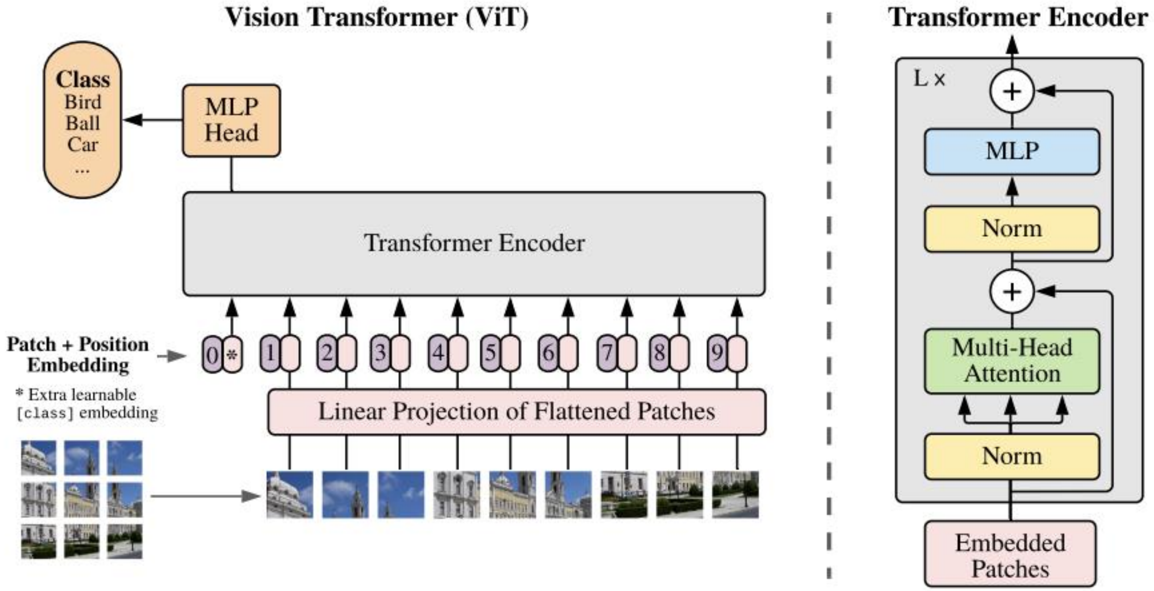


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).