Mechanizm uwagi

Transformery

Standardowa warstwa z uwagą

Wejście: dwie sekwencje $X \in \mathbb{R}^{T \times D}$ i $X' \in \mathbb{R}^{T' \times D}$

Projekcja do sekwencji zapytań Q (*queries*), K kluczy (*keys*) i V wartości (*values*)

$$Q = XW_Q \qquad K = X'W_K \qquad V = X'W_V$$

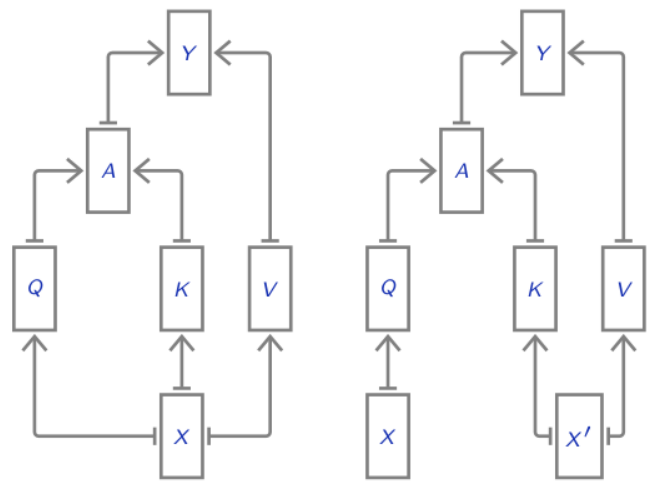
gdzie $W_Q \in \mathbb{R}^{D \times d_k}$, $W_K \in \mathbb{R}^{D \times d_k}$, $W_V \in \mathbb{R}^{D \times d_v}$

Macierz uwagi $A \in \mathbb{R}^{T \times T'}$

$$A = \text{softmax}_{\text{row}}\left(\frac{QK^\top}{\sqrt{D}}\right)$$

Sekwencja wyjściowa

$$Y = AV$$

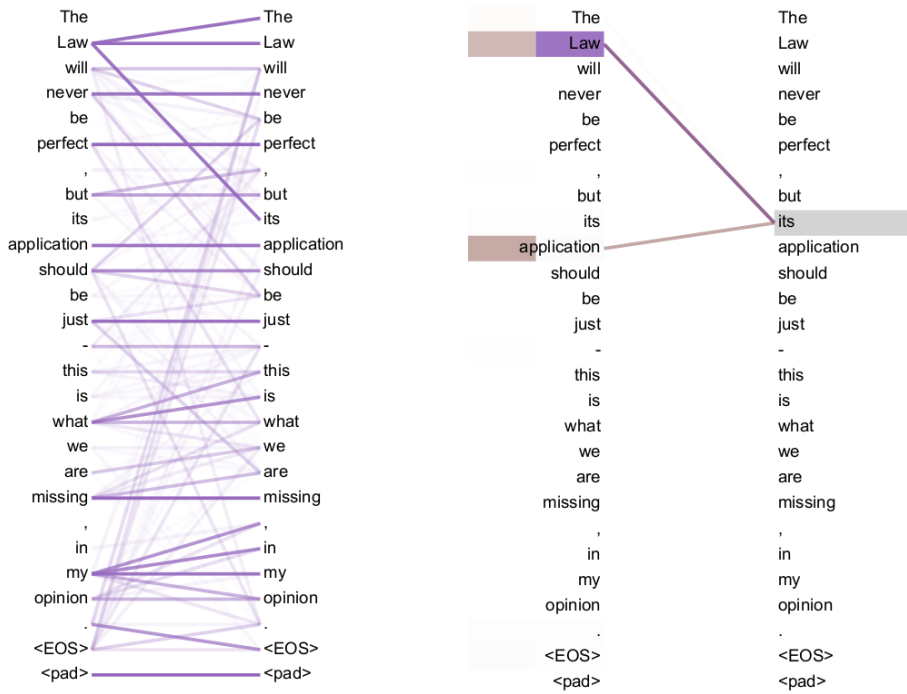


- samo-uwaga (*self attention*), jeżeli $X = X'$
- uwaga krzyżowa (*cross attention*), jeżeli $X \neq X'$

Uwaga to wszystko czego potrzebujesz

Uwaga jest uśrednieniem wartości W skojarzonych z kluczami K pasującymi do zapytań Q

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



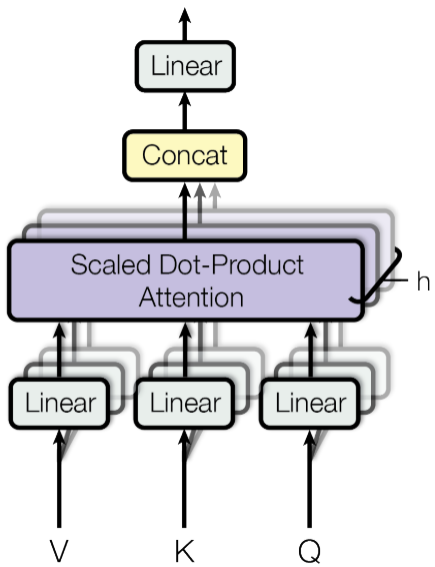
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.

Uwaga o wielu głowach (*Multi-head attention*)

Uwaga z wieloma głowicami pozwala skupić uwagę modelu na informacji zawartych w różnych reprezentacjach (osobne projekcje Q_i, K_i, V_i) i w różnych pozycjach sekwencji

$$\text{MultiHead}(Q, K, V) = \text{Concat}\,(Y_1, \ldots, Y_{\text{h}})\, W_H$$

gdzie $W_H \in \mathbb{R}^{hd_v \times D}$



W przypadku standardowego transformera ustala się $d_k = d_v = D/h$, dzięki redukcji wymiar każdej głowy całkowity koszt obliczeń pozostaje zbliżony do pojedynczej warstwy uwagi o pełnej wymiarowości

Standardowa operacja uwagi (attention) jest **niezmiennicza** względem permutacji kluczy i wartości:

$$\text{Attention}(Q, \sigma(K), \sigma(V)) = \text{Attention}(Q, K, V)$$

oraz **ekwiwariantna** względem permutacji zapytań, tzn. wynikowy tensor jest permutowany w ten sam sposób:

$$\text{Attention}(\sigma(Q), K, V) = \sigma(\text{Attention}(Q, K, V))$$

gdzie $\sigma(X)$ oznacza permutację wierszy macierzy

W konsekwencji *self-attention* i *cross-attention* są ekwiwariantne względem permutacji X, a *cross-attention* jest niezmiennicza względem permutacji X'.

Dlaczego samo-uwaga?

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

n - długość sekwencji, d - wymiar reprezentacji, k - rozmiar filtru splotu

- warstwy *self-attention* są szybsze od warstw rekurencyjnych gdy długość sekwencji n jest mniejsza od wymiarowości reprezentacji d

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.

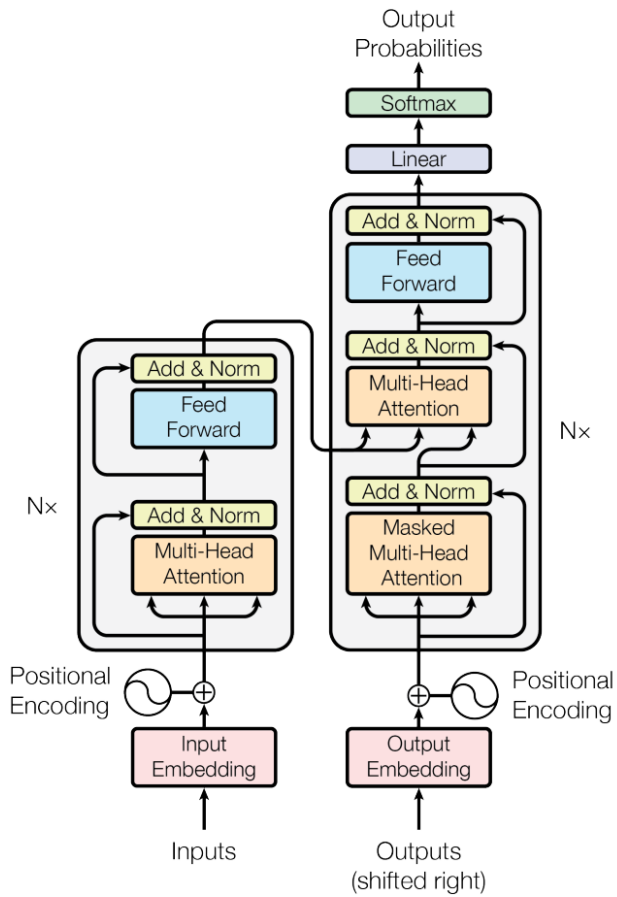
Transformer (Vaswani, 2017)

Transformery używają samo-uwagi do modelowania sekwencji w czyto jednokierunkowej sieci (bez warstw rekurencyjnych). Stosowane do budowy modelu językowych np. BERT, GPT

Pierwotny model $N = 6$,
dla GPT-3 $N = 93$

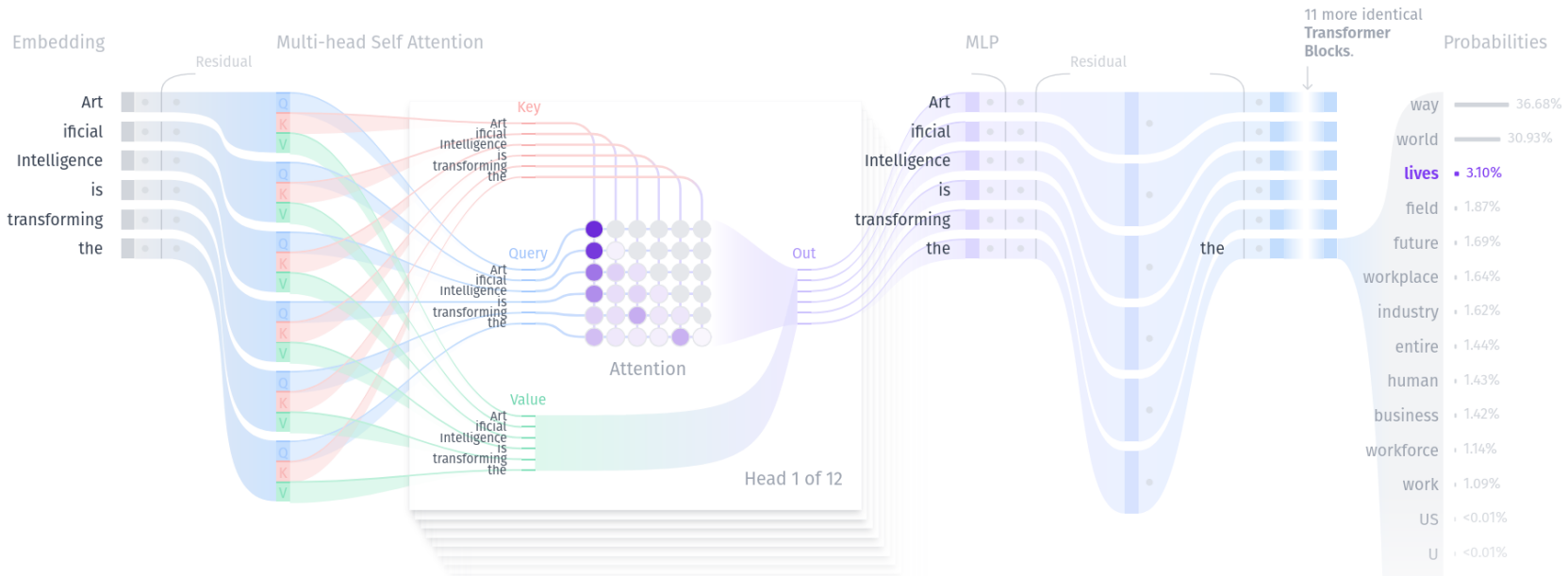
Enkoder: bloki kodujące z samo-uwagą, połączenia skrótowe między warstwami

Dekoder: bloki dekodujące z samo-uwagą, w każdym bloku dodatkowa warstwa z uwagą krzyżową ponad wyjściem bloków enkodera, działa w trybie autoregresji - przewiduje kolejne słowo używając poprzednie słowo jako wejście



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.

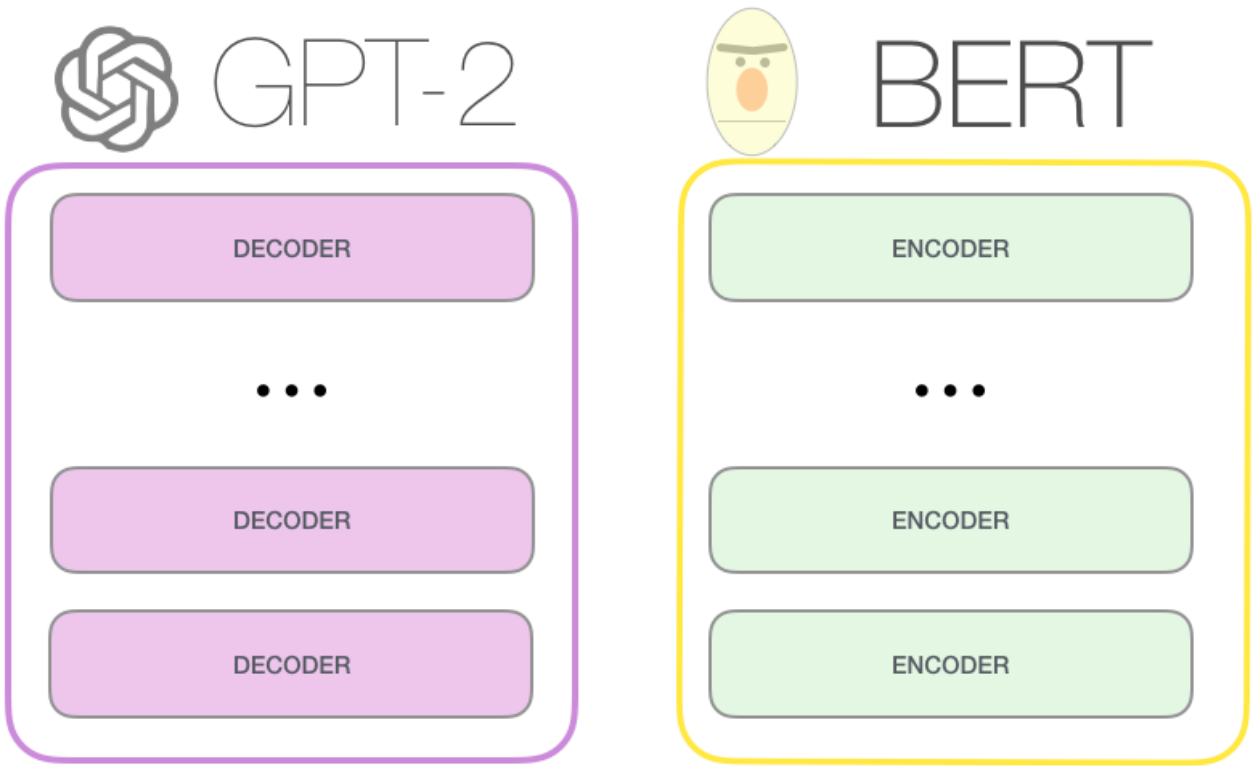
Transformer



☞ Transformer Explainer

BERT i GPT

- BERT (Bidirectional Encoder Representations from Transformers) od Google (2019)
- GPT (Generative Pre-trained Transformer) od OpenAI



☞ *Attentional neural networks*

Visual Transformer (ViT)

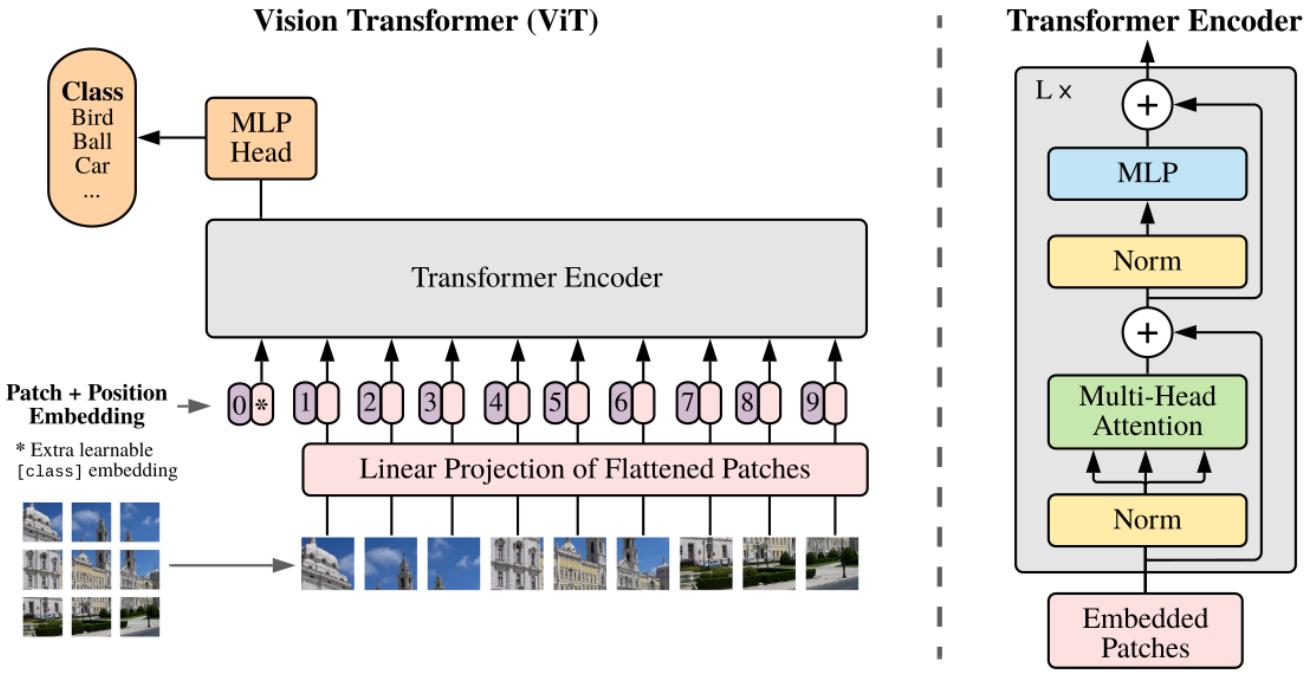


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.

ViT

Wejście: sekwencja płaskich łatek 2D obrazu

$$\mathbf{x} \in \mathbb{R}^{H \times W \times C} \qquad \longrightarrow \qquad \mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$$

$$\text{where } N = HW / P^2$$

Liniowa transformacja do reprezentacji D wymiarowej: (patch embeddings)

$$\mathbf{z}_0 = \left[\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots ;\, \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{pos}$$

$$\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D} \qquad \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$$

Enkoder transformera

$$\mathbf{z}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \ldots L$$

$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_\ell \ell\right)\right) + \mathbf{z}'_\ell, \quad \ell = 1 \ldots L$$

Głowa MLP klasyfikująca

$$\mathbf{y} = \text{LN}\left(\mathbf{z}_L^0\right)$$

Image Classification: state of the art

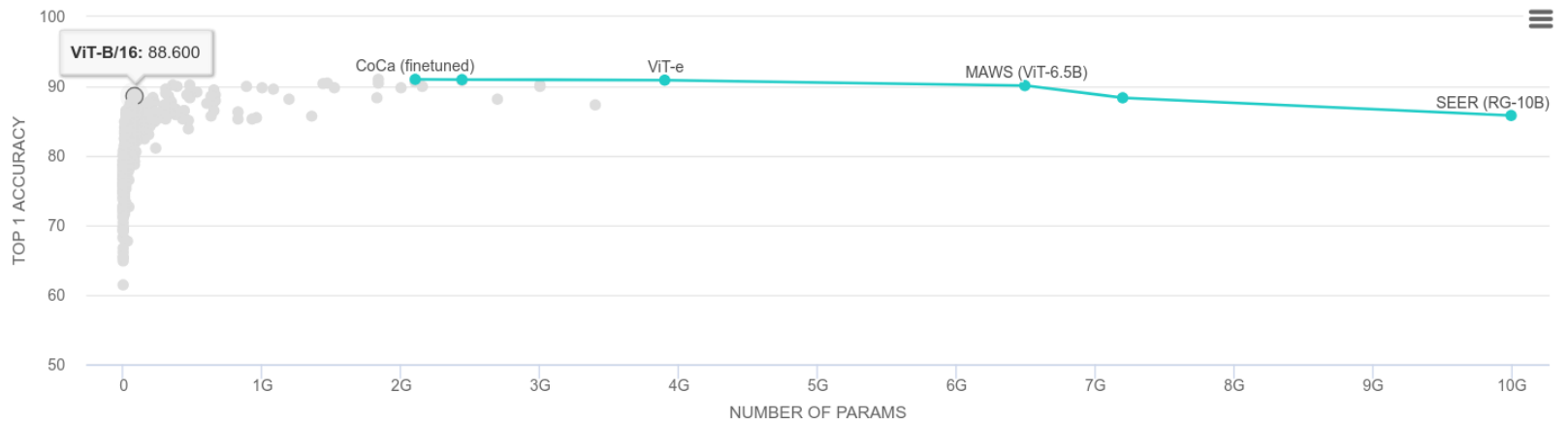
Trend	Dataset	Best Model	Paper	Code	Compare
	ImageNet	CoCa (finetuned)			See all
	CIFAR-10	ViT-H/14			See all
	CIFAR-100	EffNet-L2 (SAM)			See all
	STL-10	VIT-L/16 (Spinal FC)			See all
	ObjectNet	CoCa			See all
	MNIST	Branching/Merging CNN + Homogeneous Vector Capsules			See all
	iNaturalist 2018	OmniVec2			See all
	SVHN	E2E-M3			See all
	ImageNet ReaL	Baseline (ViT-G/14)			See all
	Clothing1M	LRA-diffusion (CC)			See all

☞ Browse State-of-the-Art

Image Classification on ImageNet

Leaderboard | Community Models | Dataset

View: Top 1 Accuracy | by: Number of params | for: All models



☞ Image classification on Imagenet