موضوع

SAM (Segment Anything Model)

Instance Segmentation

Segment Anything

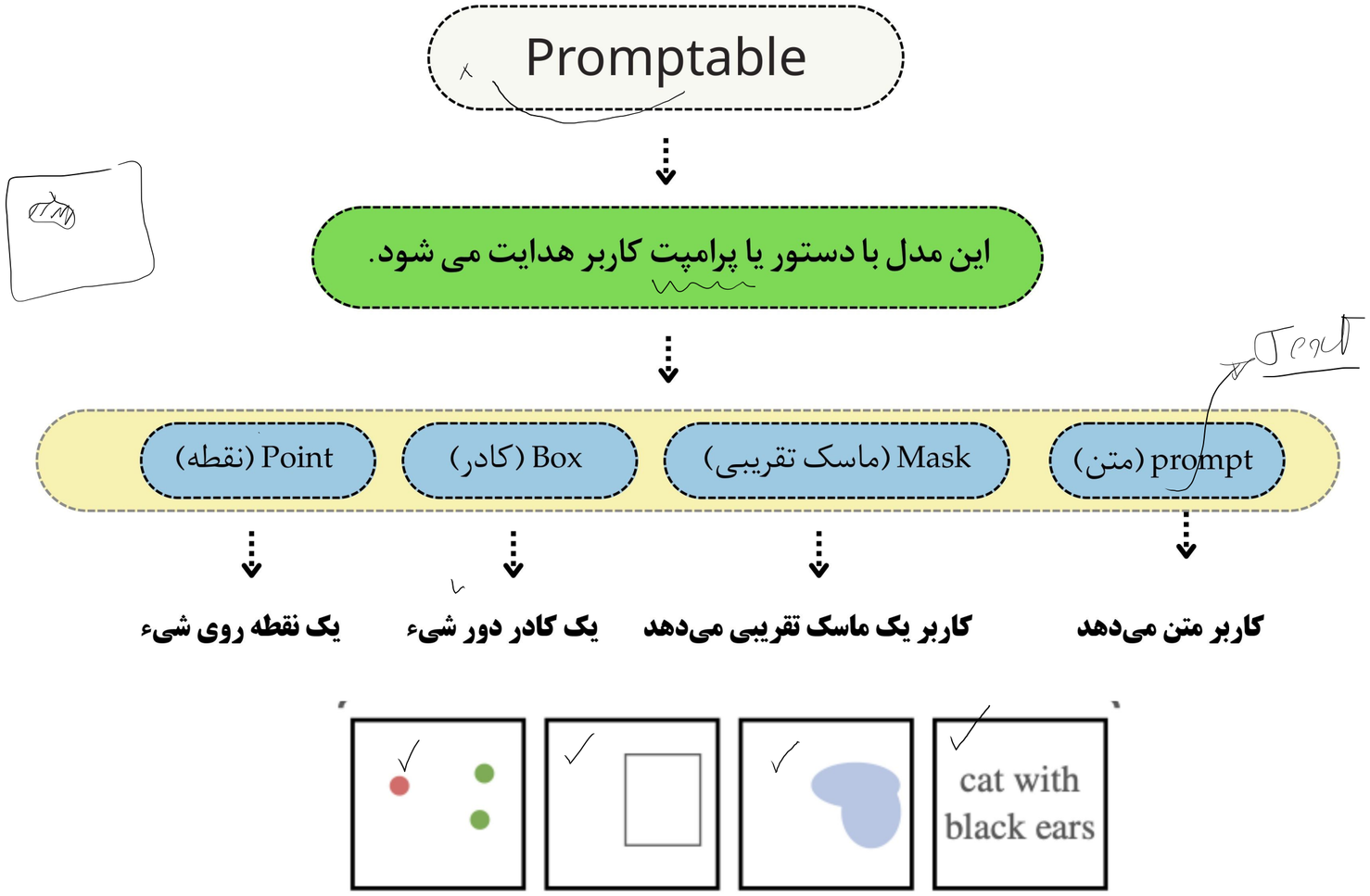
Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead
Meta AI Research, FAIR

SAM 1

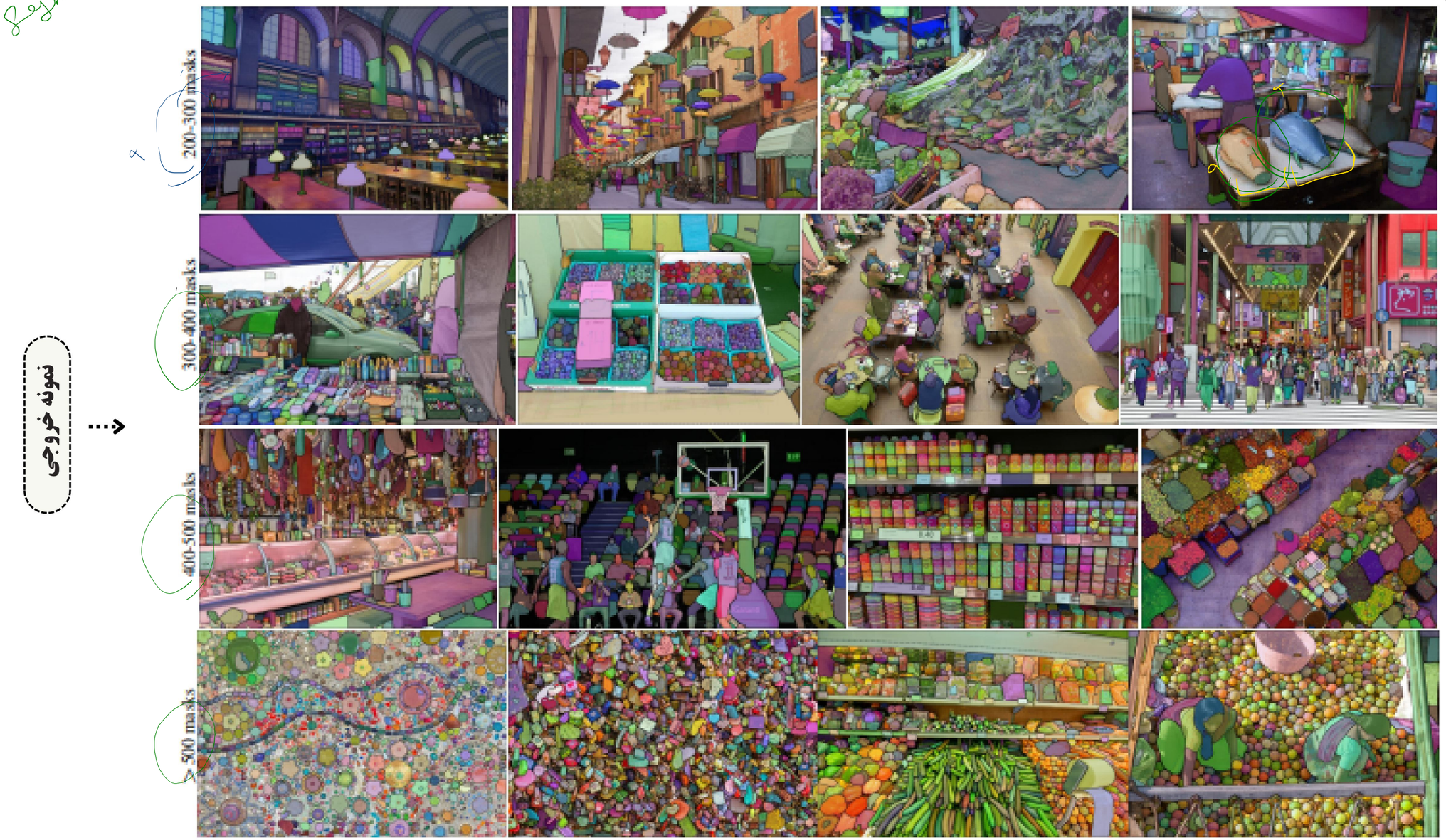
هدف

امکان تقسیم‌بندی (segmentation) هر شیء در هر تصویر فقط با دادن یک prompt
(نقطه، کادر، ماسک، یا حتی متن)

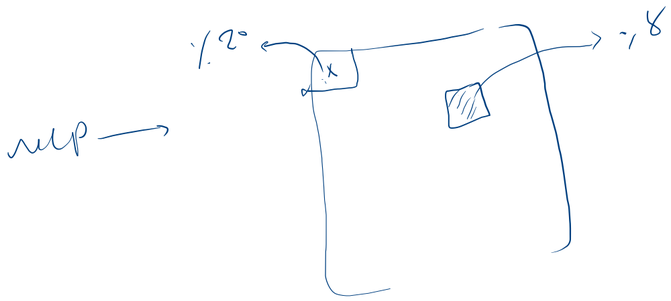
ویژگی های SAM 1

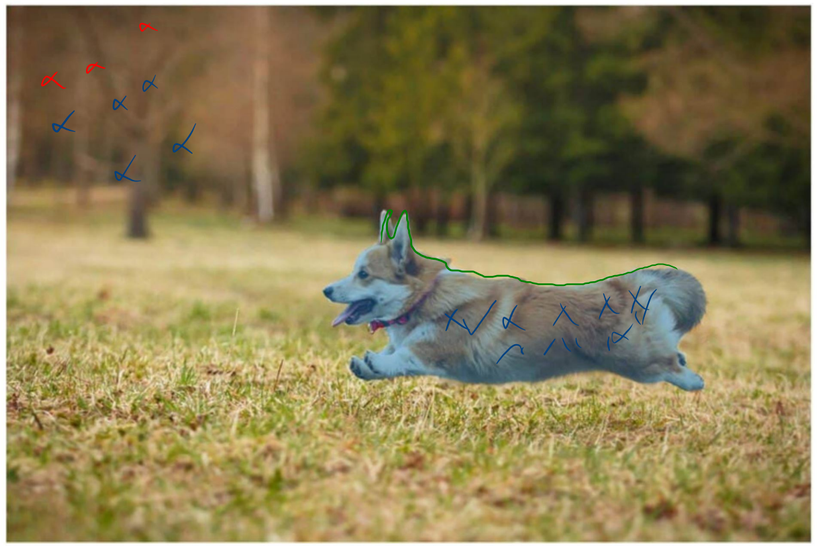


ویژگی های SAM 1



ویژگی های SAM 1





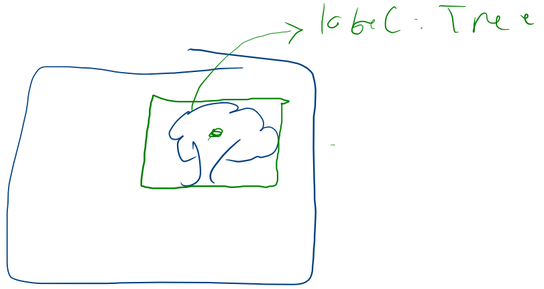
توانمندی های SAM1

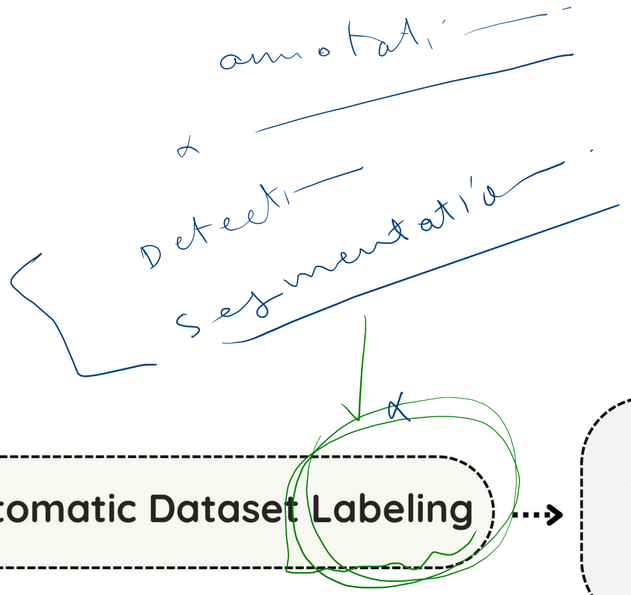
Interactive Segmentation

کاربر یک نقطه، کادر یا ماسک می‌دهد،
SAM بلافاصله شیء را جدا می‌کند.

https://segment-anything.com/assets/section-1.1a.mp4

توانمندی های SAM1



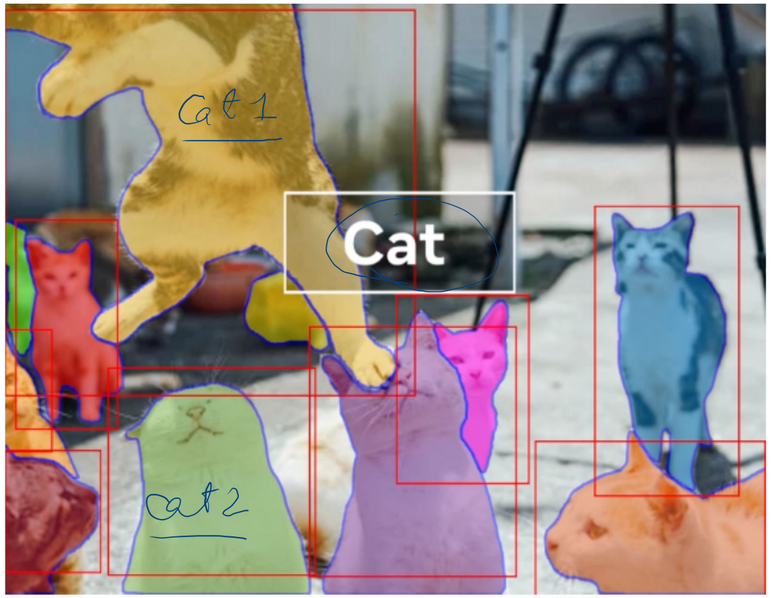


Automatic Dataset Labeling ···→ ماسک‌کردن خودکار تصاویر بزرگ، برای دیتاست‌های آموزشی

توانمندی های SAM1

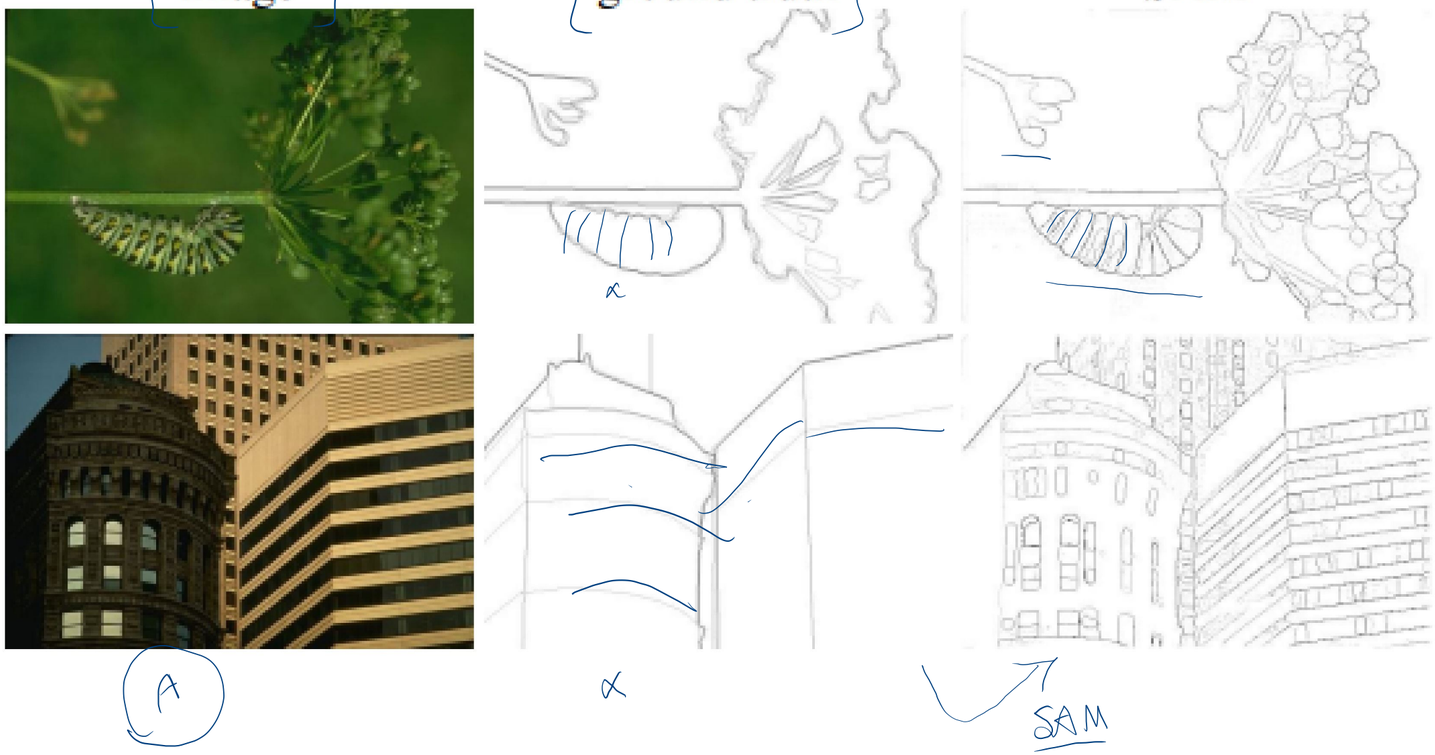
Instance Segmentation

جداسازی اشیاء جداگانه در تصویر
(مثل جداکردن هر فرد در جمعیت).

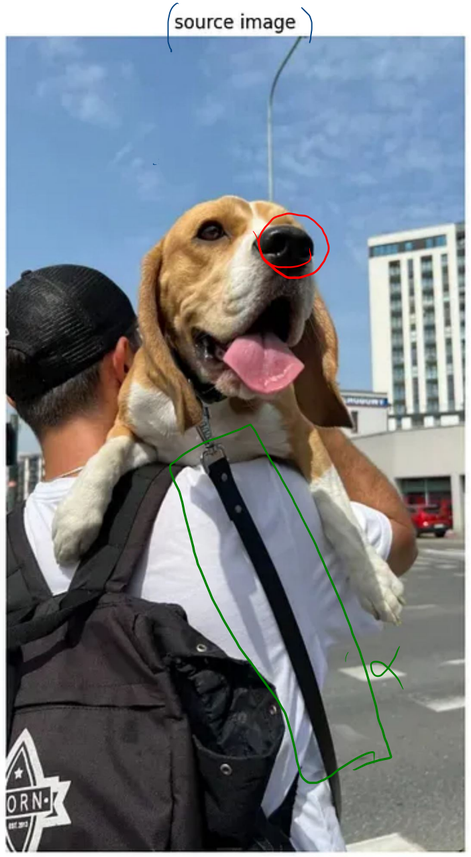


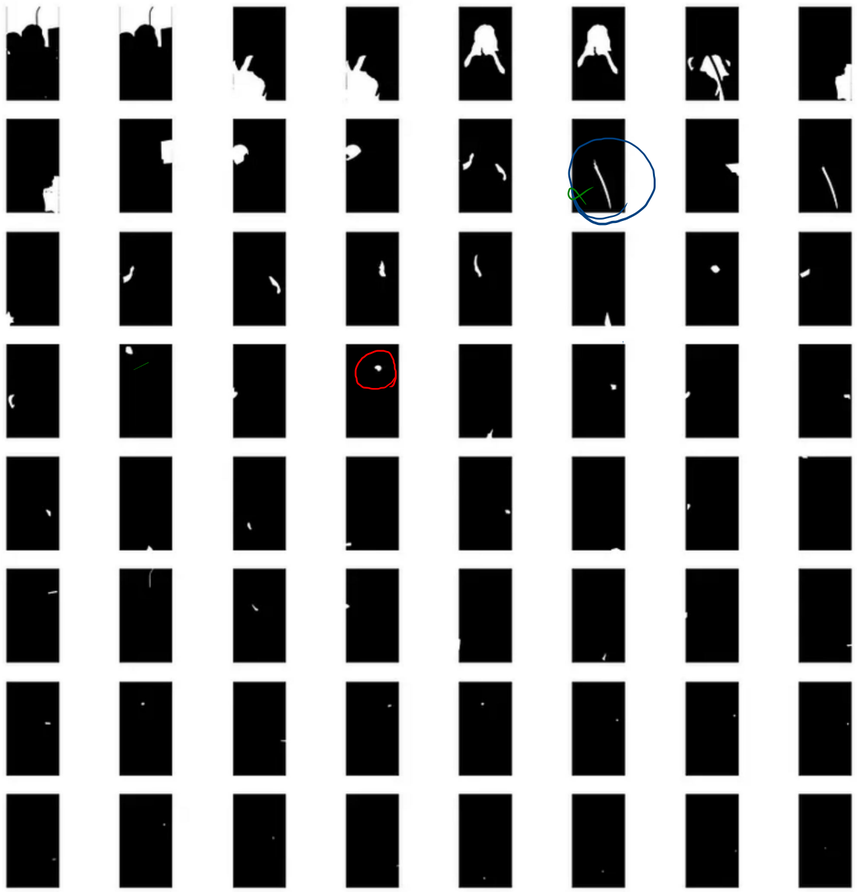
توانمندی های دیگر SAM1

Edge Detection

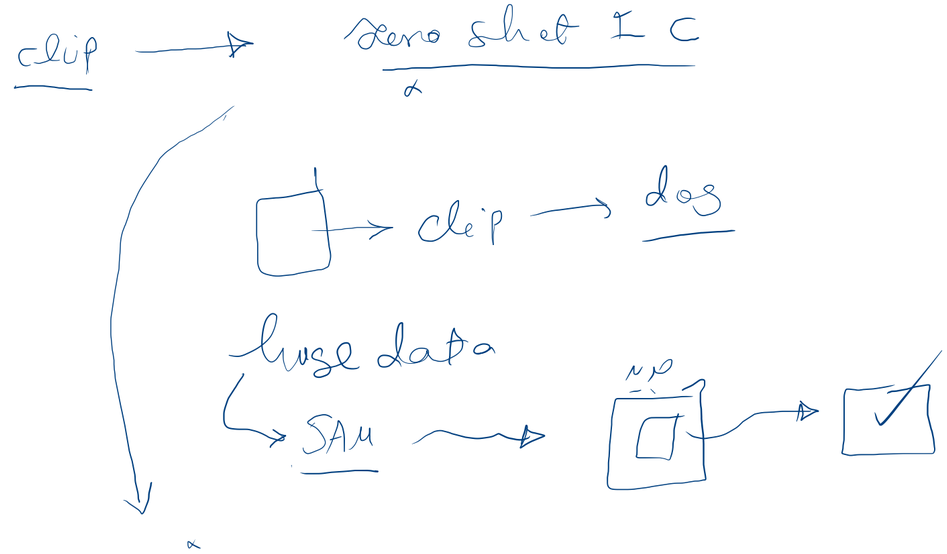


ماسک در سگمنتیشن



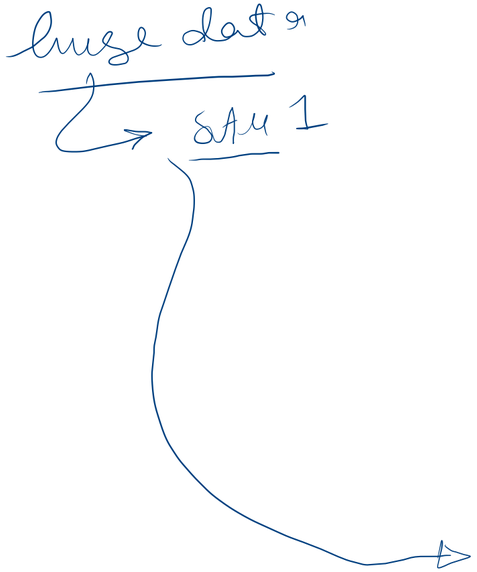


توانمندی های دیگر SAM1

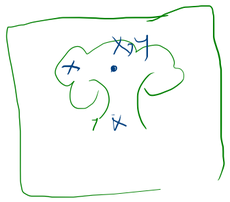


Zero-Shot Segmentation ⟶ **بدون** fine–tune **روی دیتای جدید هم کار می‌کند.**

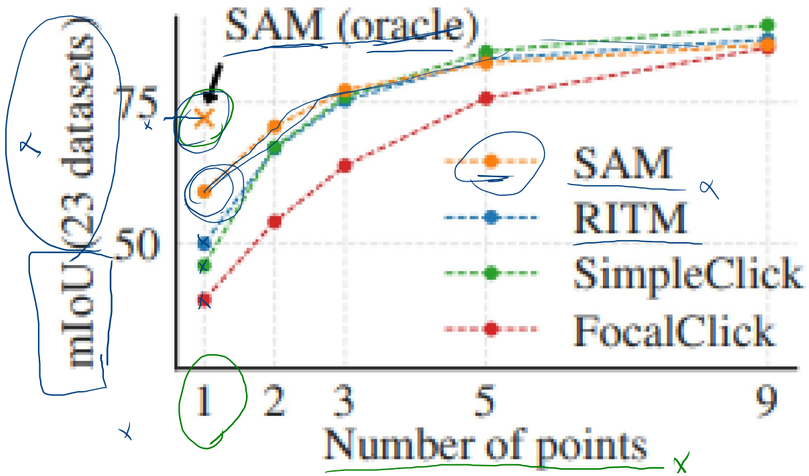
توانمندی های دیگر SAM1



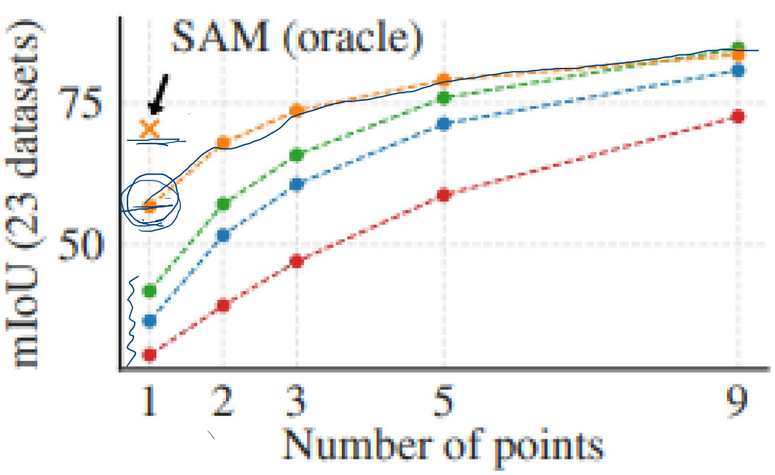
General-purpose ---> می‌تواند در هر حوزه‌ای (پزشکی، صنعتی، شهری) استفاده شود.



مزیت های مدل SAM

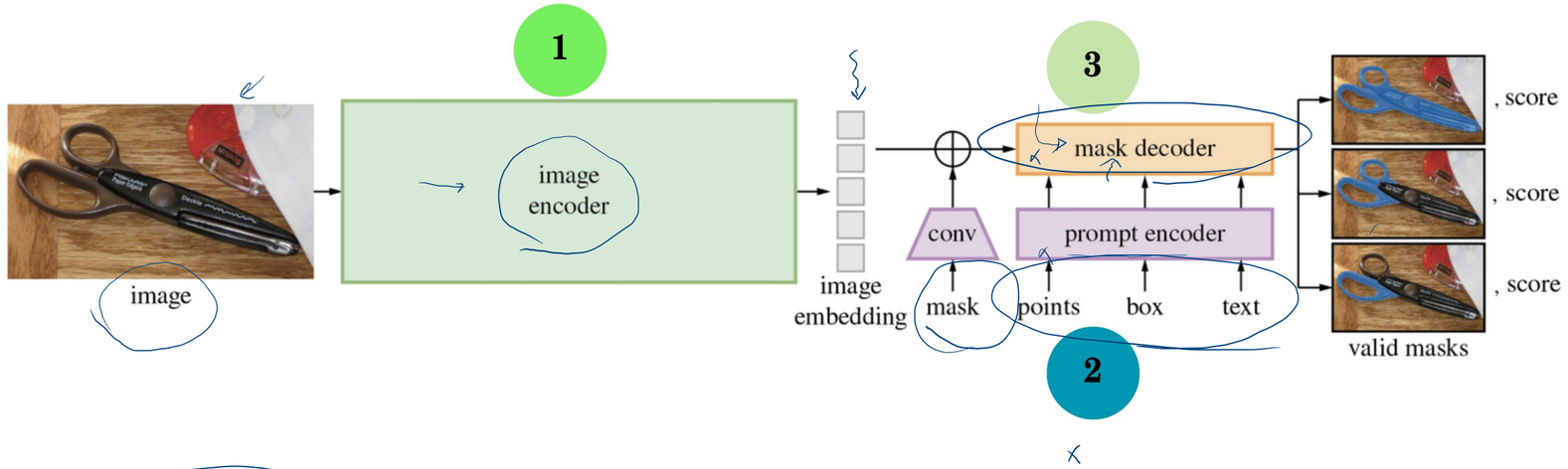


(c) Center points (default)



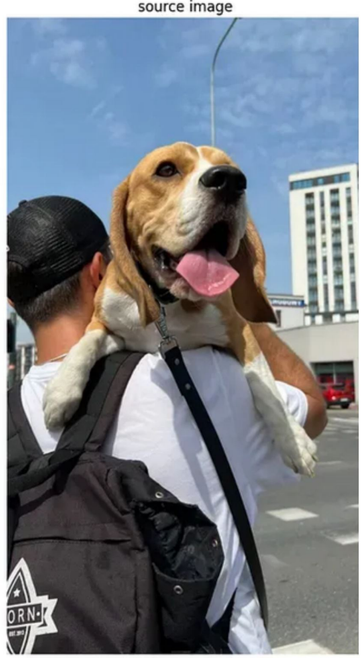
(d) Random points

ساختار مدل SAM



ساختار مدل SAM

source image



1

Image Encoder

استخراج ویژگی‌های تصویر

2

Prompt Encoder

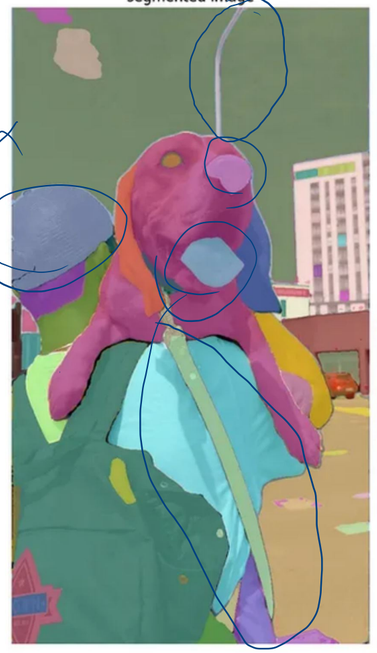
استخراج ویژگی‌های پرامپت ورودی کاربر

3

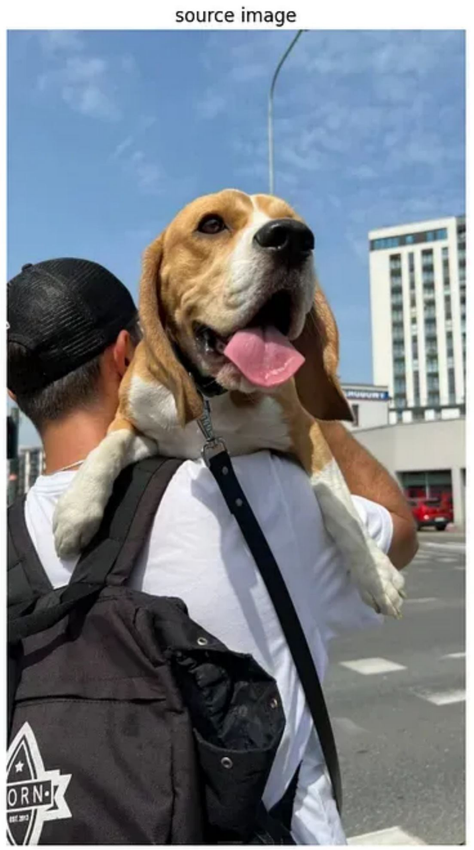
Mask Decoder

تولید ماسک نهایی

segmented image

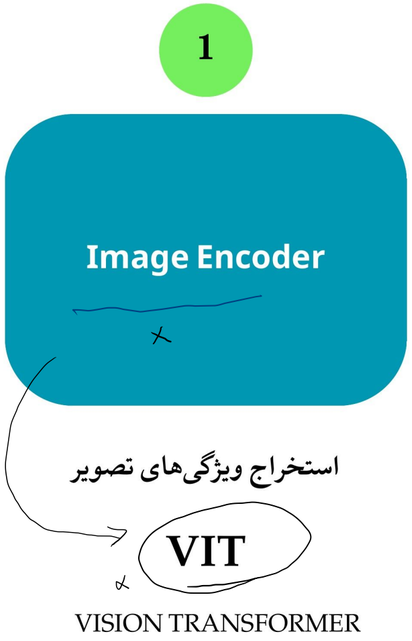


ساختار مدل SAM

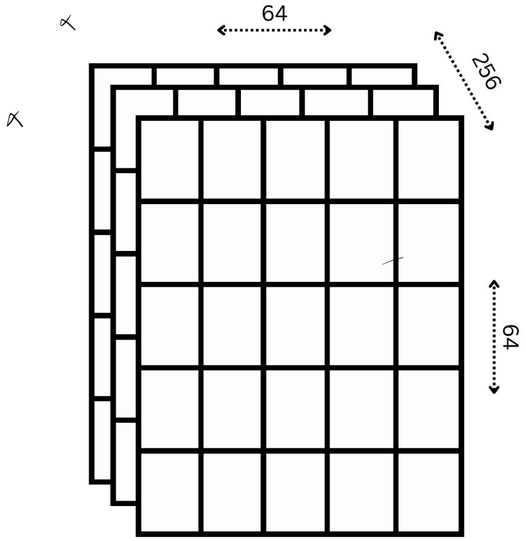


تصویر RGB (مثلاً 512×512)

ورودی



مدل



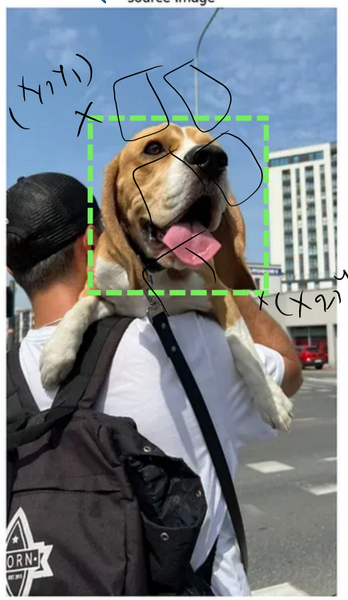
EMBEDDING تصویری (مثل 256×64×64)

خروجی

Prompt Encoder

چگونه ورودی Bounding Box انکد می شود؟

source image



Bounding Box

corners of the box

(x1, y1, x2, y2)

box

1 Normalization([0,1])

$$x1_{norm} = \frac{x1}{W}, \quad y1_{norm} = \frac{y1}{H}$$

$$x2_{norm} = \frac{x2}{W}, \quad y2_{norm} = \frac{y2}{H}$$

*Resolution Invariant

x1-norm
y1-norm
x2-norm
y2-norm

2 Positional Encoding

$$PE_{x,y} = [\sin(\omega_1 x), \cos(\omega_1 x), \ldots, \sin(\omega_d y), \cos(\omega_d y)]$$

*2D Sine-Cosine Positional Encoding
ω = fixed frequency bands.
It lets the model "understand" the location

box
point
prompt
mask

256 (Fixed)

3 Type Embedding

$$E^{box}_{\text{type}}$$

- learned Type Embedding
- it lets the model know "this is a box", not a point or mask

4 corners

256

output

4 Final Sum

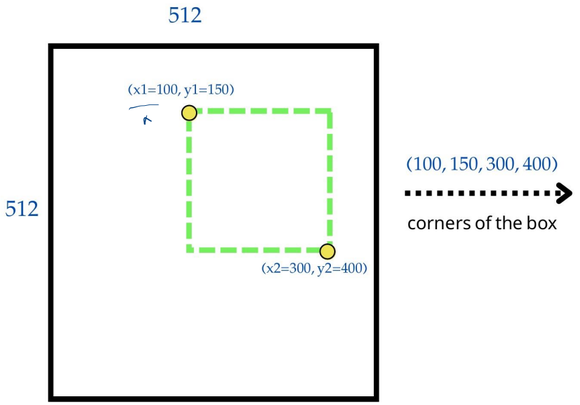
$$E_{corner} = PE_{x,y} + E^{box}_{\text{type}}$$

- For a Box, each corner gets its own embedding
- Each → Positional Encoding + Type Embedding → 256-D vector
- Four 256-dimensional vectors (one for each corner)
- These are passed to the Mask Decoder.

تبدیل ورودی کاربر (prompt) به embedding عددی که بتواند با Image Embedding ترکیب شود.

Prompt Encoder

چگونه ورودی Bounding Box انکد می شود؟ (مثال عددی)



(100, 150, 300, 400)

corners of the box

1

Normalization([0,1])

$$x1_{norm} = 100/512 \approx 0.195$$

$$y1_{norm} = 150/512 \approx 0.293$$

$$x2_{norm} = 300/512 \approx 0.586$$

$$y2_{norm} = 400/512 \approx 0.781$$

0.195
0.293
0.586
0.781

2

Positional Encoding

Sample: (0.195, 0.293):

اعمال سینوس و کوسینوس به صورت متناوب روی مختصات ورودی

۰/۴۷
۰/۸۸
...
۰/۲۹

256 (Fixed)

3

Type Embedding

$$E^{box}_{\text{type}}$$

A learned vector for "box"

E_type_box = [0.01, 0.23, ... , -0.14]

4

Final Sum

$$E_{corner} = PE_{x,y} + E^{box}_{\text{type}}$$

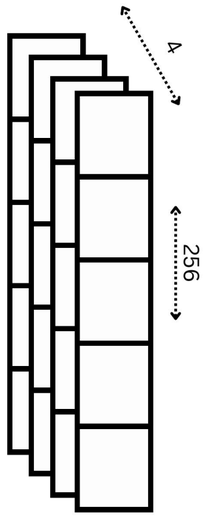
E_corner1 = [0.48, 1.11, 0.57, ... , 0.15]

E_corner2 = ...

E_corner3 = ...

E_corner4 = ...

output



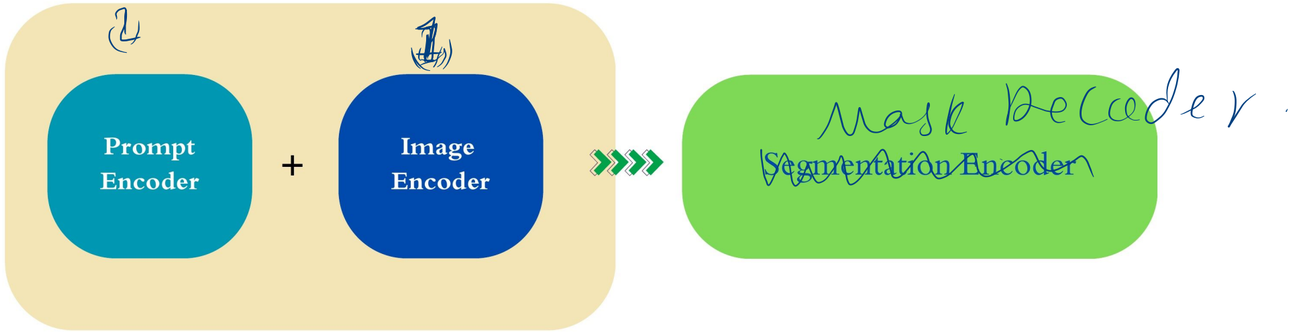
Mask Decoder

3

Mask Decoder

قلب مدل SAM است که همه‌ی اطلاعات رو ترکیب می‌کند.

با توجه به دستور کاربر، مرز دقیق شیء رو جدا می کند.



Mask Encoder

3

Mask Decoder

Mask Decoder **وظایف**

- **درک ارتباط بین تصویر و Prompt**
- **مشخص کردن ناحیه دقیق شیء مورد نظر**
- **تولید یک ماسک پیکسل به پیکسل**

Mask Encoder

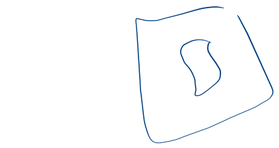


Image Embedding (ViT)

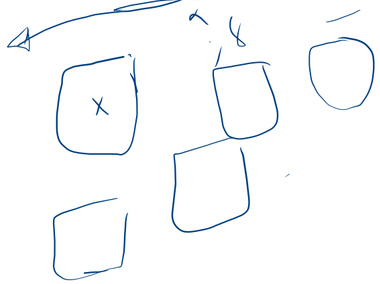
Prompt Embedding

Mask Decoder

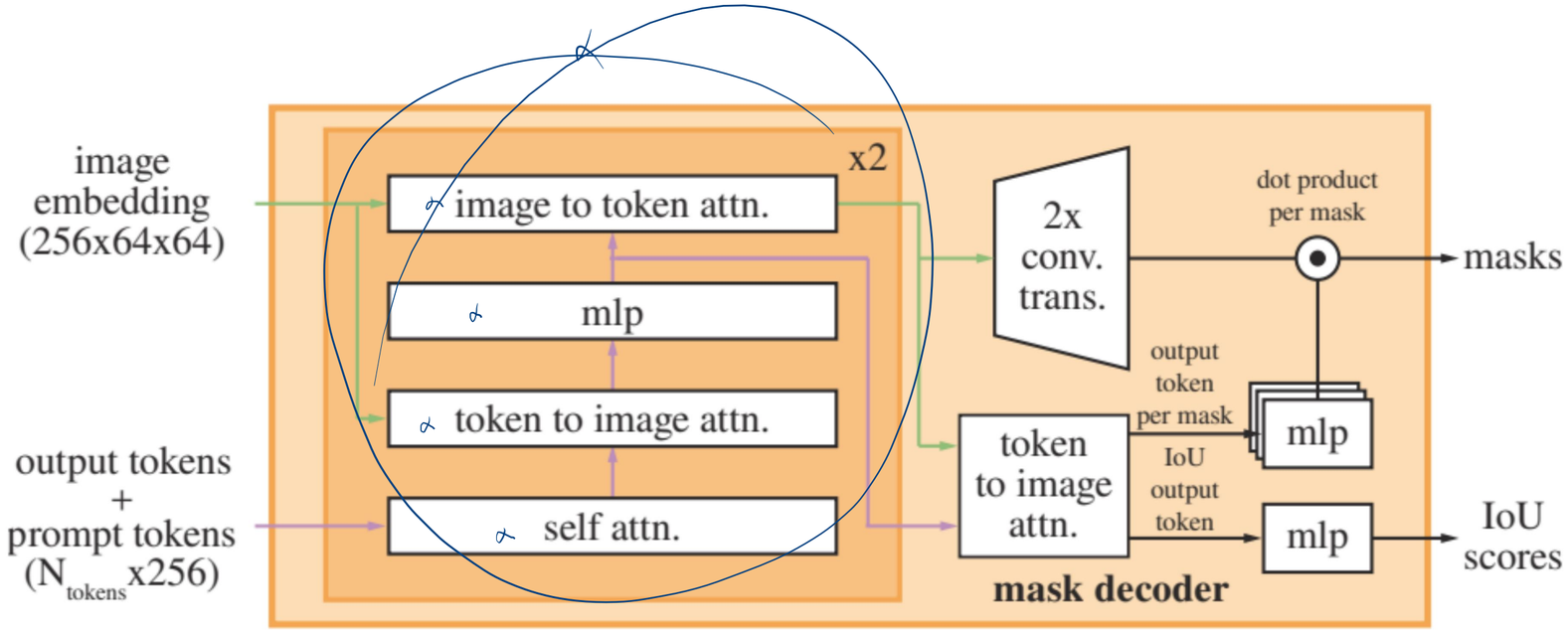
ماسک دقیق 512×512

احتمال (0~1) برای هر پیکسل

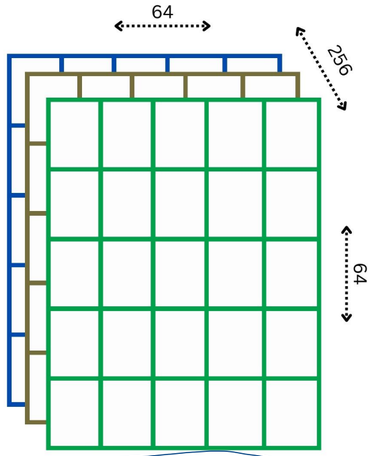
چند ماسک (با احتمالات متفاوت)



ساختار درونی Mask Encoder



ساختار درونی Mask Encoder



خروجی image encoder

ویژگی‌های کل تصویر، با موقعیت مکانی (spatial) حفظ شده

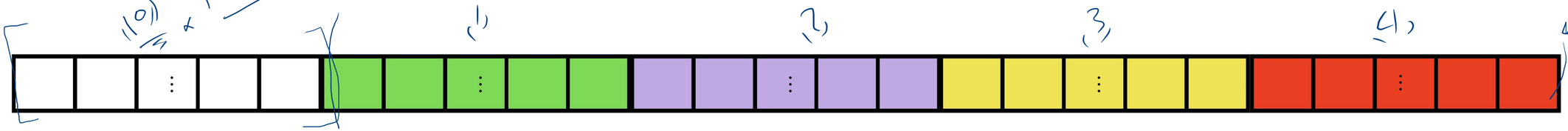
image embedding (256x64x64)

output tokens + prompt tokens (N_{tokens} x256)

Mask Encoder

inputs

خروجی prompt encoder



256 outpout tokens

256 prompt token 1

256 prompt token 2

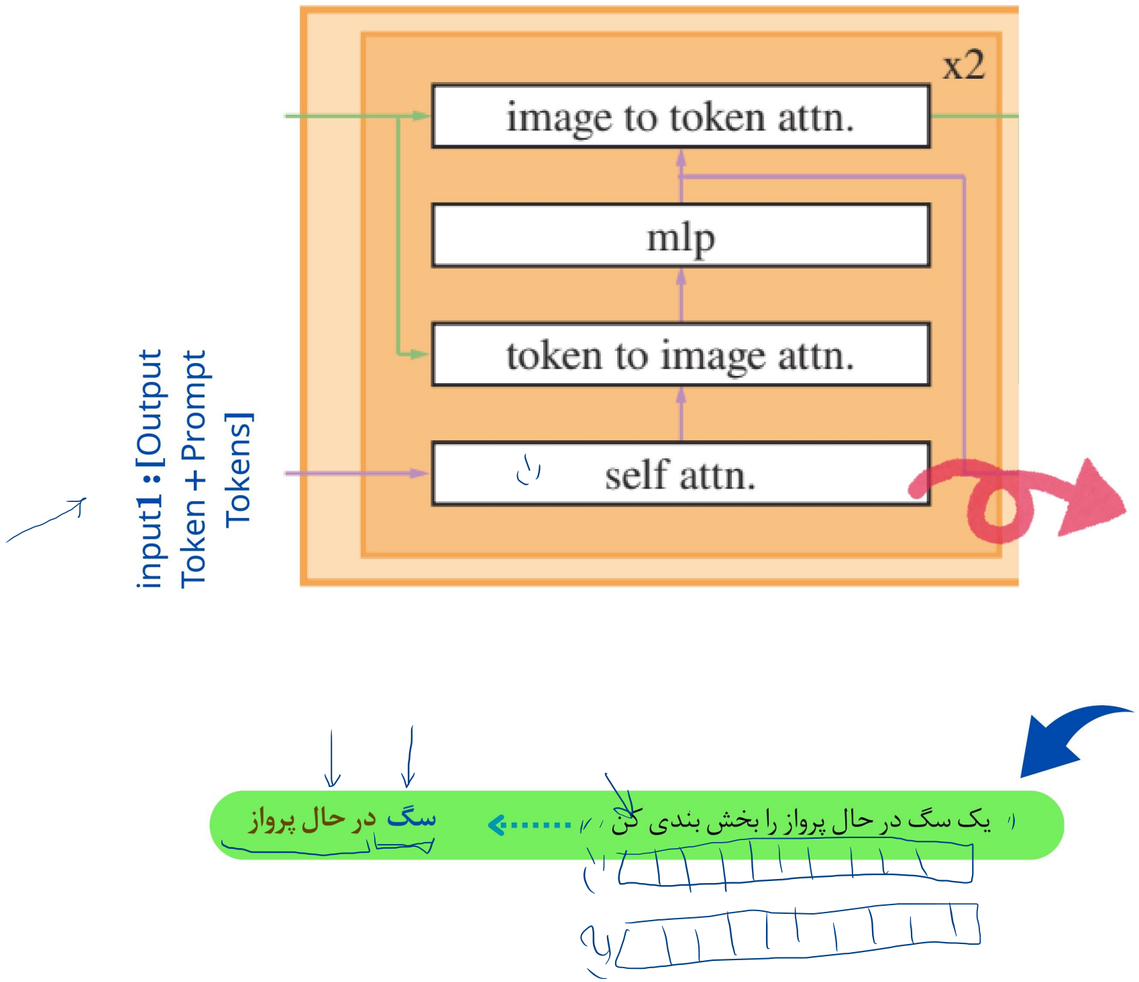
256 prompt token 3

256 prompt token 4

یک بردار ثابت برای همه ی نمونه هاست که در حین آموزش مقدار ثابت آن تعیین می شود.

ساختار درونی Mask Encoder

Self Attention



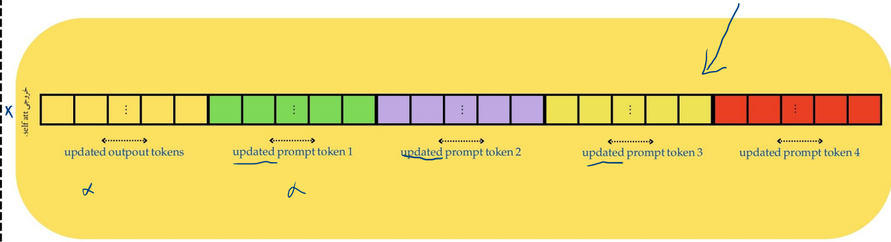
درباره self attention

- به اشتراک گذاری اطلاعات بین Prompt tokens و با Output Token و مشخص کردن نقاط مهم پرامپت ورودی و توجه بیشتر به آن

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

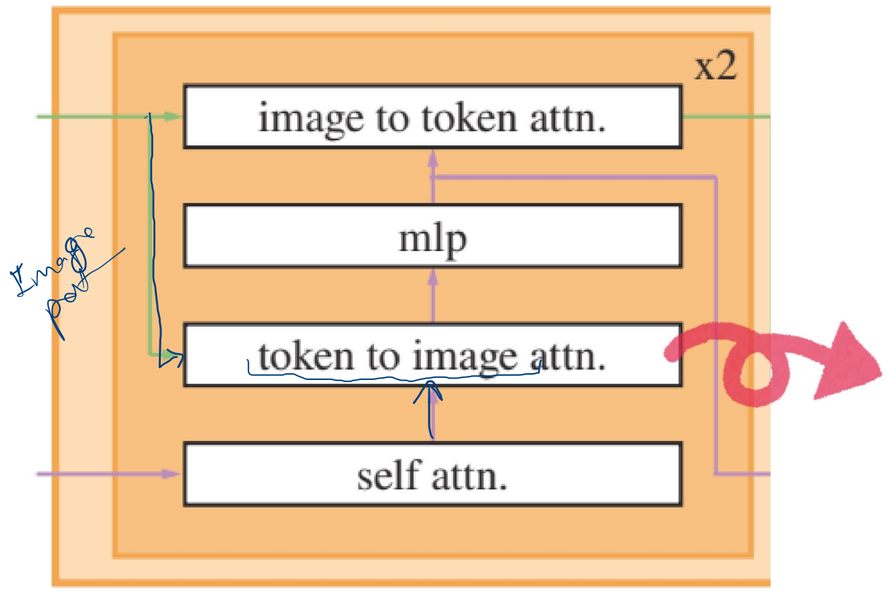
- Query (Q) = [Output Token + Prompt Tokens]
- Key (K) = [Output Token + Prompt Tokens]
- Value (V) = [Output Token + Prompt Tokens]

خروجی بخش self attention



ساختار درونی Mask Encoder

Token to Image Att.

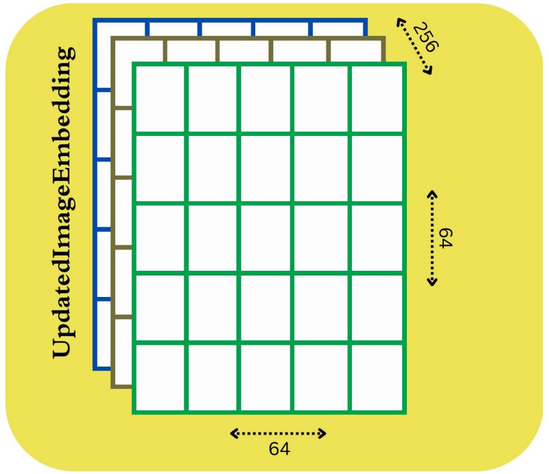


- «Image patches از Prompt (خروجی self attention) یاد میگیرد که چه چیزها و نکاتی مهمند. به نوعی در این بخش پرامپت نقش محدود کننده و شرط (Condition) را برای تصویر بازی میکند.

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

- Query (Q) = Image Embedding
- Key (K) = Prompt+Output Tokens
- Value (V) = Prompt+Output Tokens

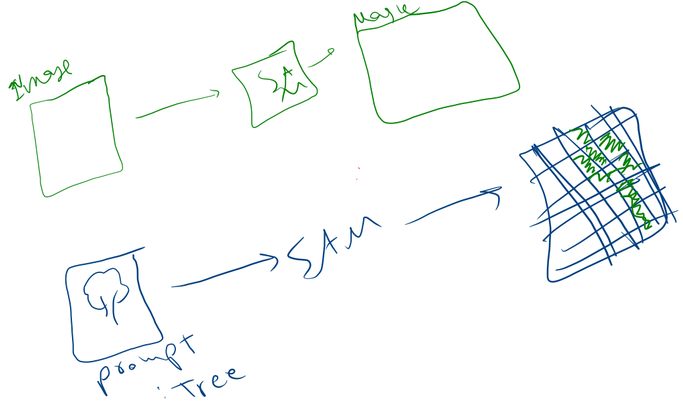
درباره token to image att.

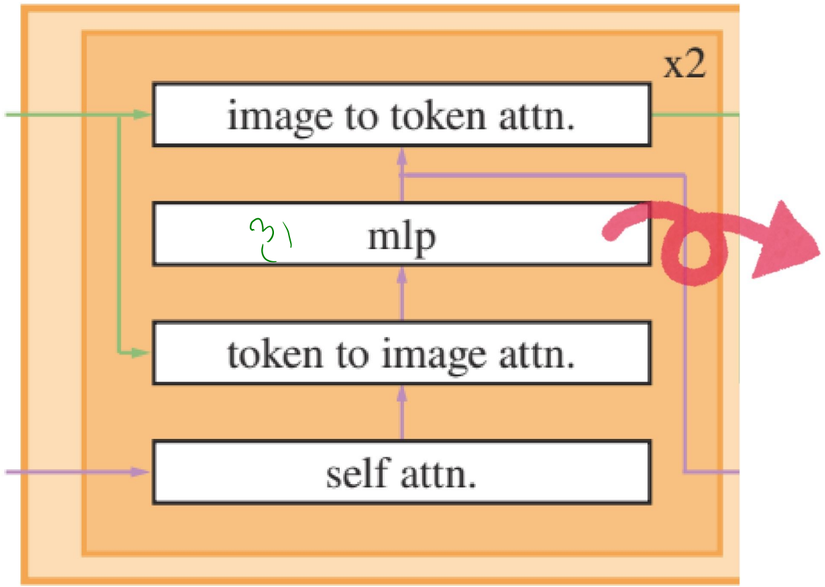


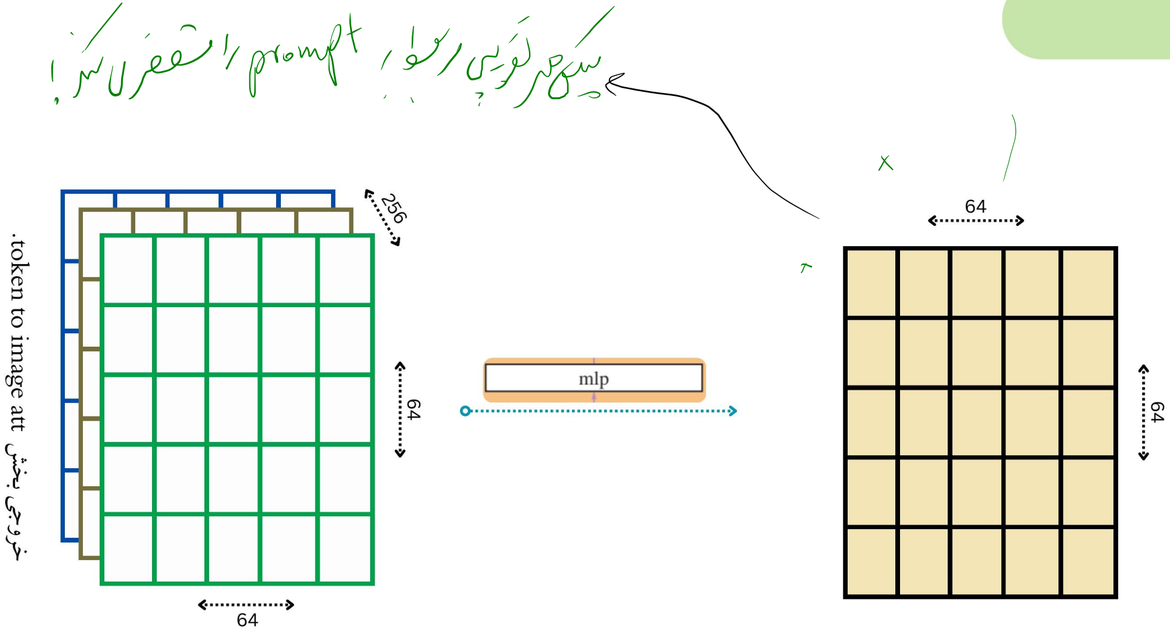
خروجی بخش token to image att.

ساختار درونی Mask Encoder

mlp







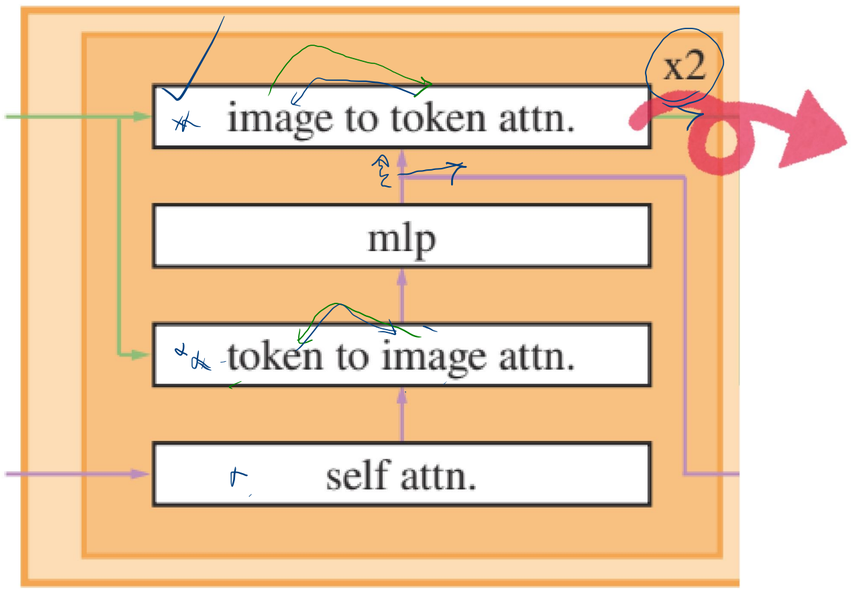
تبدیل کردن تمام فیچرمپ های به تصویر دو بعدی

Feedforward network / Multi-Layer Perceptron

این بخش مشخص می کند که این Prompt مربوط به کدام قسمت تصویر است.

ساختار درونی Mask Encoder

Image to token Att.

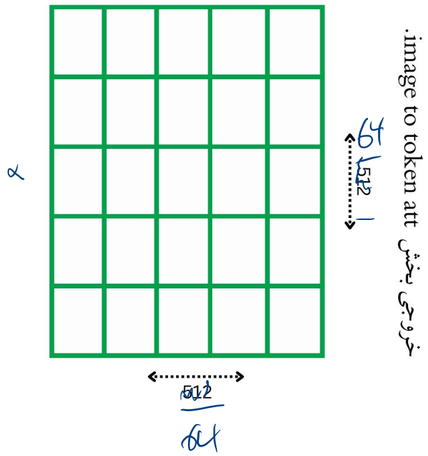


- در این بخش مدل با ا ستفاده از Cross-Attention یاد می‌گیرد که کدام بخش تصویر به Prompt ورودی مربوط است.

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

- Query (Q) = output of mlp (updated prompt token) - 256 D Vector
- Key (K) = Image Token (ViT patches)
- Value (V) = Image Token (ViT patches)

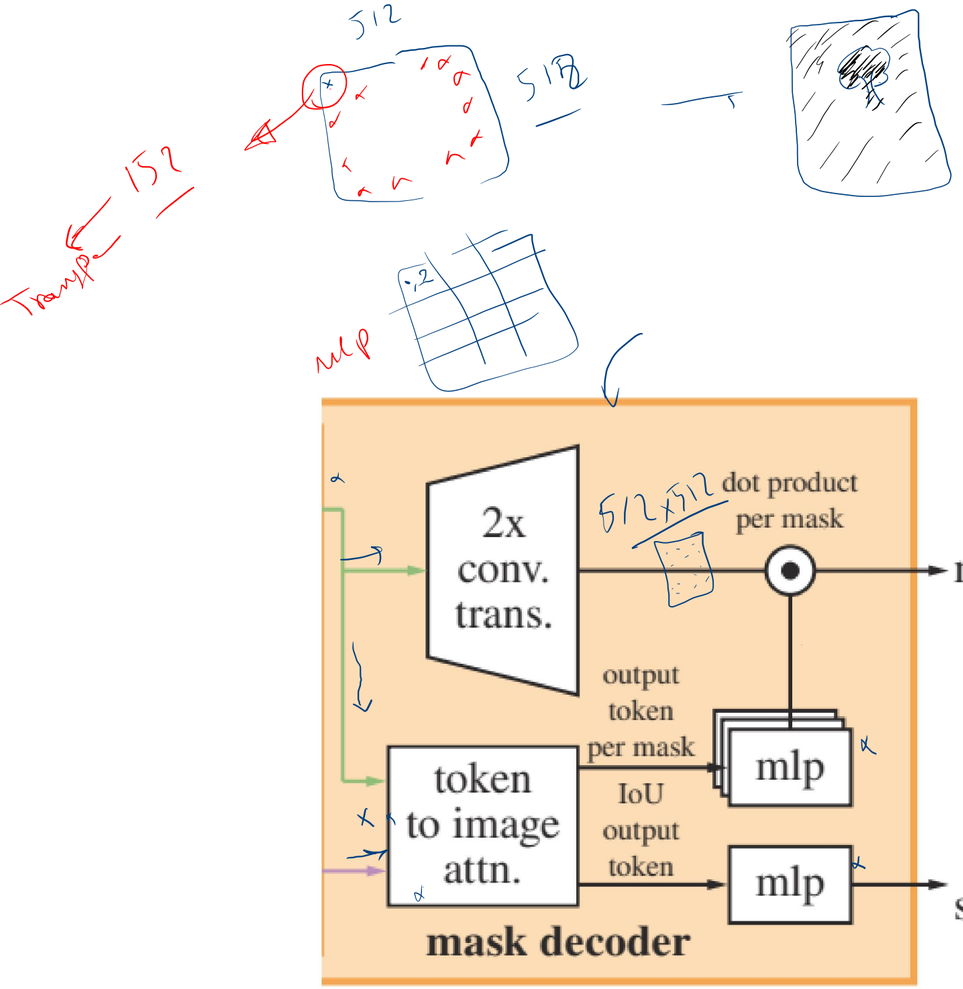
درباره image to token att

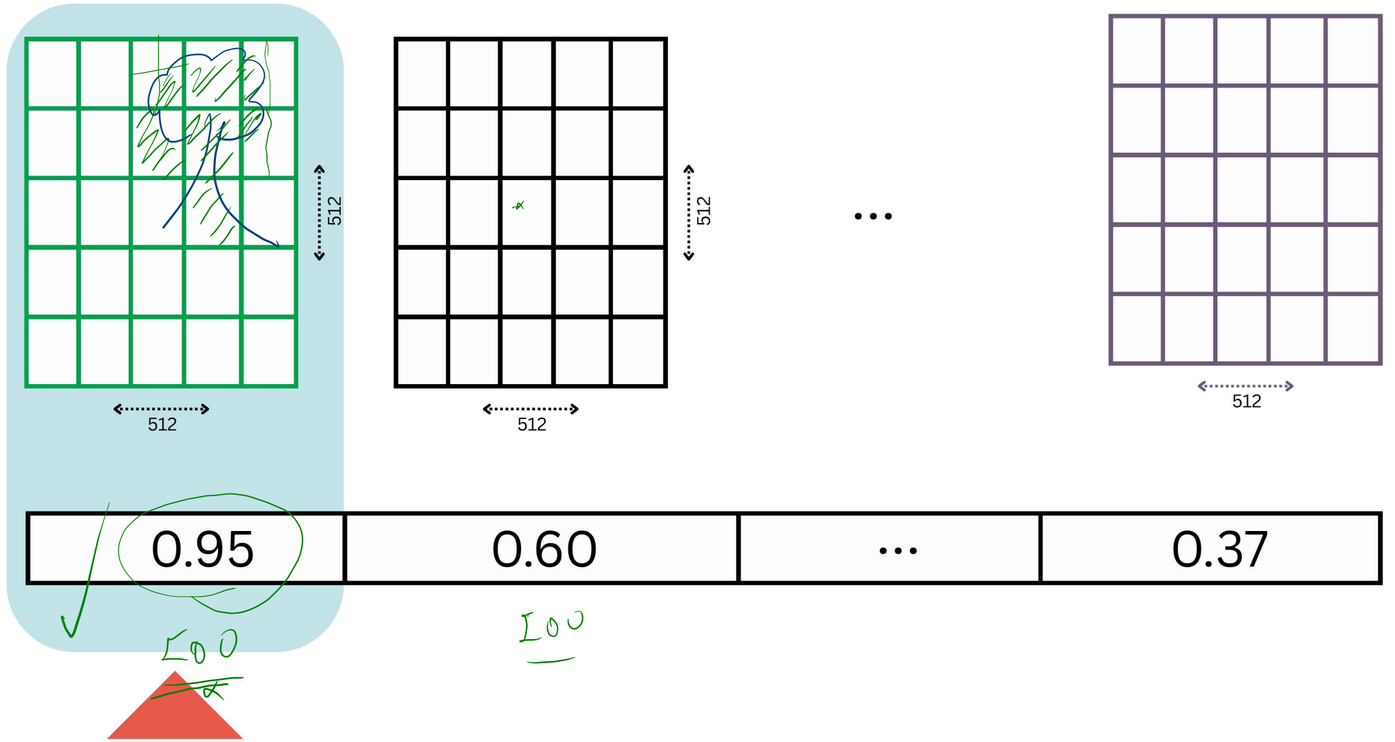


خروجی بخش image to token att

Mask Encoder ساختار درونی

Image to token Att.





Selected

The end

نوازش کی