

# Как работи генирането на изображения с изкуствен интелект

Йордан Маджунков

Генерирай ми радостни хора практикуващи  
акро йога на плажа

# Генерирай ми радостни хора практикуващи акро йога на плажа



# Генерирай ми радостни хора практикуващи акро йога на плажа





# “Разбиране” на съдържанието на изображение



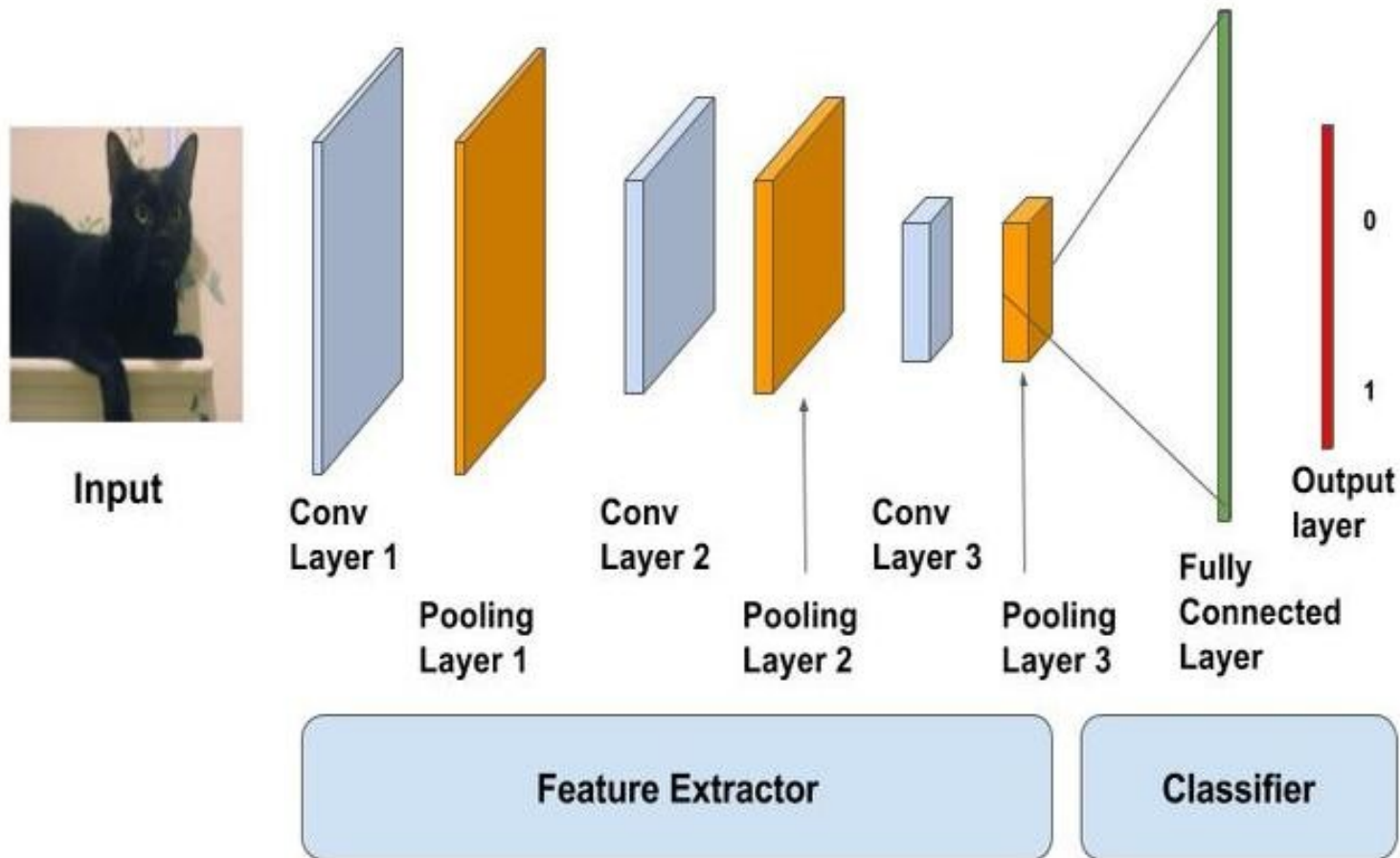
- Жълта котка в голф 2
- Изплашен мъжки котарак
- Нахално котенце се вмъкна в колата ми

# Защо е трудно за “разбиране”

## Жълта котка в кола



# Двуичен класификатор - Има ли котка ?

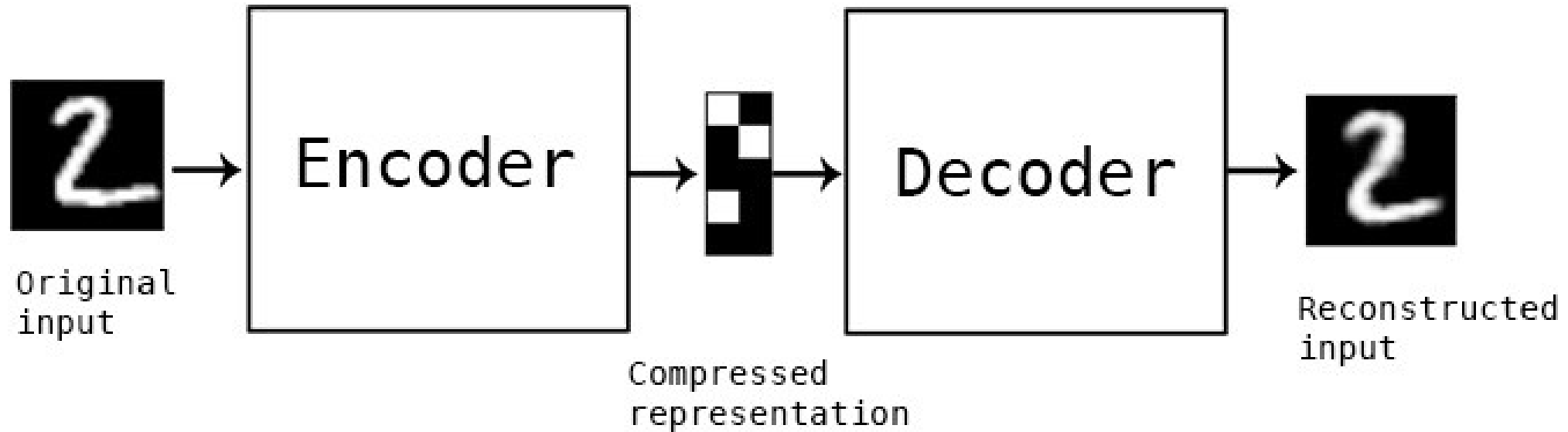


# Двуичен класификатор - Проблеми

- Трудно скалира
- Специфично за всеки клас
- Квадратично нарастване на грешката с броя класове
- Изисква ръчно анотиране на данните



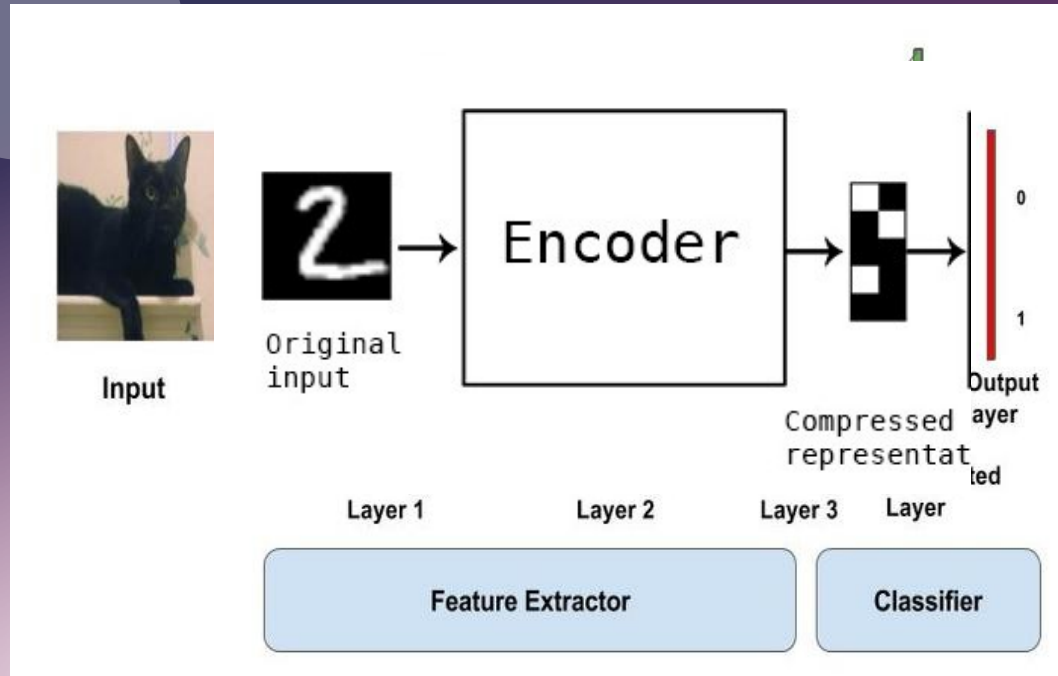
# Encoder - Decoder



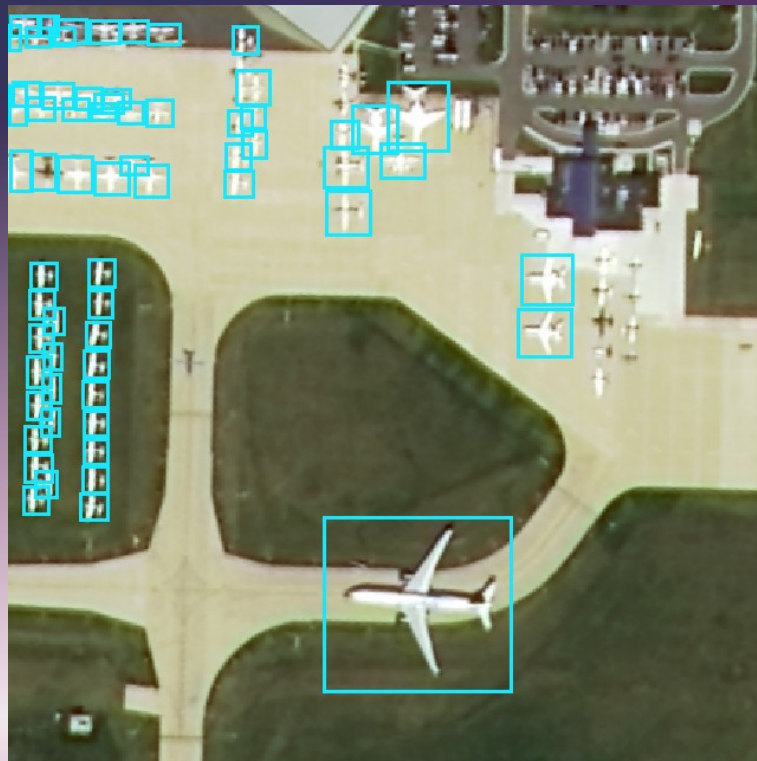
# Encoder - Decoder

- Unsupervised learning
- 400+ милион изображения
- Може да се ползват независимо

# Encoder + Има ли котка ?

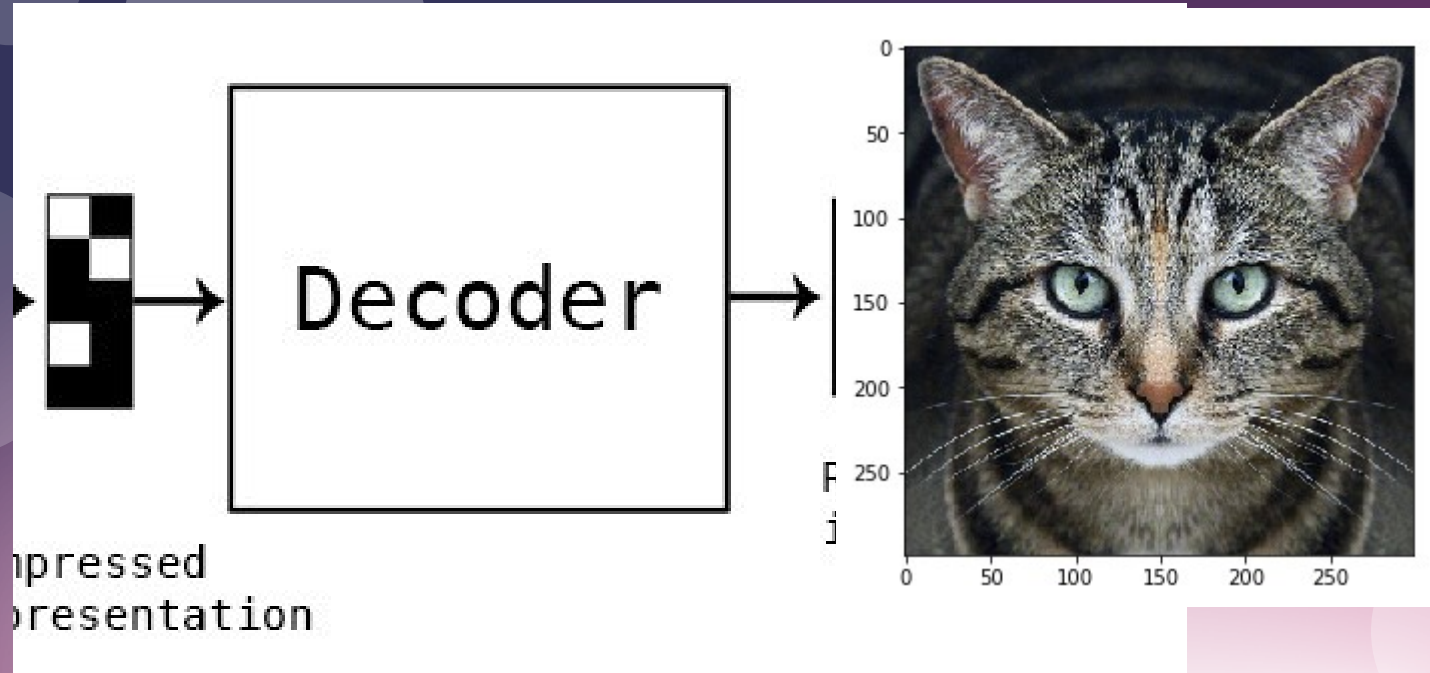


Къде има самолет на снимката ?





# Sampling + Decoder + Има ли котка ?

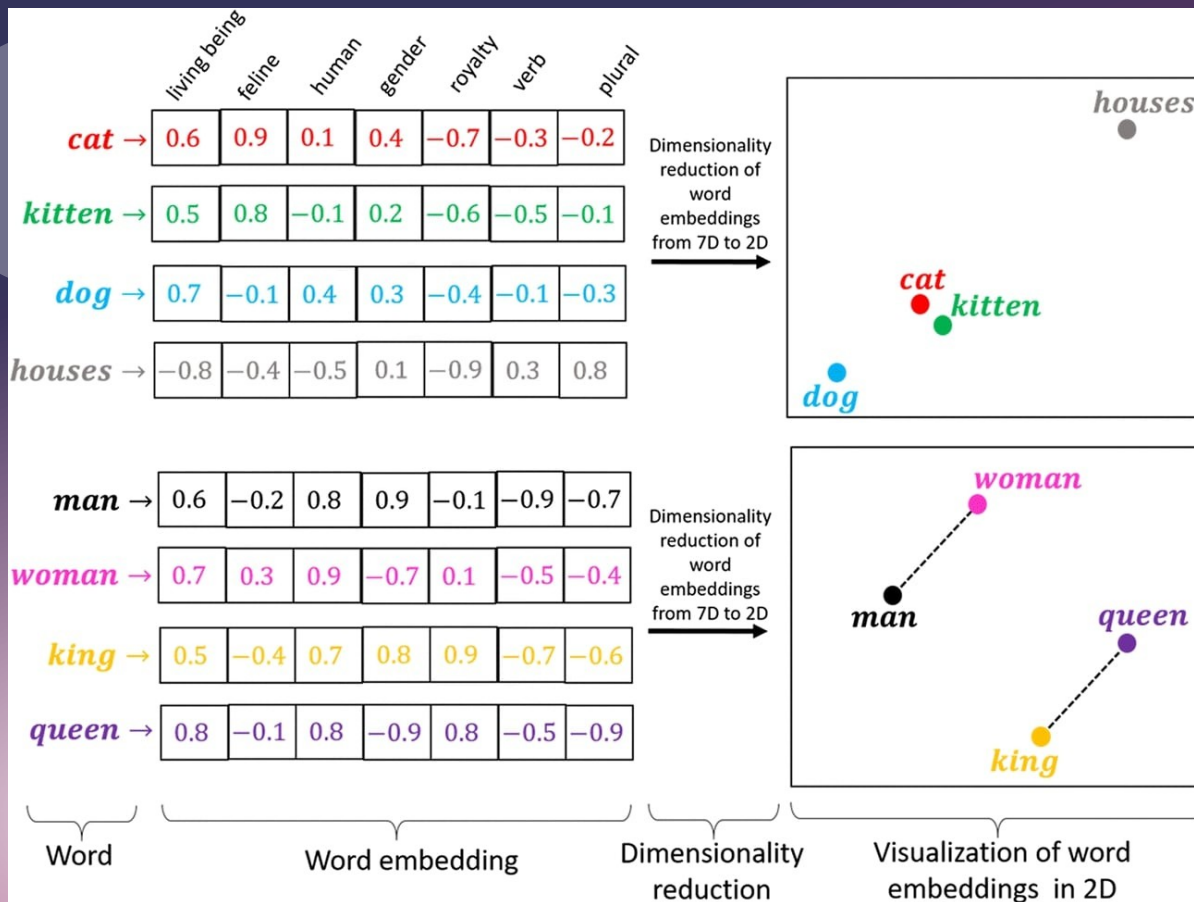


# Неостатъци

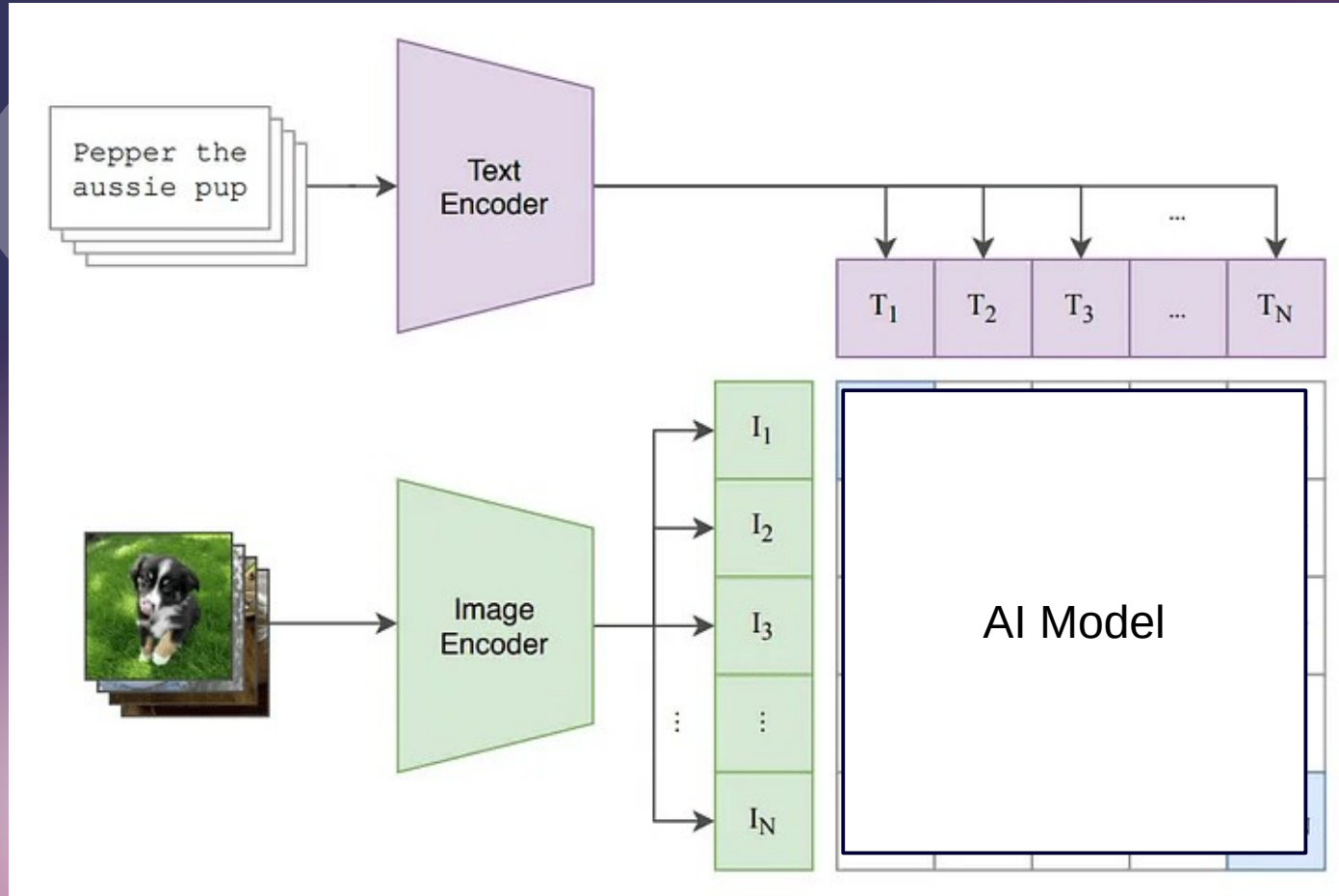
- Скъпо/Бавно/Неефективно
- Ограничен брой класове
- Шумни резултати с дефекти



# LLM Embeddings

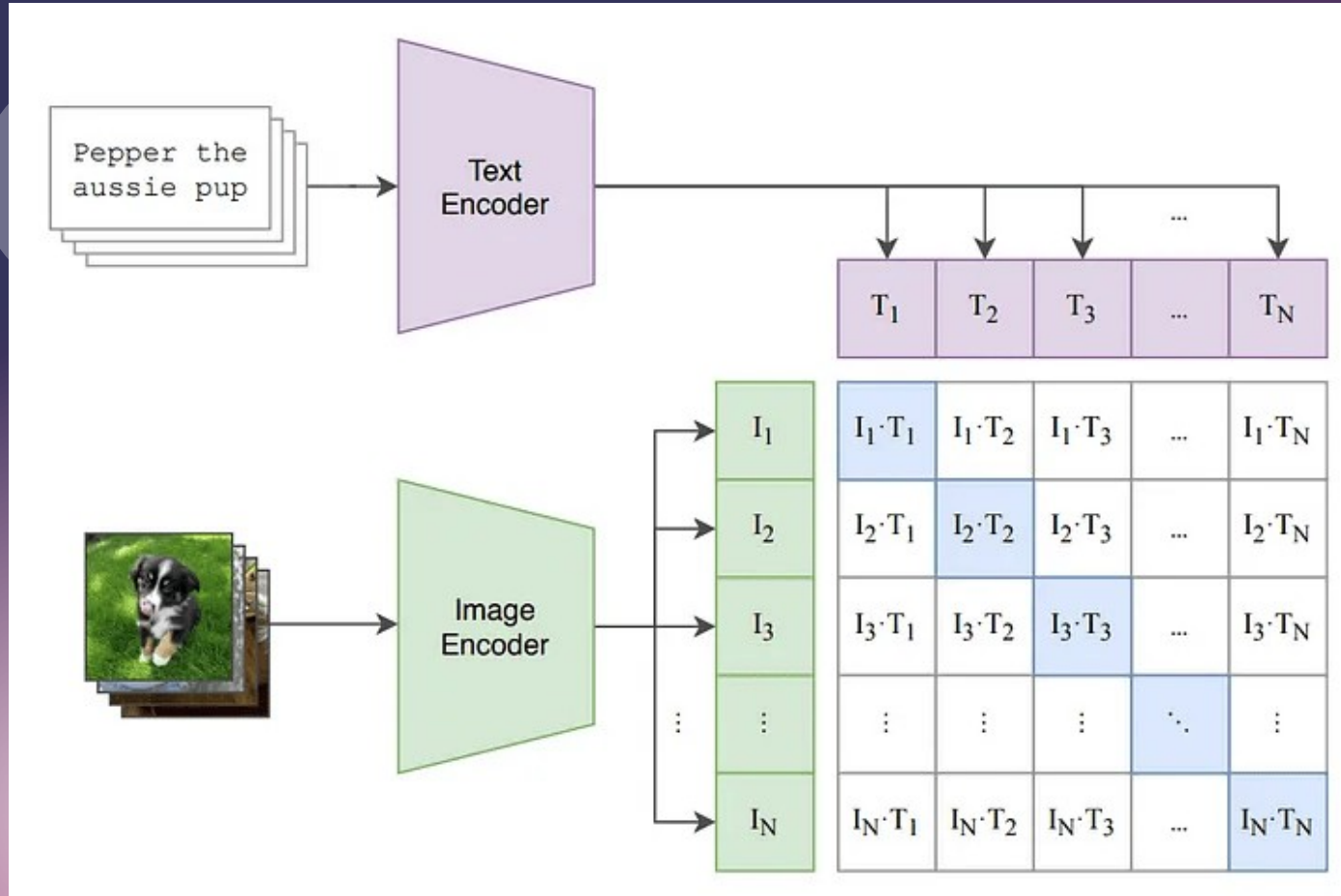


# Zero shot classification



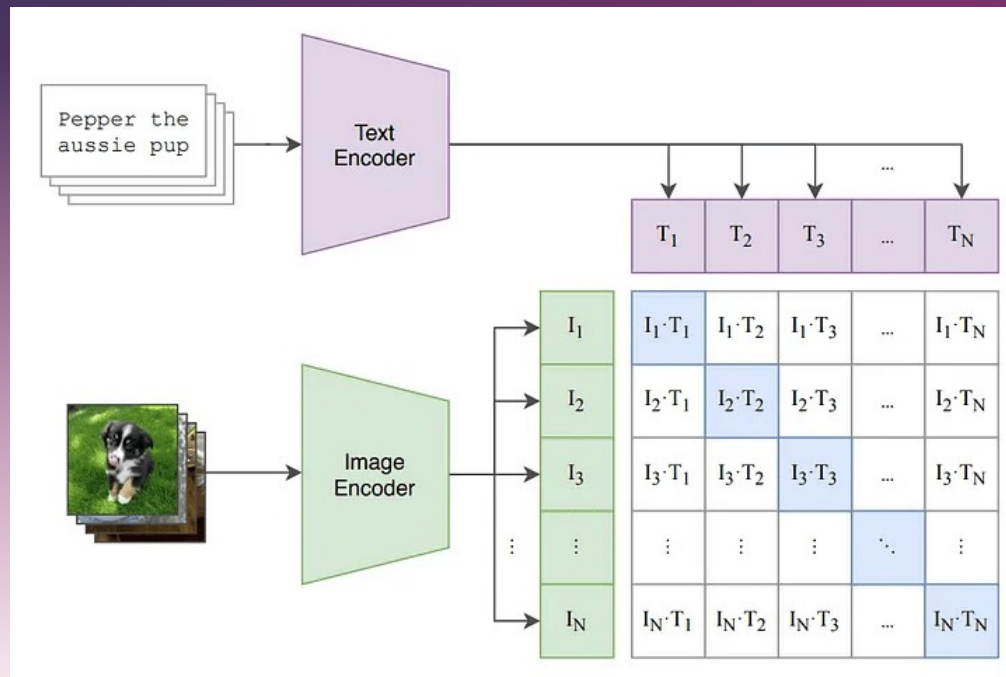


# CLIP



# CLIP – За генериране

- Скъпо/Бавно/Неефективно
- Ограничен брой класове
- Шумни резултати с дефекти
- Еднакви резултати

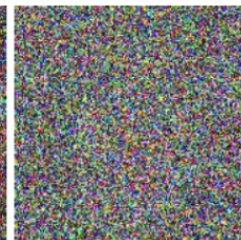
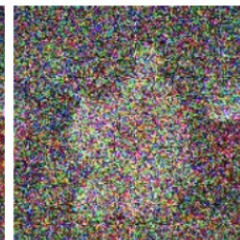
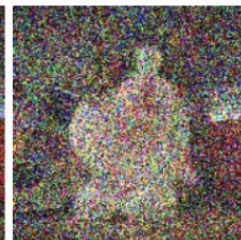
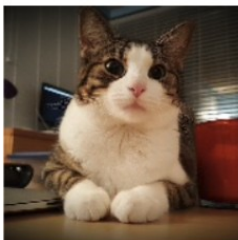


# Denoiser



# Denoiser

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$



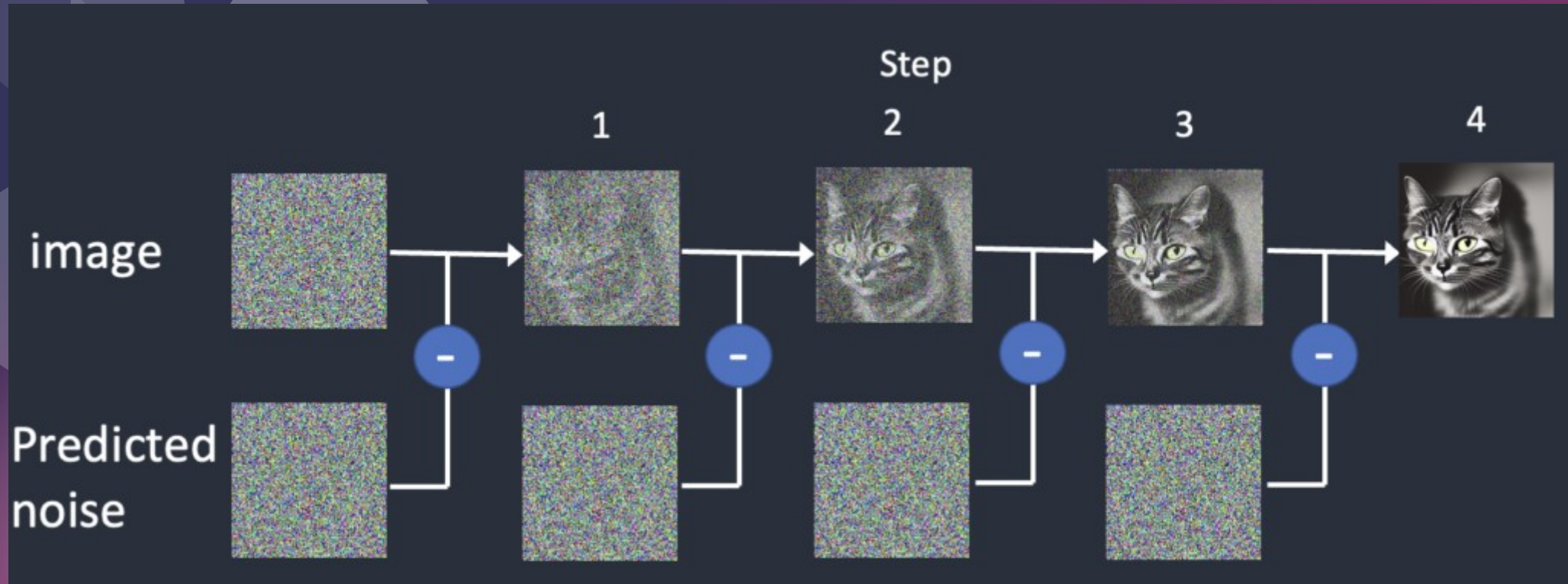
Data

Noise

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$$



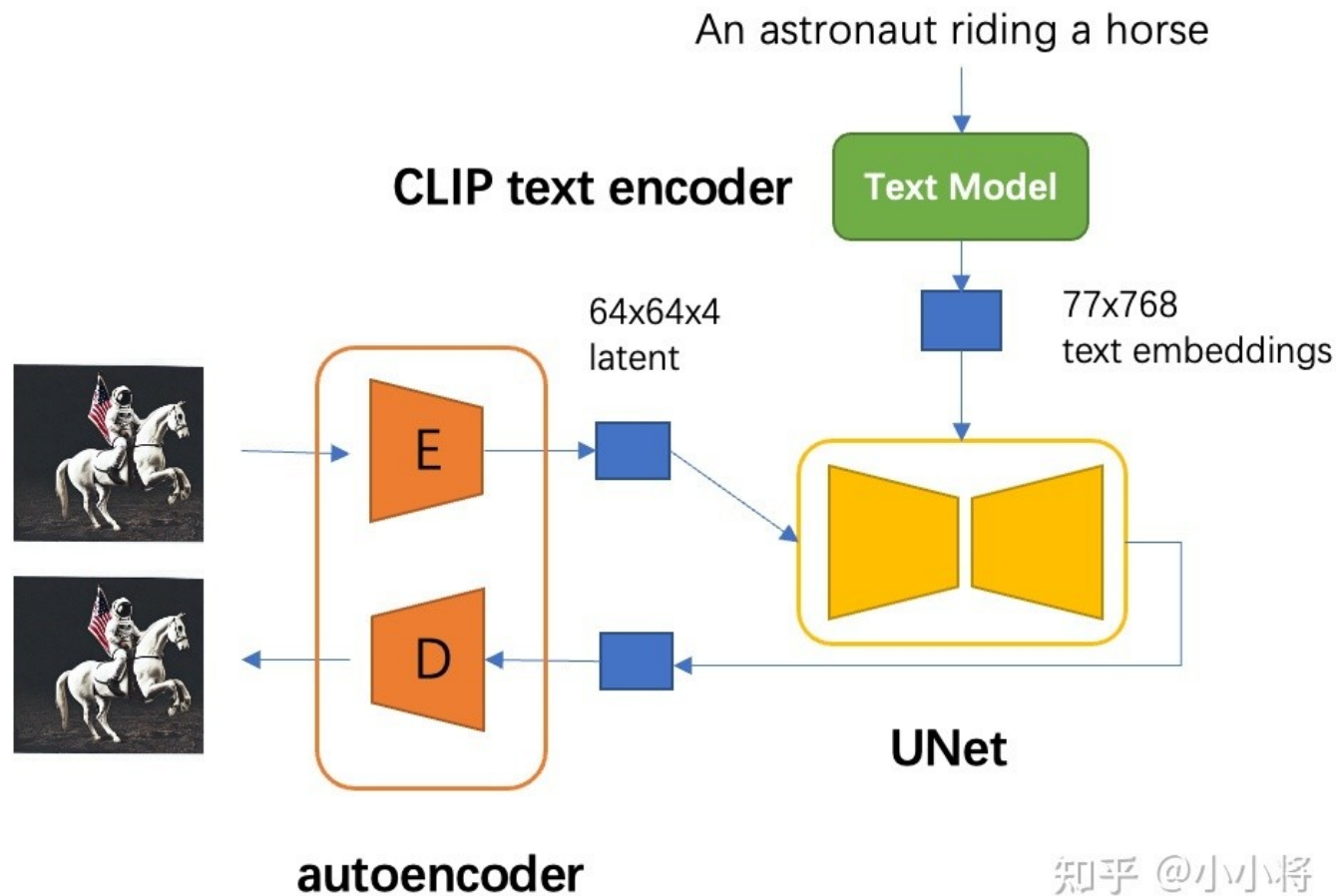
# Denoiser - unguided



# Denoiser - unguided



# Stable Diffusion = CLIP + Denoise



# Въпроси