# Multimodal NLP

MIPT
28.04.2022
Anton Emelianov



https://github.com/king-menin/mipt-nlp2022

# Multimodal NLP tasks

Sources:

- Speech
- Music
- Image
- Video
- …

# Text recognition

# Text recognition

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo, license plates in cars...) or from subtitle text superimposed on an image (for example: from a television broadcast)

# Text recognition

## Tesseract OCR

Tesseract was originally developed at Hewlett-Packard Laboratories Bristol and at Hewlett-Packard Co, Greeley Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++izing in 1998. In 2005 Tesseract was open sourced by HP. From 2006 until November 2018 it was developed by Google.

```
tesseract --tessdata-dir /usr/share imagename outputbase -l eng --psm 3
```

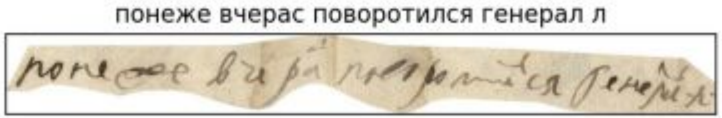
Following examples use this image which has text in multiple languages.

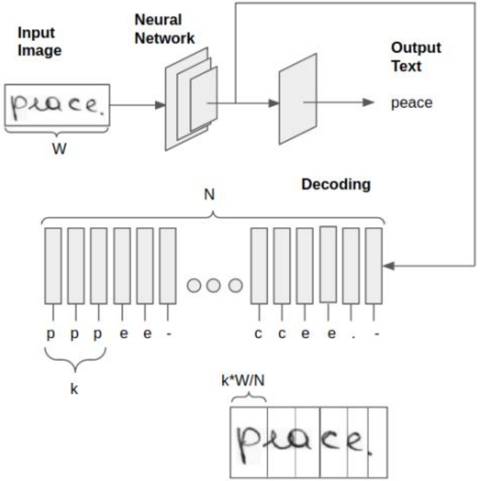
The (quick) [brown] {fox} jumps!
Over the $43,456.78 <lazy> #90 dog
& duck/goose, as 12.5% of E-mail
from aspammer@website.com is spam.
Der „schnelle" braune Fuchs springt
über den faulen Hund. Le renard brun
«rapide» saute par-dessus le chien
paresseux. La volpe marrone rapida
salta sopra il cane pigro. El zorro
marrón rápido salta sobre el perro
perezoso. A raposa marrom rápida
salta sobre o cão preguiçoso.

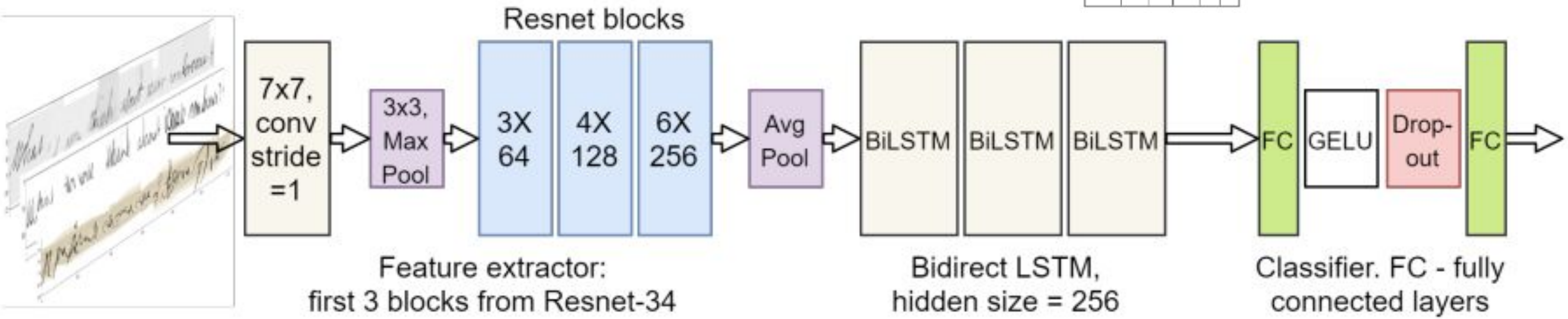
# Text recognition

[StackMix and Blot Augmentations for Handwritten Text Recognition 2021](#)

[Git](#)



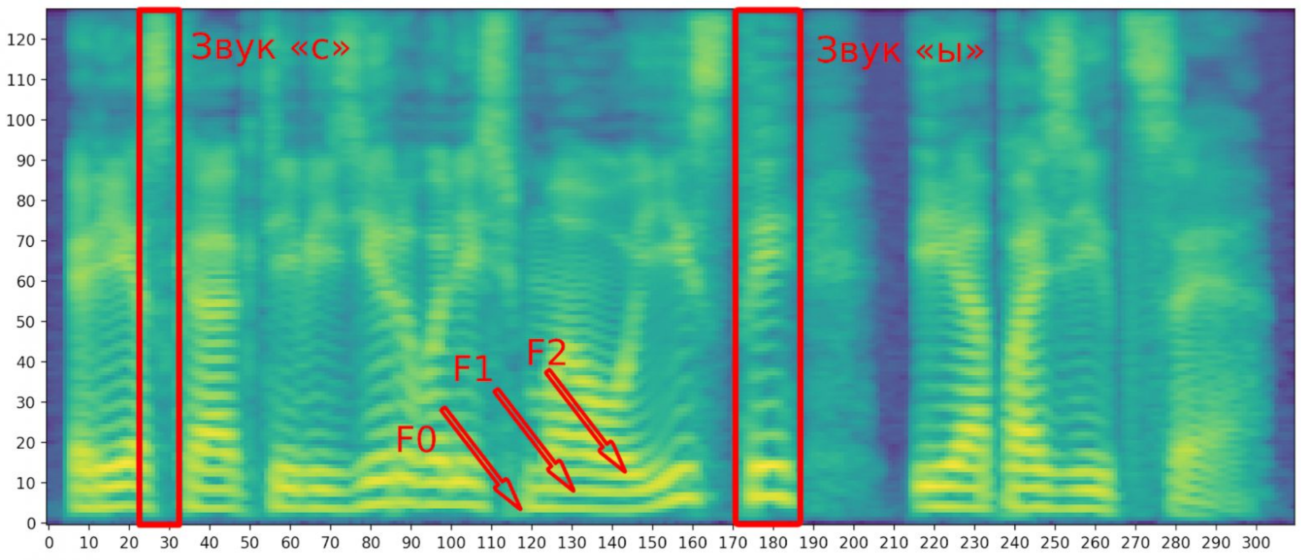



# Text to speech

# Text to speech

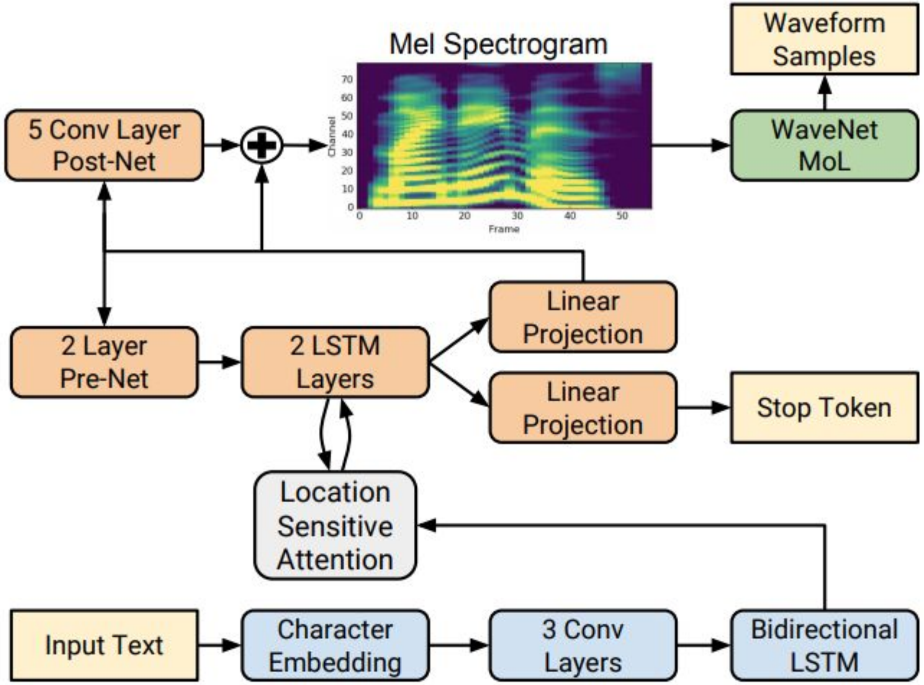
mel spectrogram



Tacotron 2

The Tacotron 2 model form a text-to-speech system that enables user to synthesise a natural sounding speech from raw transcripts without any additional prosody information. The Tacotron 2 model produces mel spectrograms from input text using encoder-decoder architecture.

WaveNet, a deep generative model of raw audio waveforms



# Image captioning

# Image captioning

- To build a model that can generate a descriptive caption for an image we provide it.

# Image captioning

Datasets:

- COCO (Microsoft Common Objects in Context)
  - The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of 328K images.
- Flickr30k
  - The Flickr30k dataset contains 31,000 images collected from Flickr, together with 5 reference sentences provided by human annotators.
- And more…

# Image captioning

Metrics

- CIDEr: Consensus-based Image Description Evaluation
- SPICE: Semantic Propositional Image Caption Evaluation
- BLUE: Bilingual Evaluation Understudy Score
- METEOR: Metric for Evaluation of Translation with Explicit ORdering

# CIDEr

CIDEr score for n-grams of length n is computed using the average cosine similarity between the candidate sentence and the reference sentences, which accounts for both precision and recall

$$\text{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{\boldsymbol{g^n}(c_i) \cdot \boldsymbol{g^n}(s_{ij})}{\|\boldsymbol{g^n}(c_i)\| \|\boldsymbol{g^n}(s_{ij})\|}$$

use higher order (longer) n-grams to capture grammatical properties as well as richer semantics. We combine the scores from n-grams of varying lengths as follows:

$$\text{CIDEr}(c_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDEr}_n(c_i, S_i)$$

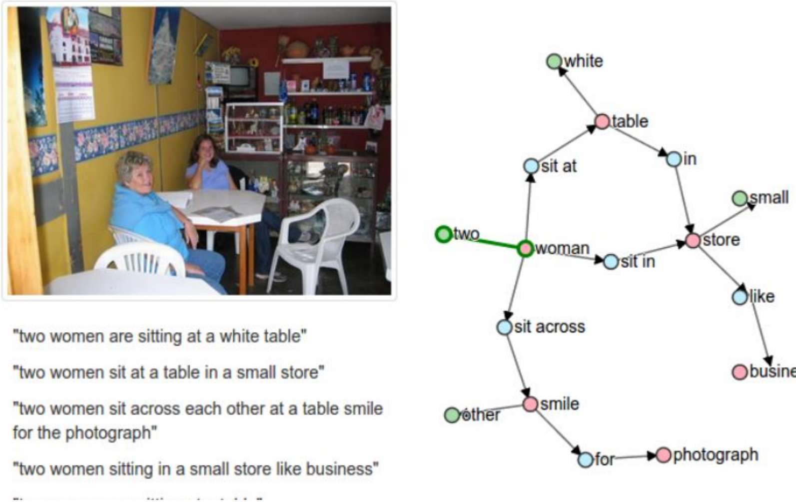
Empirically, found that uniform weights wn = 1/N work the best. Use N = 4.

# SPICE

## Semantic Parsing—Captions to Scene Graphs

$$T(G(c)) \triangleq O(c) \cup E(c) \cup K(c)$$

$$\{\ (\text{girl}),\ (\text{court}),\ (\text{girl, young}),\ (\text{girl, standing})\ (\text{court, tennis}),\ (\text{girl, on-top-of, court})\ \}$$

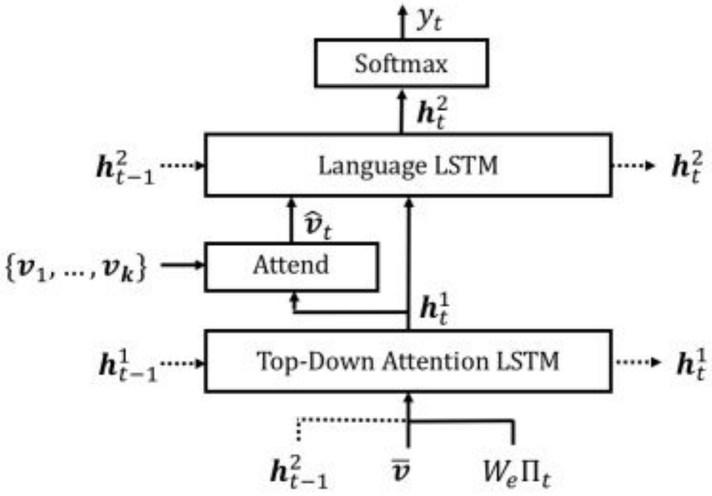


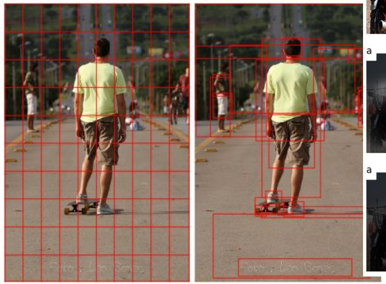

$$P(c, S) = \frac{|T(G(c)) \otimes T(G(S))|}{|T(G(c))|}$$

$$R(c, S) = \frac{|T(G(c)) \otimes T(G(S))|}{|T(G(S))|}$$

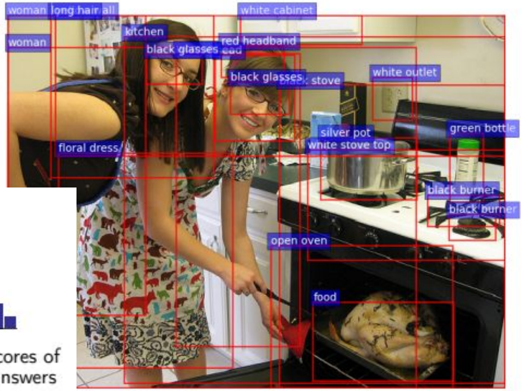
$$SPICE(c, S) = F_1(c, S) = \frac{2 \cdot P(c, S) \cdot R(c, S)}{P(c, S) + R(c, S)}$$

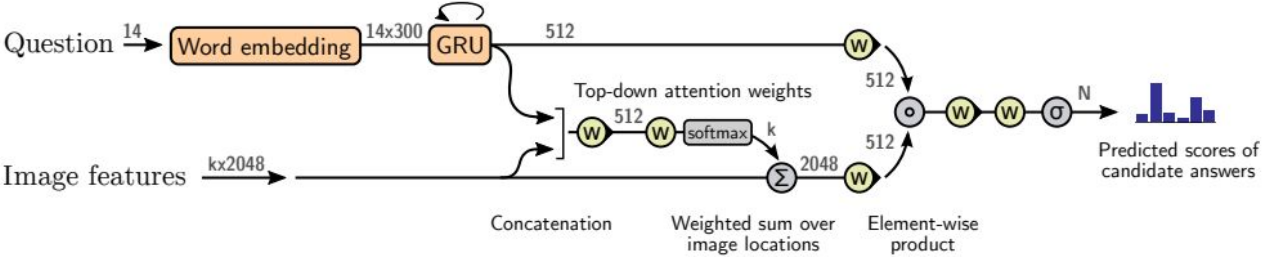
[UpDown]


$y_t$
Softmax
$h_t^2$
$h_{t-1}^2$
Language LSTM
$h_t^2$
$\hat{v}_t$
$\{v_1, \dots, v_k\}$
Attend
$h_t^1$
$h_{t-1}^1$
Top-Down Attention LSTM
$h_t^1$
$h_{t-1}^2$
$\bar{v}$
$W_e \Pi_t$



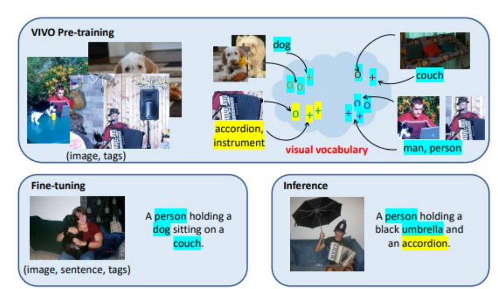

<start>
a
motorcycle
parked
on
a
dirt
field
next
to
a
fence
<end>

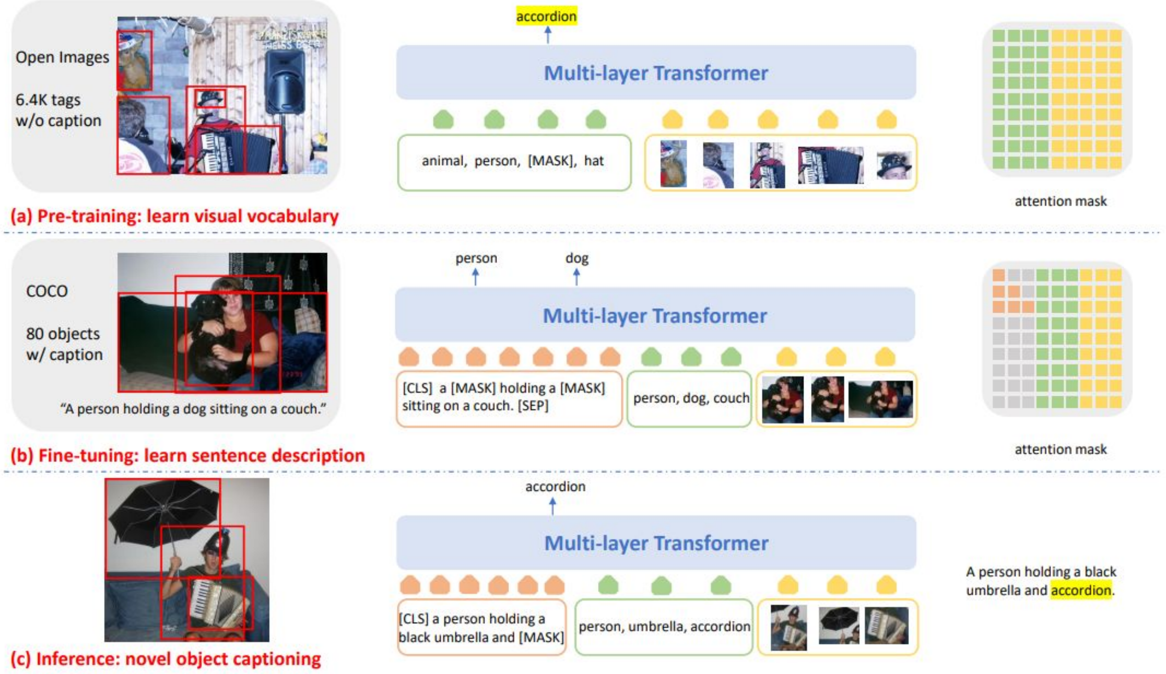

woman
white cabinet
kitchen
red headband
black glasses
black glasses
stove
white outlet
floral dress
silver pot
white stove top
green bottle
black burner
black burner
open oven
food

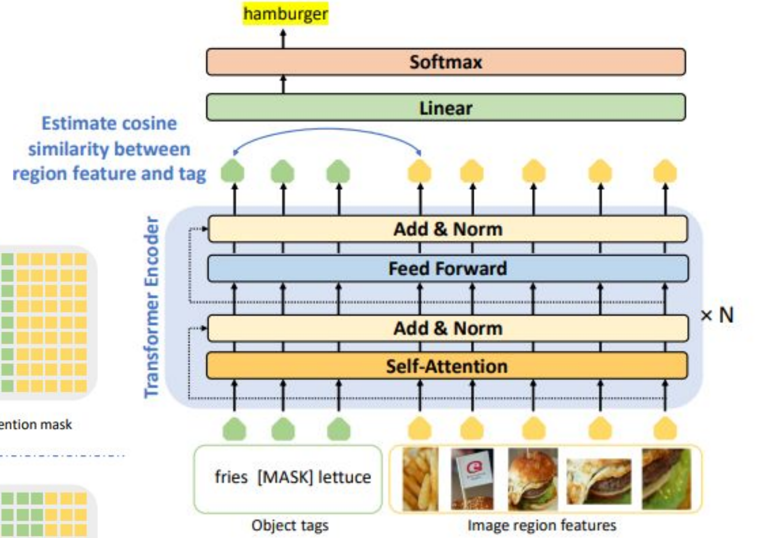

Question
14
Word embedding
14x300
GRU
512
Top-down attention weights
512
softmax
k
Image features
kx2048
2048
512
512
N
Predicted scores of candidate answers
Concatenation
Weighted sum over image locations
Element-wise product

# Image captioning

[VIVO](#)

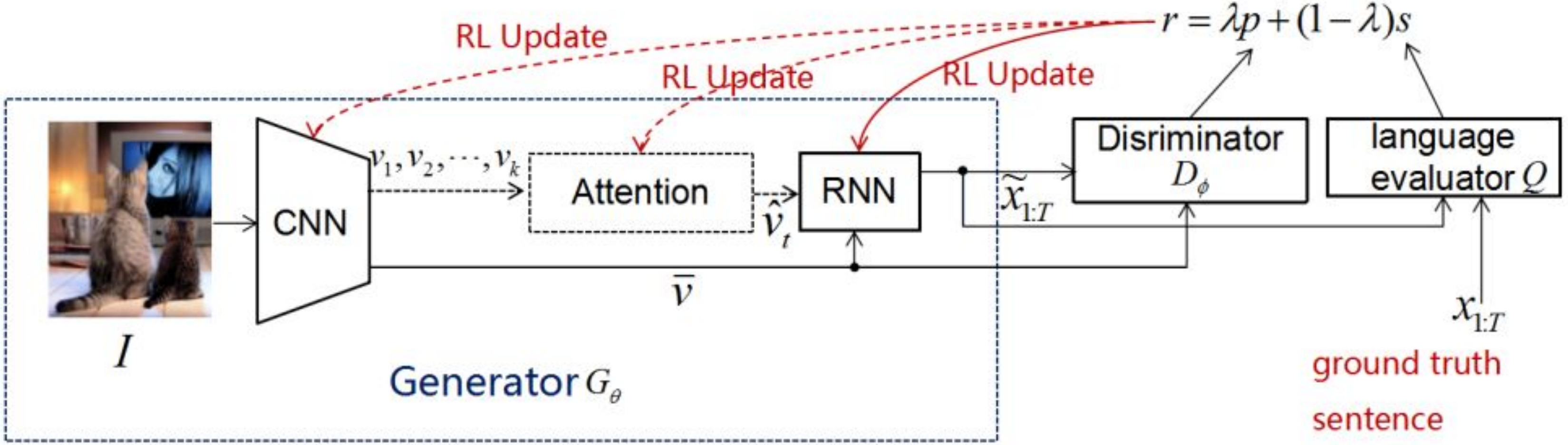


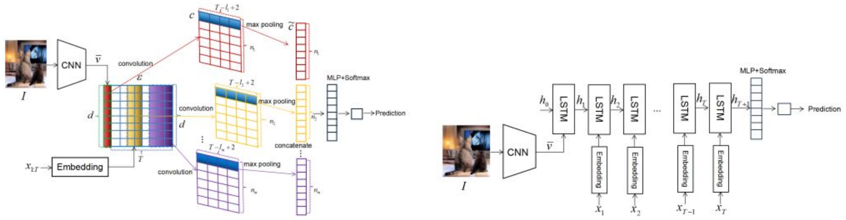



# Image captioning

## IC-GAN




(a) CNN-based discriminator (b) RNN-based discriminator

Figure 2: CNN and RNN-based discriminator architectures. Best viewed in colour.

# Image captioning

Results (28 Jul 2021) of the overall performance on MS COCO Karpathy test split

| Method | CIDEr | SPICE |
|---|---|---|
| Resnet Baseline | 111.1 | 20.2 |
| UpDown | 120.1 | 21.4 |
| MLE Maximization | 110.2 | 20.3 |
| *RL Maximization | 120.4 | 21.3 |
| *MLE + RL Maximization | 119.3 | 21.2 |
| *Meta Learning | 121.0 | 21.7 |
| IC-GAN (Updown/CNN-GAN) | 123.2 | 22.1 |
| IC-GAN (Updown/RNN-GAN) | 122.2 | 22.0 |
| IC-GAN (Updown/ensemble) | **125.9** | **22.3** |

Tutorial

# Text to image

# Text to image

DALL·E is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs.

# Text to image

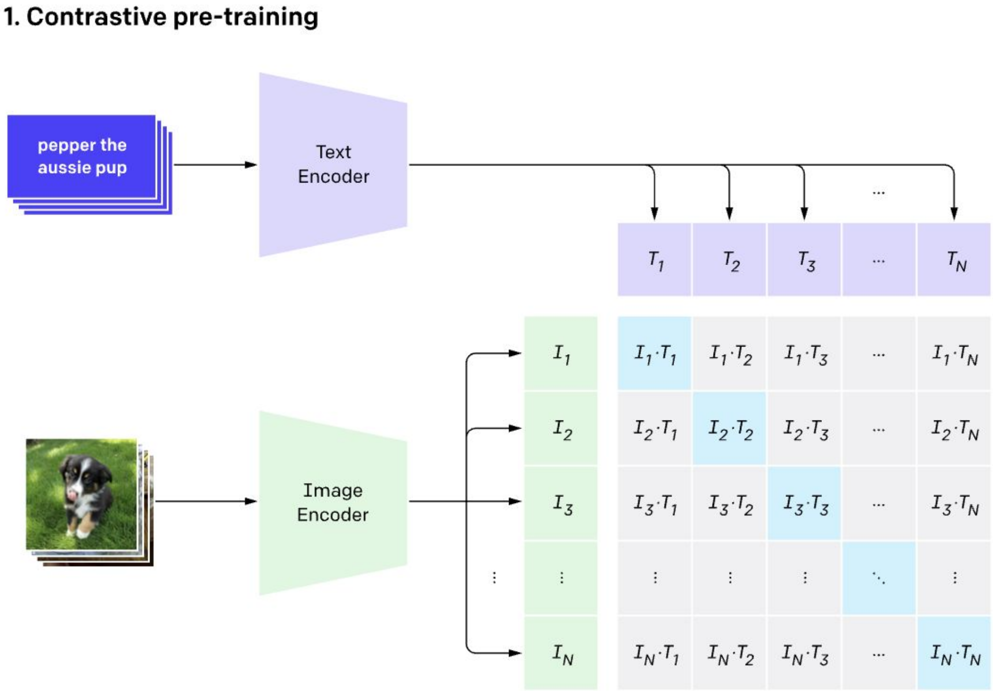
DALL·E

**Stage 1**:  Train a discrete Variational Autoencoder(DVAE) to compress each 256 X 256 RGB image to 32 X 32 grid of image tokens, each element of which can assume 8192 possible values. This reduces the context size of the transformer by a factor of 192 without a large degradation in visual quality.

**Stage 2** : Concatenate up to 256 BPE-encoded text tokens with the 32 × 32 = 1024 image tokens, and train an autoregressive transformer to model the joint distribution over the text and image tokens.

# Text to image

## CLIP (Contrastive Language–Image Pre-training)

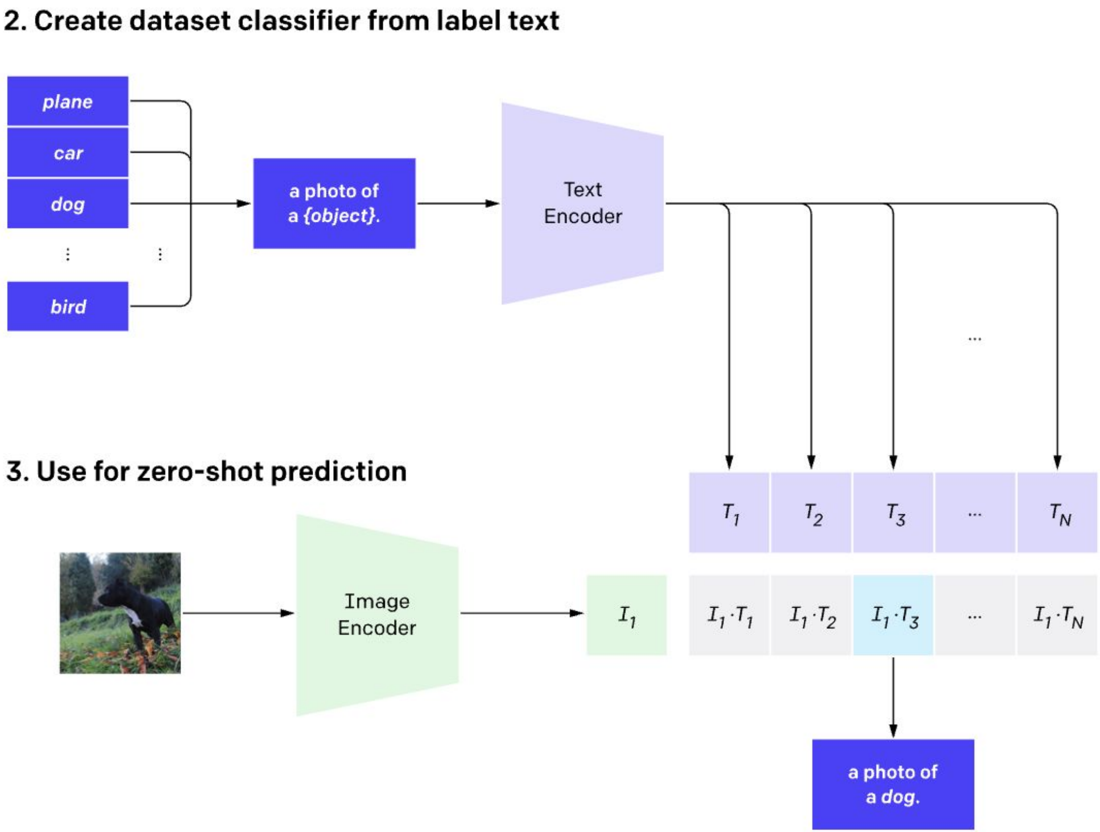



```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

*Figure 3.* Numpy-like pseudocode for the core of an implementation of CLIP.

# Text to image

## CLIP (Contrastive Language–Image Pre-training)



# Text to image

ruDALL-E

- ruDALL-E Kandinsky (XXL) - 12b parameters;
- ruDALL-E Malevich (XL) c 1.3 b parameters.

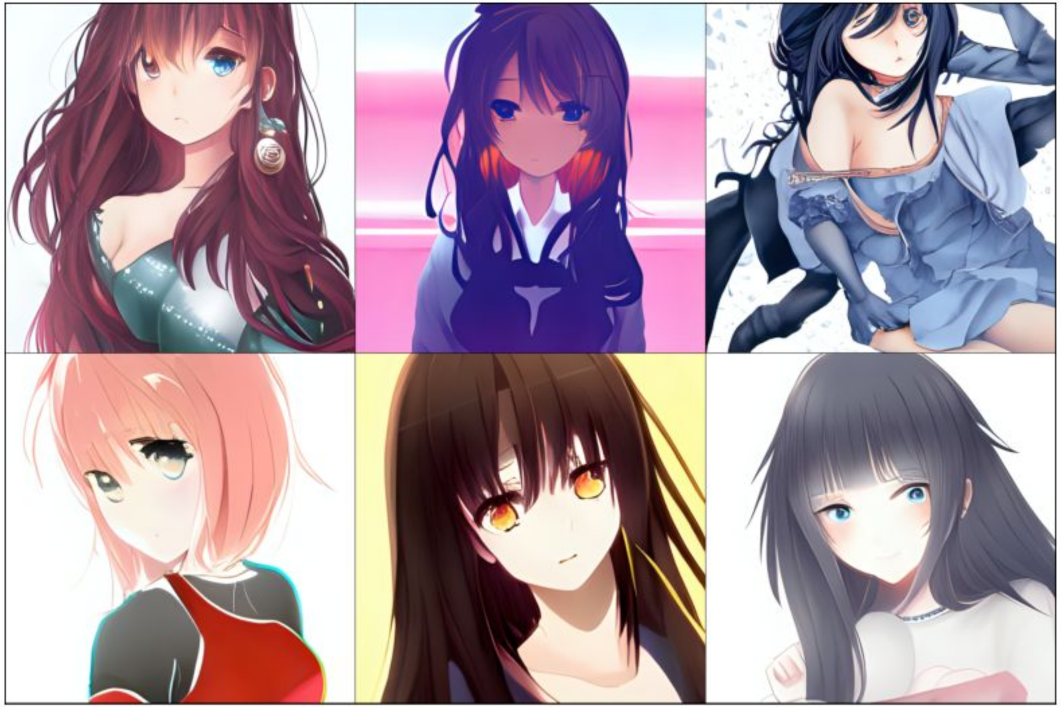
**Pipeline:**

1. ruDALL-E Malevich (XL)
2. Sber VQ-GAN
3. ruCLIP
4. Super Resolution (Real ESRGAN)

# Text to Image

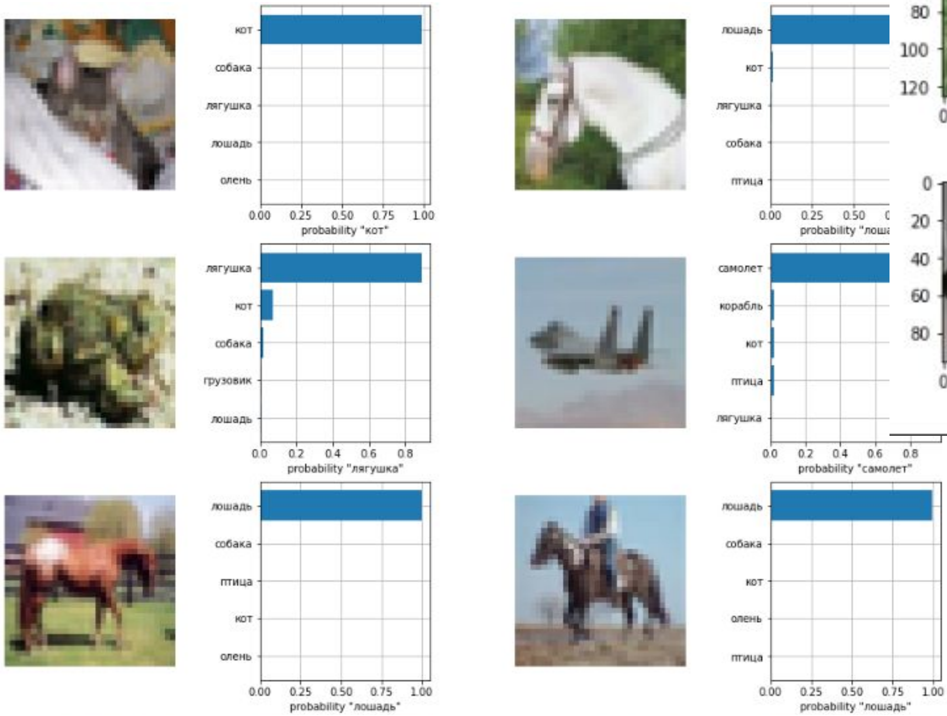
ruDALL-E, finetune colab example

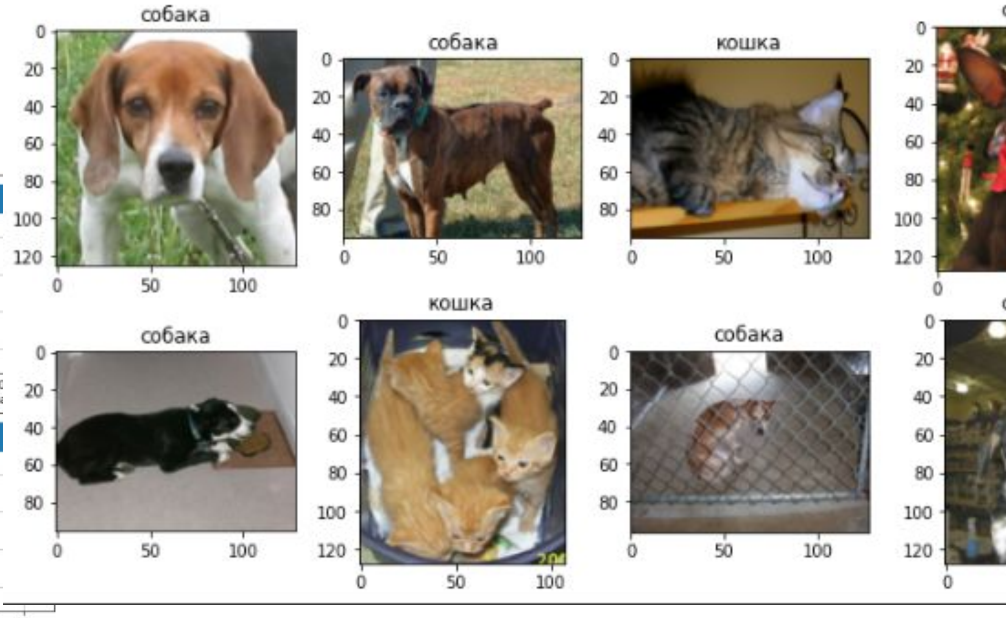
```
text, seed = 'красивая тян из аниме', 6955
```



# Text to Image

ruCLIP, example



# Questions