

# **Deep Learning Szeminárium**

**I. előadás: bemutatkozás, kedvcsináló**

Varga Dániel (Rényi Intézet)

# Előadók:

- Csiszárík Adrián
  - Lukács András
  - Varga Dániel
  - Zombori Zsolt
- 
- Benkő Beatrix
  - Csanády Bálint
  - Czifra Domonkos
  - Kiss Melinda

**<https://bit.ly/elite-ttk-deeplearning>**

**Feltételezett előismeretek:** lineáris algebrai alapismeretek, többváltozós analízis alapismeretek, python alapismeretek

**A jegyszerzést megkönnyítő további előismeretek:** gépi tanulás alapfogalmai, valószínűségszámítási alapismeretek, python tudományos programozás (numpy, scipy)

## **Jegyszerzés módja:**

- A jegyszerzéshez a kiadott 6 házi feladatból legalább 5-öt el kell készíteni.
- A házi feladatok megoldása colab notebook környezetben történik.
- További részletek a feltételekről és elvárásokról a tárgy weblapján hamarosan.

## Ajánlott olvasmányok:

- François Chollet - Deep Learning with Python
- <http://course17.fast.ai/> (tensorflow)
- <http://cs231n.stanford.edu/>
- <http://rll.berkeley.edu/deeprlcourse/>
- <http://www.deeplearningbook.org/>
- Aki nagyon naprakész szeretne lenni, annak <https://www.reddit.com/r/MachineLearning/> vagy Twitteren deep learning hírességek követését ajánljuk.
- <https://github.com/> A legizgalmasabb dolgok elég nagy része open source, vagy van elfogadható minőségű open source reimplementationációja. A kód módosítgatásából lehet a legtöbbet tanulni.

# ...ez mind közkincs:

- pip install tensorflow
- git clone <https://github.com/tensorflow/models> ; cd research/im2txt # image to text
- git clone <https://github.com/hanzhanggit/StackGAN.git> # text to image
- git clone [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN) # object detection & segmentation
- git clone <https://github.com/facebookresearch/DensePose> # human pose detection
- git clone <https://github.com/tensorflow/nmt> # machine translation (seq2seq)
- git clone <https://github.com/tomlepaine/fast-wavenet> # speech synthesis
- git clone <https://github.com/lengstrom/fast-style-transfer> # style transfer
- git clone <https://github.com/NVlabs/stylegan2> # image generation

# GPT-2: csak 174 sor kód

```
147 def model(hparams, X, past=None, scope='model', reuse=False):
148     with tf.variable_scope(scope, reuse=reuse):
149         results = {}
150         batch, sequence = shape_list(X)
151
152         wpe = tf.get_variable('wpe', [hparams.n_ctx, hparams.n_embd],
153                               initializer=tf.random_normal_initializer(stddev=0.01))
154         wte = tf.get_variable('wte', [hparams.n_vocab, hparams.n_embd],
155                               initializer=tf.random_normal_initializer(stddev=0.02))
156         past_length = 0 if past is None else tf.shape(past)[-2]
157         h = tf.gather(wte, X) + tf.gather(wpe, positions_for(X, past_length))
158
159         # Transformer
160         presents = []
161         pasts = tf.unstack(past, axis=1) if past is not None else [None] * hparams.n_layer
162         assert len(pasts) == hparams.n_layer
163         for layer, past in enumerate(pasts):
164             h, present = block(h, 'h%d' % layer, past=past, hparams=hparams)
165             presents.append(present)
166             results['present'] = tf.stack(presents, axis=1)
167             h = norm(h, 'ln_f')
168
169         # Language model loss. Do tokens <n predict token n?
170         h_flat = tf.reshape(h, [batch*sequence, hparams.n_embd])
171         logits = tf.matmul(h_flat, wte, transpose_b=True)
172         logits = tf.reshape(logits, [batch, sequence, hparams.n_vocab])
173         results['logits'] = logits
174         return results
```

**Megpróbálom végigcsinálni ezt az előadást úgy, hogy egyáltalán nem mondom el, hogy miből épül fel egy mesterséges neuronháló.**

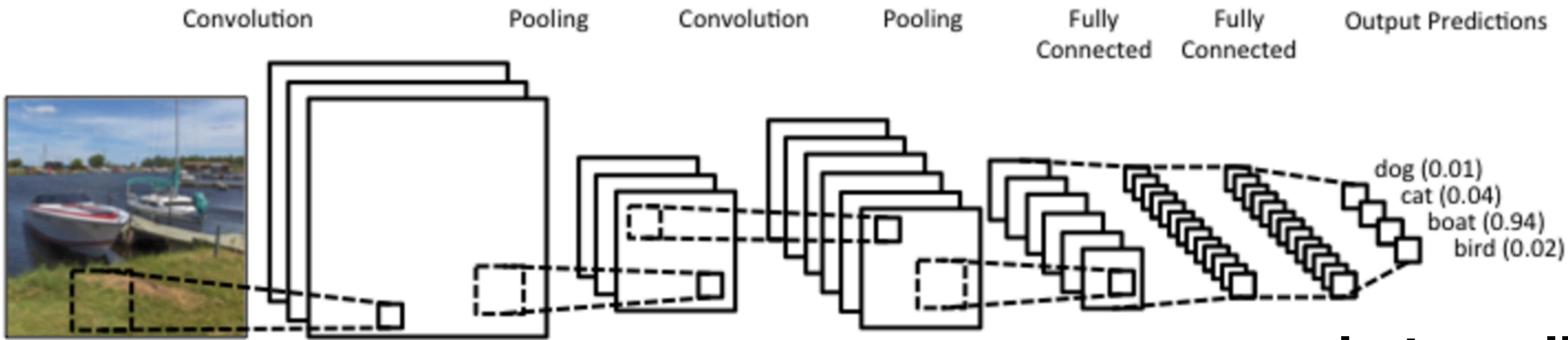
**Jövő héten Zombori Zsolt teljesen pontos definíciót ad majd.**

# Egy fekete doboz nagyon sok potméterrel



# Ennél azért egy kicsit részletesebben

- valós vektorokat valós vektorokba képező folytonos függvények paraméterezett családja
- bemenetek (kép, hang, videó, szöveg, térbeli pozíció, bármilyen idősor, ...)
- kimenetek (a fentiek közül bármelyik, klasszifikációnál valószínűségek vektora)
- veszteségfüggvény (Legegyszerűbb esetben  $L(y, \hat{y})$ , azaz hogy mennyire vagyok elégedett a kapott  $\hat{y}$  kimenettel az elvárt  $y$  kimenet ismeretében. Lehet pl. négyzetes hiba.)
- tanítás (paraméter-hangolás elvárt bemenet-kimenet párok halmazának felhasználásával, hogy a paraméterezett függvénycsaládból egy számunkra alkalmas konkrét elemet kiválasszunk.)

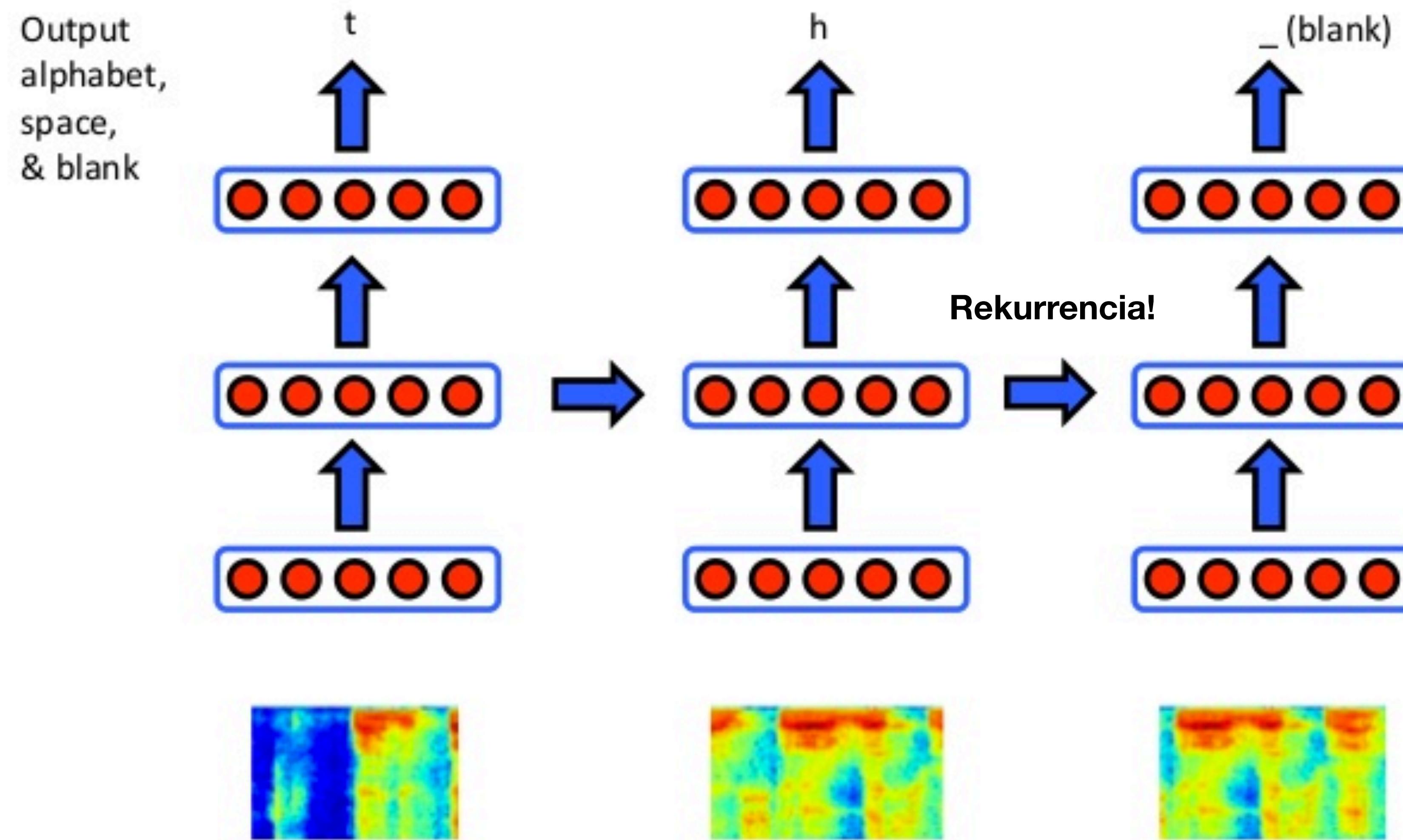


**RGB pixel bemenet  
(224x224x3 dim)**

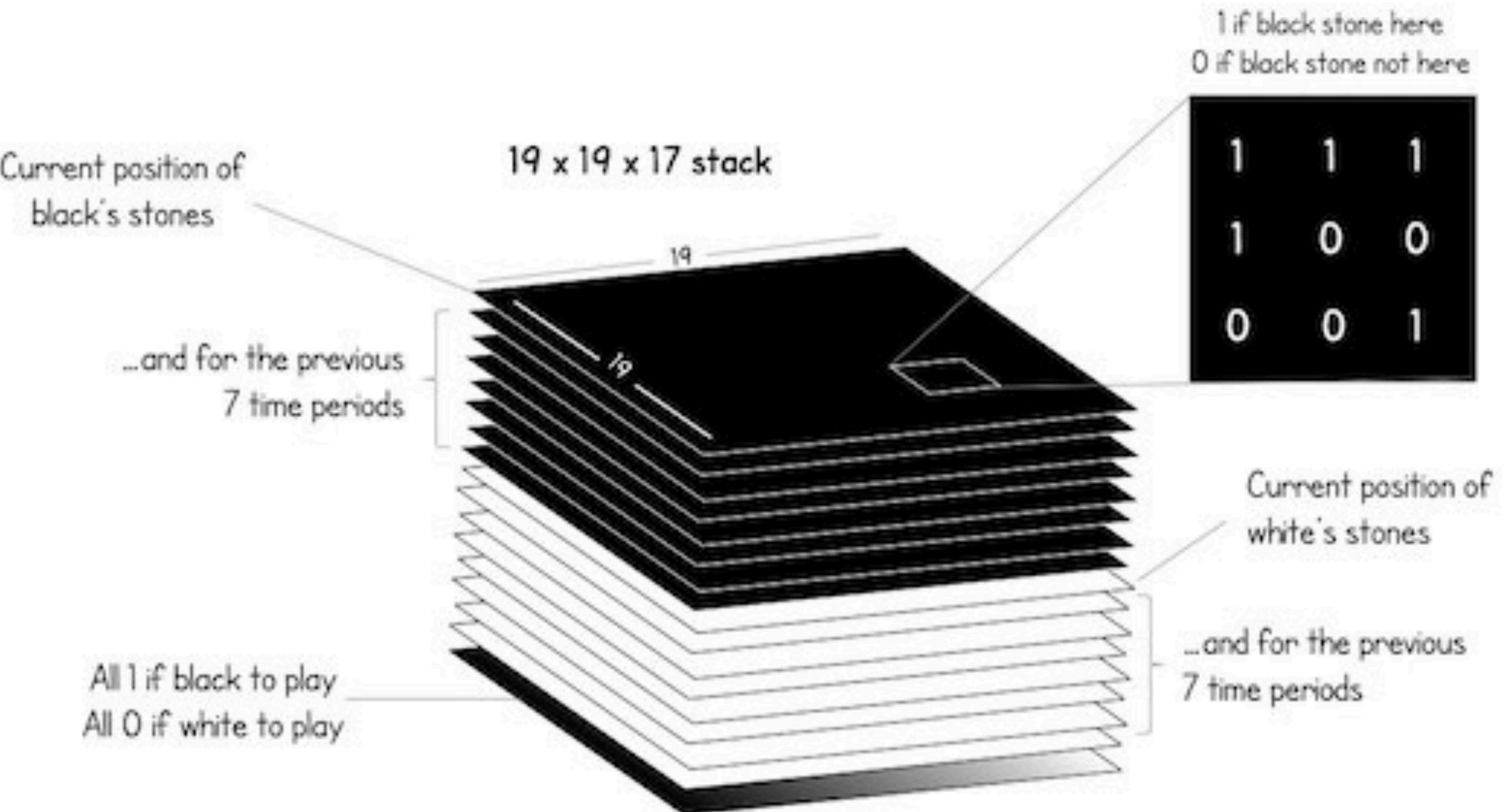
**kategorikus  
kimenet  
(4 dim)**

# Kimenet: pixelenkénti klasszifikáció (haj vagy nem haj)





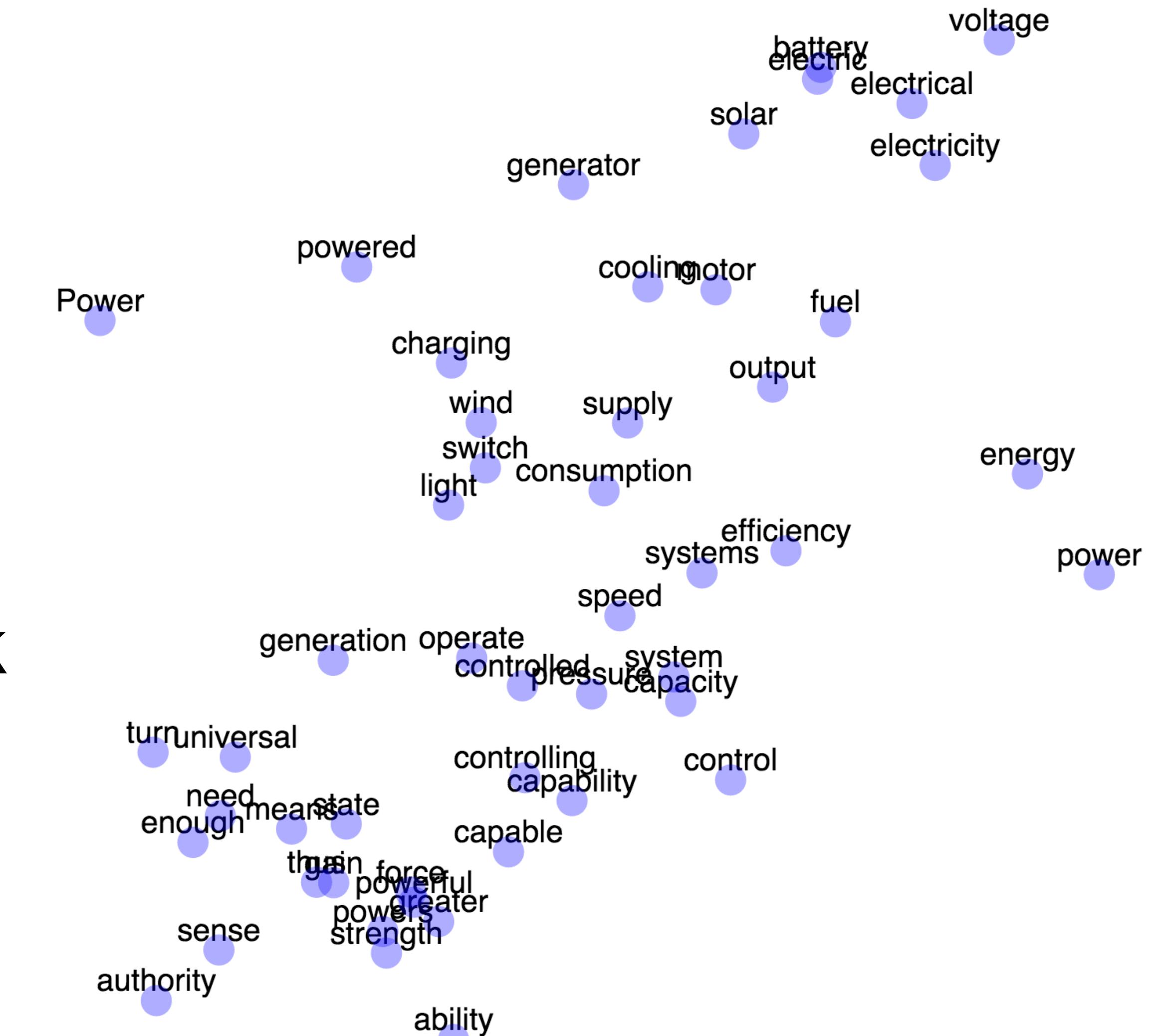
# az AlphaGo Zero játékállás -reprezentációja



This stack is the input to the deep neural network

A word embedding módszereknek hála a szavak is reprezentálhatók valós vektorokként, és így használhatók neuronhálók bemeneteként és kimeneteként.

A beágyazás során megpróbálunk közeli vektorokba képezni olyan szavakat, amiknek a szövegkörnyezetük gyakran hasonló.

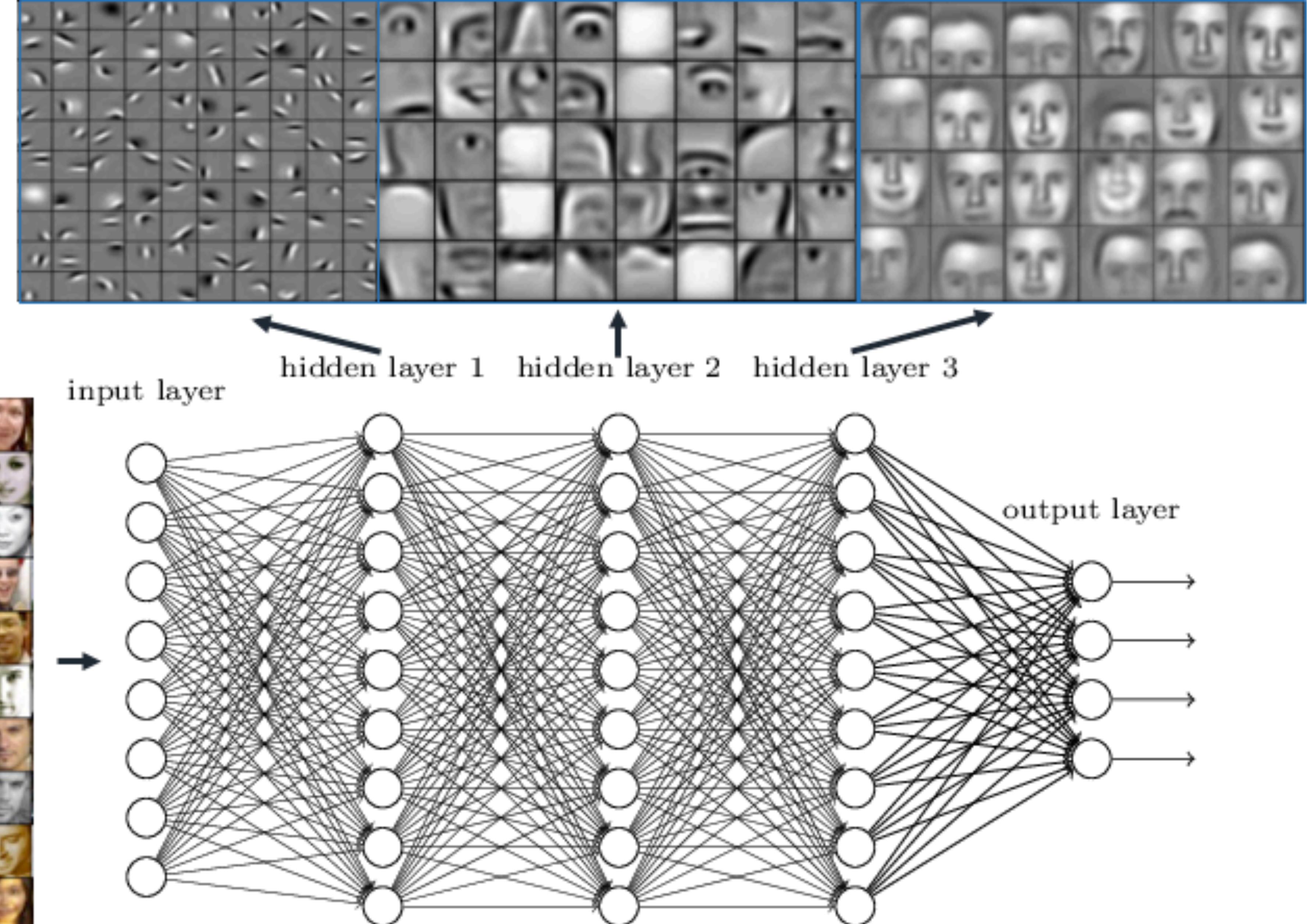


# A neuronháló modellek építésének fő lépései (nem okvetlenül pontosan ebben a sorrendben)

- tanulóadatok (bemenet-kimenet párok) összegyűjtése
- veszteségfüggvény megtervezése
- neurális architektúra megtervezése (milyen paraméterezett függénycsaládok milyen kompozíciójaként álljon elő a neuronháló)
- while kudarc:
  - hiperparaméter-hangolás, architektúra-hangolás
  - tanítás

# Látens reprezentációk

Deep neural  
networks learn  
hierarchical feature  
representations



**<https://distill.pub/2017/feature-visualization/>**

**<https://distill.pub/2017/feature-visualization/appendix/>**

## POSITIVE CHANNEL



Channel Objective



Dataset examples

## NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

## POSITIVE CHANNEL



Channel Objective

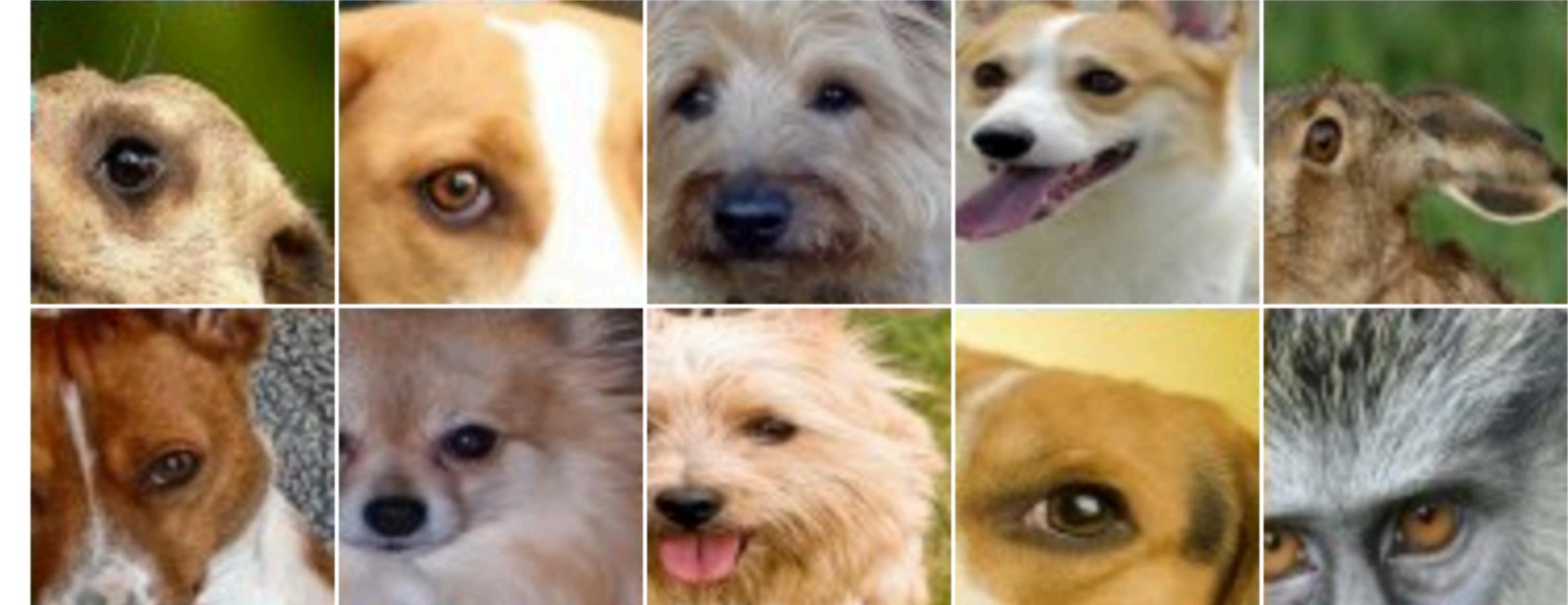


Dataset examples

## NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

## POSITIVE CHANNEL



Channel Objective



Dataset examples

## NEGATIVE CHANNEL



Negative Channel

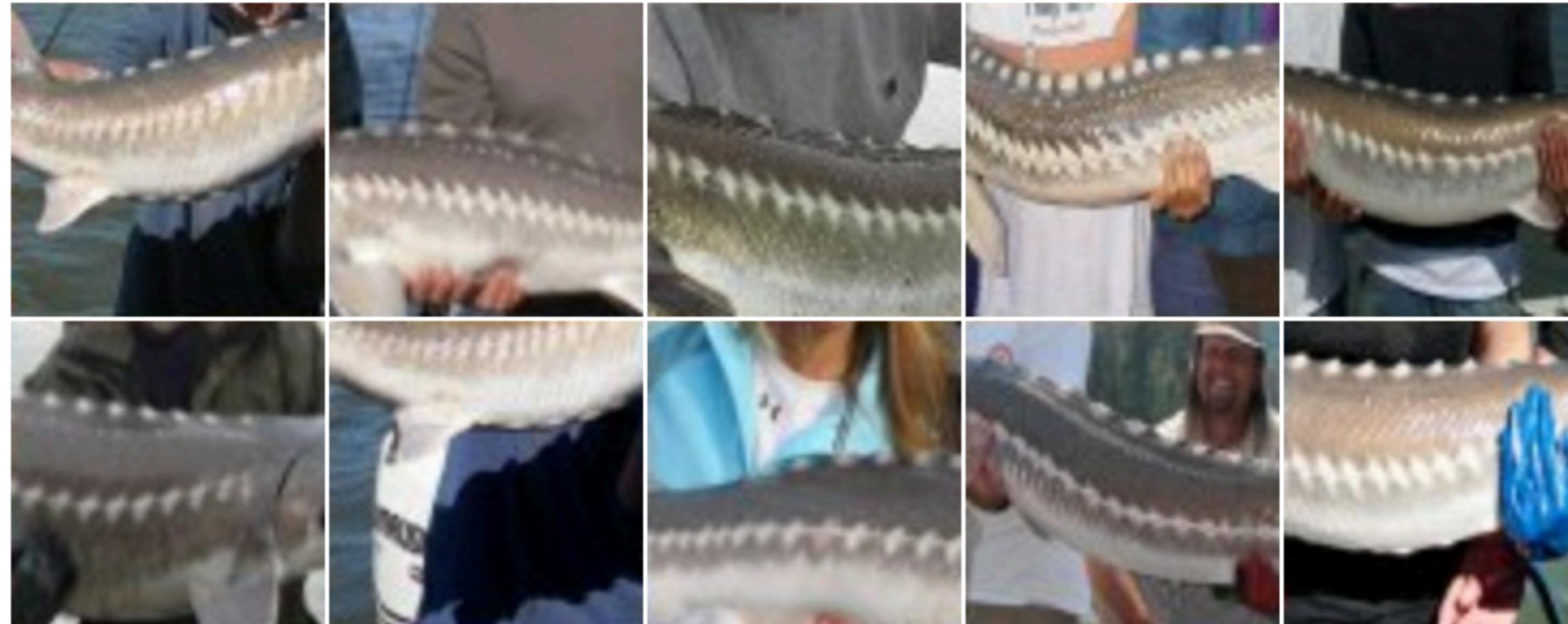


Negative dataset examples

## POSITIVE CHANNEL



Channel Objective

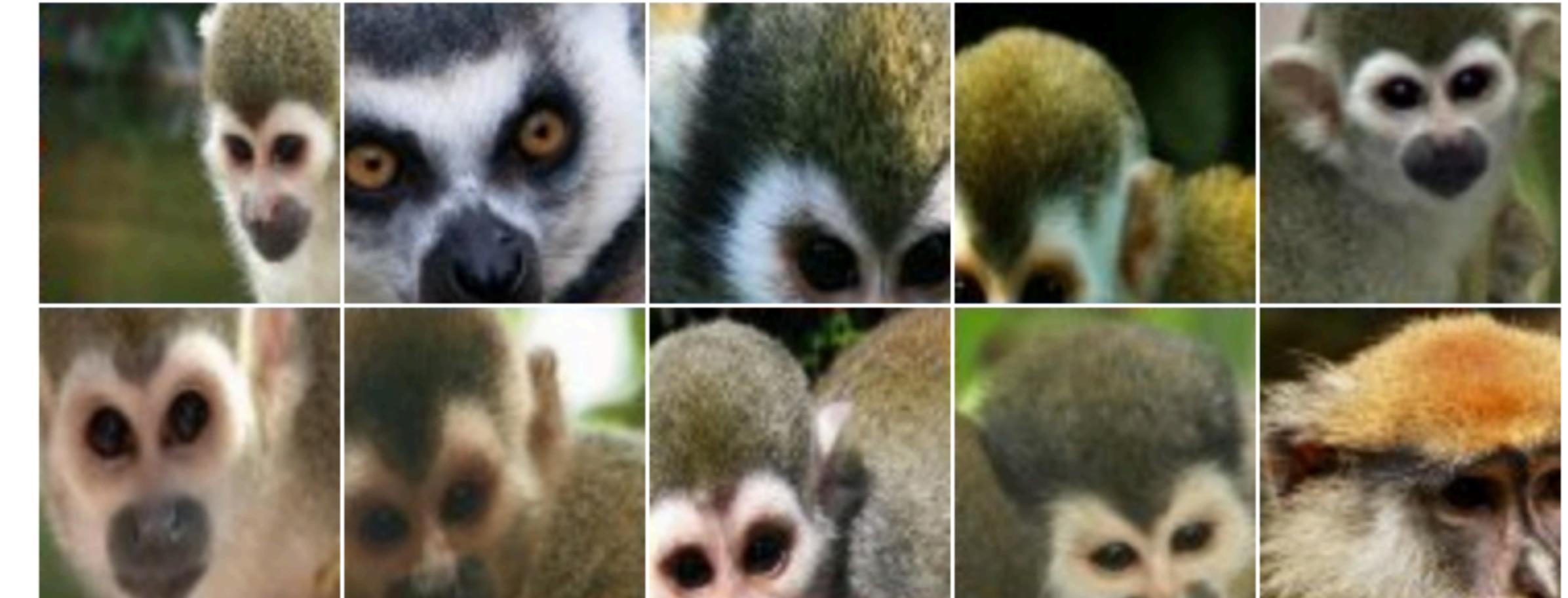


Dataset examples

## NEGATIVE CHANNEL

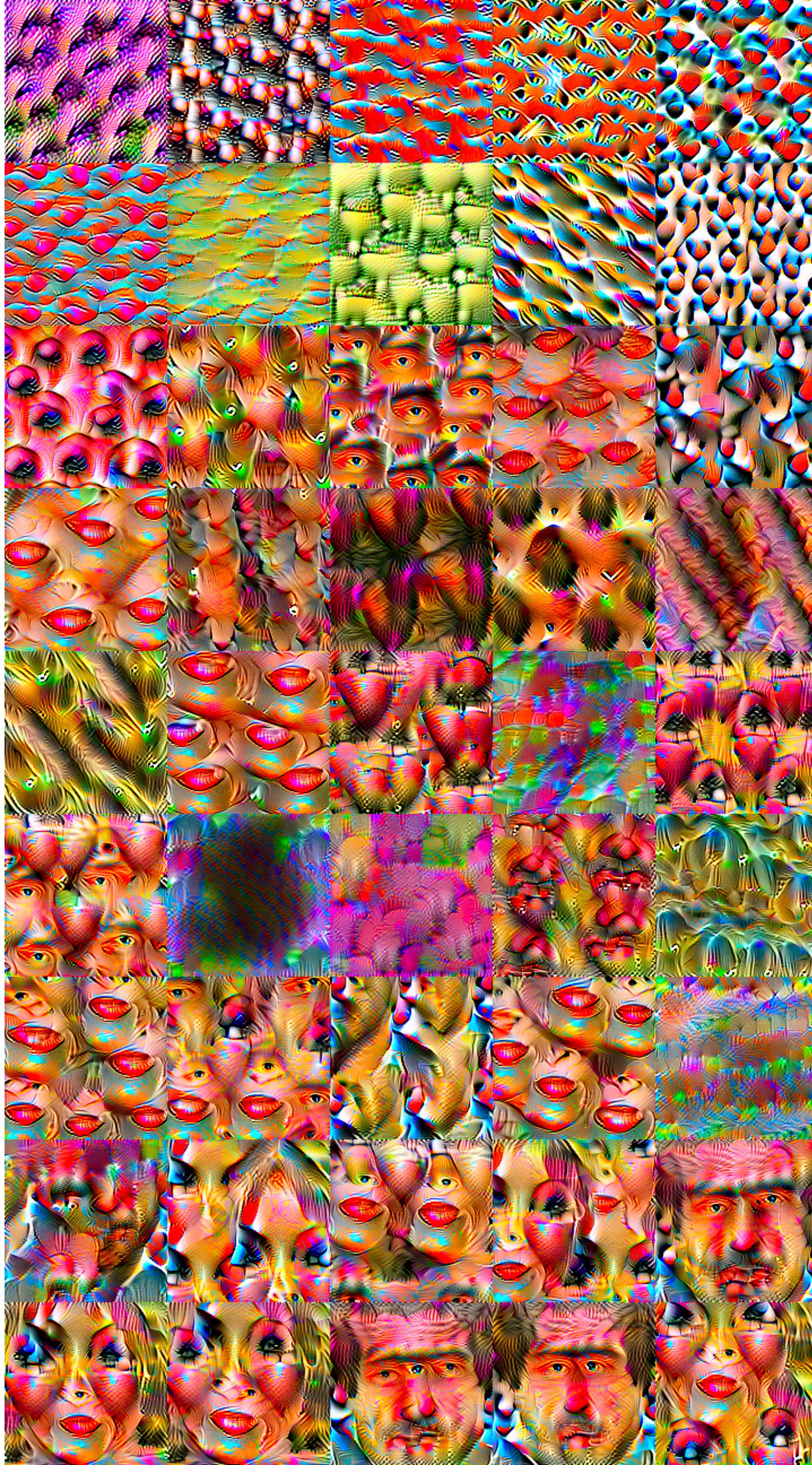


Negative Channel



Negative dataset examples

**Bemenet**



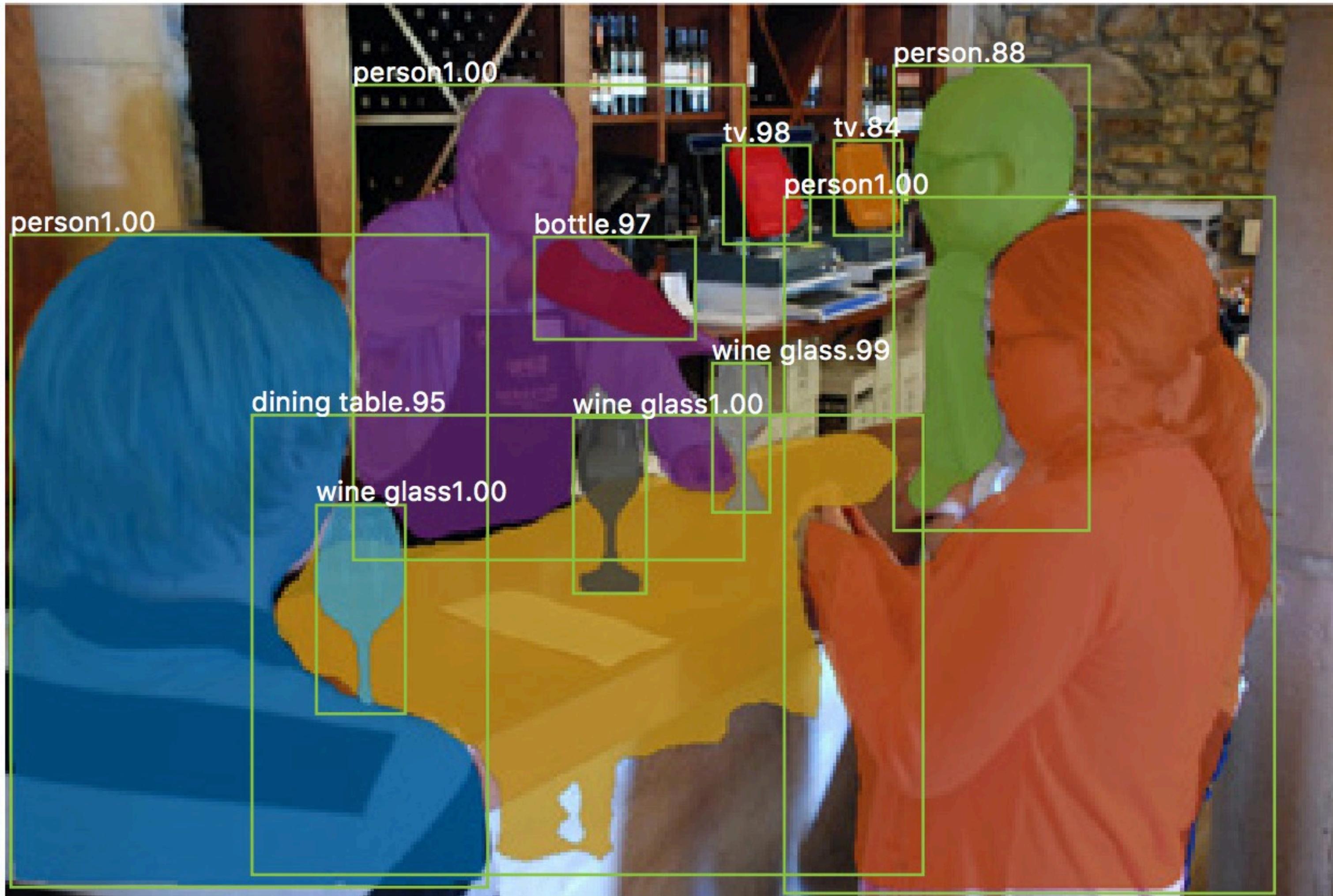
**Kimenet**

# **Alkalmazások**

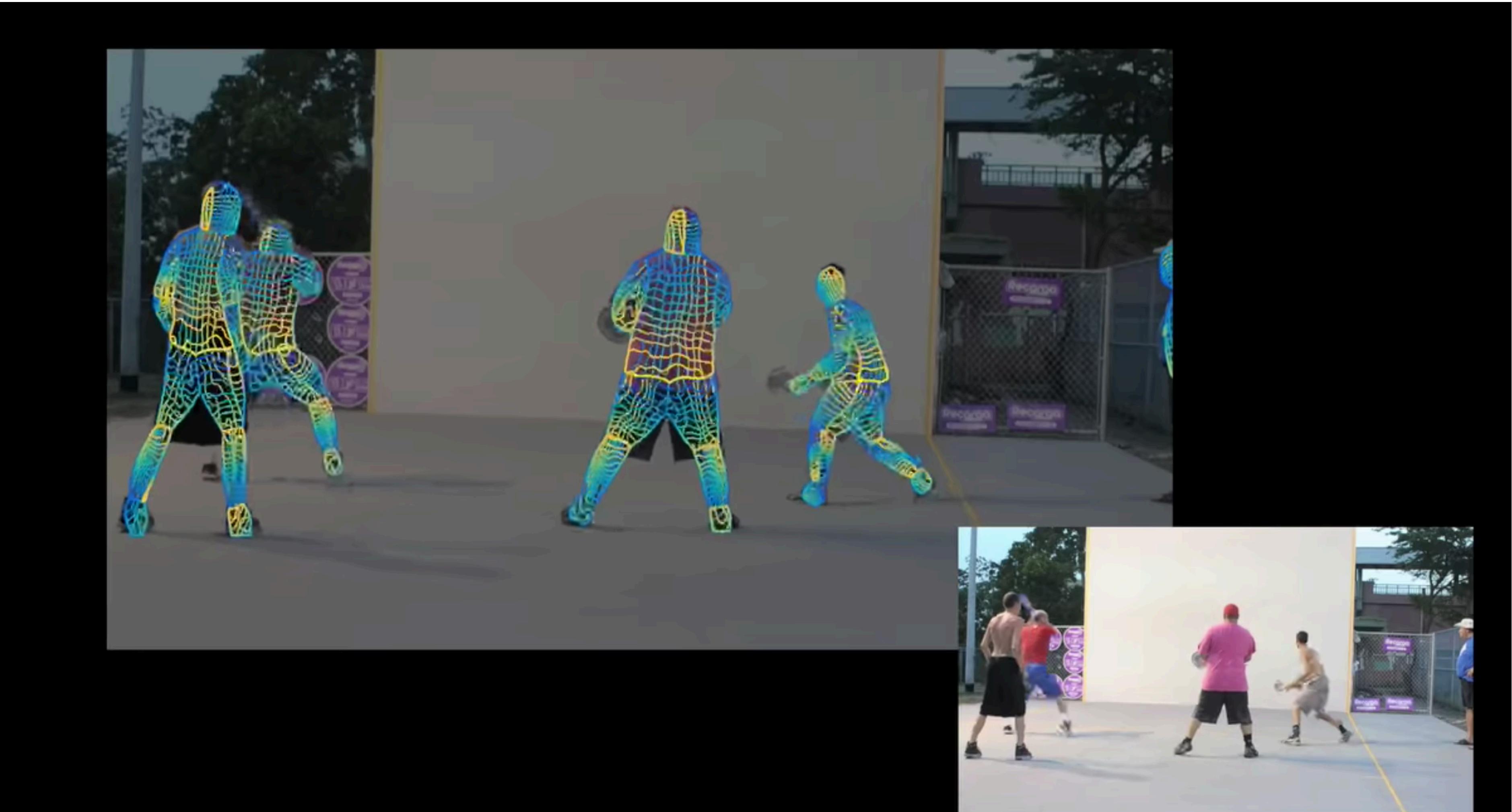
# Alkalmazások

Nem reprezentatív lista, főleg az alapján válogatva, hogy szerintem mennyire érdekes vagy látványos az alkalmazás.

# Mask R-CNN - képszegmentálás



# DensePose - pose estimation



# StackGAN - text to image

Text  
description

This bird is blue with white and has a very short beak



Stage-I  
images

This bird has wings that are brown and has a yellow belly



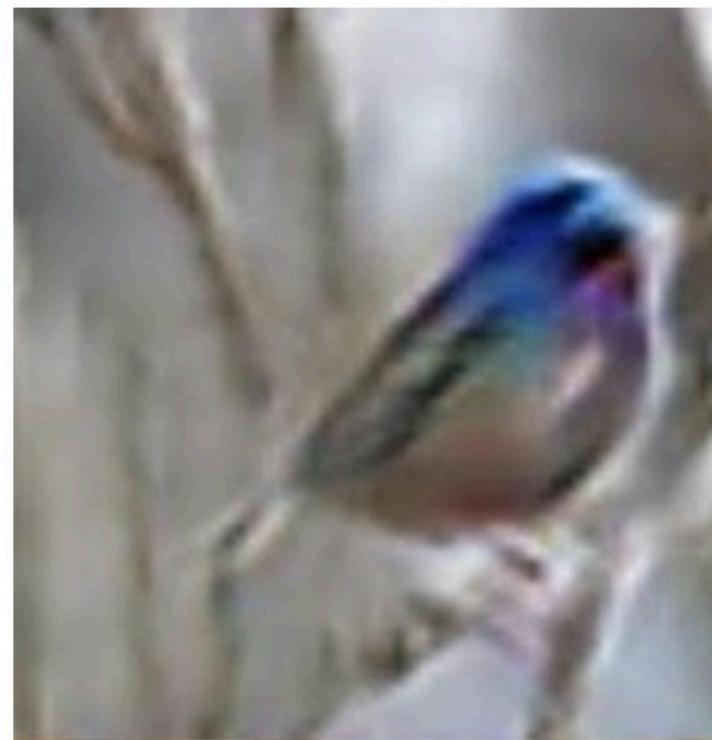
A white bird with a black crown and yellow beak



This bird is white, black, and brown in color, with a brown beak



The bird has small beak, with reddish brown crown and gray belly



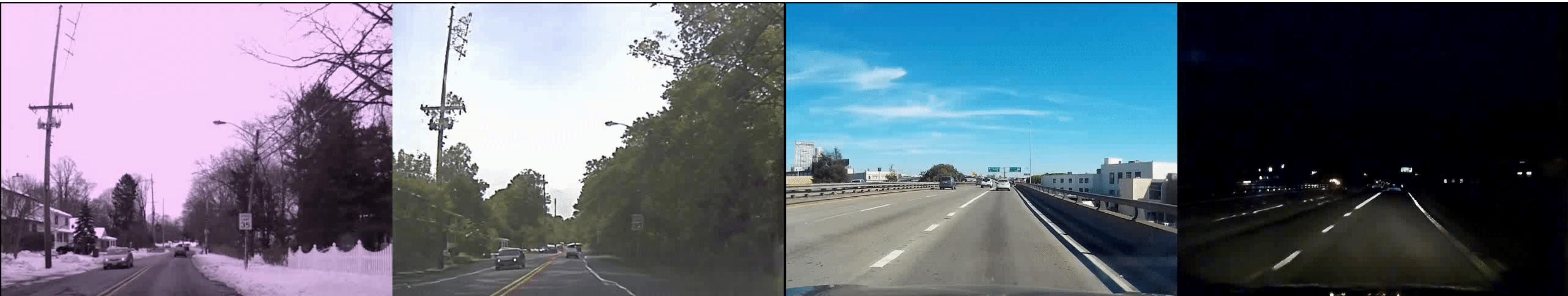
This is a small, black bird with a white breast and white on the wingbars.



Stage-II  
images



# Unsupervised Image-to-image Translation



# DeepFakes

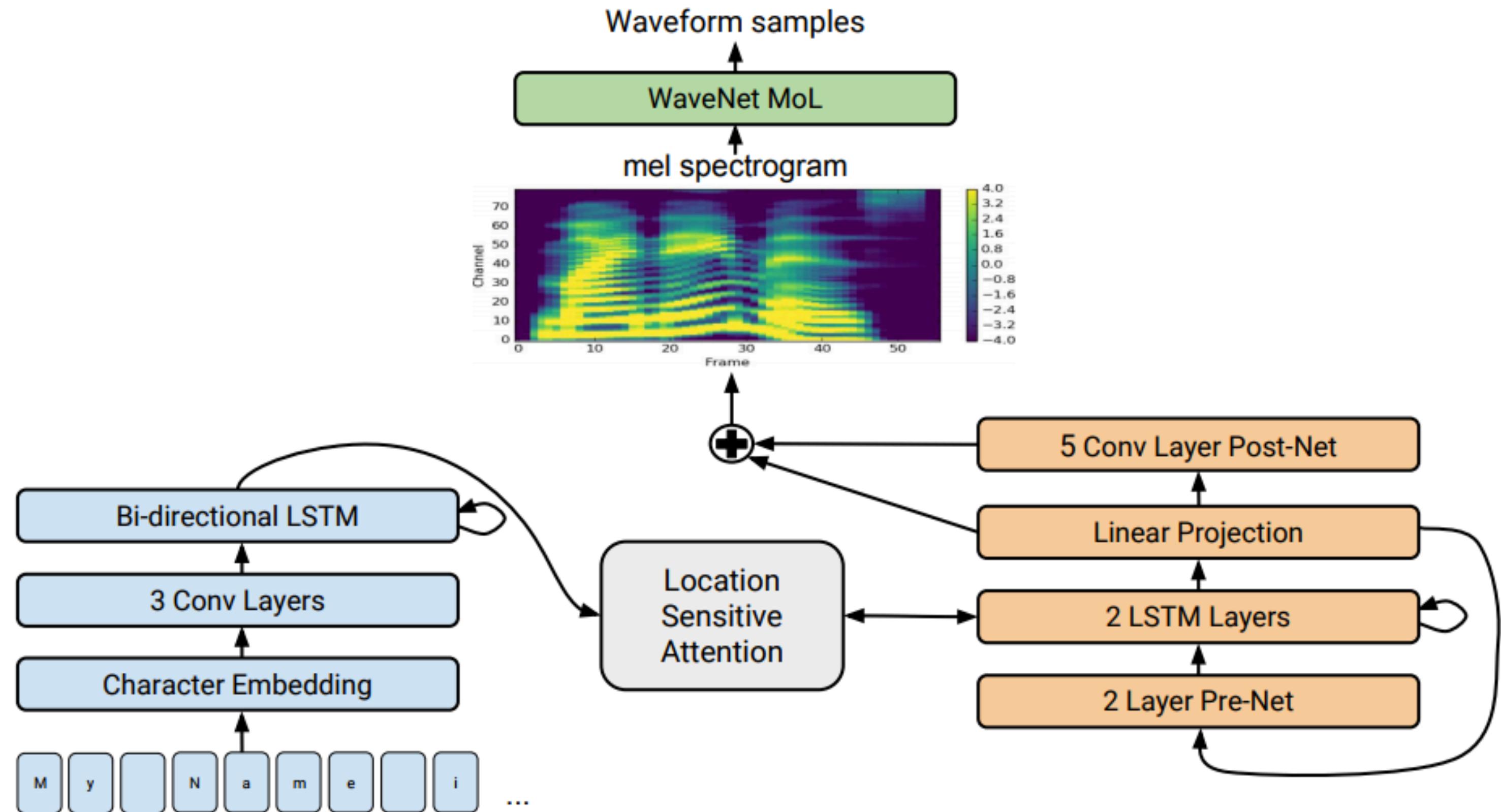


vídeo



# Tacotron 2 - beszédszintézis

[link: Tacotron 2 szintézis példák](#)



# GPT-2, nyelvmodellezés

SYSTEM PROMPT  
(HUMAN-WRITTEN)

*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

MODEL COMPLETION  
(MACHINE-WRITTEN,  
10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Cherry-picked

Kawasaki Z1000

I recently acquired this bike from its original owner. It has always been a good bike in every regard, but I found some problems along the way: It has some wear and tear on the engine bay, the front end has been modified to a wider track and the suspension was modified as well. The bike has been running very well, and the engine is still working perfectly (I have changed oil and filters and both of my oil changes since acquiring this bike).

The bike is still in working order and it has all the original items including the oil filter. I am giving this kz1000 a thorough overhaul with a complete overhaul in the oil system and front end as well as some complete brakes and exhaust work.

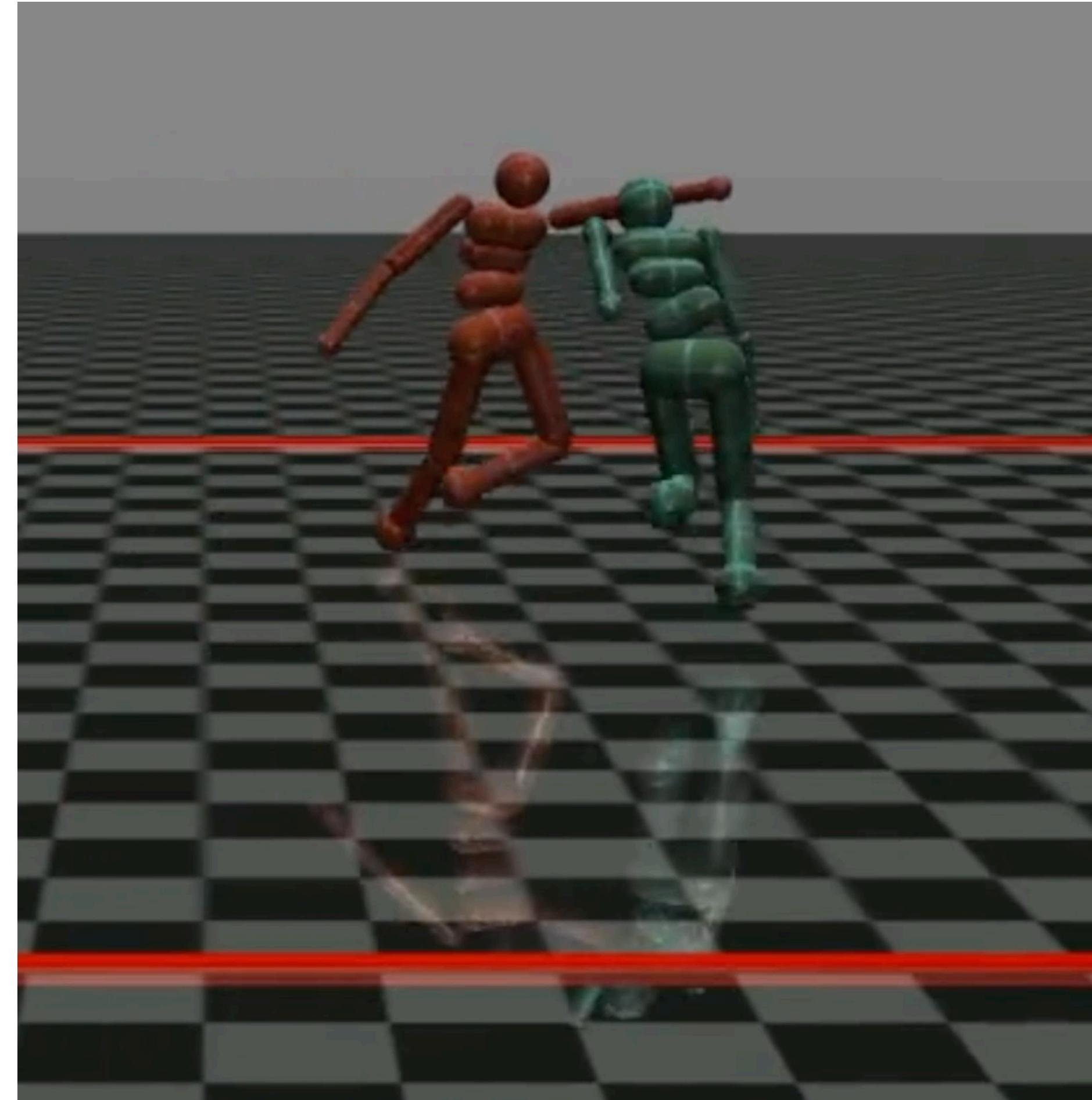
As a side project, I also have a very old motorbike that is a great engine swap to take these bikes to the next level.

Non-cherry-picked

# Mély megerősítéses tanulás szimulált 3D környezetben

videó: tanult mozgás

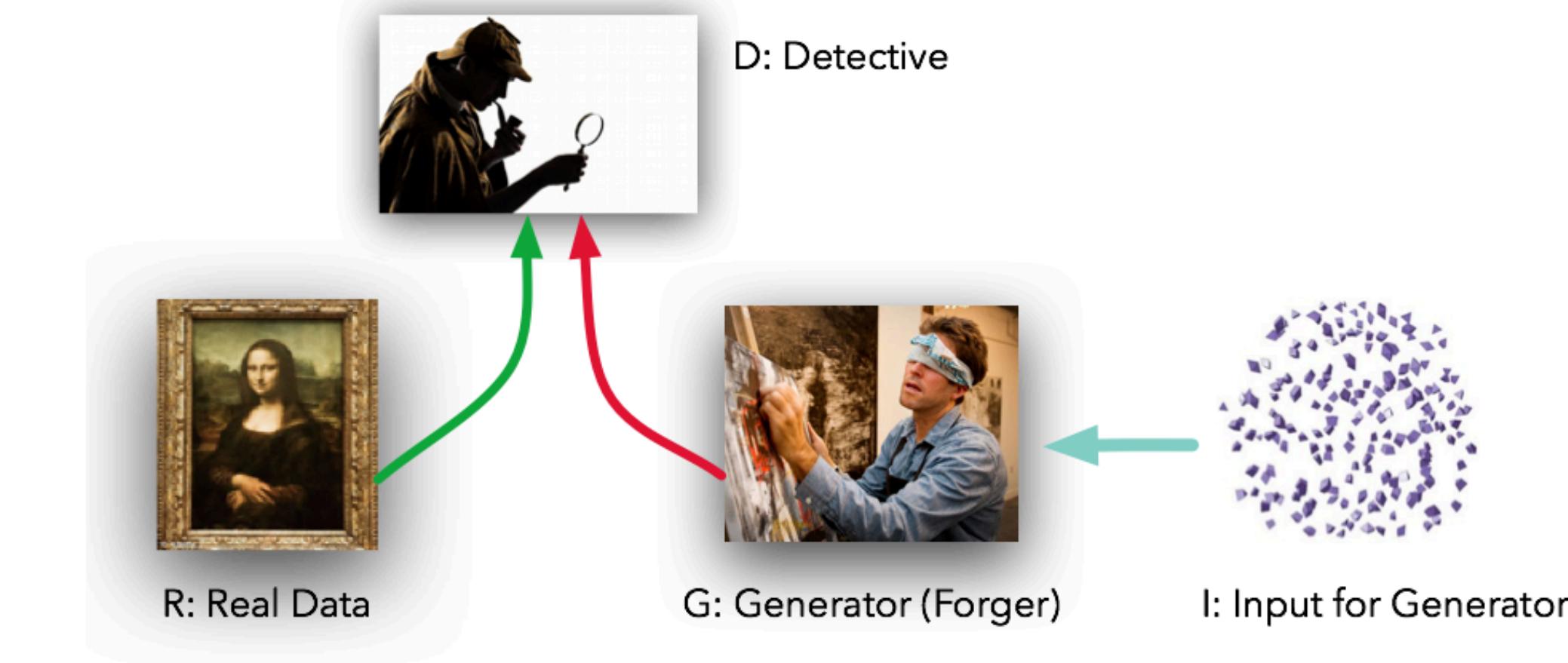
videó: competitive self-play



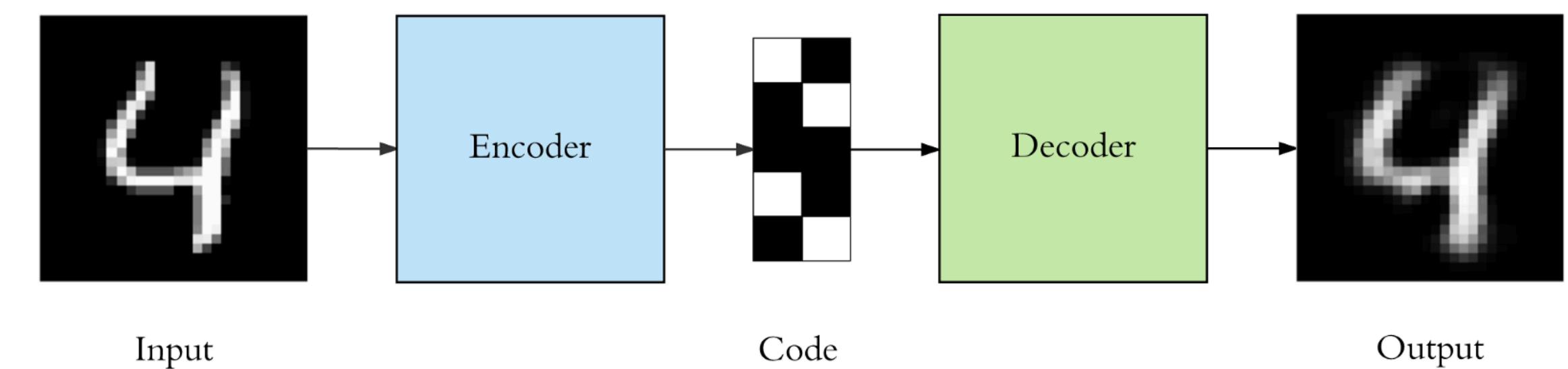
# **Generatív neuronhálók**

# Generatív neuronhálók

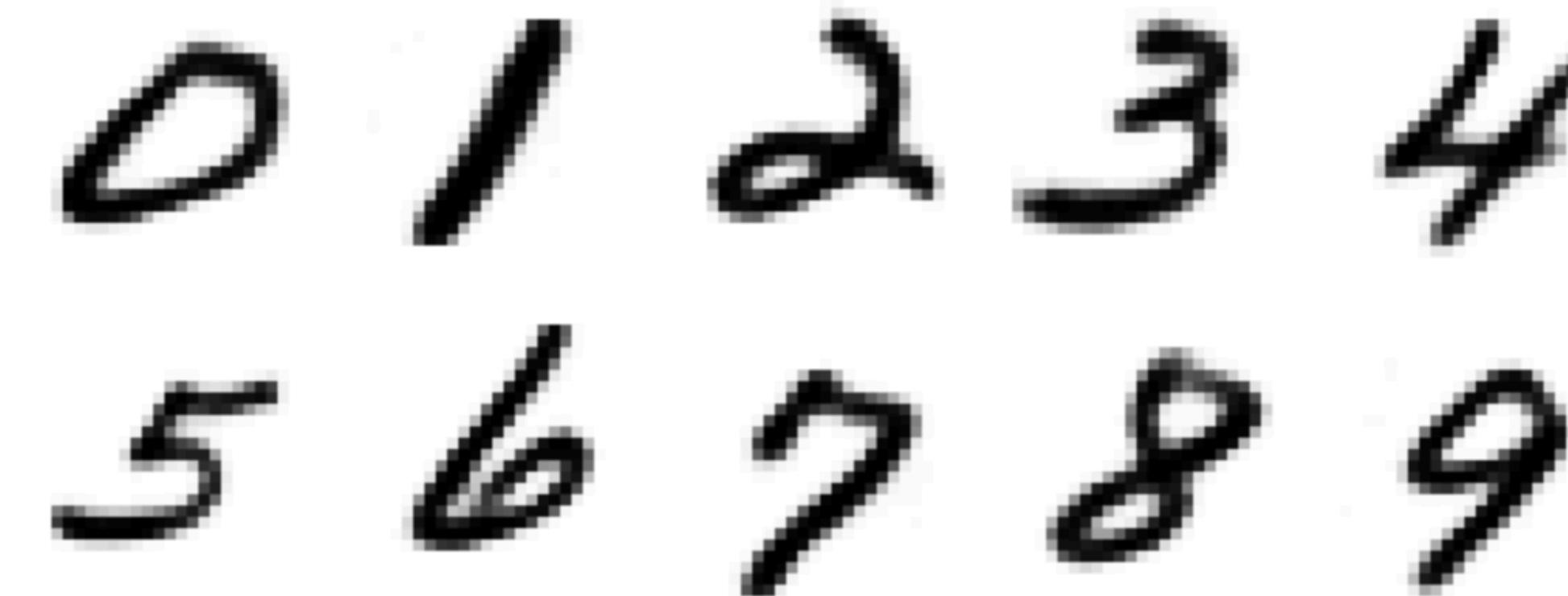
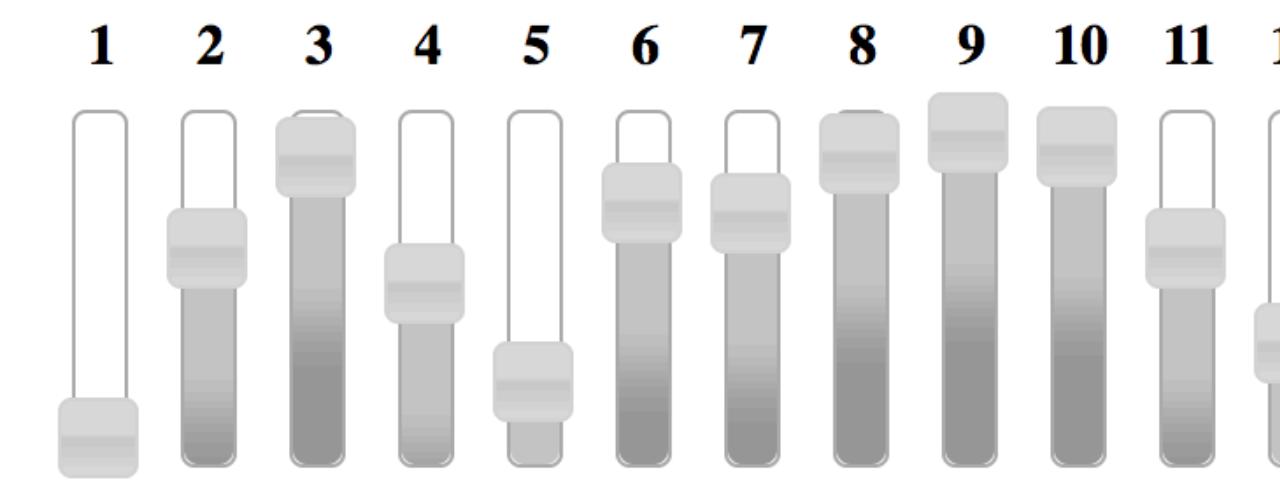
Generative Adversarial Network



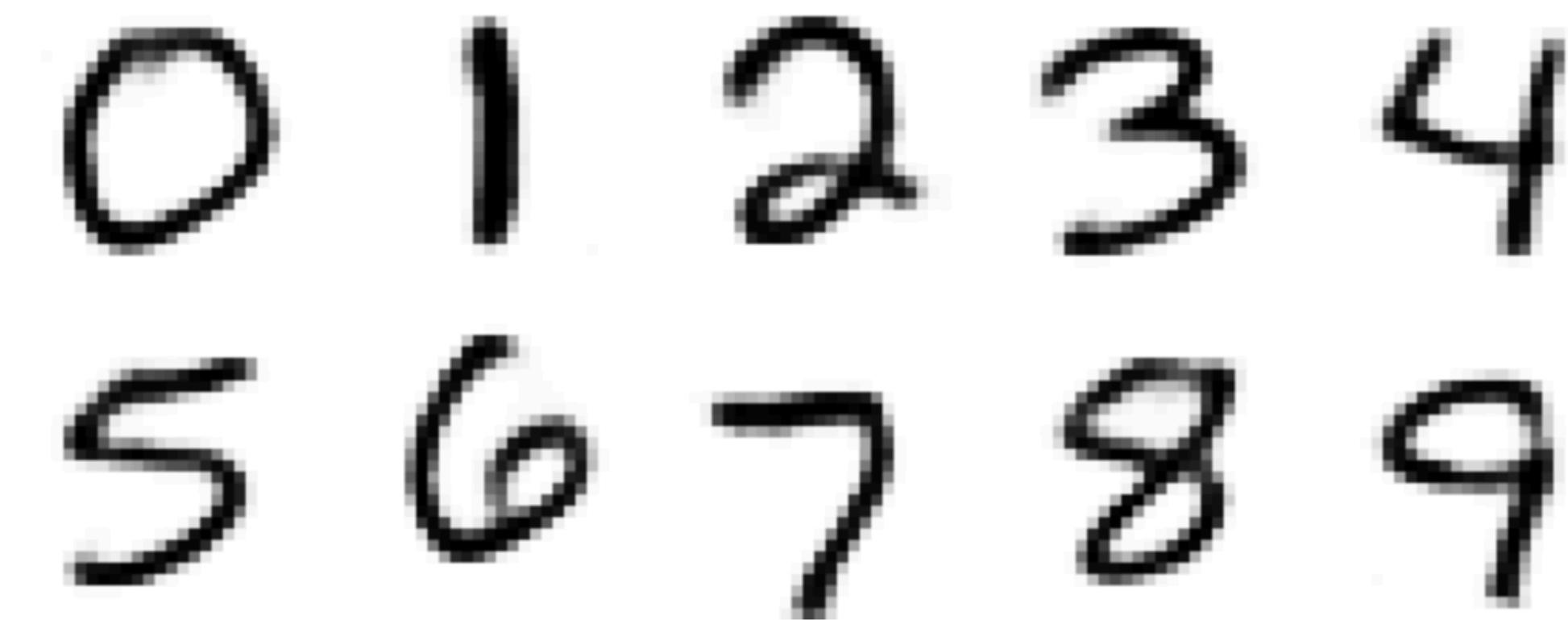
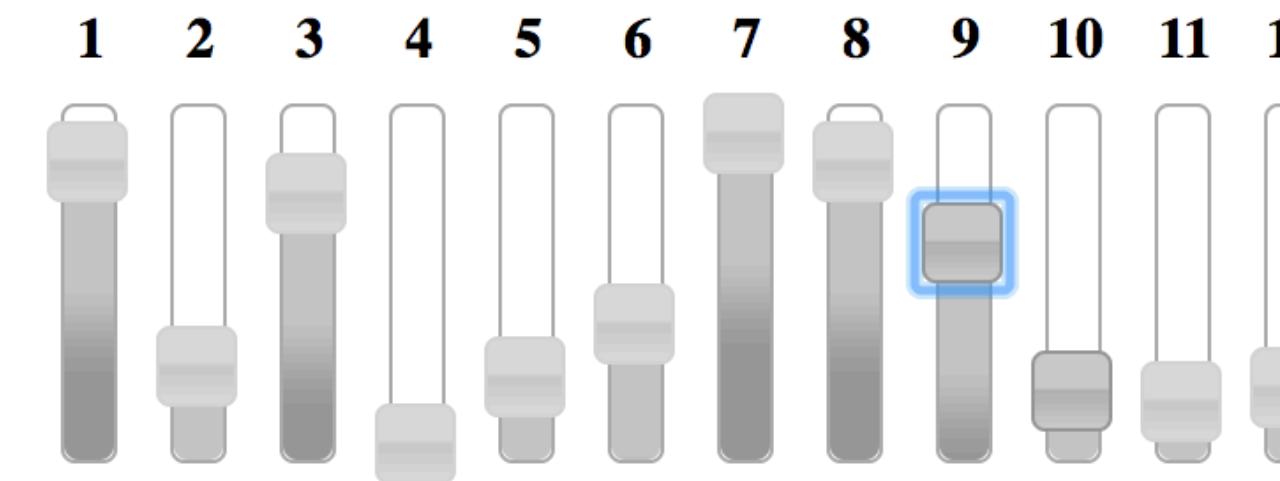
Variational Autoencoder

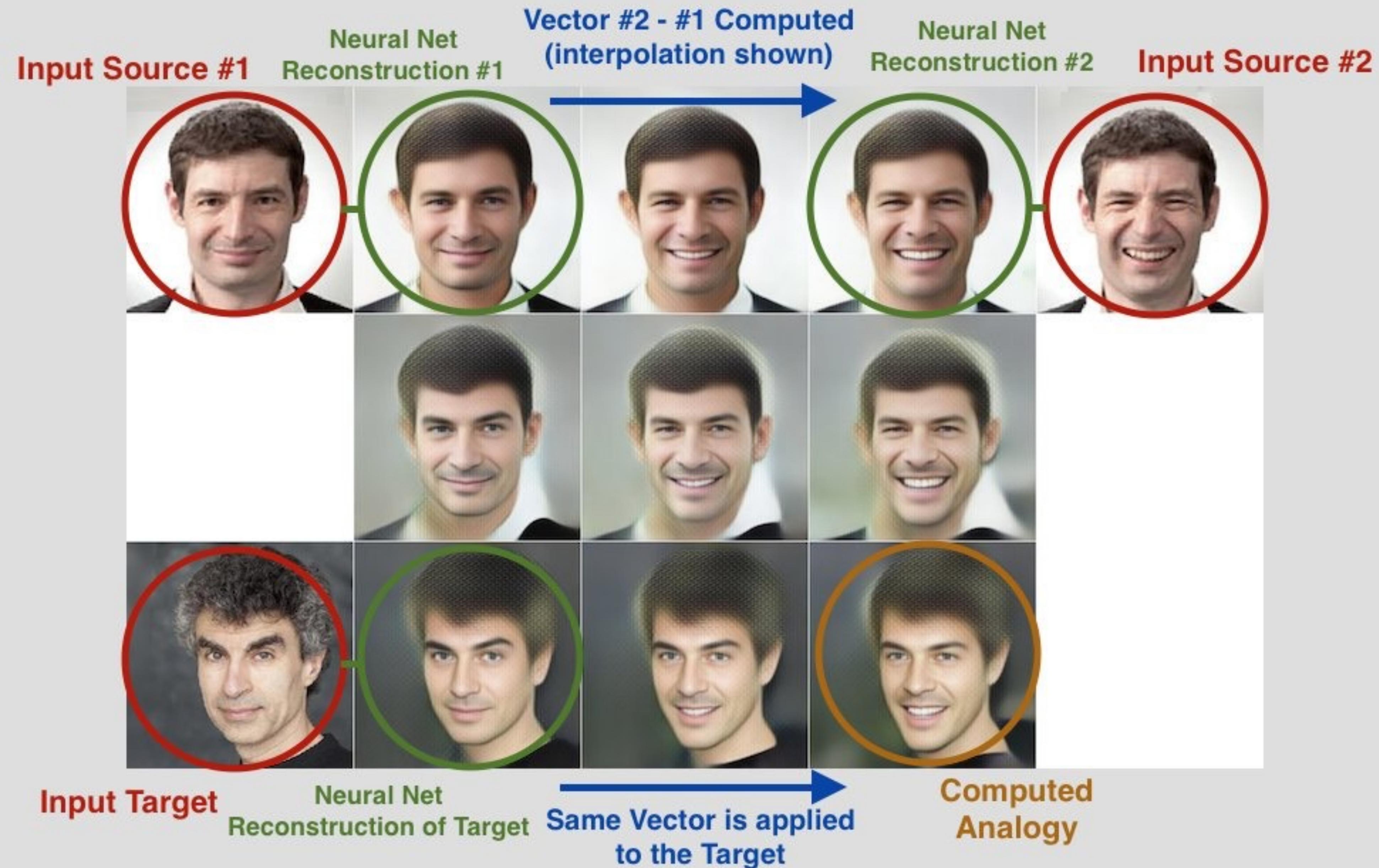


# Látens reprezentációk generatív modellekben



## Digits autoencoder demo





The relationship between Source #1 and Source #2 is applied to the target image.

Coeff: -1.0



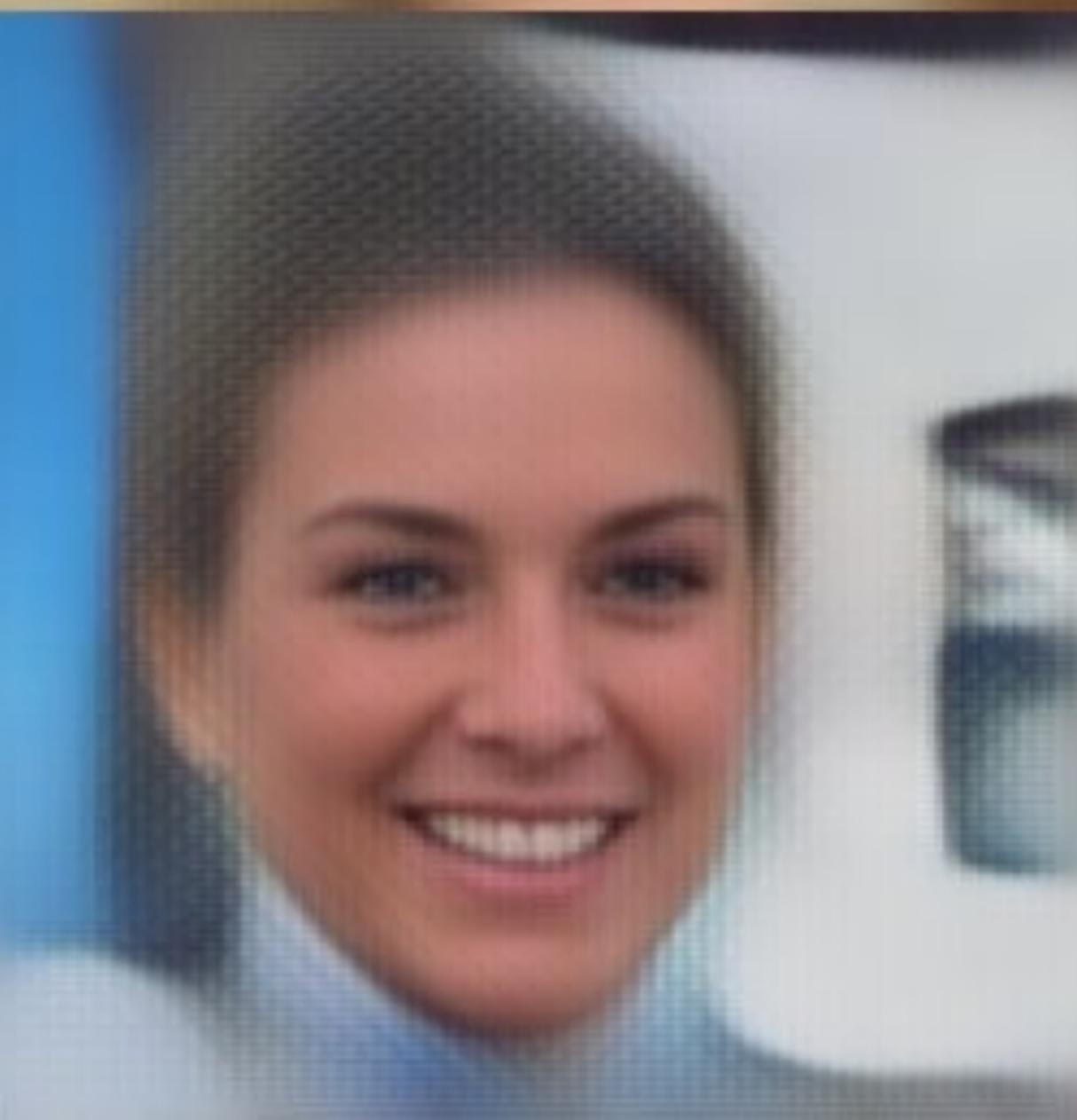
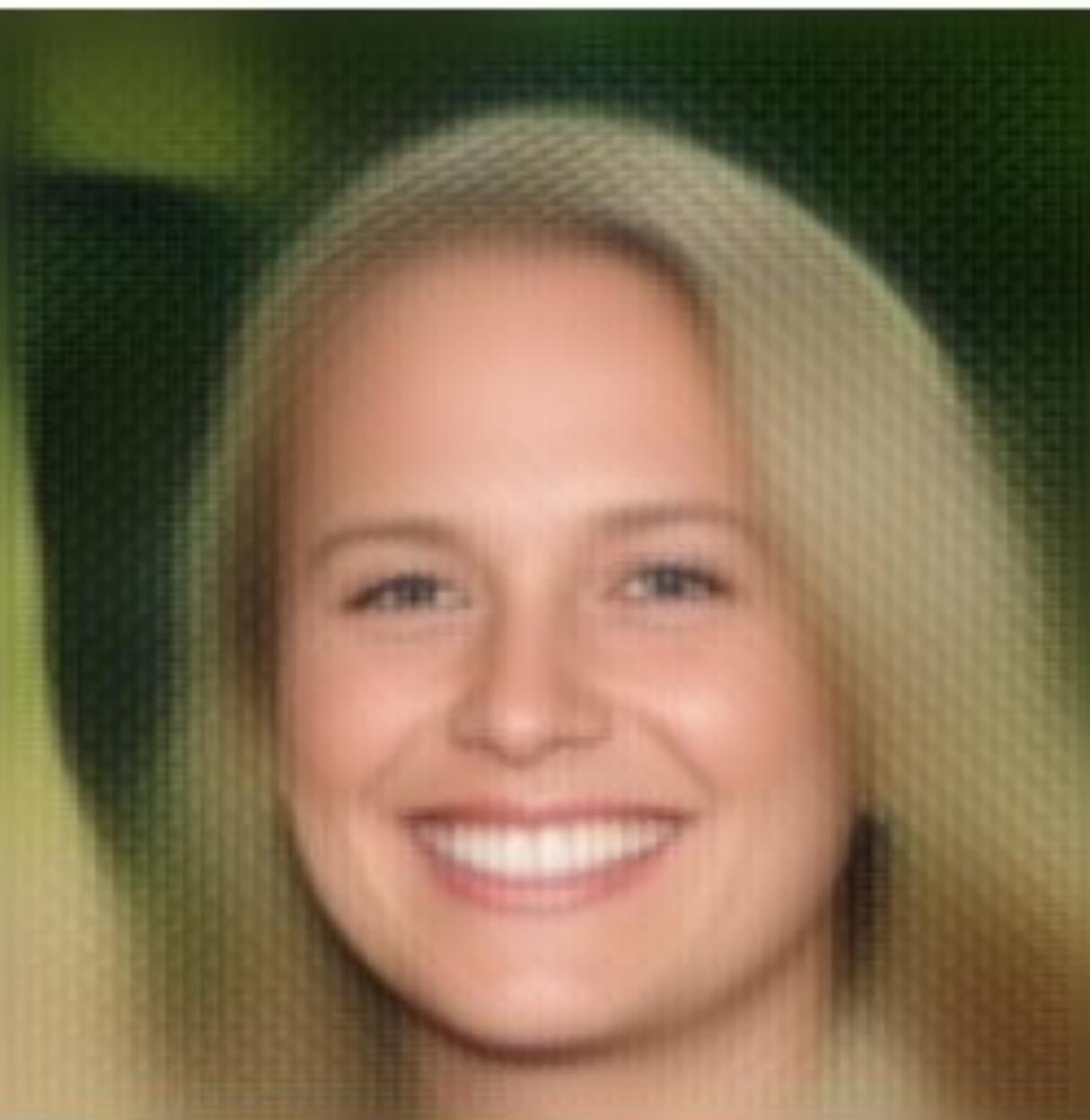
Coeff: 0.0



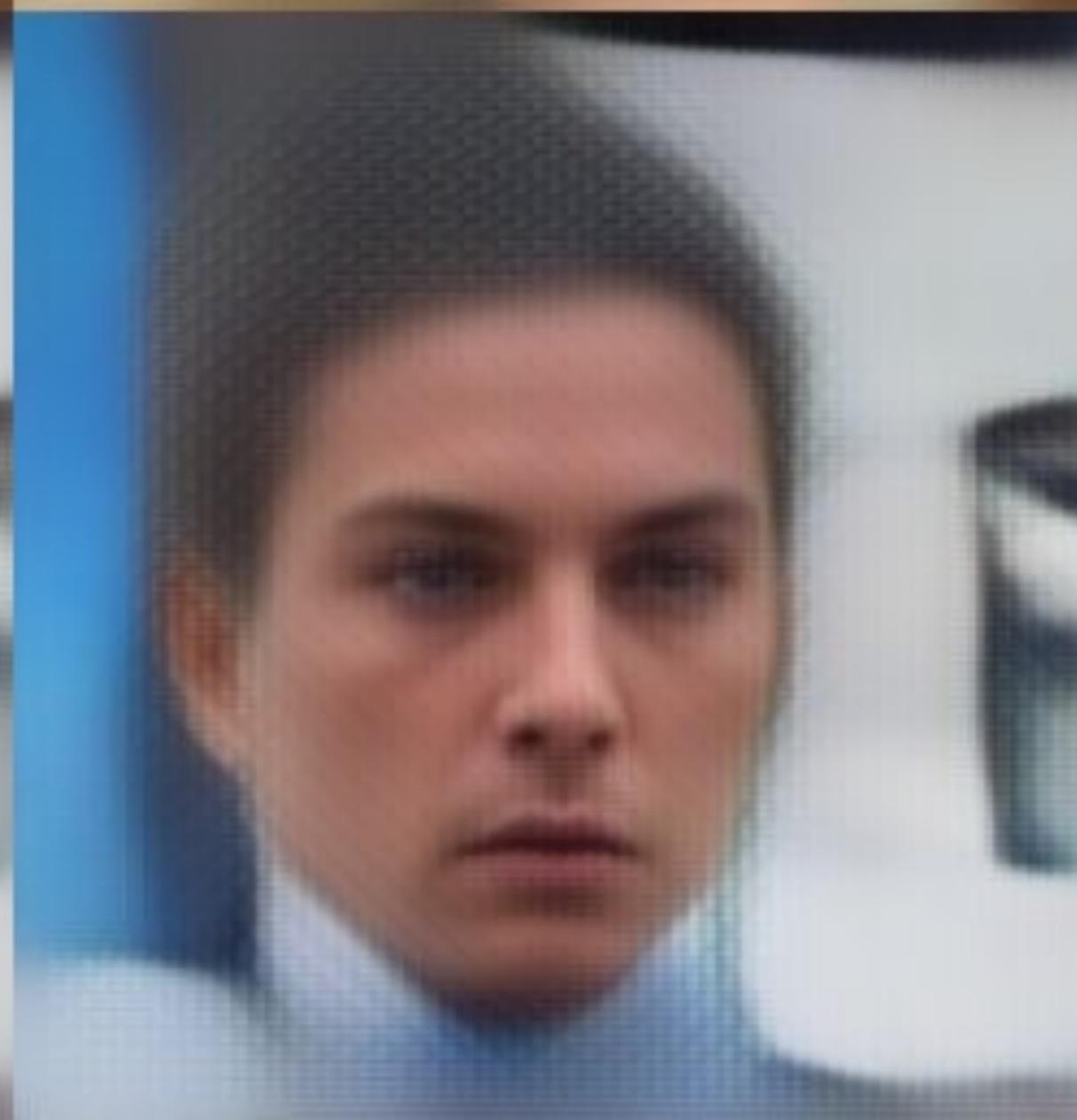
Coeff: 2.0



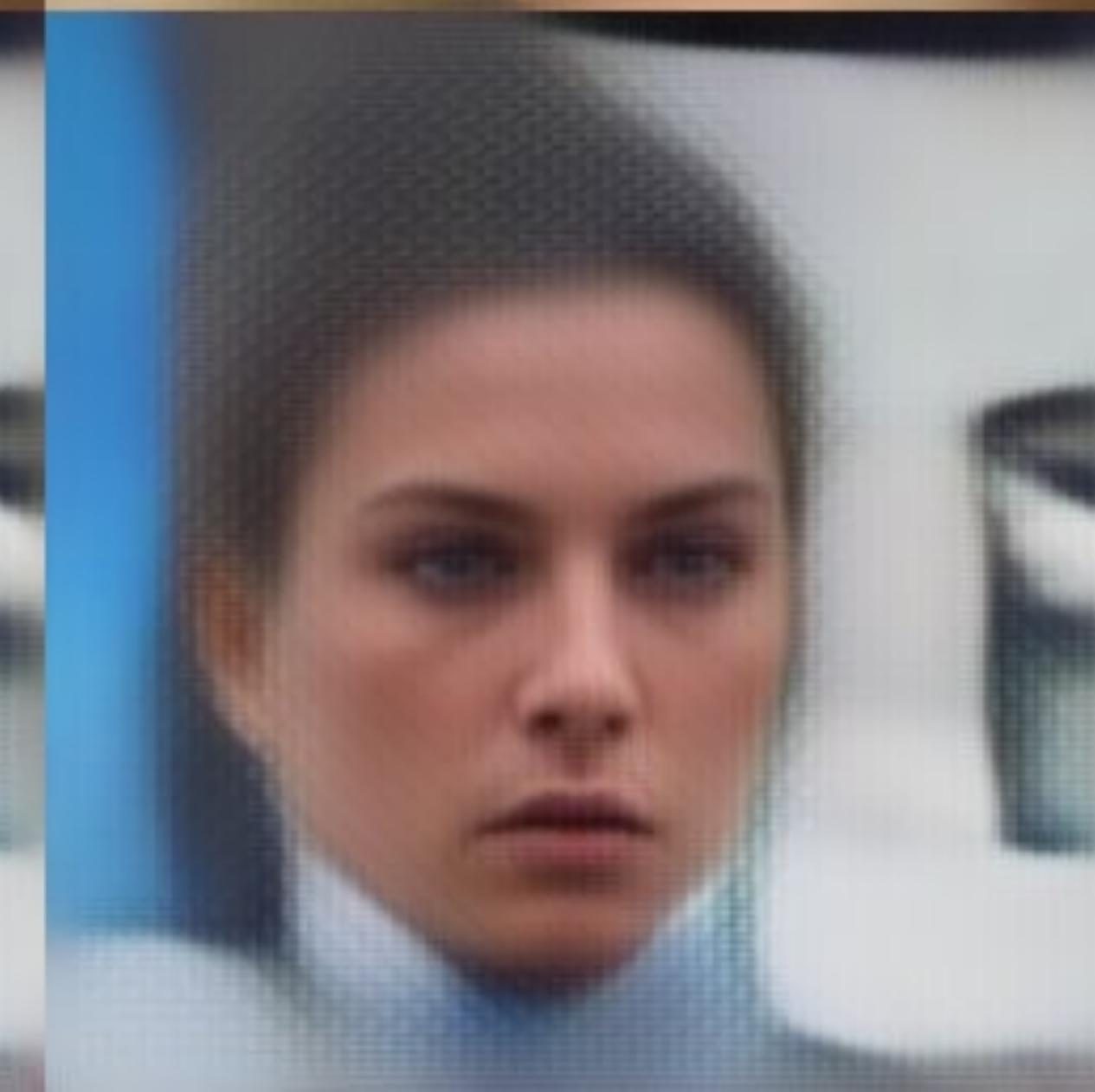
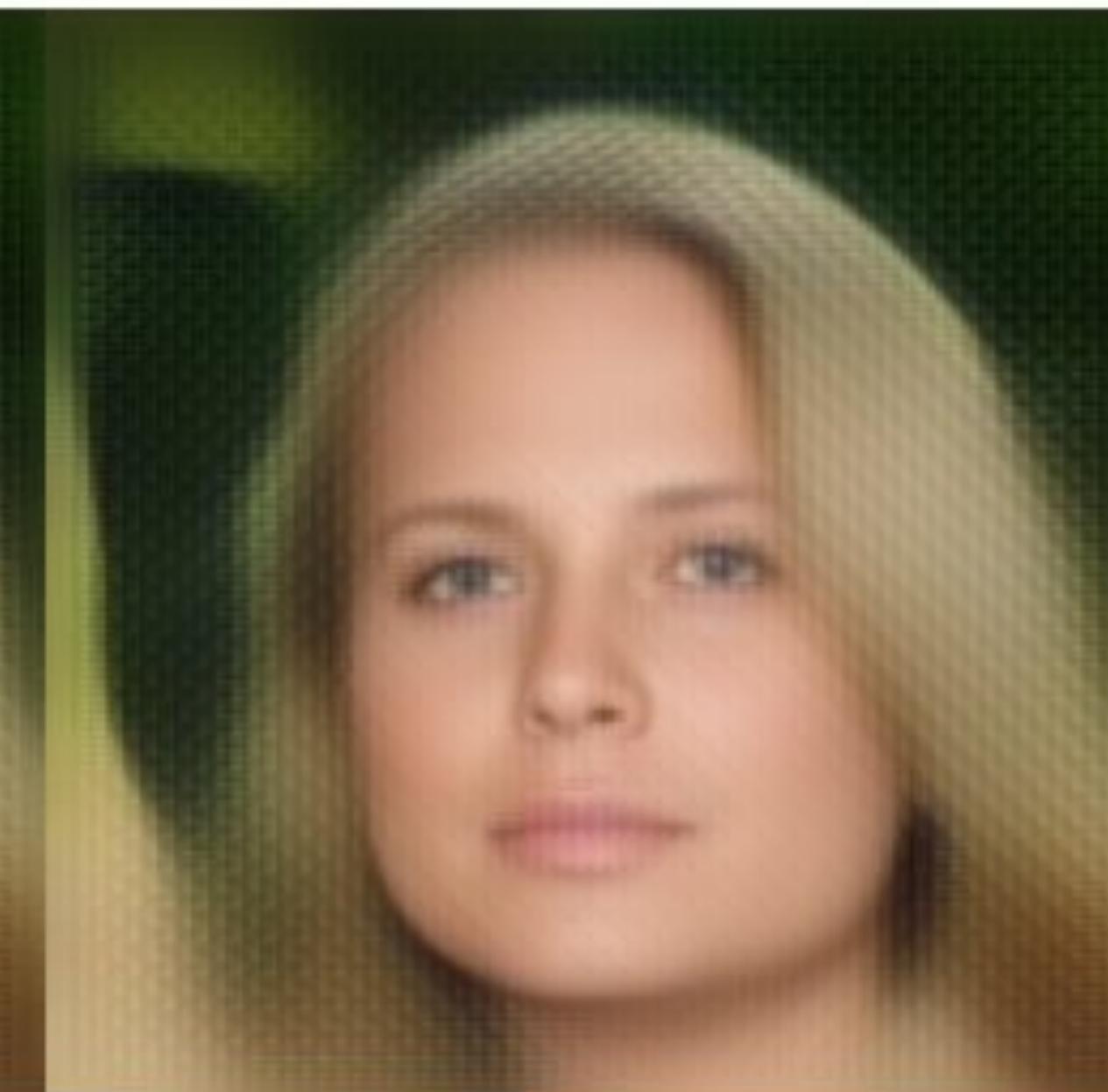
Reconstruction



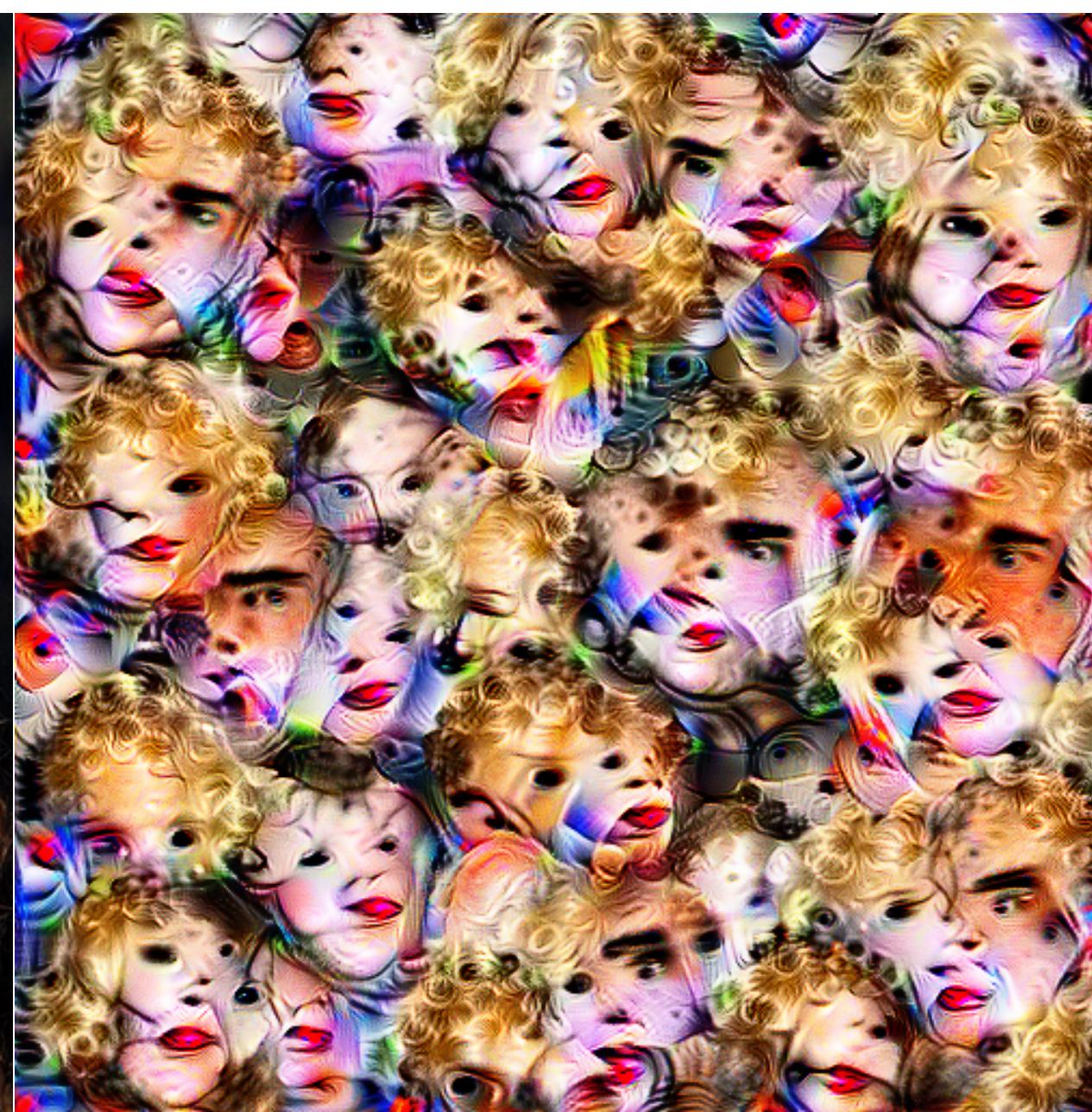
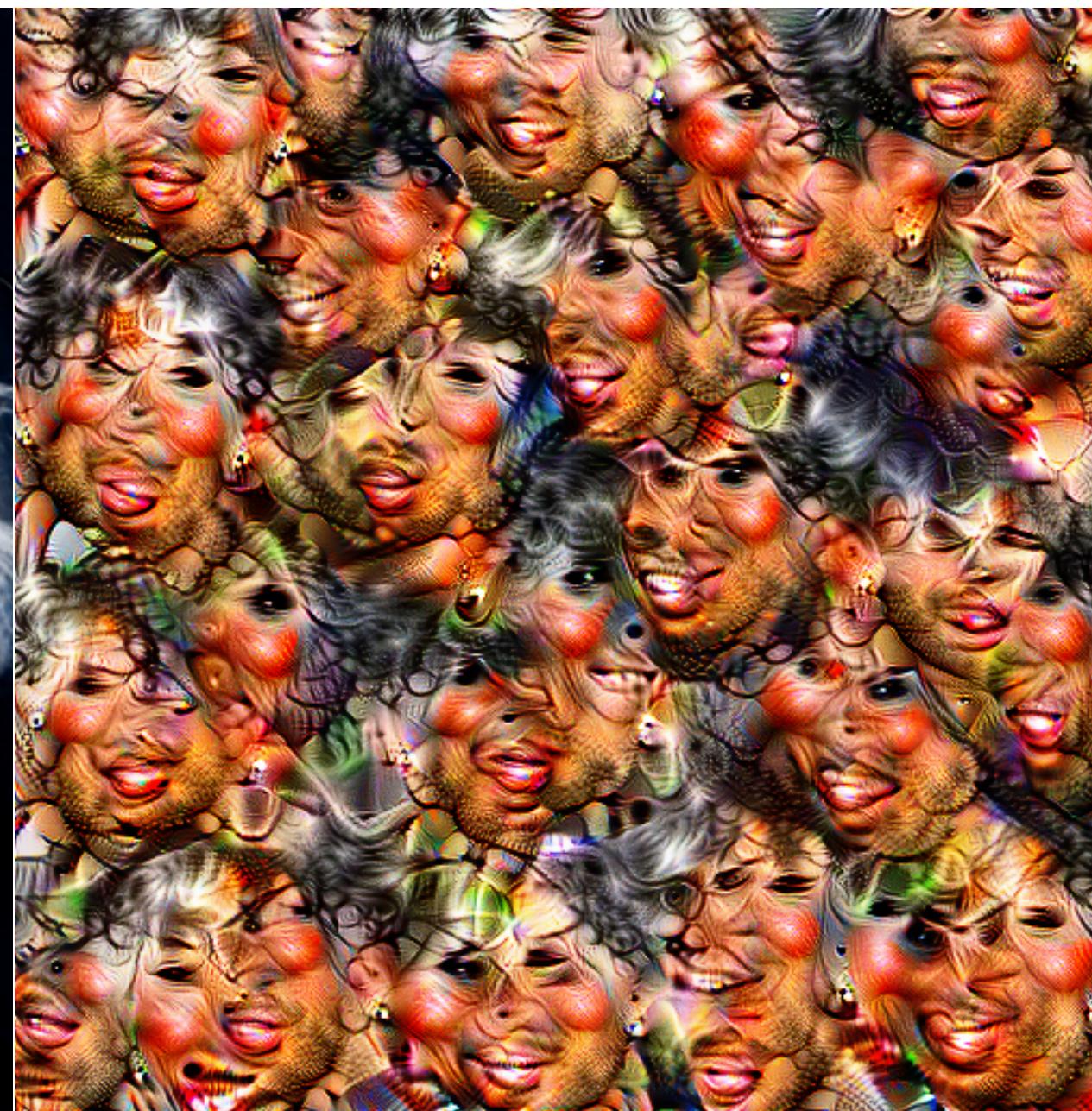
Gender-biased  
smilevector



Gender-balanced  
smilevector



# Archetípusok StyleGAN2-vel



# Bemelegítő házi feladat jövő hétre: colab hello world

<https://colab.research.google.com>

Runtime / Change runtime type / Runtime type: Python 3  
Runtime / Change runtime type / Hardware accelerator: GPU

```
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
plt.imshow(x_train[12])
```

# Köszönöm a figyelmet!

A következő héten a mesterséges  
neuronhálók fogalmilag pontos  
bevezetése, tanításuk alapelvei.