

Deep Learning

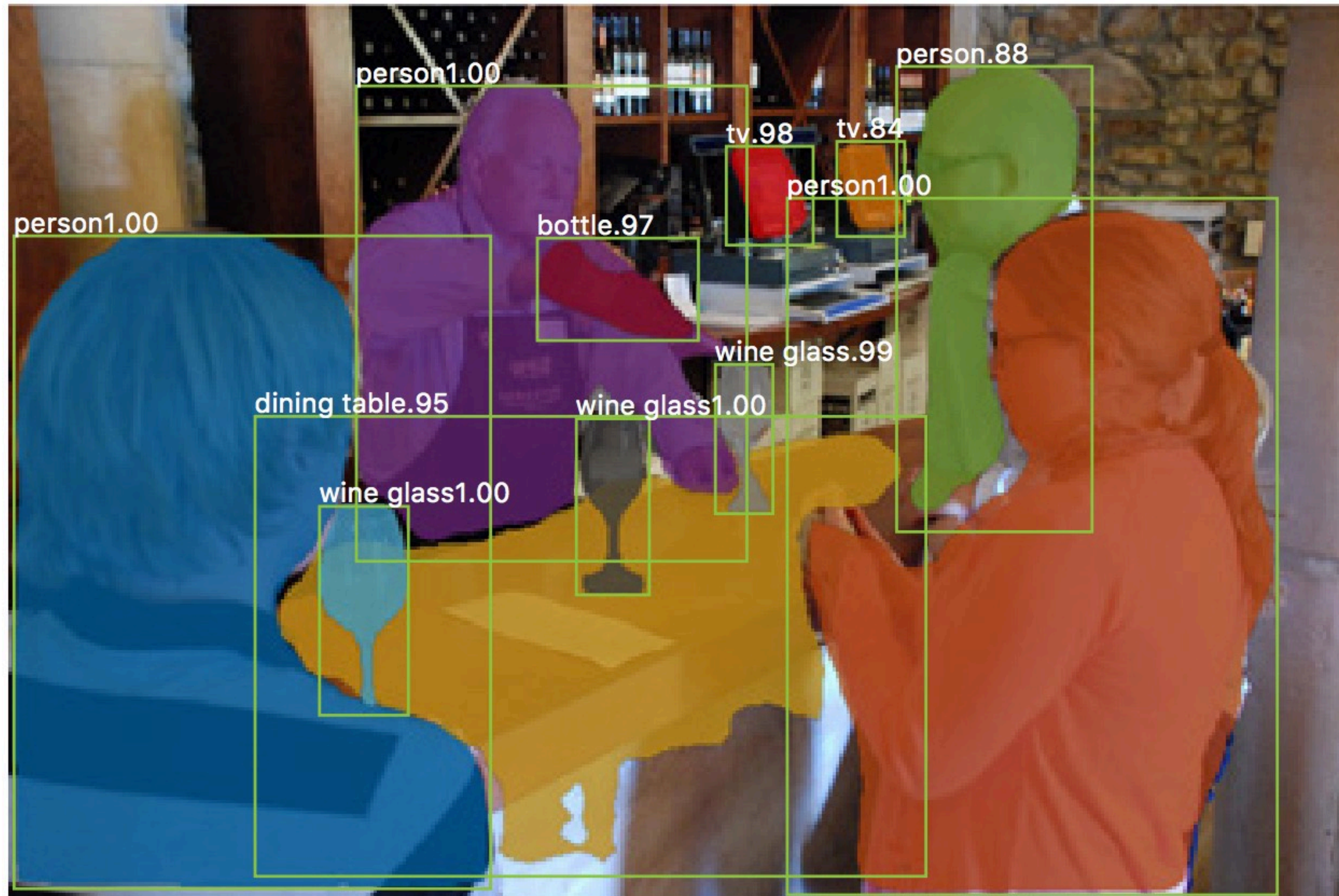
ELTE TTK Matematikai Intézet

Szeminárium

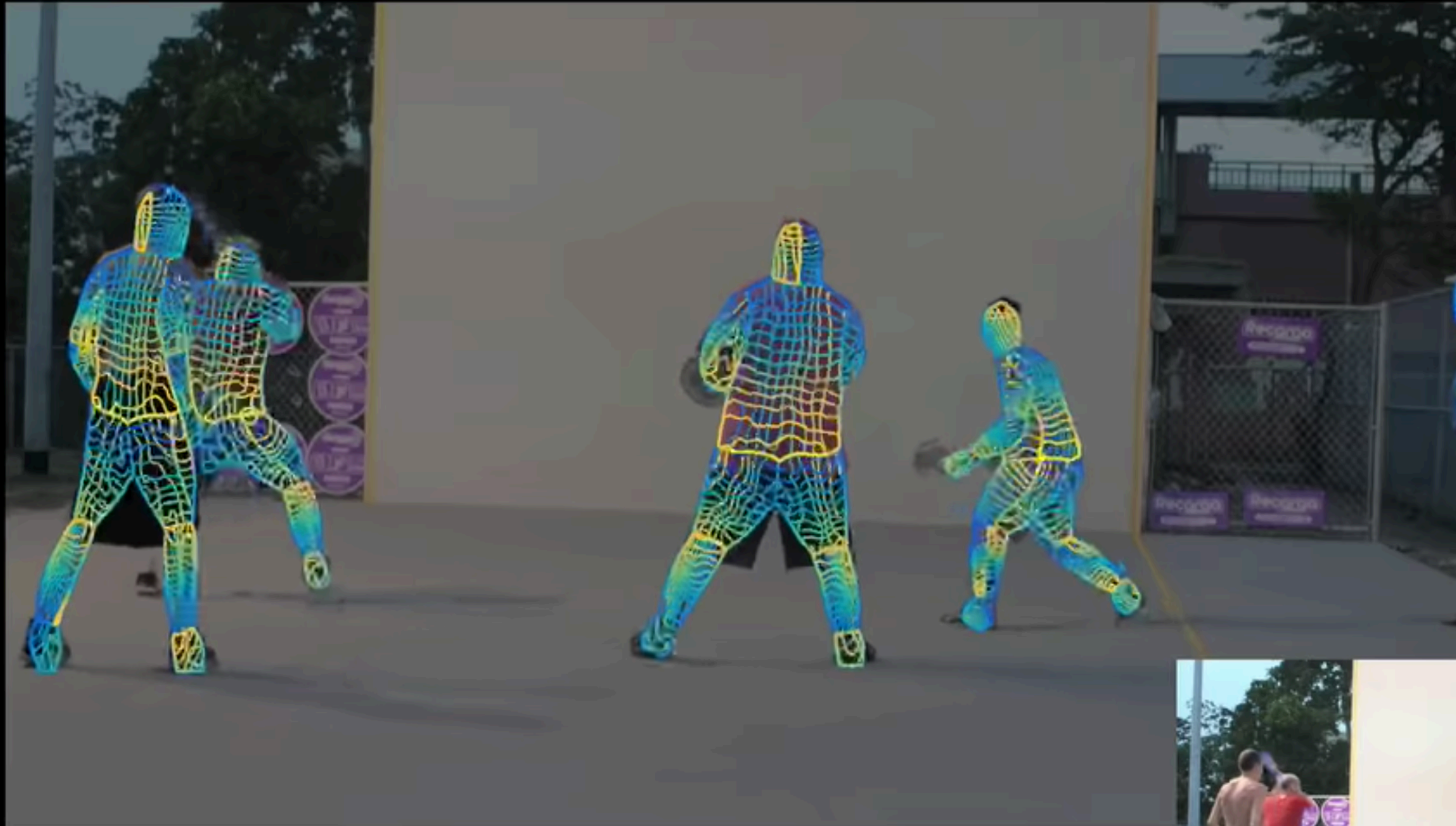
Varga Dániel (Rényi Intézet)

Példák

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

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

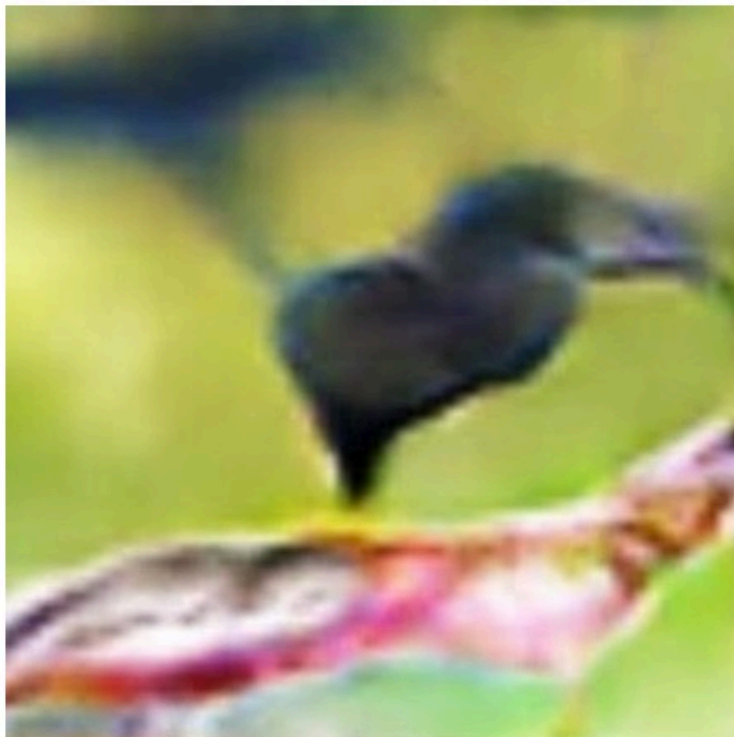
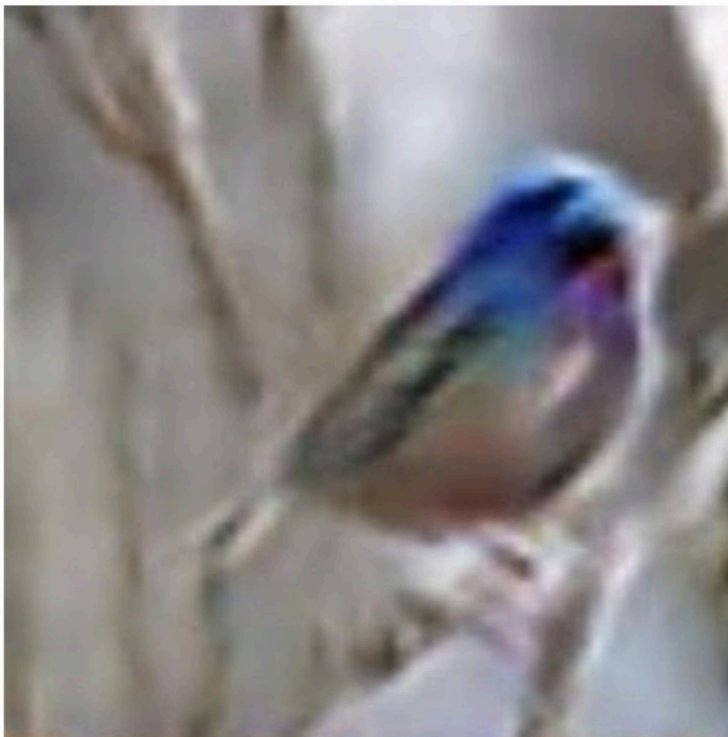
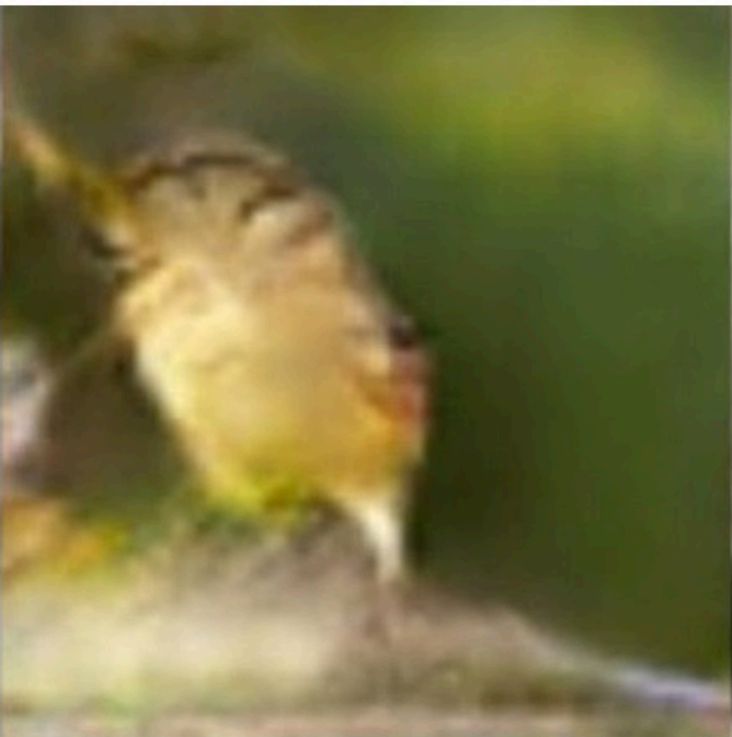
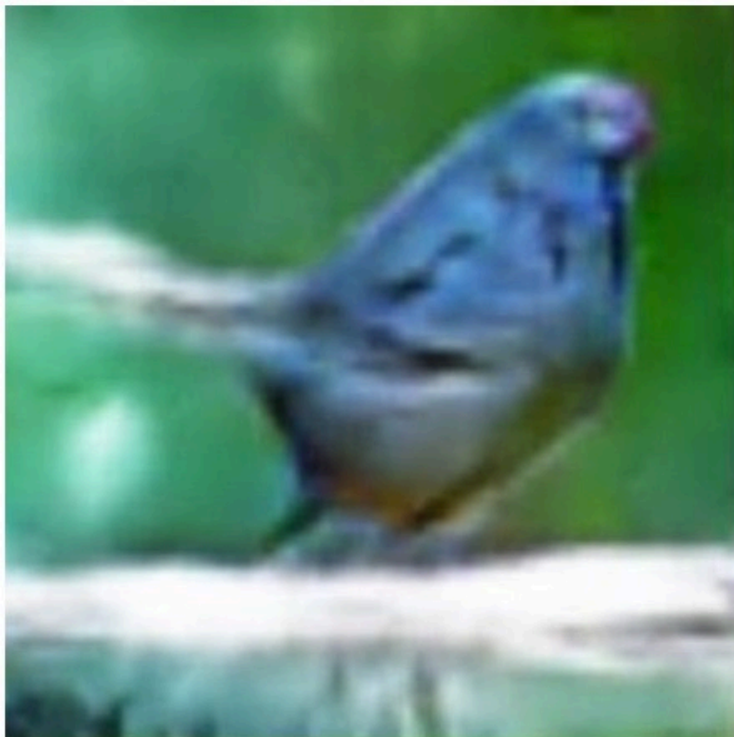
A white bird
with a black
crown and
yellow beak

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

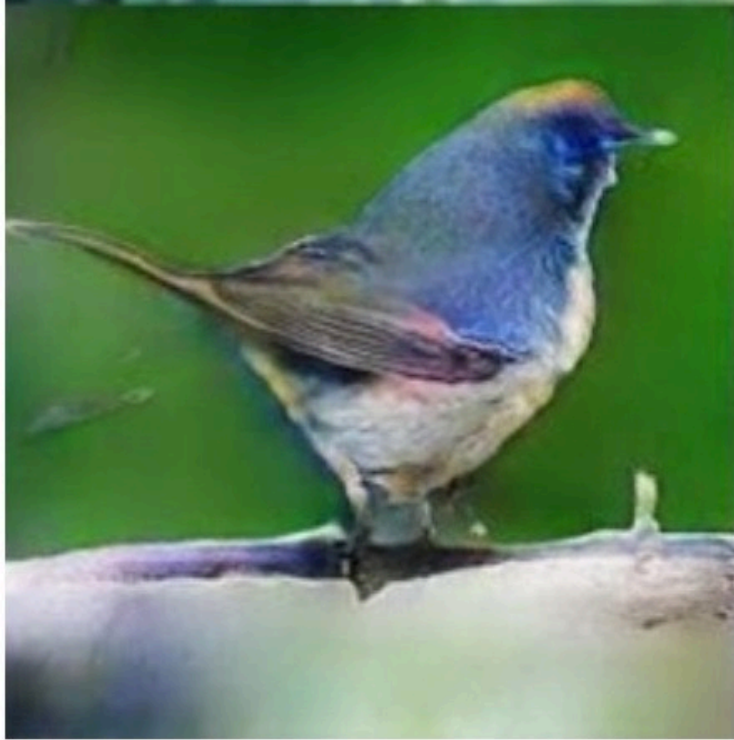
The bird has
small beak,
with reddish
brown crown
and gray belly

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

Stage-I
images

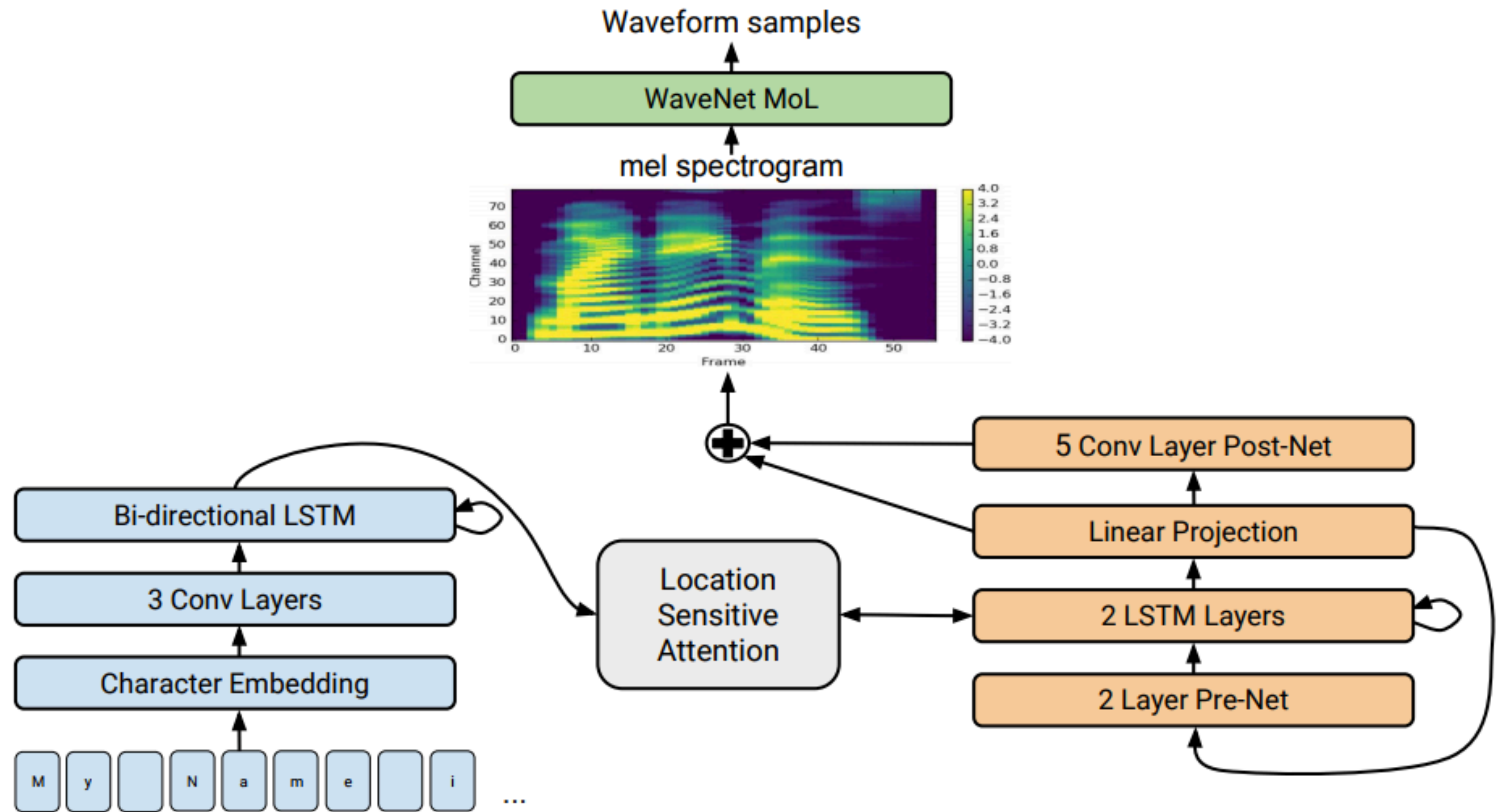


Stage-II
images



Tacotron 2 - beszédsszintézis

[link: Tacotron 2 szintézis példák](#)



Unsupervised Image-to-image Translation



Bemenetek és kimenetek reprezentációja

Convolution

Pooling

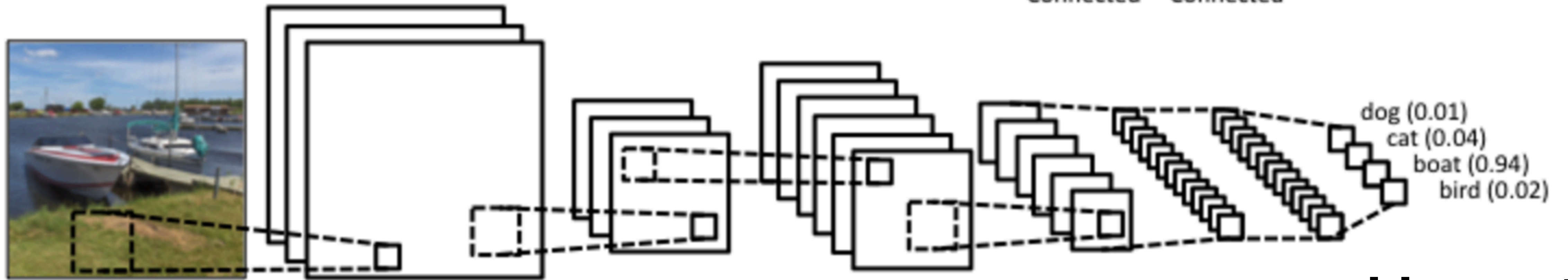
Convolution

Pooling

Fully
Connected

Fully
Connected

Output Predictions



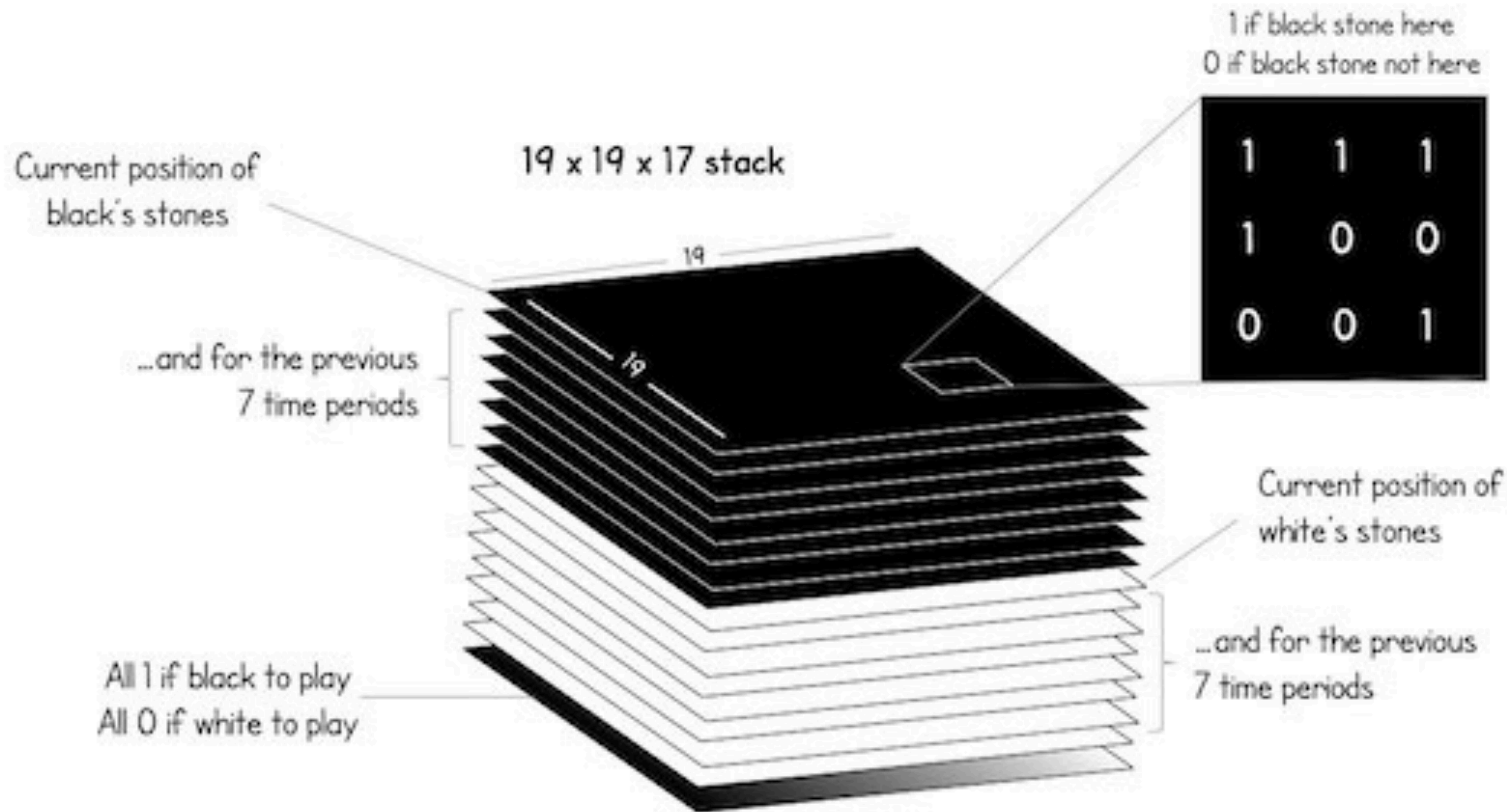
bemenet:
RGB pixel-
fényerők
(224x224x3 dim)

kimenet:
kategorikus
(4 dim)

Kimenet: pixelenkénti klasszifikáció (haj vagy nem haj)

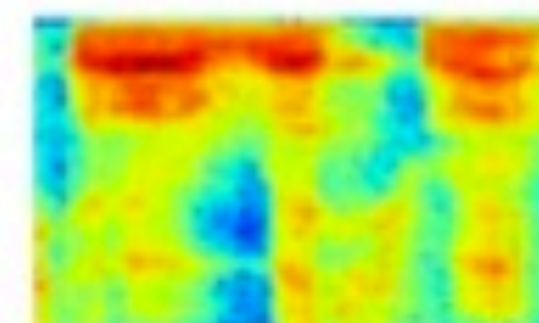
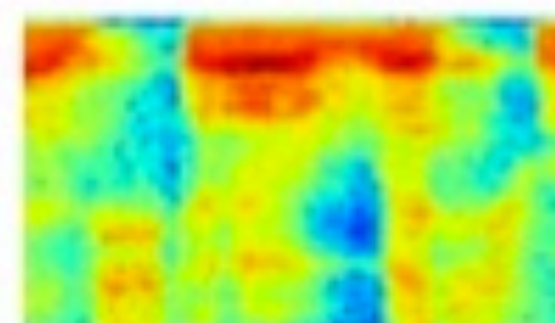
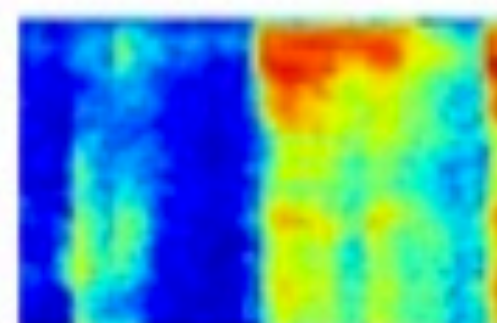
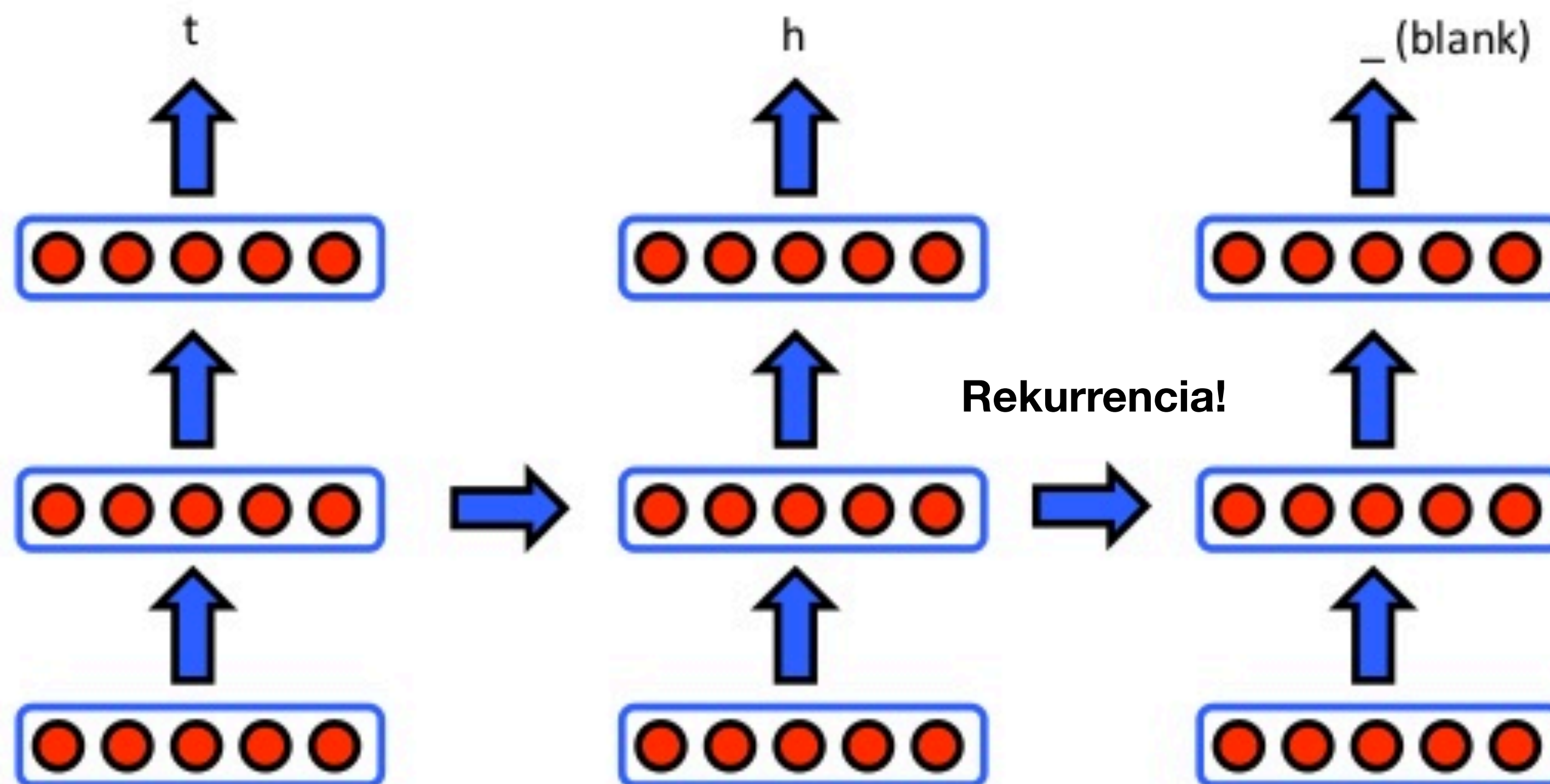


az AlphaGo Zero játékállás -reprezentációja



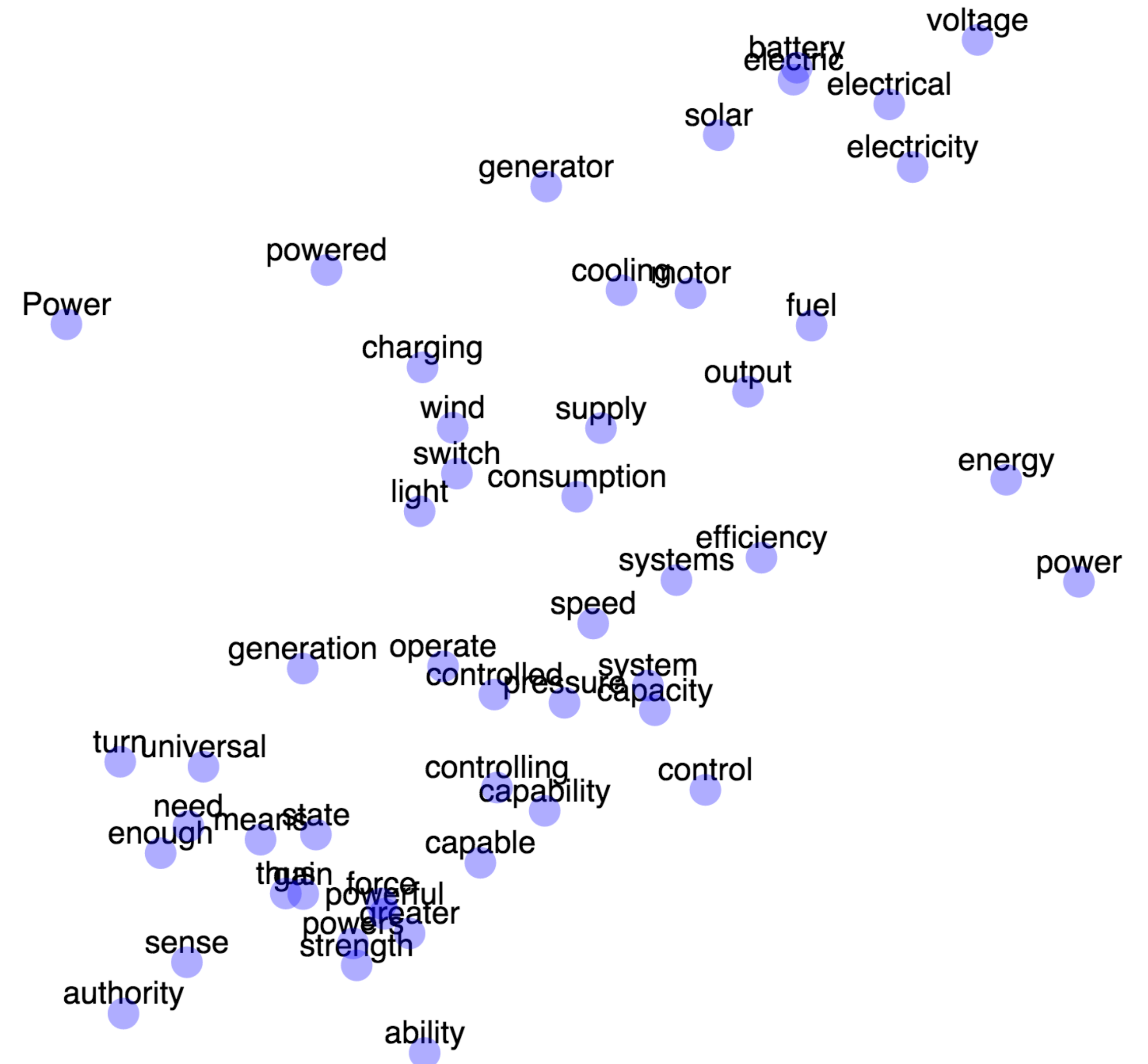
This stack is the input to the deep neural network

Output
alphabet,
space,
& blank



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.



Alapfogalmak

- $f_{\Theta} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Folytonos, majdnem mindenhol differenciálható függvények családja, amelynek egyes elemeit valós paraméterek Θ vektora jelöli ki.
- A függvény által elvárt viselkedést az $(x_1, y_1), \dots, (x_N, y_N)$ példákkal (tanítókorpusszal) adjuk meg. $x_i \in \mathbb{R}^n, y_i \in \mathbb{R}^m$.
- Továbbá az $L(\hat{y}, y)$ veszteségfüggvénnyel, amely megadja, hogy mennyire vagyunk elégedettek az $\hat{y}$ kimenettel, amikor y lenne az elvárt kimenet. (Konvenció szerint kisebb érték a jobb.)

Alapfogalmak

A fenti jelölésekkel

$$\operatorname{argmin}_{\Theta} \sum_{i=1}^N L(f_{\Theta}(x_i), y_i)$$

adja meg az optimális paraméterbeállítást. Ha egy optimalizálási folyamat végén rögzítünk egy Θ paramétervektort, amely a fenti értéket kicsivé teszi, akkor azt mondjuk, hogy a hálózatot betanítottuk a tanítókorpuszon.

Alapfogalmak

Konkrét, egyszerű, de már gyakorlatban is hasznos példa f_{Θ} függvénycsaládra:

$$f(x) = (A_k \circ \sigma \circ A_{k-1} \circ \sigma \circ \dots \circ A_1)(x),$$

ahol $A_j(z) = W_j z + b_j$ affin transzformáció, és

$$\sigma(z) = (\max(z_1, 0), \max(z_2, 0), \dots, \max(z_n, 0))$$

elemenkénti nemnegatívba csonkolás. Ebben az esetben tehát a paraméterezés:

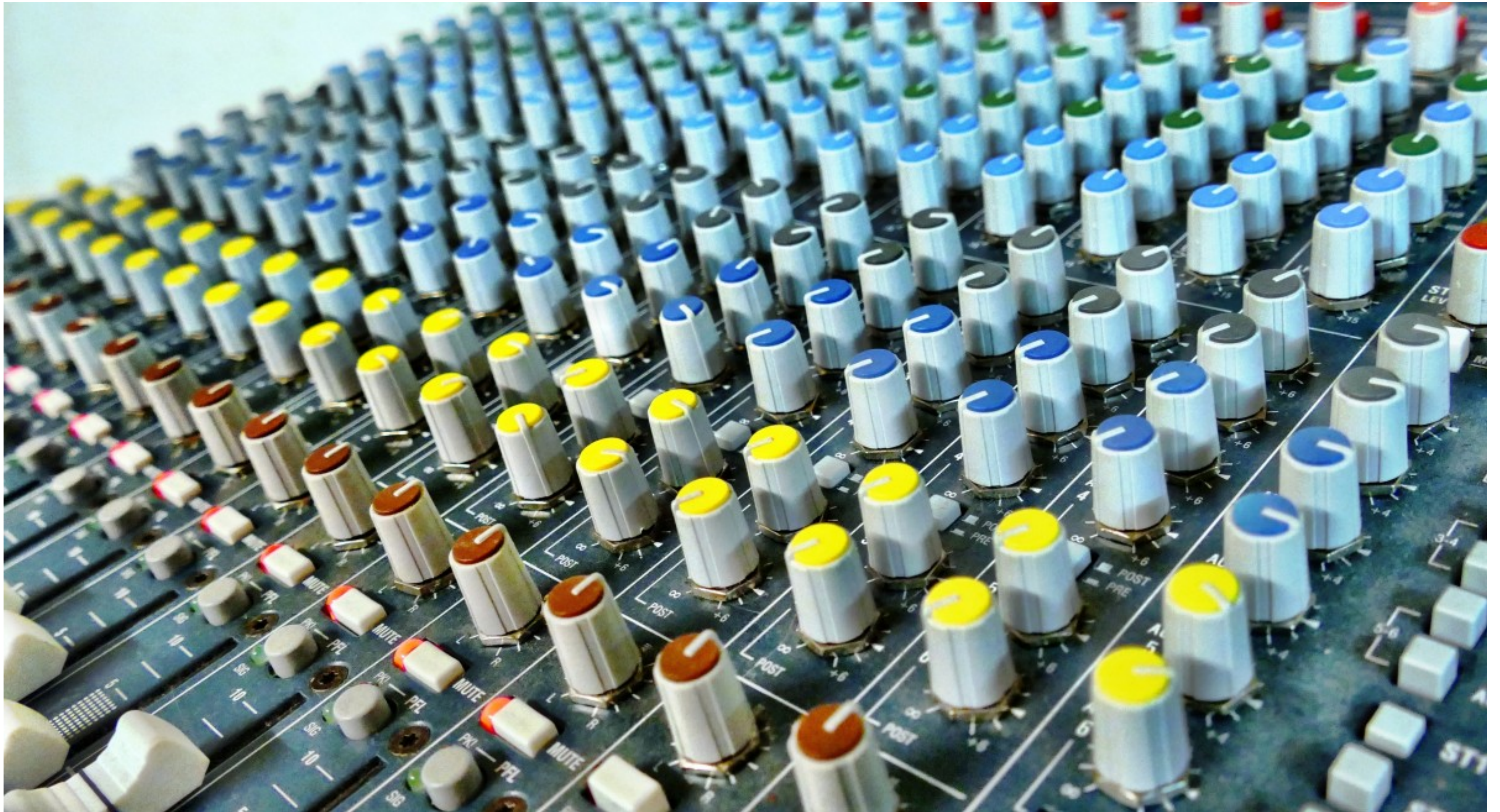
$$\Theta = (W_1, b_1, W_2, b_2, \dots, W_k, b_k)$$

Nemkonvex célfüggvény

$$\operatorname{argmin}_{\Theta} \sum_{i=1}^N L(f_{\Theta}(x_i), y_i)$$

- Ez egy nemkonvex optimalizálási feladat, amit gradiens alapú lokális optimalizáló módszerekkel próbálunk megoldani.
- Ehhez kihasználjuk, hogy a veszteségfüggvény paraméterek szerinti gradiense hatékonyan kiszámítható (backpropagation).
- Fog ez sikerülni?

Egy fekete doboz nagyon sok potméterrel



A második AI tél

- ...fog ez sikerülni?
- Amikor a 90-es években nehezebb feladatokon valóban nem sikerült, akkor logikusan a nemkonvexitásban keresték a problémát, és ez a kutatási irány kiment a divatból.
- Az egyszerűségükhöz képest meglepően eredményes konvex módszerek (logisztikus regresszió, support vector machine) vették át a helyét, lényegében speciális egyrétegű neuronhálók.
- De ma már kis túlzással azt mondhatjuk, hogy a lokális optimumok ártatlanul lettek gyanúba keverve.
- Kis túlzással. A valóság árnyaltabb, és a neuronháló optimumhelyek topológiája és geometriája izgalmas, alig feltárt kutatási terület.

A második AI tél vége

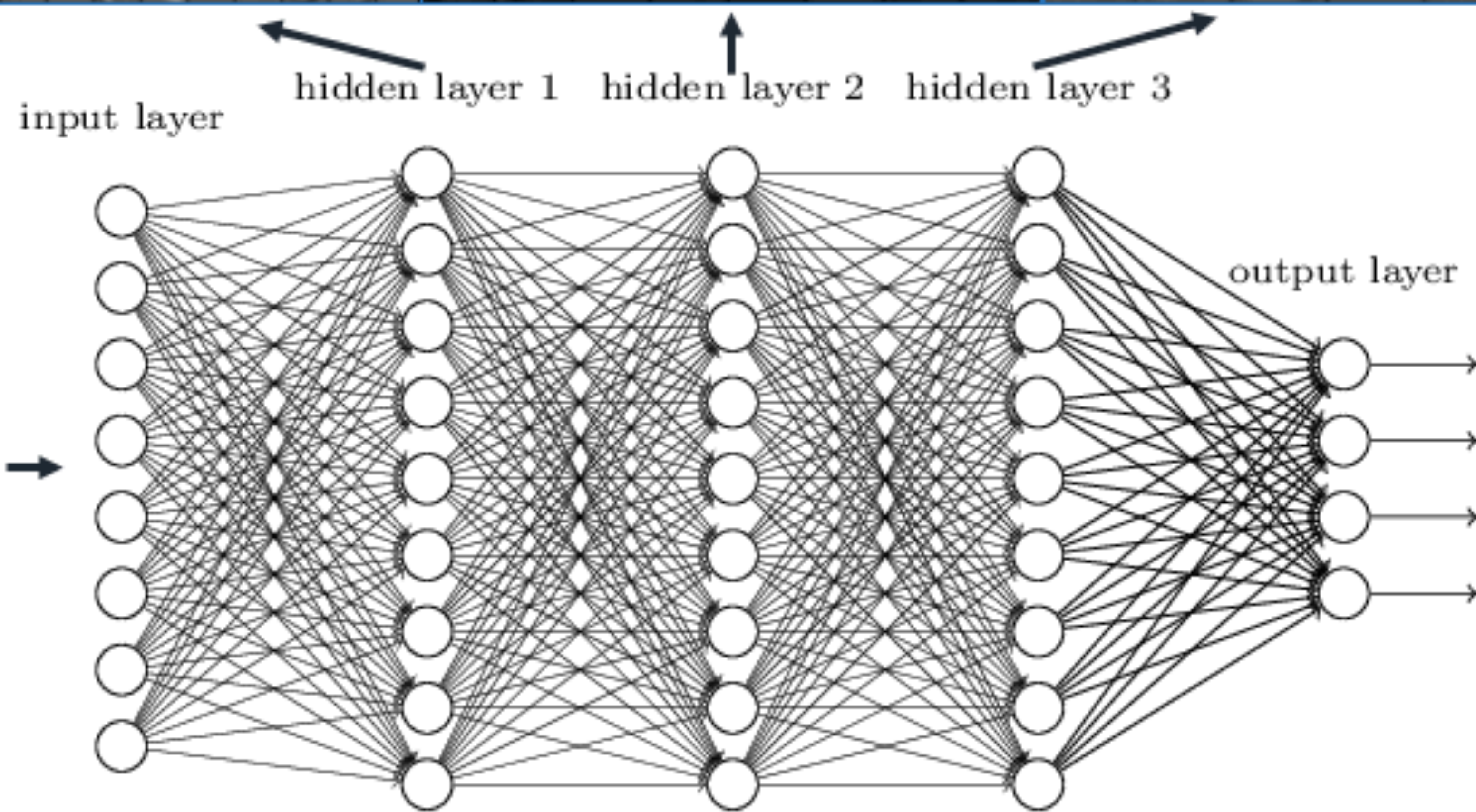
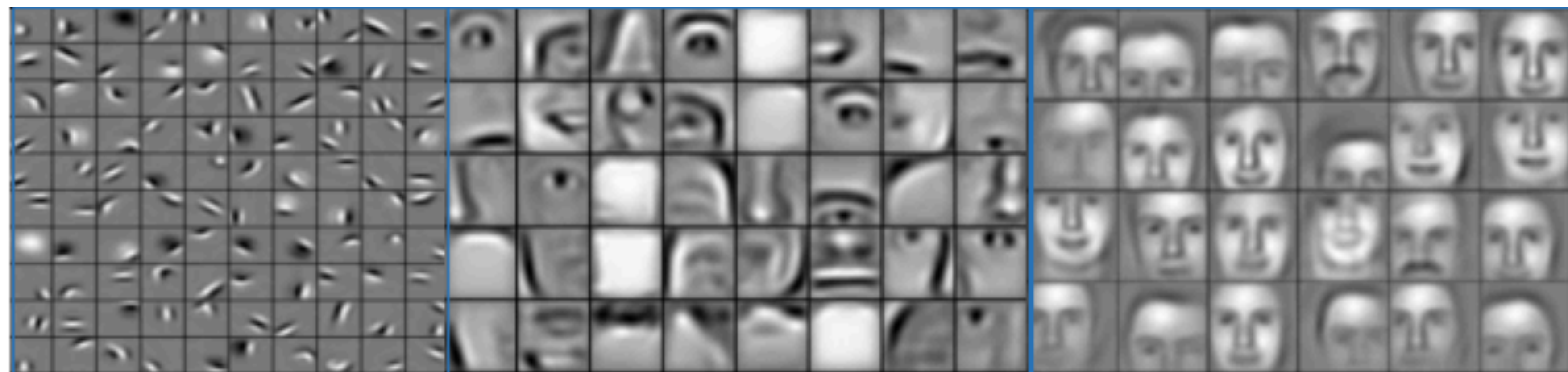
- Ha gondatlanul választjuk meg a kiindulási Θ paraméterezést, akkor rögtön az elején elakadhat az optimalizálás.
- A 90-es években használt szigmoid nemlinearitás nem bizonyult olyan sikeresnek, mint a fenti példában is szereplő modernebb ReLU azaz $\max(0, x)$. (eltűnő gradiensek)
- A számítási kapacitásnak és az elérhető tanító-korpuszok méretének növekedése is fontos segítség volt.

Konvolúciós neuronhálók

- Ha a paraméterezett függvénycsaládunk (más szóval neuronháló architektúránk) nem használja ki a bemenet (kép, hang, idősor) eltolásinvarianciáit, akkor a deep learning módszerek általában nem működnek jól.
- Ezt a fontos szempontot idő hiányában ma szőnyeg alá fogom söpörni.
- Az illusztrációként említett neuronhálók mind kihasználják az eltolásinvarianciát.

Látens reprezentációk

Deep neural networks learn hierarchical feature representations

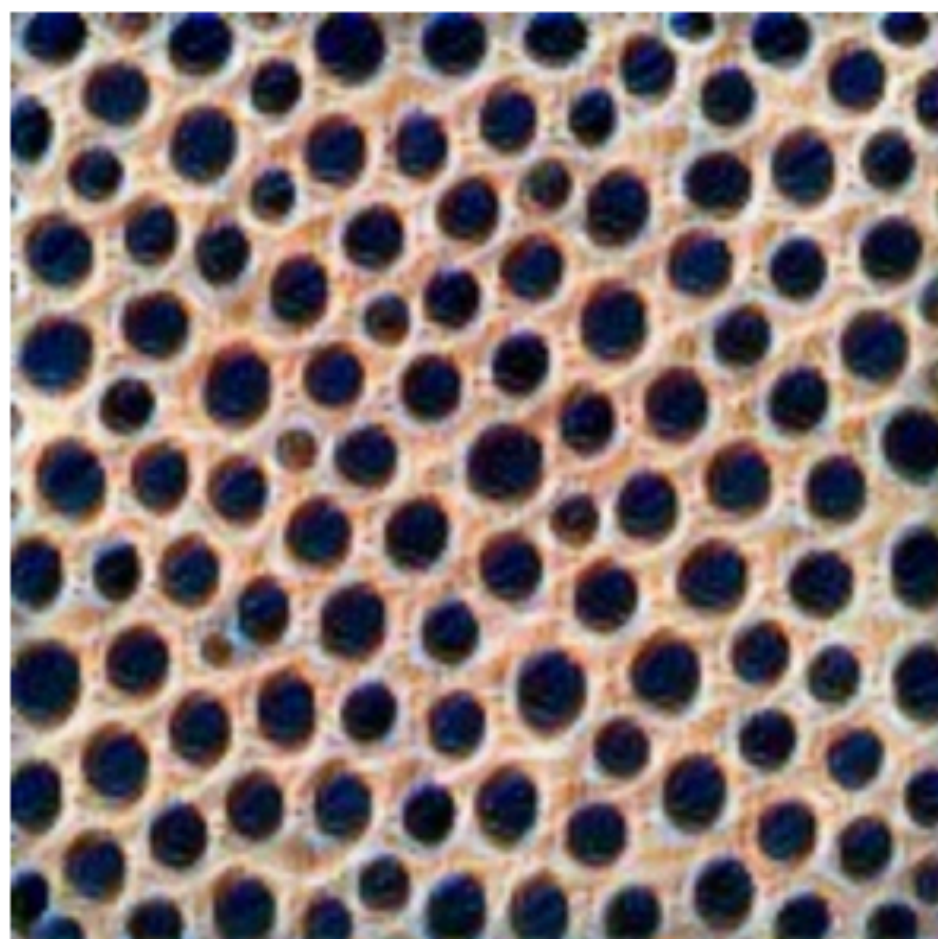


<https://distill.pub/2017/feature-visualization/>

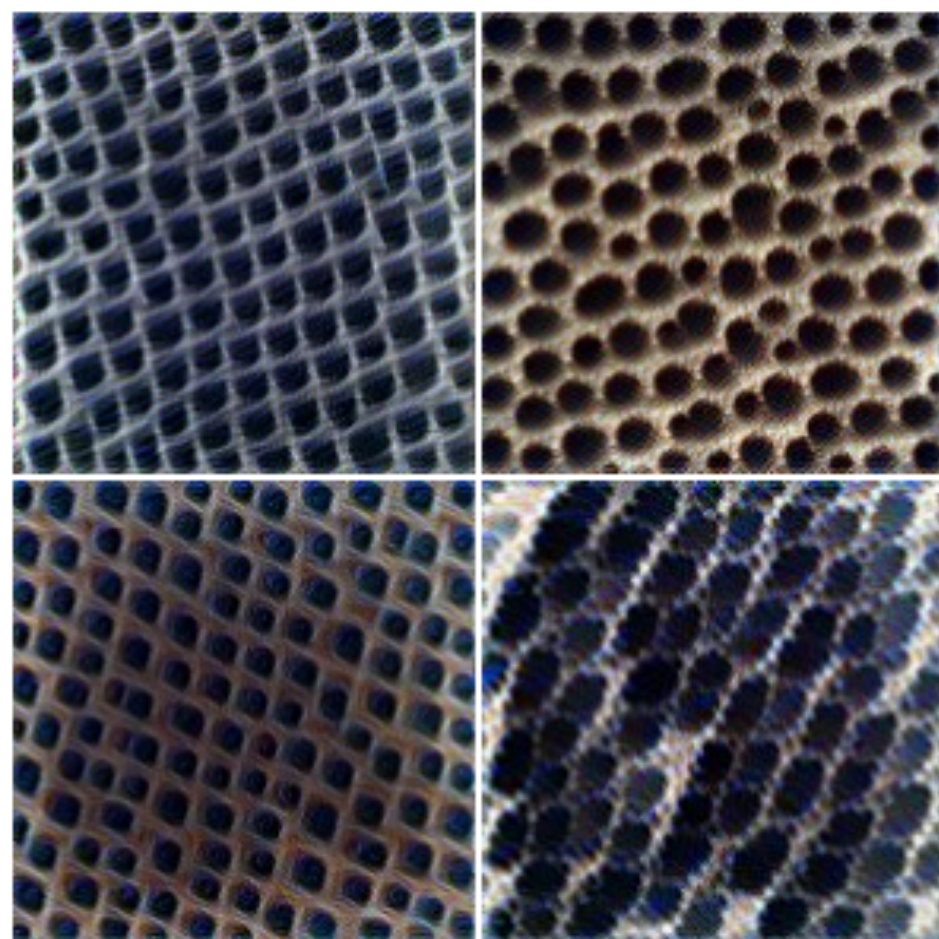
<https://distill.pub/2017/feature-visualization/appendix/>

<https://distill.pub/2018/building-blocks/>

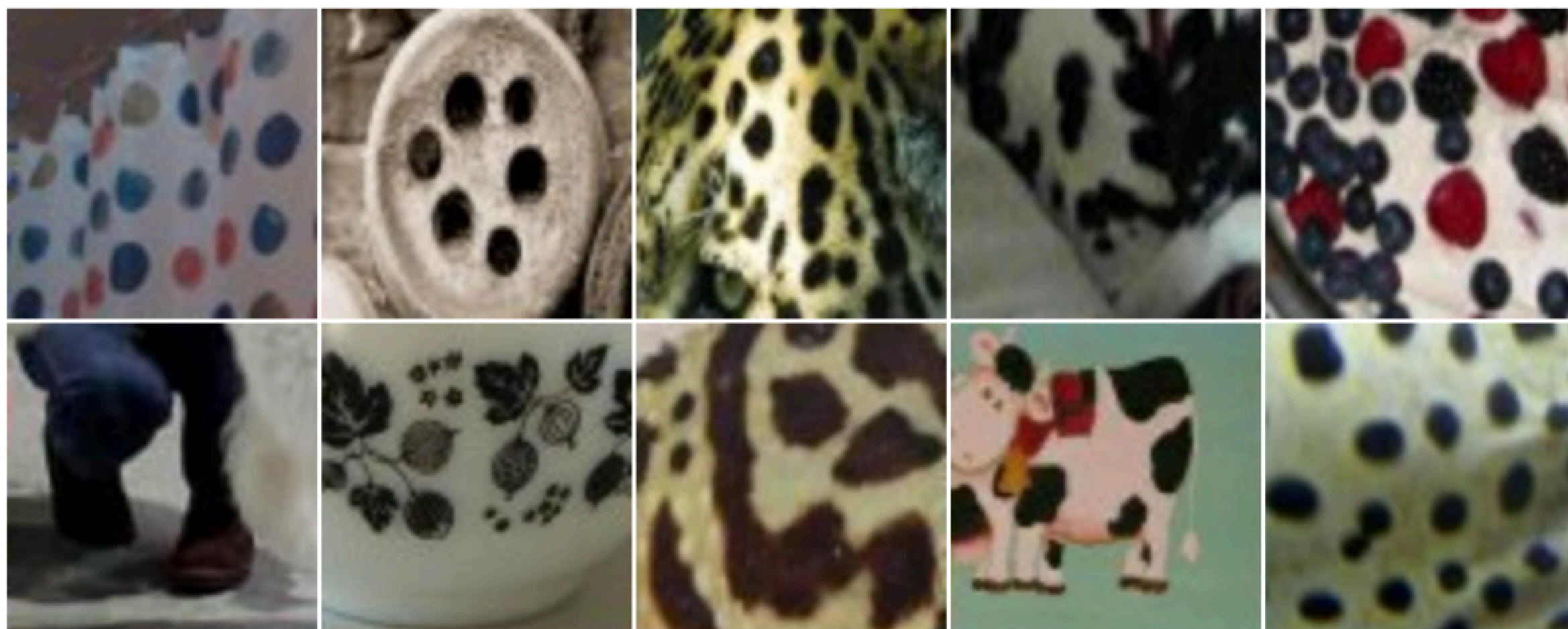
POSITIVE CHANNEL



Channel Objective

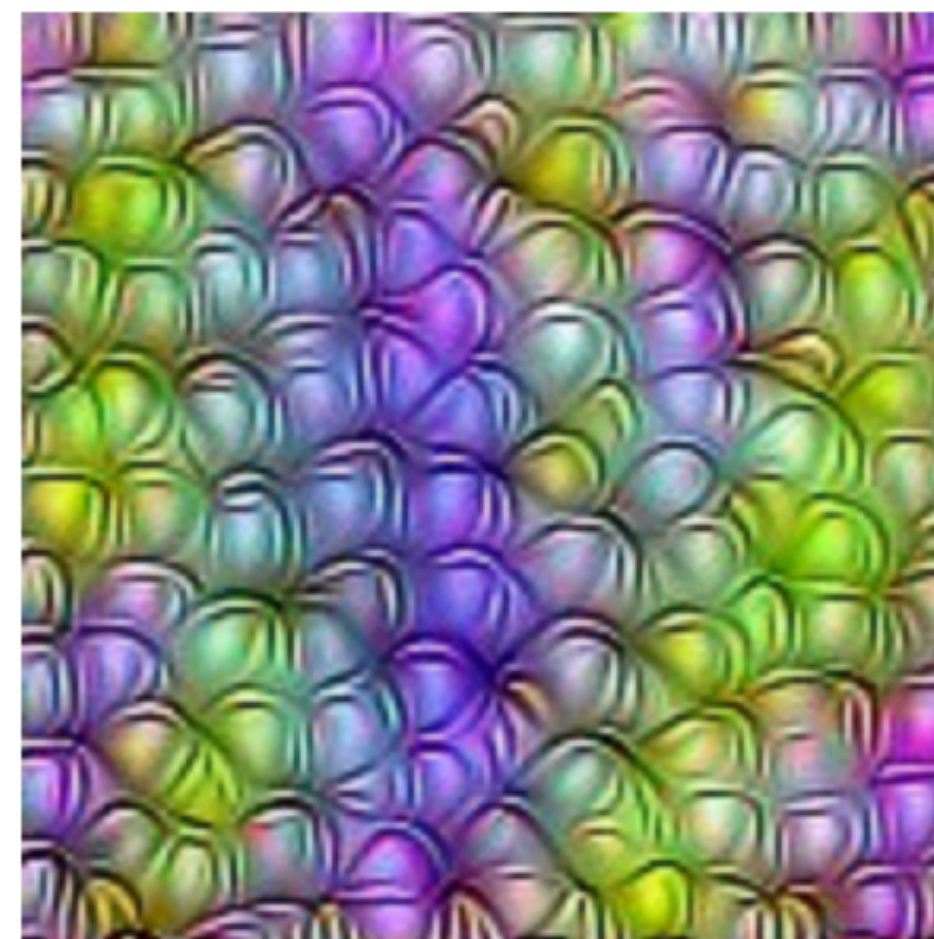


Diversity

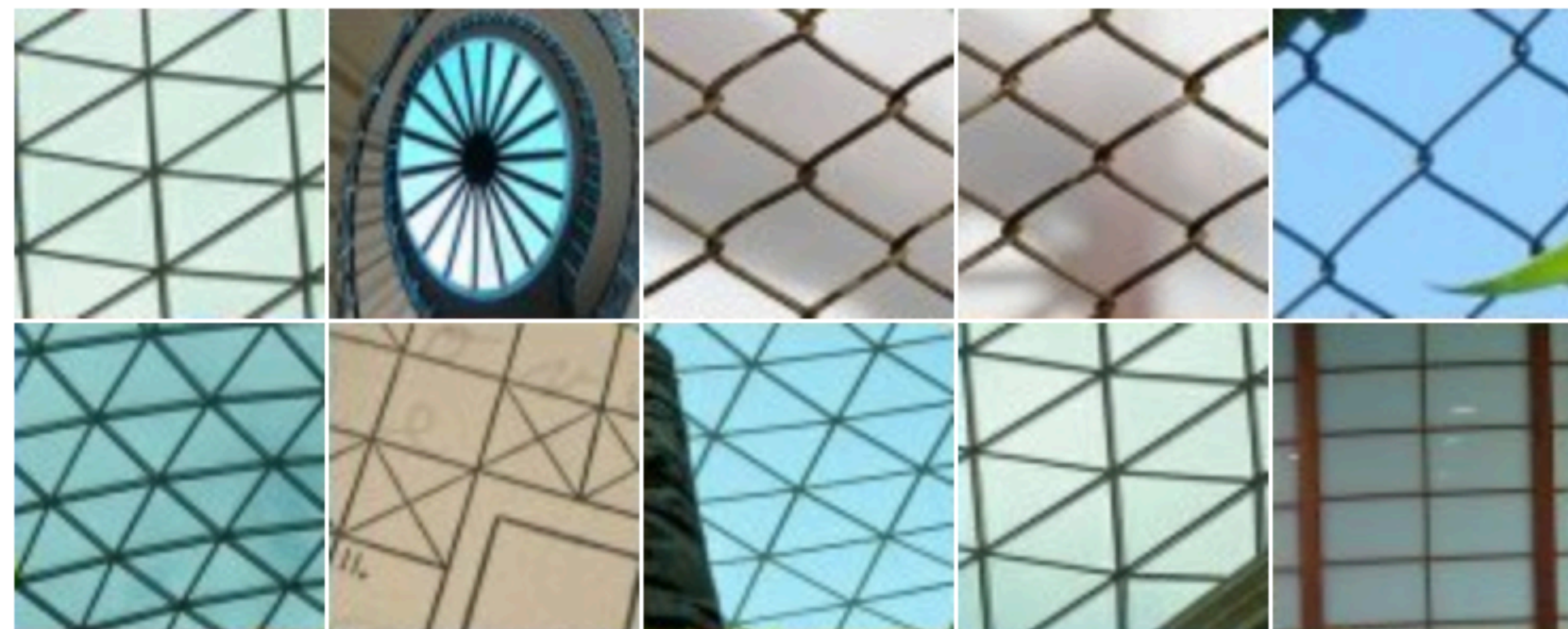


Dataset examples

NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

POSITIVE CHANNEL



Channel Objective



Diversity

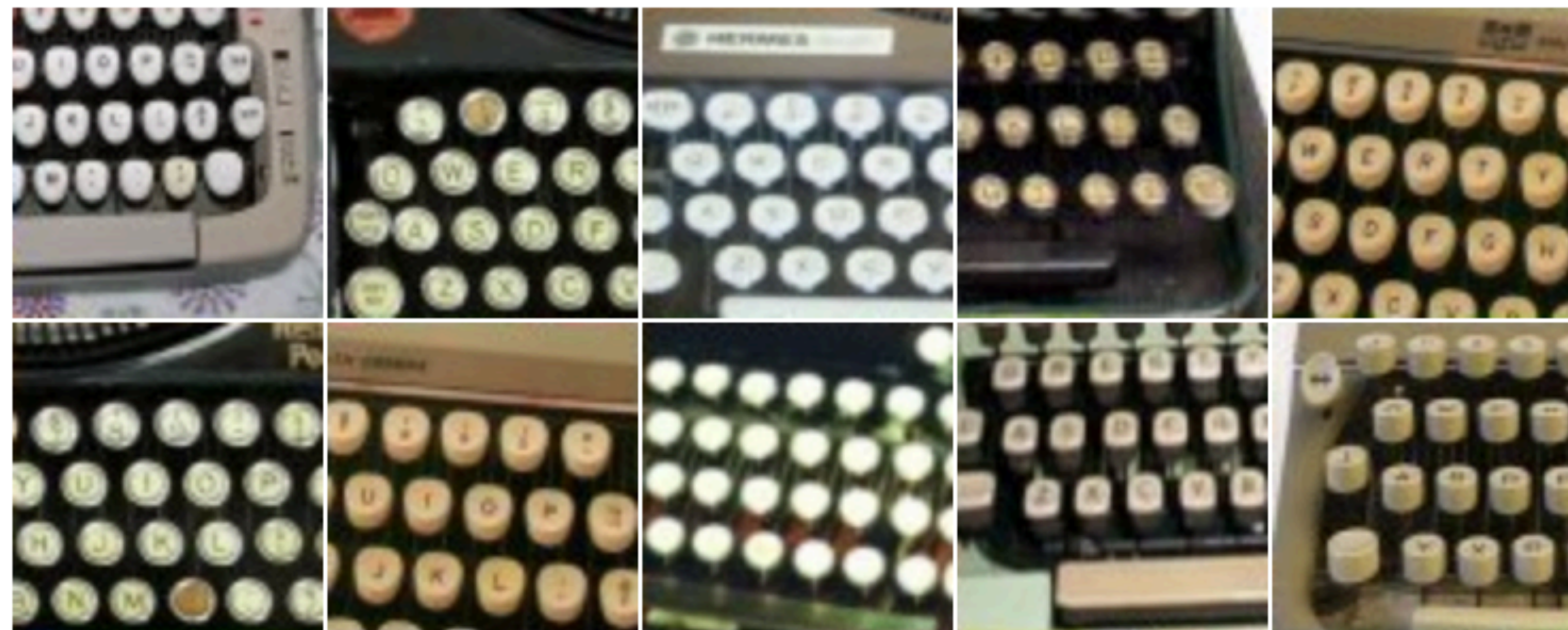


Dataset examples

NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

POSITIVE CHANNEL



Channel Objective



Diversity



Dataset examples

NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

POSITIVE CHANNEL



Channel Objective



Diversity

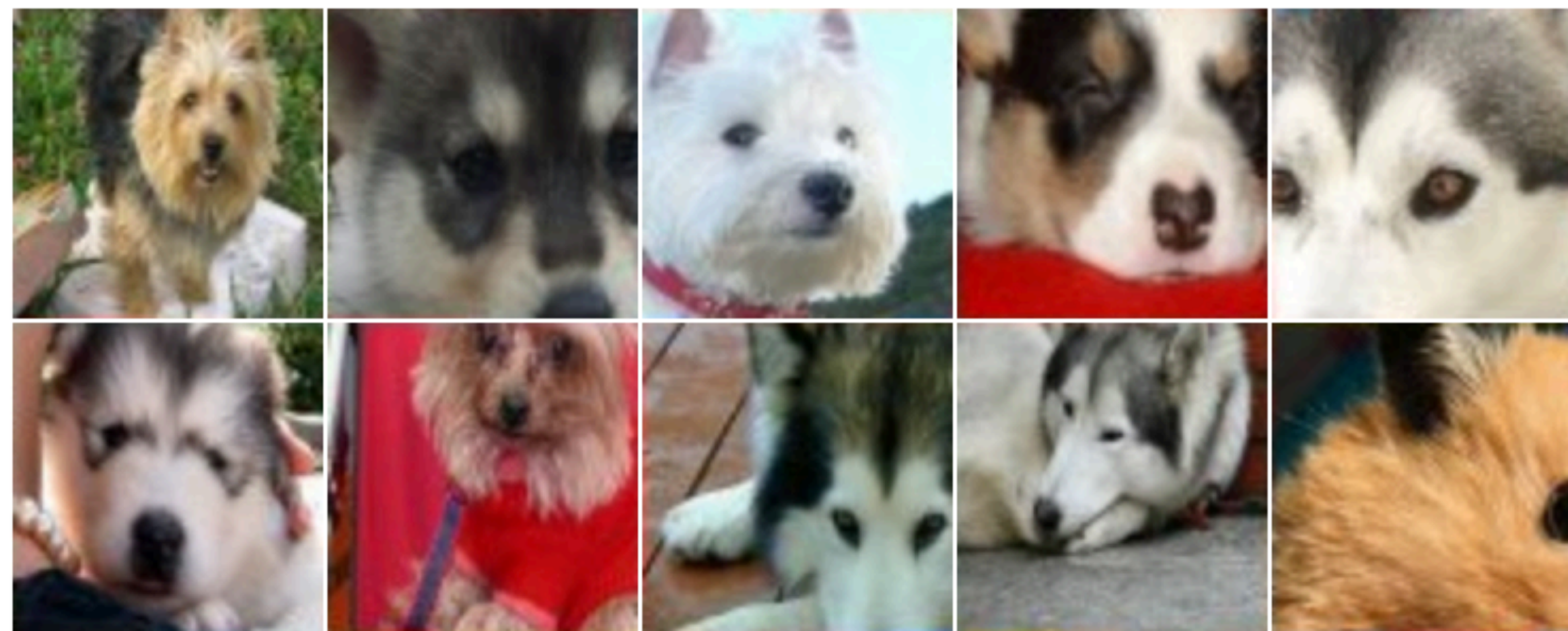


Dataset examples

NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

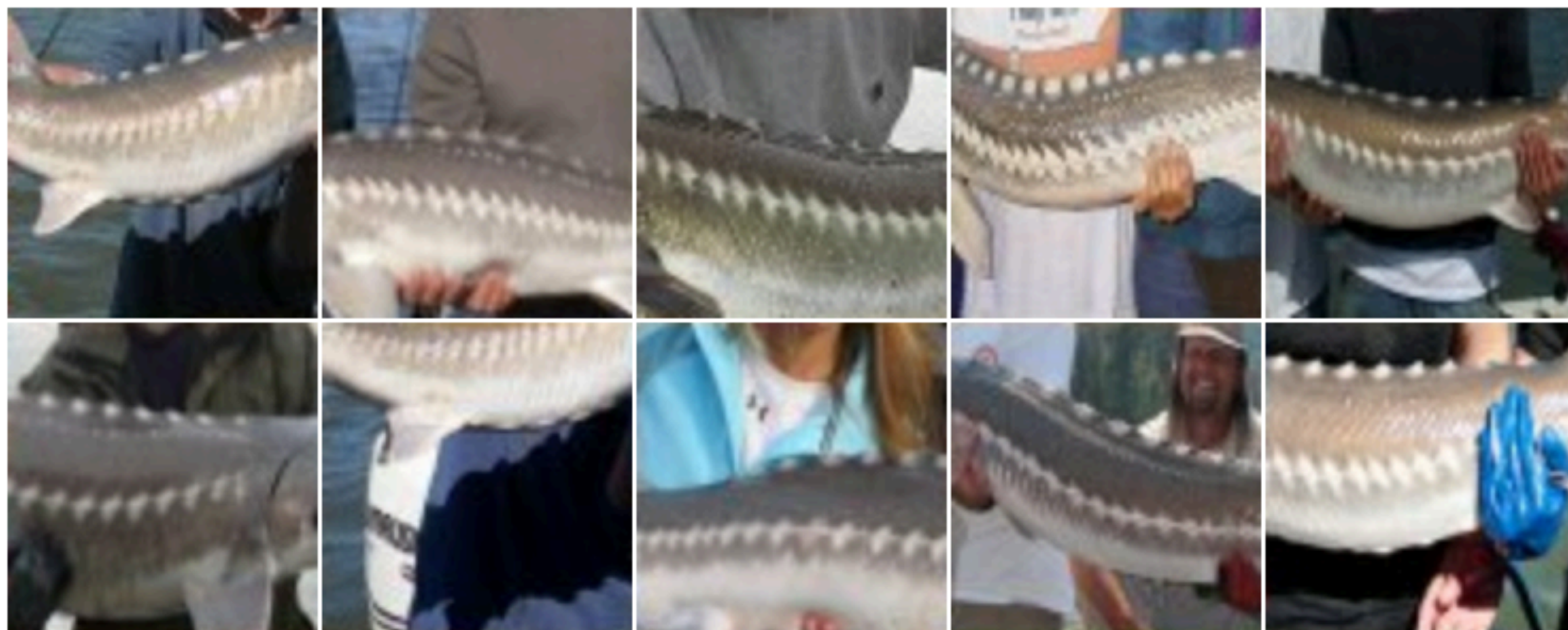
POSITIVE CHANNEL



Channel Objective



Diversity

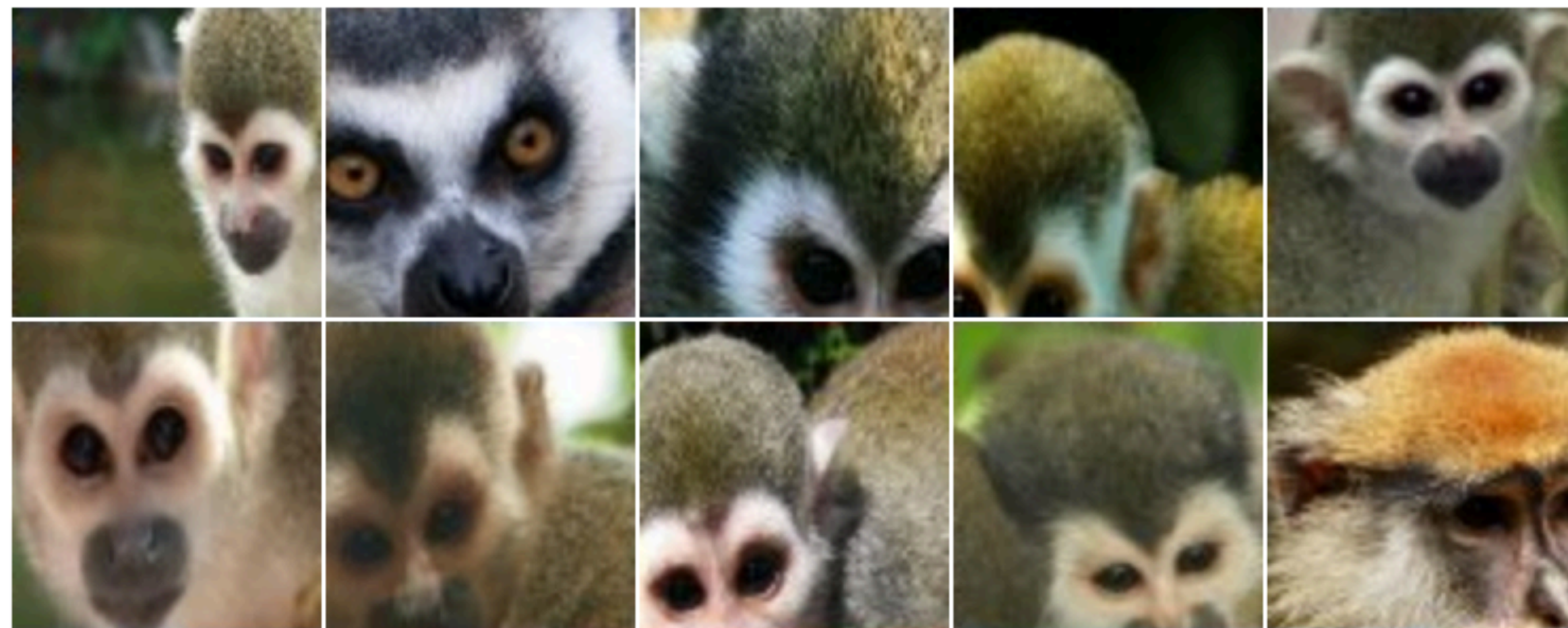


Dataset examples

NEGATIVE CHANNEL



Negative Channel



Negative dataset examples

POSITIVE CHANNEL



Channel Objective



Diversity



Dataset examples

NEGATIVE CHANNEL



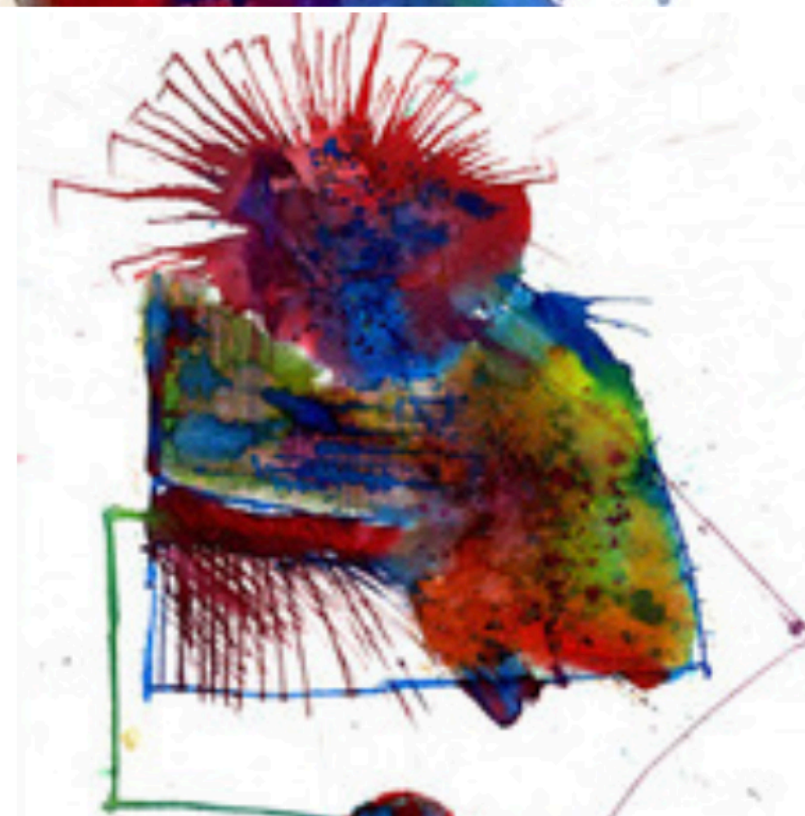
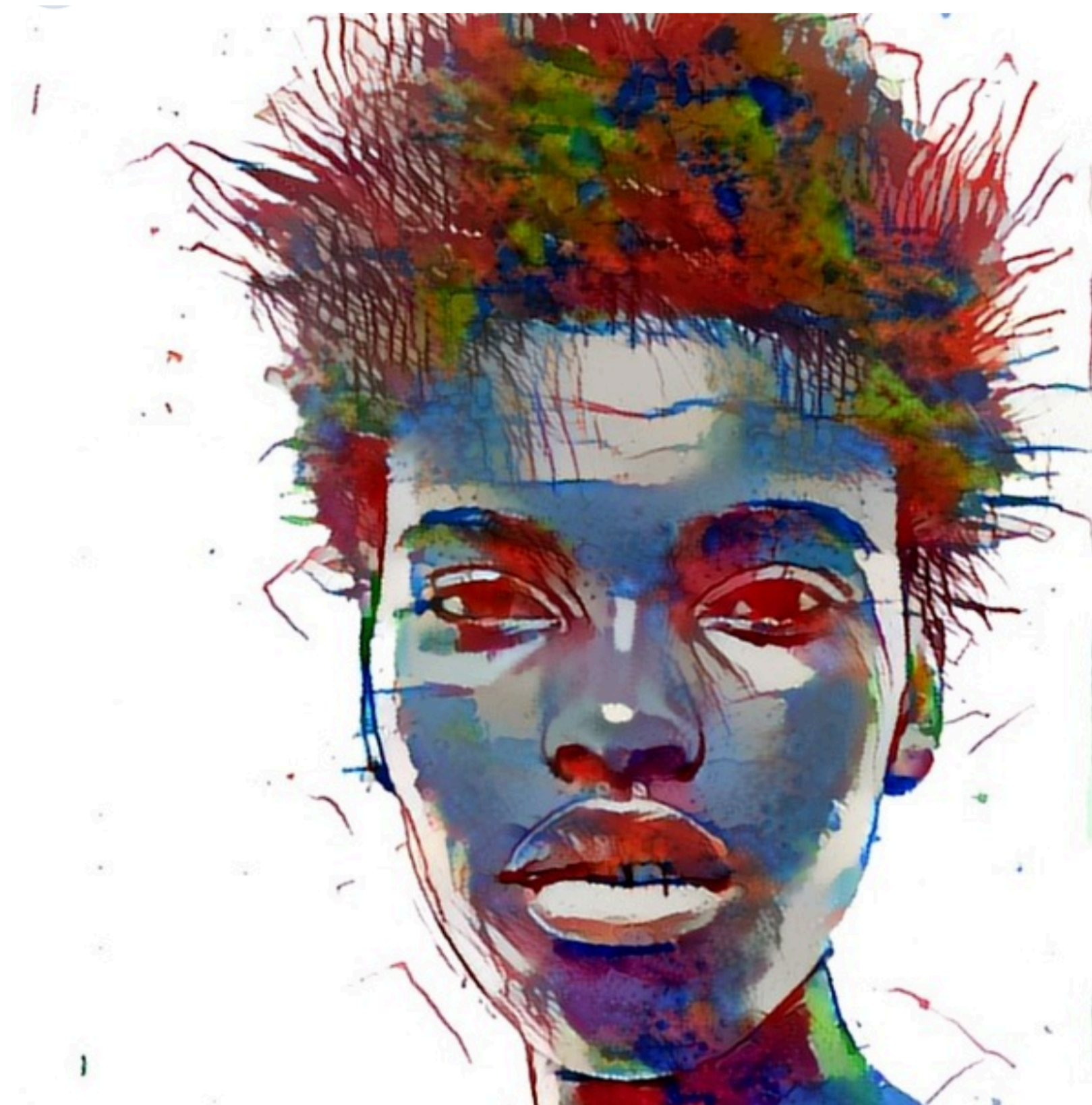
Negative Channel



Negative dataset examples

Style Transfer

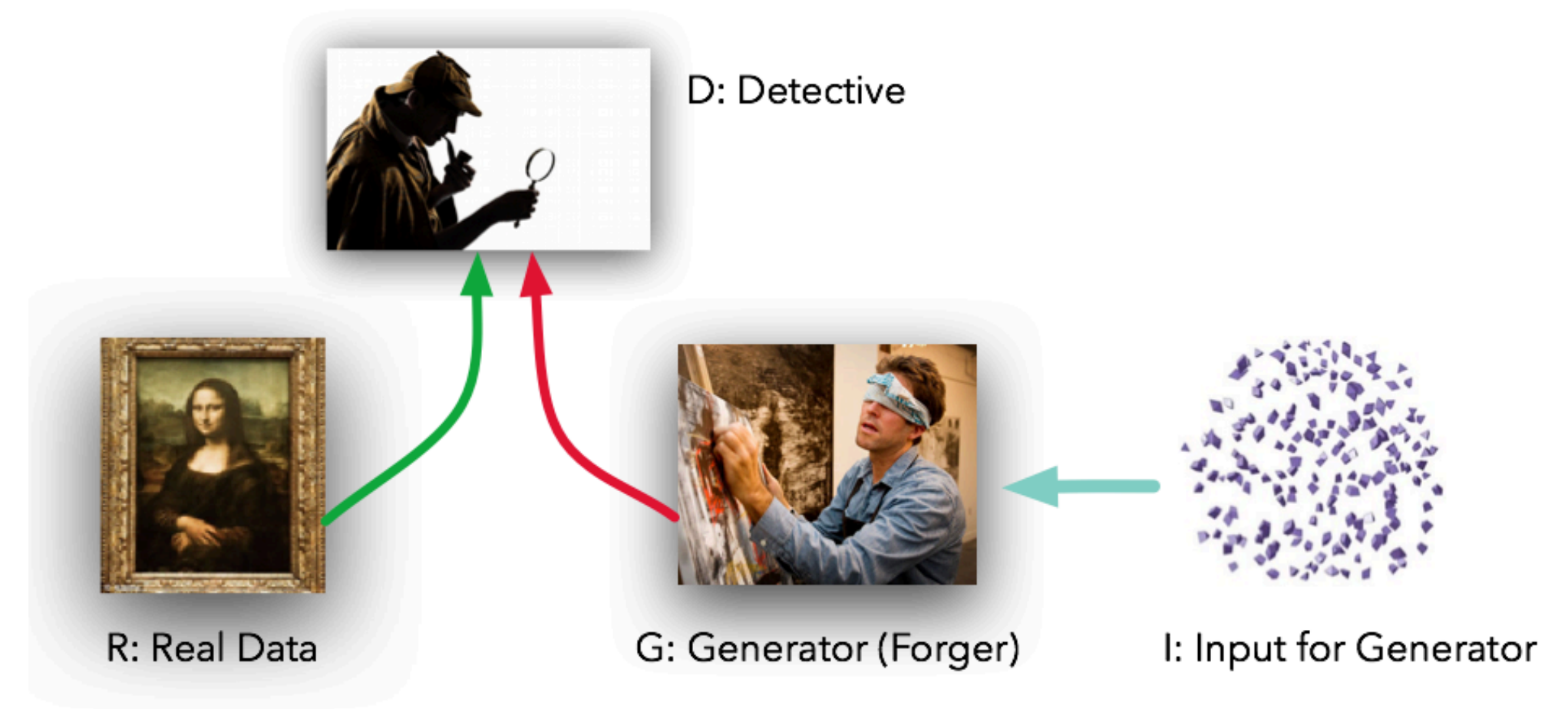
[link: Ostagram válogatás](#)



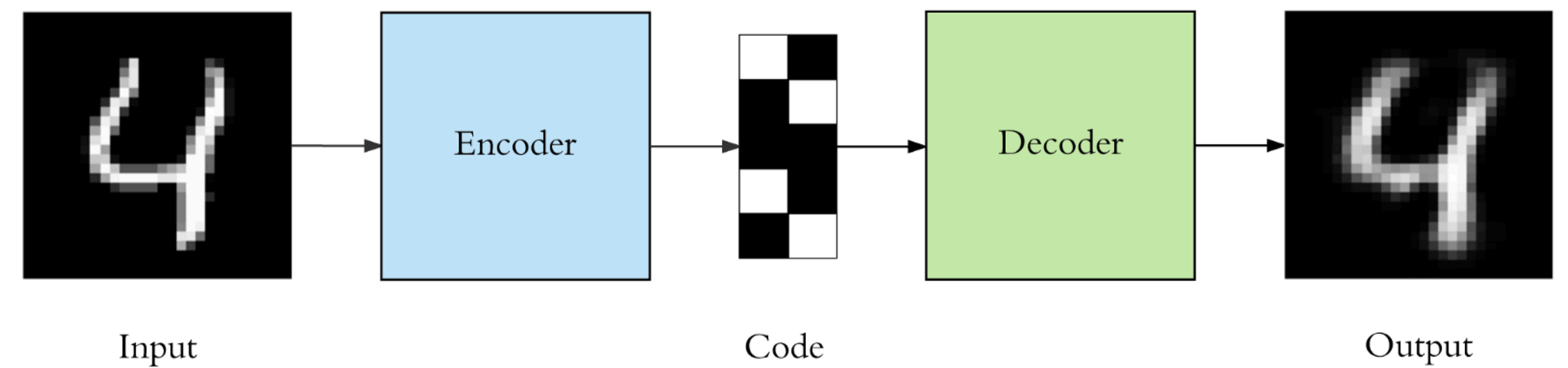
Generatív neuronhálók

Generatív neuronhálók

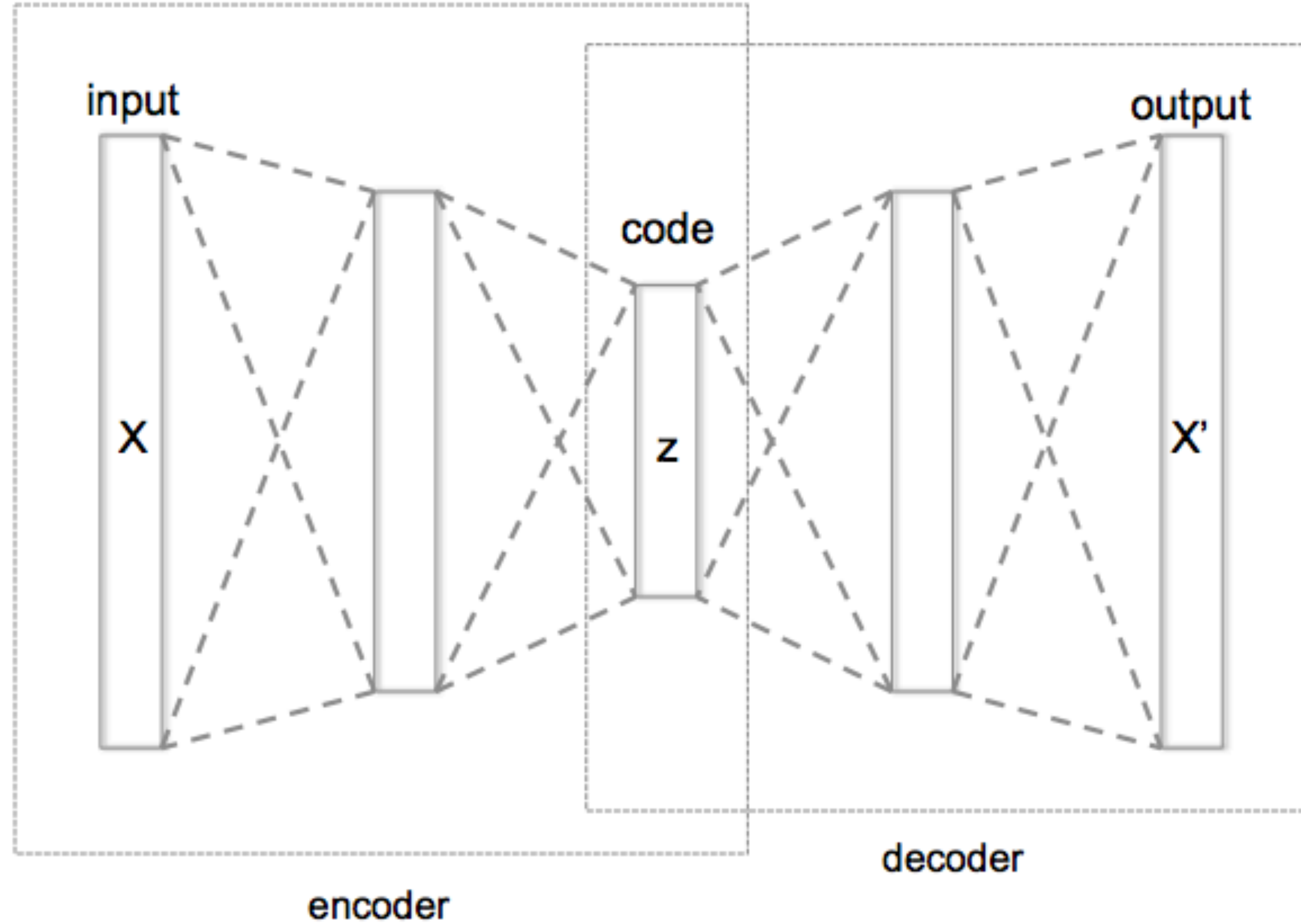
Generative Adversarial Network



Variational Autoencoder

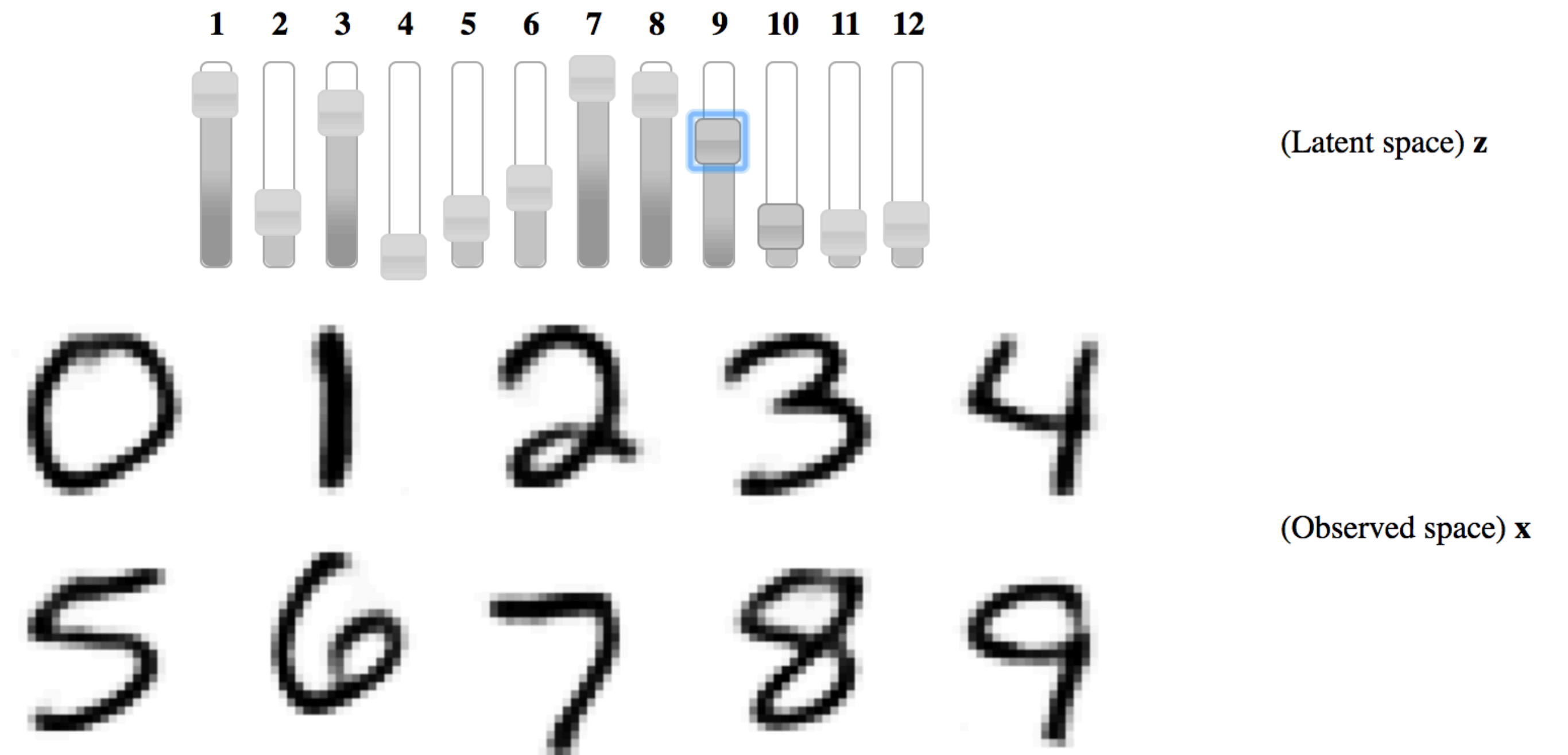
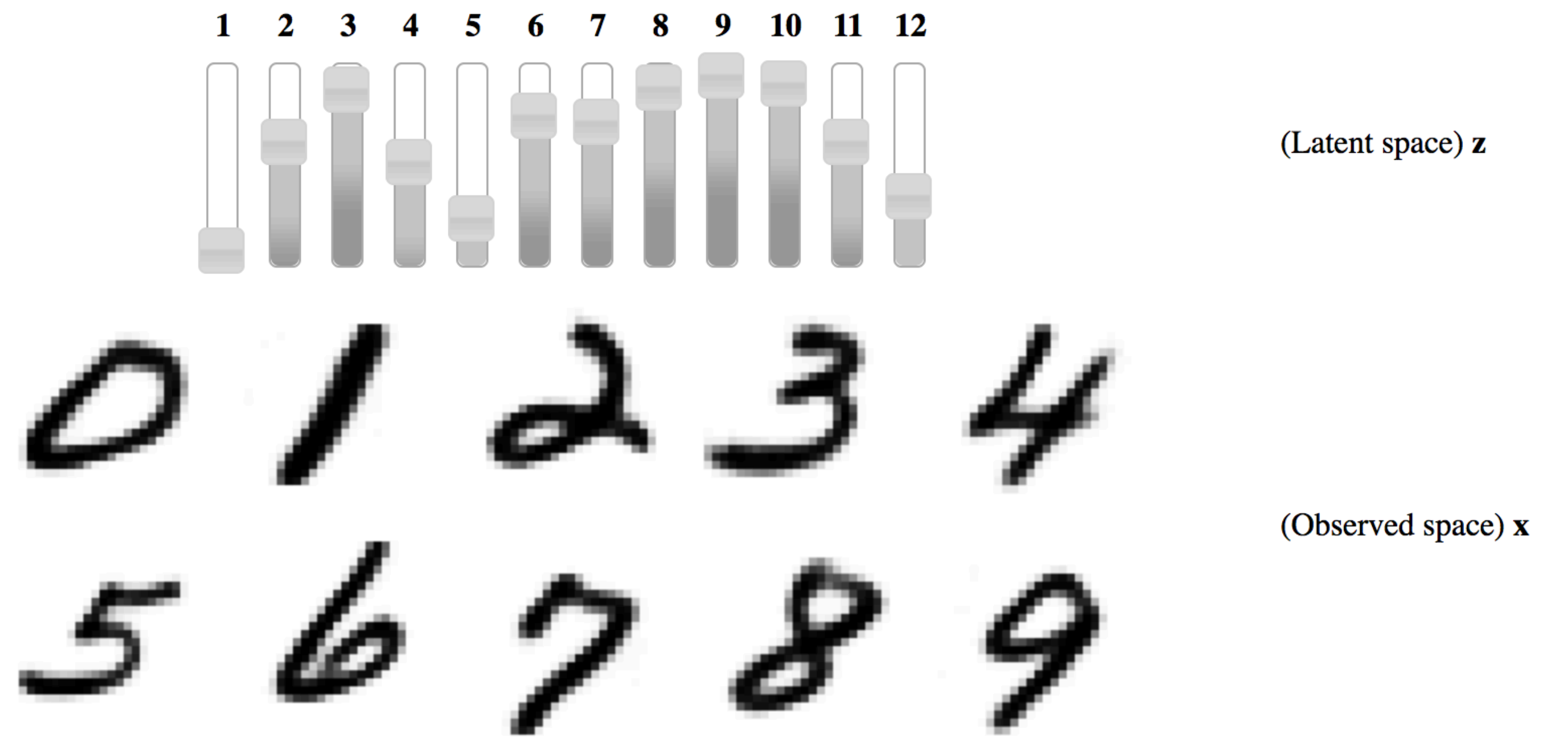


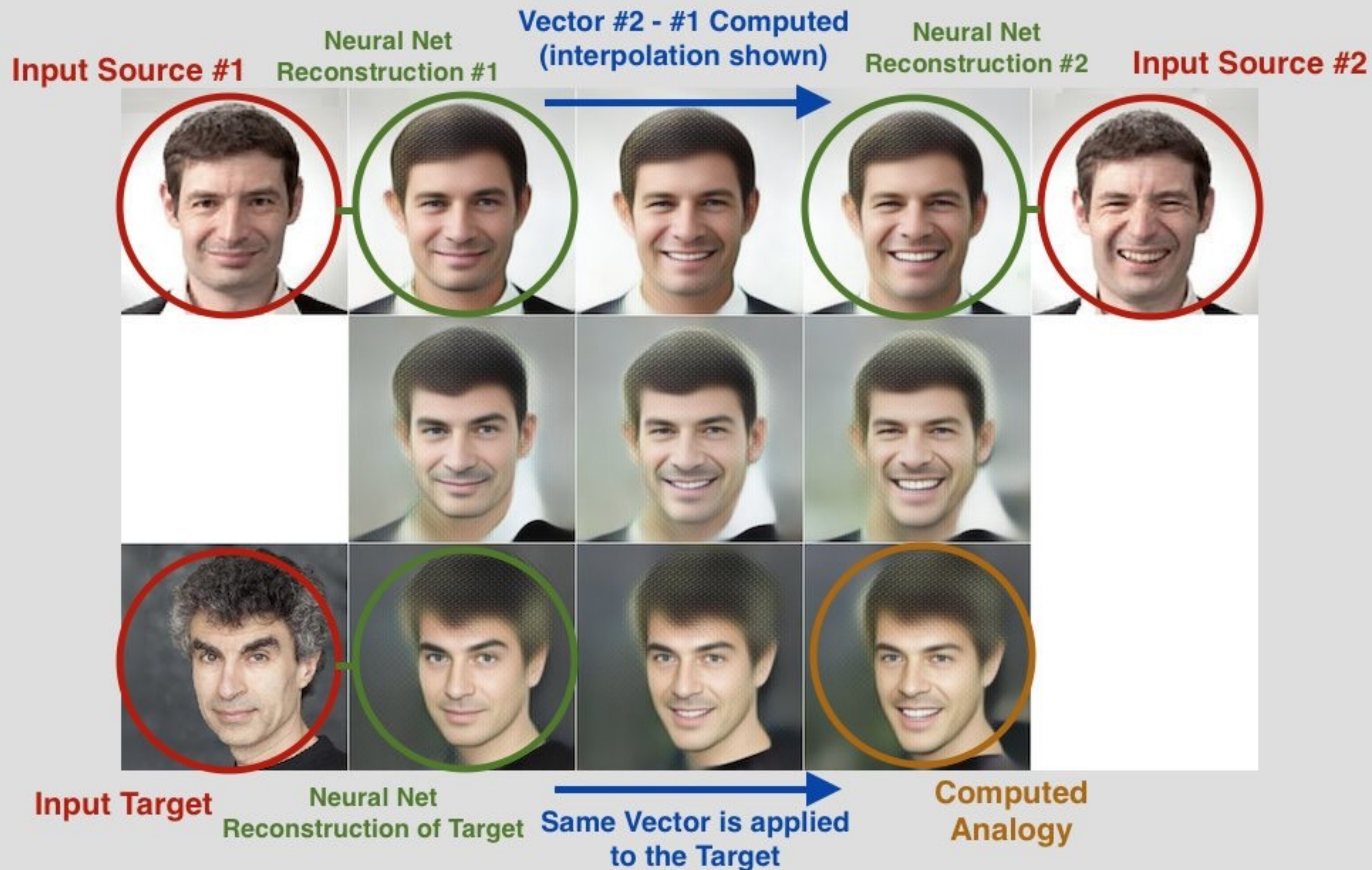
Autoencoder



Látens reprezentációk generatív modellekben

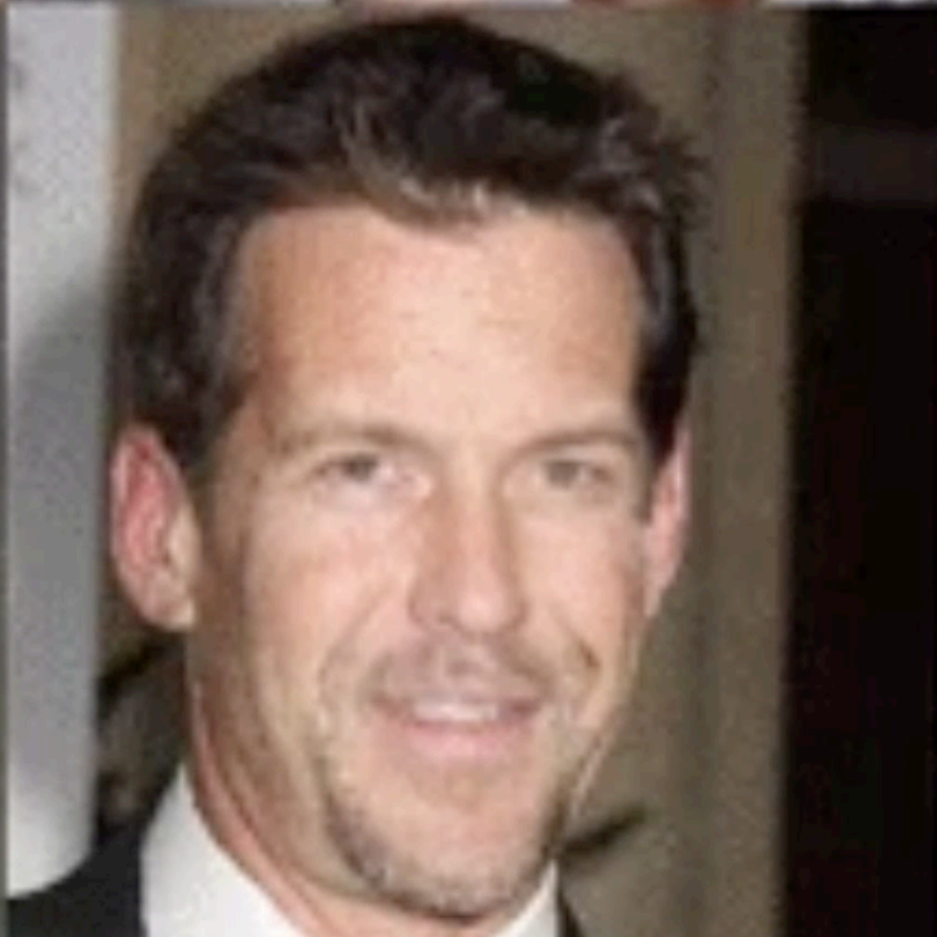
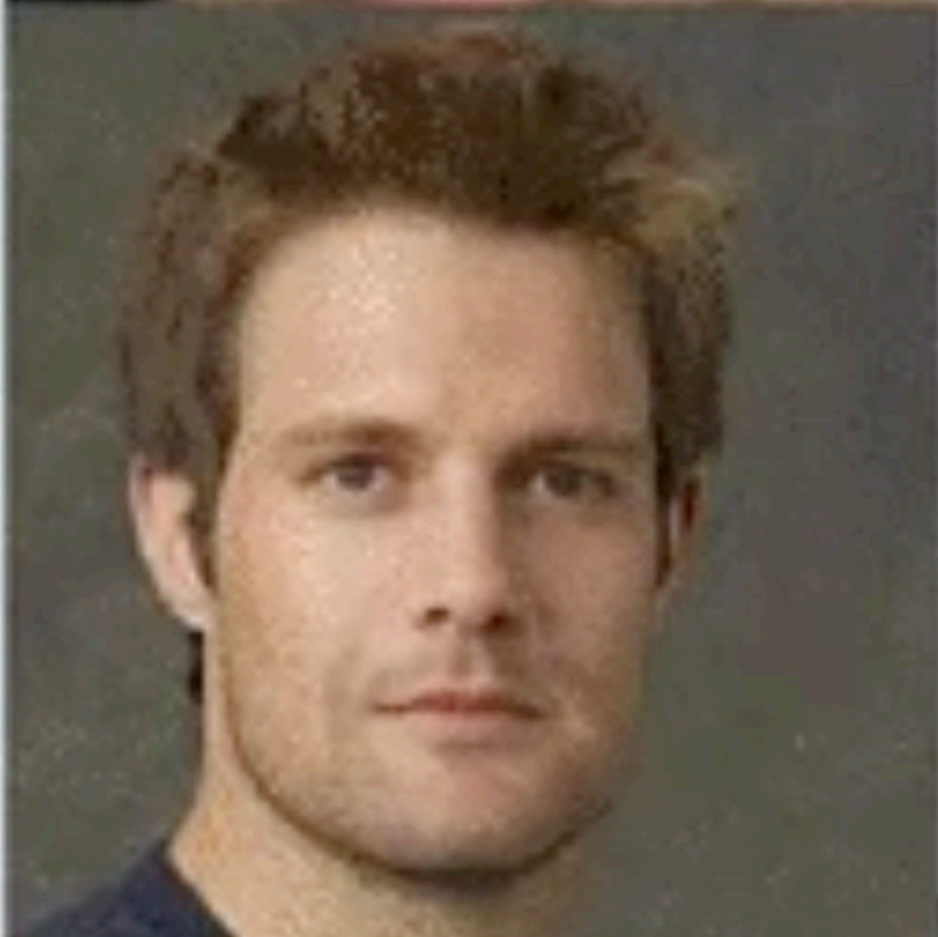
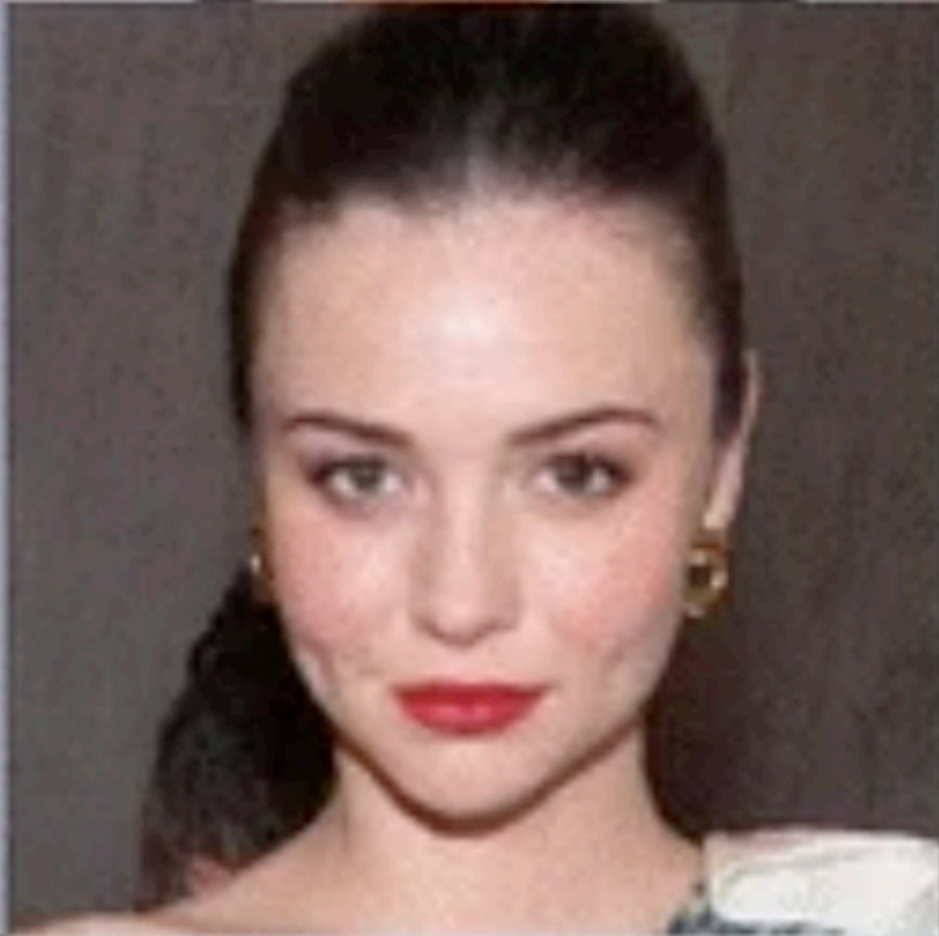
Digits autoencoder demo



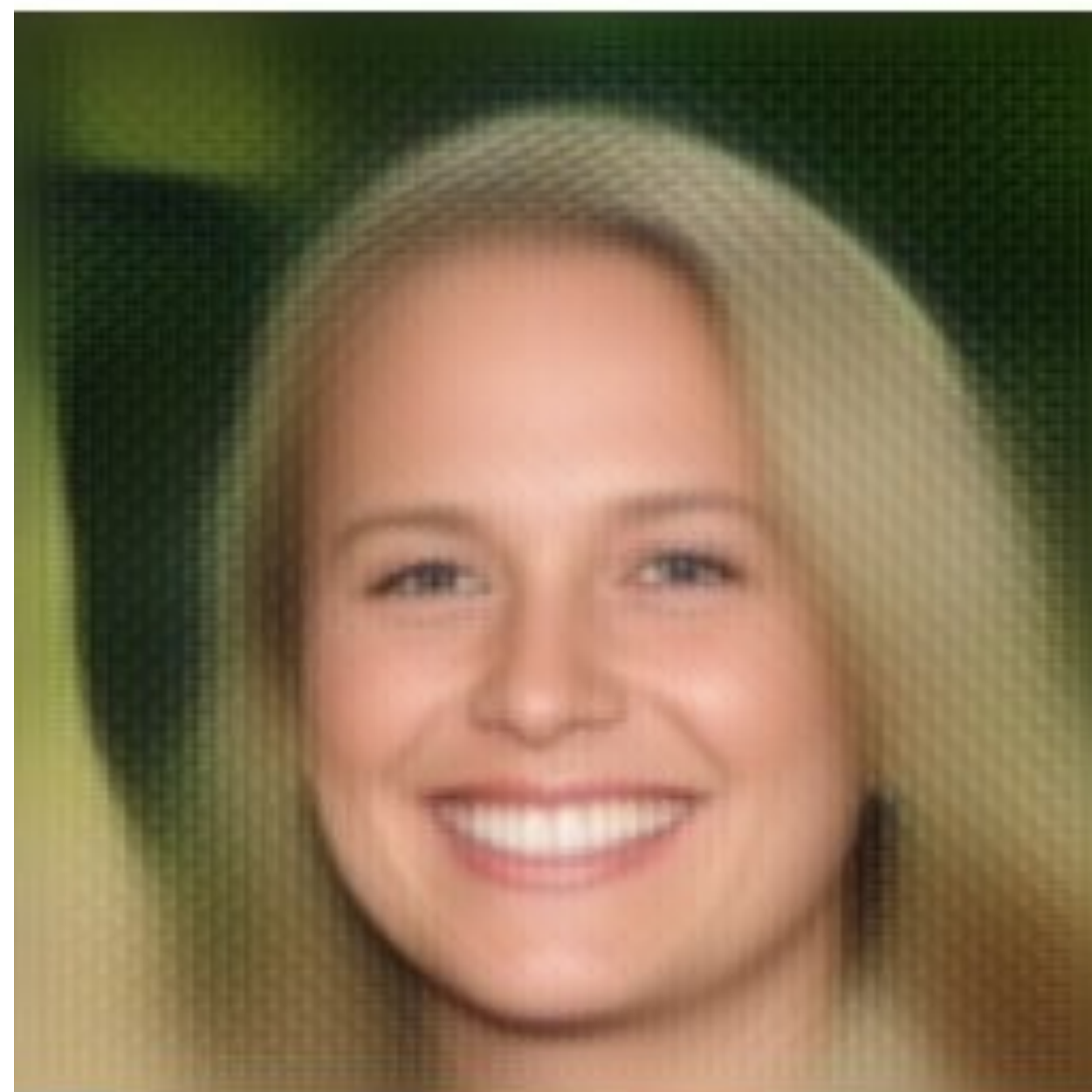


The relationship between Source #1 and Source #2 is applied to the target image.

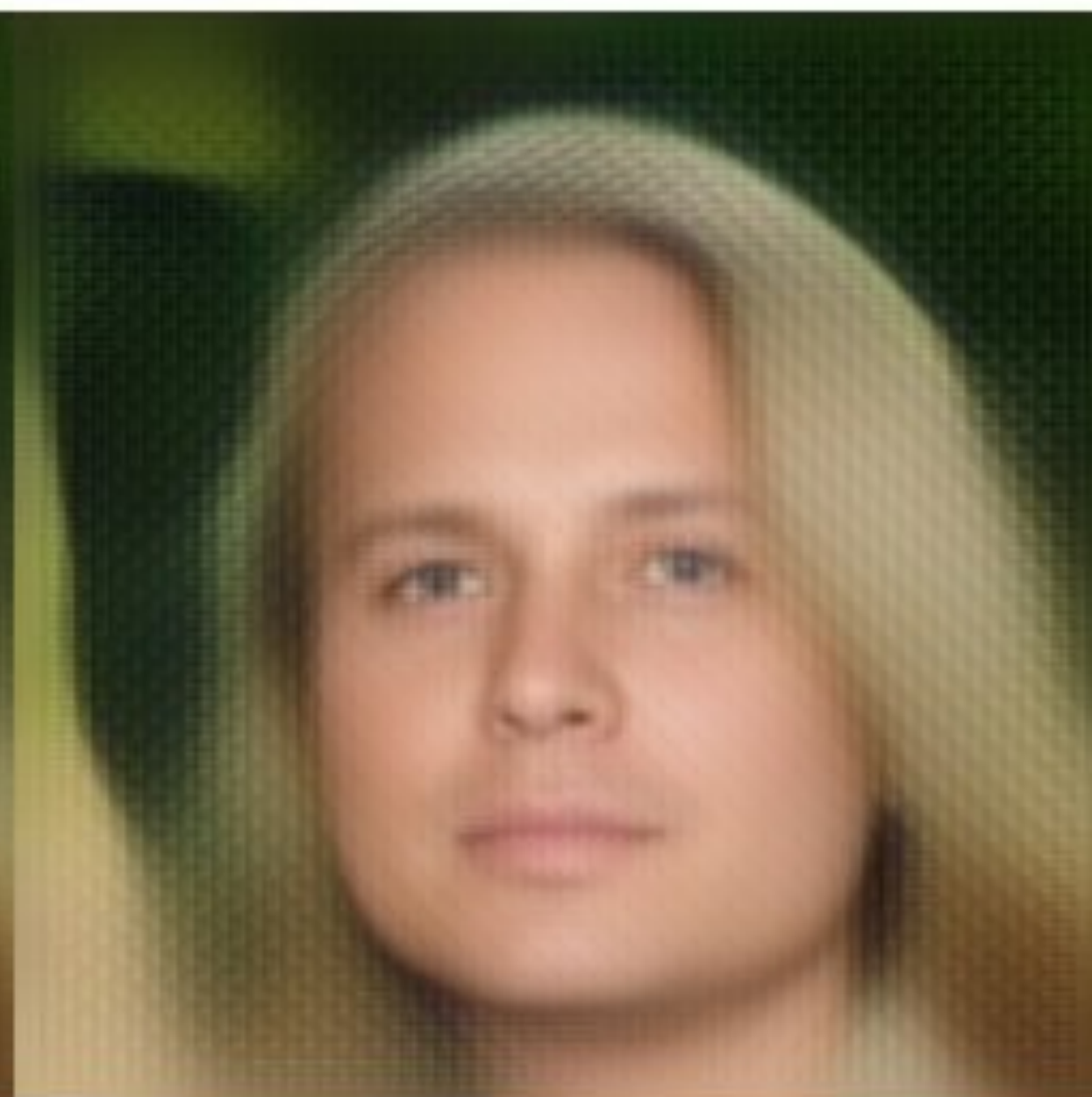




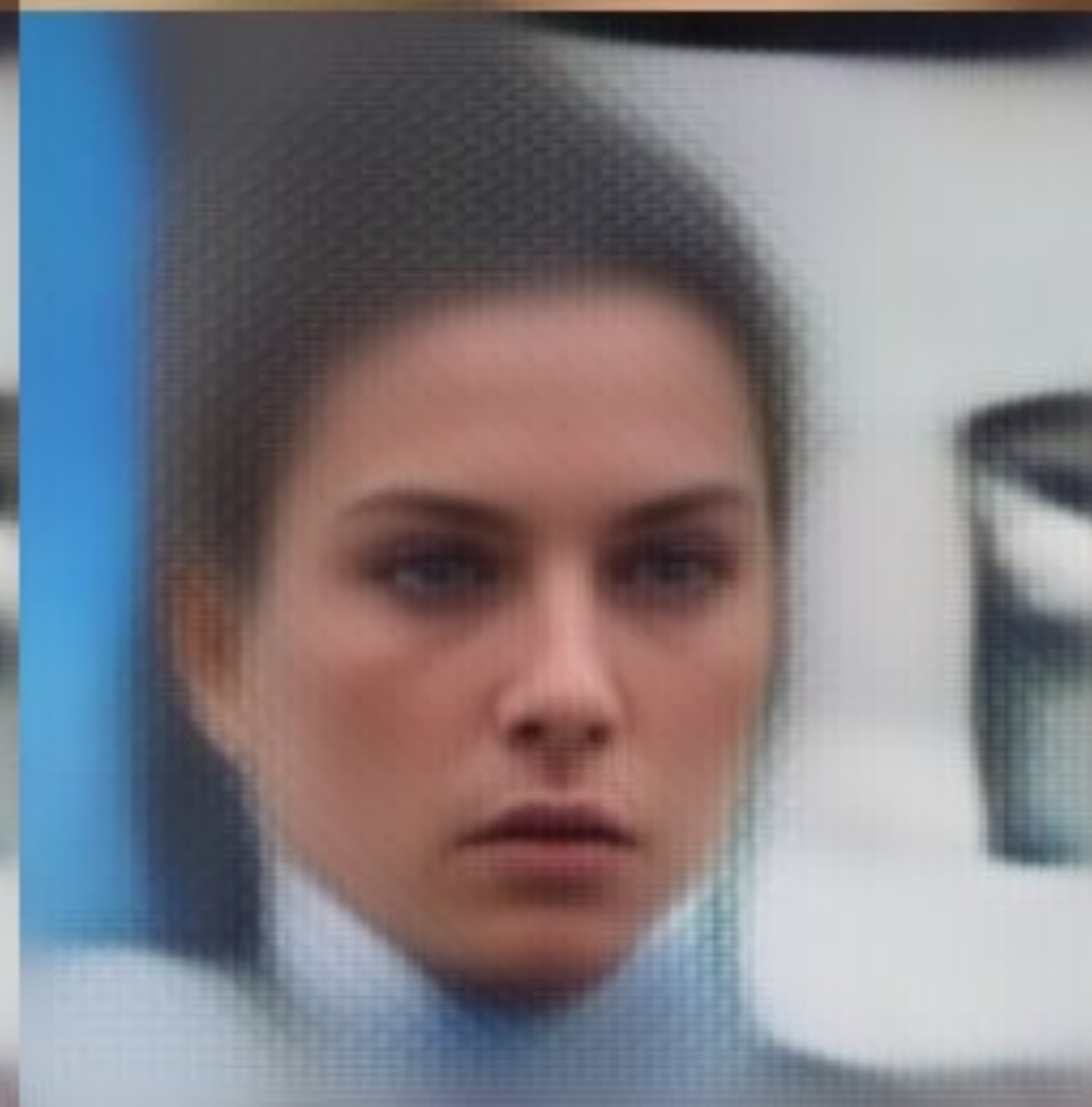
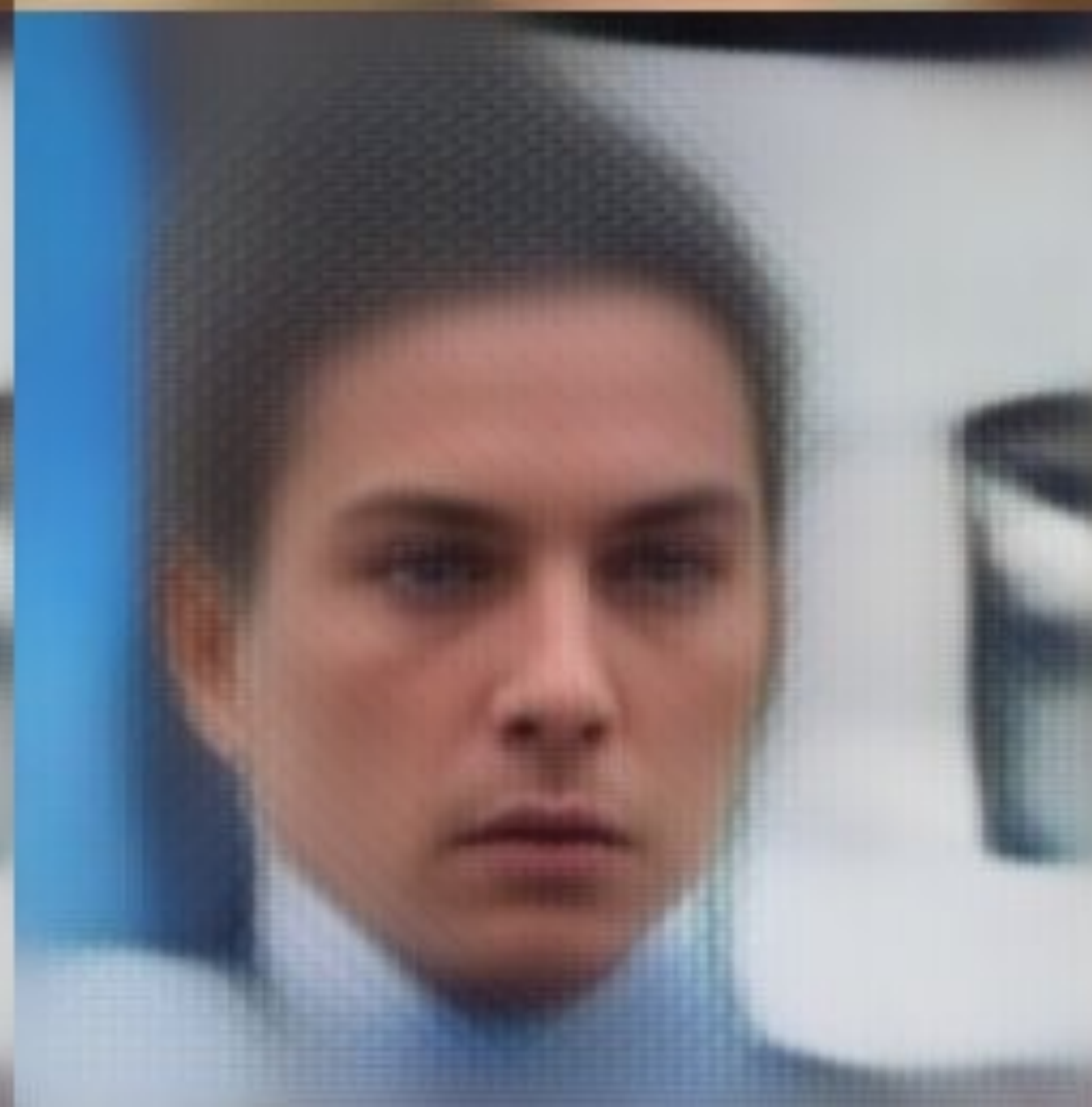
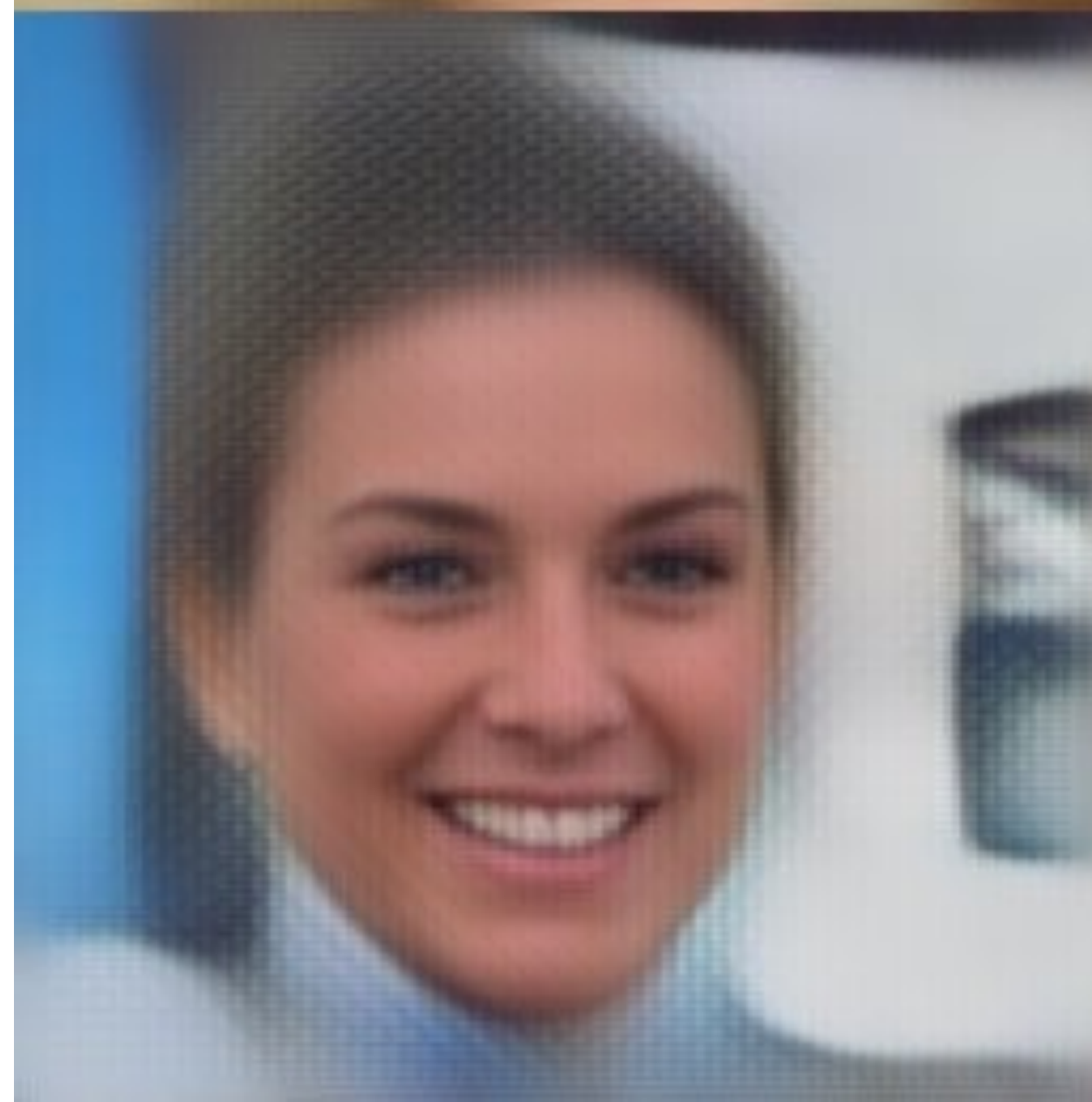
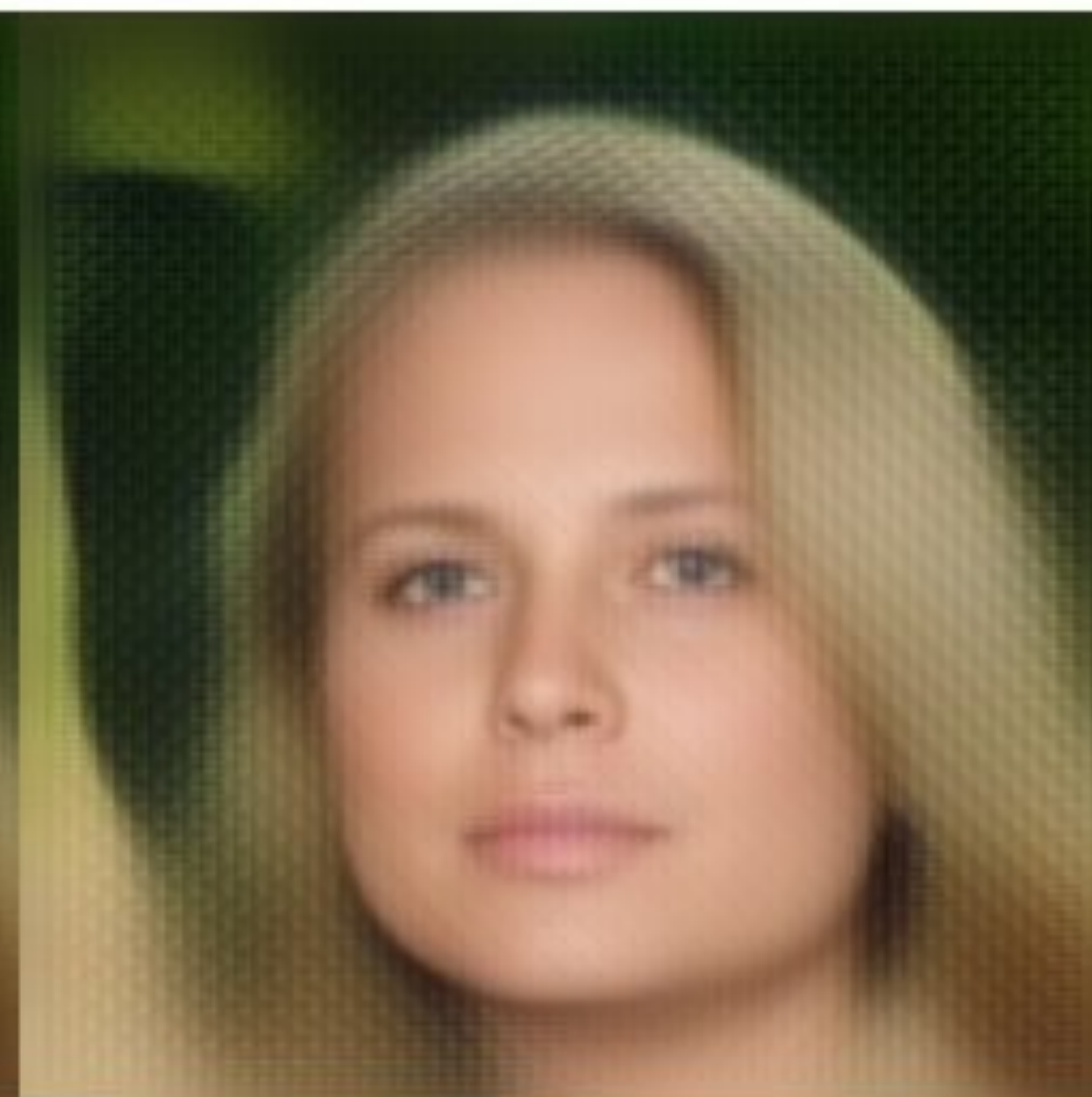
Reconstruction



Gender-biased
smilevector



Gender-balanced
smilevector





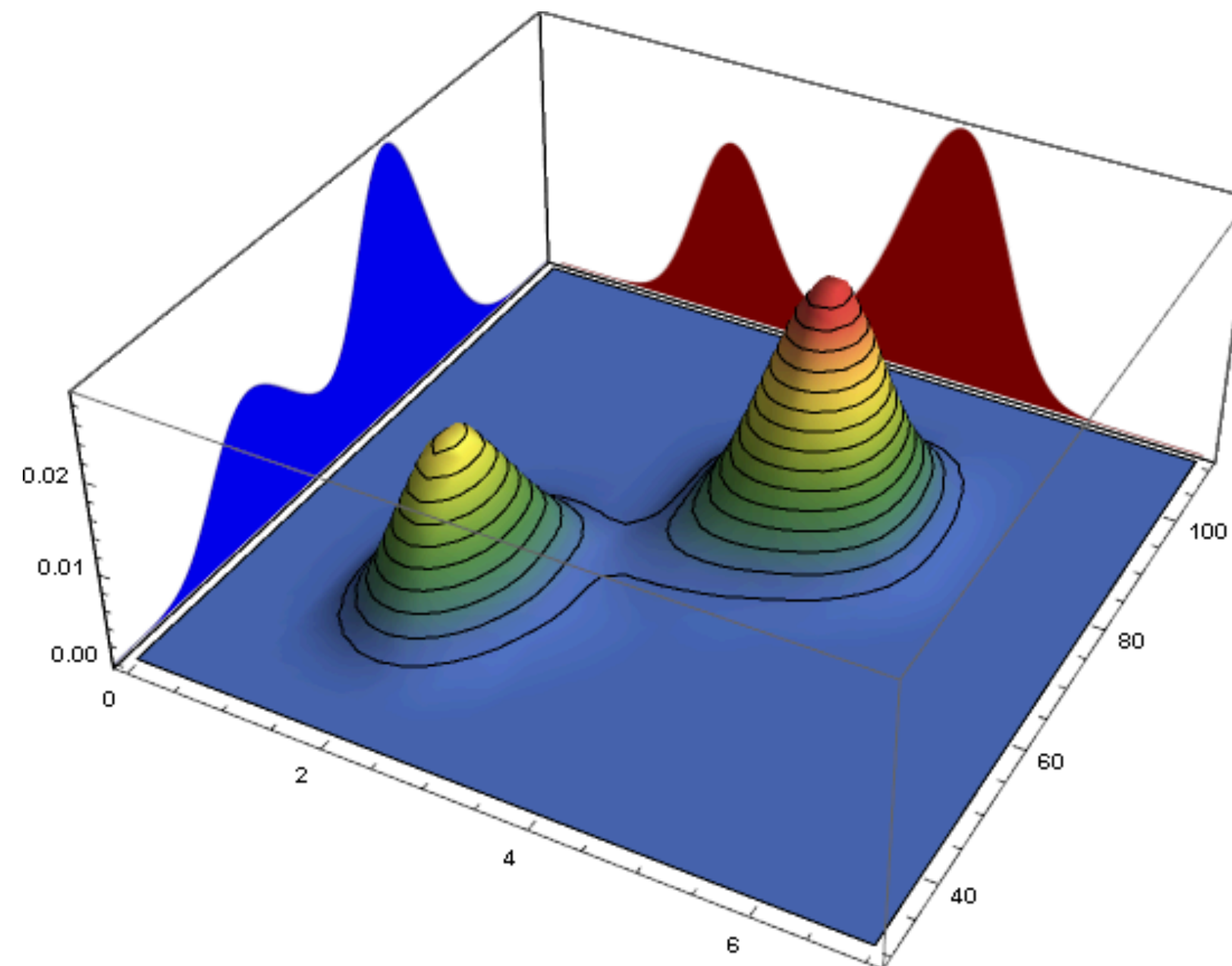


Wasserstein, távolság
más néven
earth mover's distance



$$W_1(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \mathbb{E}_{(x,y) \sim \gamma} d(x, y),$$

ahol $\Gamma(\mu, \nu)$ a csatolások halmaza. A csatolás olyan mérték $M \times M$ felett, amelynek marginális eloszlásai az első és második koordinátán μ illetve ν .



Kantorovics-Rubinstein dualitás

$$W_1(\mu, \nu) = \sup_{f: M \rightarrow \mathbb{R}, \text{Lip}(f) \leq 1} (\mathbb{E}_{x \sim \mu} f(x) - \mathbb{E}_{y \sim \nu} f(y))$$

Ez egy klasszikus lineáris programozási dualitás, és ahogy ez ilyeneknél szokásos, az egyik irányú egyenlőtlenség könnyű, a másik nehéz.

Nem fogjuk bebizonyítani, de legalább nézzük meg a könnyű irányát.

Az alvállalkozó. (Caffarelli via Villani) A μ 1 tonnás homokkupacot kellene áthordanunk a ν pozícióba. Egy tonna homok x -ből y -ba való áthordásának költsége $d(x, y)$, a költség egyenesen arányos a tömeggel. Megjelenik egy gyanús vállalkozó, és vállalja, hogy áthordja a homokkupacunkat. Rajzol nekünk egy f potenciálfüggvényt (domborzati térképet), azzal a tulajdonsággal, hogy két pont közti szintkülönbség sosem nagyobb, mint a távolságuk. Az elszámolás módjának az alábbi javasolja: ha m tonna homokot x -ből y -ba kellett szállítania, azért $(f(y) - f(x))m$ költséget számláz.

Az világos, hogy $\mathbb{E}_{y \sim \nu} f(y) - \mathbb{E}_{x \sim \mu} f(x)$ összeget fog kiszámlázni, és ez teljesen független attól, hogy ő maga hogyan oldotta meg a szállítást. Tehát feltehetjük, hogy a lehető leghatékonyabban járt el, és ez esetben az is világos, hogy nem járhatunk rosszul az ajánlatával, az f Lipschitz-tulajdonsága miatt.

A dualitástétel nehéz iránya azt mondja ki, hogy jól sem fogunk járni.

Vissza a generatív modellekhez

- Van egy $g(z)$ generatív neuronhálónk, amelynek célja egy μ eloszlás megtanulása példákból. Sajnos egyelőre a v eloszlást tanulta meg.
- Jó veszteségfüggvény lehetne a kettő közti Wasserstein távolság. Ehhez egy f tanú-függvényt kellene találnunk.
- Ötlet: Reprezentálja az f függvényt egy második neuronháló (diszkriminátornak vagy kritikusnak hívjuk), amelynek veszteségfüggvénye a minimax-tétel jobb oldalán szereplő különbség. Legyen f 1-Lipschitz.
- **Tehát a kritikust azért jutalmazzuk, ha nagy pozitív értékeket ad a tényleges példákra, és nagy negatív értékeket ad a “hamisított”, g által generált példákra.**
- A generátort (hamisítót) azért jutalmazzuk, ha megtéveszti a kritikust.

Wasserstein Generative Adversarial Network

Progressive Growing of GANs videó

videó: One hour of imaginary celebrities

...igen, az arcok rajzolgatása egy haszontalan alkalmazás,
de a gyógyszeresztéztől a beszélgetőprogramokig
sokféle ígéretes kísérlet van hasznos alkalmazásokra is.

Hiperbolikus beágyazások

Holografikus függvények

Gradiens regularizáció

Deep learning a kombinatorikus optimalizálásban

Ajánlott olvasmányok:

- <http://www.deeplearningbook.org/>
- <http://unsupervised.cs.princeton.edu/publications.html>
- <http://cs231n.stanford.edu/>
- 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 reimplementációja. A kód módosíthatásából lehet a legtöbbet tanulni.

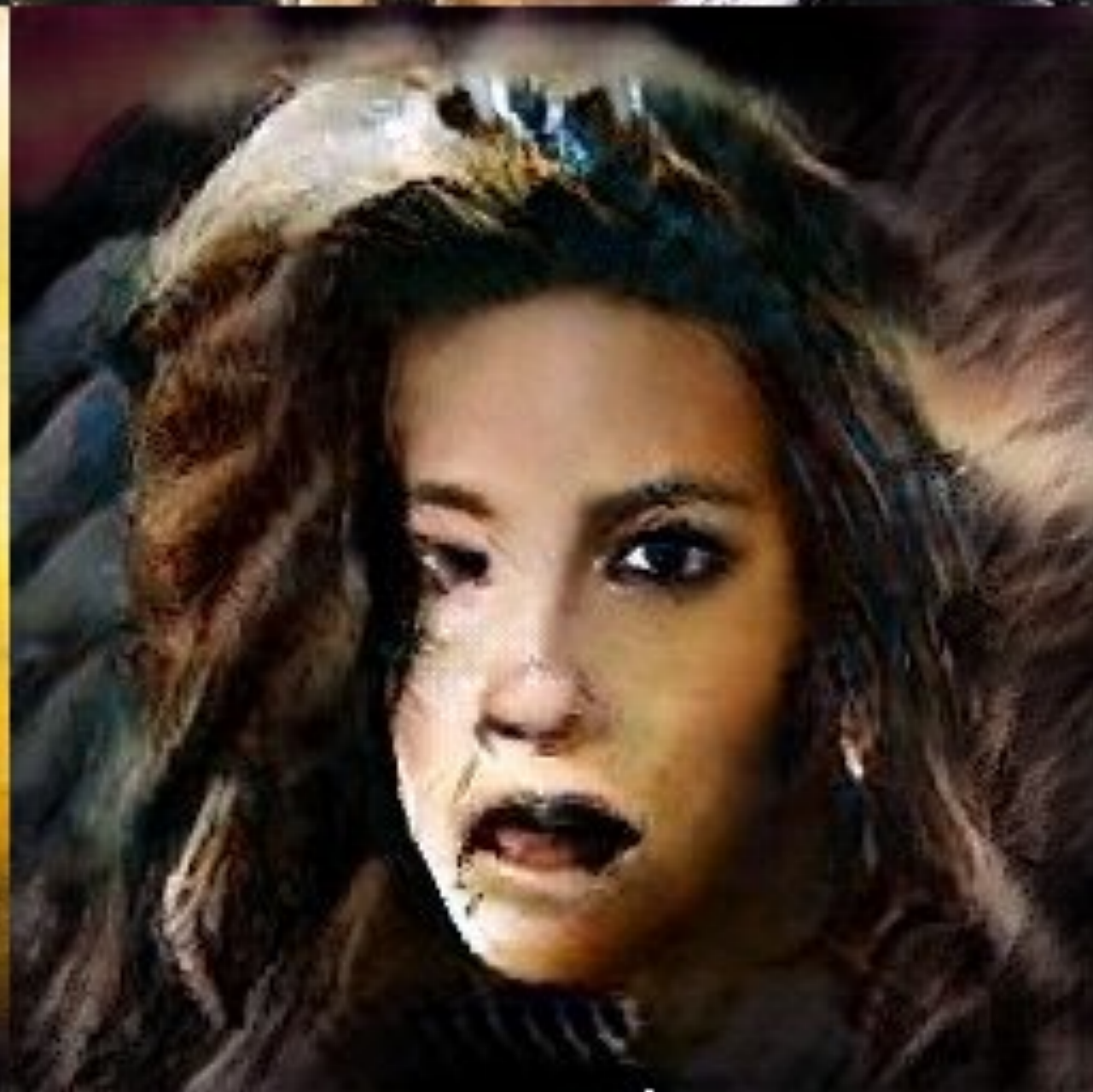
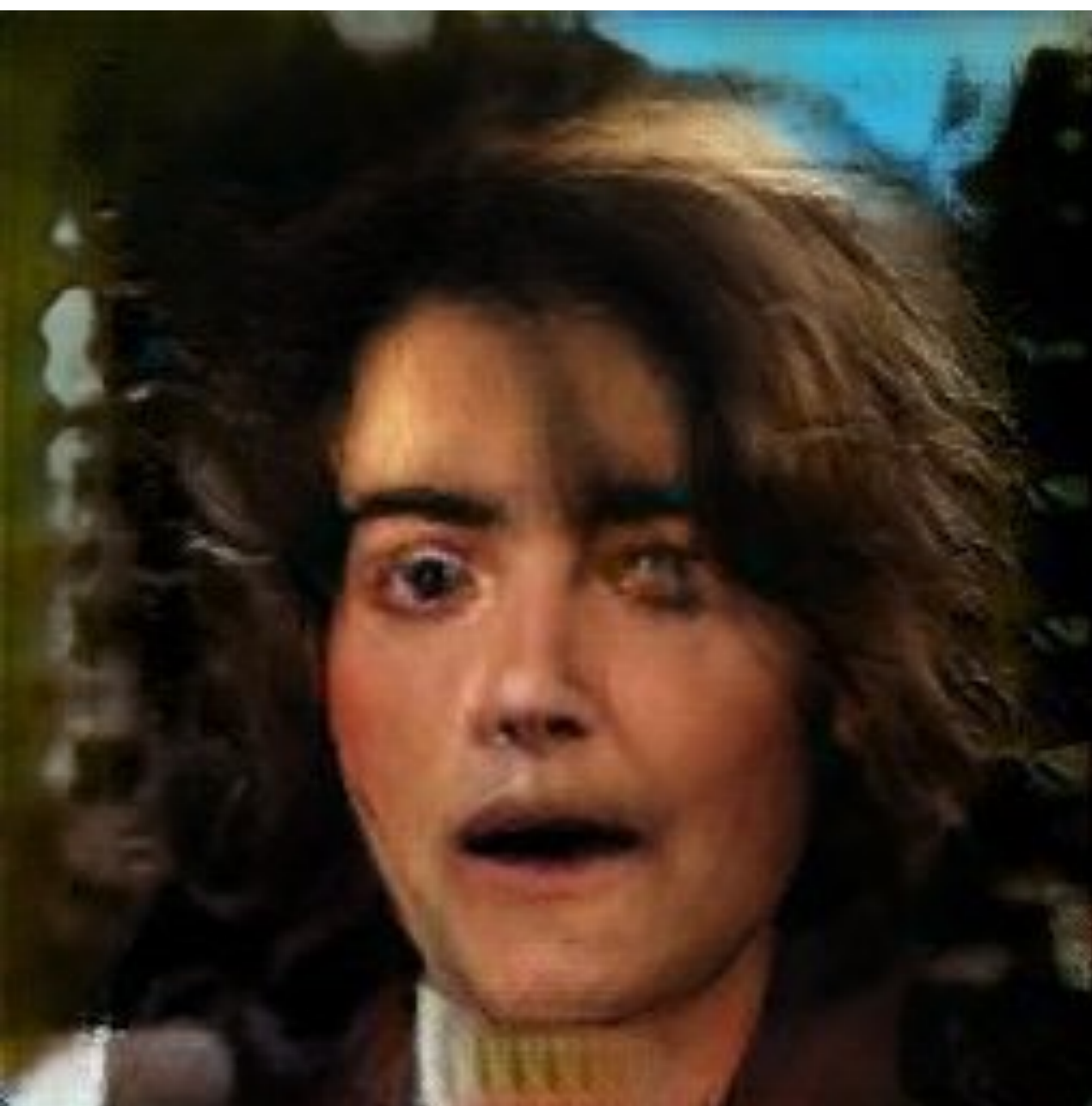
...ez mind közkincs:

- `pip install tensorflow keras`
- `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 # object detection & segmentation`
- `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`
- `pip install pytorch`
- `git clone https://github.com/tkarras/progressive\_growing\_of\_gans # image generation`

Köszönöm a figyelmet!

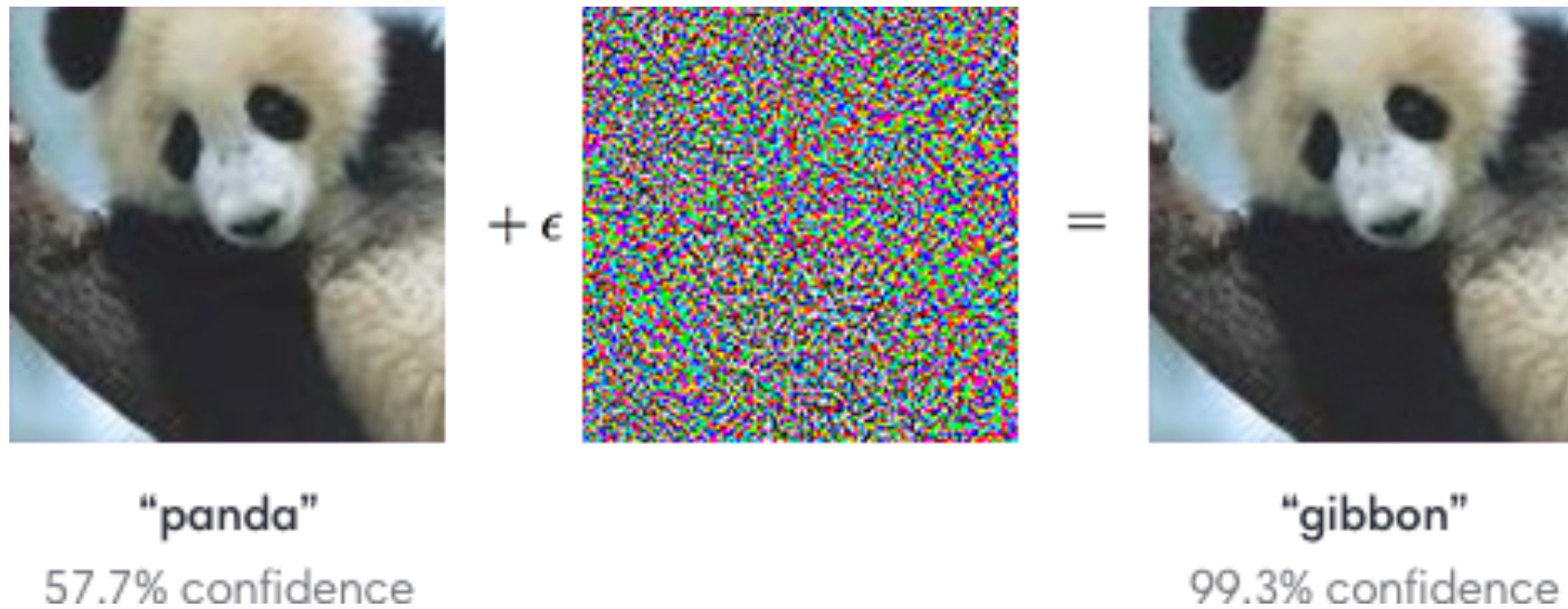
Eddig csak az eredményeit soroltam, de a deep learning nem varázsszer

- Hatalmas az adat-igénye.
- Hatalmas a számítási kapacitás-igénye.
- Nincs szisztematizált matematikai vagy mérnöki megalapozása. Főleg próbálgatás illetve szájhagyomány útján terjedő babonák alapján tervezik a hálózatokat.
- Kevés sikert mutathat fel azokon a területeken, ahol nem lehet olyan eltolás-invarianciákat kihasználni, mint a képeknél és a hangoknál.



...és még egy nagy gond

Adversarial perturbation



videó: fizikailag
megvalósított
adversarial példa

Adversarial patch



DeepFakes

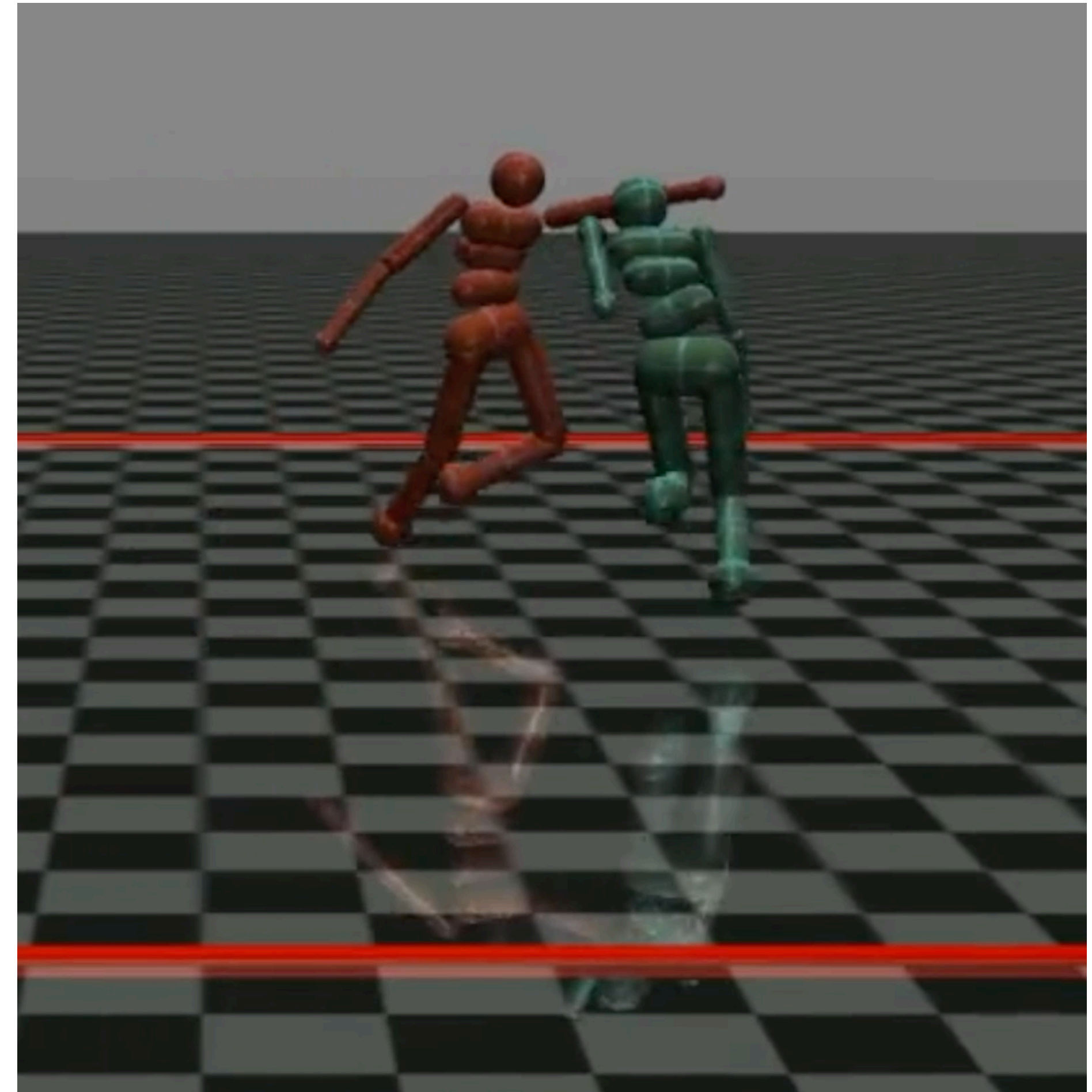
videó



Mély megerősítéssel tanulás szimulált 3D környezetben

videó: tanult mozgás

videó: competitive self-play



- 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.)