



# Neuronske mreže

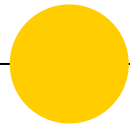
Generativni modeli



## Agenda

- *Variational Autoencoders*
- *Generative Adversarial Networks*
- Dodatno čitanje

# ***Variational Autoencoders***





## **Variational Autoencoders**

---

- Autoenkoder:
  - Veštačka neuronska mreža koja se koristi za učenje efikasnog kodiranja
  - Cilj je naučiti kompresovanu, distribuiranu reprezentaciju podataka
  - Osnovna struktura:
    - *Feed-forward MLP* sa:
      - Istim brojem ulaznih i izlaznih neurona
      - Nekoliko skrivenih slojeva



## ***Variational Autoencoders***

---

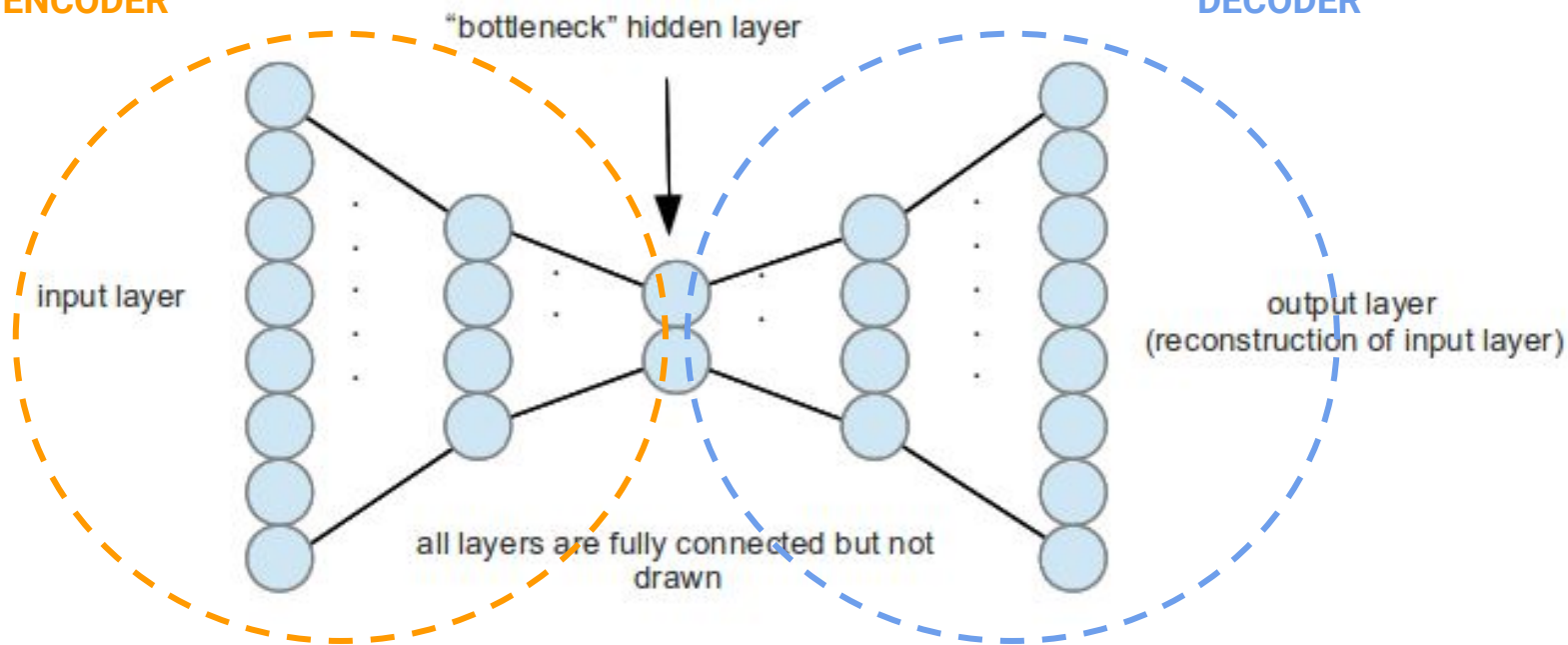
- Autoenkoder:
  - Za razliku od standardnog MLP, autoenkoder uči da rekonstruiše ulaz
  - Enkoder - deo mreže koji uči efikasno kodiranje podataka
  - Dekoder - deo mreže koji uči efikasno dekodiranje (rekonstrukciju) podataka.



# Variational Autoencoders

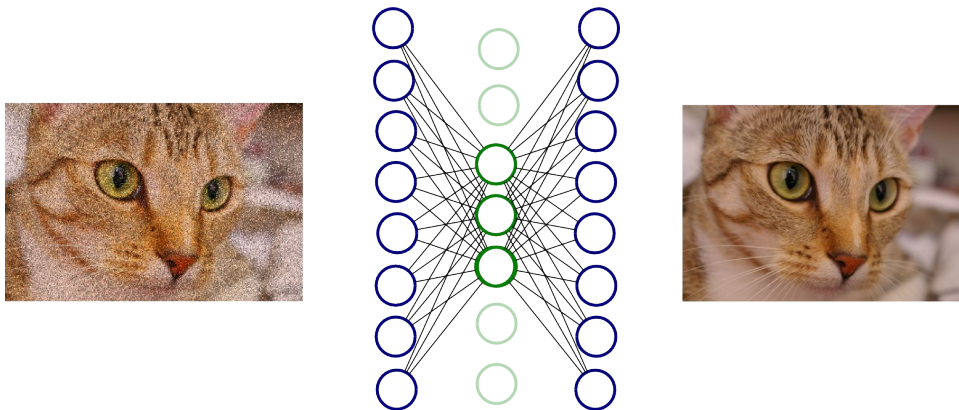
ENCODER

DECODER



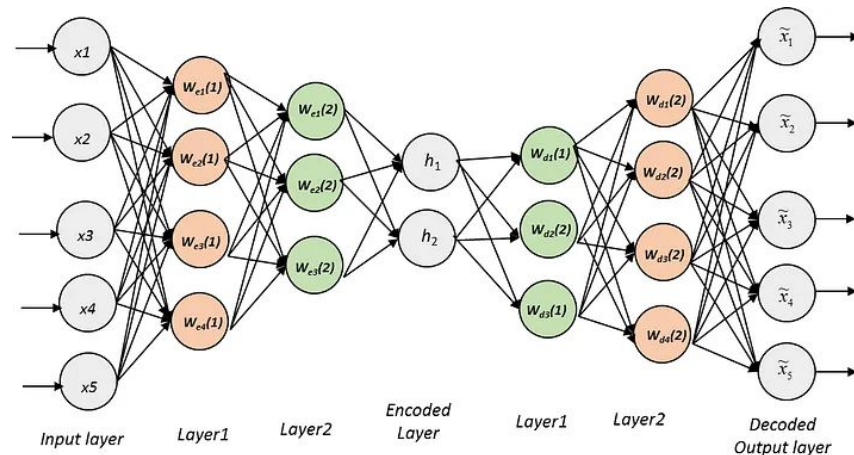
## ● Variational Autoencoders

- Varijacije autoenkodera:
  - **Denoising Autoencoder (DAE)** - ulaz sa šumom, na izlazu se očekuje rekonstruisan ulaz, ali bez šuma



# ● Variational Autoencoders

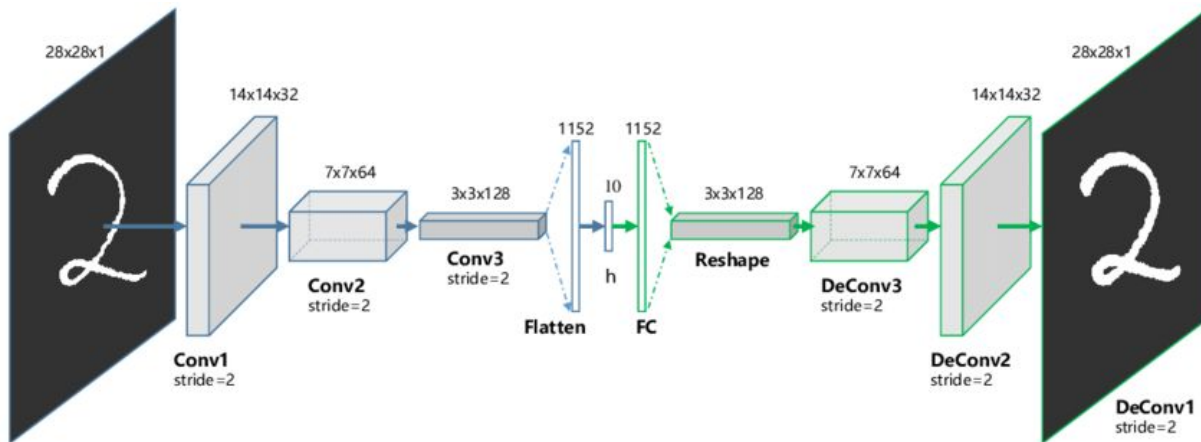
- Varijacije autoenkodera:
  - **Stacked Autoencoder (SAE)** - suštinski *deep* autoenkoder, ima više skrivenih slojeva





## ● Variational Autoencoders

- Varijacije autoenkodera:
  - *Convolutional Autoencoder (CAE)* - autoenkoder sa konvolutivnim slojevima





## **Variational Autoencoders**

---

- Obučavanje autoenkodera:
  - Standardni *Backpropagation* sa nekim od *Gradient Descent* algoritama
  - Cilj je što više smanjiti razliku između ulaza i rekonstruisanog izlaza:
    - Često se minimizuje *MSE* (*Mean Squared Error*) između svih ulaznih neurona i njima odgovarajućih izlaznih neurona.



## ***Variational Autoencoders***

---

- Upotreba autoenkodera:
  - Učenje osobina podataka
  - Smatra se da će autoenkoder tokom učenja rekonstrukcije ulaza naučiti i neke osobine podataka:
    - Konvolutivni autoenkoder nauči iste filtere kao i *CNN*

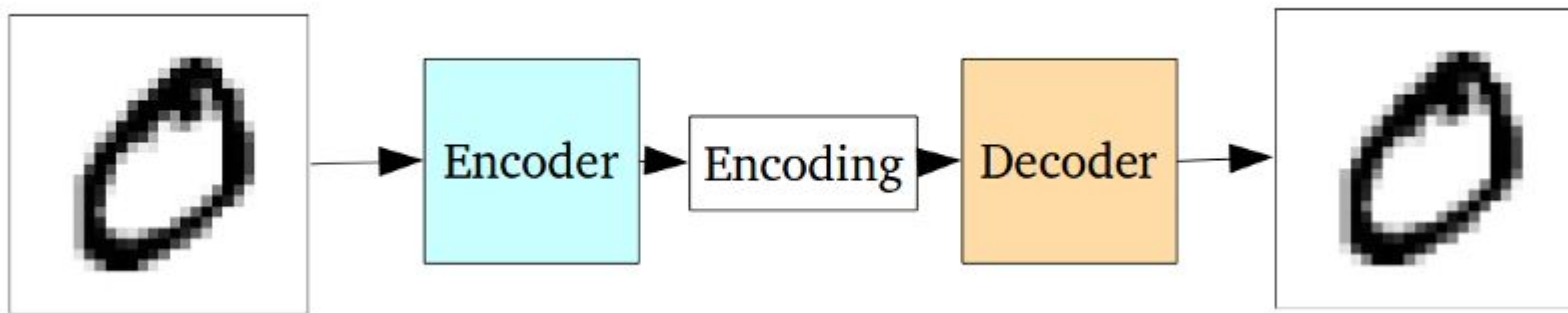


## Variational Autoencoders

- Upotreba autoenkodera:
  - Jako korisni ako nema puno obeleženih podataka:
    - Neobeleženi podaci se daju autoenkoderu - *pretraining*
    - Nakon *pretraining*-a, uzima se samo enkoder deo mreže
    - Na pretrenirani enkoder doda se još par slojeva i takva mreža se obučava nad obeleženim podacima - *fine-tuning*.

## ● Variational Autoencoders

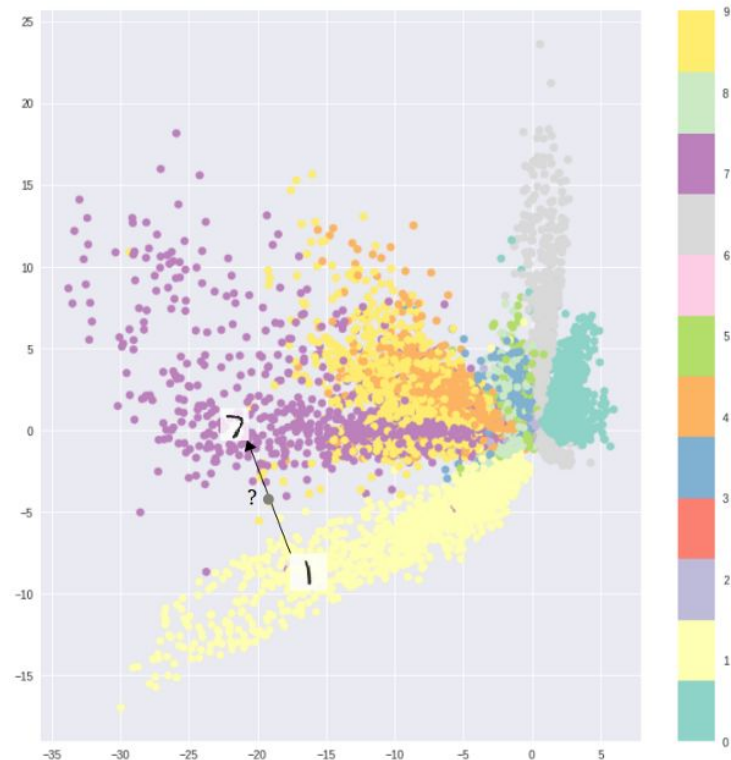
- Običan autoenkoder:



- Kako ćemo odrediti *encoding* vektor ako želimo da generišemo novi primerak?

## ● Variational Autoencoders

- Vizuelna reprezentacija 2D skrivenog (*latent*) prostora je prikazana na slici:

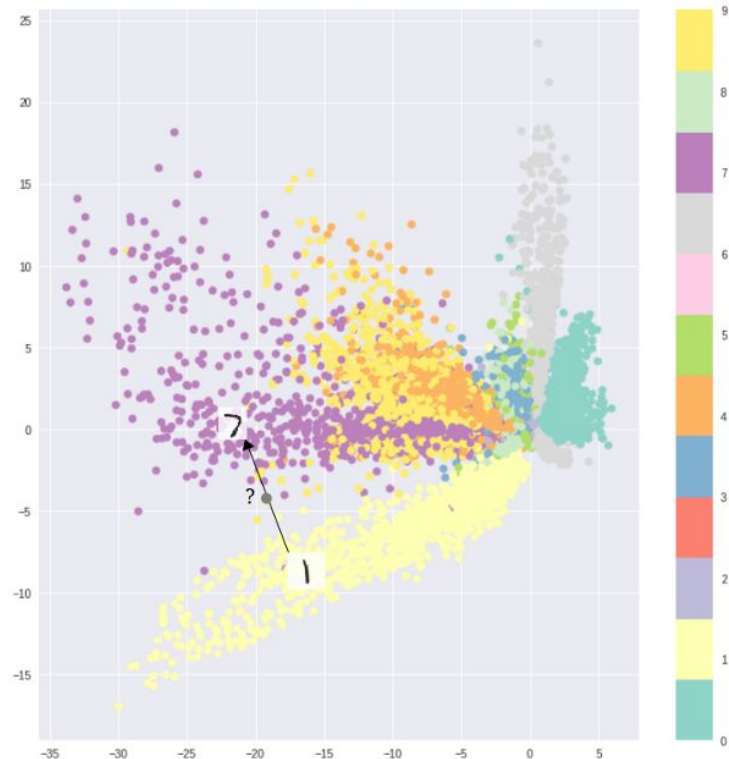




## Variational Autoencoders

Ako izaberemo *encoding* iz praznog dela prostora (u kome nije bilo primeraka), dekodler neće znati kako da se ponaša i generisaće loš izlaz.

Prostor nije modelovan kao kontinualan (postoje *gap*-ovi), što je problem u ovom slučaju jer se ne može interpolirati.





## **Variational Autoencoders**

---

- *Variational Autoencoders (VAE)* se prvi put pojavljuju 2013. godine
- Generativni model:
  - Omogućava generisanje dodatnih primeraka iz iste distribucije
  - Upravljamo u kom smeru će generisanje ići (smer varijacije u podacima)
  - Istraživanje postojeće distribucije.



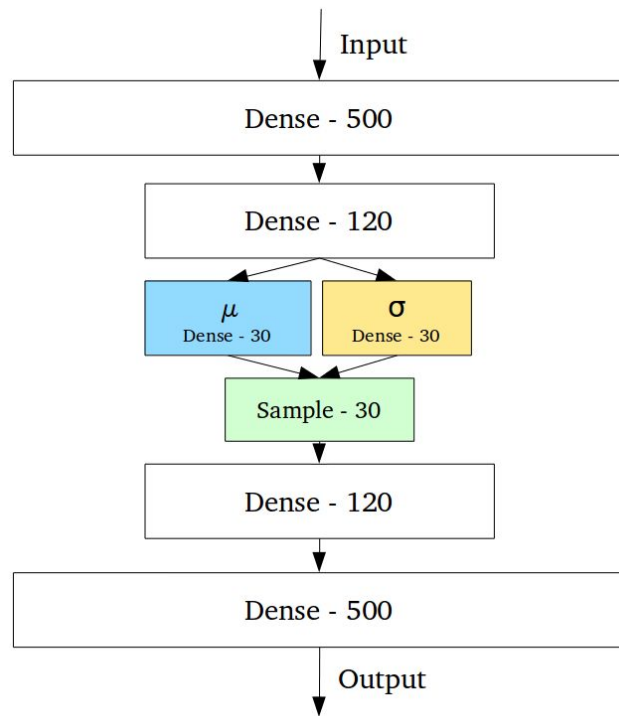
## ● **Variational Autoencoders**

- Običan autoenkoder tehnički jeste generativan model, ali radi u rekonstruktivnom režimu:
  - Nije mogao da radi bez enkodera
- VAE-u *encoding* treba samo prilikom treninga:
  - Mogu se ručno zadavati stvari za generisanje



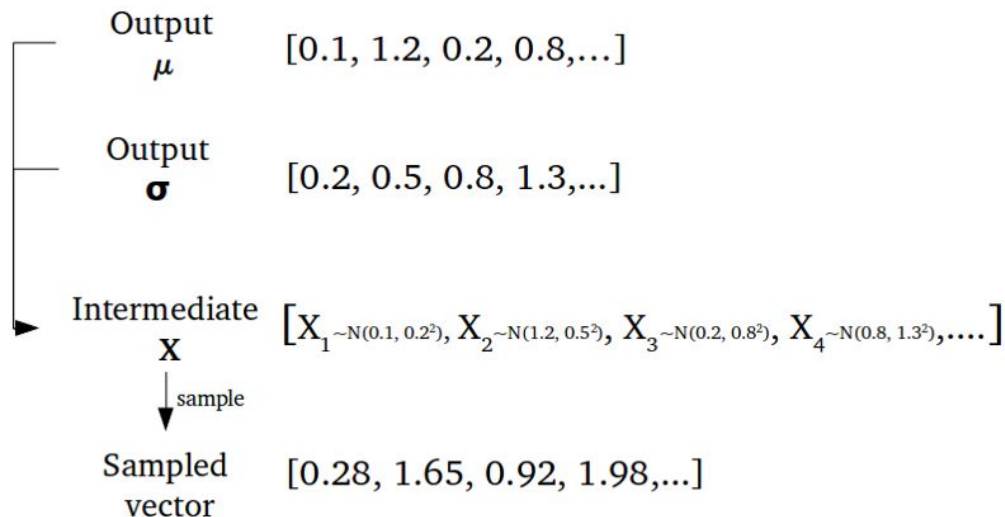
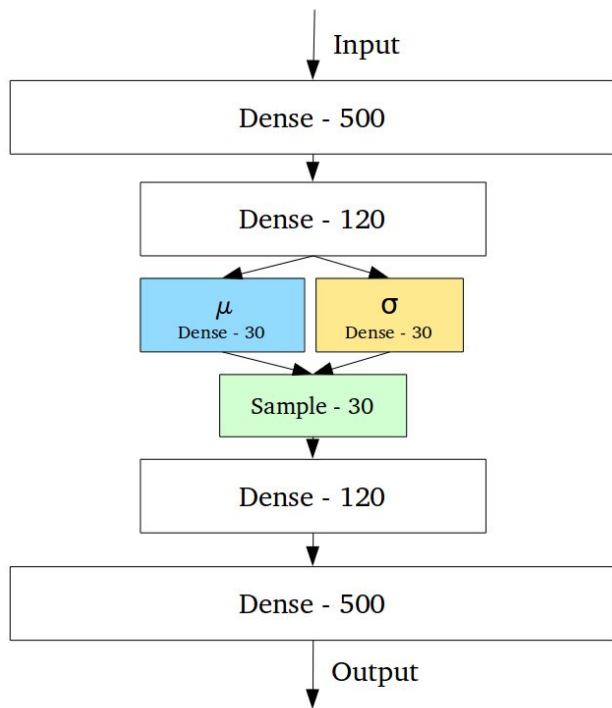
## ● Variational Autoencoders

- *Latent space* (prostor naučenih/skrivenih *encoding-a*) se modeluje kao **kontinualan**
- Umesto jednog vektora, izlaz enkodera će biti dva vektora:
  - Vektor aritmetičkih sredina
  - Vektor standardnih devijacija.





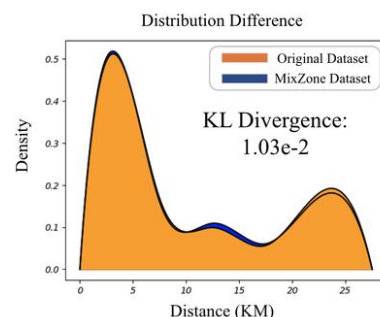
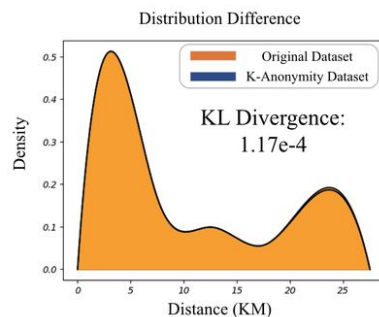
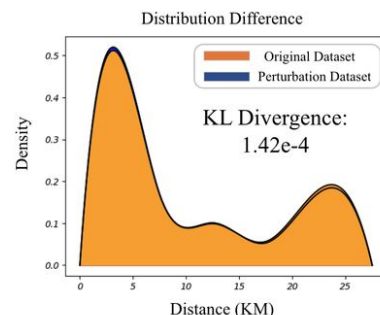
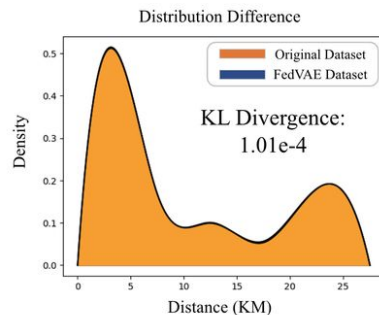
# Variational Autoencoders





## Variational Autoencoders

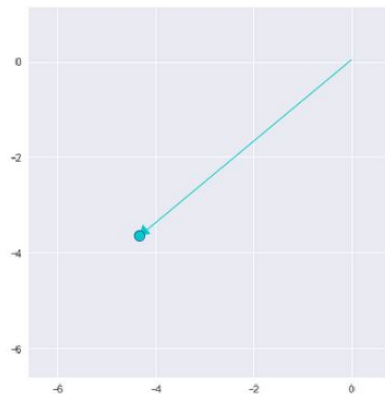
- Pošto su u pitanju slučajne promenljive, svaki put će *encoding* biti malo drugačiji
- Kao *loss* funkcija se koristi KL divergence:
  - Meri koliko dve distribucije divergiraju jedna od druge



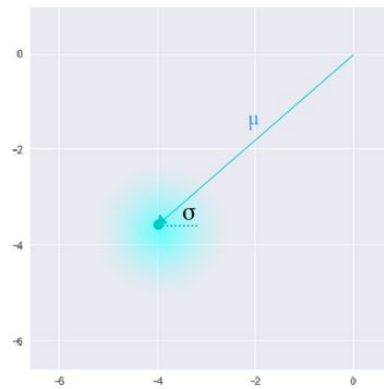


## Variational Autoencoders

- VAE će generisati vrednosti za sve primere (oko sredine gledajući vrednost standardne devijacije)



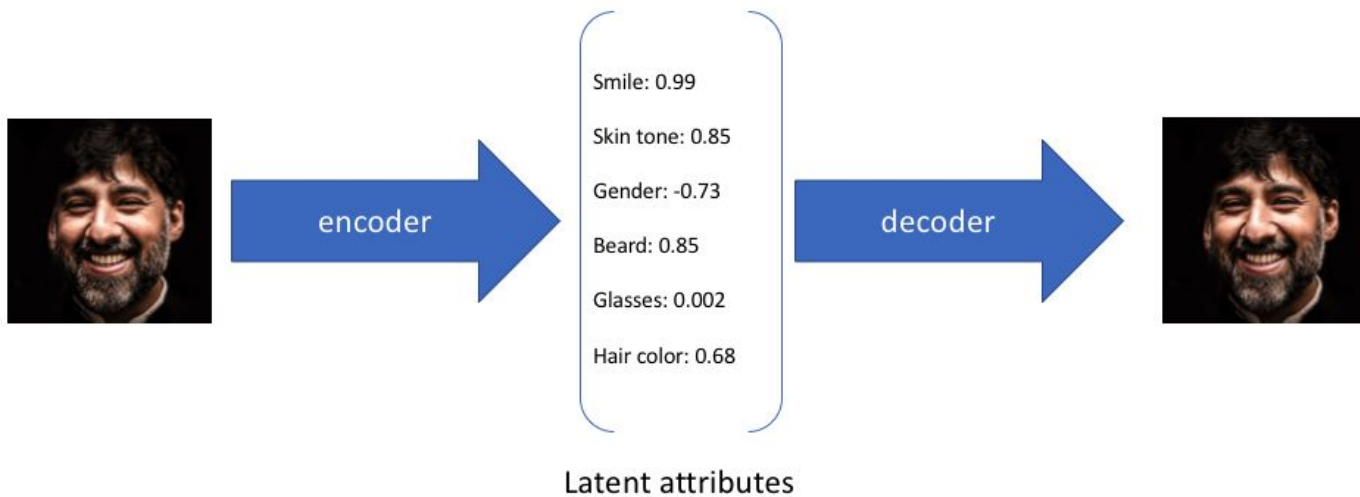
Standard Autoencoder  
(direct encoding coordinates)



Variational Autoencoder  
( $\mu$  and  $\sigma$  initialize a probability distribution)

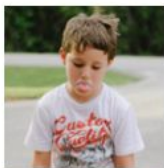


## *Variational Autoencoders*

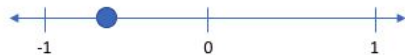




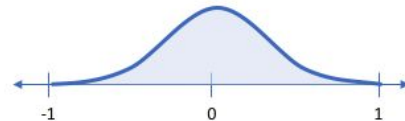
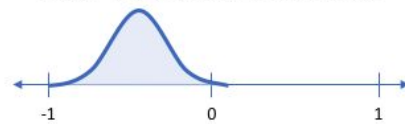
# Variational Autoencoders



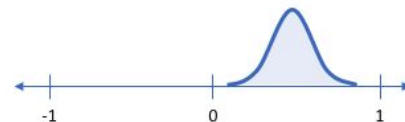
Smile (discrete value)



Smile (probability distribution)

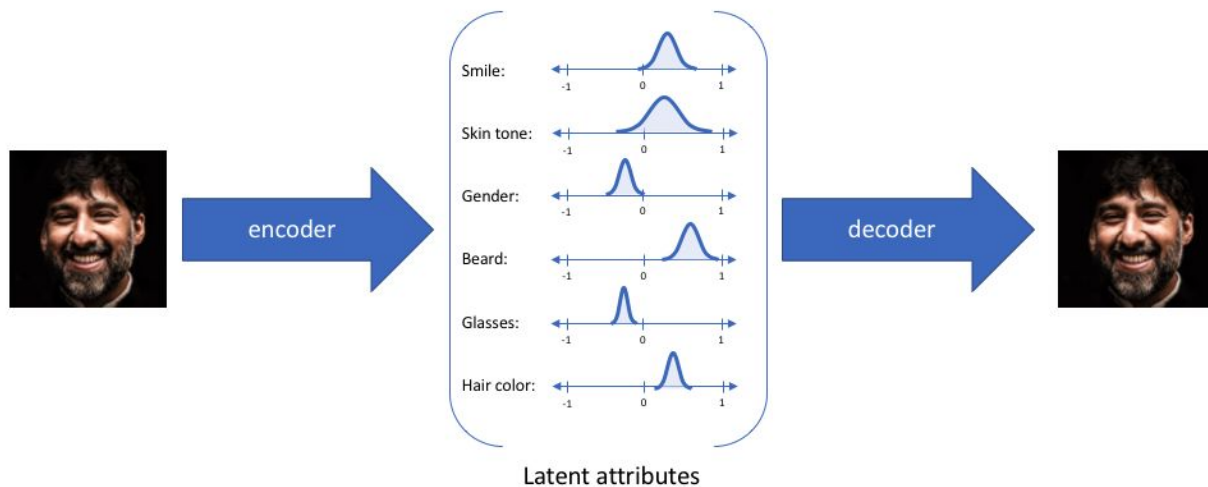


vs.





# Variational Autoencoders

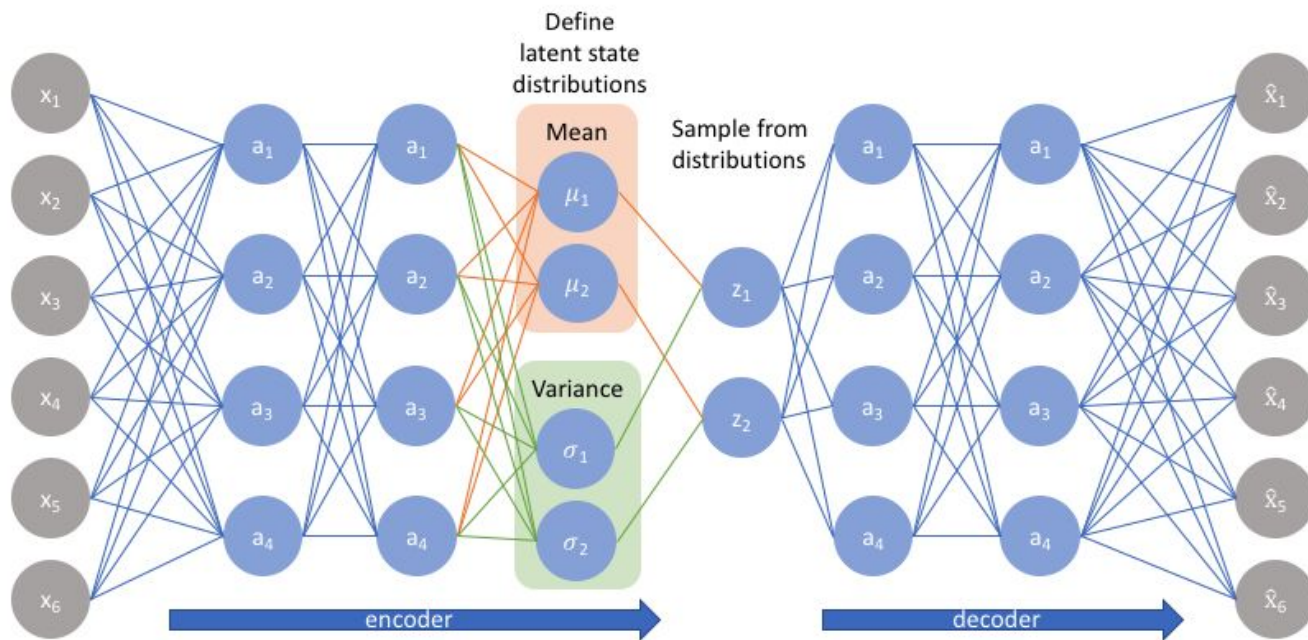


OpenAI Glow





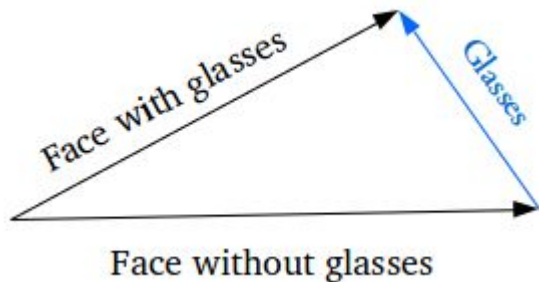
## Variational Autoencoders



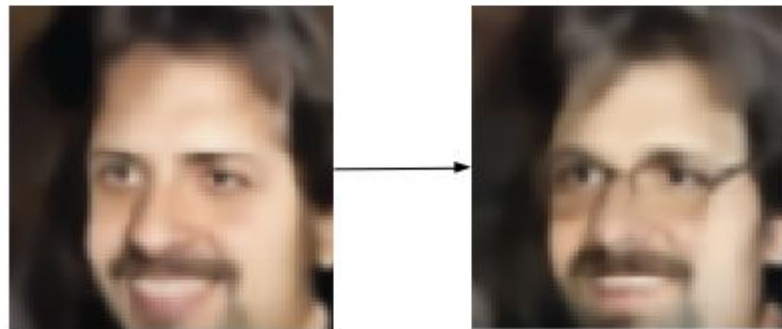
Enkoder se često naziva *recognition model*, a dekoder *generative model*.

## ● Variational Autoencoders

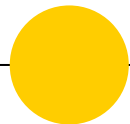
- Omogućava vektorsku aritmetiku nad *encoding* vektorima:



1. Naći primerak sa naočarima i primerak bez naočara i naći njihovu razliku
2. Sabrati dobijenu razliku (naočare) sa bilo kojim drugim licem i propustiti kroz dekođer
3. Rezultat:



# *Generative Adversarial Networks*





## **Generative Adversarial Networks**

- Problem dosadašnjih generativnih modela je u tome što nemaju mogućnost aproksimacije kompleksnih probabilističkih funkcija (u teoriji moguće, u praksi ne):
  - Potreban je jako veliki broj distribucija da bi se sve aproksimiralo
- Izlaz je bio nejasan (eng. *blurry*) zbog samog načina kodiranja:
  - VAE radi sa malim skupom osobina.

# ● Generative Adversarial Networks

- *Generative Adversarial Networks (GAN)* imaju drugačiji pristup:
  - Ideja je slična:
    - Generisanje primera podataka
  - Koristi se nasumičan šum na ulazu

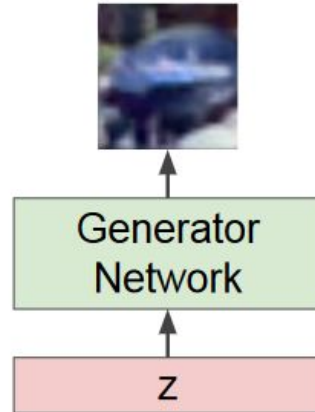
Output: Sample from training distribution



Generator Network

Input: Random noise

z





## ***Generative Adversarial Networks***

- GAN se sastoji iz dve neuronske mreže:
  - Generator
    - Na osnovu šuma generiše izlaz
  - Diskriminator
    - Na osnovu skupa podataka i izlaza generatora daje informaciju da li je izlaz generatora **fake** ili ne



## ***Generative Adversarial Networks***

- Ove dve neuronske mreže se takmiče (*adversarials*) jedna protiv druge
  - Generator pokušava da izgeneriše izlaz koji će moći da prođe kao dobar (da prevari diskriminator)
  - Diskriminator pokušava da postane pametniji i da ne dozvoli generatoru da ga prevari

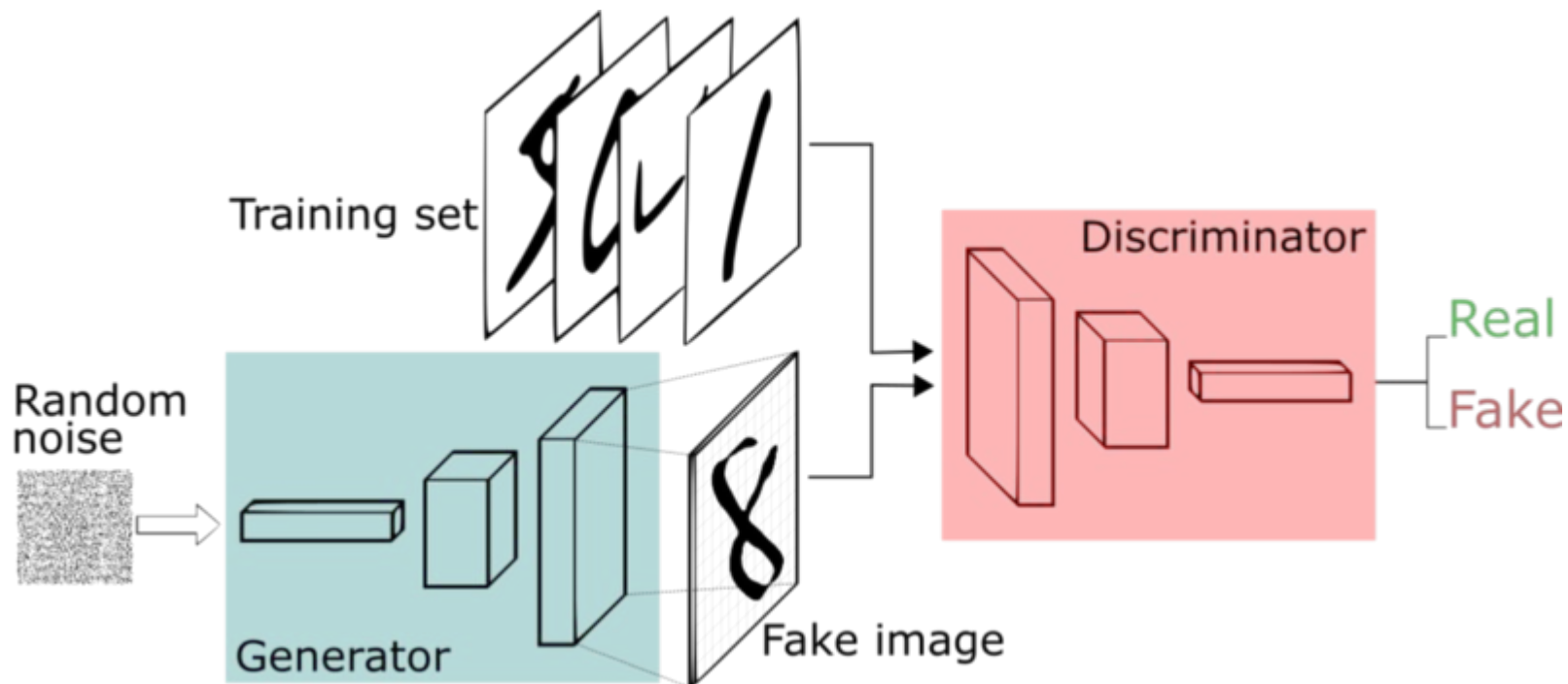


imgflip.com



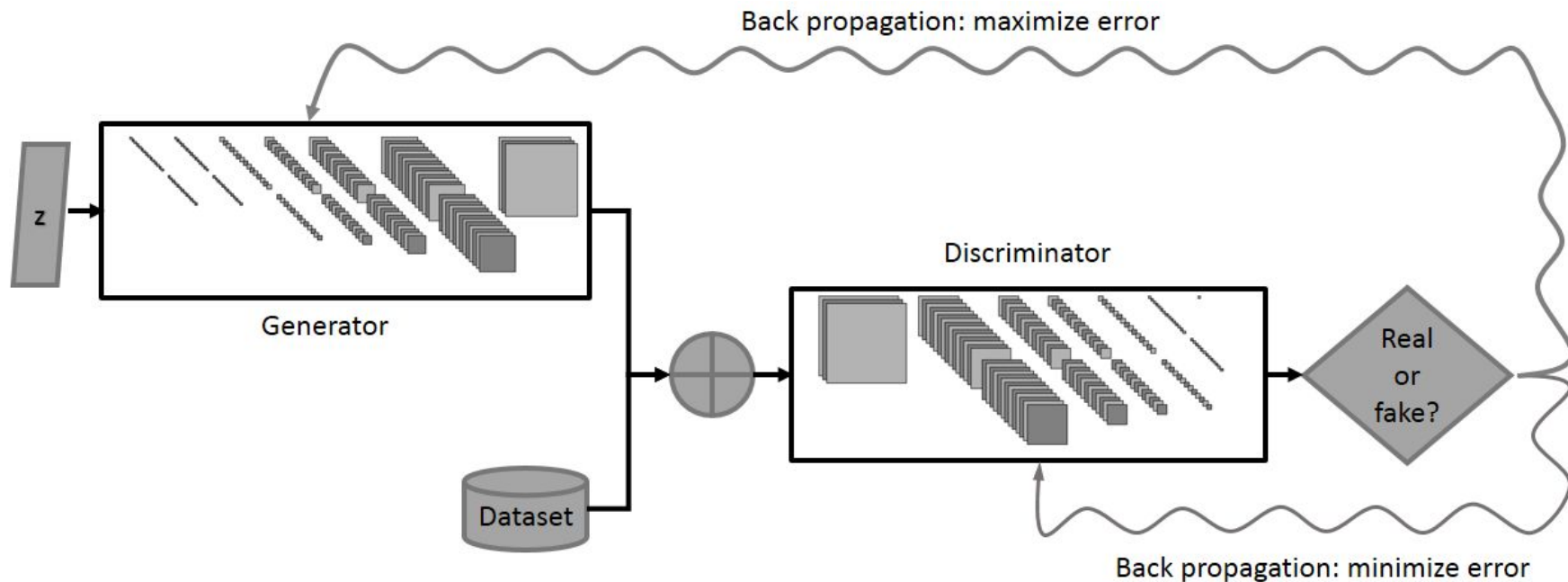


## Generative Adversarial Networks





# Generative Adversarial Networks



# ● Generative Adversarial Networks

- GAN u početku razvoja



## GAN output in paper



## Your GAN output



# Generative Adversarial Networks

- GAN danas
  - Generisanje slika visoke rezolucije



DCGAN  
11/2015



EBGAN-PT  
9/2016



BEGAN  
3/2017  
128 x 128



Progressive GAN  
10/2017  
1024 x 1024





# Generative Adversarial Networks

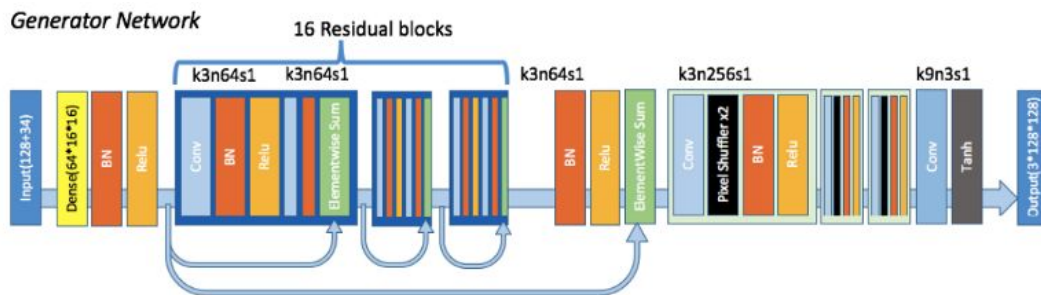


Figure 3: Generator Architecture

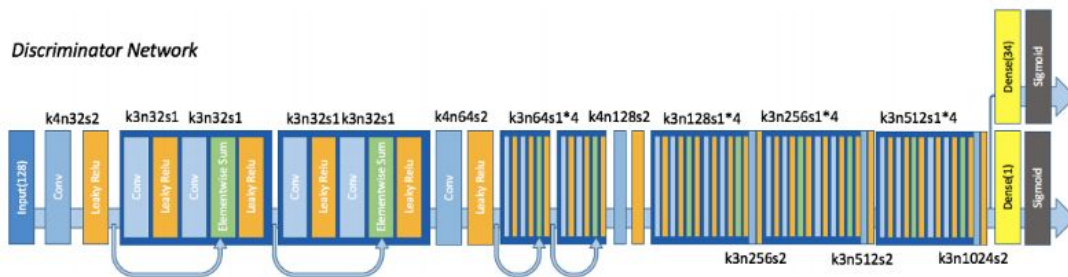
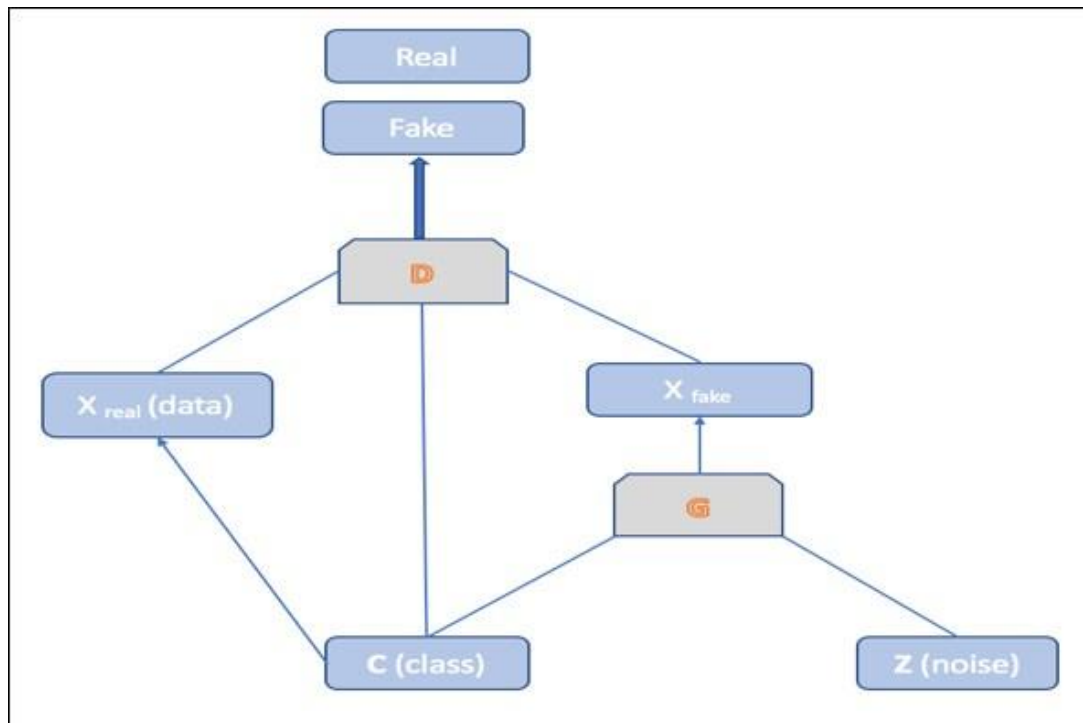


Figure 4: Discriminator Architecture

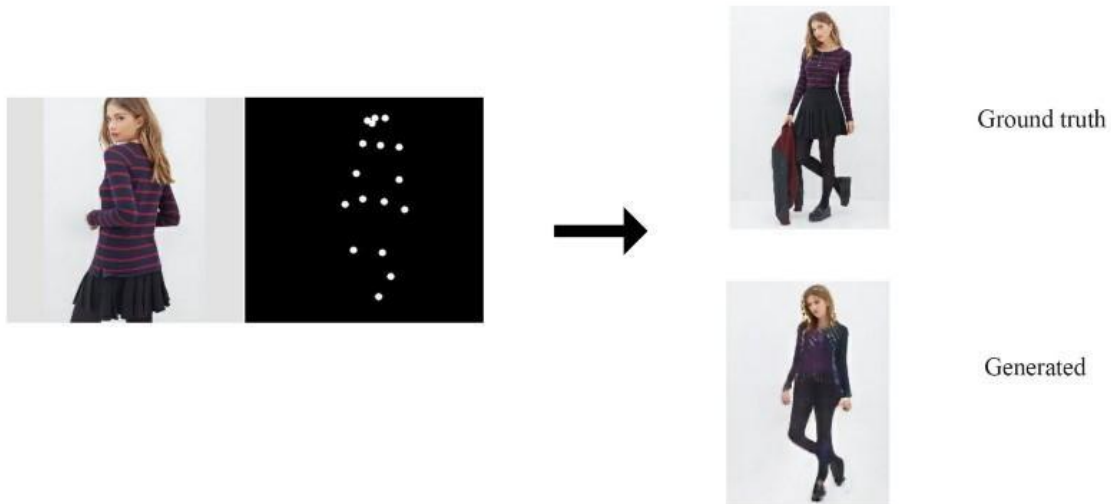
# Generative Adversarial Networks

- *Conditional GAN*
  - Primeniti i labelle kako bi uslovili kako će generisani izlaz izgledati:
    - Druga *loss* funkcija koja više kažnjava grešku u labeli.



# ● Generative Adversarial Networks

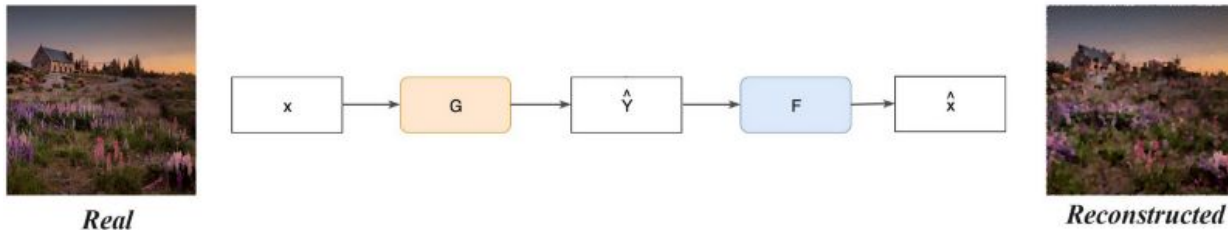
- Pose Guided Person Image Generation





# Generative Adversarial Networks

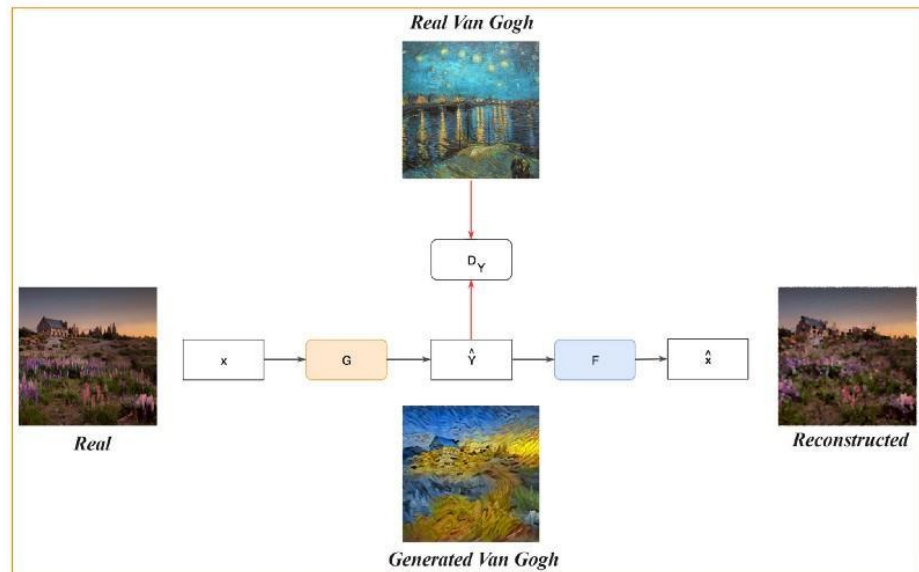
- *Cycle GAN*
  - Ideja je da proces svedemo na *image reconstruction*





# Generative Adversarial Networks

- *Cycle GAN*
  - Umesto da vršimo rekonstrukciju, diskriminator će navoditi generator da generiše određeni stil



Jedan deo mreže se intuitivno koristi za čuvanje osobina slike, a diskriminator će uticati da slika poprimi određeni stil.

Zebras ↔ Horses

Monet ↗



Monet -



zebra → horse

Winter



winter



photo -



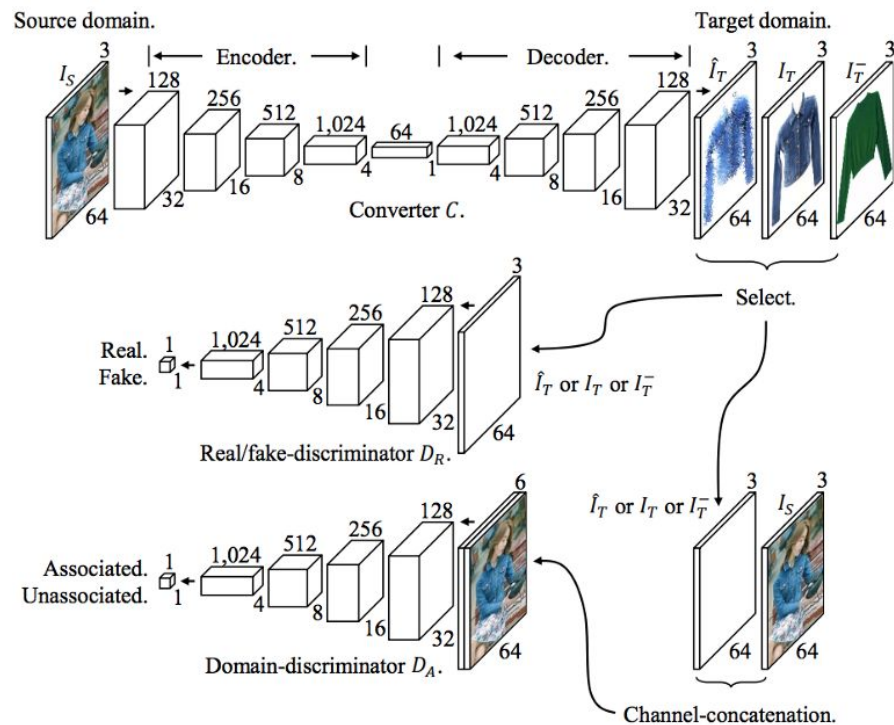
horse → zebra



summer

# PixelDTGAN

Example results on LOC  
More results will be added

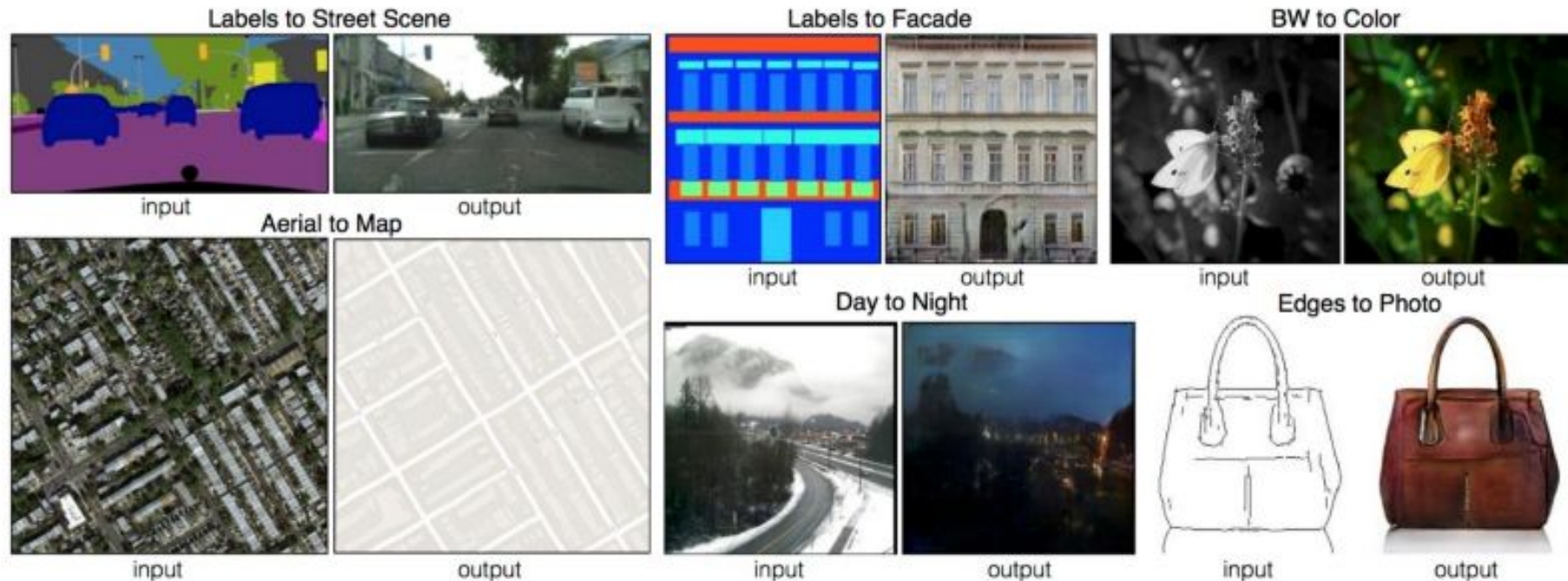


a similar dataset (bottom)



Fig. 2. Whole architecture for pixel-level domain transfer.

## *Pix2pix HD* - obrnut postupak u odnosu na semantičku segmentaciju slike





## Generative Adversarial Networks

- Image-to-Image Demo



## *StackGAN - Text to image*



## StackGAN - Text to image

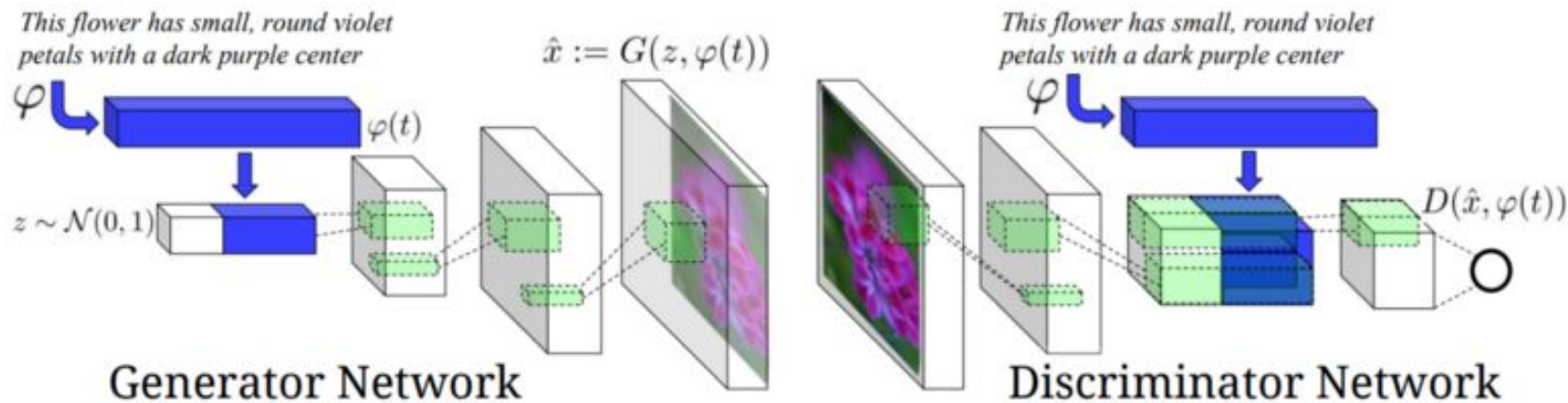
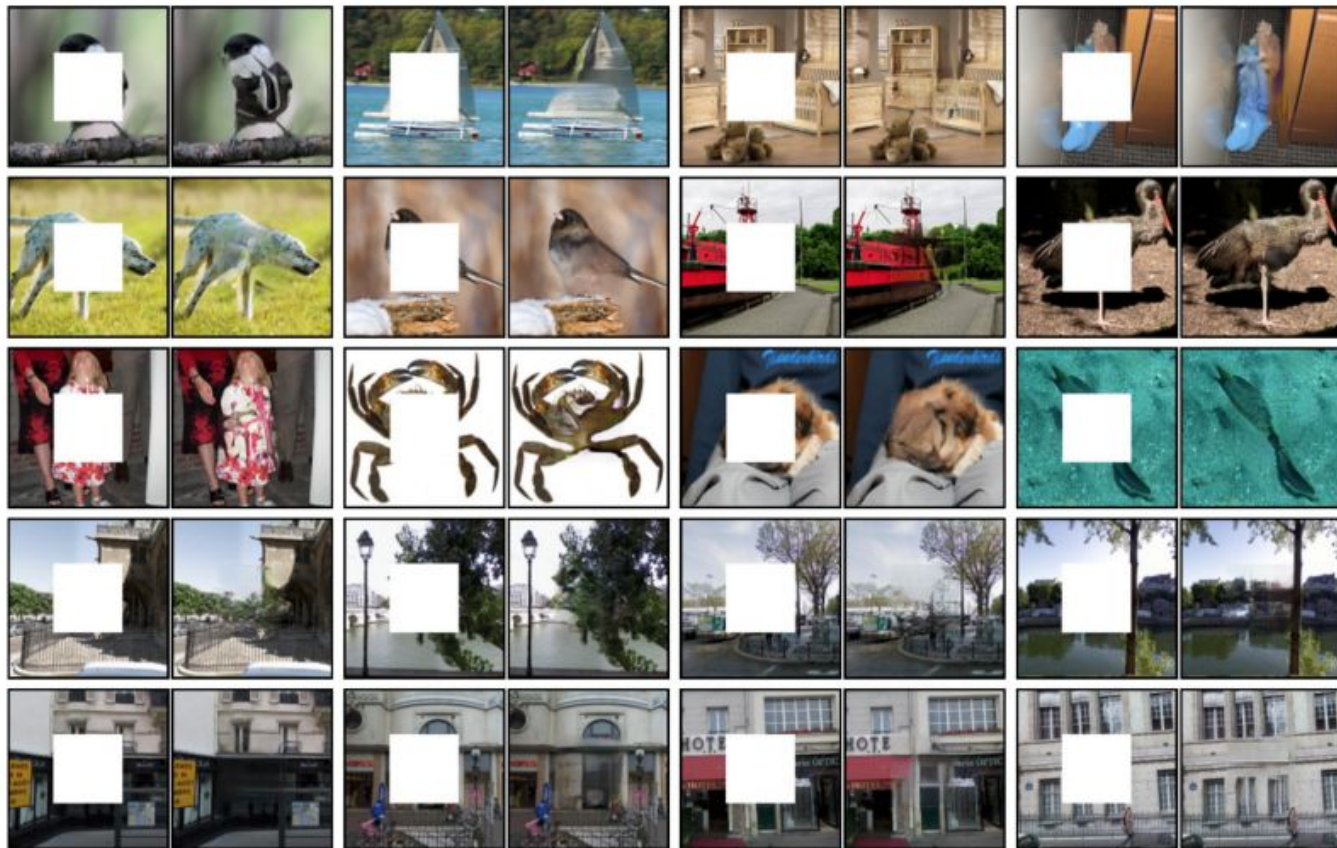


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding  $\varphi(t)$  is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

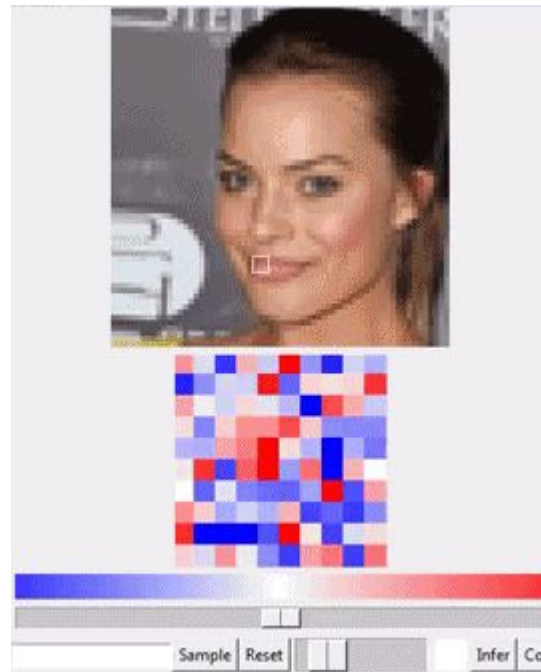


## *Image inpainting*



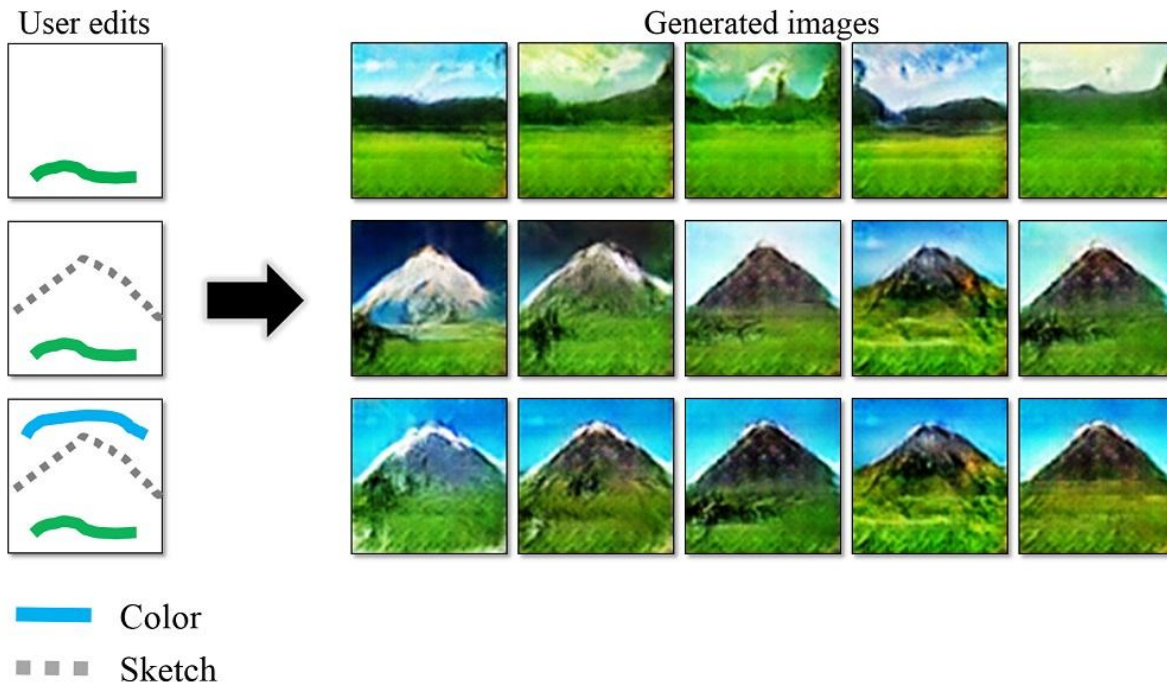
# ● Generative Adversarial Networks

- Neural Photo Editor



# Generative Adversarial Networks

- Sketchy GAN



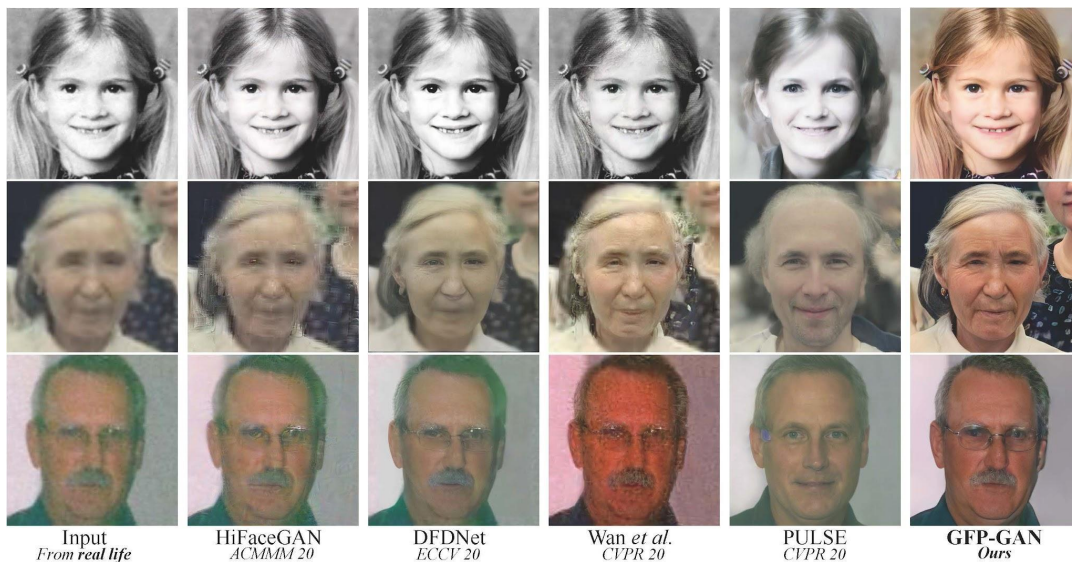


## ***Generative Adversarial Networks***

- *3D Object generation*

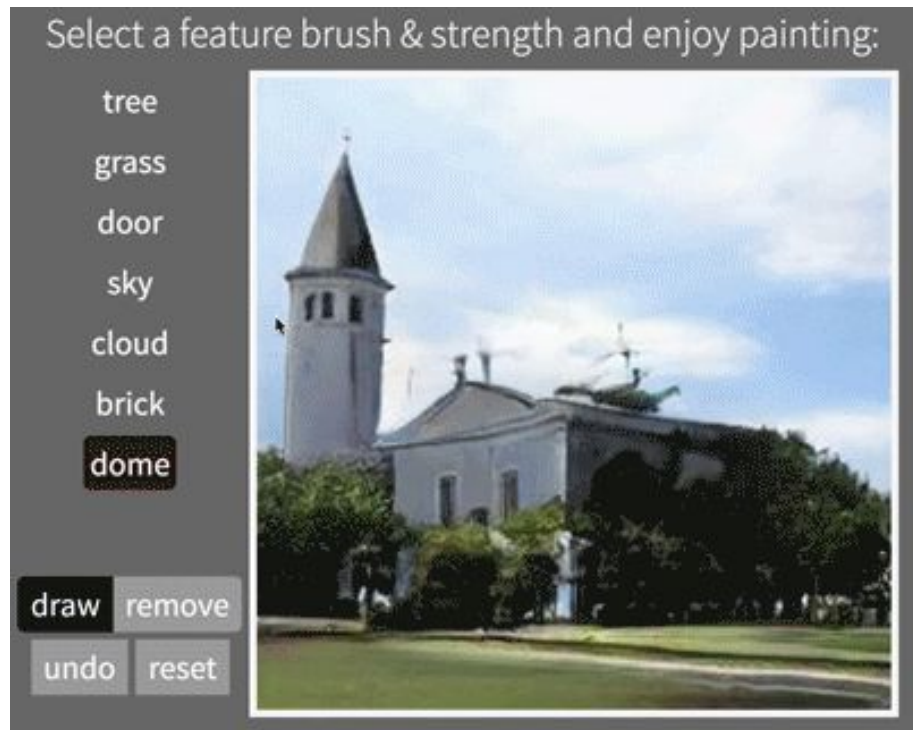
# Generative Adversarial Networks

- CFPGAN - Face Restoration

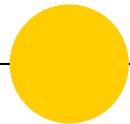


# ● Generative Adversarial Networks

- GAN Dissection



# Dodatno čitanje





## Dodatno čitanje

---

- *Auto-Encoding Variational Bayes*
- *Stanford: Generative Models*
- *Building Autoencoders in Keras*
- *Intuitively Understanding Variational Autoencoders*
- *Variational autoencoders.*
- *OpenAI Research: Generative models*





## **Dodatno čitanje**

---

- *[gans-awesome-applications](#)*
- *[GAN - Some cool applications of GAN](#)*
- *[GAN Playground](#)*
- *[Demystifying Generative Adversarial Networks](#)*
- *[CS236G Generative Adversarial Networks \(GANs\)](#)*

# Hvala na pažnji!

Pitanja?