Rekonstrukcija parcijalnih skica pomoću CVAE-bazirane duboke mreže

Ivan Cvrk ivan.cvrk@fer.hr
Davor Najev davor.najev@fer.hr
Antonio Škara antonio.skara@fer.hr
Toni Vanjak toni.vanjak@fer.hr

Abstract

U ovom radu predstavljamo arhitekturu namijenjenu generiranju potpunih skica na temelju djelomičnih slika. Nasuprot pristupima temeljenim na potezima (stroke-based), kao što je SketchRNN, nudimo alternativni pristup temeljen na pikselima (pixel-based) koji bi trebao omogućiti jednostavno i izravno učenje. Zabilježili smo zadovoljavajuće rezultate u upotpunjavanju slika.

1 Zadatak

Cilj je pomoću dubokog modela napraviti rekonstrukciju binarne slike jednostavnog predmeta. Modela mora biti generativan, tj. različiti prolazi kroz model moraju dati raznovrsne izlaze.

U sljedećim poglavljima bit će opisana naša odabrana arhitekura. Ista će biti evaluarana te će biti prikazani rezultati koji su dobiveni pomoću nje.

2 Teorijska podloga

2.1 Varijacijski autoenkoder (VAE)

Varijacijski autoencoder (VAE) predstavlja generativni probabilistički model koji kombinira duboke neuronske mreže s varijacijskom Bayesovskom inferencijom. Prvi put je formalno predstavljen u radu [Kingma and Welling, 2022] te je danas jedan od temeljnih modela za učenje latentnih reprezentacija i generiranje podataka.

Pretpostavimo da su podaci $\mathbf{x} \in \mathscr{X}$ uzorkovani iz nepoznate distribucije $p_{\text{data}}(\mathbf{x})$. Cilj generativnog modela je naučiti parametarsku distribuciju $p_\theta(\mathbf{x})$ koja aproksimira ovu nepoznatu distribuciju. U VAE-u se uvodi latentna varijabla $\mathbf{z} \in \mathscr{Z}$ te se generativni proces definira kao

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}), \tag{1}$$

gdje je prior nad latentnim prostorom tipično standardna normalna distribucija $p(\mathbf{z}) = \mathscr{N}(\mathbf{0}, \mathbf{I})$.

Izravno maksimiziranje log-vjerojatnosti $p_\theta(\mathbf{x})$ nije računalno izvedivo zbog integracije nad latentnom varijablom:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \, d\mathbf{z}. \tag{2}$$

Kako bi se taj problem zaobišao, uvodi se aproksimacijska (varijacijska) posteriorna distribucija $q_\phi(\mathbf{z} \mid \mathbf{x})$, koja se realizira pomoću koderske neuronske mreže. Primjenom Jensenove nejednakosti dobiva

se donja granica na log-vjerojatnost, tzv. *Evidence Lower Bound* (ELBO):

$$\log p_\theta(\mathbf{x}) \geq E_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\right). \tag{3}$$

Maksimizacija ELBO funkcije ekvivalentna je istovremenoj:

- minimizaciji rekonstrukcijske pogreške (log-vjerojatnosti dekodera),
- regularizaciji latentnog prostora tako da se posterior približava prioru.

Encoder proizvodi parametre distribucije $q_\phi(\mathbf{z} \mid \mathbf{x})$, najčešće srednju vrijednost $\mu_\phi(\mathbf{x})$ i log-varijancu $\log \sigma^2_\phi(\mathbf{x})$, dok se uzorkovanje omogućuje pomoću tzv. *reparameterization trick*:

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

Ova konstrukcija omogućuje treniranje cijelog modela gradijentnim metodama.

2.2 Uvjetni varijacijski autoenkoder (CVAE)

Uvjetni varijacijski autoenkoder (CVAE) predstavlja proširenje VAE modela u kojem se generiranje podataka uvjetuje dodatnom varijablom $\mathbf{c}$. CVAE omogućuje modeliranje uvjetnih distribucija oblika

$$p_\theta(\mathbf{x} \mid \mathbf{c}), \tag{5}$$

što je ključno u zadacima poput inpaintinga, super-rezolucije i rekonstrukcije djelomičnih ulaza.

U kontekstu ovog rada, uvjet $\mathbf{c}$ predstavlja djelomično nacrtanu sliku, dok je cilj rekonstruirati puni uzorak slike $\mathbf{x}$. Generativni model se definira kao

$$p_\theta(\mathbf{x}, \mathbf{z} \mid \mathbf{c}) = p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c}) p(\mathbf{z} \mid \mathbf{c}), \tag{6}$$

pri čemu se u praksi često pretpostavlja da je $p(\mathbf{z} \mid \mathbf{c}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ radi jednostavnosti.

Varijacijska posteriorna distribucija poprima oblik

$$q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}), \tag{7}$$

te se također parametrizira koderskom mrežom koja prima i ciljnu sliku i uvjet.

Analogno standardnom VAE-u, derivira se ELBO za uvjetni slučaj [Zhang et al., 2021]:

$$\log p_\theta(\mathbf{x} \mid \mathbf{c}) \geq E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{c})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c})\right] \tag{8}$$

$$- \mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}) \,\|\, p(\mathbf{z}, \mathbf{c})\right). \tag{9}$$

Tijekom treniranja, encoder koristi i $\mathbf{x}$ i $\mathbf{c}$ kako bi naučio informativnu latentnu reprezentaciju. Tijekom generiranja (inferencije), koristi se samo uvjet $\mathbf{c}$, a latentna varijabla se uzorkuje iz priora. Dekoder tada generira distribuciju nad punim slikama konzistentnim s danim djelomičnim ulazom.

Ovakva formulacija omogućuje modelu da nauči višestruka moguća rješenja (multimodalnost), što je posebno važno u zadacima gdje je rekonstrukcija inherentno nejednoznačna.

2.3 Bernoullijeva distribucija kao distribucija dekodera

Za binarne ili normalizirane slike čije su vrijednosti u intervalu $[0,1]$, uobičajeno je modelirati dekoderovu izlaznu distribuciju kao Bernoullijevu distribuciju. U tom slučaju, dekoder proizvodi parametre

$$\pi_\theta(\mathbf{z},\mathbf{c}) \in [0,1]^D, \tag{10}$$

gdje je D broj piksela, a π_i predstavlja vjerojatnost da je piksel $x_i = 1$.

Uvjetna vjerojatnost slike dana latentnoj varijabli i uvjetu tada je

$$p_\theta(\mathbf{x} \mid \mathbf{z},\mathbf{c}) = \prod_{i=1}^{D} \text{Bernoulli}(x_i \mid \pi_i). \tag{11}$$

Ovdje dolazi do problema jer diskretna Bernoullijeva distribucija nema dobro definiranu normalizacijsku konstantu [Loaiza-Ganem and Cunningham, 2019]. Kao rješenje tog problema koristi se **kontinuirana Bernoullijeva distribucija**, koja je zadana sljedećom formulom:

$$p(x \mid \lambda) = \begin{cases} \dfrac{\lambda^x (1-\lambda)^{1-x}}{C(\lambda)}, & x \in [0,1], \\ 0, & \text{inače}, \end{cases} \tag{12}$$

gdje je normalizacijska konstanta $C(\lambda)$ definirana kao:

$$C(\lambda) = \begin{cases} \dfrac{2\lambda - 1}{\ln\lambda - \ln(1-\lambda)}, & \lambda \neq \frac{1}{2}, \\ 2, & \lambda = \frac{1}{2}. \end{cases} \tag{13}$$

Ova distribucija omogućuje glatko definiranje vjerojatnosti za sve vrijednosti x u intervalu $[0,1]$, čime se izbjegava problem neodređene normalizacijske konstante kod diskretne verzije.

3 Metodologija

Za ostvarenje rješenja korišten je generativni model temeljen na uvjetnom varijacijskom autoenkoderu (Conditional Variational Autoencoder, CVAE) za generiranje i dovršavanje skica.

Za razliku od pristupa koji skice promatraju kao niz poteza olovke, ovdje se koristi *pixel-based* reprezentacija, pri čemu je svaka skica predstavljena kao dvodimenzionalna binarna slika. Takav pristup omogućuje izravnu primjenu konvolucijskih neuronskih mreža te pojednostavljuje proces obrade podataka, budući da nije potrebno eksplicitno modelirati redoslijed poteza ili stanje olovke.

3.1 Arhitektura dekodera i enkodera

Dekoder i enkoder u modelu temelje se na konvolucijskim neuronskim mrežama. Naša implementacija sadrži dva enkodera: jedan za modeliranje priorne distribucije $p(z \mid c)$ i jedan za posteriornu distribuciju $p(z \mid x,c)$. Enkoderi su modelirani modulom GaussianEncoder, a dekoder modulom BernoulliDecoder

Gaussian Encoder: Enkoder prima skicu (djelomičnu ili kombinaciju potpune i djelomične) i projektira je u latentni prostor. Arhitektura enkodera uključuje:

- Konvolucijske slojeve: niz od tri bloka konvolucijskih slojeva s LeakyReLU aktivacijom. Svaki blok sadrži 2 konvolucijska sloja i nakon prvih dva bloka prosječno spajanje (average pooling) kako bi se smanjila dimenzionalnost.
- Flatten i linearni slojevi: izlaz konvolucijskih slojeva se spljošti i prolazi kroz dva linearna sloja koji generiraju μ i $\log \sigma^2$ latentnog prostora.

Bernoulli Decoder: Dekoder prima latentni vektor z i djelomičnu skicu y te generira rekonstruiranu sliku $\hat{x}$. Arhitektura dekodera uključuje:

- Linearni sloj: latentni vektor z se projicira u početni prostorni feature map dimenzija channels $\times$ height $\times$ width.
- Konvolucijske transpozicije (ConvTranspose2D): nekoliko slojeva koji postupno povećavaju prostornu dimenziju feature mapa, uz LeakyReLU aktivacije.
- Kombinacija s djelomičnom skicom: latentni vektor je spojen s djelomičnom skicom kao dodatni kanal, čime se omogućuje uvjetovano generiranje.
- Izlazni sloj: završni konvolucijski sloj proizvodi vjerojatnosti po pikselu, odnosno generira binarnu sliku.

Ova arhitektura omogućuje modelu da:

1. Uči prostorne značajke skice kroz konvolucijske slojeve.
2. Mapira skicu u latentni prostor i uzorkuje latentni vektor z diferencijabilno.
3. Generira potpunu sliku iz latentnog vektora uz uvjetovanje na djelomičnoj skici.

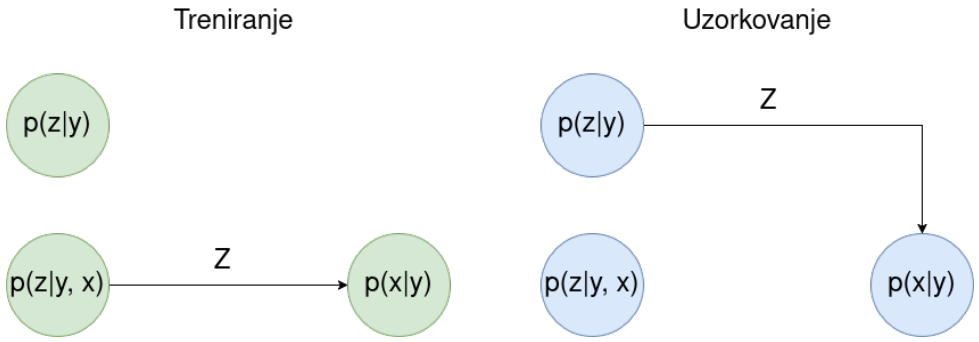


Figure 1: Dijagram Uzorkovanja i Treniranja CVAE

Kao što vidimo na

3.2 Funkcija gubitka

Učenje se provodi optimizacijom *varijacijskog donjeg ograničenja* (ELBO), koje se sastoji od rekonstrukcijskog gubitka i Kullback–Leiblerove divergencije između posteriorne i prior distribucije:

$$\mathscr{L}_{\text{ELBO}} = E_{q_\phi(z|x,y)}\big[\log p_\theta(x \mid z,y)\big] - \text{KL}\big(q_\phi(z \mid x,y) \,||\, p_\theta(z \mid y)\big), \tag{14}$$

gdje prvi član predstavlja gubitak rekonstrukcije, a drugi član služi kao regularizacija latentnog prostora.

3.3 Način treniranja

Naš model je treniran na slikama veličine 64x64 koje su stvorene pretprocesiranjem odabranih klasa predmeta iz "Quick, Draw!" skupa podataka.

Kod našeg modela su ulazi i izlazi reprezentirani pompoću piksela. Ovaj pristup ima nekoliko važnih prednosti:

- Arhitektura je jednostavna i intuitivna, a treniranje stabilno u usporedbi sa sekvencijalnim modelima.
- Dobro se prilagođava zadacima poput dovršavanja skice, uklanjanja šuma ili rekonstrukcije nedostajućih dijelova slike, jer model izravno uči prostorne odnose među pikselima.

Međutim, ovaj pristup ima i ograničenja:

- Skica se promatra isključivo kao slika, pa model ne uči eksplicitnu informaciju o redoslijedu poteza, brzini crtanja ili stilu linije.
- Generirani crteži mogu izgledati manje prirodno ili “mehanički” u usporedbi s ljudskim crtanjem.

3.4 Alternativni pristupi treniranju

Alternativa pristupu temeljenom na pikselima je pristup temeljen na sekvenci poteza olovkom. U tom slučaju je ulaz u model niz poteza, odnosno odmaka olovke. Skup podataka "Quick, Draw!" kojeg smo mi koristili za treniranje zapravo dostavlja podatke u ovom tipu, za učenje našeg modela smo ga morali prvo predprocesirati.

Kako su ulazni podatci proizvolje duljine, potrebno je koristiti neku vrstu rekuretne mreže (RNN-a) za obradu. Vrlo poznati model baš za ovu konkretnu namjenu SketchRNN upravo radi na takav način, on je ustvari VAE kojemu su koder i dekoder dvosmjerni LSTM-ovi [Ha and Eck, 2017]. Takvi modeli bolje hvataju dinamiku ljudskog crtanja i stil linije, ali su složeniji za implementaciju i treniranje, a i u principu je teže naći primjerene skupove podataka za treniranje.

4 Kvalitativna i kvantitativna analiza

4.1 Usporedba gubitaka u odnosu na različite hiperparametre

Promotrit ćemo pogreške na validacijskom skupu u ovisnosti o tome koliko smo slika koristili za treniranje modela. Budući da je treniranje modela dugotrajan proces, nadali smo se postići dobre rezultate s manjim brojem slika.

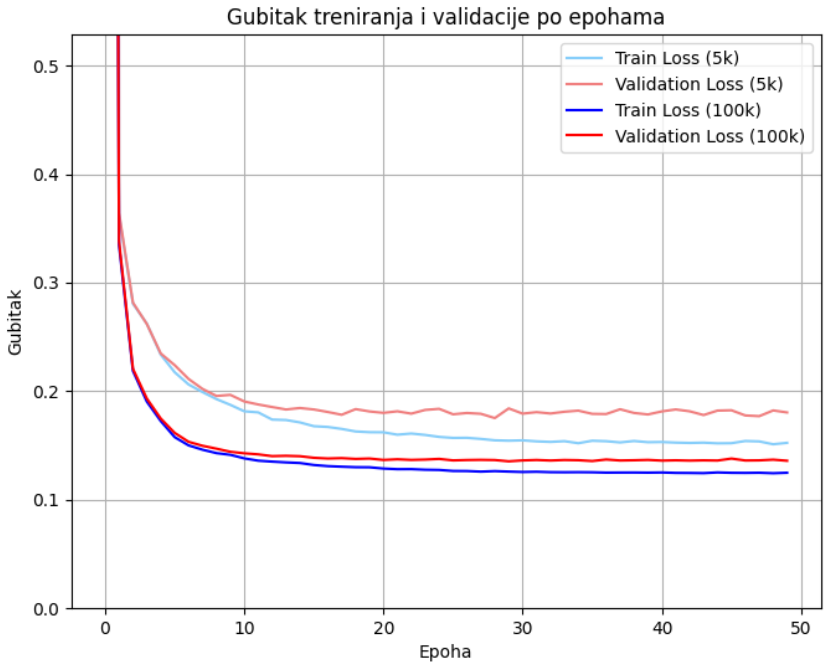
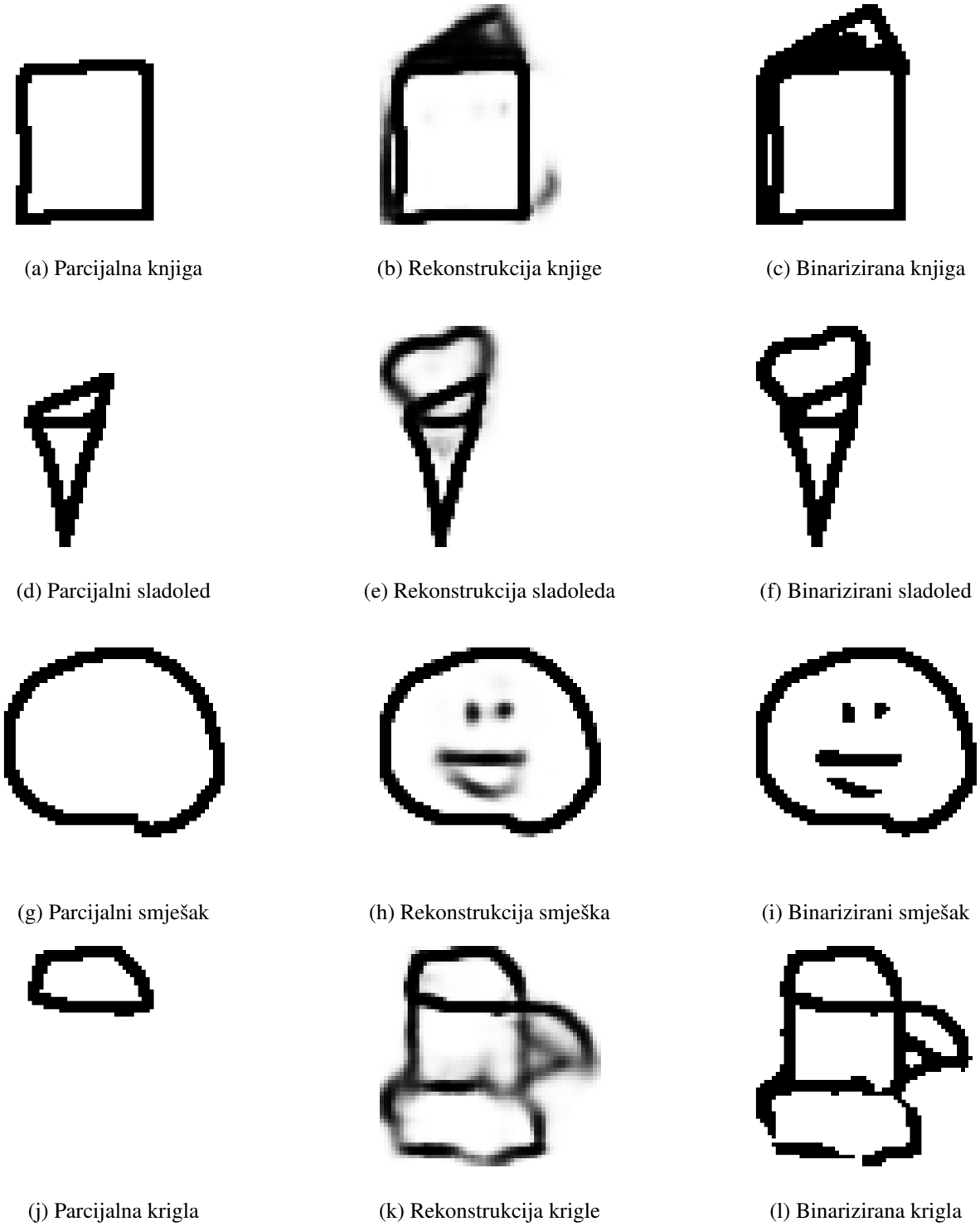


Figure 2: Usporedba gubitaka za različite veličine skupa za treniranje

Na figuri 2 možemo vidjeti kako se ponašaju validacijski gubitci za različite veličine skupova za treniranje. Vidljivo je da je razlika u gubitcima između 5 tisuća i 100 tisuća slika disproporcionalno mala. To nam govori da je model zaključio sve što je mogao o distribuciji ulaznik slika s relativno malim brojem primjera. Najvjerojatniji razlog tome je niska kvaliteta podataka u skup podataka - slike su izrađivali ljudi na internetu potpuno slobodno pa ima mnogo slika loše kvalitete. Unatoč tome, ručnom kvalitativnom analziom može se ustvrditi da veći broj slika daje primjetno bolje slike što nije neočekivano.

4.2 Vizualna usporedba rezultata

Sada ćemo vizualno usporediti završne rezultate. U sljedećim usporedbama koristit ćemo model treniran na 25 tisuća slika s dimenzijom skrivenog sloja 512.



(a) Parcijalna knjiga (b) Rekonstrukcija knjige (c) Binarizirana knjiga

(d) Parcijalni sladoled (e) Rekonstrukcija sladoleda (f) Binarizirani sladoled

(g) Parcijalni smješak (h) Rekonstrukcija smješka (i) Binarizirani smješak

(j) Parcijalna krigla (k) Rekonstrukcija krigle (l) Binarizirana krigla

Figure 3: Primjer rekonstrukcija različitih klasa

Na figuri 3 možemo vidjeti nekoliko primjera skupova ulaznih slika u model (tj. nepotpune slike koje želimo upotpuniti), izlazne slike (tj. izlazne vjerojatnosti) te binariziranu izlaznu sliku s fiksnim pragom od 0,5. Vidljivo je da model ima ispravnu ideju o tome gdje i kako nadopuniti podatke koji nedostaju.

Kako je model generativan, možemo dobiti različite rekonstrukcije uzorkovanjem različitih z-ova iz posteriorne distibucije enkodera. Na figuri **??** možemo vidjeti mnogo različitih rekonstrukcija sladoleda ako damo modelu samo jednak kornet kao ulaz.

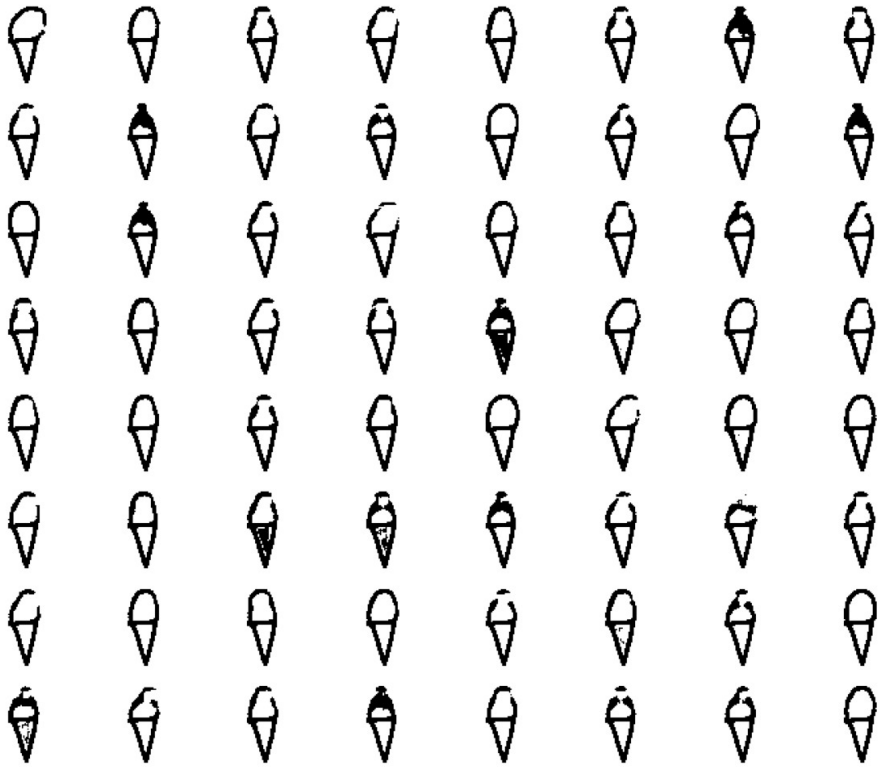


Figure 4: Primjer mnogo različitih primjera generiranih sladoleda

5 Repozitorij

Repozitorij s implementacijom ovog modela kao i sve što je potrebno za njegovo treniranje se nalazi na sljedećoj web lokaciji: http://github.com/spinzed/drawing-reconstruction

References

[Ha and Eck, 2017] Ha, D. and Eck, D. (2017). A neural representation of sketch drawings. *CoRR*, abs/1704.03477.

[Kingma and Welling, 2022] Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.

[Loaiza-Ganem and Cunningham, 2019] Loaiza-Ganem, G. and Cunningham, J. P. (2019). The continuous bernoulli: fixing a pervasive error in variational autoencoders.

[Zhang et al., 2021] Zhang, C., Barbano, R., and Jin, B. (2021). Conditional variational autoencoder for learned image reconstruction. *CoRR*, abs/2110.11681.