Prenošenje stila glazbe korištenjem difuzijskih modela

1st Fran Vučković *FER*
2nd Filip Pankretić *FER*
3rd Dominik Jambrović *FER*
4th Velimir Kovačić *FER*
5th Filip Perković *FER*
6th Luka Glavinić *FER*

I. Uvod

Popularizacijom generativnih modela umjetne inteligencije proteklih godina mogu se vidjeti brojni primjeri njihove primjene na prenošenje stila (engl. *style transfer*) [1]. Glavni cilj ovoga zadatka strojnog učenja je pretvorba postojećih slika tako da izgledaju kao da su naslikane rukom željenog slikara ili animirane u stilu zadanog studija. Kompliciraniji je problem prenošenje stila zvučnih podataka, zbog njihove promjenjivosti u vremenu, ali i poteškoća u stvaranju preciznih opisa različitih stilova. Cilj je prenijeti glazbeni žanr i zvuk zadanog stila na način da je očuvana melodijska i ritamska struktura zadanog isječka.

Difuzijski modeli [2] učinkovit su alat u zadacima s vizualnim generiranjem pa nude obećavajući okvir i za probleme iz audio domene. Ovaj rad nastoji reproducirati rezultate rada *Music Style Transfer with Time-Varying Inversion of Diffusion Models* [3]. Opisat će se prethodni pristupi na čije smo se temelje oslanjali, struktura korištenog modela, njegovo učenje i rezultati te njihova evaluacija po dvije različite objektivne metrike.

II. Pregled postojećih pristupa

Kad se govori o prenošenju stila glazbe, podrazumijeva se podijeliti glazbeni primjerak na njegov sadržaj i njegov stil. Cilj nam je izmijeniti taj stil u neki drugi zadani stil, a da sadržaj primjerka ostane uvelike nepromijenjen. U te se svrhe koriste autoregresijski modeli, generativne suparničke mreže (engl. *Generative Adversarial Networks - GAN*) [4], varijacijski autoenkoderi [5] i razni drugi modeli. U području GAN-ova ističu se modifikacije WaveGAN [6] i CycleGAN [7]. *Score Matching* još je jedna nedavna metoda korištena u generiranju glazbenih zapisa [8], no rezultati se obično fokusiraju na prenošenje samo jedne karakteristike stila i pate od artefakata na spektrogramima.

Neki nedavni radovi poput WaveGrad [9] i DiffWave [10] uspijevaju generirati audio podatke vrlo brzo i kvalitetno pomoću difuzijskih modela. Glavni uzor u tom polju ipak nam je ranije spomenut rad [3] u kojem autori opisuju karakteristike korištene podatkovne reprezentacije, korištenje latentnog difuzijskog modela za učenje i prenošenje zadanog stila glazbe, kao i dodavanje vremenski promjenjivog kodera u arhitekturu kako bi se dodatno unaprijedilo prenošenje stila.

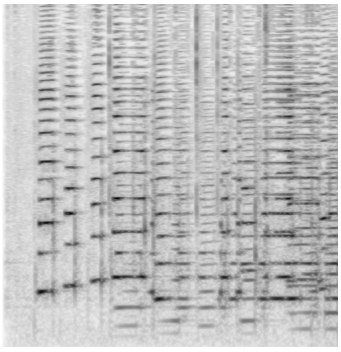
Valja napomenuti i da se glavnina tradicionalnih modela prenošenja stila bavi prebacivanjem pjesama u druge striktno definirane glazbene žanrove, dok su mnogi stilovi na kojima se naš model uči manje instrumentalni. Neki od njih su zvukovi pucketanja plamena, cvrkuta ptice, zvonjave zvona ili otkucaja srca.

III. Opis skupa podataka

Za potrebe provođenja eksperimenata preuzeli smo skup podataka iz repozitorija autora rada [3]. Skup se sastoji od 253 zvučnih zapisa .wav formata koji svaki traje po 5 sekundi. Zapisi su prikupljeni s web-stranice Pixabay te podijeljeni na zapise sadržaja (engl. *content*) i zapise stila (engl. *style*). Nakon podjele, skup podataka sadrži ukupno 179 zapisa sadržaja te 74 zapisa stila. Zapisi sadržaja podijeljeni su u 13 različitih kategorija, dok su zapisi stila podijeljeni na 18 različitih stilova. Pojedini zapisi obuhvaćaju širok spektar instrumenata i stilova kako bismo što podrobnije ispitali kapacitet i uspješnost našeg modela.

Glavna ideja našeg pristupa jest da se model nauči na zvučnim zapisima odabranog stila, npr. *harmonica*, te da se naučeni stil potom prenosi na željene zapise sadržaja.

U svrhu obrade zvučnih podataka koristit ćemo podatkovnu reprezentaciju zvanu mel-spektrogram. Mel-spektrogram služi za vizualni prikaz frekvencijskog sadržaja zvuka u vremenu. Frekvencije na spektrogramu prilagođene su na takozvanu mel skalu koja odgovara ljudskom doživljaju odnosa među pojedinim tonovima. Takvi nam spektrogrami odgovaraju jer omogućuju jednostavan prikaz zvukova u domeni ljudske percepcije. Dodatno, njima možemo izravno manipulirati tijekom postupka prenošenja stila koristeći modele koji rade sa slikama. Kako bismo pretvorili zvučni zapis .wav formata u mel-spektrogram, koristimo *Short Time Fourier Transform* (STFT). Nakon prenošenja stila, dobivene izmijenjene mel-spektrograme pretvaramo natrag u zvučne zapise pomoću Griffin-Lim algoritma.



Slika 1: Primjer jednog mel-spektrograma

IV. Arhitektura

Razvijeni model za prijenos stila glazbe sastoji se od 3 komponente:

- Okosnica Riffusion
- Koder teksta iz CLIP-a
- Modul TVE (*engl. Time - Varying Encoder*)

A. Okosnica Riffusion

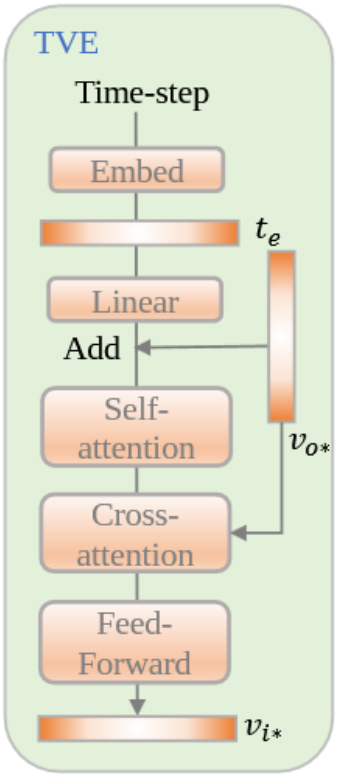
Model Riffusion [11] dizajniran je za rad sa spektrogramima zvuka za zadatke poput generiranja i transformacije glazbe. Sami model spada u latentne difuzijske modele (LDM), a na ulazu odnosno izlazu modela nalaze se mel-spektrogrami. Za zadani spektrogram na ulazu, model radi niz malih promjena (koraka uklanjanja šuma) sve dok (u ovom slučaju) spektrogram ne poprimi željeni stil. U našoj arhitekturi za prijenos stila glazbe, Riffusion se koristi kao okosnica sa zamrznutim parametrima.

B. Koder teksta iz CLIP-a

CLIP (engl. *Contrastive Language–Image Pretraining*) [12] model je strojnog učenja koji povezuje tekst i slike. Može se koristiti za povezivanje slika i pripadnih tekstualnih opisa, *zero-shot* klasifikaciju, stvaranje opisa za slike i slično. Sastoji se od 2 dijela: kodera slika i kodera teksta. Koder slika ugrađuje slikovnu reprezentaciju u prostor ugrađivanja (latentni prostor), a koder teksta ugrađuje tekstualnu reprezentaciju u isti taj prostor ugrađivanja. U našoj arhitekturi za prijenos stila glazbe, koristi se samo predtrenirani koder teksta iz CLIP-a sa zamrznutim parametrima.

C. Modul TVE (engl. Time - Varying Encoder)

Modul TVE ključna je inovacija u procesu prijenosa stila glazbe. Cilj je asocirati vremenski korak t i ugrađivanje znaka "*" sa stilom tijekom učenja, kako bi se kasnije modul mogao koristiti u procesu prijenosa stila glazbe.



Slika 2: Arhitektura modula TVE. Preuzeto iz [3].

Na ulazu, modul prima vremenski korak t i ugrađuje ga u prostor ugrađivanja sinusnim i kosinusnim transformacijama kao t_e. Zatim se vektor t_e provede kroz 3 linearna sloja neuronske mreže uz aktivacijsku funkciju $SiLU$ nakon svakog sloja osim posljednjeg.

$$SiLU(x) = x \cdot \sigma(x) \tag{1}$$

Nakon toga, vektoru t'_e pribraja se ugrađivanje konstantnog znaka "*" v_{o*} dobiveno pomoću kodera teksta iz CLIP-a. Dobiveno ugrađivanje v_0 zatim prolazi kroz sloj pažnje i sloj unakrsne pažnje. Oba sloja pažnje koriste *dropout* iznosa 0.05 i imaju 8 glava pažnje. Jednadžba pažnje općenito je:

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d}}) \cdot V \tag{2}$$

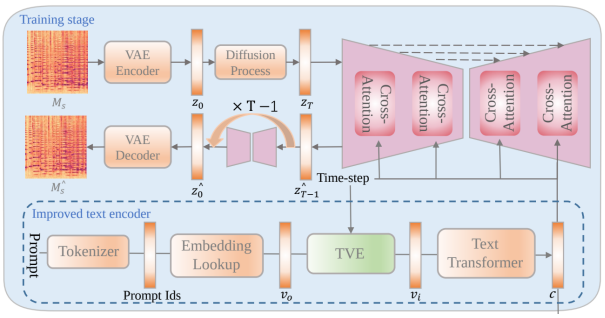
U ovom slučaju, ulaz u sloj pažnje je samo v_0, a ulaz u sloj unakrsne pažnje je v_0 i izlaz prvog sloja pažnje v_1.

$$v_1 = Attention(v_0, v_0, v_0) \tag{3}$$
$$v_i = Attention(v_1, v_0, v_0) \tag{4}$$

Konačno, vektor prolazi kroz sloj *dropout* iznosa 0.05 i još jedan linearni sloj te postaje vektor ugrađivanja v_{i*}.

D. Učenje modela

Suradnju svih dosad opisanih dijelova arhitekture najbolje vidimo u procesu učenja, prikazanog na slici 3.



Slika 3: Učenje modela. Preuzeto iz [3].

Zadani mel-spektrogram stila prolazi kroz koder varijacijskog autoenkodera i postaje latentni vektor z_0. Zatim prolazi kroz difuzijski proces (5) od T koraka i postaje zašumljeni latentni vektor z_T. Broj T nasumično uzorkujemo u svakom koraku učenja.

$$z_t = \sqrt{1-\beta_t^2} \cdot z_{t-1} + \beta_t \cdot \epsilon_t \tag{5}$$

Gdje je $0 < \beta_n^2 < 1$ i $\epsilon \sim N(0,1)$.

Zašumljeni latentni vektor z_T ulazi u model UNet koji sadrži slojeve unakrsne pažnje. Cilj je rekonstruirati vektor $\hat{z}_{T-1}$ iz vektora z_T uklanjanjem šuma. Za predviđanje šuma u trenutku T, UNet osim zašumljenog latentnog vektora z_T koristi i izlaz ranije opisanog modula TVE, koji na ulazu prima vremenski korak T i ugrađivanje znaka "*" (dobivenom koristeći CLIP-ov tekstualni koder).

Ovime je opisan jedan korak obrnute difuzije tj. uklanjanja šuma. Postupak se ponavlja za svaki korak t sve dok ne dođemo do koraka $t = 0$ i latentnog prikaza $\hat{z}_0$:

$$\hat{z}_{t-1} = f(\hat{z}_t, t, e_\theta(y,t)) \tag{6}$$

Pritom $e_\theta(y,t)$ označava izlaz modula TVE za zadano tekstualno ugrađivanje y i korak t. Ovo ugrađivanje zovemo ugrađivanje stila za korak t.

Gubitak (7) je definiran kao kvadrat L2 norme razlika stvarnog slučajnog šuma ϵ i šuma predviđenog modelom UNet ϵ_ϕ.

$$\mathcal{L} = \|\epsilon_t - \epsilon_\phi(z_t, t, e_\theta(y,t))\|_2^2, \tag{7}$$

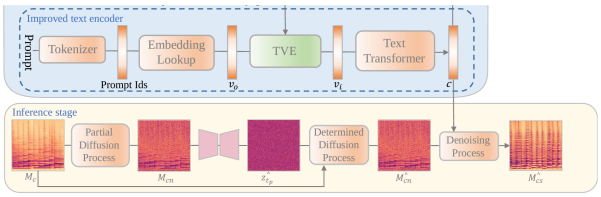
Dakle, uče se samo parametri θ modula TVE, dok su svi ostali parametri ϕ zamrznuti. Cilj optimizacije (utemeljen na prikazanom gubitku LDM-a) je:

$$\theta^* = \arg\min_\theta \mathbb{E}_{z,y,\epsilon_t,t}\left[\|\epsilon_t - \epsilon_\phi(z_t, t, e_\theta(y,t))\|_2^2\right], \tag{8}$$

Drugim riječima, cilj je pronaći parametre θ^* modula TVE za koje je očekivanje kvadrirane L2 norme razlike stvarnog slučajnog šuma $\epsilon_t \sim N(0,1)$ i predviđenog šuma modela ϵ_ϕ minimalno.

E. Prijenos stila

Nakon što smo naučili modul TVE na željenom stilu, prenosimo ga procesom stilizacije reducirane pristranosti (engl. *bias-reduced stylization*), prikazanom na slici 4. Cijeli proces sastoji se od tri glavna koraka.



Slika 4: Prijenos stila. Preuzeto iz [3].

Prvi korak prijenosa stila je parcijalni proces difuzije. Mel-spektrogramu M_c (koji predstavlja glazbu na koju želimo prenijeti stil), u latentnom prostoru VAE-a, prvo se difuzijskim procesom dodaje šum dok ne dođe do koraka $t_p = T \cdot strength$. Zašumljeni latentni vektor spektrograma u koraku t_p označavamo s M_{cn}.

Nakon prvog koraka prijenosa stila slijedi korak reduciranja pristranosti. Na temelju zašumljenog latentnog vektora M_{cn} predviđamo šum za korak t_p. Dobiveni izlaz označavamo sa $\hat{z}_{t_p}$.

$$\hat{z}_{t_p} = f(M_{cn}, t_p, e_\theta(y, t_p)) \tag{9}$$

Sada početni spektrogram M_c ponovno prolazi proces difuzije do koraka t_p, ali ovaj put slučajni se šum zamjenjuje sa $\hat{z}_{t_p}$.

$$M_{c,t} = \sqrt{1-\beta_n^2} \cdot M_{c,t-1} + \beta_n \cdot \hat{z_{t_p}} \tag{10}$$

Ovaj postupak možemo smatrati kao dodavanje pristranosti u zašumljenu sliku kako bismo smanjili utjecaj pristranosti modela. Nakon ovog koraka slijedi konačno uklanjanje šuma.

Rezultat koraka reduciranja pristranosti $\hat{M}_{cn}$ prolazi postupak obrnute difuzije tj. uklanjanja šuma s navođenjem bez klasifikatora (engl. *Classifier-Free Guidance*).

U svakom koraku t predviđaju se 2 šuma, šum navođen ugrađivanjem za ulazni znak "*" i šum bez navođenja tj. šum navođen ugrađivanjem za ulazni znak "".

$$\hat{\epsilon}_{t,text} = \epsilon_\theta(z_t, t, e_\theta(" * ", t)) \tag{11}$$

$$\hat{\epsilon}_{t,uncod} = \epsilon_\theta(z_t, t, e_\theta("", t)) \tag{12}$$

Ukupni predviđeni šum za korak t računa se po jednadžbi:

$$\hat{\epsilon}_t = \hat{\epsilon}_{t,uncod} + scale \cdot (\hat{\epsilon}_{t,text} - \hat{\epsilon}_{t,uncod}) \tag{13}$$

Pritom $scale$ označava hiperparametar koji kontrolira jačinu utjecaja navođenja. Na temelju dobivenog šuma $\hat{\epsilon}_t$, računamo djelomično odšumljenu latentnu reprezentaciju $\hat{M}_{cn,t-1}$. Ovaj postupak ponavlja se sve dok ne dođemo do latentne reprezentacije za trenutak $t = 0$.

Na kraju se, nakon dekodiranja VAE-om, dobije mel-spektrogram glazbe s prenesenim stilom $\hat{M}_{cs}$.

V. Mjere dobrote

Korištene su 2 mjere dobrote zasnovane na modelu CLAP: očuvanje sadržaja (engl. *content preservation*) i podudaranje stila (engl. *style fit*).

CLAP (engl. *Contrastive Language-Audio Pretraining*) je predtrenirani model strojnog učenja koji kodira zvuk i tekst u zajednički prostor ugrađivanja (semantički prostor). Sličniji parovi zvučnih i tekstualnih zapisa bit će blizu u ovom prostoru, dok će različiti biti udaljeni.

Mjera CLAP služi za evaluaciju usklađenosti između ugrađivanja zvučnih zapisa i tekstualnih opisa dobivenih modelom CLAP. Nadahnuta je mjerom CLIP (engl. *Contrastive Language-Image Pretraining*) koja radi na vrlo sličan način, ali između slika i tekstualnih opisa.

$$\text{CLAPscore}(x, y) = ReLU\left(\frac{E(x) \cdot E(y)}{||E(x)|| \cdot ||E(y)||}\right) \quad (14)$$

Pritom x i y mogu predstavljati zvučni zapis ili tekst, dok $E(x)$ predstavlja ugrađivanje dobivenom modelom CLAP. Rezultat je u intervalu $[0, 1]$, gdje 1 predstavlja savršenu usklađenost, a 0 da nema usklađenosti.

A. Očuvanje sadržaja

Očuvanje sadržaja mjeri sličnost između ulaznog zvučnog zapisa x_0 (prije prijenosa stila) i generiranog zvučnog zapisa $\hat{x}_0$ (nakon prijenosa stila). Mjera CP računa kosinusnu sličnost između ugrađivanja dvaju zvučnih zapisa dobivenih modelom CLAP.

$$CP(x_0, \hat{x}_0) = \text{CLAPscore}(x_0, \hat{x}_0) \quad (15)$$

Iznos je ograničen na interval $[0, 1]$. Što je vrijednost veća, to je sadržaj očuvaniji,

B. Podudaranje stila

Podudaranje stila mjeri koliko stil generiranog zvučnog zapisa $\hat{x}_0$ odgovara zadanom stilu Y. Originalna mjera SF računa kosinusnu sličnost između ugrađivanja generiranog zvučnog zapisa i tekstualnog opisa stila.

U našem radu koristimo blago izmijenjenu varijantu mjere. Y ne predstavlja tekstualni opis stila (jer isti nisu dostupni), već niz zvučnih zapisa zadanog stila. Pri izračunu kosinusne sličnosti tada koristimo prosjek ugrađivanja pojedinih zvučnih zapisa stila.

$$SF(\hat{x}_0, Y) = ReLU\left(\frac{E(\hat{x}_0) \cdot \frac{1}{N}\sum_{i=1}^{N} E(y_i)}{||E(\hat{x}_0)|| \cdot ||\frac{1}{N}\sum_{i=1}^{N} E(y_i)||}\right) \quad (16)$$

Iznos je ograničen na interval $[0, 1]$. Što je vrijednost veća, to se stil bolje podudara,

VI. Eksperimentalni rezultati

Korišteni skup podataka sadrži 179 zapisa sadržaja iz 13 različitih kategorija, kao i 74 zapisa stila podijeljenih na 18 različitih stilova. Kako bismo mogli prenijeti neki željeni stil, potrebno je prvo učiti model na zvučnim zapisima tog stila. Odabrali smo 3 različita stila: *bird*, *accordion* i *chime*. Za svaki od odabranih stilova naučen je jedan model.

Tijekom učenja modela, koristili smo optimizator Adam s početnom stopom učenja $1 \cdot 10^{-4}$, hiperparametrom β_1 iznosa 0.9 te hiperparametrom β_2 iznosa 0.999. Svaki model učili smo 3000 epoha s veličinom mini-grupe postavljenom na 1. Učenje modela je trajalo između 30 i 120 minuta na grafičkoj kartici NVIDIA GeForce RTX 3060, ovisno o broju odabranih isječaka stila.

Nakon učenja modela, isti koristimo za prijenos naučenog stila na odabrane zvučne zapise. Za prijenos stila trebamo odabrati nekoliko hiperparametara: broj koraka prijenosa, *scale* i *strength*. Broj koraka prijenosa obično fiksiramo na iznos 50, dok hiperparametre *scale* i *strength* mijenjamo ovisno o željenom odnosu očuvanje sadržaja i podudaranja stila. Velik iznos hiperparametra *scale* usmjerava predviđanje šuma strože prema zadanom stilu, dok manji iznos stavlja veći naglasak na kreativnost. Velik iznos hiperparametra *strength* prioritizira prijenos stila nad očuvanjem sadržaja.

Evaluaciju rezultata provodimo koristeći već spomenute objektivne mjere dobrote: očuvanje sadržaja (engl. *contentent preservation* - CP) i podudaranje stila (engl. *style fit* - SF). Nakon prijenosa zadanog stila na sve dostupne zvučne zapise sadržaja, izračunati su iznosi obje mjere za svaki stilizirani zvučni zapis. Konačno, kako bismo dobili iznos mjera za cijeli skup zapisa sadržaja, dobivene rezultate uprosječujemo.

Različiti odabrani stilovi imaju različit broj dostupnih zvučnih zapisa stila. Ako model učimo na većem broju zvučnih zapisa, prijenos stila će, općenito govoreći, biti uspješniji. Na primjer, uspješniji je prijenos stila *accordion* (15 zvučnih zapisa) u usporedbi s prijenosom stila *bird* (1 zvučni zapis). Dodatno, pojedini stilovi zahtijevaju pažljivo ugađanje iznosa hiperparametara *scale* i *strength*. U slučaju stilova s manje dostupnih zvučnih zapisa, općenito je preporučeno koristiti manji iznos hiperparametara, posebice iznos hiperparametra *strength*.

Tablica I: Usporedba mjera dobrote za različite stilove i iznose hiperparametara

Stil	*Scale*	*Strength*	CP	SF
Accordion	4.5	0.6	0.40	**0.49**
Bird	3.5	0.45	0.35	0.40
Chime	3.5	0.45	**0.49**	0.41
Prosjek	-	-	0.41	0.43

Možemo vidjeti da je najbolji iznos mjere podudaranja stila postignut za stil *Accordion* uz vrijednost hiperparametra *scale* iznosa 4.5 odnosno vrijednost hiperparametra *strength* iznosa 0.6. S druge strane, najbolji iznos mjere očuvanja sadržaja postignut je za stil *Chime* uz vrijednost hiperparametra *scale* iznosa 3.5 odnosno vrijednost hiperparametra *strength* iznosa

0.45. Vidimo da je u pitanju kompromis: povećanje očuvanja sadržaja općenito smanjuje podudaranje stila.

VII. Usporedba s postojećim pristupima

Za usporedbu, odabranu su sljedeća četiri modela:

- R+TI baseline - arhitektura Riffusion [11] uz korištenje tekstualne inverzije (engl. *Textual Inversion*) [13]
- SS VQ-VAE - varijacijski autoenkoder s vektoriziranom latentnom reprezentacijom učen samonadzirano, korišten za prijenos stila glazbe na temelju jednog primjera (engl. *One-shot music style transfer*) [14]
- MUSICGEN - jezični model za tekstom vođen prijenos stila glazbe uvjetovan melodijom (engl. *Text-Guided Music Stylization with Melody Conditioning*) [15]
- Originalni model iz reproduciranog rada [3]

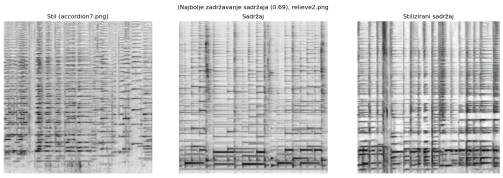
Tablica II: Usporedba mjera dobrote različitih modela

Model	CP	SF
R+TI [11] [13]	0.35	0.27
SS VQ-VAE [14]	0.24	0.28
MUSICGEN [15]	0.28	0.24
Originalni model [3]	**0.46**	0.28
Naš model	0.41	**0.43**

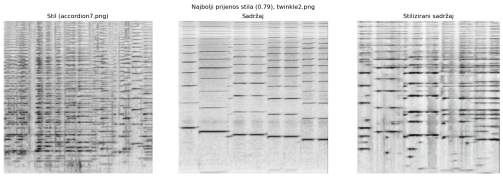
U tablici II možemo vidjeti iznos mjera očuvanja sadržaja i podudaranja stila za različite modele. Važno je napomenuti da je u slučaju odabranih modela mjera podudaranja stila izračunata na temelju ugrađivanja stiliziranih zvučnih zapisa i tekstualnih opisa stila, dok su za naš model ugrađivanja tekstualnih opisa stila zamijenjeni prosječnim ugrađivanjem zvučnih zapisa stila. U slučaju našeg modela, prikazane su prosječne mjere dobrote preko sva tri odabrana stila.

Možemo vidjeti da originalni model koji reproduciramo [3], kao i naš model, imaju bolji iznos obje mjere dobrote od preostalih odabranih modela za prijenos stila. Ako uspoređujemo našu implementaciju s originalnom, vidimo da originalni model ima nešto viši iznos mjere očuvanja sadržaja, dok naš model ima viši iznos podudaranja stila. Naravno, odnos očuvanja sadržaja i podudaranja stila veoma ovisi o konkretnom iznosu hiperparametara *scale* i *strength*. Veći iznos očuvanja sadržaja mogli bismo lako postići povećavanjem iznosa hiperparametra *strength*, ali po cijenu nižeg podudaranja stila.

Osim izračuna mjera dobrote za cijeli skup zvučnih zapisa sadržaja, dodatno smo za svaki naučeni stil pronašli primjere koji imaju najveći iznos mjere očuvanja sadržaja odnosno mjere podudaranja stila. Na slici 5 možemo vidjeti jedan od mel-spektrograma stila korišten za učenje modela, originalni mel-spektrogram sadržaja i stilizirani mel-spektrogram sadržaja za koji je iznos mjere očuvanja sadržaja najveći. S druge strane, na slici 6 možemo vidjeti jedan od mel-spektrograma stila korišten za učenje modela, originalni mel-spektrogram sadržaja i stilizirani mel-spektrogram sadržaja za koji je iznos mjere podudaranja stila sadržaja najveći.



Slika 5: Spektrogram stila, originalnog primjera i stiliziranog primjera s najvećim očuvanjem sadržaja



Slika 6: Spektrogram stila, originalnog primjera i stiliziranog primjera s s najvećim prijenosom stila

VIII. Zaključak

Pokazali smo da pristup temeljen na korištenju latentnog difuzijskog modela (LDM) uz koder teksta iz CLIP-a i vremenski promjenjiv koder (modul TVE) predstavlja obećavajuć smjer rada u području prijenosa stila glazbe.

Osim što konačni model ima dobar iznos objektivnih mjera dobrote očuvanja sadržaja i podudaranja stila, model na ulazu ne očekuje opširan tekstualni opis stila, već je dovoljno prikupiti nekoliko kratkih zvučnih zapisa koji predstavljaju isti. Dok klasični modeli često imaju problema s prijenosom stila prirodnih ili umjetno generiranih zvukova, naš model omogućava prijenos raznovrsnih stilova.

Dodatno, pošto se uče isključivo parametri modula TVE, samo učenje modela može se provoditi u razumnom vremenu na osobnom računalu. Ovo svojstvo našeg modela omogućava lokalno učenje i korištenje modela za prijenos stila bez ovisnosti o vanjskim pružateljima usluga.

Literatura

[1] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[2] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[3] H. Huang, Y. Wang, L. Li, and J. Lin, "Music style transfer with diffusion model," 2024.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[5] D. P. Kingma, M. Welling, *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[6] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," 2019.

[7] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with cyclegan," 2018.

[8] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2021.

[9] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," 2020.

[10] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," 2021.
[11] S. Forsgren and H. Martiros, "Riffusion - stable diffusion for real-time music generation." https://riffusion.com/about, 2022. Accessed: 2024-12-31.
[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.
[13] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," 2022.
[14] O. Cifka, A. Ozerov, U. Simsekli, and G. Richard, "Self-supervised vq-vae for one-shot music style transfer," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, June 2021.
[15] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2024.