

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

# **Nuotraukos spalvų korekcija naudojant vaizdų transformacijas**

## **Photo color correction using image transformations**

Bakalauro baigiamasis darbas

Atliko: Redas Serafimavičius

Darbo vadovas: j. asist. Boleslovas Dapkūnas

Darbo recenzentas: asist. dr. Andrius Vytautas Misiukas Misiūnas

Vilnius – 2024

## **Padėka už suteiktus HPC išteklius**

Darbo autorius dėkoja Vilniaus universiteto Matematikos ir informatikos fakulteto Informacinių technologijų atviros prieigos centrui už suteiktus HPC išteklius šio darbo skaičiavimams atlikti.

## Santrauka

Egzituojančios automatinės ekspozicijos ir baltos spalvos balanso korekcijos įrankiai, nenaudojantys giliojo mokymosi metodų, ekspozicijos ir baltos spalvos balanso korekciją atlieka nestabiliai bei nekokybiškai. Šiame darbe apmokomi Pix2Pix GAN ir InstructPix2Pix modeliai automatinei ekspozicijos ir baltos spalvos balanso korekcijai atlikti. Taip pat sukurta nuostolių funkcija skirta ekspozicijos ir baltos spalvos balanso uždaviniui, bei naudojant šią nuostolių funkciją buvo apmokytas neuroninių tinklų modelis. Šių apmokytų modelių sugeneruotiems rezultatams vertinti apskaičiuojamos pikselių spalvų skirtumų vidurkiai pagal CIEDE2000 formulę bei SSIM indeksai. Rezultatai lyginami tarp apmokytų modelių. Geriausias rezultatus parodęs modelis palyginamas su Adobe Lightroom Classic automatinės ekspozicijos ir baltos spalvos balanso korekcijos funkcionalumais. Lyginant modelius, modelis apmokytas su sukurta nuostolių funkcija parodė geresnius ir kokybiškesnius rezultatus nei Pix2Pix GAN, InstructPix2Pix ir Adobe Lightroom Classic automatinės ekspozicijos ir baltos spalvos balanso korekcijos funkcionalumai.

**Raktiniai žodžiai:** generatyvūs priešiški tinklai, Pix2Pix, difuzijos modeliai, Instruct-Pix2Pix, spalvų korekcija, nuostolių funkcija

## Summary

Existing automatic exposure and white balance correction tools that do not use deep learning techniques perform exposure and white balance correction in an unstable and poor quality way. In this work, Pix2Pix GAN and InstructPix2Pix models are trained to perform automatic exposure and white balance correction. A loss function has also been developed for the exposure and white balance task and a neural network model has been trained using this loss function. To evaluate the results generated by these trained models, the averages of the pixel colour differences are computed using the CIEDE2000 formula and the SSIM indices. The results are compared between the trained models. The model with the best results is compared with the automatic exposure and white balance correction functionalities of Adobe Lightroom Classic. When comparing the models, the model trained with the developed loss function showed better and higher quality results than Pix2Pix GAN, InstructPix2Pix and Adobe Lightroom Classic auto-exposure and white balance correction functionalities.

**Keywords:** generative adversarial networks, Pix2Pix, diffusion models, InstructPix2Pix, color correction, loss function

# Turinys

ĮVADAS .....	7
1. GENERATYVINIŲ PRIEŠIŠKŲ TINKLŲ MODELIAI .....	10
1.1. Generatyviniai priešiški tinklai .....	10
1.2. Sąlyginiai generatyviniai priešiški tinklai .....	10
1.3. Pix2Pix modelis .....	10
1.3.1. Duomenų rinkinys .....	11
1.3.2. Modelio architektūra .....	11
1.3.3. Pix2pix modelio tyrimo rezultatai .....	13
2. DIFUZIJOS MODELIAI .....	15
2.1. Triukšmą šalinantys tikimybiniai difuzijos modeliai .....	15
2.1.1. DDPM tiesioginis procesas .....	15
2.1.2. DDPM atgalinis procesas .....	16
2.1.3. DDPM modelio architektūra .....	16
2.1.4. DDPM nuostolių funkcija .....	16
2.1.5. DDPM mokymo procesas .....	16
2.1.6. DDPM vaizdų generavimo procesas .....	17
2.2. Latentinės difuzijos modeliai .....	17
2.2.1. Variacinis auto-enkoderis .....	17
2.2.2. CLIP transformeris .....	18
2.2.3. Stabiliosios difuzijos modelio veikimas .....	18
2.2.4. Stabiliosios difuzijos modelio panaudojimas vaizdų transformavimui .....	19
2.2.5. Stabiliosios difuzijos modelio generavimo parametrai .....	20
2.2.6. InstructPix2Pix modelis .....	20
3. EKSPERIMENTAS .....	22
3.1. Duomenų rinkinys .....	22
3.2. Pix2Pix GAN modelio mokymas .....	23
3.3. InstructPix2Pix modelio mokymas .....	24
3.3.1. Papildomas rezultatų apdorojimas .....	24
3.4. Ekspozicijos ir baltos spalvos balanso korekcijai pritaikytas modelis .....	26
3.4.1. Generatorius .....	26
3.4.2. Nuostolių funkcija .....	27
3.4.3. Modelio mokymas .....	28
3.5. Modelių lyginimas .....	29
3.5.1. Modelių vertinimo vaizdų aibės sudarymas .....	29
3.6. Rezultatų palyginimas .....	31
3.6.1. InstructPix2Pix modelio neredaguotų ir redaguotų išvesčių palyginimas .....	31
3.6.2. Modelių palyginimas tarpusavyje .....	33
3.6.3. Eksperimento metu sukurto modelio ir Adobe Lightroom Classic palyginimas .....	37
REZULTATAI IR IŠVADOS .....	39
ŠALTINIAI .....	40
PRIEDAI .....	42
1 priedas. „Adobe Lightroom Classic“ programos nustatymai kiekvienai nuokrypio varia- cijai .....	42

2 priedas. Programinis kodas .....	43
3 priedas. Sugeneruotų vertinimo vaizdų pavyzdžiai .....	44

# Įvadas

## Tyrimo objektas ir aktualumas

Fotografuojant automatinio režimu, dėl nenuspėjamų aplinkos veiksnių tokių kaip: šešėliai, saulės spinduliai šviečiantys tiesiai į objektyvą, fono spalva ir t.t., pasikeičia nuotraukos ekspozicija (angl. *exposure*) ir baltos spalvos balansas (angl. *white balance*), todėl nuotraukos spalvos gali neatitikti tikrovės. Taip pat, dėl netinkamai pasirinktų fotoaparato nustatymų taip gali nutikti ir fotografuojant rankiniu režimu.

Procesas, kuris suartina nuotraukos spalvas su realiomis, vadinamas spalvų korekcija. Spalvų korekcija įprastai susideda iš dviejų dalių:

1. **Nuotraukos ekspozicijos korekcija.** Nuotrauka gali būti per tamsi arba per šviesi. Norint sutvarkyti nuotraukos ekspozicijos, koreguojami bendras nuotraukos šviesos ryškumas, kontrastas, šešėlių ir šviesių zonų ryškumas. Tačiau ne visada galima kokybiškai atlikti ekspozicijos korekciją. Kai nuotrauka per tamsi, ir norima ją pašviesinti, vis labiau šviesinant nuotraukoje atsiranda vis daugiau triukšmo. Taip pat norint kokybiškai atlikti šviesos korekciją, labai svarbus kameros, su kuria buvo fotografuota dinaminis diapazonas (angl. *dynamic range*). Jis nurodo skirtumą tarp šviesiausios ir tamsiausios galimos nuotraukos reikšmės. Nuotraukos vietos, kuriose tamsumas arba šviesumas viršija šias reikšmes, praras visą informaciją ir galutiniam rezultate jos neturės jokių detalių ir bus atvaizduojamos tik kaip visiškai juodos arba visiškai baltos vietos. Todėl atliekant šviesumo korekciją šios vietos nebus paveikiamos ir jos atrodys visada taip pat, t.y. visiškai juodos arba visiškai baltos.
2. **Baltos spalvos balanso korekcija.** Kartais kameros automatinio baltos spalvos balanso funkcija netinkamai atpažįsta esamą apšvietimą, dėl ko nuotraukos gali turėti oranžinį, mėlyną, žalią arba violetinį atspalvius. Norint nuotraukos suartinti su realiomis spalvomis reikia atitinkamai šias spalvas pridėti. Pavyzdžiui, jei nuotrauką turi oranžinį atspalvį, norint gauti natūralias spalvas, reikia pridėti mėlynos spalvos. Pridėjus mėlynos spalvos gali atsirasti žalias atspalvis, todėl papildomai reikia pridėti violetinės spalvos. Tačiau kartais baltos spalvos balansas gali taip nutolti nuo realybės, kad grąžinti nuotraukai natūralias spalvas tampa labai sudėtinga.

Įprastai baltos spalvos balanso ir ekspozicijos korekcija atliekama nuotraukų redagavimo programose tokiose kaip „Adobe Lightroom Classic“. Šios programos turi įvairių įrankių, kurie leidžia manipuliuoti nuotraukos ekspozicija ir spalvomis. Taip pat jose neretai galima rasti ir automatinės baltos spalvos balanso ir ekspozicijos korekcijos funkciją. Deja, bet automatinė baltos spalvos balanso ir ekspozicijos korekcija šiose programose veikia gan prastai ir dažniausiai korekciją vis tiek tenka atlikti rankiniu būdu naudojantis kitais programos įrankiais.

Žmogui, kuris nesidomi fotografija ir nuotraukų retušavimu, netgi paprasta baltos spalvos balanso ir ekspozicijos korekcija gali kelti iššūkių, o tam tikrais atvejais net patyrusiam profesionalui tai gali būti sudėtinga arba net neįmanoma atlikti. Todėl reikalingas metodas, kuris galėtų automatiškai atlikti baltos spalvos balanso ir ekspozicijos korekciją. Vienas iš galimų būdų šiam metodui sukurti – pasinaudoti generatyvinių neuroninių tinklų modeliais. Pastaraisiais metais ypač išpopuliarėjus šiems modeliams, jie sėkmingai buvo pritaikyti daugelyje dalykinių sričių,

todėl verta patyrinėti generatyvinių neuroninių tinklų galimybes atlikti automatinę baltos spalvos balanso ir ekspozicijos korekciją.

Šiuo metu populiariausi, plačiausiai dokumentuoti, bei didžiausią bendruomenę turintys generatyvinių neuroninių tinklų modeliai skirti vaizdų generavimui bei transformavimui yra difuzijos (angl. *Diffusion*) ir generatyvinių priešiško neuroninių tinklų modeliai (angl. *Generative adversarial networks*) arba trumpiau – GAN modeliai. Todėl šiame darbe bus tyrinėjami būtent tokio tipo modeliai.

Nors ir ankščiau minėtų modelių didžiausias pranašumas yra pritaikomumas įvairiose srityse, bei įvairiose vaizdų transformavimo uždaviniuose, tačiau ši šių modelių savybė gali tapti trūkumu tokiuose vaizdų transformavimo uždaviniuose, kur rezultatas yra labai specifinis, bei gali būti aiškiai pamatuojamas. Būtent toks ir yra automatinės baltos spalvos balanso ir ekspozicijos korekcijos uždavinys. Turint poruotą duomenų rinkinį, galima pamatuoti spalvų skirtumą tarp dviejų nuotraukų ir taip įvertinti korekcijos kokybę. Todėl šiame darbe taip pat bus tyrinėjamas spalvų skirtumo, kaip nuostolių funkcijos apmokant neuroninių tinklų modelį, panaudojumas.

Verta paminėti, kad galima rasti darbų, kurie tyrinėja neuroninių tinklų modelius tik ekspozicijos korekcijai [ADO<sup>+</sup>21] arba tik baltos spalvos balanso korekcijai [AB20], tačiau nėra viešai prieinamų atliktų tyrimų, kurie tyrinėtų tiek ekspozicijos tiek baltos spalvos balanso korekcijas kartu, vienoje nuotraukoje.

## Tyrimo metodai

Išsiaiškinti kokius Stabilios difuzijos bei GAN modelių variantus treniruoti, bei kokią architektūrą pasirinkti naujam modeliui bus pasitelkti šie metodai:

- Mokslinės literatūros apžvalga
- Atliktas empirinis tyrimas

## Darbo tikslas

Šio darbo tikslas yra sukurti metodą atliekantį automatinę baltos spalvos balanso ir ekspozicijos korekciją neiškraipant nuotraukos ir nepridedant triukšmo, naudojant generatyvinių neuroninių tinklų modelius.

## Darbo uždaviniai

1. Sukurti duomenų rinkinį skirtą apmokyti modelius atlikti automatinę baltos spalvos balanso ir ekspozicijos korekciją kaip duomenų rinkinio pagrindą naudojant Adobe MIT FiveK duomenų rinkinį.
2. Atlikti literatūros analizę apie difuzijos bei GAN modelius skirtus vaizdų transformavimui.
3. Atlikti empirinį tyrimą apie įvairias neuroninių tinklų modelių architektūras naudojamas vaizdų transformavimui



4. Sukurti metodą automatinei baltos spalvos balanso ir ekspozicijos korekcijai atlikti naudojant įvairius neuroninių tinklų modelius.
5. Palyginti sukurtą metodą su egzistuojančiais automatinės baltos spalvos balanso ir ekspozicijos korekcijos metodais naudojant spalvų skirtumo, skaičiuojamo pagal CIEDE2000 formulę, bei SSIM indekso metrikas.

### **Darbo atlikimo procesas**

1. Sukurti suporuotų nuotraukų duomenų rinkinį.
2. Apmokyti generatyvinių priešiškių tinklų modelį naudojant sukurtą suporuotų duomenų rinkinį.
3. Adaptuoti difuzijos modelį naudojant sukurtą suporuotų duomenų rinkinį.
4. Sukurti bei apmokyti generatyvinių neuroninių tinklų architektūrą automatinei baltos spalvos balanso ir ekspozicijos korekcijai atlikti naudojant spalvų skirtumą kaip nuostolių funkciją
5. Palyginti apmokytų modelių rezultatus tarpusavyje bei su esamais automatinės baltos spalvos balanso ir ekspozicijos korekcijos metodais.

### **Laukiami rezultatai**

Šio darbo pagrindinis laukiamas rezultatas – sukurtas metodas, kuris atliktų automatinę baltos spalvos balanso ir ekspozicijos korekciją įvairių scenų JPEG formato nuotraukose stabiliau ir tiksliau nei egzistuojantys automatinės baltos spalvos balanso ir ekspozicijos korekcijos metodai nenaudojantys neuroninių tinklų modelių, kur kaip įvestis naudojama tik pati nuotrauka. Taip pat šis metodas neturėtų iškraipyti nuotraukos, bei nepridėti papildomo triukšmo.

# 1. Generatyvinių priešišku tinklų modeliai

## 1.1. Generatyviniai priešiški tinklai

Generatyviniai priešiški tinklų modelis (angl. *Generative Adversarial Networks*, toliau GAN) pirmą kart buvo pristatytas 2014 m. Montrealio universiteto mokslininkų [GPM<sup>+</sup>14]. Šiame darbe buvo pristatytas nauja generatyvinių modelių vertinimo sistema naudojant priešpriešinį procesą, kuris susideda iš dviejų neuroninių tinklų modelių: diskriminacinio modelio (D) ir generatyvinio modelio (G).

**Diskriminacinis modelis** yra dviejų klasių klasifikatorius. Jis kaip įvestį priima nuotrauką ir bando prognozuoti ar nuotrauka yra tikra (paimta iš duomenų rinkinio), ar sugeneruota generatyvinio modelio. Mokymosi procese šis modelis vis tobulėja atskirti modelio sugeneruotas ir tikras nuotraukas, taip priversdamas generatyvinį modelį generuoti kuo įtikinamesnius vaizdus.

**Generatyvinis modelis** iš atsitiktinio triukšmo bando sugeneruoti nuotrauką. Mokymosi procese modelis tobulėja bandant apgauti diskriminacinį modelį generuojant vis panašesnes nuotraukas į esančias duomenų rinkinyje.

GAN modeliai nepriima jokios įvesties, todėl neįmanoma po mokymo proceso įtakoti modelio išvesties. Todėl tokių modelių, be papildomų vaizdų transformavimui pritaikytų modifikacijų, negalima panaudoti vaizdų transformavimui.

## 1.2. Sąlyginiai generatyviniai priešiški tinklai

Sąlyginiai generatyviniai priešiški tinklų (angl. *Conditional Generative Adversarial Networks*, toliau CGAN) modelis pirmą kart buvo pristatytas Montrealio universiteto mokslininkų 2014 m. [MO14]. CGAN modelio struktūra panaši į GAN modelio, t.y. abu susideda iš dviejų neuroninių tinklų modelių: diskriminacinio ir generatyvinio. Tačiau priešingai nei paprasti GAN modeliai, CGAN modelis priima įvestį, kuri gali būti pateikta bet kokia forma (pvz. tekstas, nuotrauka ir t.t...). Ši įvestis perduodama tiek generatyviniam tiek diskriminaciniam modeliui kaip papildomas įvesties sluoksnis.

Kadangi CGAN modeliai gali priimti įvestį kuri gali būti nuotrauka, tai įgalina tokio tipo modelius pritaikyti vaizdų transformavimui.

## 1.3. Pix2Pix modelis

2016 m. Berklio universiteto mokslininkai publikavo straipsnį [IZZ<sup>+</sup>17], kuriame tyrinėjama galimybė pritaikyti CGAN modelį vaizdų transformavimui. Prieš šį darbą buvę vaizdų transformavimo metodai, naudojantys neuroninius tinklus, buvo stipriai priklausomi nuo dalykinės srities ir tinkamai žemu lygiu sukonstruotos nuostolių funkcijos būtent tai dalykinei sričiai. Šio darbo tikslas buvo sukurti universalų metodą atlikti vaizdų transformavimą, kuriam nereiktų kiekvienai dalykinei sričiai atskirai kurti vis kitokios nuostolių funkcijos, o ją būtų galima apibrėžti aukštu lygiu. Su GAN ir CGAN modeliu atsiradimu tai tapo įmanoma, kadangi šiuose modeliuose

nuostolių funkcija yra išmokstama mokymo metu ir už ją atsakingas diskriminacinis modelis. Kartu su šiuo darbu autoriai pateikė ir šio modelio implementaciją pavadinimu Pix2Pix.

### 1.3.1. Duomenų rinkinys

Šiam modeliui mokytis reikalingas suporuotų nuotraukų duomenų rinkinys. Viena nuotrauka yra įvesties pavyzdys, o kita – kokia turėtų būti išvestis. Kitaip ši išvestis gali būti vadinama etalonu (angl. *ground truth*) [IZZ<sup>+</sup>17].



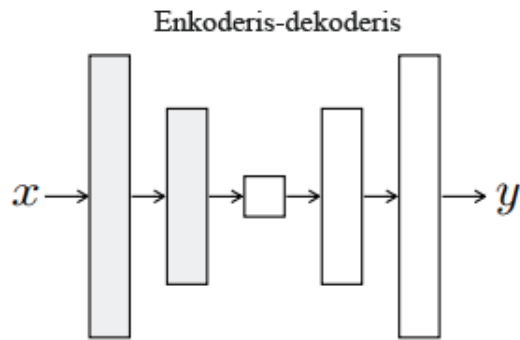
1 pav. Duomenų rinkinio pavyzdys skirtas apmokyti Pix2Pix modelį transformuoti dienos nuotraukas į nakties

### 1.3.2. Modelio architektūra

Kadangi Pix2Pix yra CGAN modelis, jis susideda iš dviejų, jau anksčiau minėtų dalių: diskriminacinio ir generatyvinio modelių [IZZ<sup>+</sup>17].

**Generatyvinis modelis:** Pix2Pix modelyje generatyvinis modelis yra U-NET tipo neuroninis tinklas [IZZ<sup>+</sup>17]. Norint suprasti kaip veikia U-NET tipo neuroniniai tinklai, iš pradžių reikia suprasti enkoderio-dekoderio tipo tinklus.

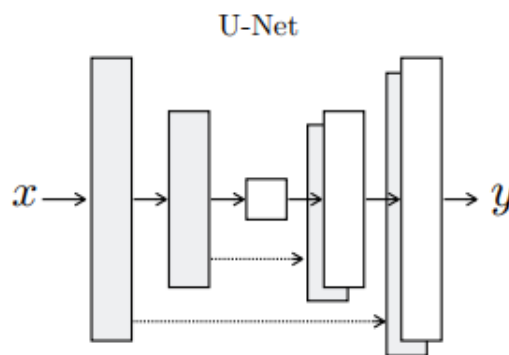
Enkoderio-dekoderio tipo tinklai susideda, kaip sufleruoja pavadinimas, iš dviejų dalių – enkoderio ir dekoderio. Enkoderis turi kelis sluoksnius, pro kuriuos eidama įvesties duomenų seka, yra palaipsniui mažinama ir galų gale paverčiama į mažesnę, fiksuoto dydžio, tenzorių, kuris yra tarsi abstraktus įvesties apibūdinimas ir reprezentuoja svarbiausią įvesties informaciją. Tuo tarpu dekoderis taip pat turi kelis sluoksnius, kurie veikia atvirkščiai nei enkoderio sluoksniai. Pro juos einantis iš enkoderio gautas tenzorius palaipsniui didinamas ir taip iš pradinės įvesties abstrakčios reprezentacijos sugeneruojama nauja išvestis [VSP<sup>+</sup>17].



2 pav. Enkoderio-dekoderio tipo neuroninio tinklo struktūros reprezentacija

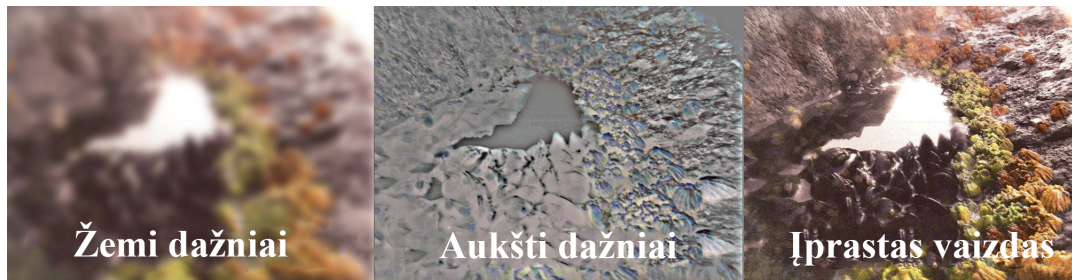
U-NET tipo tinklai yra labai panašūs į enkoderio-dekoderio tipo tinklus, tačiau jie turi vieną esminį skirtumą. Enkoderio-dekoderio tipo tinkluose sluoksniai yra sujungti vienas su kitu nuosekliai, todėl, kad gauti išvestį, įvestis turi praeiti pro visus sluoksnius. Tuo tarpu U-NET tipo tinkluose enkoderio ir dekoderio sluoksniai gali būti sujungti, todėl kai kurie sluoksniai gali būti praleisti. Kadangi daug įvesties detalių enkoderyje yra prarandamos, tam tikrų enkoderio ir dekoderio sluoksnių sujungimas leidžia išsaugoti tam tikras smulkias detales iš įvesties, kas yra ypatingai svarbu vaizdų transformavimo uždaviniuose [RFB15].

Pix2pix modelyje  $i$  sluoksnis yra sujungtas su  $n-1$  sluoksniu, kur  $n$  yra visų tinklo sluoksnių skaičius [ZZ<sup>+</sup>17].



3 pav. Supaprastinta Pix2pix modelyje naudojamo U-NET tipo neuroninio tinklo struktūros reprezentacija

**Diskriminacinis modelis:** Kad suprasti, kodėl Pix2pix modelyje buvo pasirinkta tam tikra diskriminatoriaus architektūra ir kaip ji veikia, reikia suprasti ką vadiname aušto ir žemo dažnio dalimis vaizduose. Aukšti dažniai vaizduose reprezentuoja vaizdo kontūrus bei briaunas, bet turi mažai informacijos apie spalvas. Tuo tarpu žemi dažniai reprezentuoja vaizdo spalvas bei jų pasiskirstymą, tačiau turi mažai informacijos apie vaizdo detales, bei kontūrus. Sudėjus šiuos du dažnius gauname įprastą, pilną vaizdą [Zen18].



4 pav. Nuotraukos išskirstytos į žemus ir aukštus dažnius, bei jos originalo pavyzdys

Šiame darbe [IZZ<sup>+</sup>17] buvo tyrinėjamas L1 nuostolių funkcijos pritaikymas vaizdų transformavimui. L1 nuostolių funkcija, kitaip vadinama *mažiausių absoliučių nukrypimų* funkcija, naudojama siekiant sumažinti paklaidą ir apskaičiuojama kaip visų absoliučių skirtumų tarp tikrosios vertės ir prognozuojamos vertės suma

$$L1 = \sum_{i=1}^n |y_{tikroji} - y_{prognozuojama}| \quad (1)$$

Naudojant L1 funkciją, kaip vienintelę nuostolių funkciją šiame modelyje, buvo pastebėta, kad sugeneruotos išvesties nuotraukos yra neryškios, jose trūksta detalių, tačiau puikiai perteikia norimo rezultato nuotraukos žemo dažnio dalis. Todėl L1 nuostolių funkcija naudojama kartu su GAN modeliams būdingu diskriminaciniu modeliu [IZZ<sup>+</sup>17].

Aukšto dažnio vaizdo dalių korektiškumui užtikrinti naudojamas *PatchGAN* diskriminacinis modelis. Jis vietoje to, kad klasifikuotų visą vaizdą į tikrą ar netikrą, išskaido vaizdą į  $N \times N$  dydžio lopinėlius ir klasifikuoja kiekvieną atskirai, tada išveda vidurkį iš visų lopinėlių rezultatų ir prieina vienos išvados - vaizdas tikras arba netikras [IZZ<sup>+</sup>17].

Iš generatyvaus modelio mokymo metu gautas rezultatas vertinamas pagal abu L1 nuostolių funkciją ir *PatchGAN* diskriminacinį modelį. Eksperimentų metu nustatyta, kad taikant būtent tokį metodą gaunami geriausi rezultatai [IZZ<sup>+</sup>17].

### 1.3.3. Pix2pix modelio tyrimo rezultatai

Pix2pix modelis buvo testuotas su įvairiais vaizdų transformavimo uždaviniais: dienos vaizdų pavertimas nakties vaizdais, vaizdo nuspalvinimas, semantinių žymių transformavimas į realaus pasaulio nuotrauką, įpiešimas, bei daugelį kitų. Atlikus šiuos bandymus buvo pastebėta, kad šis modelis duoda puikius rezultatus labai skirtingose ir įvairiuose domenuose. Taip pat šis modelis gali duoti gerus rezultatus net su nedideliu duomenų rinkiniu ir sąlyginai trumpu mokymo laiku [IZZ<sup>+</sup>17].



5 pav. Pix2pix modelio rezultatas apmokyto su duomenų rinkiniu iš 400 nuotraukų porų

## 2. Difuzijos modeliai

Difuzijos modeliai – tai generatyvinių modelių klasė. Šių modelių idėja ir pirminis metodas pirmą kartą buvo pristatytas 2015 m. Standfordo ir Berklio universitetų mokslininkų [SWM<sup>+</sup>15]. Šiame darbe buvo pristatytas naujas metodas, kuris remiasi nepusiausvyrinės statistinės fizikos ir sekvencinio Monte Carlo metodo įdėjomis. Šis metodas, naudojant Markovo grandinę, paverčia paprastą, žinomą skirstinį, (pvz. Gauso skirstinį) į tikslinį (duomenų) skirstinį. Tam įgyvendinti buvo pasiūlytas karkasas kuris susideda iš dviejų procesų:

1. **Tiesioginis (difuzijos) procesas** – tai procesas, kurio metu duomenims pamažu yra pridedamas triukšmas, kol visiškai prarandama pradinių duomenų struktūra.
2. **Atgalinis (triukšmo šalinimo) procesas** – šis procesas pamažu šalina triukšmą, tol kol gaunami reikalingi duomenys. Atlikti šį procesą yra apmokomas modelis.

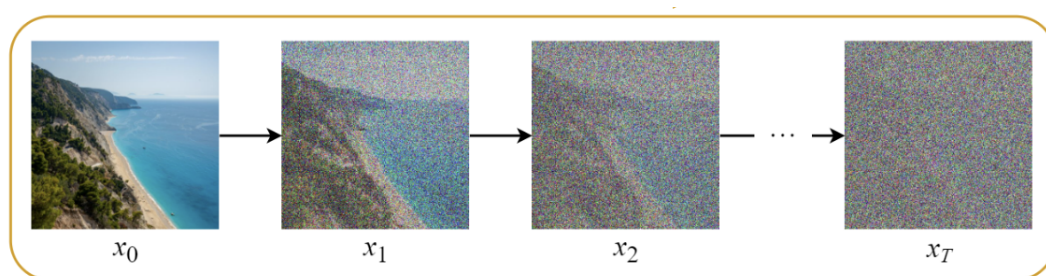
Modelis apmokytas atlikti atgalinį procesą, vėliau gali būti naudojamas naujų duomenų generavimui iš triukšmo.

### 2.1. Triukšmą šalinantys tikimybiniai difuzijos modeliai

Triukšmą šalinantys tikimybiniai difuzijos modeliai (*trumpiau – DDPM*) buvo pristatyti 2020 m. Berklio universiteto mokslininkų [HJA20]. Šiame darbe buvo pristatytas konkretus mokymo procesas bei DDPM modelio implementacija, kuri geba generuoti aukštos kokybės ir raiškos vaizdus. Taip pat šiame darbe buvo aprašytos tiesioginio ir atgalinio procesų implementacijos.

#### 2.1.1. DDPM tiesioginis procesas

Tiesioginiame procese duomenims iteratyviai pamažu pridedama triukšmo tol, kol pradinių duomenų struktūra yra visiškai prarandama ir gaunamas Gauso skirstinys.



6 pav. Tiesioginis procesas

Triukšmo kiekis kiekvienoje šio proceso iteracijoje pridedamas ne atsitiktinai. Už jį atsakingas dispersijos keitimo planuotojas (angl. *variance scheduler*). Šis planuotojas nustato kiekvieno žingsnio pridedamo triukšmo kiekį, bei su kiekvienu žingsniu jį didina. Triukšmo kiekis žymimas  $\beta_t$ , kur  $t$  yra laiko žingsnio indeksas. Šiame darbe buvo naudota  $T = 1000$  žingsnių, pradinė triukšmo kiekio vertė  $\beta_1 = 10^{-4}$ , o galutinė triukšmo vertė  $\beta_T = 0,02$

### 2.1.2. DDPM atgalinis procesas

Atgalinis arba triukšmo šalinimo procesas siekia iš triukšmo, gauto po tiesioginio proceso, atkurti originalų vaizdą. Visas triukšmas nėra pašalinamas iškart, o iteratyviai, per lygiai tiek pat žingsnių kiek tiesioginiame procese.

### 2.1.3. DDPM modelio architektūra

Atlikti atgalinį procesą yra mokomas U-NET tipo modelis. Tačiau jis šiek tiek skiriasi nuo Pix2Pix modelyje naudojamo U-NET architektūros. DDPM naudojamame U-NET modelyje, tarp konvoliucinių blokų taip pat yra panaudoti savaiminio dėmesio sutelkimo (angl. *self-attention*) blokai. Šie blokai leidžia modeliui sutelkti dėmesį į specifinius vaizdo regionus, bei atsižvelgti į informaciją kitose vaizdo vietose, taip atsižvelgiant į viso vaizdo kontekstą [HLJ]<sup>+</sup>23].

### 2.1.4. DDPM nuostolių funkcija

DDPM modelyje buvo panaudota tokia nuostolių funkcija:

$$L(\theta) := \mathbb{E}_{t, x_0, \epsilon} [\|\epsilon - \epsilon_{\theta}(x_t, t)\|^2] \quad (2)$$

- $\theta$  - mokomo modelio parametrai
- $t$  - laiko žingsnis
- $x_0$  - originalus vaizdas be triukšmo
- $x_t$  - vaizdas su triukšmu laiko žingsnyje  $t$
- $\epsilon$  - standartinis Gauso triukšmo tenzorius
- $\epsilon_{\theta}$  - modelio sugeneruoto triukšmo tenzorius

Šioje nuostolių funkcijoje apskaičiuojama vidutinė kvadratinė paklaida (angl. *Mean squared error*) tarp standartinio Gauso triukšmo ir triukšmo sugeneruoto modelio tam tikrame laiko žingsnyje. Tokia nuostolių funkcija skatina modelį tiksliai numatyti triukšmą, kuris buvo pridėtas prie pradinių duomenų kiekviename tiesioginio proceso etape.

### 2.1.5. DDPM mokymo procesas

DDPM mokymo procese, modelis yra mokomas nuspėti triukšmą, kuris buvo pridėtas laiko žingsnyje  $t$ . Mokymosi proceso žingsniai yra tokie:

1. Atsitiktinai parenkamas laiko žingsnis  $t$ .
2. Modeliui kaip įvestis duodamas vaizdas iš duomenų rinkinio su triukšmu atsitiktinai parinktame laiko žingsnyje  $x_t$ .
3. Modelis bando nuspėti pridėta triukšmą laiko žingsnyje  $t$ .
4. Naudojant nuostolių funkciją, modelio nuspėtas triukšmas palyginamas su tikruoju pridėtu triukšmu laiko žingsnyje  $t$ .
5. Pagal nuostolių funkcijos rezultatą atnaujinami modelio svoriai.



### 2.1.6. DDPM vaizdų generavimo procesas

Apmokius DDPM modelį, galima pradėti generuoti vaizdus. Vaizdų generavimo proceso žingsniai yra tokie

1. Apmokytam modeliui iš pradžių duodamas vaizdas sudarytas tik iš triukšmo ir laiko žingsnio indeksas, kuris lygus visų laiko žingsnių kiekiui  $T$ .
2. Iš duoto triukšmo modelis bando nuspėti koks triukšmas buvo pridėtas.
3. Iš modelio gautas triukšmo skirtumas atimamas iš įvesties triukšmo ir gaunamas vaizdas laiko žingsnyje  $T - 1$ .
4. Tada šis gautas vaizdas vėl naudojamas kaip įvestis modeliui tol, kol pasiekiamas laiko žingsnis  $t = 0$ .

Gautas vaizdas yra visiškai naujas vaizdas, kuris atspindi duomenų rinkinyje buvusius vaizdus.

## 2.2. Latentinės difuzijos modeliai

Latentinės difuzijos modeliai (*trumpiau - LDM*) - tai teksto transformavimo į vaizdus modeliai, pirmą kartą pristatyti 2022 m. Miuncheno universiteto mokslininkų [RBL<sup>+</sup>22]. Šie modeliai, paverčia teksto įvestį į vaizdą naudojant difuzijos modelį.

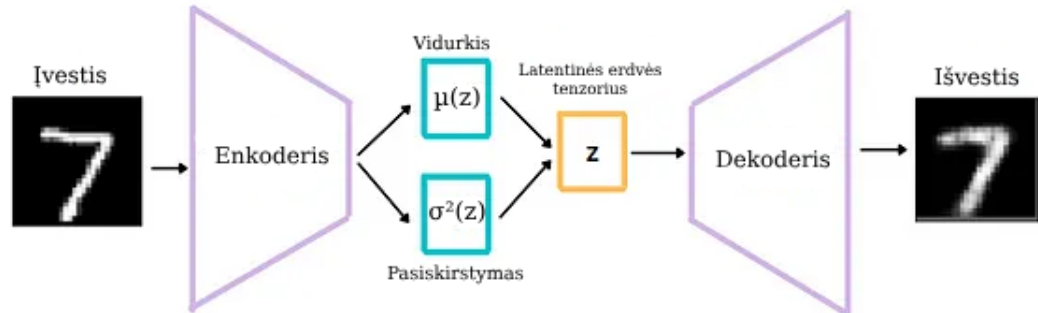
Tai nėra pirmasis teksto transformavimo į vaizdus modelis. Tačiau prieš tai egzistavę teksto transformavimo į vaizdus modeliai, tokie kaip DALL-E modelis [RPG<sup>+</sup>21], reikalavo labai daug resursų tiek apmokymo, tiek vaizdų generavimo procese, todėl naudoti šiuos modelius yra labai brangu. LDM modeliai siekia sumažinti resursų sąnaudas teksto transformavimo į vaizdus uždaviniuose neprarandant rezultatų kokybės.

Populiariausias LDM sukurtas remiantis šiuo darbu - tai Stabiliosios difuzijos modelis sukurtas įmonės Stability AI. Stabiliosios difuzijos naudoja patobulintą difuzijos procesą, kuris leidžia ženkliai sumažinti resursų sąnaudas. Prieš tai egzistavę modeliai difuzijos ir triukšmo šalinimo veiksmus atlikdavo su visu vaizdu. Atsižvelgiant į tai, jog triukšmo šalinimo procesas reikalauja pakartotinio atlikimo siekiant išgauti vieną atvaizdą, tai reikalauja didelio resursų kiekio. Stabiliosios difuzijos modelis triukšmo šalinimo ir difuzijos veiksmus perkėlė į latentinę erdvę - suspaustą pirminių duomenų atvaizdą, kur kiekvienas matmuo atitinka konkrečią savybę ar charakteristiką. [AT23].

### 2.2.1. Variacinis auto-enkoderis

Auto-enkoderiai - tai neuroniniai tinklai, kurie suspaudžia įvestį į žemesnės dimensijos tenzorių ir iš to tenzoriaus bando atkurti originalius duomenis [BKG23]. Šis suspaustas tenzorius atsiranda latentinėje erdvėje. Tačiau Stabiliosios difuzijos modeliuose naudojamas Variacinis auto-enkoderis. Variacinis auto-enkoderis kaip ir paprastas auto-enkoderis turi enkoderį ir dekode-derį, tačiau enkoderis vietoje tenzoriaus, kuris nurodo konkretų tašką latentinėje erdvėje, grąžina tenzorių, kuris nurodo įvesties duomenų pasiskirstymą latentinėje erdvėje, pagal kurį sukuria- ma dekoduoja pradinio vaizdo reprezentacija [KW13]. Dėl to latentinė erdvė tampa labiau

struktūrizuota, todėl dekoduojant tenzorių, galima gauti visiškai naujų išvesčių keičiant tenzorius parametrus. Tai padaro Variacinį auto-enkoderį generatyviniu neuroniniu tinklu.



7 pav. Variacinio auto-enkoderio veikimo schema

### 2.2.2. CLIP transformeris

Kadangi Stabiliosios difuzijos modelis yra teksto transformavimo į vaizdus modelis, reikia apdoroti teksto įvestį bei panaudoti ją kaip vieną iš įvesčių. Už tai atsakingas CLIP transformeris, kuris buvo sukurtas 2021 m. įmonės OpenAI [RKH<sup>+</sup>21]. Šis modelis sukurtas ir apmokytas suprasti ryšius tarp teksto ir vaizdų. Kaip įvestį šis modelis priima teksto eilutę ir vaizdą. Šios įvestys, naudojant atitinkamus enkoderius, paverčiamos į teksto ir vaizdo įterpinius ir pagal juos yra apskaičiuojamas panašumo balas, kuris nurodo kiek įvesties teksto eilutė sutampa su duotu vaizdu.

### 2.2.3. Stabiliosios difuzijos modelio veikimas

Stabiliosios difuzijos modelio veikimo principas panašus kaip DDPM modelio, tačiau Stabiliosios difuzijos modelis turi kelis esminius patobulinimus. Šis modelis susideda iš trijų, anksčiau minėtų, dalių:

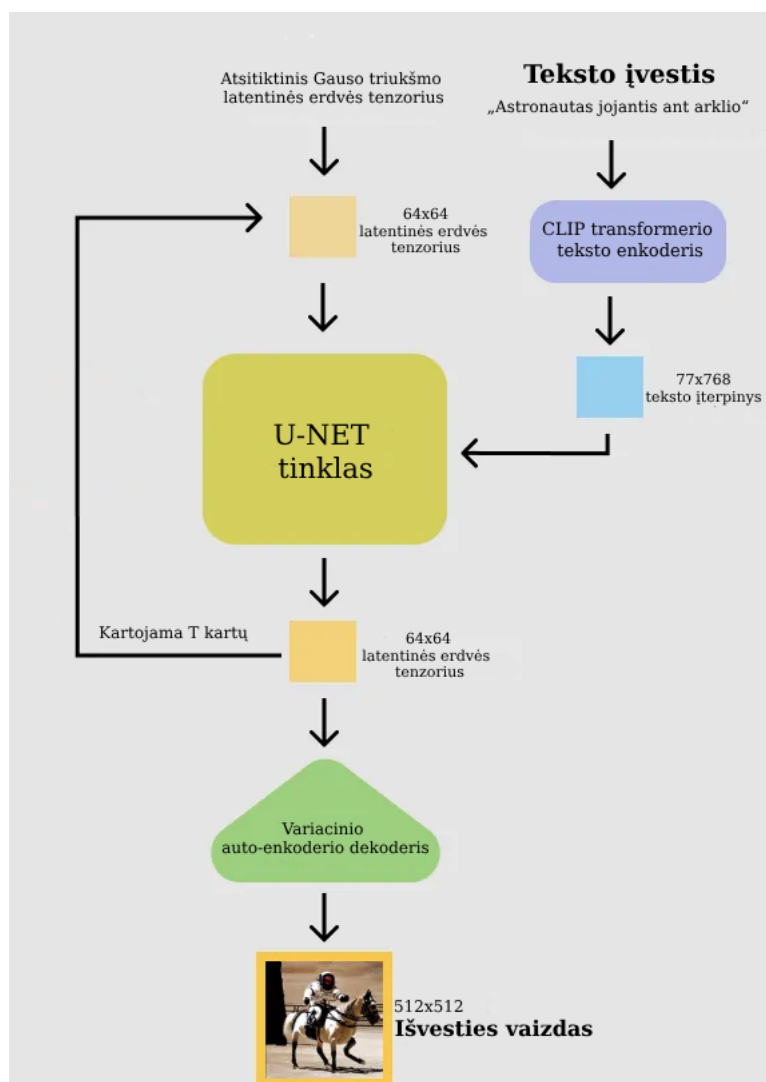
1. Variacinio auto-enkoderio
2. CLIP transformerio
3. U-NET neuroninio tinklo

Visų pirma įvesties vaizdas naudojant Variacinį auto-enkoderį yra užkoduojamas į latentinės erdvės tenzorių. Tiek tiesioginis tiek atgalinis procesai yra vykdomi su šiuo užkoduotu tenzoriumi. Tai leidžia atlikti triukšmo šalinimo operacijas žymiai efektyviau, naudojant žymiai mažiau resursų, nei tai atliekant su neužkoduotu vaizdu. Pavyzdžiui (3, 512, 512) formos vaizdas yra užkoduojamas į (4, 64, 64) formos tenzorių. Todėl šiuo atveju veiksmas su tenzoriumi reikalauja 48 kartus mažiau atminties nei su pilnu vaizdu. Modelio panaudojimo naujam vaizdui sugeneruoti atveju, nenau-

dojamas Variacinio auto-enkoderio enkoderis, tiesiog paduodamas atsitiktinis Gauso triukšmas [RBL<sup>+</sup>22].

Toliau, naudojant CLIP transformerį, iš vartotojo įvestos teksto eilutės sukuriamas teksto įterpinys. Šis teksto įterpinys kartu su vaizdo latentinės erdvės tenzoriu paduodami į U-NET neuroninį tinklą. Šiame tinkle vyksta triukšmo šalinimo procesas ir kaip išvestis gražinamas naujas latentinės erdvės tenzorius su mažiau triukšmo. Šis tenzorius kartu su ankščiau gautu teksto įterpiniu paduodamas atgali į U-NET tinklą ir taip kartojama, kol pašalinamas visas triukšmas, t.y. kai prieinama  $t = 0$  laiko žingsnio.

Kai visas triukšmas iš tenzorius pašalintas, jis, naudojant Variacinio auto-enkoderio dekoderį, dekoduojamas į galutinį vaizdą.



8 pav. Stabiliosios difuzijos veikimo schema naujo vaizdo generavimui

#### 2.2.4. Stabiliosios difuzijos modelio panaudojimas vaizdų transformavimui

Stabiliosios difuzijos modelis taip pat gali būti pritaikytas vaizdų transformavimo uždaviniams. Vaizdų transformavimo procesas labai panašus į vaizdų generavimo procesą, tačiau pačioje pradžioje, vietoje atsitiktinio Gauso triukšmo, įvesties vaizdas paverčiama į latentinės erdvės tenzorių naudojant Variacinį auto-enkoderį, o prie šio tenzorius pridamas atsitiktinis triukšmas. Su

šiuo tenzoriumi toliau vyksta įprastas triukšmo šalinimo procesas. Po triukšmo šalinimo proceso gautas latentinės erdvės tenzorius dekoduojamas, naudojant Variacinio auto-enkoderio dekoderį, į naują vaizdą kuris atspindi įvesties nuotrauką [fac22].

### 2.2.5. Stabiliosios difuzijos modelio generavimo parametrai

Norint gauti kuo įmanoma geresnius rezultatus, galima keisti tam tikrus apmokyto Stabiliosios difuzijos modelio parametrus [Ash24]:

1. **Pradinio triukšmo reikšmė (angl. *Seed*)** – galima nurodyti skaičių pagal kurį bus sugeneruotas triukšmas. Jei šis parametras nenurodomas, triukšmas sugeneruojamas atsitiktinai.
2. **Orientacinė skalė (angl. *Guidance scale*)** – ši skalė nurodo, kiek generavimo procese atsižvelgiama į teksto įvestį. Kuo reikšmė didesnė – tuo griežčiau atsižvelgiama į teksto įvestį ir sugeneruotas vaizdas ją labiau atitinka.
3. **Stiprumas (angl. *Strength*)** – šis parametras naudojamas vaizdų transformavimo uždaviniuose. Kuo didesnė šio parametro vertė – tuo daugiau triukšmo bus pridėta į įvesties vaizdą. Kuo daugiau triukšmo pridedama į įvesties vaizdą – tuo mažiau išvesties vaizdas panašus į įvesties.
4. **Triukšmo valdiklis (angl. *Sampler*)** – galima parinkti triukšmo pridėjimo bei spėjimo metodą ir strategiją. Skirtingi metodai gali generuoti mažiau arba labiau kūrybingus vaizdus. Taip pat skirtingi metodai reikalauja skirtingo kiekio resursų.
5. **Žingsnių kiekis (angl. *Steps*)** – šis parametras nurodo skaičių žingsnių, per kiek iš vaizdo bus pašalinamas visas triukšmas. (t.y. parametras  $T$ )

### 2.2.6. InstructPix2Pix modelis

InstructPix2Pix – tai vaizdų transformavimo modelis, kuris buvo pristatytas 2023 m. Berklio universiteto mokslininkų [BHE23]. Šis modelis yra specialiai apmokytas Stabiliosios difuzijos modelis atlikti vaizdų redagavimą. Naudojant šį modelį teksto įvestyje nereikia apibūdinti viso vaizdo, užtenka tik liepiamąja nuosaka nurodyti ką reikia pakeisti arba pridėti.



9 pav. InstructPix2Pix modelio vaizdo ir teksto įvestys ir išvestys

Šis modelis buvo apmokytas su visiškai sugeneruotu duomenų rinkiniu. Visų pirma buvo surinkti įvairių vaizdų aprašymai. Tada naudojant naudojant GPT-3 kalbos modelį, kiekvienam

vaizdo aprašymui buvo sugeneruota atsitiktinė vaizdo redagavimo instrukcija ir redaguoto vaizdo aprašymas. Toliau, iš pradinio vaizdo aprašymo ir redaguoto vaizdo aprašymo, naudojant Stabiliosios difuzijos modelį, buvo sugeneruoti du vaizdai: prieš redagavimą ir po redagavimo.

Galutinis duomenų rinkinys susideda iš trijų dalių:

1. Pradinis vaizdas
2. Redagavimo instrukcija liepiamąja nuosaka
3. Vaizdas po redagavimo

### 3. Eksperimentas

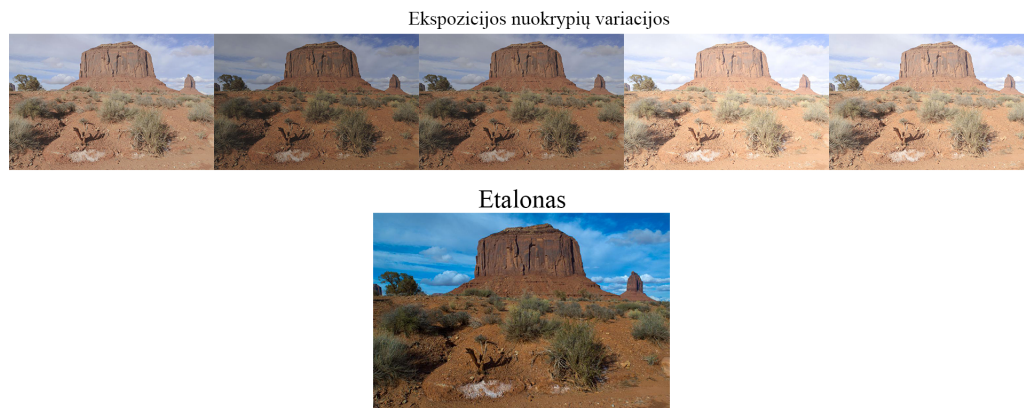
Atliekant šį darbą buvo atliktas eksperimentas, kurio metu buvo apmokyti du egzistuojantys modeliai: Pix2Pix GAN modelis ir InstructPix2Pix modelis, kuris paremtas Stabiliosios difuzijos modeliu. Kadangi šie modeliai yra bendrosios paskirties, ir yra skirti įvairiems vaizdų transformavimo uždaviniams, taip pat buvo sukurtas ir apmokytas modelis skirtas būtent automatinei ekspozicijos ir baltos spalvos balanso korekcijai atlikti.

#### 3.1. Duomenų rinkinys

Apmokyti šiuos modelius atlikti automatinę spalvų korekciją reikalingas toks duomenų rinkinys, kuris atspindėtų realias situacijas, t.y. kiekviena įvesties nuotrauka turi būti su neteisinga ekspozicija ir baltos spalvos balansu. Ekspozicija nuotraukose gali būti nukrypusi į dvi puses – į tamsesnę arba į šviesesnę. Baltų spalvų balansas taip pat gali nukrypti į dvi puses – į mėlyną arba į oranžinę pusę. Taip pat nuokrypiai gali būti skirtingo stiprumo. Todėl reikalingas toks duomenų rinkinys, kuriame būtų įvairių stiprumų ekspozicijos ir baltų spalvų balanso nuokrypių variacijų, arba reikalinga tik viena iš spalvų korekcijos dalių: ekspozicijos arba baltų spalvų balanso korekcija. Taip pat svarbu paminėti, kad ne visada spalvų korekcija gali būti reikalinga, t.y. nuotraukos ekspozicija ir baltų spalvų balansas yra korektiškas, todėl būtina į duomenų rinkinį įtraukti tokių nuotraukų, kur spalvų korekcija nereikalinga.

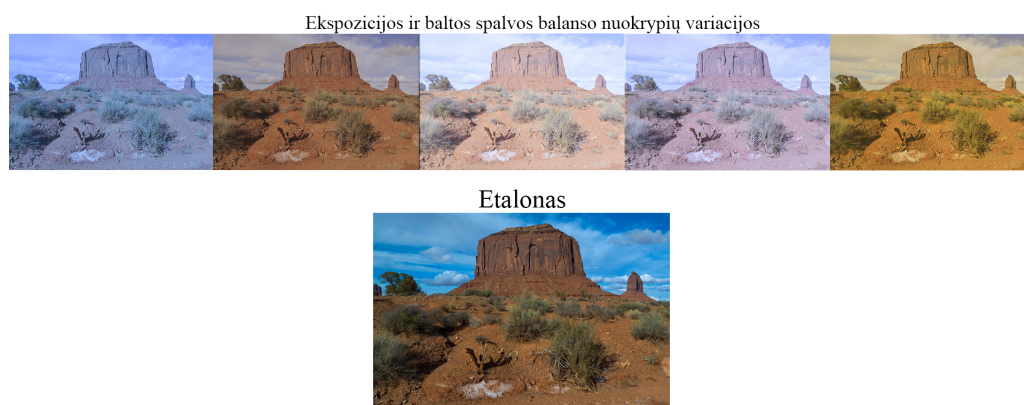
Visiems modeliams apmokyti reikalingas suporuotų nuotraukų duomenų rinkinys, kur vieną duomenų rinkinio porą sudaro nuotrauką atspindinti įvestį, o kita – etalonas.

Rasti viešai prieinamų suporuotų duomenų rinkinių, kuriose viena nuotrauka būtų su neteisinga ekspozicija ir neteisingu baltos spalvos balansu, o kita – korektiška, rasti nepavyko, todėl reikėjo jį susikurti. Tam buvo paimtas Jorko universiteto mokslininkų 2021 m. tyrime [ADO<sup>+</sup>21] naudotas duomenų rinkinys. Šiame darbe buvo tyrinėjamos neuroninių tinklų galimybės atlikti tik ekspozicijos korekciją. Šis duomenų rinkinys yra sudarytas iš „MIT-Adobe FiveK“ duomenų rinkinio, kuris turi 5000 įvairių nuotraukų, kurios rankinių būdu buvo retušuotos nuotraukų redagavimo profesionalų. Daliai šių nuotraukų buvo su simuliuoti 5 skirtingi, tikroviški ekspozicijos nuokrypiai. Galutinis duomenų rinkinys yra sudarytas iš 3535 skirtingų nuotraukų, kiekviena nuotrauka turi po 5 skirtingas ekspozicijos nuokrypio variacijas ir vieną etaloną. Todėl šį duomenų rinkinį sudaro 17675 poros. Į šį duomenų rinkinį taip pat įeina nuotraukos, kurioms ekspozicijos korekcija nereikalinga.



10 pav. Ekspozicijos nuokrypių variacijų duomenų rinkinio vieneto pavyzdys

Iš viso yra 17675 nuotraukų įvairiais ekspozicijos nuokrypiais, tačiau dar reikia kiekvienai iš jų pridėti baltos spalvos balanso nuokrypį. Tam pasitelktas Github platformos vartotojo „mahmoudnafifi“ sukurtas įrankis [AB19] skirtas pridėti nuotraukai baltos spalvos balanso nuokrypį. Šis įrankis, pasitelkęs neuroninius tinklus, duotai nuotraukai atsitiktinai parenka nuokrypio pusę (mėlyną arba oranžinę), bei stiprumą. Taip pat šis įrankis duotai nuotraukai gali ir nepridėti baltos spalvos balanso nuokrypio.



11 pav. Ekspozicijos ir baltos spalvos balanso nuokrypių variacijų duomenų rinkinio vieneto pavyzdys

### 3.2. Pix2Pix GAN modelio mokymas

Šiame eksperimente buvo naudota „Pytorch“ karkasą naudojanti Pix2Pix GAN modelio implementacija, kuri patalpinta „GitHub“ platformos repozitorijoje naudotojo „junyanz“. Šioje repozitorijoje yra ne tik modelio implementacija, tačiau ir pagalbinių kodo fragmentai skirti duomenų rinkinio paruošimui, modelio mokymui ir testavimui.

Kadangi Pix2Pix modeliui mokyti reikalingos tokios nuotraukų poros, kur vienoje nuotraukoje yra abi poros nuotraukos viena šalia kitos, pasinaudoję Pix2Pix implementacijos repozitorijoje esančiu pagalbiniu kodo fragmentu, turimas atskiras įvesties ir etalono nuotraukas iš duomenų rinkinio, pagal failo vardą sujungiame šias nuotraukas į vieną failą, kur abi nuotraukos yra viena šalia kitos.

Modelis buvo mokomas VU MIF Paskirstytų skaičiavimų tinkle naudojant vieną vaizdo plokštę. Kas keletą epochų buvo tikrinamas sugeneruotas rezultatų failas, su tos epochos modelio sugeneruotų vaizdų pavyzdžiais. Modelis buvo mokomas tol, kol nebebuvo pastebima ryškių rezultatų pagerėjimų. Iš viso modelis buvo mokytas 185 epochas ir užtruko apytiksliai 100 valandų.

### 3.3. InstructPix2Pix modelio mokymas

InstructPix2Pix modeliui apmokyti buvo naudojamas „Hugging face“ įmonės paruoštas šio modelio mokymo kodas, kuris randamas tos pačios įmonės sukurtos „Diffusers“ bibliotekos repozitorijoje.

Visų pirma reikėjo paruošti duomenų rinkinį į formatą tinkamą naudoti su šiuo mokymo kodu. Mokymo kodas naudoja „Datasets“ biblioteką kuri taip pat sukurta įmonės „Hugging face“. Norint naudoti turimą duomenų rinkinį, reikėjo sukurti CSV failą su trim stulpeliais kiekvienai duomenų rinkinio nuotraukai. Šiame faile turi būti 3 stulpeliai:

1. Įvesties nuotraukos nuoroda
2. Nuotraukos redagavimo instrukcija tekstu
3. Išvesties nuotraukos nuoroda

Kiekvienai nuotraukai buvo naudojama ta pati redagavimo instrukcija: „color correct this image“ (liet. *ištaisyti šio vaizdo spalvas*). Toliau reikėjo pakoreguoti mokymo kodą, kad duomenų rinkinys būtų užkrautas naudojant būtent šį CSV failą.

Šis modelis, kaip ir Pix2Pix GAN modelis, buvo mokomas VU MIF Paskirstytų skaičiavimų tinkle, naudojant vieną vaizdo plokštę. Kas 15000 žingsnių modelis buvo išsaugojamas ir testuojamas su įvairių situacijų vaizdais. Modelis buvo mokomas tol, kol nebebuvo pastebima ryškių rezultatų pagerėjimų. Iš viso modelis buvo mokomas 75000 žingsnių. ir truko apytiksliai 40 valandų.

#### 3.3.1. Papildomas rezultatų apdorojimas

Ištreniravus modelį ir patikrinus jo rezultatus, buvo pastebėta, kad iš modelio gautuose vaizduose yra iškraipyti užrašai bei žmonių veidai.



12 pav. Iškraipyto teksto pavyzdys. Kairėje – InstructPix2Pix modelio sugeneruotas vaizdas, dešinėje – originalus vaizdas





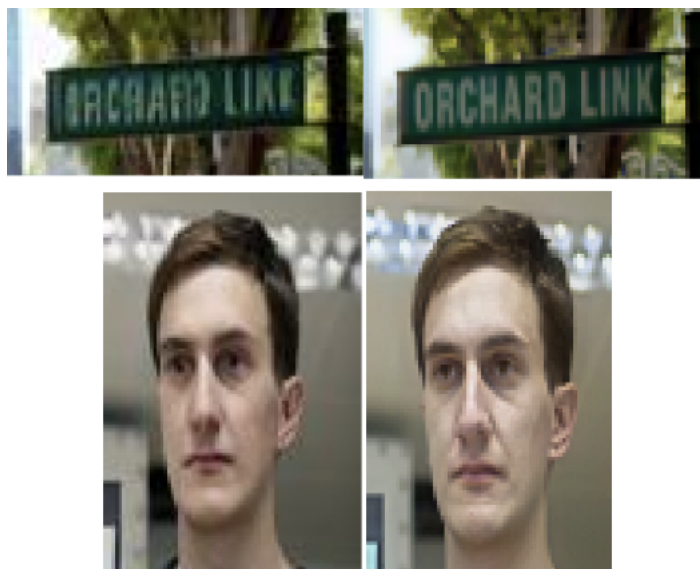
13 pav. Iškraipyto žmogaus veido pavyzdys. Kairėje - InstructPix2Pix modelio sugeneruotas vaizdas, dešinėje - originalus vaizdas

Tačiau galima pastebėti, kad iškraipyta informacija yra vaizdo aukšto dažnio informacijos dalis. O ekspozicijos ir baltos spalvos balanso korekcija vyksta vaizdo žemo dažnio informacijos dalyje. Todėl ištaisyti šiuos iškraipymus galima sujungus originalaus vaizdo aukšto dažnio dalį su modelio sugeneruoto vaizdo žemo dažnio dalimi.



14 pav. Modelio sugeneruoto vaizdo žemo dažnio dalies ir originalaus vaizdo aukšto dažnio dalies suma

Toks dviejų vaizdų atskirų dažnių dalių sujungimas leidžia išlaikyti originalaus vaizdo detales, bei turėti koreguotą nuotraukos ekspoziciją ir baltos spalvos balansą, sugeneruotą modelio.



15 pav. Vaizdų pavyzdžiai prieš ir po skirtingų dažnių sujungimo

Šis vaizdo redagavimas atliekamas su „Adobe Photoshop“ programine įranga. Visų pirma abiem, originaliam ir iš modelio gautam vaizdams pridedamas Gauso išliejimas (angl. *Gaussian blur*). Toliau pridedama dar viena originalaus vaizdo kopija, ir šis vaizdas atimamas iš išlieto originalaus vaizdo, gauname skirtumą tarp vaizdų. Išlietas vaizdas – žemo dažnio informacija, skirtumas – aukšto dažnio informacija. Galiausiai, originalaus vaizdo aukšto dažnio informacija sujungiama su modelio sugeneruoto vaizdo žemo dažnio informacija.

Verta paminėti, kad aukšto dažnio vaizdo dalyje yra laikoma dalis vaizdo kontrasto. Su tam tikrais ekspozicijos nuokrypiais vaizdo kontrastas gali sumažėti arba padidėti lyginant su korektišku vaizdu, todėl sujungus šiuos du dažnius, nors ir nežymiai, tačiau gali pakisti vaizdo kontrastas (matoma 15 pav. pavyzdyje su žmogaus veidu).

Dar vienas šio metodo trūkumas – atskiriant dažnius, kiekvienai situacijai reikalingi skirtingos išliejimo reikšmės, todėl procesą sunku automatizuoti.

### 3.4. Ekspozicijos ir baltos spalvos balanso korekcijai pritaikytas modelis

Tiek Pix2Pix GAN, tiek InstructPix2Pix modeliai yra bendros paskirties modeliai, gebantys atlikti įvairius vaizdų transformavimo uždavinius. Jie mokymo metu bando sugeneruoti nuotrauką pagal duotą etaloną ir neturi aiškiai išreikštos nuostolių funkcijos, specifinės tam tikram uždaviniui. Tačiau ekspozicijos ir baltos spalvos balanso korekcijos uždavinyje, tikslą galima aiškiai išreikšti naudojant įvairias metrikas kaip nuostolių funkciją. Todėl šiame darbe buvo sukurtas vaizdų transformavimo modelis su specialiai ekspozicijos ir baltos spalvos balanso korekcijai pritaikyta nuostolių funkcija.

#### 3.4.1. Generatorius

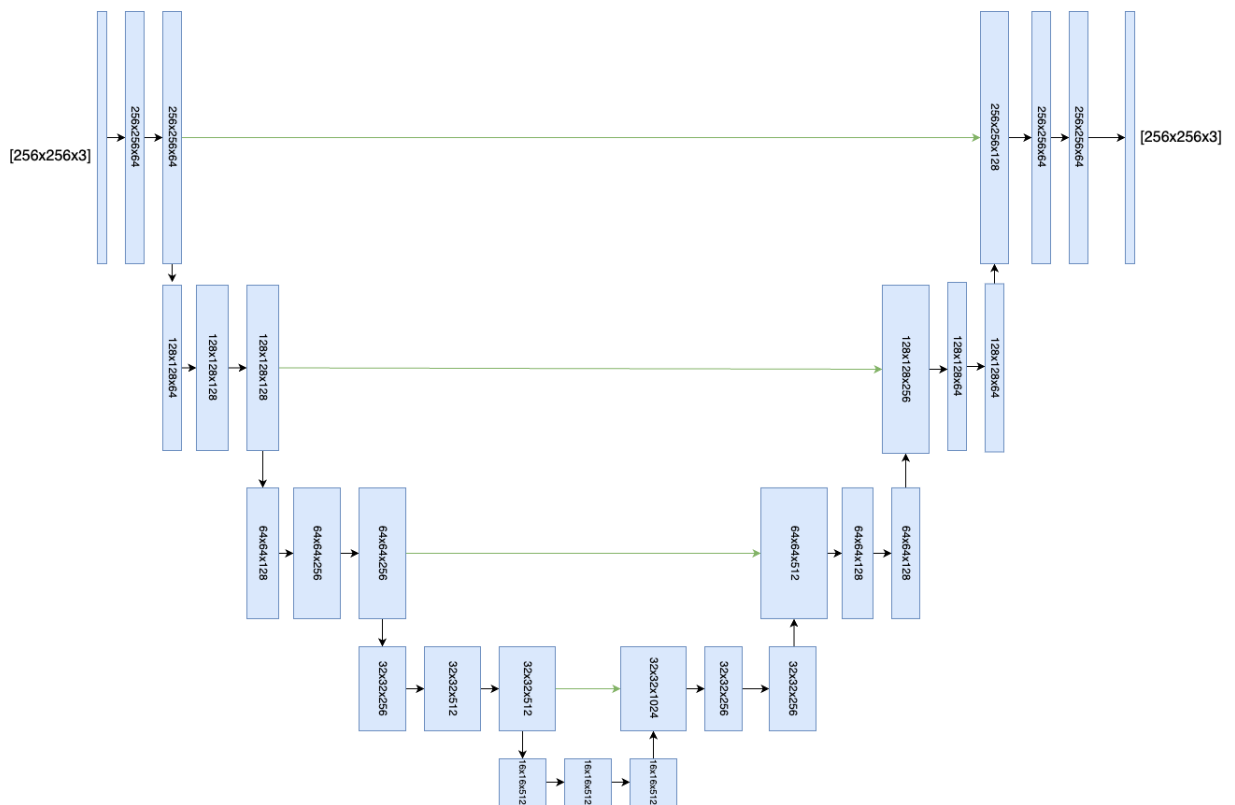
Šiame modelyje generatoriui panaudota U-NET tipo architektūra. Ši architektūra buvo pasirinkta, nes U-NET architektūros modeliai dažnai naudojami įvairiuose vaizdų transformavimo uždaviniuose, bei rodo gerus rezultatus. Taip pat, šis modelis dėl turimų praleidimo jungčių (angl.

*skip connections*), leidžia išsaugoti įvesties vaizdo detales, kas yra ypatingai svarbu ekspozicijos ir baltos spalvos balanso korekcijos uždavinyje. Verta paminėti, kad tiek Pix2Pix GAN tiek Stabiliosios difuzijos modeliai naudoja U-NET tipo architektūros tinklus.

Eksperimento metu sukurto modelio architektūra galima išskaidyti į šias dalis:

- **Dvigubos konvoliucijos sluoksnis** – šiame sluoksnyje atliekamos dvi konvoliucijos su filtro branduolio dydžiu (angl. *kernel size*) – 3 ir papildymo transformacija (angl. *padding*) – 1. Po kiekvienos konvoliucijos atliekama ReLu aktyvacija.
- **Mažinimo konvoliucijos sluoksnis** – šiame sluoksnyje vaizdo aukštis ir plotis sumažinamas dvigubai naudojant sutelkimą imant maksimalią reikšmę (angl. *Max pooling*), bei atliekama dviguba konvoliucija.
- **Didinimo konvoliucijos sluoksnis** – šiame sluoksnyje vaizdo aukštis ir plotis padidina-  
mas dvigubai ir atliekama dviguba konvoliucija.
- **Paskutinės konvoliucijos sluoksnis** – atliekama paskutinė konvoliucija su filtro branduolio dydžiu – 1, kurios išėitis yra galutinis vaizdas.

Šiame modelyje įvesties vaizdai visų pirma pritaikoma dviguba konvoliucija, tada atliekamos 4 mažinimo konvoliucijos bei 4 didinimo konvoliucijos ir gale pritaikoma paskutinė konvoliucija.



16 pav. Modelyje naudojamo U-NET tinklo architektūra

### 3.4.2. Nuostolių funkcija

Kaip nuostolių funkcija naudojamos dviejų metrikų santykis. Pirmoji – visų vaizdo pikselių spalvų skirtumų, skaičiuojamų pagal CIEDE2000 formulę (*žymima*  $\Delta E_{00}$ ), vidurkis. Antroji – SSIM metrika kuri matuoja struktūrinį panašumą tarp dviejų vaizdų. Šią nuostolių funkciją galima

apibrėžti tokia formule, kur  $n$  - vaizdo pločio pikselių skaičius,  $m$  - vaizdo aukščio pikselių skaičius,  $x$  - įvesties nuotrauka,  $y$  - etalonas.

$$LOSS_{cc} = \frac{1}{n * m} \frac{\sum_{i=0}^n \sum_{j=0}^m \Delta E_{00}(x_{ij}, y_{ij})}{SSIM(x, y)} \quad (3)$$

Skirtumas suskaičiuotas pagal CIEDE2000 formulę labiausiai atspindi tokį skirtumą tarp spalvų, kaip jį supranta žmogaus akis [LCR01]. Šios formulės rezultato mažiausia reikšmė yra 0, kai spalvos visiškai vienodos, o didžiausia reikšmė, nors ir teoriškai yra  $\infty$ , tačiau praktiškai niekada neviršija 100 (lyginant visiškai baltą RGB(255,255,255) ir visiškai juodą RGB(0,0,0) spalvas rezultatas = 100). Apskaičiuoti šiam skirtumui tarp dviejų spalvų naudojant CIEDE2000 formulę nuostolių funkcijoje naudojama „colour-science“ biblioteka.

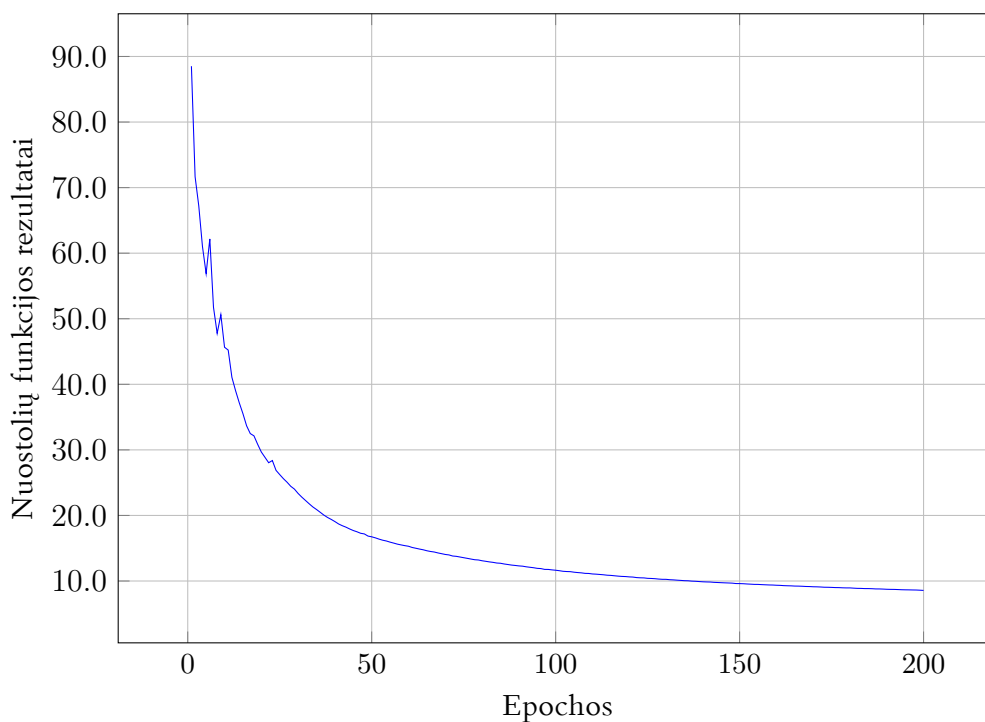
Kita nuostolių funkcijoje naudojama metrika yra struktūrinio panašumo indekso metrika (SSIM). Ji nurodo kiek du paveikslėliai yra struktūriškai panašūs. Šio indekso skaičiavime vertinami trys kriterijai: vaizdo ryškumas, kontrastas bei struktūra. Indekso reikšmės gali varijuoti tarp -1 ir 1, kur -1 reiškia, kad vaizdai yra visiškai nepanašūs, o 1 - vaizdai identiški [NA20]. Šiam indeksui tarp dviejų nuotraukų suskaičiuoti naudojama „pytorch-ssim“ biblioteka.

Kadangi  $\Delta E$  vidurkį daliname iš SSIM indekso, o SSIM indeksas gali būti 0 arba įgauti neigiamą reikšmę, tai gali pakenkti arba nutraukti mokymo procesą. Todėl tokias reikšmes reikia transformuoti į teigiamas. Jei SSIM indeksas yra 0, tada gražinama 0.01. Jei SSIM indeksas yra neigiamas, tada paimamas indekso modulis ir jis atimamas iš 1. Toliau skirtumas yra pakeliamas kvadratu ir padauginamas iš 0.01. Taip, SSIM reikšmei artėjant prie -1, gaunama vis mažesnė teigiama reikšmė. Ši transformacija aprašyta 4 formulėje, kur  $x$  - neigiamas SSIM indeksas.

$$SS\tilde{I}M = (1 - |x|)^2 * 0.01 \quad (4)$$

### 3.4.3. Modelio mokymas

Šio modelio mokymo metu buvo naudojamas Adam optimizatorius, su mokymosi žingsniu 0.0002. Po kiekvienos epochos būdavo išsaugojama viena pavyzdinė nuotrauka sugeneruota modelio. Taip pat po kiekvienos epochos buvo išsaugomas modelis. Taip pat kaip Pix2Pix GAN ir InstructPix2Pix modeliai, šis buvo mokomas VU MIF Paskirstytų skaičiavimų tinkle, naudojant vieną vaizdo plokštę. Modelis buvo mokomas tol, kol nuostolių funkcijos rezultatas pradėjo kristi labai lėtai. Iš viso mokymas užtruko 200 epochų ir truko apytiksliai 96 valandas. Mokymosi metu nuostolių funkcijos rezultatas nukrito nuo 88.5182 iki 8.5650.



17 pav. Nuostolių funkcijos rezultatai per visas epochas

### 3.5. Modelių lyginimas

Apmokytų modelių rezultatams vertinti pasitelkiama tokia strategija:

1. Iš 3 skirtingų situacijų nuotraukų, kurios turi korektišką ekspoziciją bei baltos spalvos balansą, sukuriamos 3 nuotraukų aibės, kurias sudarys tos pačios nuotraukos su skirtingos ekspozicijos ir baltos spalvos balanso variacijomis.
2. Visi trys modeliai pritaikomi su šiomis nuotraukų aibėmis kaip įvestimi.
3. Pritaikyti „Adobe Lightroom Classic“ nuotraukų redagavimo programos siūlomą automatinės ekspozicijos ir baltos spalvos balanso korekcijos funkcionalumus, kuris nenaudoja neuroninių tinklų.
4. Įvertinti rezultatų spalvų tikslumui, kiekvienam rezultatui bus apskaičiuojamas visų įvesties ir išvesties nuotraukų pikselių skirtumų ( $\Delta E$ ), skaičiuojamų pagal CIEDE2000 formulę, vidurkis.
5. Įvertinti rezultatų struktūrinį panašumą, kiekvienam rezultatui bus skaičiuojama SSIM metrika.

#### 3.5.1. Modelių vertinimo vaizdų aibės sudarymas

Rezultatų vertinimo nuotraukų aibės sudarymui paimtos 3 skirtingų situacijų nuotraukos: 1 situacija – lauko kadras, 2 situacija – vidaus kadras, 3 situacija – kadras, kuriame aiškiai matomas žmogus. Visose šitose nuotraukose matomos spalvų korekcijai skirtos lentelės. Pagal jas rankiniu būdu buvo atlikta ekspozicijos ir baltos spalvos balanso korekcija kiekvienai nuotraukai.

## Lauko kadras

## Vidaus kadras

Kadras su aiškiai matomu žmogumi



18 pav. Pasirinktos skirtingų situacijų nuotraukos vertinimo aibei sudaryti

Ekspozicijos ir baltos spalvos balanso nuokrypių variacijoms sudaryti, buvo apibrėžtos 4 ekspozicijos nuokrypių variacijos bei 4 baltos spalvos balanso nuokrypių variacijos. Taip pat joms suteikti identifikaciniai kodai.

Ekspozicijos nuokrypių variacijos ir jų identifikaciniai kodai:

- **SD** – šiek tiek per tamsi
- **SL** – šiek tiek per šviesi
- **VD** – labai tamsi
- **VL** – labai šviesi

Baltos spalvos balanso nuokrypių variacijos ir jų identifikaciniai kodai:

- **SC** – šiek tiek mėlyna
- **SW** – šiek tiek oranžinė
- **VC** – labai mėlyna
- **VW** – labai oranžinė

Iš jų buvo sudarytos naujos, jungtinės ekspozicijos ir baltos spalvos balanso nuokrypių variacijos.

1 lentelė. Jungtinės ekspozicijos ir baltos spalvos balanso nuokrypių variacijos pagal identifikacinius kodus

	SD	SL	VD	VL
SC	SD_SC	SL_SC	VD_SC	VL_SC
SW	SD_SW	SL_SW	VD_SW	VL_SW
VC	SD_VC	SL_VC	VD_VC	VL_VC
VW	SD_VW	SL_VW	VD_VW	VL_VW

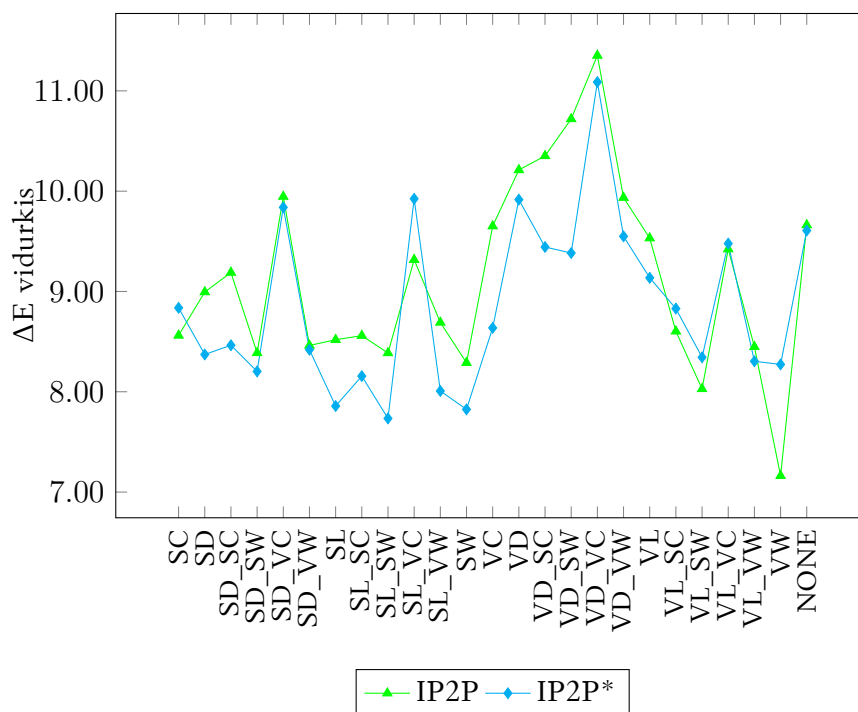
Iš 4 ekspozicijos nuokrypių, 4 baltos spalvos balanso nuokrypių sudarytos 16 jungtinių ekspozicijos ir baltos spalvos balanso nuokrypių variacijos. Įskaičiavus nuotrauką, kuri nuokrypių neturi ir iš viso yra 25 skirtingos nuokrypių variacijos kiekvienai nuotraukai. Iš viso yra 75 nuotraukos iš 3 aibių.

Šie nuokrypiai nuotraukoms buvo pritaikyti naudojant „Adobe Lightroom Classic“ nuotraukų redagavimo programą. Konkretūs programos nustatymai kiekvienam nuokrypiui pateikti 1 priede.

### 3.6. Rezultatų palyginimas

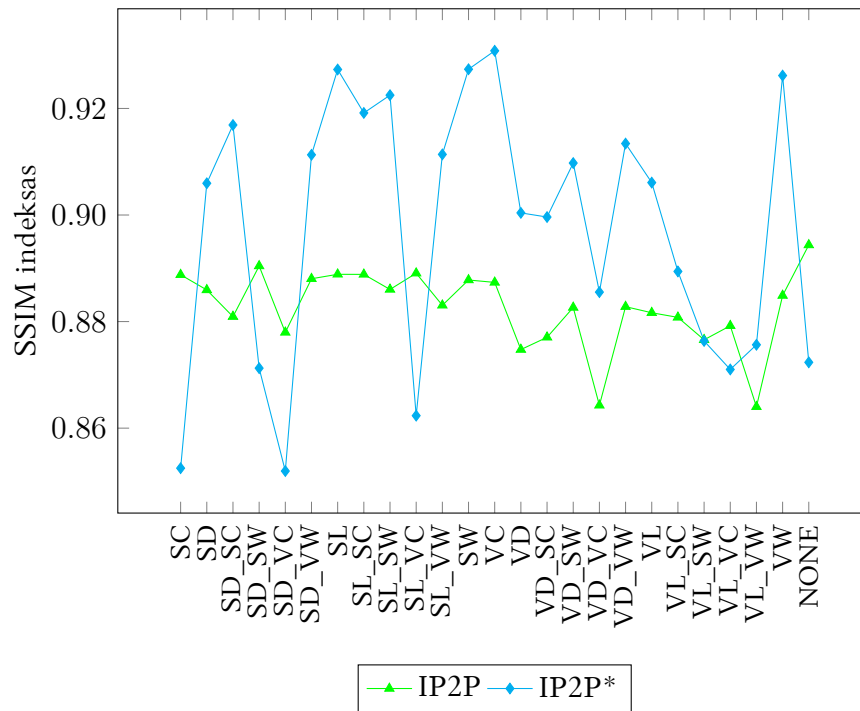
#### 3.6.1. InstructPix2Pix modelio neredaguotų ir redaguotų išvesčių palyginimas

Visų pirma palyginkime rezultatus tarp neredaguotų ir redaguotų InstructPix2Pix modelio išvesčių.



19 pav. Bendras InstructPix2Pix neredaguotų ir redaguotų išvesčių  $\Delta E$  vidurkio vertinimas kiekvienai nuokrypių variacijai.

Pagal  $\Delta E$  vidurkio vertinimą rezultatai ganėtinai panašūs tarp abiejų, neredaguotų ir redaguotų išvesčių. Tačiau galima pastebėti, kad bendru atveju, redaguoti vaizdai rodo šiek tiek geresnius rezultatus nei neredaguoti. Šį skirtumą galima paaiškinti tuo, kad neredaguotose nuotraukose atsiradę iškraipymai, stipriai pakeičia atitinkamų pikselių reikšmes, todėl ištaisius šiuos iškraipymus bendras rezultatas pagerėja



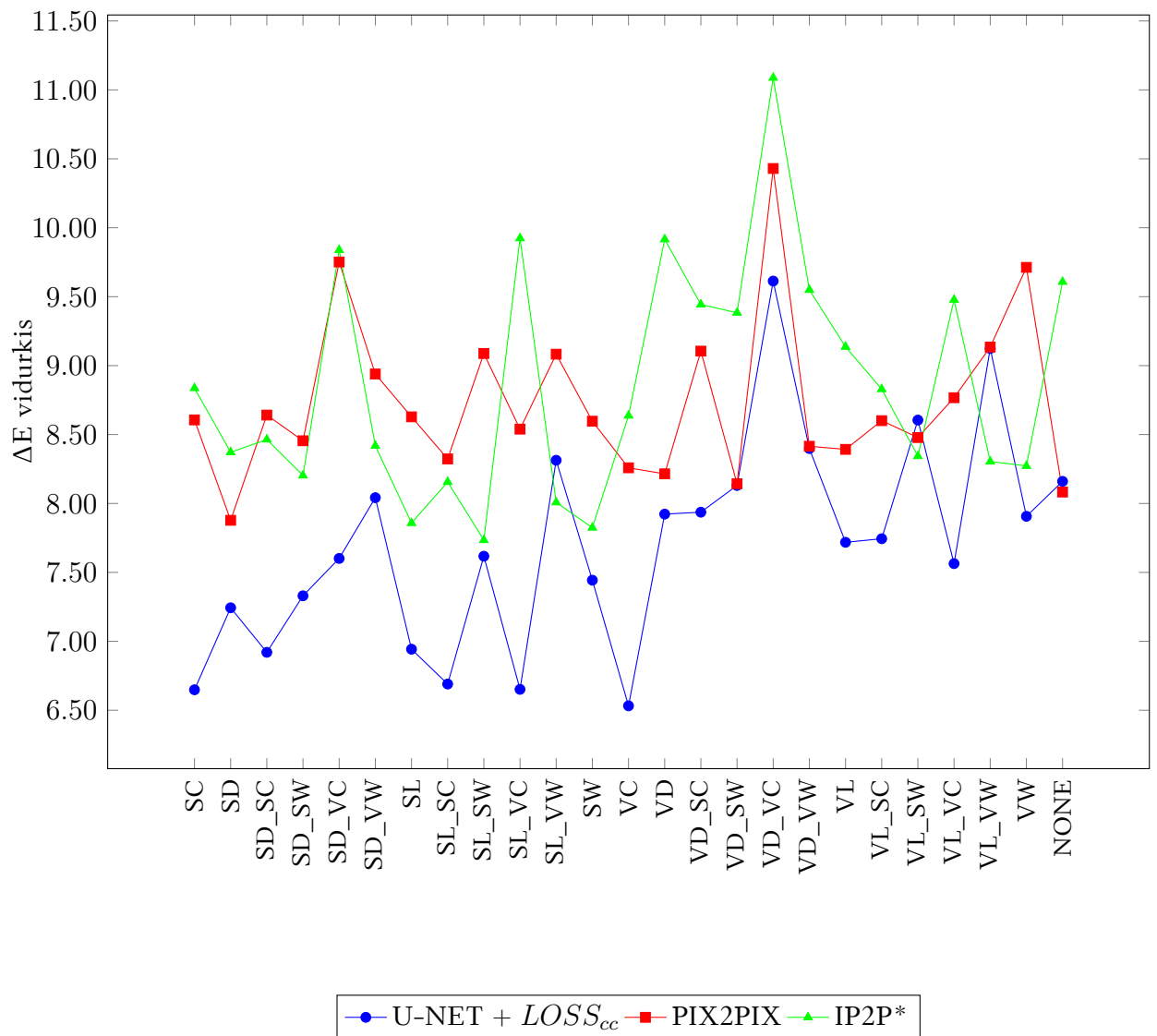
20 pav. Bendras InstructPix2Pix neredaguotų ir redaguotų išvesčių SSIM kiekvienai nuokrypių variacijai.

Toliau palyginkime neredaguotų ir redaguotų vaizdų rezultatus pagal SSIM indeksą. Nors svyravimai tarp rezultatų skiriasi tik per 0.07, tačiau galime pastebėti, kad dažnu atveju redaguoti vaizdai rodo geresnius rezultatus, bet tam tikrais atvejais SSIM indekso reikšmės redaguotų vaizdų rezultatai yra prastesni nei neredaguotų. Tai galima paaiškinti tuo, kad sujungiant originalaus vaizdo aukšto dažnio dalį su modelio sugeneruoto vaizdo žemo dažnio dalimi, šiek tiek pakinta vaizdo kontrastas, dėl kurio tam tikrais atvejais nežymiai nukrenta SSIM indeksas.

Iš abiejų šių palyginimų pagal  $\Delta E$  vidurkio ir SSIM metrikas, galime daryti išvadą, kad redaguotas vaizdas yra arčiau etalono, todėl kituose palyginimuose bus naudojami tik redaguotų vaizdų rezultatai.



### 3.6.2. Modelių palyginimas tarpusavyje



21 pav. Bendras kiekvieno metodo  $\Delta E$  vidurkio vertinimas kiekvienai nuokrypių variacijai. U-NET +  $LOSS_{cc}$  - eksperimento metu sukurto modelio rezultatai, PIX2PIX - apmokyto Pix2Pix GAN modelio rezultatai, IP2P\* - apmokyto InstructPix2Pix modelio redaguotų vaizdų rezultatai

Visų pirma palyginkime visų modelių rezultatus tarpusavyje pagal  $\Delta E$  vidurkį. Iškart galime pastebėti, kad visi trys modeliai savo blogiausių rezultatų parodė su labai tamsios ekspozicijos ir labai mėlyno baltos spalvos balanso nuokrypio variacija (**VD\_VC**).

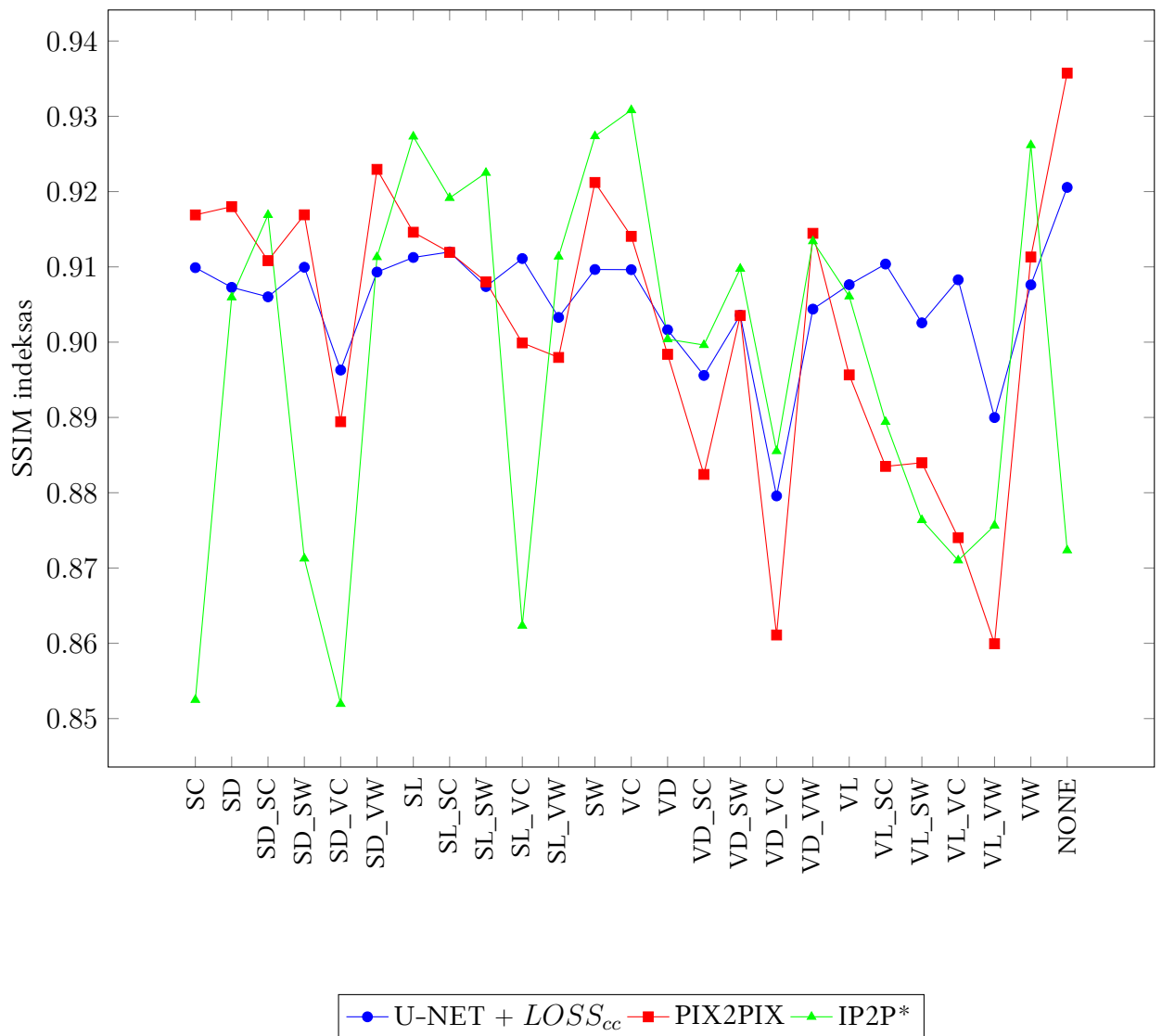
Tiek InstructPix2Pix modelio sugeneruotas ir redaguotas vaizdas, tiek Pix2Pix GAN modelio sugeneruotas vaizdas, sunkiau susitvarkė su šiek tiek tamsesnės ekspozicijos ir labai mėlyno baltos spalvos balanso nuokrypio variacija (**SD\_VC**). Taip pat InstructPix2Pix modelio sugeneruotas ir redaguotas vaizdas parodė vienus iš savo prasčiausių rezultatų su šiek tiek šviesesnės ekspozicijos ir labai mėlyno spalvos balanso (**SL\_VC**), bei su labai tamsios ekspozicijos (**VD**) nuokrypių variacijomis. Galima pastebėti, kad su variacijomis, kurių viena iš sudedamųjų yra labai tamsios ekspozicijos nuokrypio variacija (**VD**) ir/arba labai mėlynos baltos spalvos balanso nuokrypio variacija (**VC**), InstructPix2Pix modelio sugeneruotų ir redaguotų vaizdų metodas

parodė prasčiausius rezultatus palyginus su kitais savo rezultatais. Todėl galime daryti išvadą, kad InstructPix2Pix modelis sunkiausiai susitvarko su labai tamsiais ir labai mėlynais vaizdais, spalvų tikslumo atžvilgiu.

Tuo tarpu žiūrint į Pix2Pix modelio rezultatus, sunku pastebėti kažkokių tendencijų, su kuriomis nuokrypių variacijomis šis metodas rodytų stabiliai prastesnius rezultatus. Rezultatai tarp visų variacijų gana panašūs, išskyrus 3 atvejus (**SD\_SW**, **VD\_VC**, **VW**), kai šis metodas parodė išskirtinai prastesnius rezultatus lyginant su kitais šio metodo rezultatais.

Eksperto metu sukurtas modelis savo prasčiausius rezultatus parodė su tomis nuokrypių variacijomis, kur viena iš nuokrypio sudedamųjų yra labai oranžinis baltos spalvos balansas (**VW**), todėl galima daryti išvadą, kad eksperto metu sukurtas modelis prasčiausiai, spalvų tikslumo atžvilgiu, susitvarko su labai oranžinį baltos spalvos balansą turinčiais vaizdais. Tačiau šis metodas rodė labai gerus rezultatus tais atvejais, kai viena iš nuokrypio sudedamųjų yra šiek tiek šviesesnė arba tamsesnė ekspozicija ir šiek tiek mėlynas baltos spalvos balansas.

Žiūrint į rezultatus bendru atveju, matome, kad eksperto metu sukurto modelio rezultatai dažniausiai rodo geriausius rezultatus tarp visų trijų metodų. Tuo tarpu InstructPix2Pix modelio rezultatai dažniausiai yra patys prasčiausi tarp visų trijų metodų. Pix2Pix GAN modelis nors ir rodo stabiliausius rezultatus tarp visų metodų, tačiau ne geresnius nei eksperto metu sukurto modelio.



22 pav. Bendras kiekvieno metodo SSIM indekso vertinimas kiekvienai nuokrypių variacijai.

Toliau modeliai lyginami pagal SSIM indeksą. Rezultatai tarp visų metodų salyginai arti 1, todėl galima daryti išvadą, kad visi trys metodai išlaiko vaizdo struktūrą, šviesumą bei kontrastą.

Kaip ir vertinime pagal  $\Delta E$  vidurkį, visi trys metodai parodė labai prastus rezultatus su labai tamsios ekspozicijos ir labai mėlynos baltos spalvos balanso nuokrypio variacija (**VD\_VC**). Taip pat visi trys metodai parodė gan prastus rezultatus su šiek tiek tamsesnės ekspozicijos ir labai mėlyno baltos spalvos balanso nuokrypio variacija (**SD\_VC**).

InstructPix2Pix modelio sugeneruotų ir redaguotų vaizdų atveju, rezultatų kreivė labai nestabili. Tam tikrais atvejais šis metodas rodo pačius geriausius rezultatus tarp visų trijų metodų, tačiau kitais atvejais rodo pačius prasčiausius rezultatus. Tai galima paaiškinti dėl nuotraukų redagavimo ištaisant modelio sugeneruotus iškreipimus. Kadangi kopijuojama aukšto dažnio reikšmė iš modelio nesugeneruoto vaizdo, tam tikrais atvejais SSIM vertė yra didesnė palyginti su kitais dviem metodais. Tačiau kitais atvejais kopijuota aukšto dažnio informacija daro įtaką vaizdo kontrastui, dėl ko SSIM indeksas yra žemesnis.

Pix2Pix GAN ir eksperimento metu sukurto modelio rezultatų kreivės stabilesnės nei InstructPix2Pix modelio sugeneruotų ir redaguotų vaizdų metodo. Tačiau eksperimento metu

sukurto modelio rezultatai, nors ir ne visada geresni, tačiau vis tiek stabilesni nei Pix2Pix GAN modelio. Didelis skirtumas tarp šių dviejų metodų matomas kai viena iš nuokrypio sudedamųjų yra labai šviesi ekspozicija (**VL**). Todėl ir vertinime pagal SSIM indeksą geriausius rezultatus parodė eksperimento metu sukurtas modelis.

Kadangi tiek vertinime pagal  $\Delta E$  vidurkį, tiek pagal SSIM indeksą geriausius rezultatus parodė eksperimento metu sukurtas modelis, galime daryti išvadą, kad pritaikius modelio nuostolių funkciją specifiniam ekspozicijos ir baltos spalvos balanso korekcijos uždaviniui, modelis parodo geresnius rezultatus nei bendros paskirties priešiškių generatyvinių neuroninių tinklų ir difuzijos modeliai.

Taip pat svarbu paminėti, kad eksperimento metu sukurtas modelis ir Pix2Pix GAN modelis yra konvoliuciniai modeliai, o InstructPix2Pix jau kitos – difuzijos klasės modelis. Tiek eksperimento metu sukurtas, tiek Pix2Pix GAN modelis parodė geresnius rezultatus nei InstructPix2Pix, todėl taip pat galima daryti išvadą, kad ekspozicijos ir baltos spalvos balanso korekcijos uždavinyje, konvoliuciniai tinklai rodo geresnius rezultatus nei difuzijos modeliai.

Taip pat buvo palygintas modelių vieno vaizdo sugeneravimo vidutinis greitis naudojant MacBook Pro kompiuterį su M1 Pro procesoriumi:

2 lentelė. Modelių vienos nuotraukos vidutinis generavimo greitis

$U\text{-NET} + LOSS_{cc}$ $\sim 0,3$ s.	Pix2Pix GAN $\sim 0,6$ s.	InstructPix2Pix $\sim 12,7$ s.
---	------------------------------	-----------------------------------

Šiame palyginime aiškiai matomas vienas esminis skirtumas tarp difuzijos ir konvoliucinių modelių. Pix2Pix GAN ir eksperimento metu sukurto modelio vieno vaizdo generavimo greitis įprastai nesiekia nei 1 sekundės. Tuo tarpu InstructPix2Pix, vieno vaizdo generavimas su 10 žingsnių, trunka daugiau nei 12 sekundžių.

Kitas svarbus aspektas yra tai, kad nors ir Pix2Pix GAN modelis neiškraipo vaizdo aukšto dažnio informacijos, tačiau šviesesnėse vaizdo vietose sugeneruoja artefaktus.



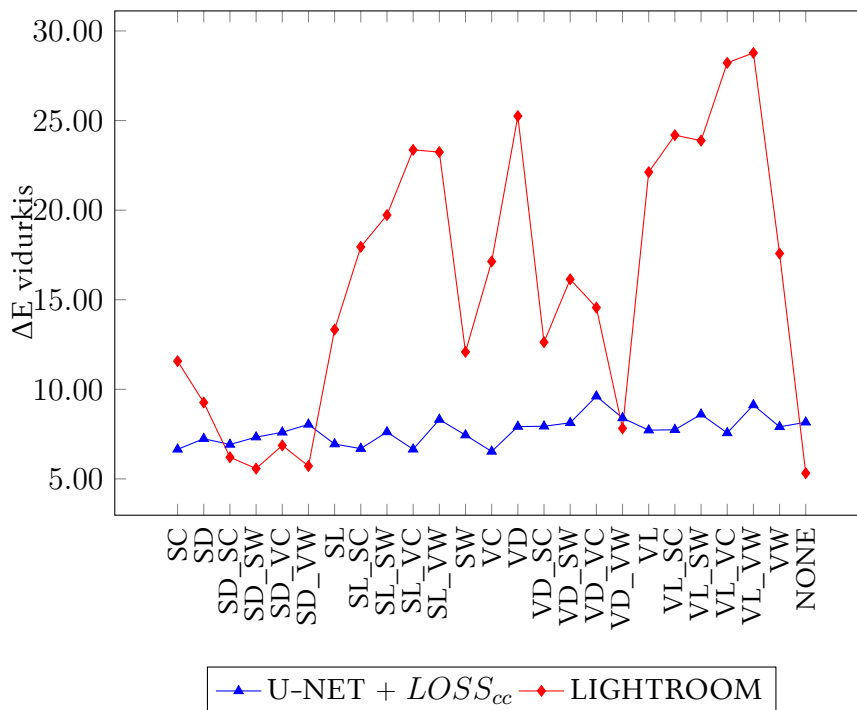
23 pav. Pix2Pix GAN modelio sugeneruotų artefaktų pavyzdys

Žiūrint į sugeneruotus vaizdus, iš pirmo žvilgsnio InstructPix2Pix modelio redaguotų

nuotraukų gali pasirodyti geresni nei kitų dviejų modelių. Tačiau InstructPix2Pix dažnai generuoja tamsesnius vaizdus, labiau kontrastingus vaizdus. Kadangi CIEDE2000 formulė jautri spalvų šviesumų skirtumui, todėl vertinimas pagal  $\Delta E$  vidurkį gali suprastėti. Sugeneruotų vaizdų pavyzdžiai pateikti 3 priede.

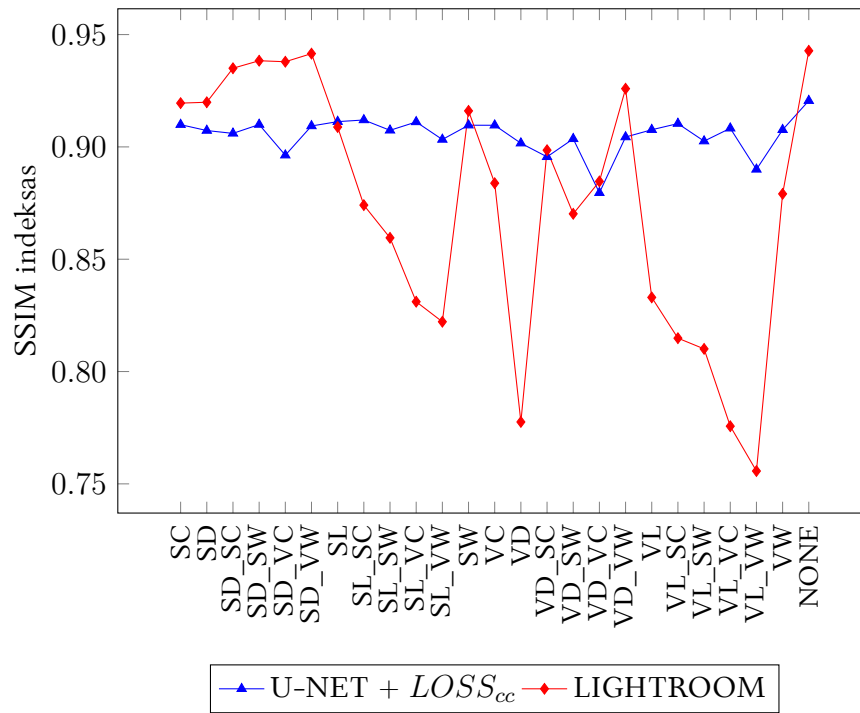
### 3.6.3. Eksperimento metu sukurtu modelio ir Adobe Lightroom Classic palyginimas

Kadangi geriausiai iš visų trijų metodų, nors ir labai nežymiai, geriausius rezultatus parodė eksperimento metu sukurtas modelis, palyginkime jį su „Adobe Lightroom Classic“ nuotraukų generavimo programinės įrangos siūlomu automatinės ekspozicijos ir baltos spalvos balanso korekcijos įrankiu, kuris nenaudoja neuroninių tinklų.



24 pav. Eksperimento metu sukurtu modelio ir Adobe Lightroom Classic siūlomo automatinės ekspozicijos ir baltos spalvos balanso korekcijos įrankio bendri vertinimai pagal  $\Delta E$  vidurkį.

Palyginus su „Adobe Lightroom Classic“, eksperimento metu sukurtas modelis, nors ir keliais atvejais pasirodė šiek tiek prasčiau, tačiau dažniausiai rodo žymiai stabilesnius ir geresnius rezultatus spalvų tikslumo atžvilgiu.



25 pav. Eksperimento metu sukurto modelio ir Adobe Lightroom Classic siūlomo automatinės ekspozicijos ir baltos spalvos balanso korekcijos įrankio bendri vertinimai pagal SSIM indeksą.

Lygiai taip pat kaip ir šių dviejų modelių lyginime pagal  $\Delta E$  vidurkį, lyginant su „Adobe Lightroom Classic“, eksperimento metu sukurtas modelis, nors kartais rodo šiek tiek prastesnius rezultatus, tačiau bendru atveju rodo žymiai stabilesnius ir geresnius rezultatus.

## Rezultatai ir išvados

Šiame darbe buvo apžvelgti Generatyvinių priešišku neuroninių tinklų ir difuzijos modeliai skirti vaizdų transformavimui, apžvelgta jų architektūra, veikimo bei mokymo procesai. Darbo metu buvo apmokyti Pix2Pix GAN ir InstructPix2Pix modeliai atlikti automatinę ekspozicijos ir baltos spalvos balanso korekciją. Taip pat buvo sukurtas modelis su specialiai ekspozicijos ir baltos spalvos balanso korekcijai pritaikyta nuostolių funkcija. Visų šių trijų metodų rezultatai buvo palyginti tarpusavyje, o geriausiai pasirodęs modelis palygintas su „Adobe Lightroom Classic“ siūlomu automatinės ekspozicijos ir baltos spalvos balanso korekcijos įrankiu.

### Darbo rezultatai:

1. Sukurtas poruotas duomenų rinkinys skirtas apmokyti modelius atlikti automatinę ekspozicijos ir baltos spalvos balanso korekciją.
2. Apmokyti Pix2Pix GAN ir InstructPix2Pix modeliai atlikti automatinę ekspozicijos ir baltos spalvos balanso korekciją naudojant sukurtą duomenų rinkinį.
3. Sukurta nuostolių funkcija pritaikyta ekspozicijos ir baltos spalvos balanso korekcijai, bei su ja apmokytas U-NET architektūros modelis, kuris kuris atlieka automatinę ekspozicijos ir baltos spalvos balanso korekciją geriau ir stabiliau nei neuroninių tinklų nenaudojantis „Adobe Lightroom Classic“ automatiniai įrankiai.
4. InstructPix2Pix modelis sunkiausiai susitvarko su labai tamsios ekspoziciją ir labai mėlyno baltos spalvos balansą turinčiais vaizdais.
5. Eksperimento metu sukurtas modelis sunkiausiai susitvarko su labai oranžinį baltos spalvos balansą turinčiais vaizdais.

### Darbo išvados:

1. Neuroninių tinklų modeliai gali būti pritaikyti atlikti automatinę ekspozicijos ir baltos spalvos balanso korekciją.
2. Difuzijos modeliai vaizdų transformavimo uždaviniuose iškraipo pradinio vaizdo aukšto dažnio informaciją.
3. Konvoliuciniai neuroniniai tinklai, palyginus su difuzijos modeliais, geriau atlieka automatinę ekspozicijos ir baltos spalvos balanso korekciją ir neiškraipo pradinio vaizdo aukšto dažnio informacijos.
4. Neuroninių tinklų modeliai gali atlikti automatinę ekspozicijos ir baltos spalvos balanso korekciją žymiai geriau ir stabiliau nei neuroninių tinklų nenaudojantys įrankiai.
5. Geriausi ekspozicijos ir baltos spalvos balanso korekcijos rezultatai su sukurta nuostolių funkcija pritaikyta būtent šiam uždaviniui.
6. Konvoliuciniai neuroniniai tinklai vaizdą sugeneruoja ženkliai greičiau nei difuzijos modeliai

## Šaltiniai

- [AB19] M. Afifi, M. S. Brown. What Else Can Fool Deep Learning? Addressing Color Constancy Errors on Deep Neural Network Performance. Iš: *The IEEE International Conference on Computer Vision (ICCV)*. 2019.
- [AB20] M. Afifi, M. S. Brown. Deep White-Balance Editing. Iš: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.
- [ADO<sup>+</sup>21] M. Afifi, K. G. Derpanis, B. Ommer, M. S. Brown. Learning Multi-Scale Photo Exposure Correction. Iš: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021.
- [Ash24] S. K. Ashvanth.S. *A Comprehensive Guide to Stable Diffusion Parameters for Image Generation* [<https://blog.segmind.com/a-comprehensive-guide-to-stable-diffusion-parameters-for-image-generation/>]. 2024.
- [AT23] A. Asperti, V. Tonelli. Comparing the latent space of generative models. *Neural Computing and Applications*. 2023, tomas 35, numeris 4, p. 3155–3172.
- [BHE23] T. Brooks, A. Holynski, A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. Iš: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, p. 18392–18402.
- [BKG23] D. Bank, N. Koenigstein, R. Giryes. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*. 2023, p. 353–374.
- [fac22] H. face. *Image-to-image — huggingface.co* [<https://huggingface.co/docs/diffusers/using-diffusers/img2img>]. 2022.
- [GPM<sup>+</sup>14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*. 2014, tomas 27.
- [HJA20] J. Ho, A. Jain, P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*. 2020, tomas 33, p. 6840–6851.
- [HLJ<sup>+</sup>23] S. Hong, G. Lee, W. Jang, S. Kim. *Improving Sample Quality of Diffusion Models Using Self-Attention Guidance*. 2023.
- [IZZ<sup>+</sup>17] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros. Image-to-image translation with conditional adversarial networks. Iš: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 1125–1134.
- [KW13] D. P. Kingma, M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*. 2013.
- [LCR01] M. Luo, G. Cui, B. Rigg. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research Application*. 2001, tomas 26, p. 340–350. Prieiga per internetą: <https://doi.org/10.1002/col.1049>.



- [MO14] M. Mirza, S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*. 2014.
- [NA20] J. Nilsson, T. Akenine-Möller. Understanding ssim. *arXiv preprint arXiv:2006.13846*. 2020.
- [RBL<sup>+</sup>22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer. High-resolution image synthesis with latent diffusion models. Iš: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 10684–10695.
- [RFB15] O. Ronneberger, P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation. Iš: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*. Springer, 2015, p. 234–241.
- [RKH<sup>+</sup>21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh ir kiti. Learning transferable visual models from natural language supervision. Iš: *International conference on machine learning*. PMLR, 2021, p. 8748–8763.
- [RPG<sup>+</sup>21] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever. *Zero-Shot Text-to-Image Generation*. 2021.
- [SWM<sup>+</sup>15] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. Iš: *International conference on machine learning*. PMLR, 2015, p. 2256–2265.
- [VSP<sup>+</sup>17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*. 2017, tomas 30.
- [Zen18] A. Zeng. Frequencies and Gradients [<https://inst.eecs.berkeley.edu/~cs194-26/fa17/upload/files/proj3/cs194-26-aec/>]. 2018.

## Priedai

### Priedas nr. 1. „Adobe Lightroom Classic“ programos nustatymai kiekvienai nuokrypio variacijai

Kiekvienai nuokrypių variacijai testavimo vaizdų aibėje buvo pritaikyti tokie „Adobe Lightroom Classic“ programos nustatymai:

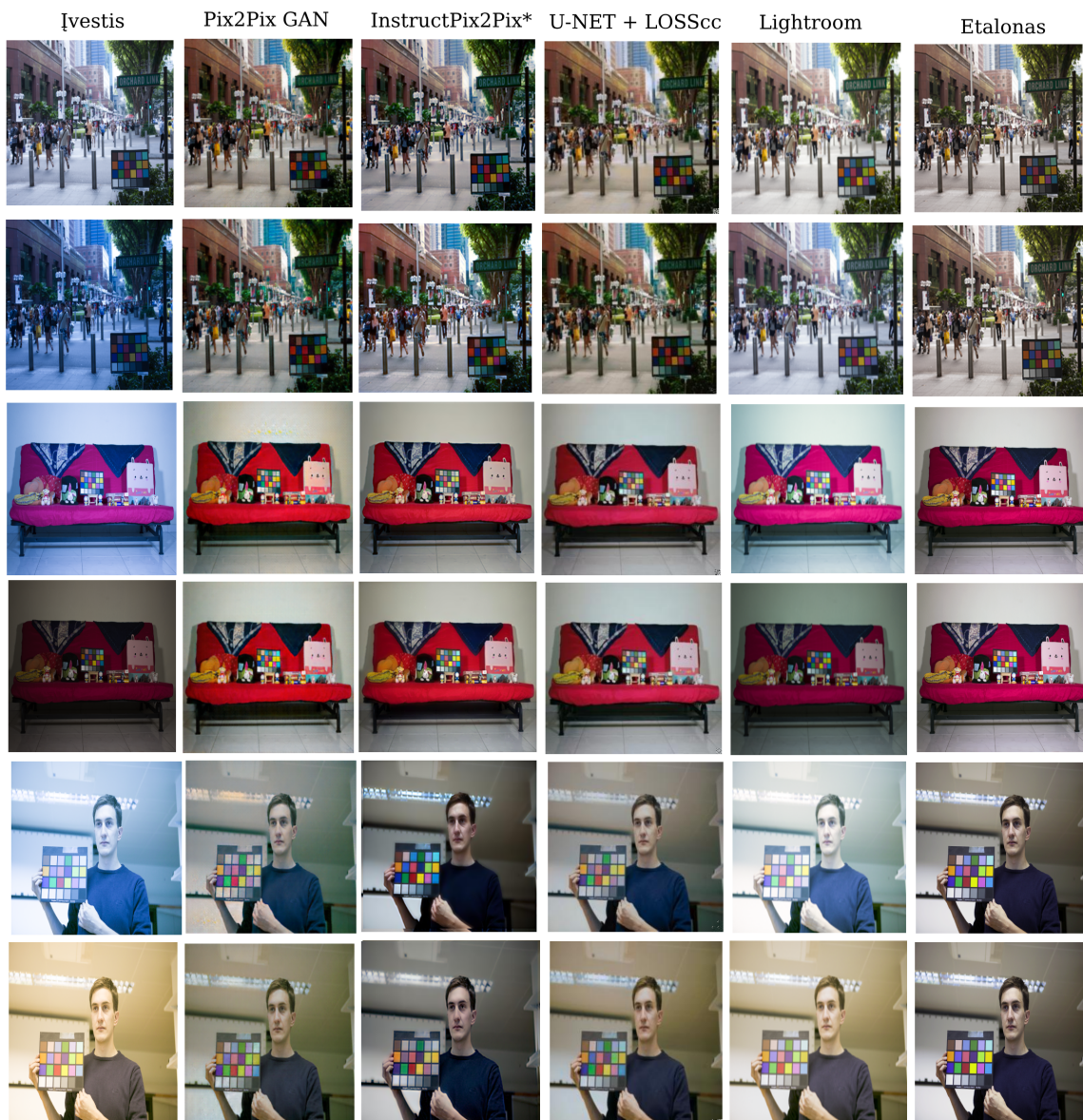
- **SD** (šiek tiek per tamsi) - *Exposure* -1
- **SL** (šiek tiek per šviesi) - *Exposure* +1
- **VD** (labai tamsi) - *Exposure* -2
- **VL** (labai šviesi) - *Exposure* +2
- **SC** (šiek tiek mėlyna) - *Temperature* -25
- **SW** (šiek tiek oranžinė) - *Temperature* +25
- **VC** (labai mėlyna) - *Temperature* -50
- **VW** (labai oranžinė) - *Temperature* +50

## **Priedas nr. 2. Programinis kodas**

Eksperimento metu sukurto modelio programinį kodą galima rasti šia nuoroda:

[https://github.com/redasSer/VU-MIF\\_bachelors\\_color-correction-model](https://github.com/redasSer/VU-MIF_bachelors_color-correction-model)

### Priedas nr. 3. Sugeneruotų vertinimo vaizdų pavyzdžiai



26 pav. Sugeneruotų vertinimo vaizdų pavyzdžiai