

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Biometrické systémy 2023 / 24

Bc. Petr Pouč - xpoucp01
Bc. Jan Polišenský - xpolis04

Obsah

| | | |
|----------|-------------------------------------------|----------|
| 1 | Úvod | 2 |
| 2 | Generování syntetických snímků žil | 3 |
| 2.1 | Předzpracování Snímků | 3 |
| 2.1.1 | Normalizace a Standardizace | 3 |
| 2.1.2 | Zvýraznění Žil Prstů | 4 |
| 2.1.3 | Optimalizace Rozlišení | 4 |
| 2.2 | Zpracování štítků | 4 |
| 2.3 | Implementace soupeřících sítí | 5 |
| 2.3.1 | Generátor | 5 |
| 2.3.2 | Diskriminátor | 5 |
| 2.4 | Zhodnocení výsledků | 7 |
| 3 | Závěr | 9 |

Kapitola 1

Úvod

Tento projekt si klade za cíl vytvořit generátor syntetických snímků žil prstu pomocí metod strojového učení. Pro dosažení tohoto cíle bude využit poskytnutý dataset¹. Prvním krokem bylo vytvoření syntetických snímků, které budou následně sloučeny do nové kolekce. Tato nová kolekce bude podrobena sérii testů, a výsledky budou důkladně ověřeny a zkontrolovány.

¹<https://strade.fit.vutbr.cz/data/s/ZiMQ4fMkbHjS9dS?path=%2FBIO2021-Data%2Fdata>

Kapitola 2

Generování syntetických snímků žil

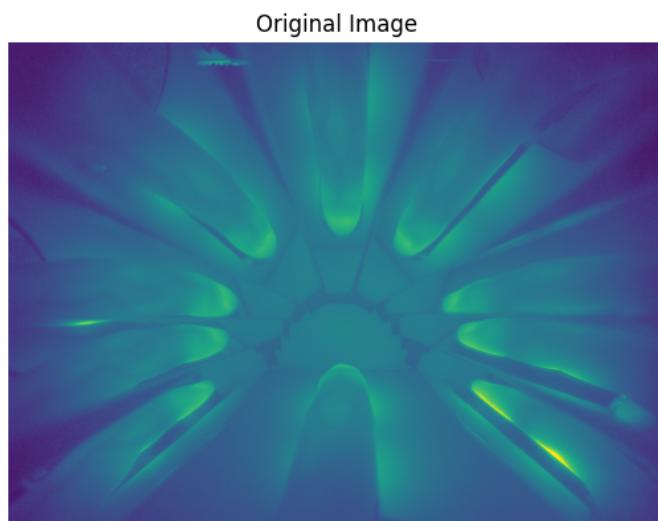
2.1 Předzpracování Snímků

Předzpracování dat, v našem případě snímků žilního řečiště prstu, je nezbytný krok před následnou generací snímků syntetických. Tento krok zabezpečuje, že jsou data ve vhodném formátu a optimalizována pro zvolenou metodu.

Předzpracování zajistí nejen kvalitní, ale také mnohem rychlejší proces generování syntetických snímků. Předzpracování dat pomáhá eliminovat nečistoty, artefakty a nesrovnatnosti ve snímcích žil.

Samotný proces úpravy snímků se nachází v souboru `compiler.py` a obsahuje kroky normalizace obrazu, aplikace histogramové ekvalizace, změny rozlišení, korekci kontrastu a jasu, a zajištění jednotné kvality v celém datasetu.

Následuje předzpracování štítků, které jsou kódovány pro použití ve strojovém učení - kategorická data jsou zakódována pomocí `OneHotEncoder` a `LabelEncoder`, zatímco numerická data jsou připravena pro další zpracování.



Obrázek 2.1: Originální snímek z poskytnuté datové sady

2.1.1 Normalizace a Standardizace

Normalizace a standardizace jsou klíčové pro zajištění konzistence a srovnatelnosti mezi snímkami. Tímto způsobem jsou eliminovány rozdíly ve světelných podmínkách, rozlišení a barevném podání. Předzpracování

zajišťuje, že všechny snímky mají podobné charakteristiky, což usnadňuje generaci snímků s homogenním vizuálním vzhledem.

Škálování je provedeno tak, aby se hodnota pixelů pohybovala od -1 do 1. Tento krok je důležitý pro stabilní, efektivní a spolehlivé trénování modelů strojového učení a umožňuje lépe zpracovat vstupní data tak, aby modely lépe generalizovaly na nová data a dosahovaly lepších výsledků.

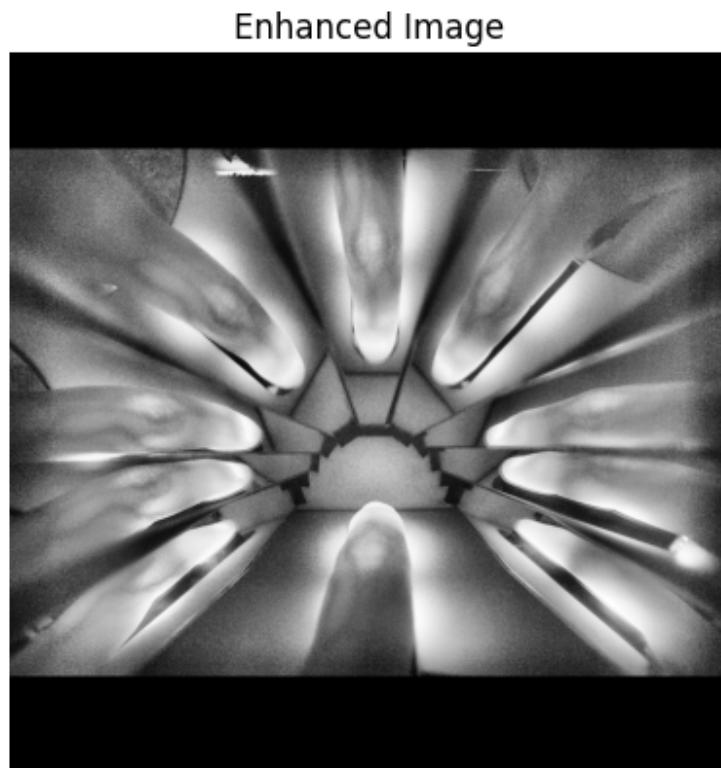
2.1.2 Zvýraznění Žil Prstů

Zpracování dat může zahrnovat identifikaci konkrétních regionů na snímcích, které jsou klíčové pro generaci. V případě snímků žil to může znamenat důkladné vyznačení oblastí obsahujících žilové struktury. Tímto způsobem může generativní model lépe soustředit svou pozornost na relevantní části snímku.

Pro zvýraznění žil prstů používáme například histogramová ekvalizace, což pomáhá zlepšit kontrast obrazu a zvýraznit detaily.

2.1.3 Optimalizace Rozlišení

Rozlišení jednotlivých snímků jsme upravili na velikost 1024x1024, což urychlilo a zjednodušilo samotný proces generování syntetických snímků.



Obrázek 2.2: Předzpracovaný snímek žil otisku prstů

2.2 Zpracování štítků

Funkcionalita tohoto kroku se nachází ve funkci `preprocess_labels`. Funkce identifikuje kategorické atributy (například pohlaví, kuřák a další), které je třeba transformovat pro použití ve strojovém učení. Tato data jsou transformována pomocí metody One-Hot Encoding, což převede kategorické atributy do binární formy. Tato forma je běžná a nejvhodnější pro strojové učení.

2.3 Implementace soupeřících sítí

Pro samotné generování syntetických snímků žil jsme zvolili přístup pomocí GAN – GAN – Generative Adversarial Networks (soupeřící sítě). GAN se skládá ze dvou hlavních částí: generátoru a diskriminátoru, které spolu soupeří.

2.3.1 Generátor

Generátor v našem případě vytváří syntetické snímky žil, založené na náhodném šumu a podmíněné na dodaných datech, která jsou doplněná o štítky (v kódu označované jako labels), jako jsou věk, pohlaví, melanin a další). Cílem generátoru je vytvořit realistické snímky, které je nemožné odlišit od snímků skutečných.

2.3.2 Diskriminátor

Diskriminátor funguje jako klasifikátor, který se snaží rozlišit syntetické snímky vygenerované generátorem od skutečných snímků. Jeho cílem je co nejpřesněji rozpoznat, které snímky jsou skutečné a které jsou generované.

Tyto dvě části jsou trénovány současně v opakujících se cyklech. Sítě se tedy navzájem zlepšují a díky tomu dosahují lepších výsledků - věrohodnějších snímků žil prstů.

```
class Generator():

    # Limit the number of images for testing
    SAMPLE_LIMIT = 1000

    # Loading preprocessed images and their labels
    # ... Code

    # Initialization of the generator and specification of image creation parameters
    z_dim = 10    # Latent space dimension
    generator = Generator(z_dim, preprocessed_labels.shape[1], (1024, 1024, 1))    # Image
    ↵ shape specification

    # Initialization of optimizer and loss function for the generator
    generator_optimizer = Adam(learning_rate=0.015)
    mse_loss = MeanSquaredError()

    # Compilation of the generator model
    generator.model.compile(optimizer=generator_optimizer, loss=mse_loss)

    # Saving the generator model
    generator.model.save('generator_model.h5')
    import json
    # Additional code for parallelization and image generation

    # Training loop for generating images
    for i in range(len(preprocessed_images)):
        for j in range(5):
            Generating synthetic images using the generator model
            Adjusting labels, saving the synthetic image to a directory
            Processing labels, saving labels of synthetic images
            # ... Code
```

Kód začíná inicializací diskriminátoru, optimizéru a ztrátové funkce pro trénování generátoru. Generátor a diskriminátor jsou inicializovány pomocí odpovídajících tříd: Generator a Discriminator. Generátor bere šum a podmíněný vstup a generuje obrázek. Diskriminátor bere obrázek a podmínu a vrací skóre, které určuje, zda je obrázek reálný nebo synteticky generovaný. Součástí kódu je také manipulace s popisky vytvořených syntetických obrázků, aby bylo možné sledovat příslušnost a informace o jednotlivých vygenerovaných snímcích.

Během trénování modelu jsou použity předem připravené obrázky (preprocessed_images) a jejich popisky (preprocessed_labels). V cyklu jsou náhodně vybrány předzpracované obrázky a pro každý z nich je provedeno několik iterací generování syntetických obrázků.

Klíčové aspekty architektury soupeřících sítí:

- `z_dim`

Tento parametr představuje rozměr latentního prostoru, což je v podstatě rozměr vektoru náhodného šumu. Čím vyšší `z_dim`, tím větší je prostor pro generování rozmanitějších snímků.

- počet iterací

Počet iterací definuje, kolikrát se provede trénování pomocí daného datasetu. Každá iterace zahrnuje aktualizaci vah generátoru a diskriminátoru.

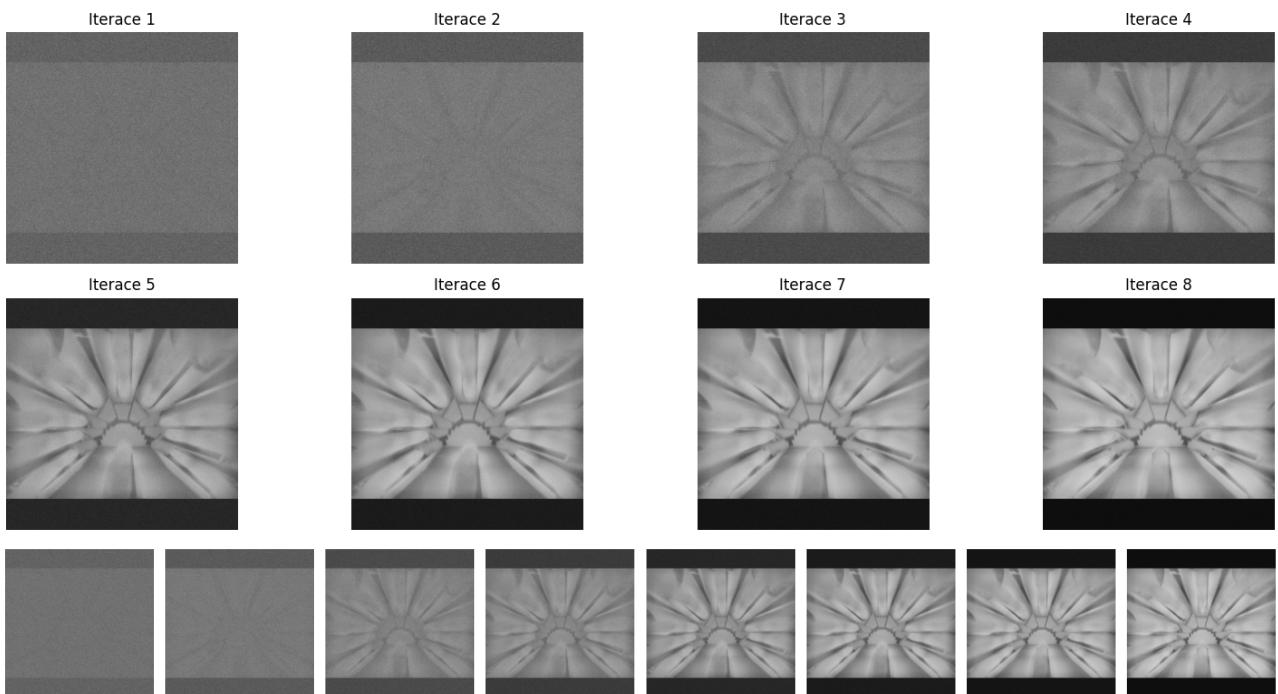
- ztrátová funkce (loss function)

Jako ztrátovou funkci jsme v našel projektu použili střední kvadratickou chybu (Mean Squared Error - MSE). Ztrátová funkce slouží pro měření rozdílu mezi syntetickými a reálnými snímkami. Tato metrika porovnává hodnoty pixelů mezi oběma obrázky a optimalizuje parametry generátoru tak, aby se rozdíl minimalizoval.

- generator_optimizer

Optimalizér slouží k optimalizaci definování vah modelu. V našem případě jsme oužili optimalizér zvaný Adam. Tento optimalizér aktualizuje váhy sítě podle derivací ztrátové funkce. Rychlosť učení (learning rate) jsme nastavili na hodnotu 0.015. Tato hodnota 0.015 určuje, jak velké kroky se při tréninku vah mají podnikat.

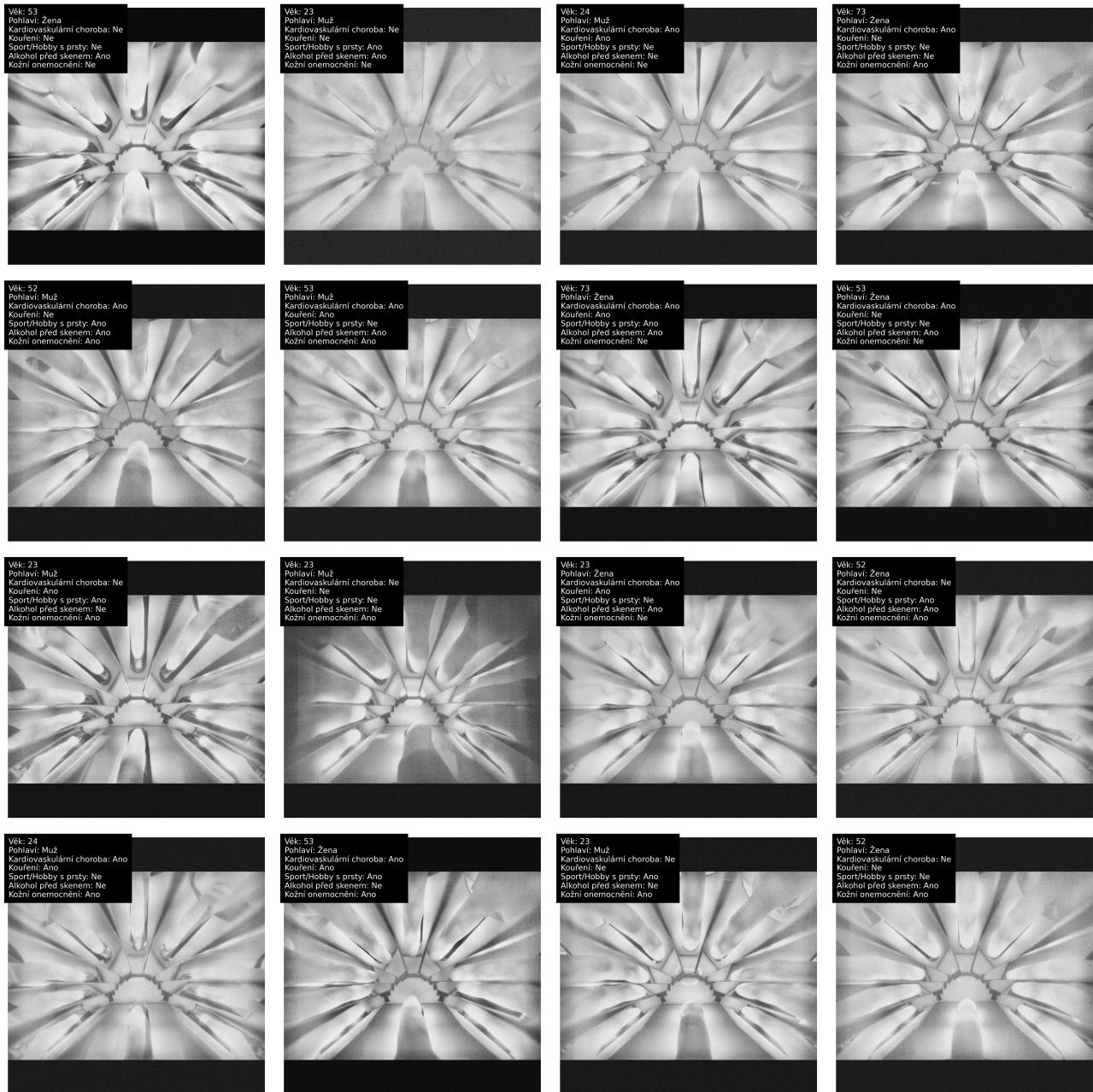
Generátor vytváří syntetické obrázky na základě náhodného šumu.



Obrázek 2.3: Proces učení

2.4 Zhodnocení výsledků

Podařilo se nám celkem vygenerovat více než 100 syntetických snímků žilního řečiště prstů. N anásledující matici lze vidět některé z nich, včetně jejich přiřazených štítků.

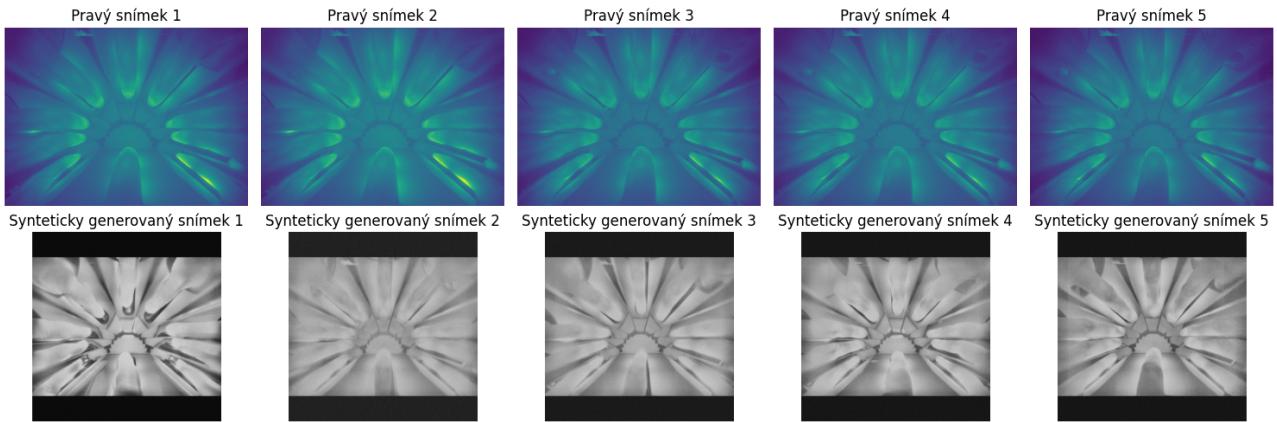


Obrázek 2.4: Synteticky generované snímky žilního řečiště

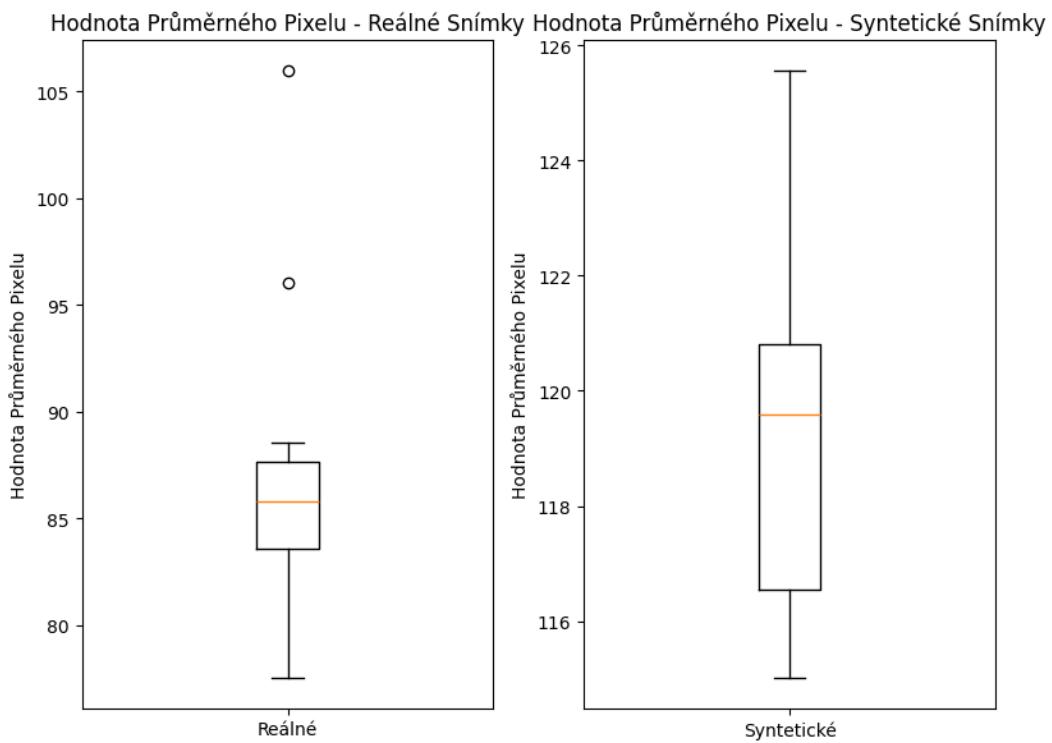
Na první pohled lze konstatovat, že syntetické obrázky žil prstů dosáhly vysoké míry realismu a podobnosti s reálnými snímkami. Generované obrázky navíc obsahují detailní strukturu žil, což je klíčovým aspektem pro jejich použitelnost v různých aplikacích.

Kromě vizuálního porovnání jsme se v rámci hodnocení zaměřili na statistické hodnoty vzniklých snímků, vůči těm originálním. Syntetické obrázky dosáhly v průměru vyšší intenzity pixelů než obrázky reálné (119 x 87).

Směrodatná odchylka reálného obrazu je 29.72, zatímco směrodatná odchylka syntetického obrazu je vyšší a činí 64.03. Vyšší hodnota směrodatné odchylky v syntetickém obrazu naznačuje větší rozptyl hodnot intenzity pixelů od jejich průměru ve srovnání s reálným obrazem. To může značit větší variabilitu hodnot intenzity pixelů v syntetickém obrazu.



Obrázek 2.5: Srovnání pravých a synteticky generovaných snímků



Obrázek 2.6: Srovnání pravých a synteticky generovaných snímků

Kapitola 3

Závěr

Tento projekt se zaměřil na vytvoření generátoru syntetických snímků žil prstu pomocí metod strojového učení, konkrétně využitím soupeřících sítí (GAN - Generative Adversarial Networks). Celkem se nám podařilo úspěšně vygenerovat více než 100 syntetických snímků žilního řečiště prstů. Vizuální analýza ukázala vysokou míru realismu a podobnosti syntetických snímků s reálnými snímky žil prstů. Přes dosažené výsledky v generování syntetických snímků je stále prostor pro optimalizaci a zdokonalení generátoru a diskriminátoru. Například významný rozdíl v průměrné hodnotě a směrodatné odchylce mezi reálnými a syntetickými obrázky naznačuje rozdíly ve struktuře, osvětlení nebo distribuci barev mezi těmito dvěma typy obrázků.