



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**GENEROVÁNÍ HODNOVĚRNÝCH POZADÍ OBRÁZKŮ  
LATENTNÍCH OTISKŮ PRSTŮ**

GENERATION OF AUTHENTIC LATENT FINGERPRINTS BACKGROUND

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ADAM GAJDA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ONDŘEJ KANICH, Ph.D.**

BRNO 2022

## Zadání bakalářské práce



Student: **Gajda Adam**  
Program: Informační technologie  
Název: **Generování hodnověrných pozadí obrázků latentních otisků prstů**  
**Generation of Authentic Latent Fingerprints Background**  
Kategorie: Zpracování obrazu

### Zadání:

1. Prostudujte literaturu týkající se daktyloskopie a snímání otisků prstů, včetně metod a technologií snímání. Seznamte se s generátory syntetických otisků prstů a s možnými poškozeními, které mohou ovlivňovat kvalitu nasnímaného otisku prstu, zejména u latentních snímků.
2. Navrhněte vhodné modely a posléze algoritmy pro generování hodnověrných pozadí snímků otisků prstů, vycházející z latentních záznamů otisků. Použijte minimálně tři odlišné typy pozadí (např. dokument, bankovka, daktyloskopická karta, znečištěný povrch) všechny s různou úrovní poškození. Vzniklé pozadí pak integrujte do snímků syntetických otisků prstů.
3. Implementujte navržený algoritmus z předchozího bodu.
4. Otestujte algoritmus z předchozího bodu vygenerováním databáze syntetických otisků prstů s pozadím s různými nastaveními. Analyzujte autenticitu vzniklých snímků. Ověřte výstupy pomocí algoritmů na určení kvality otisku prstu.
5. Dosažené výsledky shrňte a diskutujte. Uveďte možná rozšíření vašeho řešení.

### Literatura:

- Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer, 2009, p. 512. ISBN 978-1-8488-2254-2.
- Kanich, O.: *Research in Fingerprint Damage Simulations*, Doctoral thesis, FIT BUT in Brno, Brno, 2018, p. 148.
- U.S. Department of Justice: *The Fingerprint Sourcebook*. CreateSpace Independent Publishing Platform, 2014, p. 428. ISBN 978-1502828422.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Kanich Ondřej, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 3. listopadu 2021

## Abstrakt

Tato bakalářská práce se zabývá generováním autentických latentních pozadí otisků prstů pomocí hlubokého učení, konkrétněji pomocí podmíněné generativní adversariální sítě a jiných konvenčnějších metod. Tato práce shrnuje základní teoretické informace o biometrii včetně syntetických otisků prstů a úvodu do umělé inteligence. Hlavní model navržený v této práci se nepovedl kvůli nedostatku unikátních trénovacích dat. Byly diskutovány i další možné důvody. Byl tak vyvinut alternativní způsob generování pozadí latentních otisků prstů a po vizuálním vyhodnocení konečných výsledků a reálných dat byl závěr pozitivní.

## Abstract

This bachelor's thesis deals with the generation of authentic latent fingerprint backgrounds, through the use of deep learning, more specifically with the help of conditional generative adversarial network and other more conventional methods. This work summarizes the basic theoretical information about biometrics including synthetic fingerprints and a introduction into artificial intelligence. The main model proposed in this thesis has not come into fruition due to lack of unique training data. Other possible reasons were discussed. Thus an alternative way of generating latent fingerprint backgrounds was developed and after visual evaluation of the final results and real data the conclusion was positive.

## Klíčová slova

pozadí latentních otisků prstů, generování pozadí otisků prstů, hluboké neuronové sítě, CGAN

## Keywords

latent fingerprint background, generating fingerprint background, deep neural networks, CGAN

## Citace

GAJDA, Adam. *Generování hodnověrných pozadí obrázků latentních otisků prstů*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Kanich, Ph.D.

# Generování hodnověrných pozadí obrázků latentních otisků prstů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Kanich Ondřej, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Adam Gajda  
11. května 2022

## Poděkování

Tímto děkuji vedoucímu práce a těm, kteří mi poskytli morální podporu.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Souhrn dnešních technologií otisků prstů</b>	<b>3</b>
2.1	Úvod do biometrie . . . . .	3
2.2	Daktyloskopie . . . . .	4
2.3	Klasifikace otisků prstů . . . . .	6
2.4	Markanty otisků prstů . . . . .	7
2.5	Z pohledu kriminalistiky . . . . .	7
2.6	Technologie snímání otisků prstů . . . . .	8
2.7	Postup zpracování otisků prstů . . . . .	13
2.8	Syntetický otisk prstu . . . . .	14
<b>3</b>	<b>Umělá inteligence</b>	<b>16</b>
3.1	Strojové učení . . . . .	16
3.2	Hluboké učení . . . . .	17
3.3	Strojové učení versus hluboké učení . . . . .	17
3.4	Umělý neuron . . . . .	18
3.5	Umělá neuronová síť . . . . .	19
3.6	Dopředné neuronové sítě . . . . .	20
3.7	Neuronové sítě autoenkodéru . . . . .	20
3.8	Opakující se neuronové sítě . . . . .	21
3.9	Konvoluční neuronová síť . . . . .	21
<b>4</b>	<b>Návrh generátoru pozadí</b>	<b>22</b>
4.1	Data na trénování modelu . . . . .	22
4.2	Generování nových pozadí . . . . .	23
<b>5</b>	<b>Implementace generátoru a jeho výsledky</b>	<b>25</b>
5.1	Generování za pomoci umělé neuronové sítě . . . . .	26
5.2	Generování za pomoci konvenčních metod . . . . .	32
<b>6</b>	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>38</b>
<b>A</b>	<b>Obsah přiloženého média</b>	<b>41</b>
<b>B</b>	<b>Ukázka pozadí - konvenční metoda</b>	<b>42</b>

# Kapitola 1

## Úvod

Biometrické údaje jsou již z historie velmi populární způsob jak identifikovat jedince. Ku příkladu takový otisk prstu je spolehlivý způsob pro autentizaci a získání takových to otisků není nijak složité. Tyto důvody nám napomáhají porozumět proč jsou v dnešní době otisky prstů jedny z nejpoužívanějších biometrických údajů. S postupem času se ale vyvíjejí pokročilejší a sofistikovanější techniky, které své nároky na množství dat převážně zvyšují a to jak cenou tak i časem, což znamená obzvláště v dnešní době GDPR, že je třeba najít alternativu. Touto alternativou jsou generátory syntetických otisků prstů a tyto generátory potřebují nějaké pozadí na které svůj nově vygenerovaný otisk umístí.

O tomto je tato práce, generování pozadí k syntetickým otiskům prstů, pro upřesnění otiskům latentním - špatně viditelným/částečným. Tohoto jsme schopni dosáhnout za pomocí generativních modelů, v našem případě typ generativní soupeřivé sítě, která je pro takový to úkol ideální.

V kapitole plné teorie 2 bude nastíněna biometrie, kriminalistika, otisky reálné i syntetické, jejich pozadí a poškození. Další kapitolou teorie je 3 umělá inteligence - na co se dělí; neuronová síť - z čeho se skládá a její různé typy. Posléze v kapitole 4, která je kombinace teorie a praktického návrhu generativního modelu. Kapitola 5 bude implementace dříve zmíněného návrhu, implementace alternativy a následné vyhodnocení výsledků. A na závěr 6 bude shrnutí celé práce a případné návrhy na vylepšení do budoucna.

## Kapitola 2

# Souhrn dnešních technologií otisků prstů

V této kapitole budou shrnuty základní znalosti potřebné k pochopení návrhu této práce v kontextu otisků prstů a její následné implementaci. Obecně bude shrnuta biometrie, kde se blíže zaměříme na daktyloskopii a informace s ní spřažené. Posléze si vysvětlíme jak se klasifikují otisky prstů a jaké rozlišujeme markanty (a jak vypadají). Následně se přiblížíme ke kriminalistice kde si řekneme jaké existují typy otisků prstů se zaměřením na otisk latentní a jaké faktory nám ovlivňují kvalitu otisku prstu. Dále si řekneme jakým způsobem se takovéto otisky zaznamenat/snímat. A jako poslední podkapitola nám zbývá syntetické otisky prstů (námi uměle vytvořené) a jejich výroba (generování).

### 2.1 Úvod do biometrie

O biometrii se můžeme bavit v kontextu biologie (taktéž známé jako biostatistika) [33] a hlavně pro nás důležité v kontextu informačních technologií. Biometrie (IT) je definována jako automatizované rozpoznání lidí na základě jejich fyziologických (otisk prstu, obličej, sítnice, ...) a behaviorálních (styl chůze, manýrismus, mimika obličej, ...) vlastností. [9, 19]

Při zpracování biometrických signálů se často setkáváme s problémy jako: segmentace, zašuměný vstup, výkonnost systému (rychlost, náklady, ...), jednoznačnost vlastnosti, fúze vícero vlastností, rozšiřitelnost, útoky na systém, otázky privátních dat a další. [9] Ještě je nutno podotknout mezitřídní a vnitrotřídní variabilitu kde mezitřídní nám určuje jak velký je rozdíl mezi vlastnostmi u rozdílných tříd (e.g. dva lidé vypadající identicky) a vnitrotřídní (e.g. jeden a ten samý člověk vypadá na jiných fotkách odlišně) [9, 19]

#### Charakteristiky biometrických vlastností:

- Univerzalita (U) - každý člověk by tuto vlastnost měl mít
- Jedinečnost (J) - žádní dva lidé by neměli mít tuto vlastnost stejnou
- Konstantnost (K) - daná vlastnost se v čase nemění
- Získatelnost (Z) - daná vlastnost se dá kvantitativně změřit (získat)

- Výkonnost (V) - daná vlastnost by se neměla změnit či zestárnout (ovlivňuje výkonnost)
- Akceptace (A) - ochota lidí nechat si vlastnost změřit
- Zfalšovatelnost (F) - jak složité je danou vlastnost zfalšovat
- Finanční náklady (N) - jak moc hraje cena roli při nasazení systému s touto vlastností
- Údržba - jak moc bude v nákladech hrát roli údržba systému

Tabulka 2.1: Charakteristiky biometrických vlastností (+ vysoká, ÷ střední, - nízká) [9]

Vlastnost	U	J	K	Z	V	A	F	N
Obličej	+	-	÷	+	-	+	-	-
Otisk prstu	÷	+	+	÷	+	÷	+	-
Geometrie ruky	÷	÷	÷	+	÷	÷	÷	÷
Žíly ruky	÷	÷	÷	÷	÷	÷	+	÷
Duhovka oka	+	+	+	÷	+	-	+	+
Sítnice oka	+	+	÷	-	+	-	+	+
Podpis	-	-	-	+	-	+	-	-
Hlas	÷	-	-	÷	-	+	-	-
Termogram	+	+	-	+	÷	+	+	+

## 2.2 Daktyloskopie

Daktyloskopie je nauka o kožních otiscích (přesněji jejich papilárních liniích vyskytující se na prstech, dlaních a chodidlech) a jejich vzájemném porovnávání z důvodu identifikace osob. Nejstarší příklad využití otisků prstů k identifikaci osoby pochází z Číny (221-206 př.n.l.). Této biometrické vlastnosti se často využívá v kriminalistice z důvodu vysoké jedinečnosti a konstantnosti. [17]

### 2.2.1 Papilární linie

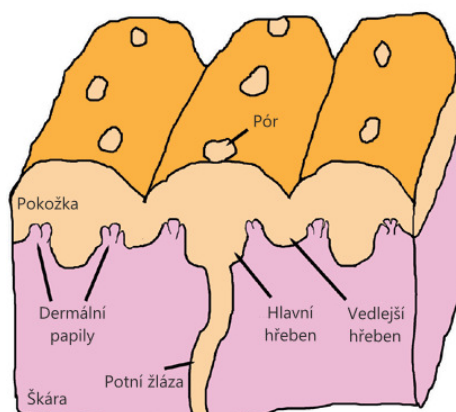
Otisk prstu vznikne při zachycení výběžků na prstu (papilárních linií), které vznikají v pokožce (Epidermis), kde samotný výběžek nám vzniká v hlavním hřebenu (Primary Ridge) a údolí mezi výběžky nám zase vzniká ve vedlejším hřebenu (Secondary Ridge) (viz obrázek 2.1). [19, 17]

Papilární linie jsou složeny z keratinocytů což jsou buňky, které jsou tvořeny z nejnižší vrstvy zvané *stratum basale* a v případě hlavních hřebenů i vrstvy *supra-basal* (viz obrázek 2.2). Tudíž jakékoliv poškození (e.g. popáleniny, odřeniny, říznutí, ...) pokožky nepůsobí žádnou změnu papilárních linií za předpokladu, že nebude poškozena právě vrstva *Stratum Basale*. Pokud však ano, tak se navždy změní struktura otisku prstu v oblasti zranění (viz obrázek 2.3). [19, 17]

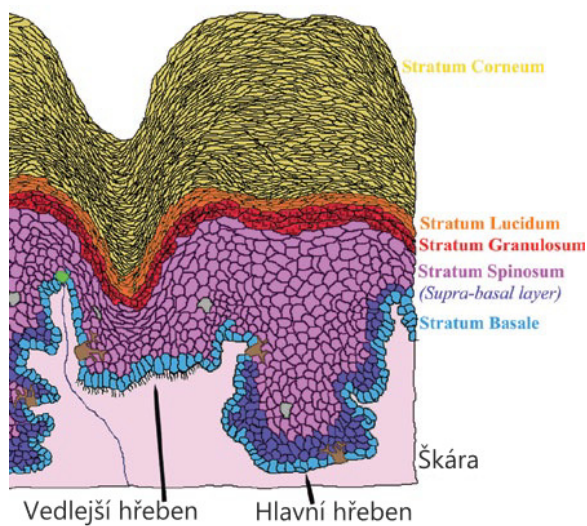
Vrásky nám také mohou změnit strukturu otisku prstu. A jsou uzpůsobené mechanickými změnami kůže v čase (stárnutí). Hlavně je tímto ovlivněná škůra (Dermis), která tímto mechanickým opotřebením trpí nejvíce (ztrácí elasticitu, vlákna kolagenu a elastinu



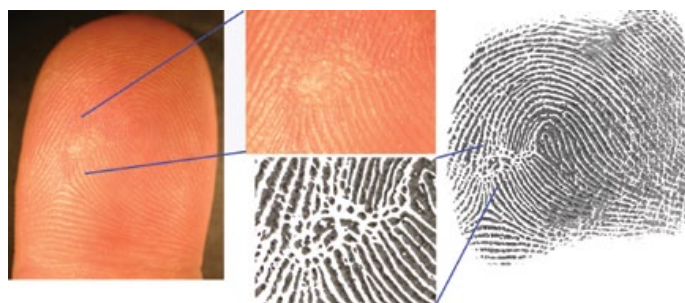
se smršťují, ...). Také je časem výškový rozdíl mezi papilárními liniemi a údolími čím dál tím menší, což vyústí v méně kvalitní otisk prstu. [17]



Obrázek 2.1: Struktura pokožky a škóry. [17]



Obrázek 2.2: Vrstvy pokožky. [17]



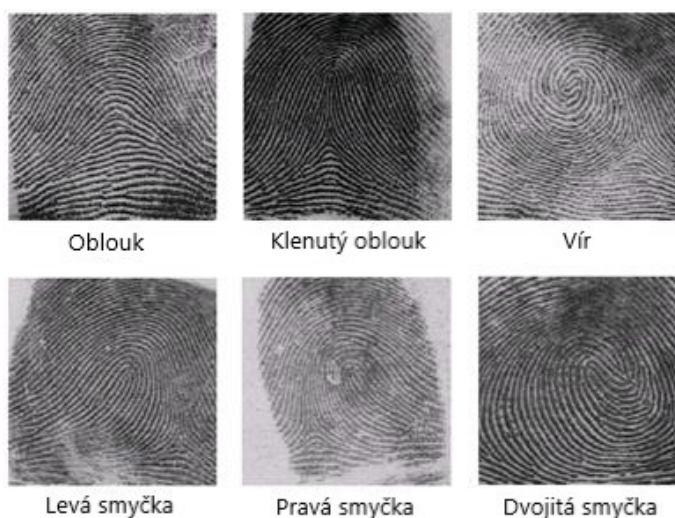
Obrázek 2.3: Otisk prstu s jizvou. [17]

## 2.3 Klasifikace otisků prstů

Při porovnávání otisků prstů je výhodné kategorizovat je do tříd. Na základě kterých jsme schopni rychle určit zda-li jsou porovnávané otisky stejné třídy či nikoli. Tento postup je obzvláště důležitý u velkých databází otisků prstů jako má FBI (**F**ederal **B**ureau of **I**nvestigation) a další podobné organizace. [9, 19, 17]

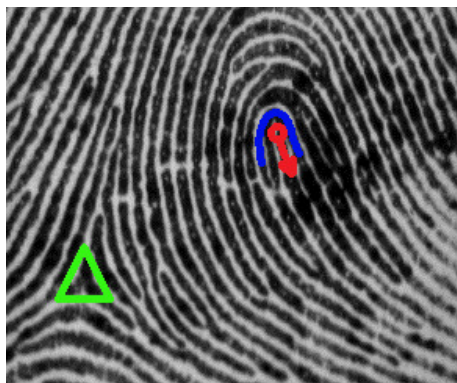
Konkrétně u FBI v Americe používají IAFIS (**I**ntegrated **A**utomated **F**ingerprint **I**dentification **S**ystem), což je automatizovaný porovnávací systém, který využívá Henryho klasifikačního systému. Ten otisky rozděluje do tří tříd: oblouk, smyčka, vír (přibližný výskyt v populaci: 6,6 %; 65,5 %; 27,9 %). [19, 9]

V dnešní době se už však používají systémy, které tyto třídy dělí na specifitější podtřídy (viz obrázek 2.4). [19, 17]



Obrázek 2.4: Třídy otisků prstů. [19, 9]

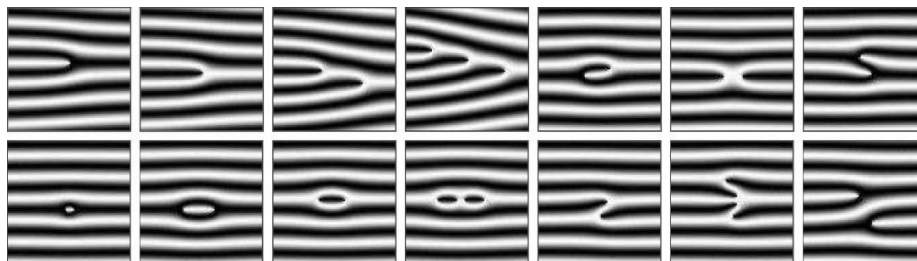
Před pokračováním je nutné definovat tyto pojmy: Delta – místo v otisku prstu, kde probíhají papilární linie do tří směrů. Jádru – střed otisku prstu, nachází se na nejspodnějším vyklenutí v průběhu papilárních linií. Na obrázku 2.5 jsou barevně vyznačeny. [9, 19]



Obrázek 2.5: Vyznačená delta (zelený trojúhelník), jádru (modrý oblouk) a směr jádru (červená šipka). [19]

## 2.4 Markanty otisků prstů

Samotné třídy nám však nestačí pro identifikaci osob. Právě proto nás zajímají útvary, které nesou název **markant**. Tyto útvary jsou tvořeny papilárními liniemi a údolími mezi nimi. V obrázku 2.6 máme vyznačeny základní typy markantů a ty jsou (pořadí zleva doprava, ze shora dolů): *ukončení*, *jednoduchá vidlička / rozdvojení*, *dvojitá vidlička*, *trojitá vidlička*, *hák*, *křížení*, *boční kontakt*, *bod*, *interval*, *jednoduchá smyčka*, *dvojitá smyčka*, *jednoduchý most*, *dvojitý most*, *průsečná linie*. [9, 19]



Obrázek 2.6: Základní typy markantů. [19, 9]

## 2.5 Z pohledu kriminalistiky

Kriminalistika je důležitým prvkem trestního soudního systému. Forenzní vědci zkoumají a analyzují důkazy z míst činu a na jiných místech, aby získali objektivní zjištění, která mohou pomoci při vyšetřování a stíhání pachatelů trestných činů nebo zbavit nevinného člověka podezření.[16]

Existují tři typy otisků prstů [9, 17]:

- Válený – většinou zachycen na daktyloskopickou kartu za pomoci inkoustu či barvy
- Živý – otisk zachycen na nějakém povrchu většinou neúmyslně
- Latentní – otisk, který je většinou obtížné zpozorovat pouhým okem a často bývá neúplný (tento druh otisku nás v této práci zajímá nejvíce)

### 2.5.1 Podtypy latentních otisků prstů

Patentní otisk prstu – Hodně variant těchto otisků prstů lze detekovat pouhým okem. Důležitý je pro tento typ otisku dobrý zdroj světla. Dobrý příklad takového otisku je mastný otisk prstu na okenici. Jsme schopni je nalézt také v krvi, barvě, inkoustu, blátě nebo prachu.[17]

Latentní otisk prstu – Tyto otisky jsou nedetekovatelné dokud neprojdou fyzickými či chemickými procesy na zvýraznění jejich rysů.[17]

Plastický otisk prstu – Vznikne když v momentě kontaktu prstu a povrchu je daný povrch dostatečně tvarovatelný, aby byl schopen zachytit tří-dimenzionální charakteristiku prstu. Tímto se nám vytvoří forma prstu ve které jsou znázorněny papilární linie a propadliny. Jsme schopni je nalézt v jílu, tmelu, měkkém vosku, změkklý či tavící se plast, těžký tuk, lepkavá barva a jiné.[17] (Pozn.: obecně se všem těmto podtypům říká latentní otisk prstu)

## 2.5.2 Faktory ovlivňující kvalitu otisku

Řadí se do tří kategorií [17]:

- Před obtiskem – otisk ovlivňuje stav/zdraví papilárních linií prstu (věk, pohlaví, stimulus, povolání, nemoc, ...) a množství/typ usazenin na prstu (materiál kterého se prst před obtiskem dotkl).
- Při obtisku – otisk ovlivňuje materiál na kterém bude zanechán otisk (textura, plocha/zakřivení/tvar/teplota povrchu, kondenzace, kontaminanty, jiné materiály na povrchu, ...). Dále závisí na tlaku vyvíjený na daný povrch.
- Po obtisku – otisk ovlivňuje okolní prostředí po zanechání otisku (kontakt s jiným povrchem, voda, vlhkost, teplota).

## 2.6 Technologie snímání otisků prstů

U technologií kde se otisky prstů získávají za pomoci snímačů nás zajímají parametry jako rozlišení (běžně kolem 500bpi), snímací plocha (0,7cm × 0,7cm pro přístupové systémy, 10cm × 6cm pro daktyloskopické systémy), počet bitů (pro zakódování barvy, běžně 8 bitů), geometrická přesnost (množství zkreslení otisku prstu vůči realitě) a kvalita obrazu. [9]

U modernějších technologií často vídáme snahu o maximalizování odhalení podvrhu (detekování živosti, detekování potních žláz, ...) [9, 19]

**Jednotlivé snímače se převážně dělí na:**

- Dotykový – prst je položen na plochu snímače, kde bez dalšího pohybu pár sekund setrvá. Výsledkem je otisk prstu[19]. Výhody/nevýhody:
  - jednoduchý na použití i pro nezaškolené uživatele
  - náchylný na špatnou orientaci prstu
  - různorodé velikosti a tvary prstu vedou k horším výsledkům
  - může po snímání zůstat latentní otisk prstu (riziko zneužití)
  - náchylný na zašpinění, což vede k horším výsledkům
- Tahový – prst provede plynulý vertikální pohyb po povrchu snímače tak, aby snímač byl v kontaktu s každou částí prstu, kde se nachází papilární linie, které utváří náš požadovaný otisk. Výsledkem je rekonstrukce otisku prstu, které bylo dosaženo díky malým částem otisku prstu získané ze snímače[19]. Výhody/nevýhody:
  - o něco širší než dotykový snímač, ale za to je výška jen pár milimetrů
  - levnější než alternativy
  - pouze malá část latentního otisku prstu může zůstat
  - díky tahovému pohybu prstu znatelně menší náchylnost na zašpinění
  - těžké na použití (nutnost správné a stálé rychlosti, pozice, stabilnosti)
  - rychlost rekonstrukce otisku prstu je malá a může být nepřesná

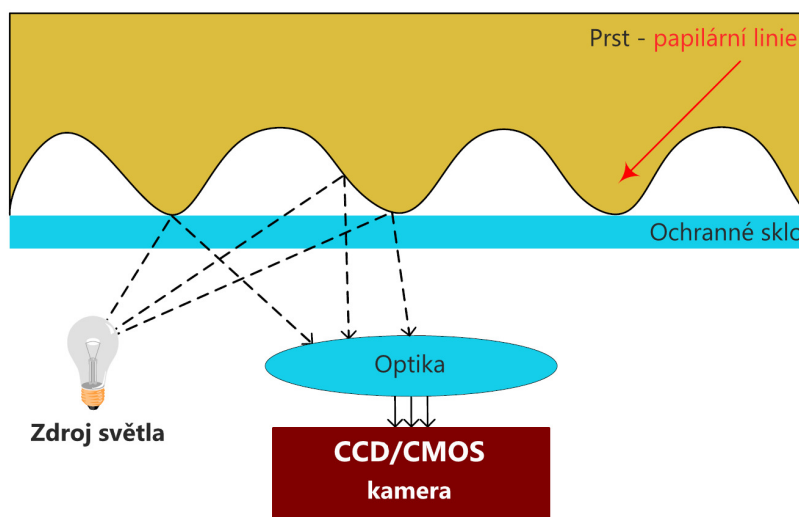
- Bezdotykový – pracují na podobném principu jako dotykové, ale bez nutnosti dotyku snímače. Výsledkem je otisk prstu[19]. Výhody/nevýhody:
  - žádné riziko latentních otisků prstů či zašpinění
  - zařízení většinou obklopuje celý prst, což znamená větší cenu a méně kompatibilních prstů

### 2.6.1 Daktyloskopické karty

Způsob při kterém dojde k otištění prstu na papírovou daktyloskopickou kartu, která se v minulosti uschovala do repositáře s dalšími kartami, které měly mezi sebou jistou podobnost (pro snazší vyhledávání). Bohužel čím více mělo policejní oddělení kartiček tím déle trvalo vyhodnocení požadavku. V dnešní době se daktyloskopické karty načítají do počítače pomocí skeneru. Jedná se o nejstarší metodu používanou při daktyloskopii. [9, 17]

### 2.6.2 Optická technologie

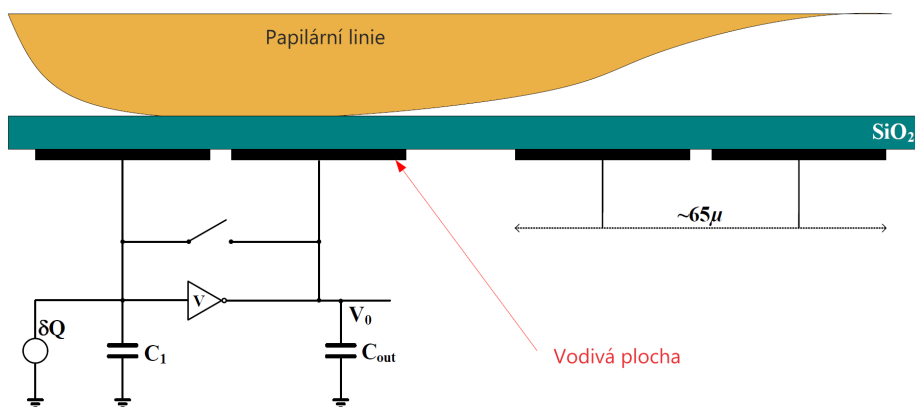
Jeden ze starších způsobů, který je založen na principu FTIR (**F**rustrated **T**otal **I**nternal **R**eflection). Kde prst je položen na ochranné sklo, tak aby se jej papilární dotýkaly a propadliny naopak nikoli. Paprsek ze zdroje světla (LED) se odrazí od papilárních linií, ale v propadlinách se velká míra energie fotonu absorbuje do pokožky (díky rozptýlení). Odražené paprsky projdou přes optiku k CCD (**C**harged-**C**oupled **D**evice) nebo CMOS (**C**omplementary **M**etal-**O**xide-**S**emiconductor) kameře. Když zaměníme ochranné sklo za průsvitnou trubičku schopnou rotace a patřičně přizpůsobíme ostatní komponenty, jsme schopni z dotykového snímače udělat snímač tahový. V jistých případech lze technologii přizpůsobit, aby se snímač stal bezdotykovým (velmi podobné fotografickým zařízením). Další typ optického snímače využívá technologie OCT (**O**ptical **C**oherence **T**omography). Toto řešení je velmi drahé, ale má schopnost získat obrázek z hlubší vrstvy pokožky. Tudíž je mnohem složitější tento snímač oklamat. [9, 19]



Obrázek 2.7: Princip optické technologie. [19]

### 2.6.3 Kapacitní technologie

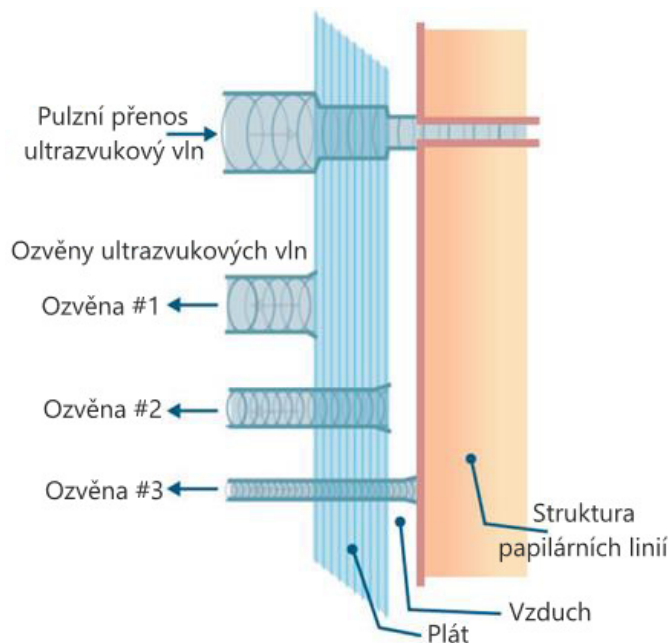
Snímač je složen z dvou dimenzionální matice malých vodivých plošek, na nichž je napařená ochranná vrstva nevodivého oxidu křemičitého ( $SiO_2$ ). Četnost vodivých plošek je vyšší než četnost papilárních linií. Přiložením prstu vzniknou nad ploškami těchto plošek kondenzátory (jednou elektrodou je papilární v místě styku s ploškou matice snímače a druhou je právě tato ploška), jejichž výstupem je hodnota odpovídající překryvu plochy plošky. Je-li výška snímače v řádech milimetrů jedná se o snímač tahový. Elektrostatické napětí může způsobit poškození snímače. Ochranná vrstva musí být co nejtenčí, aby pořad šlo přesně měřit rozdíl mezi papilární linií a propadlinou, ale zároveň musí být dostatečně silná, aby ochránila zbylé komponenty před nežádoucími látkami jako je pot (ten obsahuje látky, které by mohly snímač poškodit). [9, 19]



Obrázek 2.8: Princip kapacitní technologie. [19]

### 2.6.4 Ultrazvuková technologie

Je založena na rotujícím ultrazvukovém vysílači, v němž je zabudován i přijímač. Tento rotuje po kruhové dráze a snímá otisk prstu. Frekvence používaná těmito snímači je mezi 20 kHz až několik GHz. Obecně řečeno čím větší frekvence tím vyšší rozlišení obrazce. Ultrazvukové vlny proniknou i pod povrch kůže, což umožňuje odhalit pokusy o oklamání a ušpiněné prsty (či jiné neduhy) nehrají velkou roli. Tato technologie je jednou z nejpřesnějších technologií snímání otisků prstů, ale zároveň velmi drahá. [9, 19]



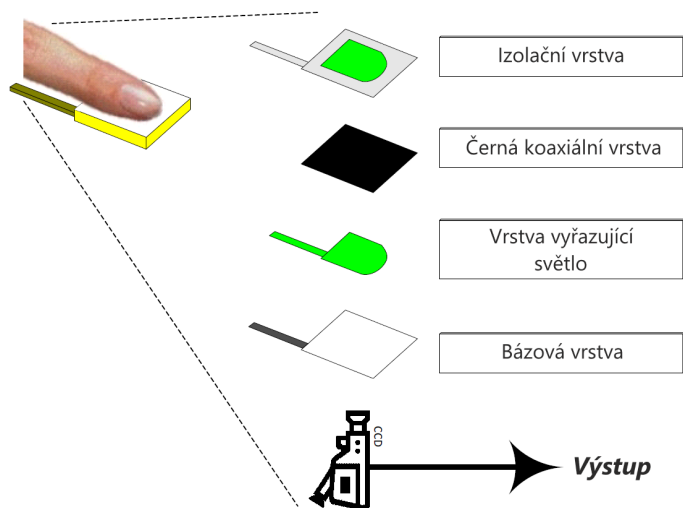
Obrázek 2.9: Pohyb senzoru ultrazvukové technologie. [19]

### 2.6.5 Technologie elektrického pole

Čidlo se skládá z hnacího prstence a matice antén. Hnací prstenec generuje sinusoidní rádiový signál a matice aktivních antén tento signál modulovaný kůží (přesněji škárou) přijímá. Obdobně jako u ultrazvukové technologie umožňuje tato technologie odhalit pokusy o oklamání a ušpiněné prsty (či jiné neduhy) nehrají velkou roli. [19]

### 2.6.6 Elektrooptická technologie

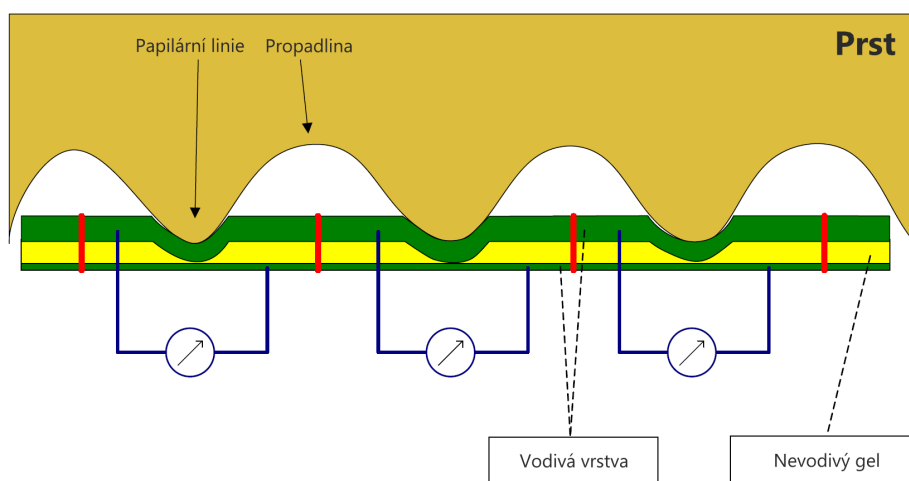
Se skládá ze 4 vrstev. Vrstva izolační, černá koaxiální vrstva, vrstva vyřazující světlo (fosfor), bázová vrstva. Pod těmito vrstvami je CCD/CMOS kamera. Když papilární vrstva je ve styku se snímačem zapříčiní to dotyk černé koaxiální vrstvy s fosforovou vrstvou vyřazující světlo. Toto světlo projde přes bázovou vrstvu kde jej zachytí kamera. Přítlakem prstu vybudí styk černé koaxiální vrstvy emitování světla ve fosforové vrstvě. Toto záření projde základní vrstvou do senzoru. [9, 19]



Obrázek 2.10: Princip elektrooptické technologie [19]

### 2.6.7 Tlaková technologie

Je složena ze tří vrstev, přičemž mezi elektrovedivé vrstvy je vložen nevodivý gel. Přiložením prstu na plochu snímače dojde ke stisku nevodivého gelu v místě dotyku papilárních linií tak, že se elektrovedivé vrstvy dotknou. U této technologie je důležité použít dostatečně citlivý materiál, aby bylo možné odlišit papilární linii od propadlin. [9, 19]

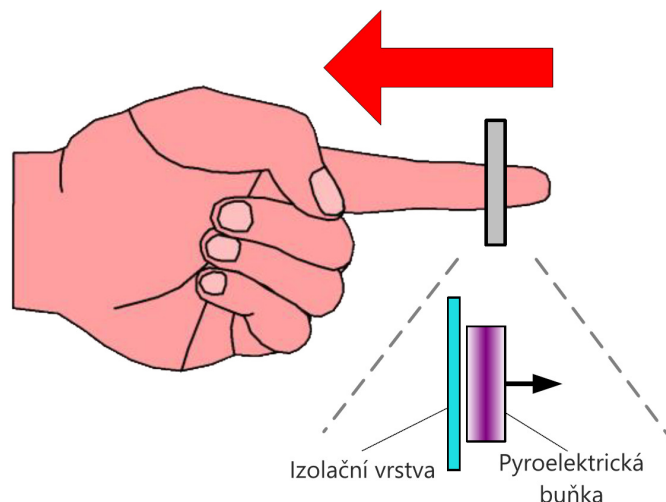


Obrázek 2.11: Princip tlakové technologie. [19]

### 2.6.8 Termická technologie

Princip je založen na tepelném záření. Papilární mají vyšší vyzařování tepla než propadliny mezi nimi a vzhledem k tomu, že teplota mezi nimi má tendenci se vyrovnávat je nutno využít snímačů tahových. Prst je protažen přes pyroelektrickou buňku, která snímá tepelné vyzařování. [9, 19]





Obrázek 2.12: Princip termické technologie. [19]

### 2.6.9 Technologie MEMS

MEMS (**M**icro **E**lectro-**M**echanical **S**ystems) používá malé části o velikosti mikrometrů k získání otisku prstu. Jedna z implementací této technologie používá miniaturní piezorezistivní ramena. Uživatel tahem sjede prst po snímači, který se skládá z tří řad piezorezistivních měřidel. Jejich rovnoběžné vychýlení změny v napětí, která je potom změřena a z nich následně vznikne výsledný otisk prstu. Výsledek je v binární podobě (papilární linie = 1, údolí = 0). Další variantou je užití miniaturních topných tělísek, které prstu zvýší teplotu a výsledný otisk prstu získá z teplotního rozdílu papilárních linií a propadlin. [19, 5]

## 2.7 Postup zpracování otisků prstů

Nejdříve je nutno získat digitální obrázek otisku prstu technologiemi popsanými výše (popřípadě jiné). A dále je tento obrázek vylepšit a to tak, že v každém jeho bodě a okolí se spočítá směr papilárních linií. Pokud bod leží přímo na nějaké papilární linii tak s velkou pravděpodobností jsme schopni určit jeho směr. V této fázi se dá použít spousta různých algoritmů někdy i vícero z nich (e.g. filtry Gabor, frekvenční, Butterworth, Ikonomopoulos, ...). [19]

Dalším krokem je binarizace (prahování), což nám převede obrázek do binární podoby (papilární linie = 1, údolí = 0).[19]

Dále je nutno pro extrakci markantů mít papilární linie o tloušťce 1 px, tudíž převedeme veškeré linie tlustší než 1 px právě na 1 px. Důležité je u tohoto kroku, aby vztah mezi jednotlivými papilárními liniemi zůstal stejný.[19]

Teď nám zbývá už jen extrahovat pozici markantů dále jejich typ, gradient. Na základě těchto informací už jsme schopni zjistit zda-li se otisky (přesněji jejich markanty) shodují. Na toto jsme schopni jít mnoha způsoby (e.g. lokální/globální porovnání markantů, MCC (Minutia Cylinder-Code), ...). Na obrázku 2.13 je ukázána výsledná podoba jednotlivých kroků (zleva doprava). [19, 2]



Obrázek 2.13: Zpracování otisků prstů. [19]

## 2.8 Syntetický otisk prstu

V dnešní době se technologie snímání otisků prstů vyvíjejí za pomoci technologií, které vyžadují hodně vstupních dat. Ať už na testování nějakého algoritmu nebo přímo na jeho tvorbu (machine learning). Sbírání takového velkého množství otisků prstů je drahé, zdlouhavé a nese riziko porušení zákonů (e.g. obecné nařízení o ochraně osobních údajů – GDPR). Čím menší množství otisků je použito k testování/tvorbě algoritmu, tím je jeho přesnost menší a vzrůstá nám riziko, že výsledky tohoto algoritmu budou závislé na datech (pro běžně používané otisky bude fungovat znatelně přesněji než pro ty méně běžné).[19]

Právě proto by bylo dobré mít program, který je schopen vygenerovat věrohodnou aproximaci otisku a udělat jich tolik kolik jich bude potřeba, navíc levněji a rychleji. [19]

### Generování syntetického otisku prstu

Je v podstatě opačný postup při zpracování reálných otisků prstů (inverzní biometrie). U obrázku 2.13 jdeme zprava doleva. [35]

Existuje spousta metod popsanych zde [27, 4, 30, 3, 1, 14, 25]. Obecně lze říci, že tyto metody jsou založeny na tom samém principu. Nejstarší a zároveň nejznámější metoda se zdá být SFinGe (**S**ynthetic **F**ingerprint **G**enerator - generátor syntetických otisků prstů)<sup>1</sup> (tuto metodu budu brát jako základ pro vysvětlování). [19, 18]

Následující kroky jsou znázorněny na obrázku 2.14.

V prvním kroku je třeba určit tvar otisku prstu. Základním tvarem je ovál, který následně můžeme poupravit do požadovaného stavu. [19]

V kroku druhém je vybrána třída otisku a také umístění jader a delt (singularit) na základě čehož bude vytvořeno pole orientací. Tento krok využívá Sherlock a Monroe model toku papilárních linií k vygenerování konzistentního pole orientací. [27, 19]

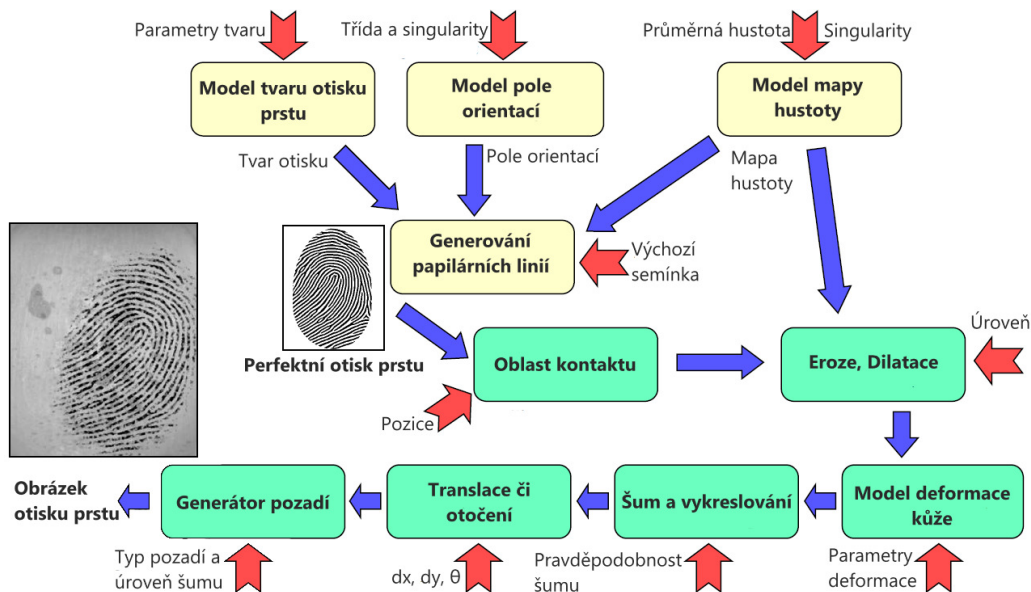
V dalším kroku figuruje tvorba mapy hustoty. Tento krok probíhá na základě poznatků získaných z reálných otisků prstů (e.g. pozice jader a delt). [19]

Ve čtvrtém kroku generujeme papilární linie. Tento krok zužitkuje všechny předešlé kroky a navíc i nějaké úvodní hodnoty (semínka). Iterativně je použit Gabor filtr, kde jeho orientace a frekvence jsou upraveny na základě pole orientací a mapy hustoty. Markanty a jejich typy jsou náhodně rozmístěny/určeny. Jako výstup generování získáme perfektní otisk prstu (na obrázku 2.13 ve fázi binarizace).[19]

<sup>1</sup>Dostupné z: <https://biolab.csr.unibo.it/research.asp?organize=Activities&select=%s&selobj=12&pathSubj=111%7C%7C12&>

V pátém kroku je třeba ještě zavést do našeho otisku nějaké poškození/deformaci a šum. Jak je již zmíněno zde 2.5 je mnoho faktorů, které nám ovlivňují kvalitu/podobu reálného otisku prstu a ty je třeba v tomto kroku se snažit zreplikovat, abychom se co nejvíce je jen možné přiblížili k realisticky vypadajícímu otisku.[19]

V posledním kroku nám stačí otisk náležitě orotovat a umístit na pozadí. V této práci bude právě řešeno generování tohoto pozadí.



Obrázek 2.14: Proces generování otisku prstu přes SFInGe. [3, 19]

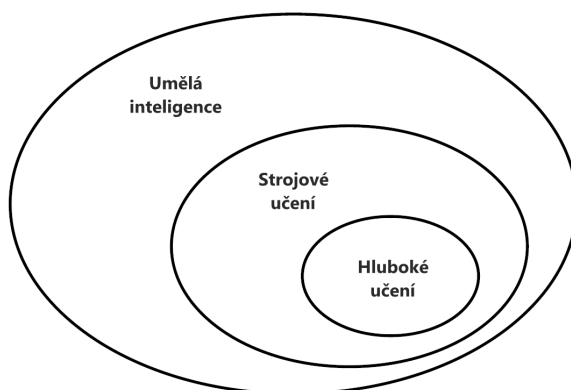
### 2.8.1 Poškození syntetického otisku prstu

Jednotlivá poškození syntetického otisku prstu, jako poškozený/úzký senzor, tlak/vlhkost, nežádoucí pohyb, zdravotní problémy ovlivňující kvalitu otisku (bradavice, jizvy, atopická dermatitida, psoriáza, ...) nám umožňují simulovat věrohodnější otisk prstu. Jeden ze způsobů je využití petriho sítí (SyFDaS - **S**ynthetic **F**ingerprint **D**amage **S**imulator and **G**enerator - simulátor/generátor poškození syntetických otisků prstů). [19, 20]

## Kapitola 3

# Umělá inteligence

Umělá inteligence nebo také anglická zkratka AI (**A**rificial **I**ntelligence) je široké odvětví, které v sobě zahrnuje jak strojové učení tak hluboké učení (viz obrázek 3.1), ale také jiné metody které nezahrnují učení (e.g. Symbolic AI). Obecně se dá toto odvětví definovat jako snaha o automatizaci intelektuálních úkolů běžně vykonávaných lidmi[6]. Je nutno podotknout že tato kapitola úzce souvisí s kapitolou 4.



Obrázek 3.1: Hluboké učení v kontextu strojového učení a umělé inteligenci. [6]

### 3.1 Strojové učení

Jedná se o jedno z nejpobulárnějších a nejúspěšnějších odvětví AI, které úzce souvisí se statistikou. Oproti statistice se ale strojové učení zabývá velkými a komplexními data-sety (e.g. databáze miliónů obrázků, každý v řádu desítky až stovky tisíc pixelů). Díky této velké náročnosti/složitosti při zpracování dat je třeba se více soustředit na praktičnost, empirické nálezy a méně na teorii (opačný přístup jako u teoretické fyziky, matematiky).[6]

Při klasickém programování nám programátor vytvoří množinu pravidel na základě kterých získáme ze vstupních dat požadovaný výsledek. Strojové učení je však nové programovací paradigma u kterého je snaha o získání pravidel ze vstupních dat a výsledků.[6]

K tomu aby strojové učení vůbec fungovalo ještě musíme vymyslet tu část které říkáme „učení“ e.i. způsob kterým zjistíme vzdálenost mezi vstupními daty a požadovaným výsledkem. Díky této vzdálenosti jsme schopni určit zda-li se v čase přibližujeme k cíli nebo naopak vzdalujeme a na základě toho dělat rozhodnutí, které nás dostanou co nejbližše k cíli.[6]

Když tento proces manipulace se vstupními/výstupními daty a učení zautomatizujeme a předáme počítači (stroji) tak získáme „Strojové učení“. Toto nám umožňuje řešit překvapivě široký prostor intelektuálních úkolů (e.g. autonomní řízení, rozeznávání hlasu).[6]

## 3.2 Hluboké učení

Hluboké učení je podmnožina strojového učení. Slovo „hluboké“ v tomto kontextu představuje počet reprezentativních vrstev modelu. V současné době hluboké učení zahrnuje desítky až stovky po sobě jdoucích reprezentačních vrstev. Tyto vrstvené reprezentace se učí za pomoci modelů kterým se říká **neuronové sítě** (viz sekce 3.5).[6]

Je příhodné si toto představit jako vícestupňový destilační proces, kde s každou vrstvou dostáváme hodnotnější (koncentrovanější) informace, které posléze využijeme ke splnění našeho cíle.

### Důležité pojmy:

- tensor - objekt, který slouží jako zobecnění vektoru či matice na vícero dimenzí.
- optimizátor - algoritmus, který na základě ztrátové funkce rozhoduje jak upravit váhy jednotlivých vrstev (e.g. SGD [**S**tochastic **G**radient **D**escend], RMSprop [**R**oot **M**ean **S**quared **p**ropagation], Adam, ...).

## 3.3 Strojové učení versus hluboké učení

Hluboké učení se od klasického strojového učení odlišuje typem dat, se kterými pracuje, a metodami, kterými se učí.[11, 6]

Algoritmy strojového učení využívají k vytváření předpovědí strukturovaná a označená data, e.i. že konkrétní funkce jsou definovány ze vstupních dat pro model a organizovány do tabulek. Znamená to pouze, že pokud ano, obvykle prochází určitým předzpracováním, aby byl uspořádan do strukturovaného formátu.[11, 6]

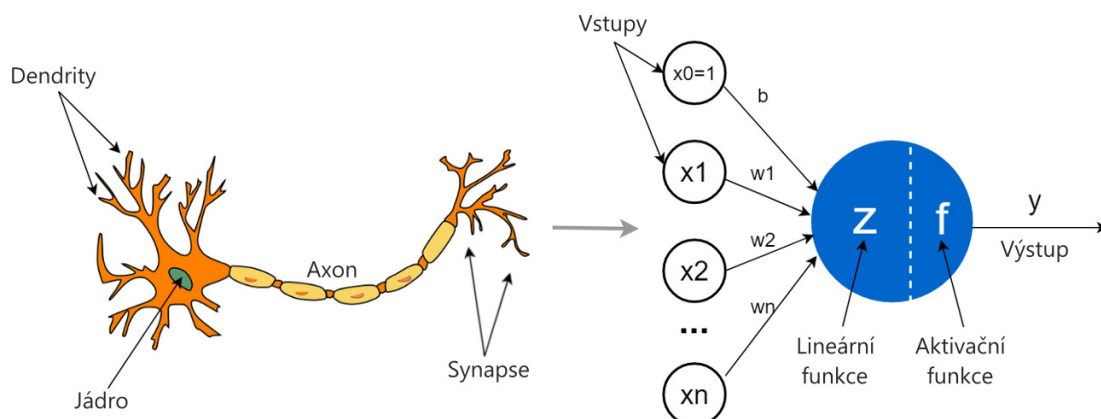
Hluboké učení eliminuje některé předzpracování dat, které je obvykle spojeno se strojovým učení. Tyto algoritmy mohou přijímat a zpracovávat nestrukturovaná data, jako je text a obrázky, a automatizuje extrakci prvků, čímž odstraňuje určitou závislost na lidských expertech. Představme si, že máme sadu fotografií různých domácích mazlíčků a chceme je roztrždit do kategorií „kočka“, „pes“, „křeček“, ... . Algoritmy hlubokého učení mohou určit, které vlastnosti (e.g. uši) jsou nejdůležitější pro odlišení každého zvířete od druhého. Ve strojovém učení je tato hierarchie funkcí vytvořena ručně lidským expertem.[11, 6]

Následně, za pomoci gradientního sestupu a algoritmu zpětného šíření chyby, se nám algoritmus hlubokého učení upraví a přizpůsobí tak aby došlo ke zvýšení přesnosti, což mu umožňuje provádět předpovědi o nové fotografii zvířete úspěšněji.[11, 6, 26]

Modely strojového učení a hlubokého učení jsou také schopné různých typů učení, které se obvykle dělí na učení pod dohledem, učení bez dozoru a učení na posílení. Řízené učení využívá označené datové sady ke kategorizaci nebo vytváření předpovědí. To vyžaduje určitý druh lidského zásahu pro správné označení vstupních dat. Naproti tomu učení bez dozoru nevyžaduje označené datové sady a místo toho zjišťuje vzory v datech a seskupuje je podle jakýchkoli rozlišujících charakteristik. Posílení učení je proces, ve kterém se model učí být přesnější pro provádění akce v prostředí na základě zpětné vazby s cílem maximalizovat odměnu.[11, 6]

### 3.4 Umělý neuron

Umělý neuron, někdy se mu také říká perceptron/jednotka/uzel, je matematický model, který je inspirován neurony lidského mozku. Tyto neurony jsou základní stavební jednotkou neuronových sítí. Na obrázku 3.2 lze vidět jak umělý neuron tak zjednodušenou reprezentaci toho pravého. [29]



Obrázek 3.2: Reprezentace reálného neuronu vlevo a umělého neuronu vpravo. [29]

#### 3.4.1 Umělý neuron se skládá z: [29, 12]

- Vstupy - jsou výstupy jiných neuronů nebo takzvaných „vstupních neuronů“, které slouží jako vstup dat do naší neuronové sítě.
- Váhy - jsou hodnoty, které určují hodnotu (důležitost) daného vstupu. Při inicializaci neuronové sítě je běžné nastavit tyto hodnoty náhodně. Tyto hodnoty se mění v čase na základě optimalizačního algoritmu a reprezentují stav sítě.
- Bias (zaujetí) - hodnota, která se může použít k zaručení nenulovosti parametru aktivační funkce.
- Lineární funkce - je součet všech součinů vstupů  $x_i$  a jejich vah  $w_i$ , s následným přičtením biasu  $b$ .

$$z = \sum_{i=1}^n x_i w_i + b \quad (3.1)$$

- Aktivační (nelineární) funkce - je použití aktivační funkce  $f$  na výsledek naší lineární funkce  $z$ . Primárně slouží k zanesení nelinearity do naší neuronové sítě.

$$y = f(z) \quad (3.2)$$

$$y = f\left(\sum_{i=1}^n x_i w_i + b\right) \quad (3.3)$$

- Výstup - výsledek aktivační funkce, který je předán jako vstup dalšímu neuronu nebo brán jako výsledek neuronové sítě

### 3.4.2 Konkrétní aktivační funkce (nejedná se o vyčerpávající seznam): [21]

- Binární kroková funkce

$$\text{binarystep}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad (3.4)$$

- ReLU (**R**ectified **L**inear **U**nit)

$$\text{relu}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (3.5)$$

- LeakyReLU (**L**eaky **R**ectified **L**inear **U**nit)

$$\text{leakyrelu}(x, \alpha) = \begin{cases} x\alpha & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (3.6)$$

- Sigmoid

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.7)$$

- TanH (**H**yperbolic **T**angent)

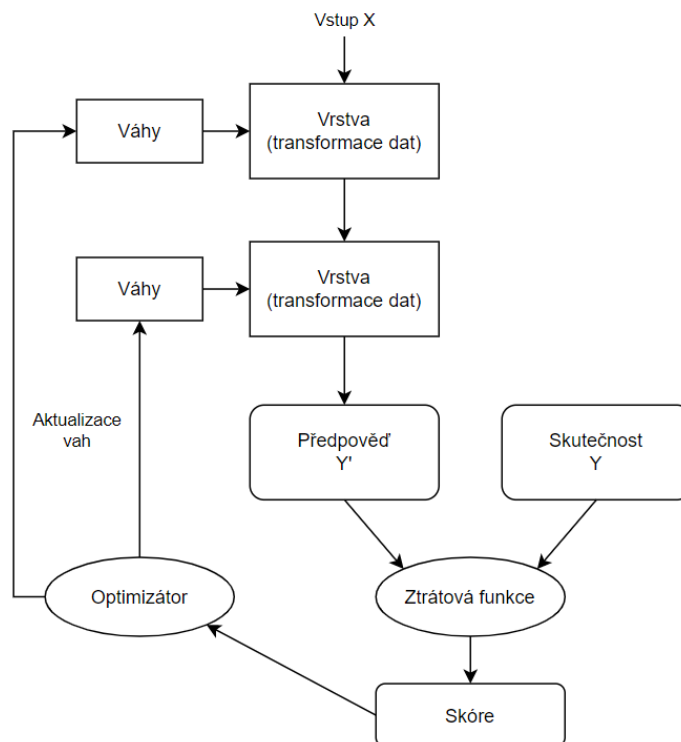
$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.8)$$

- Softmax

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^n \exp(z_j)} \quad (3.9)$$

## 3.5 Umělá neuronová síť

Umělé neuronové sítě se skládají z uzlů (umělých neuronů) seskupeny do jednotlivých vrstev. Tyto vrstvy se skládají ze vstupní vrstvy (zde vstupují data do naší neuronové sítě), alespoň jedné skryté vrstvy (zde se provádí užitečná práce) a výstupní vrstvy (zde již upravená data naši síť opouštějí). Počáteční design neuronových sítí byl inspirován skutečným mozkiem. [12, 6, 34]



Obrázek 3.3: Neuronová síť se ztrátovou funkcí a optimizátorem.[6]

### 3.5.1 Učení pod dohledem versus učení bez dozoru

Rozdíl mezi učením neuronových sítí je hlavně v tom, jaká úroveň zásahu do modelu expertem, je nutná. Při učení s dohledem se můžeme setkat s nutností označit data (vstupy, výstupy), upravování parametrů modelu či zásah do běžícího modelu. [15]

U učení bez dozoru se model učím sám od sebe, hledá skryté vzory z neoznačených dat, které využije ke svému cíli. Modely učení bez dozoru se používají pro tři hlavní úkoly: shlukování, asociace a snižování dimenzí. [15]

## 3.6 Dopředné neuronové sítě

Ve kterých je každý umělý neuron v jedné vrstvě připojen ke každému umělému neuronu z další vrstvy. Informace jsou předávány z jedné vrstvy do druhé pouze dopředným směrem. Neexistuje žádná zpětná vazba.[7]

## 3.7 Neuronové sítě autoenkodéru

Se používají k vytváření abstrakcí nazývaných kodéry, vytvořených z dané sady vstupů. Ačkoli se podobá tradičnějším neuronovým sítím, autokodéry se snaží modelovat vstupy samy, a proto je metoda považována za nekontrolovanou. Předpokladem autokodérů je desenzibilizovat vstupy irelevantní a senzibilizovat vstupy relevantní. Jak se přidávají vrstvy, další abstrakce jsou formulovány ve vyšších vrstvách (vrstvy nejbližší k bodu, ve kterém je zavedena vrstva dekodéru). [26]



## 3.8 Opakující se neuronové sítě

Používají sekvenční informace (e.g. data s časovým razítkem ze sensorového zařízení, mluvená věta složená z posloupnosti pojmů, ...). Na rozdíl od tradičních neuronových sítí nejsou všechny vstupy do opakující se neuronové sítě na sobě nezávislé a výstup pro každý prvek závisí na výpočtech jeho předchozích prvků. Tyto sítě mají své využití při předpovědi počasí, analýze sentimentu, tam kde se zužitkují časové řady. [13]

## 3.9 Konvoluční neuronová síť

Konvoluční neuronová síť (anglicky **C**onvolutio**n**al **N**eural **N**etwork - CNN) je známá svým skvělým výkonem co se týká zpracování obrázků, mluvení, zvuku. CNN se skládá z [10, 6]:

- Konvoluční vrstvy - je základním stavebním kamenem CNN a v ní probíhá většina výpočtů. Vyžaduje vstupní data, filtr a mapu rysů. Vstupem bude obrázek, který je tvořen tří-dimenzionální maticí (šířka  $\times$  výška  $\times$  barevná hloubka).[10, 6]

Důležitou komponentou je detektor rysů (jádro/filtr), který se bude pohybovat po podmnožinách obrázku (e.i. šířka a výška je limitována, ale barevná hloubka nikoli) a hledat rysy. Tomuto procesu se říká konvoluce. Detektor rysů je dvourozměrné pole vah reprezentující část obrázku. Filtry mohou mít různé velikosti (většinou 2D matice  $3 \times 3$ ) a ta se rovná velikosti pod-množiny obrázku. Filtr se poté aplikuje na oblast obrázku a mezi vstupními pixely a filtrem se vypočítá skalární součin jehož výsledek se vloží do výstupního pole. Poté se filtr posune o libovolný krok (v pixelech; velikosti posunu se také říká **stride**) a proces se opakuje, dokud filtr neprojde celým obrázkem. Finálnímu výsledku se říká mapa rysů. Po každé konvoluci se aplikuje aktivační funkce (zmněno v sekci 3.4).[10, 6]

Váhy v detektoru rysů mezi jednotlivými iteracemi zůstávají fixní čemuž se říká sdílení parametrů. Rozhodneme-li se použít několik filtrů současně, tak váhy mezi těmito různorodými filtry již fixní nejsou tudíž se mohou lišit. Použití vícero filtrů současně nám udává hloubku výsledné množiny map rysů. [10, 6]

Padding (vycpávka) se občas používá, když filtr velikostně neodpovídá vstupnímu obrázku (e.g.  $3 \times 3$  filtr s krokem 2px aplikován na  $6 \times 6$  obrázek). To znamená mírné zvýšení rozlišení obrázku kde nově rozšířené pixely jsou nastaveny na nulu. [10, 6]

Diagram hierarchie rysů v konvolučních neuronových sítích Konvoluční vrstva nakonec převádí obraz na číselné hodnoty, což umožňuje neuronové síti interpretovat a extrahovat relevantní vzory.[10, 6]

- Sdružovací vrstvy - slouží ke redukci dimenzí, snižování počtu vstupních parametrů. Obdobně jako u konvoluční vrstvy se pohybuje filtr po podmnožinách vstupu s tím rozdílem, že sdružovací vrstva nemá žádné váhy. Místo toho používá agregační funkci (e.g. maximální sdružování - pošle dál nejvyšší hodnotu, průměrné sdružování - pošle dál průměr získaný ze všech hodnot podmnožiny). Vhodné k redukci komplexity, snížení rizika přetrénování.[10, 6]
- Plně propojené vrstvy - jak již z názvu vypovídá takovéto vrstvy jsou mezi sebou plně propojeny, což znamená jsou neurony propojeny každý s každým. Tyto vrstvy mají tendenci fungovat dobře se softmax (3.9) aktivační funkcí. [10, 6]

## Kapitola 4

# Návrh generátoru pozadí

V současné době se vygenerované syntetické otisky prstů buď umístí na reálné pozadí (e.g. fotografie dokumentu, předmětu) nebo se na obrázek použije nějaká transformační metoda (e.g. Karhunen-Loève-ova transformace)[3, 19].

V této kapitole půjdu primárně cestou hlubokého učení, ale pokud-li nastane neúspěch a nebudu schopen generovat použitelné pozadí, tak půjdu konvenční cestou generování pozadí skrze sérii interpolací, translací a různých jiných operací, které by dosáhly úspěšného napodobení reálných pozadí latentních otisků prstů.

### 4.1 Data na trénování modelu

V této práci budou použity latentní otisky prstů z NIST Special Database 27<sup>1</sup>. Dále bude tato množina dat označována jako **S**.

Nejdříve je třeba roztrždit latentní otisky prstů do jednotlivých kategorií podle typu pozadí (e.g. Bankovka/šek, daktyloskopická karta, ...) a kvality (e.g. Dobrá, špatná, otřesná). Na základě těchto kategorií lze natrénovat generativní model.

Existují metody zvané augmentace dat [32], které nám umožňují naše vstupní data transformovat s cílem snížit pravděpodobnost přetrénování (navýšení množství dat). Mezi tyto metody se řadí: změna RGB intenzity [24], maskování regionů [8], otočení/zrcadlení/ořezávání, míchání dat [36], a další.

Jeden z potenciálních problémů, který mě může zastihnout je velmi omezený počet dat v jednotlivých kategoriích. Tudíž po vyzkoušení modelu mohu dojít k závěru, že kategorie sloučím do jedné a doufat, že toto řešení drasticky neredukuje kvalitu. Toto rozhodnutí by znamenalo, že bych šel cestou obyčejného DCGAN (**D**eep **C**onvolutional **G**enerative **A**dversarial **N**etwork) potenciálně s vylepšením [32] (diferencovatelná augmentace). Tato metoda nám umožňuje augmentaci dat a to bez nežádoucích následků jako zhoršená konvergence diskriminátoru **D** a generátoru **G** či zhoršení kvality výstupu výsledného **G**. Pro naše účely bude stačit jednoduché otáčení/zrcadlení/ořezávání a maskování regionů.

Pro dosažení lepších výsledků bude třeba patričný latentní otisk prstu odstranit. K tomuto lze dojít zprůměrováním oblasti otisku s jeho okolím, popřípadě ruční úpravou.

---

<sup>1</sup>Dostupné z: <https://www.nist.gov/itl/iad/image-group/nist-special-database-27>

## 4.2 Generování nových pozadí

Mám v plánu použít architekturu CGAN (Conditional Generative Adversarial Network). Ta se dělí na Generátor (dále jako  $G$ ) a na Diskriminátor (klasifikátor; dále jako  $D$ ) kde každý z nich je implementován jako CNN (Convolutional Neural Network) e.i. regularizované verze vícevrstevných perceptronů. Generátor se snaží vygenerovat realistické obrázky ke každé kategorii (zprva zcela náhodně) a diskriminátor se zase snaží zjistit jestli daný obrázek je reálný nebo falešný a pokud je reálný tak určit v jaké kategorii tento obrázek je. Tyto dvě neuronové sítě mezi sebou neustále soupeří a snaží se být lepší než oponent. Tento proces nakonec vyústí v graduální zlepšování jak generátoru (který my použijeme potom dále) tak i diskriminátoru. [26]

### 4.2.1 Abstraktní popis trénovacího algoritmu CGAN [26, 28]

1. Trénování diskriminátoru:

- Vezmi pseudo-náhodnou dávku reálných obrázků  $x$  z množiny  $S$  a jejich označení  $y$ . (  $p = (x, y)$  )
- Spočítej ztrátovou funkci pro  $D(p)$  a zpětně šíř (backpropagation) binární klasifikační ztrátu k aktualizaci vah a zkreslení  $\theta_D$  s cílem ji minimalizovat.
- Vezmi dávku pseudo-náhodných vektorů šumu  $z$  a označení  $y$  a s ní vygeneruj dávku falešných obrázků  $x'$ . (  $G(z, y) = x'|y$  )
- Spočítej ztrátovou funkci pro  $D(G(z, y), y)$  a zpětně šíř (backpropagation) binární klasifikační ztrátu k aktualizaci vah a zkreslení  $\theta_D$  s cílem ji minimalizovat.

2. Trénování generátoru:

- Vezmi dávku pseudo-náhodných vektorů šumu  $z$  a označení  $y$  a s ní vygeneruj dávku falešných obrázků  $x'$ . (  $G(z, y) = x'|y$  )
- Spočítej ztrátovou funkci pro  $D(G(z, y), y)$  a zpětně šíř (backpropagation) binární klasifikační ztrátu k aktualizaci vah a zkreslení  $\theta_G$  s cílem ji maximalizovat.

3. Opakuj krok 1 dokud nedosáhneme Nashovy rovnováhy (Nash equilibrium) e.i. opakuj dokud pravděpodobnost, že diskriminátor  $D$  uhádne reálný obrázek od falešného je 50 % (nemůže je od sebe rozeznat; v praxi bohužel tohoto lze těžko dosáhnout kvůli příliš velké komplexitě/variabilitě dat, proto se s iteracemi přestává jakmile se od ideálního stavu 50 %/50 % začneme vzdalovat).

Výše zmíněným algoritmem bychom již mohli naše pozadí vygenerovat, ale problém nastává v tom, že bychom museli mít hodně vstupních dat. Nemáme-li dostatečně velkou množinu dat  $S$  (což je náš případ) šance na úspěch je malá.

Právě proto na základě úspěchu předchozí metody se může stát, že budu nucen data z různých kategorií sloučit do jedné a použít DCGAN architekturu. Cílem této architektury je víceméně to samé jako u CGAN s tím rozdílem, že vstupní data nejsou rozřazena do jednotlivých kategorií.

### 4.2.2 Abstraktní popis trénovacího algoritmu DCGAN [26, 31]

1. Trénování diskriminátoru:

- Vezmi pseudo-náhodnou dávku reálných obrázků  $\mathbf{x}$  z množiny  $\mathcal{S}$ .
- Vezmi dávku pseudo-náhodných vektorů šumu  $\mathbf{z}$  a s ní vygeneruj dávku falešných obrázků  $\mathbf{x}'$ . ( $G(\mathbf{z}) = \mathbf{x}'$ )
- Spočítej ztrátovou funkci pro  $D(\mathbf{x})$  a  $D(\mathbf{x}')$  a zpětně šíř (backpropagation) binární klasifikační ztrátu k aktualizaci vah a zkresení  $\theta_D$  s cílem ji minimalizovat.

2. Trénování generátoru:

- Vezmi dávku pseudo-náhodných vektorů šumu  $\mathbf{z}$  a s ní vygeneruj dávku falešných obrázků  $\mathbf{x}'$ . ( $G(\mathbf{z}) = \mathbf{x}'$ )
- Spočítej ztrátovou funkci pro  $D(G(\mathbf{z}), \mathbf{y})$  a zpětně šíř (backpropagation) binární klasifikační ztrátu k aktualizaci vah a zkresení  $\theta_G$  s cílem ji maximalizovat.

3. Opakuj krok 1 dokud nedosáhneme Nashovy rovnováhy (Nash equilibrium) e.i. opakuj dokud pravděpodobnost, že diskriminátor  $D$  uhádne reálný obrázek od falešného je 50 % (nemůže je od sebe rozeznat; v praxi bohužel tohoto lze těžko dosáhnout kvůli příliš velké komplexitě/variabilitě dat, proto se s iteracemi přestává jakmile se od ideálního stavu 50 %/50 % začneme vzdalovat).

### 4.2.3 Ztrátové funkce

Založeny na NS-GAN (Non-Saturating GAN). Za pomoci těchto funkcí má možnost generátor  $G$  či diskriminátor  $D$  vědět jestli se zlepšuje/zhoršuje a na základě toho aktualizovat své parametry  $\theta$ .

$$J_G = E_{z \sim p_g} \log[D(G(z))] \quad (4.1)$$

$$J_D = E_{x \sim p_r} \log[D(x)] + E_{z \sim p_g} \log[1 - D(G(z))] \quad (4.2)$$

Ty  $E$ -čka znamenají očekávání nad  $x$  (skutečné rozložení dat) nebo  $z$  (šum, který získáváme z latentního prostoru),  $D$  značí funkci diskriminátoru (mapování obrázku na pravděpodobnost) a  $G$  značí funkci generátoru (mapování latentního vektoru na obrázek). [26]

## Kapitola 5

# Implementace generátoru a jeho výsledky

Tato kapitola bude pojednávat jak o implementaci generátoru pozadí za pomoci umělých neuronových sítí, ale také za pomoci konvenčních metod. Oba tyto způsoby byly implementovány v Pythonu a to primárně kvůli jednoduchosti použití a spoustu užitečných knihoven či frameworků, obzvláště co se týče hlubokého učení. Výsledky prošly vizuální kontrolou. V předešlých sekcích je vysvětleno vše potřebné co bylo použito k implementaci modelů.

### 5.0.1 TensorFlow

Zdarma open-source knihovna vyvíjena týmem Google Brain určená ke strojovému učení a jiným aplikacím umělé inteligence. Jak již z názvu vyplývá tak pracuje s tensory, což jsou abstraktní multidimenzionální objekty pro uchovávání dat se kterými bude náš model pracovat. Tato knihovna implementuje spoustu algoritmů a funkcionalit, které slouží k usnadnění vývoje modelů strojového učení. Je velmi populární díky své flexibilitě, dobrému výkonu a vzhledem k tomu, že ji má pod sebou společnost Google, tak i rozsáhlou a aktivní komunitu. Velmi populární aplikační rozhraní Keras využívá exkluzivně TensorFlow jako své jádro/backend (dříve i možnost využití jádra Theano).

### 5.0.2 Keras

Slouží jako aplikační rozhraní pro TensorFlow, které poskytuje spoustu vysokoúrovňových funkcionalit. Jedna z nich je možnost reprezentovat celý model, který se snažíme vytvořit, jako ucelený objekt, který má celou řadu užitečných metod, vlastností. Dále jsme schopni takový to model reprezentovat jako sekvenci vrstev (včetně vrstvy vstupní, vrstev skrytých a vrstvy výstupní) a to rychle a snadno. Tato skutečnost mi umožnila otestovat vícero různých možností při implementování modelu a to v relativně malém časovém rozmezí. Kromě řazení vrstev za sebou a tím vytvoření sekvence modelu Keras také umožňuje vytvoření komplexnějších struktur (e.g. sdílené vrstvy, vícero vstupů/výstupů, ...).

### 5.0.3 Anaconda

Anaconda je distribuce Pythonu zaměřená na vědecké výpočty, což mimo jiné obsahuje i hluboké učení. Díky ní je jednoduché a pohodlné zaručit bezproblémovou instalaci všech potřebných komponent, které byly použity při implementaci (komponenty jako Python, TensorFlow, Jupyter notebook, ...).

### 5.0.4 Jupyter notebook, Colaboratory

Jupyter notebook (Jupyter zápisník) je interaktivní webové prostředí sloužící k pohodlné dokumentaci a testování Python kódu. Pro nás je hlavně důležité prostředí Google Colaboratory (Colab), které kombinuje výpočetní sílu cloudu a právě Jupyter notebooku. Colab poskytuje širokou škálu možností, ale primárně je určen ke strojovému učení, kde Jupyter notebook běží na vzdáleném virtuálním počítači. Samotná neplacená verze umožňuje bezpoplatně (ikdyž s jistými kapacitními a časovými omezeními) přístup ke GPU (**G**raphical **P**rocessing **U**nit) či dokonce přístup k TPU (**T**ensor **P**rocessing **U**nit), což je výpočetní jednotka určená právě na operace nad tensory (tudíž ideální na strojové učení). Díky této skutečnosti jsem schopen provozovat hluboké učení v rozumném čase a to i přestože je můj lokální hardware naprosto nedostačující.

### 5.0.5 Spyder

Zdarma open-source IDE (**I**ntegrated **D**evelopment **E**nvironment - Integrované vývojové prostředí), které bylo navrženo s ohledem na programování v Pythonu primárně pro vědecké účely. Má k dispozici intuitivní grafické rozhraní, ve kterém je mimo jiné integrována interaktivní Python konzole a prohlížeč grafů, což jej činí ideální IDE pro tuto práci.

## 5.1 Generování za pomoci umělé neuronové sítě

Konkrétněji implementace natrhnutého modelu v předešlé kapitole. Neboli příprava vstupních dat modelu, implementace vhodných typů vrstev (a jejich adekvátního počtu) a to jak generátoru tak diskriminátoru, zvolení vhodných hyperparametrů a způsobu reprezentace výstup z již natrénovaného modelu.

### 5.1.1 Příprava dat

Data u kterých mám k dispozici použitelné pozadí u latentních otisků prstů jsou bohužel značně limitovaná množstvím. Přesněji mám k dispozici pouze ty snímky, které byly obsaženy v databázi NIST SD27. To činí celkově 290 snímků s unikátním pozadím u latentních otisků prstů.

Díky tomuto jsem se rozhodl snímky s relativně velkým rozlišením rozčtvrtit a vzhledem k tomu, že budu potřebovat k trénování snímky se stejným rozlišením tak i na ně aplikovat interpolaci.

K tomu, abych zvolil správné rozlišení jsem si zjistil statistiky o NIST SD27 za pomoci krátké skriptu co jsem napsal. Statistiky (delta udává rozdíl mezi šířkou a výškou):

- minimum:
  - šířka - 828px
  - výška - 874px
  - delta - 4px
- maximum:
  - šířka - 1652px
  - výška - 1565px

– delta - 601px

- medián:
  - šířka - 1440px
  - výška - 1409px
  - delta - 117px

Na základě výše zmíněných statistik jsem zvolil cílové rozlišení interpolace -  $1024 \times 1024 \times 1$ . Rozhodoval jsem se mezi bikubickou a Lanczos interpolací. Nakonec jsem zvolil Lanczos, kvůli obecně lepším výsledkům obzvláště co se týká snižování rozlišení (což je majorita u naší databáze).

Po rozčtvrcení jsem se dostal z 290 snímků na 1160 snímků s rozlišením  $512 \times 512 \times 1$ . Tyto snímky jsem ručně (s pomocí ručně napsaného AutoHotkey skriptu) rozřídil do tří tříd:

- **dokument** (soubor označen písmenem q) - do této třídy se řadí, kterékoliv pozadí ve kterém je dominantní text (tisknutý, psaný) nebo obrazec (bankovka, logo dokumentu).
- **čistší** (soubor označen písmenem w) - do této třídy se řadí pozadí, která jsou primárně světlejšího rázu a neobsahují přílišně mnoho poškození (škrábance, otisky, ...)
- **špinavší** (soubor označen písmenem d) - zde patří pozadí, která jsou těžko přiřaditelná do jiných tříd, převážně tmavšího rázu se spoustou poškození.

Důvod proč existují třídy **čistší** a **špinavší** (oproti něčemu konkrétnějšímu) je protože na obrázcích z NIST SD27 mnohdy nejde rozeznat o jaký typ pozadí se jedná a řadit už takto malou databázi do vícero specifitějších tříd by znamenalo téměř jistý neúspěch.

Následující krok není doporučen je-li k dispozici dostatečně velká databáze obrázků k trénování. Každý z těchto obrázků byl zrcadlen po horizontální a vertikální ose. Takže se dostaneme na 3480 obrázků, kde každý z nich třikrát otrotujeme o  $90^\circ$ , což ve výsledku činí 13920 obrázků s rozlišením  $512 \times 512 \times 1$  z původních 290 obrázků s různorodým rozlišením.

### 5.1.2 Hyperparametry

Je nutno si prvně krátce definovat pár pojmů:

- epocha - jeden průchod všech obrázků modelem (v našem případě 1160)
- dávka - skupina obrázků použita na průchod modelem

Výsledný model byl trénován za pomoci Google Colab (cloudový výpočet), protože trénování na lokálním CPU (**C**entral **P**rocessing **U**nit) je příliš pomalé a přístup k lokálnímu GPU, na kterém jsem schopen trénovat nemám. Trénovalo se po dobu 250 epoch s dávkami o velikosti 64 obrázků. Váhy byly inicializovány za pomoci Normálního rozdělení vycentrovaného na nulu se směrodatnou odchylkou 0,02 [31]. U aktivační funkce LeakyReLU byla nastavena  $\alpha$  (alpha; e.i. sklon „odtékání“) na 0,2. Padding (vycpávka) je nastavena na doplnění nul tam kde by jinak chyběly hodnoty. Filtr bude o velikosti  $3 \times 3$  pixelů a bude se pohybovat po x-ose a y-ose po dvou pixelech. Počet jader pro vrstvu MinibatchDiscrimination byl zvolen 200 a její dimenzionalita na  $90^1$ . Adam optimizátor byl zvolen [23] s

<sup>1</sup>Pro podrobnější informace: <https://github.com/King-Of-Knights/Keras-ACGAN-CIFAR10/blob/c5a89becb4e5a56dbf34bf7c45516bbf6c451d36/Minibatch.py>

rychlostí učení 0,0002 a  $\lambda_1$  (exponenciální míra rozpadu pro odhady prvního okamžiku) na 0,5 dle tohoto doporučení [31].

### 5.1.3 Implementace modelu

Model byl primárně testován za pomoci Google Colaboration. Implementace proběhla v Pythonu verze 3.7.13 . Primárně za pomoci API Keras verze 2.8.0, které využívá TensorFlow jako svůj back-end, taktéž o verzi 2.8.0 . Další moduly co stojí za zmínku jsou NumPy (1.21.6) a OpenCV (4.1.2).

Finální implementace modelu byla inspirována z mnoha zdrojů. Mezi ty nejdůležitější patří [26, 6], ale taktéž [22] (kód na vrstvu `MinibatchDiscrimination`).

Nejprve je potřeba vytvořit sekvenční model diskriminátoru podle 5.1 a zároveň uzpůsobit označení (které reprezentuje jednu z těchto tříd 5.1) tak, aby šlo spojit s obrázkem kterému toto označení náleží. Obdobně provedeme pro sekvenční model generátoru podle 5.2, kde jej označení taktéž sloučíme. Z těchto dvou sítí se vytvoří samotný CGAN model, kde se diskriminátoru nastaví argument `trainable` na `False`, což nám způsobí, že při trénování generátoru budou váhy diskriminátoru konstantní.

V následujícím úseku lze vidět pseudokód tvorby kombinovaného modelu CGAN. Zde můžeme vidět vytvoření diskriminátoru dle 5.1, nastavení konstantnosti vah diskriminátoru a následné kompilace diskriminátoru (příprava na trénování). Dále je vytvořen generátor dle 5.2, který bude zkombinován s již připraveným diskriminátorem. Finální model je již připraven na trénování.

```
1 def cgan_model():
2     diskriminator = vytvorit_diskriminator()
3     diskriminator.trainable = False
4     diskriminator.kompilovat()
5
6     generator = vytvorit_generator()
7
8     cgan = kombinovat_modely(diskriminator, generator)
9
10    return cgan
```



Tabulka 5.1: Sekvenční struktura diskriminátoru

Typ vrstvy	Parametry	Vstup	Výstup
GaussianNoise	odchylka=0.02		
Conv2D	filtry=32; velikost_jádra=3×3; posun=2×2	512×512×2	256×256×32
LeakyReLU	alpha=0.2		
Conv2D	filtry=64; velikost_jádra=3×3; posun=2×2	256×256×32	128×128×64
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=128; velikost_jádra=3×3; posun=2×2	128×128×64	64×64×128
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=256; velikost_jádra=3×3; posun=2×2	64×64×128	32×32×256
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=512; velikost_jádra=3×3; posun=2×2	32×32×256	16×16×512
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=1024; velikost_jádra=3×3; posun=2×2	16×16×512	8×8×1024
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=2048; velikost_jádra=3×3; posun=2×2	8×8×1024	4×4×2048
BatchNormalization			
LeakyReLU	alpha=0.2		
Conv2D	filtry=4096; velikost_jádra=3×3; posun=2×2	4×4×2048	2×2×4096
BatchNormalization			
LeakyReLU	alpha=0.2		
Flatten		2×2×4096	16384
MinibatchDiscrimination	počet_jader=200; dimenzionalita=90	16384	16584
Dense	dimenze_výstupu=1; aktivační_funkce=Sigmoid (viz 3.7)	16584	1

Dále můžeme vidět pseudokód pro trénování našeho modelu. Tato procedura začíná cyklem opakující se na základě proměnné `pocet_iteraci`, kterou lze odvodit následovně -  $pocet\_iteraci = \frac{pocet\_obrazku}{velikost\_davky} * pocet\_epoch$ . Následuje získání dávky (jejíž velikost udává proměnná `velikost_davky`) náhodných obrázků z naší databáze reálných obrázků (proměnná `data`) společně s jejich značkami. Posléze dojde k vygenerování dávky falešných obrázků za pomoci našeho generátoru a trénování diskriminátoru jak na reálných párech obrázků a značek tak těch falešných.

Nyní je třeba vygenerovat náhodnou dávku vektorů šumu a k nim jejich náhodné značky (s každou iterací budou méně náhodné). Teď jen trénování generátoru na právě vygenerovaném šumu a jejich značek a `iterace` je dokončena.

```
1 def trenovat(data, velikost_davky, pocet_iteraci):
2     for iterace in pocet_iteraci:
3         # diskriminator
4         ziskat nahodnou davku realnych obrazku
5         ziskat davku realnych znacek patrici k obrazkum z radku 4
6         vygenerovat davku falesnych obrazku na zaklade znacek z radku 5
7
8         trenovat diskriminator na zaklade obrazku z radku 4, 6
9
10        # generator
11        vygenerovat nahodnou davku vektoru sumu
12        vygenerovat nahodne znacky
13        trenovat generator
```

Tabulka 5.2: Sekvenční struktura generátoru

Typ vrstvy	Parametry	Vstup	Výstup
Dense	dimenze_výstupu=16384; aktivační_funkce=Sigmoid (viz 3.7)	16384	$2 \times 2 \times 4096$
Reshape	cílový tvar= $2 \times 2 \times 4096$		
Conv2DTranspose	filtry=2048; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$2 \times 2 \times 4096$	$4 \times 4 \times 2048$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=1024; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$4 \times 4 \times 2048$	$8 \times 8 \times 1024$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=512; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$8 \times 8 \times 1024$	$16 \times 16 \times 512$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=256; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$16 \times 16 \times 512$	$32 \times 32 \times 256$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=128; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$32 \times 32 \times 256$	$64 \times 64 \times 128$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=64; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$64 \times 64 \times 128$	$128 \times 128 \times 64$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=32; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$	$128 \times 128 \times 64$	$256 \times 256 \times 32$
BatchNormalization			
ReLU			
Conv2DTranspose	filtry=1; velikost_jádra= $3 \times 3$ ; posun= $2 \times 2$ ; aktivační_funkce=TanH (viz 3.8)	$256 \times 256 \times 32$	$512 \times 512 \times 1$

### 5.1.4 Vyhodnocení modelu

Za normálních okolností bych nyní provedl vizuální zhodnocení vygenerovaných výsledků a použil metodu Fréchetovy počáteční vzdálenosti (Fréchet inception distance), která slouží nejenom k vyhodnocení distribuce vygenerovaných obrázků (původní metrika Inception Score), ale taktéž k porovnání této distribuce s distribucí obrázků použitých při trénování modelu.

Bohužel jsou tyto vyhodnocení bezpředmětné, jelikož implementovaný model generuje něco co by se dalo kategorizovat pouze jako šum. Potencionální příčiny:

- Přetrénování - generované obrázky jsou přílišně podobné či dokonce identické datům použitých při trénování (není náš případ).
- Nedostatek unikátních dat - málo unikátních dat je častou příčinou nedostatečné kvality generovaných obrázků výsledného modelu. Čím menší množství dat tím větší riziko přetrénování či nedosažení konvergence. Větší množství dat by i umožňovalo navrnutí modelu s menšími hardwareovými požadavky.
- Špatné nastavení modelu - špatné zvolení vrstev, aktivačních funkcí, optimizátorů, hyperparametrů, . . .
- Problém mizejícího gradientu (Vanishing Gradients) - je-li diskriminátor až příliš úspěšný v rozeznávání reálných od falešných obrázků může nastat problém takový, že generátoru není poskytnuto dostatek informací k tomu, aby se zlepšil a tudíž může dojít k jeho stagnaci (již se nic nového nenaučí). Což může být obzvláště problematické když k tomuto dojde na počátku učení.
- Kolaps režimu (Mode Collapse) - generátor generuje velmi malé množství obrázků, protože se diskriminátor zasekl v lokálním minimu a generátor proto generuje pouze v této oblasti. Tudíž je model neschopen nalézt globální minimum (optimální stav).

V mém případě se s velkou pravděpodobností jedná o „mode collapse“ (jedná se o jeden z nejčastějších problémů těchto sítí) s kombinací nedostatku dat, protože v prvotních fázích trénování jsem neměl implementovanou vrstvu `MinibatchDiscrimination` a ztráta diskriminátoru či generátoru šla k nule znatelně rychleji než když byla vrstva implementována.

## 5.2 Generování za pomoci konvenčních metod

Vzhledem k tomu, že se předchozí způsob nepodařilo úspěšně implementovat jsem se rozhodl alespoň splnit zadání generováním pozadí latentních otisků prstů mimo strojové učení.

Tohoto cíle dosáhnou implementováním programu, který mi ze předpřipravených vstupních dat sérií operací (e.g. translace, interpolace, maskování, . . .) učiní dostatečně důvěryhodné pozadí na které se posléze umístí syntetický otisk prstu, kvůli porovnání s reálným obrázkem (třeba z NIST SD27).

### 5.2.1 Příprava vrstev (dat)

Nejdříve bylo potřeba vytvořit databázi obrázků s vhodným rozlišením a tématikou, které budou sloužit jako základ ke všem dalším operacím. Tyto obrázky byly nalezeny za pomoci Google webového vyhledávače, které byly roztříděny do jednotlivých tříd (kategorií) ze kterých si uživatel následně může zvolit jaký typ pozadí bude použit. Dále byly z těchto obrázků ručně odstraněny nežádoucí prvky (okraje, popisky, označení, . . .).

Jednotlivé třídy:

- Bankovka - 30 unikátních prvků; medián rozlišení - 1543,5×774
- Šek - 10 unikátních prvků; medián rozlišení - 1409,5×663
- Ručně psaný dopis - 35 unikátních prvků; medián rozlišení - 1450×2048
- Právní dokument - 10 unikátních prvků; medián 1275×1650

Jako další bylo třeba najít různé vrstvy, které budou posléze aplikovány na vrstvu základovou. Byly testovány vrstvy z různých zdrojů (i vlastní výroby), ale nakonec se podařilo najít takové, které jsou dostatečně kvalitní a unikátní<sup>2</sup>. Taktéž bylo třeba ruční úpravy převážně co se týká skokových přechodů mezi odlišnými částmi vrstvy.

Potom bylo třeba připravit samotné syntetické otisky prstů, které budou nakonec aplikovány na nově vygenerované pozadí, což nám umožní korektně vyhodnotit (při porovnání reálných a syntetických pozadí) kvalitu našeho generátoru. Tyto otisky lze vygenerovat za pomoci programů, určených k jejich syntetizaci, jako SFinGe či Anguli (založeno na SFinGe). Pro naše účely bohatě stačí si tyto otisky předgenerovat či přímo stáhnout již předgenerované ze stránek výše zmíněných generátorů<sup>3</sup>.

## 5.2.2 Implementace generátoru

Generátor byl implementován v Pythonu 3.9.7, který je obsažen v distribuci Anaconda (již zmíněno v sekci 5) a dalších modulů zmíněných v souboru *requirements.txt*.

Nejprve je třeba se ujistit, že následující složky *output*, *output\_cutout* a *output\_with\_fingerprint* jsou prázdné, aby se minulý výstup z generátoru nemísil se současným. Potom je třeba náhodně načíst určitý počet obrázků (počet specifikován uživatelem) z jedné z tříd specifikovaných výše (konkrétní třída taktéž zvolena uživatelem). Tyto obrázky se při načtení převedou do černobílé podoby. Cílové rozlišení generovaného pozadí je taktéž zvoleno uživatelem.

Pokud jedna z dimenzí obrázku neodpovídá cílovému rozlišení, budou obě dimenze zvětšeny tak, aby poměr stran zůstal zachován a tudíž nevznikaly problémy s roztaženými prvky obrázku (e.g. písmo). Pro tuto i budoucí interpolace byla použita Lanczosova metoda. Na obrázcích 5.1 lze pro představu vidět pozadí před budoucími úpravami.

---

<sup>2</sup>Převážně z: <https://fixthephoto.com/free-photoshop-overlays>

<sup>3</sup>Lze získat z: <https://dsl.cds.iisc.ac.in/projects/Anguli/>

Pittsburg, Pa. Feb. 18, 1902  
 Mr F. M. Maddox  
 Austin, Texas.  
 Dear Mr. Frank:  
 I am sending you  
 in this a lot of letters, being a  
 very few of many that I have re-  
 ceived of a like nature. I ask  
 you to read these letters, which will  
 give you some idea of the encourage-  
 ment I am receiving in my work.  
 I call your attention especially  
 to the letter from Gilman Hall, editor  
 of "Reveries" Magazine, dated Feb. 8<sup>th</sup>  
 in answer to one of mine to him  
 saying that I had decided to renounce  
 to New York. I am sure that you  
 will agree with me in thinking  
 that it has the very breath of en-  
 couragement. It was my hope  
 to set out last week for New York,

справедливые!  
 Я изучаю русский язык.  
 У меня не было возможности  
 поговорить с вами лично. Но я  
 хотела поделиться с вами очень  
 важной информацией.  
 Я участвую в работе,  
 которая проводится добровольцами  
 более чем в том, чтобы помочь  
 желающим получить библийские  
 ответы на такие важные вопросы:  
 Почему мы стареем и умираем?  
 В чем смысл жизни? как достичь  
 счастья?  
 Мы занимаемся этой  
 деятельностью не коммерческой.  
 Мы надеемся что вскоре сможем  
 побеседовать с вами лично. С  
 нами можно связаться адрес,  
 указанному вверху.  
 Больше информации вы  
 можете найти на JW.org  
 С УВАЖЕНИЕМ  
 ТАНЯ

The lovely Song night may some long things  
 Western and faraway my heart was beating  
 The roselod on the near the most beautiful  
 The night evening my own love my life  
 To bringe words like most as long sweet  
 Now I love this little but where all heard here  
 Walking near with cold days through the leaves  
 Like things through the dark and cool spring perfume  
 And the dark trees having low shot in case on the road  
 How beautiful the calmness of this lovely summer night!  
 From the old fields with happiness in this the place of gold!  
 I can scarcely grasp the bliss, yet heaven, I could share  
 A thousand nights like this if my darling granted me  
 The lovely Song night may some long things  
 Western and faraway my heart was beating  
 The roselod on the near the most beautiful  
 The night evening my own love my life  
 To bringe words like most as long sweet  
 Now I love this little but where all heard here  
 Walking near with cold days through the leaves  
 Like things through the dark and cool spring perfume  
 And the dark trees having low shot in case on the road  
 How beautiful the calmness of this lovely summer night!  
 From the old fields with happiness in this the place of gold!  
 I can scarcely grasp the bliss, yet heaven, I could share  
 A thousand nights like this if my darling granted me

Obrázek 5.1: Pozadí z třídy „ručně psaný dopis” před úpravami (rozlišení libovolné).

Nyní je třeba načíst obrázek ze kterého extrahujeme  $\alpha$  kanál (ten kanál který nám určuje průhlednost ostatních barev). Tuto vrstvu přes gaussovský filtr s jádrem  $3 \times 3$  rozmážeme, náhodně orotujeme s krokem  $90^\circ$  (pro zvýšení míry unikátnosti). A vrstvu dle pseudo rovnice 5.1 aplikujeme na vrstvu základovou.

$$new\_pixel = stary\_pixel - (0 + stary\_pixel) \frac{\alpha\_pixel}{255} \quad (5.1)$$

Neboli každá nová hodnota pixelu základového obrázku se bude pohybovat v rozmezí od 0 do staré hodnoty pixelu základového obrázku. O tom jak moc se nový pixel přiblíží k nule (pozn.: 0 = černá, 255 = bílá) rozhoduje právě intenzita  $\alpha$  kanálu v daném pixelu. Vzhledem k tomu, že velikost základového obrázku je takřka vždy větší než velikost alpha obrázku, je nutno alpha obrázek buďto roztáhnout (zhoršení kvality) nebo jej, jako v našem případě, nechat se opakovat (za pomoci operace modulo). Toto opakování lze jednoduše vidět prostředním obrázku 5.2.



Obrázek 5.2: Pozadí z třídy „ručně psaný dopis” s libovolnou úrovní poškození (rozlišení libovolné).

Dále bude tento obrázek náhodně orotován s inkrementem  $1^\circ$  a oříznut na požadované rozlišení. Pozice výřezu je taktéž náhodná s tou limitací, že výřez nemůže překročit původní hranici obrázku. V obrázcích 5.3 bylo zvoleno cílové rozlišení  $512 \times 512$ .

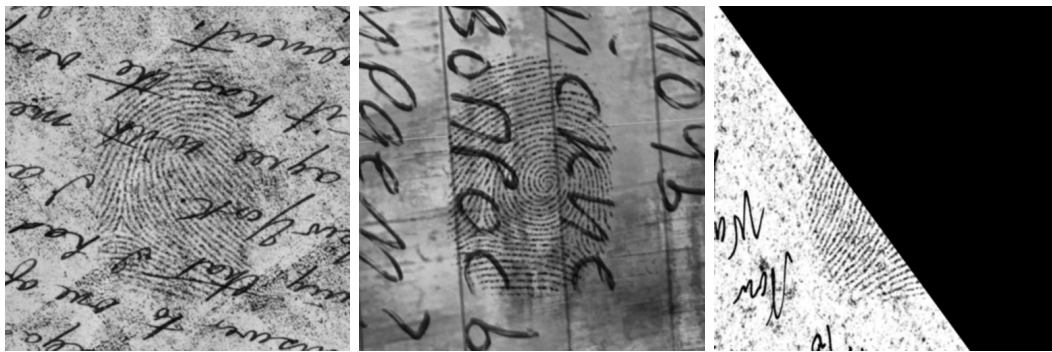


Obrázek 5.3: Orotované a oříznuté pozadí z třídy „ručně psaný dopis” (rozlišení  $512 \times 512 \times 1$ ).

Teď je třeba načíst a umístit syntetický otisk prstu na nově vytvořené pozadí. Kdyby nastala situace při které je otisk větší než samotné pozadí, tak bude rozlišení otisku adekvátně zmenšeno. Posléze se náhodně zvolí jádro gaussovského filtru mezi  $3 \times 3$  a  $5 \times 5$  (větší jádro  $\rightarrow$  více rozostření). Následuje samotná aplikace otisku na pozadí dle pseudo rovnice 5.2.

$$\text{novy\_pixel} = \text{stary\_pixel} - (0 + \text{stary\_pixel}) \left(1 - \frac{\text{print\_pixel}}{255}\right) \quad (5.2)$$

Je zde hodně podobností s rovnicí 5.1 s tím rozdílem, že otisk je vycentrován na pozadí a význam hodnoty v pixelu otisku je opačný oproti významu  $\alpha$  pixelu. A naposled je na celý obrázek aplikovat gaussovský filtr ( $3 \times 3$ ), aby jednotlivé vrstvy lépe splývaly. Finální podobu pozadí společně s otiskem lze vidět na obrázcích 5.4.



Obrázek 5.4: Finální pozadí z třídy „ručně psaný dopis” (rozlišení 512×512×1).

### 5.2.3 Vyhodnocení generátoru

Při pohledu na finální obrázek lze říci, že je kde se zlepšovat obzvláště co se týká výkonu. Ale na to, že je to vcelku skromný program, který slouží jako záložní plán v případě neúspěchu hlavního plánu, generuje slušné výsledky.



# Kapitola 6

## Závěr

Cílem této práce bylo vytvořit generátor pozadí latentních syntetických otisků prstů za pomoci CGAN. Samotné řešení bohužel skončilo neúspěchem díky neschopnosti generovat cokoli jiného než šum. Tento problém by se dal v budoucnu vyřešit implementací dalších metod na zvýšení stability modelu a získání vícero reálných unikátních dat k trénování.

Alternativní řešení však skončilo úspěchem. Tento verdikt potvrdila vizuální kontrola. Co se týká vylepšení je určitě třeba zapracovat na výkonu a zlepšit kvalitu vrstev, dále by nebylo na škodu udělat nějaké základní UI.

V této úvodní kapitole 1 je zasazení práce do širšího kontextu, nastínění o čem práce bude a základní výčet co v jaké bude. V kapitole plné teorie 2 byla nastíněna biometrie, kriminalistika, otisky reálné i syntetické, jejich pozadí a poškození. V další kapitole byla 3 umělá inteligence - na co se dělí; neuronová síť - z čeho se skládá a její různé typy. Posléze v kapitole 4, která je kombinace teorie a praktického návrhu generativního modelu. Kapitola 5 byla implementace dříve předešle zmíněného návrhu, implementace alternativy a následné vyhodnocení výsledků.

# Literatura

- [1] ANSARI, A. H. Generation and storage of large synthetic fingerprint database. *ME Thesis, Jul.* 2011, s. 47.
- [2] CAPPELLI, R., FERRARA, M. a MALTONI, D. Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. *IEEE transactions on pattern analysis and machine intelligence*. Prosinec 2010, sv. 32, s. 2128–41. DOI: 10.1109/TPAMI.2010.52.
- [3] CAPPELLI, R., MAIO, D. a MALTONI, D. SFinGe: an approach to synthetic fingerprint generation. In: *International Workshop on Biometric Technologies (BT2004)*. 2004, s. 147–154.
- [4] CHALOUPKA, R. Generátor otisků prstů. *Bakalárska práca, Vysoké učenie technické v Brne, Fakulta informačných technológií*. 2007, s. 47.
- [5] CHARLOT, B., PARRAIN, F., GALY, N., BASROUR, S. a COURTOIS, B. A sweeping mode integrated fingerprint sensor with 256 tactile microbeams. *Journal of Microelectromechanical Systems*. IEEE. 2004, sv. 13, č. 4, s. 636–644.
- [6] CHOLLET, F. o. *Deep learning with python, second edition*. Manning Publications, 2022.
- [7] DEEPAI. *Feed Forward Neural Network* [online]. DeepAI [cit. 2022-4-01]. Dostupné z: <https://deeptai.org/machine-learning-glossary-and-terms/feed-forward-neural-network>.
- [8] DEVRIES, T. a TAYLOR, G. W. Improved regularization of convolutional neural networks with cutout. *ArXiv preprint arXiv:1708.04552*. 2017.
- [9] DRAHANSKÝ, M., ORSÁG, F., DOLEŽEL, M. et al. *Biometrie*. 1. vyd. Computer Press a.s., 2011. ISBN 978-80-254-8979-6.
- [10] EDUCATION, I. C. *Convolutional Neural Networks* [online]. IBM, říjen 2020 [cit. 2022-4-01]. Dostupné z: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>.
- [11] EDUCATION, I. C. *Deep Learning* [online]. IBM, květen 2020 [cit. 2022-4-01]. Dostupné z: <https://www.ibm.com/cloud/learn/deep-learning>.
- [12] EDUCATION, I. C. *Neural Networks* [online]. IBM, srpen 2020 [cit. 2022-4-01]. Dostupné z: <https://www.ibm.com/cloud/learn/neural-networks>.

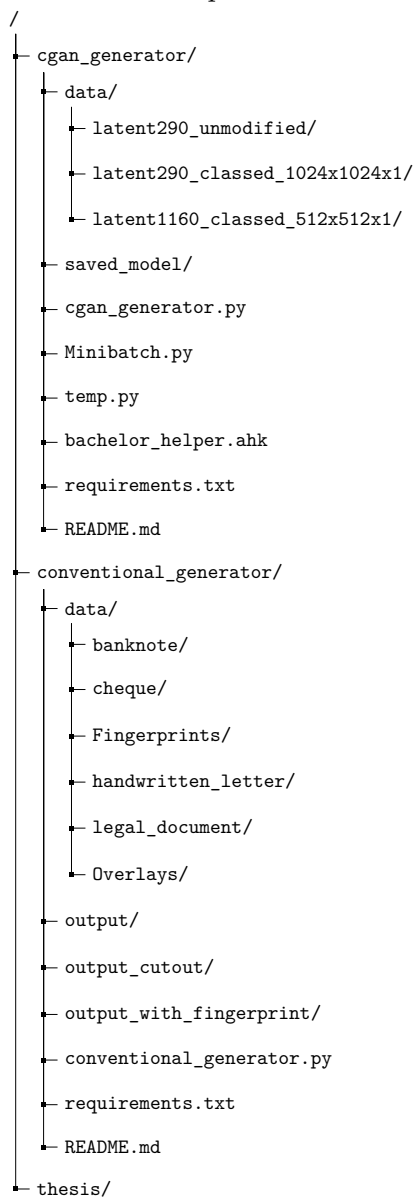
- [13] EDUCATION, I. C. *Recurrent Neural Networks* [online]. IBM, říjen 2020 [cit. 2022-4-01]. Dostupné z: <https://www.ibm.com/cloud/learn/recurrent-neural-networks>.
- [14] FENG, J. a JAIN, A. K. Fingerprint reconstruction: from minutiae to phase. *IEEE transactions on pattern analysis and machine intelligence*. IEEE. 2010, sv. 33, č. 2, s. 209–223.
- [15] JULIANNA DELUA, I. A. D. S. L. *Supervised vs. Unsupervised Learning: What's the Difference?* [online]. IBM, březen 2021 [cit. 2022-4-01]. Dostupné z: <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning>.
- [16] JUSTICE, T. U. S. D. of. *Forensic science*. Nov 2021 [cit. 26.12.2021]. Dostupné z: <https://www.justice.gov/olp/forensic-science>.
- [17] JUSTICE, U. S. D. of. *The Fingerprint Sourcebook*. CreateSpace Independent Publishing Platform, 2014. ISBN 9781502828422. Dostupné z: <https://books.google.cz/books?id=L0sgogEACAAJ>.
- [18] KANICH, O. a DRAHANSKÝ, M. Simulation of synthetic fingerprint generation using Petri nets. *IET Biometrics*. IET. 2017, sv. 6, č. 6, s. 402–408.
- [19] KANICH, O. *Research in Fingerprint Damage Simulations*. Brno, CZ, 2018. Disertační práce. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [20] KANICH, O., KOŠŤÁK, D. a DRAHANSKÝ, M. Psoriasis Damage Simulation into Synthetic Fingerprint. In: *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2019, s. 1–4.
- [21] KERAS. *Keras layers API* [online]. Keras [cit. 2022-4-01]. Dostupné z: <https://keras.io/api/layers/>.
- [22] KING OF KNIGHTS. *Keras-ACGAN-CIFAR10* [online]. King-Of-Knights, leden 2018 [cit. 2022-4-01]. Dostupné z: <https://github.com/King-Of-Knights/Keras-ACGAN-CIFAR10>.
- [23] KINGMA, D. P. a BA, J. *Adam: A Method for Stochastic Optimization*. arXiv, 2014. DOI: 10.48550/ARXIV.1412.6980. Dostupné z: <https://arxiv.org/abs/1412.6980>.
- [24] KRIZHEVSKY, A., SUTSKEVER, I. a HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012, sv. 25, s. 1097–1105.
- [25] KÜCKEN, M. a NEWELL, A. C. A model for fingerprint formation. *EPL (Europhysics Letters)*. IOP Publishing. 2004, sv. 68, č. 1, s. 141.
- [26] LANGR, J. a BOK, V. *Gans in action: Deep learning with generative adversarial networks*. Manning, 2019.
- [27] MALTONI, D. *Handbook of Fingerprint Recognition*. Springer, 2009. 512 s.
- [28] MIRZA, M. a OSINDERO, S. Conditional generative adversarial nets. *ArXiv preprint arXiv:1411.1784*. 2014.

- [29] PRAMODITHA, R. *The Concept of Artificial Neurons (Perceptrons) in Neural Networks* [online]. Towards Data Science, prosinec 2021 [cit. 2022-4-01]. Dostupné z: <https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc>.
- [30] Q., Z., K., J. A., G., P. N. et al. : Fingerprint Image Synthesis Based on Statistical Feature Models. *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), IEEE*. 2012, 978-14-673-1384-1, s. 23–30.
- [31] RADFORD, A., METZ, L. a CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv, 2015. DOI: 10.48550/ARXIV.1511.06434. Dostupné z: <https://arxiv.org/abs/1511.06434>.
- [32] SIMARD, P. Y., LECUN, Y. A., DENKER, J. S. a VICTORRI, B. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In: *Neural networks: tricks of the trade*. Springer, 1998, s. 239–274.
- [33] SOKAL, R. R. a ROHLF, F. J. Biostatistics. *Francise & Co, New York*. 1987.
- [34] UNIVERSITY, W. G. *Neural networks and deep learning explained* [online]. Western Governors University, březen 2020 [cit. 2022-4-01]. Dostupné z: <https://www.wgu.edu/blog/neural-networks-deep-learning-explained2003.html#openSubscriberModal>.
- [35] YANUSHKEVICH, S. Synthetic Biometrics: A Survey. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. 2006, s. 676–683. DOI: 10.1109/IJCNN.2006.246749.
- [36] ZHANG, H., CISSE, M., DAUPHIN, Y. N. a LOPEZ PAZ, D. Mixup: Beyond empirical risk minimization. *ArXiv preprint arXiv:1710.09412*. 2017.

# Příloha A

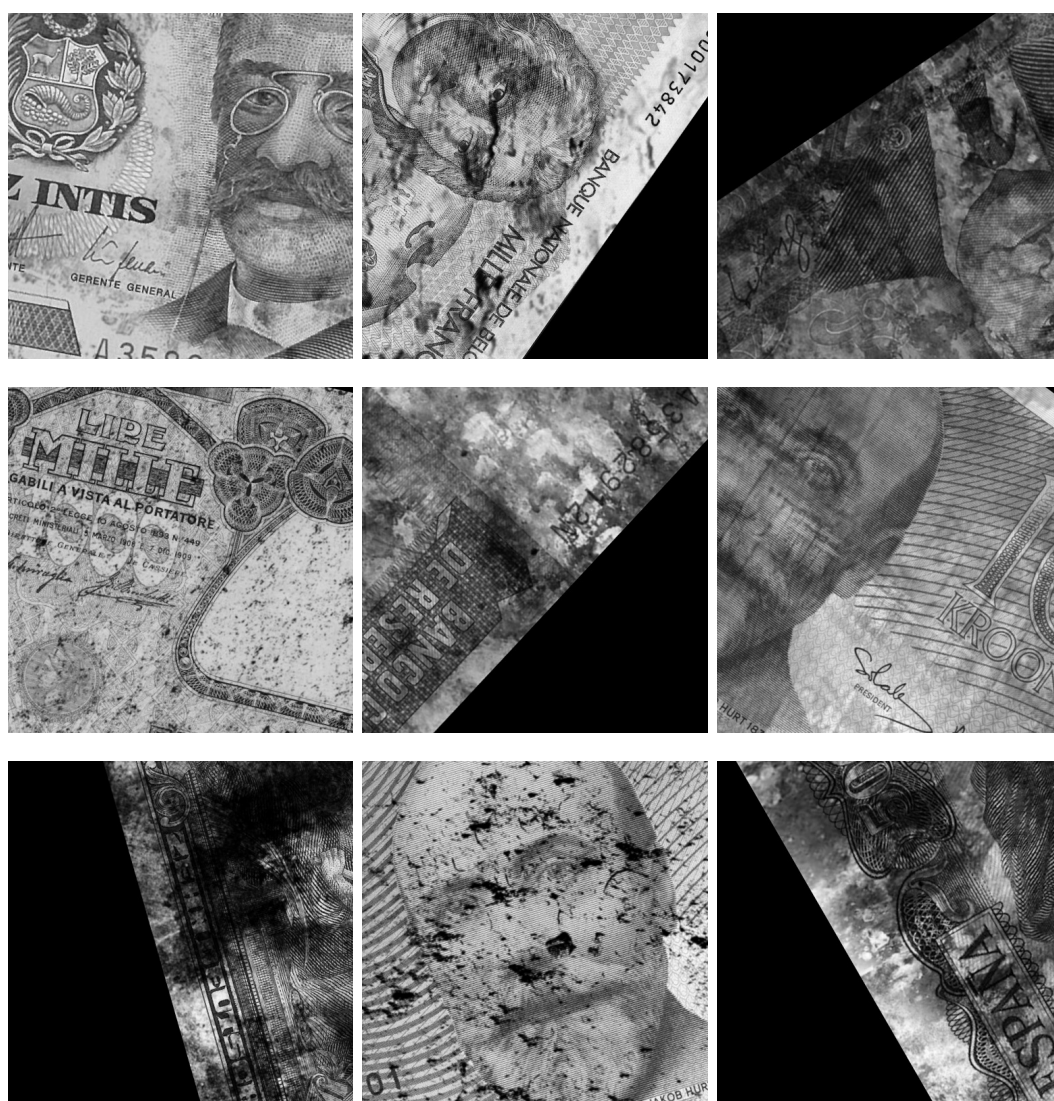
## Obsah přiloženého média

Stromová struktura přiloženého média.

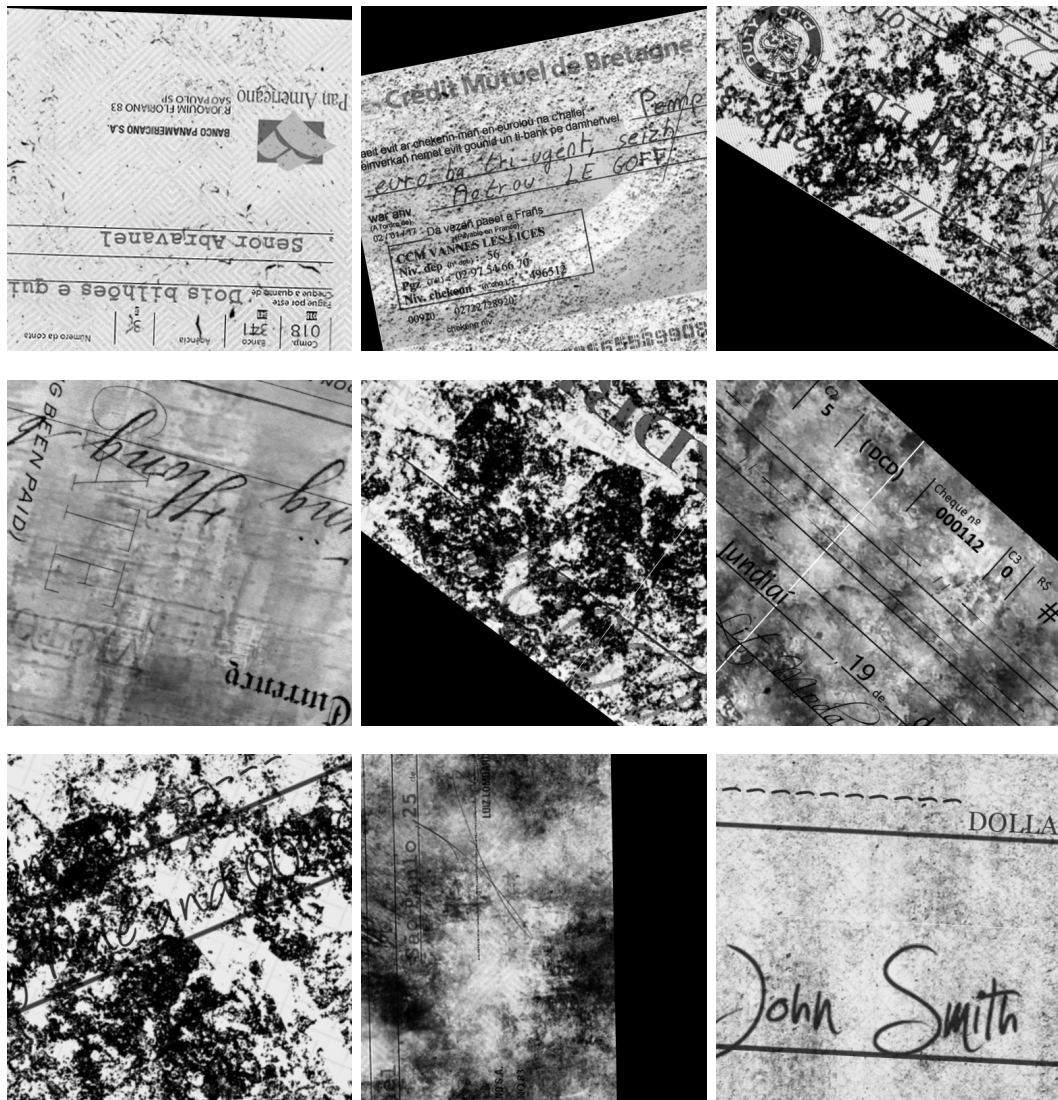


## Příloha B

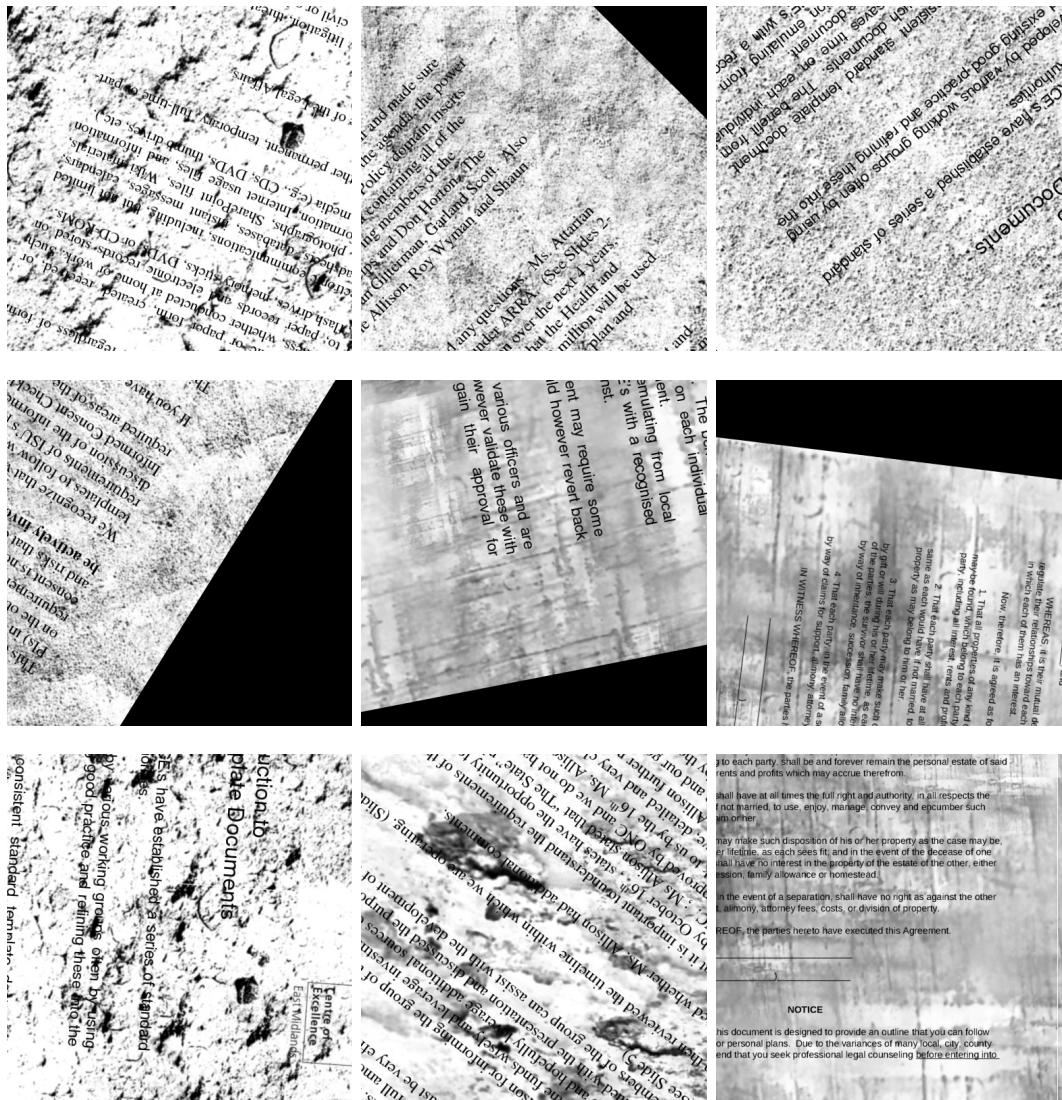
### Ukázka pozadí - konvenční metoda



Obrázek B.1: Vygenerované pozadí konvenční metodou. Pozadí z třídy „bankovka” s libovolnou úrovní poškození (rozlišení 512×512×1).



Obrázek B.2: Vygenerované pozadí konvenční metodou. Pozadí z třídy „šek” s libovolnou úrovní poškození (rozlišení 512×512×1).



Obrázek B.3: Vygenerované pozadí konvenční metodou. Pozadí z třídy „Právní dokument“ s libovolnou úrovní poškození (rozlišení 512×512×1).





Obrázek B.4: Vygenerované pozadí konvenční metodou. Pozadí z třídy „Ručně psaný dopis” s libovolnou úrovní poškození (rozlišení 512×512×1).