

# **Vizualizace a hodnocení augmentačních technik v počítačovém vidění**

Bc. Petr Buček









## PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

### Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

### Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne .....

.....  
podpis autora



## **ABSTRAKT**

Text abstraktu v jazyce práce.

Klíčová slova: klíčové slovo, klíčové slovo

## **ABSTRACT**

Text abstraktu ve světovém jazyce (angličtině).

Keywords: keyword, keyword

Zde je místo pro případné poděkování, popř. motto, úryvky knih atp.



# OBSAH

<b>ÚVOD .....</b>	<b>11</b>
<b>I    TEORETICKÁ ČÁST .....</b>	<b>12</b>
<b>1    AUGMENTACE DAT V POČÍTAČOVÉM VIDĚNÍ .....</b>	<b>13</b>
1.1    PROBLEMATIKA KVALITY A MNOŽSTVÍ DAT V POČÍTAČOVÉM VIDĚNÍ .....	13
1.2    ÚLOHA AUGMENTACE DAT PŘI ZLEPŠOVÁNÍ VÝKONU MODELŮ HLUBO- KÉHO UČENÍ .....	14
1.3    KLASIFIKACE A CHARAKTERISTIKA AUGMENTAČNÍCH METOD .....	15
1.3.1    Geometrické transformace (rotace, škálování, ořez, zrcadlení) .....	15
1.3.2    Intenzivní transformace (změna jasu, kontrastu, aplikace šumu) .....	16
1.3.3    Pokročilé augmentační metody (Cutout, Mixup, CutMix, GAN-based augmentace) .....	16
1.3.4    Doménově specifické augmentační techniky .....	18
<b>2    VIZUALIZACE DOPADU AUGMENTACÍ .....</b>	<b>19</b>
2.1    PRINCIP METODY T-SNE A JEJÍ MATEMATICKÝ ZÁKLAD .....	19
2.2    APLIKACE T-SNE V KONTEXTU ANALÝZY AUGMENTOVANÝCH DAT .....	20
2.3    MOŽNOSTI INTERPRETACE VÝSLEDKŮ T-SNE VIZUALIZACE .....	21
<b>3    HODNOCENÍ VLIVU AUGMENTACÍ NA DIVERZITU DAT .....</b>	<b>23</b>
3.1    METRIKY PRO KVANTIFIKACI DIVERZITY A ROZMANITOSTI DAT .....	23
3.2    ANALÝZA VLIVU JEDNOTLIVÝCH AUGMENTAČNÍCH TECHNIK NA DIS- TRIBUCI DAT .....	26
3.2.1    Kvantifikace dopadu augmentací na distribuci dat .....	27
3.3    KOMPARATIVNÍ HODNOCENÍ EFEKTIVITY RŮZNÝCH PŘÍSTUPŮ K AUG- MENTACI .....	27
<b>II   PRAKTICKÁ ČÁST .....</b>	<b>29</b>
<b>4    IMPLEMENTACE A EXPERIMENTÁLNÍ POSTUP .....</b>	<b>30</b>
4.1    POPIS DATOVÉHO SOUBORU .....	30
4.2    VÝBĚR AUGMENTACÍ .....	31
4.3    IMPLEMENTACE AUGMENTACÍ .....	32

4.4	VÝBĚR METRIK PRO HODNOCENÍ AUGMENTACÍ .....	34
4.5	IMPLEMENTACE VYBRANÝCH METRIK .....	34
4.5.1	Variance Diversity .....	34
4.5.2	Affinity, Diversity Index .....	35
4.5.3	Feature Space Coverage .....	36
4.5.4	Domain Shift Score .....	36
4.5.5	Cluster Purity .....	37
4.5.6	KL Divergence.....	38
4.5.7	Silhouette Score .....	38
<b>ZÁVĚR.....</b>		<b>39</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>		<b>41</b>
<b>SEZNAM OBRÁZKŮ .....</b>		<b>43</b>
<b>SEZNAM TABULEK .....</b>		<b>44</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>		<b>45</b>
<b>SEZNAM PŘÍLOH.....</b>		<b>46</b>

## ÚVOD

První řádek prvního odstavce v kapitole či podkapitole se neodsazuje, ostatní ano. Vertikální odsazení mezi odstavci je typické pro anglickou sazbu; czech babel toto respektuje, netřeba do textu přidávat jakékoliv explicitní formátování, následuje ukázka sazby tohoto a následujících odstavců). Je vhodné znovu připomenout, že autor práce by si měl pohlídat, čím končí řádky. Určitě by to neměly být jednohláskové spojky ani předložky. Pro takové případy zde máme tildu (~), kterou L<sup>A</sup>T<sub>E</sub>X vyhodnocuje jako nezlomitelnou mezeru. Příklad použití, ze kterého se můžete poučit, je uveden např. právě na konci tohoto řádku. Konec řádku zde a další řádek hned za ním, aneb podívejte se do zdrojového kódu za spojku „a“.

Šablona je nastavena na oboustranný tisk. Za tímto účelem obsahuje na svém počátku místy prázdnou stranu, aby např. zadání začínalo na liché stránce (vpravo).

## **I      TEORETICKÁ ČÁST**

## 1 AUGMENTACE DAT V POČÍTAČOVÉM VIDĚNÍ

Velké pokroky v posledních letech v oblastech strojového učení a počítačového vidění nám odhalily nové překážky na vyřešení. Zejména efektivita počítačové vidění závisí na třech složkách: schopný hardware, kvalitní algoritmy a početné a rozmanité datové sady. Zatímco jsme byly svědky toho jak efektivita a výkon hardwaru roste a algoritmy se stávají sofistikovanější, tak sběr kvalitních dat je stále stejně drahý a náročný proces. Tento problém se snaží řešit právě augmentace dat. Augmentace dat zaštiťuje soubor metod, které upravují nebo rozšiřují trénovací datové sady a tím zajišťují robustnost a přesnost modelů. [1] Tato kapitola se bude zabývat přehledem augmentačních technik v oblasti počítačového vidění. V následujících sekcích jsou analyzovány jednotlivé aspekty augmentací datových sad počínaje problematikou kvality datových sad, přes samotnou úlohou augmentací ve zlepšování modelů až po jejich praktické aplikace.

### 1.1 Problematika kvality a množství dat v počítačovém vidění

V disciplíně počítačového vidění je kvalita a velký objem rozmanitých trénovacích dat stěžejním faktorem, který výrazně ovlivňuje úspěšnost modelů. Ovšem reálné datové sady jsou ve skutečnosti nekvalitní, neúplné a často nevyvážené a vedou tedy k nepřesnosti a preferenci určitých tříd nad ostatními.

Kvalitou dat rozumíme několik aspektů, kde prvním z nich je rozmanitost zachycené scény, osvětlení, pozic a dalších různých proměnných, které odráží dynamiku reálného světa. Pokud je model trénován na omezeném spektru této rozmanitosti, tak je jeho efektivita v neobvyklých situacích častokrát nedostačující. To je způsobeno dynamickou povahou reálného světa, kdy se světelné podmínky, úhly pohledu, pozice předmětů, rotace kamery a další aspekty v čase mění. V ideálním případě by měla trénovací datová sada všechny tyto skutečnosti odrážet, ovšem v realitě je to v některých případech nemožné. Dalším důležitým aspektem je přesnost anotací popisů scény, jelikož s nepřesnými nebo chybnými vstupy je supervizované strojové učení předem odsouzeno k nezdaru. Další je aspekt rovnoměrnosti jednotlivých tříd datové sady, zajišťující, že model nebude držet preferenci k určitým třídám. Posledním aspektem je pak samotná velikost trénovacích dat. Hluboké neuronové sítě vyžadují nepřeborné množství dat pro efektivní natrénování s velkou přesností.

## 1.2 Úloha augmentace dat při zlepšování výkonu modelů hlubokého učení

Jak již bylo nastíněno augmentace dat nabízí řešení pro mnoho výzev spojených s množstvím a kvalitou trénovacích datových sad. Největší přínos spočívá v umělém rozšíření vzorů, změnou domény, nebo zvýšení rozmanitosti bez nutnosti dalšího sběru a anotace dat. Tím augmentace zaštiťují efektivní způsob navýšení velikosti a rozmanitosti trénovacích dat, což vede k lepšímu výsledku modelu.

Efektivita konvolučních neuronových sítí v oblasti hlubokého učení je úzce spjata s dostupností rozsáhlé a kvalitní datové sady. Velké množství kvalitních dat zajišťuje snížení rizika přeučení modelu. Ovšem realita je taková, že v mnoha aplikačních odvětvích, jako je například analýza medicínských snímků, často nejsou rozsáhlé datové sady dostupné [2]. To může být způsobeno různými faktory:

- Sběr takto specializovaných dat je vysoce nákladný
- V datové sadě se nachází nerovnoměrné zastoupení různých tříd a data jsou tedy nevyvážená
- Anotace dat je vysoce časově náročná
- Existence etických a právních omezení týkající se ochrany osobních údajů

Z těchto důvodů často vzejde neúplná datová sada a jejím klíčovým řešením se stává právě augmentace dat, která dokáže původní datovou sadu obohatit a model může dosahovat výsledků srovnatelných s těmi, které by byly dosaženy na robustních a kvalitních reálných datových sadách.

V oblasti počítačového vidění jeden ze skalních problémů je přeučení, k němuž dochází když se model přílišně naučí kopírovat trénovací data včetně artefaktů jako je šum nebo ne-realistických či neexistujících souvislostí. Augmentace dat tento problém řeší následujícími metodami:

- Zvýšení variability dat: Původní vzorky jsou obohaceny různými transformacemi, čímž se zajistí odolnost modelu vůči specifickým variacím, které transformace odrážejí. Model se poté dokáže lépe zaměřit na obecnější charakteristiky v datové sadě místo kopírování specifických detailů dané trénovací sady. - Umělé rozšíření: Augmentační metody dokáží s

efektivitou zvýšit počet trénovacích vzorků, to vede k lepší generalizaci modelu. - Vyvážení dat: Augmentace pomáhají zvýšit a tím vyrovnat zastoupení málo reprezentovaných tříd v dané datové sadě.

### 1.3 Klasifikace a charakteristika augmentačních metod

Klasifikovat augmentační metody lze různými způsoby, nejčastěji je lze klasifikovat podle typu transformací prováděnými nad daty. V této podkapitole jsou popsány hlavní kategorie augmentačních metod. Dále jsou zmíněny jejich charakteristiky, omezení a praktické využití v oblasti trénování počítačového vidění.

#### 1.3.1 Geometrické transformace (rotace, škálování, ořez, zrcadlení)

Tyto transformace znázorňují nejjednodušší možné metody augmentace dat. Modifikují prostorové uspořádání obrazu, ale přitom zachovávají jeho obsah. Mezi nejpoužívanější techniky patří:

- **Rotace:** Otáčení obrazu o daný úhel, čímž zvyšujeme odolnost modelu vůči změně orientaci objektů v prostoru.
- **Škálování:** Změna velikosti obrazu v dané ose o daný násobek, který pomáhá modelu rozpoznávat různě vzdálené objekty, nebo různé velikosti stejných objektů.
- **Ořez:** Ořezání dané části obrazu, tedy zobrazení specifické části obrazu, simuluje částečně viditelné objekty.
- **Zrcadlení:** Horizontální nebo vertikální převrácení obrazu, obecně zvyšuje rozmanitost dat.

Tyto metody jsou snadné na implementaci, jsou pro lidi intuitivní a výpočetně nenáročné. Intuitivnost vychází z faktu, že tyto transformace odpovídají reálným změnám, ke kterým dochází při pořizování obrazu, jako jsou různé úhly pohledu, vzdálenost a orientace kamery a pozice objektů v prostoru. Tyto transformace zachovávají obsah obrazu, tedy transformovaný obraz stále patří do stejné třídy jako ten původní. Ovšem některé transformace mohou vést ke ztrátě informací jako je to například u ořezu. Omezení takových augmentačních metod pramení z faktu, že nepomohou modelu porozumět změnám negeometrické povahy, jako jsou například změny osvětlení.

### 1.3.2 Intenzivní transformace (změna jasu, kontrastu, aplikace šumu)

Tyto transformační techniky jsou stěžejní pro zlepšování odolnosti modelů počítačového vidění vůči různým světelným podmínkám, nebo různorodé kvality v obrazových datových sadách. Tato odolnost modelu je důležitá zejména tak, protože reálná obrazová data jsou přímo ovlivněna světelnými podmínkami, ve kterých byly zachyceny, nebo také kvalitou zařízení jež je zachytilo. Tyto transformační techniky jsou často jednoduše implementovatelné, přičemž ale nabízí zvýšení přesnosti modelu v reálných podmínkách s horší kvalitou vstupních dat.

- **Změna jasu:** Úprava jasu je na datovou sadu aplikována tak, že se modifikuje intenzita pixelů v obraze, tedy nezmění se jeho geometrické vlastnosti. Tato úprava simuluje reálné odchylky ve světelných podmínkách a modely, které jsou trénované na takových datových sadách jsou lepší v rozpoznávání objektů, které například byly zachyceny v jinou denní dobu, než data původní. Bez takových augmentací by se mohlo stát, že model by byl přetrénovaný při specifických světelných podmínkách, například by mohl správně určit objekt pouze při denním světle. [3]
- **Změna kontrastu:** Úprava kontrastu je na datovou sadu aplikována tak, že mapuje hodnoty intenzity obrazu na nový rozsah. Tím se mění rozdíl mezi nejtmašími a nejsvětlejšími částmi obrazu. [4] Model natrénovaný na datové sadě augmentované touto metodou je lepší v rozlišování důležitých rysů při různých kontrastních podmínkách.
- **Aplikace šumu:** Tato augmentační technika záměrně zanáší do obrazových dat různé typy šumu, jako například Gaussův šum, Poissonův šum, nebo tzv. salt-and-pepper šum. Cílem je obrnit daný model vůči drobným nepřesnostem a poruchám v obrazových datech. Šum se totiž velmi často vyskytuje v reálných obrazových datech, většinou vlivem špatných světelných podmínek, nebo kvalitou zachytávajícího zařízení.

### 1.3.3 Pokročilé augmentační metody (Cutout, Mixup, CutMix, GAN-based augmentace)

Pokročilé augmentační metody jsou robustnější a sofistikovanější metody augmentace dat. Tyto metody vznikly s cílem zlepšit generalizaci modelů a snížit riziko přetrénování.

- **Cutout:** Tato metoda náhodně zakrývá části obrazových dat. Potřeba těchto augmentací pramení z faktu, že v reálných obrazových datech zachycené objekty mohou být



částečně skryty. Podle zdroje [5] model naučený na takto augmentovaných datech bere více v potaz kontext obrazu při rozhodování. Tato metoda může pomoci proti přílišné fixaci modelu na nepodstatné prvky. Například model, který je trénovaný na medicínských snímcích se může lehce přetrénovat na artefakty zanechané samotným zařízením, jako je například značka skeneru někde na snímku, ale tato augmentační metoda mu pomůže se zaměřit na důležitý obsah snímku, tím že tyto artefakty zakryje [6].

- **Mixup:** Je metodou, která se snaží o minimalizaci okolního rizika, tedy generuje nové vzorky v blízkosti původních dat, spojením dvou obrazových dat. Tedy nevytváří nové varianty původních dat, ale vytváří zcela nové trénovací data kombinací existujících dat. Tyto nová data jsou vytvořena například pomocí lineární interpolací dvou obrazových dat a jejich odpovídajících štítků. Tato metoda může vést k lepší generalizaci a robustnosti modelu [7].
- **CutMix:** Kombinuje obě dříve zmíněné augmentační metody v jednu. Jedná se tedy o kombinaci metody Cutout a Mixup. Princip této metody spočívá v tom, že části obrazových dat jsou vyřezávány a poté jsou zase vkládány mezi trénovací obrazová data, přičemž štítky jsou také míchány proporcionálně k ploše a poloze vyřezaných oblastí.
- **GAN-based augmentace:** Představuje augmentační metodu, která umí vytvářet zcela nová realistická data na základě distribuce trénovacích dat. Toho dosahují pomocí generativních adversiálních sítí (GAN). GAN je kombinací dvou neuronových sítí určené ke generování nových dat, která se snaží co nejvíce napodobit vlastnosti trénovacích vzorků. Jedná se dvě neuronové sítě: generátor a diskriminátor, které spolu soupeří v tzv. adversiálním tréninku. Generátor se snaží vygenerovat výstupní nové vzorky dat, které budou co nejvíce podobné původním datům z trénovací množiny. Diskriminátor se poté snaží rozeznat nově generovaná data od těch původních trénovacích. Tím je zajištěno střídavé zlepšování těchto dvou neuronových sítí a vznikne tak generativní model, který dokáže s velkou přesností napodobit data z trénovací množiny [8]. Ve srovnání s ostatními metodami augmentací dat, tato metoda poskytuje robustnější obohacení datové sady zcela novými vzorky dat, namísto pouhých transformací dat původních.

### 1.3.4 Doménově specifické augmentační techniky

Doménově specifické augmentační techniky představují soubor pokročilých metod obohacování datových sad, které jsou uzpůsobeny konkrétním charakteristikám dané aplikační oblasti počítačového vidění. Tyto metody jsou využívány zejména v kontextu vysoce specializovaných typů úloh. Například v oblasti zdravotnictví jsou tyto metody často využívány zejména proto, že reálné medicínské snímky jsou často v počtu omezené, zvláště tak u chorob, které jsou vzácné. Doménově specifické augmentace v medicínském snímkování musí zachovat relevantní informace pro diagnózu [9]. Dále když navštívíme oblast autonomních vozidel, zde jsou doménově specifické augmentace klíčové pro zajištění dobrého výkonu modelu v různých podmínkách. Aby bylo zajištěno, že model dokáže správně fungovat za různých světelných podmínek a za různého počasí, jsou použity specializované augmentační metody, které musí umět tyto podmínky nasimulovat a napasovat na původní data. Tyto metody lze charakterizovat těmito klíčovými vlastnostmi:

- Zachovávají důležité informace specifické pro danou doménu
- Využívají znalostí struktury a sémantiky dat v dané doméně
- Simulují realistické variace specifické pro danou doménu

Tímto způsobem doménově specifické augmentační techniky nabízejí mocný nástroj na efektivní tvorbu nových dat, které mohou být obtížné na získání a dokáží natrénovat modely, které excelují na specifických typech úloh.

## 2 VIZUALIZACE DOPADU AUGMENTACÍ

### 2.1 Princip metody t-SNE a její matematický základ

Vizualizace dat je klíčový prvek k pochopení a interpretaci jak augmentace upravují původní datovou sadu. Tato kapitola se zaměří na metodu t-SNE (t-Distributed Stochastic Neighbor Embedding), která slouží k vizualizaci vysokodimenzionálních dat ve 2D nebo 3D prostoru, při zachování shluků ve vysokých dimenzích. Tím lze získat jakousi intuici, jak jsou data uspořádána ve vyšších dimenzích.

Základ této metody vychází z metody SNE, celým názvem Stochastic Neighbor Embedding, která převádí vysokodimenzionální euklidovské vzdálenosti datových bodů do podmíněných pravděpodobností představujících jejich podobnost. Podmíněná pravděpodobnost má tento matematický vzorec:

$$p_{j|i} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$

Podmíněná pravděpodobnost  $p_{j|i}$  přímo odpovídá podobnosti bodu  $x_i$  a bodu  $x_j$ .  $\sigma_i$  značí Gaussovo rozdělení, které je vycentrováno nad bodem  $x_i$ . Ve své podstatě  $p_{j|i}$  určuje, zda by si bod  $x_i$  vybral bod  $x_j$  za svého souseda, pokud by sousedé byly vybírány podle proporce pravděpodobností hustoty Gaussova rozdělení  $\sigma_i$ . Pro datové body, které jsou blízko sebe nabývá  $p_{j|i}$  relativně vysokých hodnot, zatímco body, které jsou od sebe daleko nabývá hodnot téměř nulových. [10]

Metoda SNE sice dokáže sestavit dobrou vizualizaci, ale je těžké optimalizovat nákladovou funkci a vzniká tak problém přeplnění bodů. Další problém nastává z povahy SNE a to, že podmíněné pravděpodobnosti jsou v asymetrii. Tato metoda je tedy rozšířena o novou techniku zvanou "t-Distributed Stochastic Neighbor Embedding", která tyto problémy řeší. Děje se to pomocí zavedením symetrické verze pravděpodobnosti a následním použitím t-distribuce namísto normálního rozdělení.

Symetrická verze SNE metody pracuje se společnými pravděpodobnostmi namísto podmíněných pravděpodobností, tedy místo výpočtu  $p_{j|i}$  se používá symetrická pravděpodobnost:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

kde  $n$  značí celkový počet datových vzorků. Symetrizace zjednodušuje gradient při optimalizaci a algoritmus je stabilnější.

T-distribuce v nízkodimenzionálním prostoru využívá Student-t distribuci na rozdíl od normálního rozdělení pro výpočet podobnosti bodů s jedním stupněm volnosti:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

kde  $y_i$  a  $y_j$  jsou body v nízkodimenzionálním prostoru. Distribuce Student-t má takovou vlastnost, že vzdálené body ve vysokodimenzionální prostoru jsou reprezentovány jako vzdálené i v nízkodimenzionálním prostoru, přičemž ale nedochází k penalizaci jako u normálního rozdělení.

Perplexita je klíčový parametr algoritmu t-SNE, který určuje efektivní počet nejbližších sousedů pro každý datový bod a je definována jako:

$$\text{Perp}(P_i) = 2^{H(P_i)}, \quad \text{kde } H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

kde  $H(P_i)$  je Shannonova entropie pravděpodobnostního rozdělení  $P_i$ . Provádí se binární hledání hodnoty ( $\omega_i$ ) tak, aby perplexita u každého z bodů odpovídala zadané hodnotě.

## 2.2 Aplikace t-SNE v kontextu analýzy augmentovaných dat

Vizualizace vysokodimenzionálních dat pomocí t-SNE je účinným nástrojem pro analýzu dopadu augmentačních metod na distribuci dat v kontextu počítačového vidění. Díky této vizualizační metodě jsme schopni porovnat distribuce dat před a po aplikaci augmentací v dvourozměrném prostoru, to nám pomáhá posoudit, jak jednotlivé augmentační metody ovlivňují strukturu dat. Při analýze je také důležité zhodnotit, jestli augmentační metoda správně rozšiřuje datový prostor, nebo zda vytváří vzory nerealistické pro přirozenou variabilitu původních dat. Podle příkladů augmentací z první kapitoly této práce může být struktura ovlivněna například:

- **Geometrické transformace** by měli vytvářet body, které vytvářejí shluky kolem původních neaugmentovaných bodů, přičemž ale jejich distribuci jemně rozšiřují.

- **Intenzivní transformace** by měli ve vizualizaci vytvářet jakési "paprsky", které vychází z původních bodů. Paprsky jsou tvořeny tak, že tento typ transformací ve vizualizaci body lehce rozptyluje do blízkého okolí původních bodů buďto směrem od nebo do středu, či radiálně. Každý z paprsků pak reprezentuje konkrétní transformaci.
- **Pokročilé augmentační metody** mohou ve vizualizaci vytvářet zcela nové shluky.

Standardní implementace t-SNE vizualizace zachovává lokální strukturu dat, to následně může vést ke ztrátě informací o globální struktuře, což se může projevit při hodnocení vlivu augmentací, jež mění i globální vlastnosti dat. Tento problém řeší rozšířený algoritmus globální t-SNE (g-SNE), který uvažuje navíc s globální nákladovou funkcí k původní nákladové funkci t-SNE algoritmu:

$$C_{\text{g-SNE}} = C_{\text{t-SNE}} + \lambda C_{\text{global}}$$

Kde váhový parametr  $\lambda$  kontroluje rovnováhu mezi lokálními shluky a globální distribucí dat. Tato globální nákladová funkce na rozdíl od původní nákladové funkce je citlivá na velké vzdálenosti v datovém prostoru a tedy dokáže lépe zachovat vztahy mezi shluky [1].

Dopad augmentačních metod lze sledovat i v průběhu učení modelu. Extrahované příznaky lze vizualizovat z jakékoliv vrstvy sítě. Pokud tak učiníme před a po aplikaci augmentačních metod, získáme poznatky o tom, jak síť interpretuje augmentovaná data v různých fázích učení a to nám dokáže nabídnout lepší porozumění jak augmentace dokáží ovlivnit učení modelu.

Pomocí tohoto vizualizačního nástroje lze identifikovat augmentace, které dokáží pomoci modelu lépe generalizovat, nebo také rozpoznat augmentace, které naopak mohou způsobit přetrénování nebo jiné problémy při učení modelu. Také lze optimalizovat augmentační strategii dle vizuální zpětné vazby jejího dopadu na augmentovaná data.

### 2.3 Možnosti interpretace výsledků t-SNE vizualizace

Při interpretaci t-SNE vizualizace je potřeba abychom měli na paměti několik úskalí této metody. Kvůli povaze t-SNE zachovávání lokálních vzdáleností, nemusí vzdálenosti mezi jednotlivými shluky odpovídat vzdálenostem v původním datovém prostoru. Také jejich velikost a hustota nemusí být reprezentativní k reálné velikosti a hustotě v datovém prostoru. Dále

změna parametrů algoritmu, jako je například parametr perplexity, může ovlivnit výslednou vizualizaci. Stochastická povaha algoritmu znamená, že algoritmus může produkovat mírně odlišné vizualizace pro různé běhy.

Z vizualizace můžeme zkoumat tyto aspekty:

- **Překrývání tříd:** Pokud se po augmentaci třídy více překrývají, může to znamenat, že augmentace vytvářejí datové vzory, které mohou vytvořit zmatek při klasifikaci. To lze ve vizualizaci lehce poznat.
- **Rozšíření datového prostoru:** Že augmentace správně rozšiřují datový prostor ve vizualizaci zjistíme tím, že existující shluky jaksí "přetíkají" nebo se postupně rozšiřují.
- **Zachování identity tříd:** Správně by augmentační metoda měla zachovávat identitu tříd po augmentaci. Ve vizualizaci by měly augmentovaná data zůstat poblíž původních dat stejné třídy.
- **Nové shluky:** Vznik nových shluků mohou do datového prostoru zanést některé z pokročilých augmentačních metod. Zde je důležité zjistit, zda-li tyto nové shluky smysluplně rozšiřují datový prostor, nebo zda-li vytváří zavádějící a odlehlé hodnoty, které mohou spíše ublížit efektivitě modelu.

### 3 HODNOCENÍ VLIVU AUGMENTACÍ NA DIVERZITU DAT

Tato kapitola se zabývá hodnocením vlivu augmentačních technik na diverzitu dat pro strojové učení a počítačové vidění. Zahrnuje metriky pro kvantifikaci augmentací a popisuje jednotlivé metody.

#### 3.1 Metriky pro kvantifikaci diverzity a rozmanitosti dat

**Metriky posunu distribuce:**

- **Afinita:** Tato metrika určuje míru posunutí distribuce augmentovaných dat od distribuce dat čistých, na kterých se model naučil. To nám poskytuje pohled na to jak se augmentovaná data odchylují od modelu naučeného na čistých datech. Afinita je definována jako rozdíl validačních přesností modelu naučeného na datech čistých a modelu naučeného na datech augmentovaných. Pokud afinita nabývá nulové hodnoty, mezi rozděleními není žádný posun. Záporná hodnota pak značí, že augmentovaná data se nacházejí mimo distribuční oblast očekávanou modelem. [11]

$D_{\text{train}}$  a  $D_{\text{val}}$  označují trénovací a validační datové sady, které jsou nezávislé a identicky rozdělené dle stejné distribuční funkce. Dále  $a$  je stochastická augmentační strategie, přičemž  $D'_{\text{val}}$  je augmentovaná verze validační datové sady u které byla strategie  $a$  aplikována na každý vzorek z datové sady právě jednou:

$$D'_{\text{val}} = \{(a(x), y) \mid (x, y) \in D_{\text{val}}\}$$

Model  $m$  je natrénovaný na trénovací datové sadě  $D_{\text{train}}$ . Dále  $A(m, D)$  značí přesnost modelu  $m$  na datové sadě  $D$ . Samotná metrika afinity  $T[a; m; D_{\text{val}}]$  je potom definována jako rozdíl přesnosti modelu na čistých a augmentovaných datech:

$$T[a; m; D_{\text{val}}] = A(m, D'_{\text{val}}) - A(m, D_{\text{val}})$$

- **Diverzita:** Je metrikou, která kvantifikuje augmentaci z hlediska jak dokáže rozšiřovat trénovací datovou sadu. Diverzita reflektuje jakousi intuici, že komplexní nebo vícefaktorové augmentační strategie zvyšují počet vzorků v trénovací datové sadě, čímž snižují riziko přeučení modelu [11].

$D'_{\text{train}}$  je trénovací datová sada, na kterou byla aplikována stochastická augmentační strategie  $a$ . Dále  $L_{\text{train}}$  značí trénovací ztrátovou funkci modelu  $m$ . Diverzita  $D[a; m; D_{\text{train}}]$  je definována jako očekávaná hodnota trénovací ztráty:

$$D[a; m; D_{\text{train}}] := E_{D'_{\text{train}}} [L_{\text{train}}]$$

- **KL divergence** : Celým názvem Kullbackova-Leiblerova divergence, kvantifikuje míru rozdílnosti mezi dvěma pravděpodobnostními rozděleními. V kontextu augmentace dat pro počítačové vidění se používá na kontrolu změny rozdělení dat po augmentaci. Matematicky se značí jako  $D_{\text{KL}}(P||Q)$  a nabývá hodnot  $D_{\text{KL}}(P||Q) \geq 0$ , přičemž když se KL divergence rovná nule, jsou obě rozdělení shodná.

- **P** je skutečné rozdělení (např. původní datová sada)
- **Q** je předpokládané rozdělení (např. augmentovaná datová sada)

Matematický vzorec:

$$D_{\text{KL}}(P||Q) = \sum_x P(x) \log \left( \frac{P(x)}{Q(x)} \right)$$

### Metriky diverzity a pokrytí prostoru příznaků:

- **Rozptylová diverzita**: Anglicky "Variance diversity" je metrikou kvantifikující diverzitu augmentovaných dat. Dle zdroje [12] bylo prokázáno experimentem, že existuje korelace mezi touto metrikou a lepší generalizací modelu. Nízká hodnota rozptylové diverzity znamená, že augmentovaná data jsou hodně podobná původním datům, tedy mají nízkou diverzitu. Pokud je hodnota vysoká, znamená to, že data jsou dostatečně různorodá a mají potenciál vylepšit generalizaci modelu.

Pro daný vstup  $x_i$  je definována jako průměrný rozptyl embeddingů jednotlivých augmentací  $a_j(x_i)$  vzhledem k jejich centroidu (průměrná reprezentace všech augmentací v prostoru příznaků):

$$VD(x_i) = \frac{1}{k-1} \sum_{j=1}^k |f_{\theta}(a_j(x_i)) - \mu_i|_2^2$$

Kde  $f_{\theta}(\cdot)$  je model trénovaný na neaugmentovaných datech extrahující příznaky a  $\mu_i$  je centroid.



- **Pokrytí prostoru příznaků:** Tato metrika, kvantifikuje jak dobře nová augmentovaná data zaplňují prostor příznaků. Pokrytí prostoru příznaků lze měřit různými způsoby. Jednou z variant je výpočet objemu konvexního obalu redukovaného prostoru příznaků, například po PCA transformaci, tím získáme informaci o tom, jak velkou část prostoru augmentovaná data pokrývají. Čím je větší objem, tím se zvětšuje i diverzita a pokrytí prostoru příznaků.

$X$  je množina všech bodů v prostoru příznaků po redukci dimenze pomocí PCA algoritmu. Konvexní obal  $\text{Conv}(X)$  je poté množina všech konvexních kombinací těchto bodů:

$$\text{Conv}(X) = \left\{ \sum_{i=1}^n \lambda_i x_i \mid \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1 \right\}$$

Samotný objem konvexního obalu  $V$  je následovně:

$$V = \text{Vol}_d(\text{Conv}(X))$$

Kde  $\text{Vol}_d$  znázorňuje objem o dimenzi  $d$ , typicky pro dvě dimenze je to plocha a pro tři se jedná o objem.

- **Skóre posunu domény:** V oblasti počítačové vidění výkon modelu může dramaticky klesat, při střetnutí s reálnými daty, která jsou statisticky odlišná od dat, na kterých byl natrénován. Tento problém je znám jako problém posunu domény. Tato metrika kvantifikuje míru změny mezi původní a cílovou doménou. Může být měřena jako statistická vzdálenost mezi distribucemi původních a augmentovaných dat. Velmi vysoký posun může vést k špatné reprezentaci augmentovaných dat vůči těm původním, zatímco příliš malý posun může značit, že augmentace je neefektivní [13].

### Metriky shlukování a homogenity dat

- **Čistota shluku:** Anglicky "cluster purity" je metrikou hodnotící, jak moc jsou shluky dat homogenní po aplikaci augmentačních metod. V oblasti augmentačních metod je totiž velice důležité, aby augmentovaná data náležely do stejné sémantické kategorie jako jejich původní neaugmentované verze. Díky této metrice je možné identifikovat případy, kdy augmentace vytváří vzorky překračující hranice samotných tříd a narušují

tak původní strukturu dat. Vysoká hodnota této metriky může značit, že augmentační metody zachovávají sémantickou integritu dat, zatímco malé hodnoty mohou naznačovat, že augmentace není vhodná.

- **Siluetové skóre:** Tato metrika analyzuje shluky augmentovaných dat a hodnotí jak dobře jsou nově augmentované vzorky seskupeny. Tato metrika je počítána pro každý vzorek a jedná se zejména o její kohezi a separaci. Koheze určuje míru podobnosti vzorku vůči svému vlastnímu shluku. Separace poté uvádí míru rozdílnosti vzorku vůči ostatním shlukům. Tím nám tato metrika může naznačit, jestli augmentační metody zachovávají nebo naopak narušují původní struktury v datech. Toto skóre může nabývat hodnot od -1 do 1, přičemž čím vyšší hodnota značí, že shluky jsou dobře definované, to může pomoci modelu lépe rozlišovat samotné třídy.

### 3.2 Analýza vlivu jednotlivých augmentačních technik na distribuci dat

Tato podkapitola se zaměří na to, jak augmentace ovlivňují diverzitu obrazových dat. Pochoopení těchto vlivů je klíčové pro správné vybrání vhodné augmentační strategie pro konkrétní scénáře v oblasti počítačového vidění, což je i dalším bodem této práce. Chápání těchto metrik je také důležité pro správnou interpretaci jejich dopadů na trénovací proces a následný výkon modelu počítačového vidění.

Augmentační metody jsou transformacemi nad původními datovými sadami s účelem vytvoření obohaceného a rozmanitějšího trénovacího souboru, to lze kvantifikovat pomocí různých metrik a vizualizovat pomocí algoritmů jako je t-SNE, což nám umožní nahlédnout na změnu distribuce dat, jak se nově augmentovaná data rozprostírají dimenzemi a jak se například mění jejich shlukování oproti původním datům. Augmentační metody také dokáží modifikovat strukturu a rozmanitost původních dat, což může mít vážný dopad na výkon modelu a je potřeba zhodnotit, zda je tento dopad přínosný nebo ne.

- **Geometrické transformace (rotace, škálování, ořez, zrcadlení)** zachovávají sémantický obsah a charakteristiku obrazu, ale mění jeho prostorovou orientaci, tedy rozšiřují datový prostor o další možné variace a pomáhají modelu se obrnit vůči těmto změnám v reálných datech.
- **Intenzivní transformace (změna jasu, kontrastu, šum)** mění vizuální vlastnosti ob-

razu jako takového, to může pomoci zvětšit robustnost modelu vůči změnám v osvětlení a šumu v reálných datech.

- **Pokročilé metody (Cutout, Mixup, GAN-based)** vytváří zcela nové umělé vzory, ty mohou výrazně rozšířit původní distribuci dat mimo jejich původní vzory a shluky, model poté lépe dokáže generalizovat. Ovšem tyto metody také mohou zavést artefakty do augmentovaných dat, proto je velmi důležité tyto metody správně vyhodnotit.

### 3.2.1 Kvantifikace dopadu augmentací na distribuci dat

Pro analýzu vlivu augmentačních metod na distribuci dat se dají použít tyto metriky:

- **Statistické metriky rozptylu a hustoty dat:** tyto metriky kvantifikují rozptyl dat v různých dimenzích po augmentaci.
- **Metriky divergence distribucí:** kvantifikují rozdíly mezi distribucí původní a augmentované datové sady.
- **Metriky vzdáleností v latentním prostoru:** tyto metriky odrážejí, jak se augmentační metody projevují v prostoru reprezentací modelu, jde například o průměrnou vzdálenost mezi body.

Z výsledků těchto analýz lze vidět, že různé augmentační metody mají různé dopady na rozmanitost a distribuci dat. Jednoduché geometrické augmentace často vedou k mírnému rozšíření distribuce, naopak metody pokročilé, které dokáží vytvořit zcela nové vzorky dokáží distribuci výrazně rozšířit, ale s rizikem zavedení artefaktů a vzorů neodpovídajících těm reálným.

Pro správné nastavení tréninkového procesu modelu je klíčové pochopit, jak augmentace mění distribuci dat. Přílišné augmentace, které ostře rozšiřují distribuci dat, mohou vést k učení nereálných vzorů. Naopak nedostatečné augmentace nemusí dostatečně pokrýt variabilitu reálných dat a model se může lehce přeučit. Je tedy důležité vyvážit kvalitu a diverzitu augmentovaných dat i s ohledem na danou úlohu a datovou sadu.

### 3.3 Komparativní hodnocení efektivity různých přístupů k augmentaci

Komparativní hodnocení efektivity augmentačních metod je klíčové pro výběr optimální strategie pro danou úlohu. Augmentační metody se liší v principech jak rozšiřují distribuci

dat, v dopadu na výkon modelu, robustnosti vůči variabilitě v reálných datech a schopnosti generalizace.

Kritéria hodnocení efektivity augmentací:

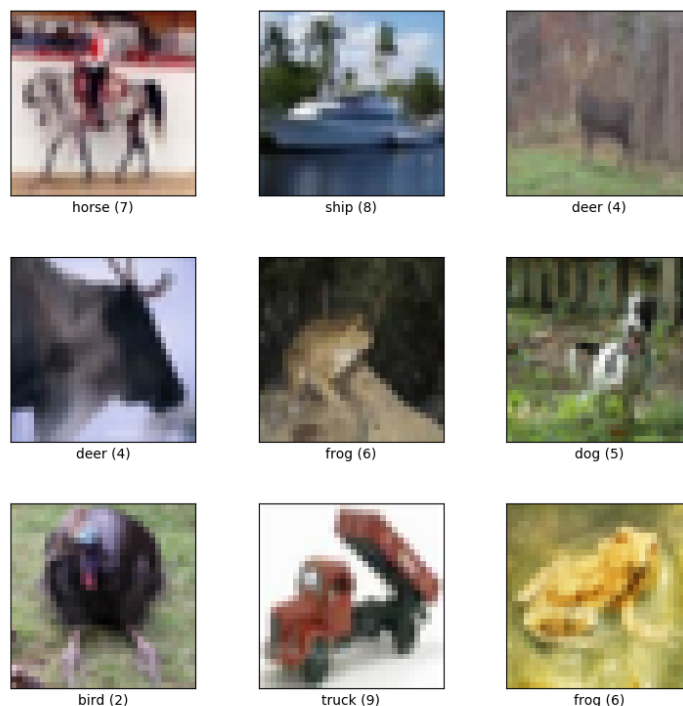
- Robustnost modelu vůči rušivým vlivům jako je například šum, deformace, variace v osvětlení.
- Výpočetní náročnost a složitost na implementaci jednotlivých augmentačních metod.
- Diverzita augmentovaných dat, která ovlivňuje jak se model dokáže naučit variabilitu reálných dat.
- Dopad na konvergenci tréninku, tedy jak rychle a stabilně model dokáže dosáhnout optimálních parametrů.
- Zlepšení přesnosti modelu na validačních a testovacích datových sadách, to dokáže odhlait přínos augmentační metody pro schopnost modelu lépe generalizovat.

## **II PRAKTICKÁ ČÁST**

## 4 IMPLEMENTACE A EXPERIMENTÁLNÍ POSTUP

### 4.1 Popis datového souboru

Pro účely této práce byl vybrán datový soubor CIFAR-10. Tento datový soubor je standardizovaný dataset často využívaný v oblasti strojového učení a počítačového vidění pro výzkum. CIFAR-10 obsahuje v celku 60 000 obrazových vzorků s rozlišením 32x32 pixelů. Ty jsou rozděleny do 10 sémantických tříd a každá třída je reprezentována přesně 6000 vzorky. Dále je datový soubor rozdělen na trénovací a testovací sadu v poměru 5:1, tedy 50 000 obrazových vzorků pro trénování a 10 000 vzorků pro testování. Obrazové vzorky jsou uloženy ve formátu trojrozměrných tenzorů odpovídající kanálům rgb tedy 32x32x3, kde každý pixel má hodnotu v rozsahu 0-255 reprezentovaný 8 bity [14].



Obrázek 4.1 Ukázka datasetu CIFAR-10

(zdroj: 15)

Sémantické třídy datového souboru CIFAR-10 jsou následující:

- Letadla

- Automobily
- Ptáci
- Kočky
- Jeleni
- Psi
- Žáby
- Koně
- Lodě
- Nákladní automobily

Výběr datové sady CIFAR-10 jako základu pro experimenty vychází z několika jeho vlastností. Jednak jde o nízké rozlišení obrazových vzorků díky němuž je trénovací proces rychlý a umožňuje rychlejší iteraci při testování. Dále jeho skromnější komplexita co se množství tříd týče poskytuje rozumný testovací prostor pro přehlednou vizualizaci a vyhodnocení dopadu augmentací. Navíc jeho využití v jiných studiích poskytuje přímé srovnání s již existujícími metodami či výzkumy.

## 4.2 Výběr augmentací

Pro tuto práci byly vybrány augmentační metody, které pokrývají několik kategorií augmentačních technik. Výběr je založen na schopnostech rozšíření distribuce datové sady z hlediska geometrických změn, fotometrických transformací, simulace šumu a kombinace obrazových vzorů. Tabulka níže uvádí jednotlivé augmentace, kategorii do které patří, parametry dané augmentace a její očekávaný vliv na data.

- **Geometrické transformace** byly zvoleny pro rozšíření prostorové variability datových vzorků, s účelem pomoci modelu naučit se odchylky orientací a pozic objektů v obrazových datech.

Tabulka 4.1 Tabulka použitých augmentací

Exp.	Název augmentace	Kategorie	Parametry	Očekávaný vliv na data
1	Rotace	Geometrická	$\pm 30^\circ$	Změna orientace, větší variabilita pozice objektů
2	Horizontální převrácení	Geometrická	$p = 0.5$	Otočení horizontálně, mírná diverzita bez ztráty informace
3	Náhodné ořezání + Změna velikosti	Geometrická	Měřítko: (0.8, 1.0)	Náhodný výřez a změna velikosti, modifikace pozice a části objektu
4	Změna barevnosti	Fotometrická	Jas, kontrast, saturace, odstín (0.4)	Změna barevnosti, rozšíření barevné variability
5	Stupnice šedi	Fotometrická	Převod na odstíny šedi	Ztráta barevné informace, testování odolnosti na barevné změny
6	Gaussův šum	Šum	mean = 0, std = 0.1	Přidání zrnění, simulace reálného šumu v obrazech
7	Náhodné mazání	Šum/Okluze	Měřítko = (0.02, 0.33)	Náhodné vymazání částí obrazu, simulace výpadků a překryvů

(zdroj: vlastní)

- **Fotometrické augmentace** by měli pomoci modelu lépe rozpoznávat objekty nebo scény při změně osvětlení, barevnosti vstupních dat nebo i při absenci informací o barvě, což odráží reálné scénáře při sběru dat.
- **Augmentace založené na šumu** napodobují šum zachycený na reálných obrazových vzorcích, zvyšující robustnost modelu vůči nekvalitním vstupním datům.

Výběr těchto augmentací umožňuje analýzu jejich vlivu na diverzitu datového souboru CIFAR-10, poskytuje také podklad pro t-SNE vizualizaci prostoru příznaků. Analýzou jednotlivých augmentačních metod poté budeme sledovat, jak se mění rozložení dat v prostoru příznaků. Výpočtem a analýzou různých metrik budeme uvažovat, jak mohou přispět k lepší generalizaci, robustnosti a výkonu modelu.

### 4.3 Implementace augmentací

Augmentace jsou definovány v Python slovníku, kde klíčem je název augmentace a číslo experimentu, hodnotou je pak samotná augmentace. Většina augmentací jsou transformační operace z knihovny `torchvision.transforms` a u gaussova šumu se jedná o vlastně implementovanou třídu.



```
1 AUGMENTATIONS = {
2     "1_Rotation": transforms.RandomRotation(30),
3     "2_HFlip": transforms.RandomHorizontalFlip(p=0.5),
4     "3_CropResize": transforms.RandomResizedCrop(size
5         =(224,224), scale=(0.8, 1.0)),
6     "4_ColorJitter": transforms.ColorJitter(brightness=0.4,
7         contrast=0.4, saturation=0.4, hue=0.1),
8     "5_Grayscale": transforms.Grayscale(num_output_channels
9         =3),
10    "6_GaussianNoise": GaussianNoise(std=0.1),
11    "7_RandomErasing": transforms.RandomErasing(),
12 }
```

---

#### Výpis 4.1 Implementace augmentací

Gaussův šum je implementován pomocí třídy `GaussianNoise`, která přijímá dva parametry a funguje jako standardní transformační operace. Šum lze řídit pomocí dvou parametrů: `mean`, který ovládá posun šumu a `std`, který určuje intenzitu šumu. Návrátová hodnota je tensor stejného tvaru jako ten vstupní, ovšem s přidávanými náhodnými hodnotami s gaussovým rozdělením se zadaným středem `mean` a směrodatnou odchylkou `std`.

```
1 class GaussianNoise:
2     def __init__(self, mean=0., std=0.1):
3         self.mean = mean
4         self.std = std
5     def __call__(self, tensor):
6         return tensor + torch.randn_like(tensor) * self.std +
7             self.mean
```

---

#### Výpis 4.2 Implementace gaussova šumu

Augmentace jsou poté do celé pipeline vkládány pomocí funkce `get_transform(aug_key)`, která následně sestaví sekvenci transformací pro trénovací data. Každý obraz je převeden na tensor a následně normalizován, poté se přidá daná augmentace na začátek seznamu transformací. Funkce navrácí objekt, který představuje sekvenci transformačních operací.

```
1 def get_transform(aug_key=None):
2     base = [
3         transforms.ToTensor(),
4         transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
5     ]
6     if aug_key in AUGMENTATIONS:
7         base.insert(0, AUGMENTATIONS[aug_key])
8     return transforms.Compose(base)
```

---

#### Výpis 4.3 Funkce get\_transform

#### 4.4 Výběr metrik pro hodnocení augmentací

Pro hodnocení vlivu augmentačních metod na diverzitu dat, pokrytí prostoru příznaků, separaci shluků a posunu domény byly vybrány metriky popsané v tabulce 4.2 níže.

Tabulka 4.2 Tabulka vybraných metrik

Metrika	Rozsah hodnot	Význam nízké hodnoty	Význam vysoké hodnoty
Variance Diversity	$\geq 0.0$	Nízká variabilita, data jsou homogenní	Vysoká rozmanitost, data pokrývají širší spektrum
Affinity Index	0.0 - 1.0	Nízká diverzita nebo přílišná odlišnost od původních dat	Vysoká podobnost k originálním datům, zachování klíčových rysů
Diversity Index	0.0 - 1.0	Nízká variabilita, data jsou homogenní	Vysoká rozmanitost, data pokrývají širší spektrum variací
Feature Space Coverage	$\geq 0.0$	Špatné pokrytí prostoru příznaků, omezená variabilita	Dobré pokrytí prostoru příznaků, široká reprezentace dat
Domain Shift Score	$\geq 0.0$	Malý posun domény, data jsou podobná originálu	Velký posun, riziko ztráty relevance dat
Cluster Purity	0.0 - 1.0	Špatné shlukování, třídy se překrývají	Dobré shlukování, třídy jsou jasně oddělené
KL Divergence	$\geq 0.0$	Malý rozdíl mezi distribucemi, dobrá věrnost vůči původní distribuci	Velký rozdíl, augmentace mění distribuci výrazně
Silhouette Score	-1.0 - 0.0	Špatná separace a kompaktnost shluků	Dobrá separace a kompaktnost shluků

(zdroj: vlastní)

Tyto vybrané metriky poskytnou ucelený pohled na kvalitu augmentací z hlediska variability, pokrytí prostoru příznaků, udržení konzistence vůči původní datové sadě, zachování struktury a posunu domény. To umožní kvantitativní hodnocení efektivity jednotlivých augmentačních metod a jejich vhodnost pro konkrétní scénáře.

#### 4.5 Implementace vybraných metrik

Tato podkapitola se zaměří na popis jednotlivých implementací metrik vybraných pro tuto práci.

##### 4.5.1 Variance Diversity

Funkce `variance_diversity` vyhodnotí rozptyl pravděpodobnostních výstupů modelu na datech. Pro každý obrazový vzorek se vypočítá pravděpodobnosti tříd pomocí funkce `nn`.

functional . softmax. Tyto pravděpodobnosti jsou uloženy do pole a následně se z nich spočítá průměrná variance přes všechny třídy.

---

```
1 def variance_diversity(model, dataloader):
2     model.eval()
3     probs = []
4     with torch.no_grad():
5         for images, _ in dataloader:
6             images = images.to(DEVICE)
7             outputs = model(images)
8             prob = nn.functional.softmax(outputs, dim=1)
9             probs.append(prob.cpu().numpy())
10    probs = np.concatenate(probs, axis=0)
11    var_div = np.var(probs, axis=0).mean()
12    return var_div
```

---

#### Výpis 4.4 Implementace variance diversity

### 4.5.2 Affinity, Diversity Index

Funkce `affinity_diversity_index` vypočítává dvě metriky naráz. Metriku affinity, která měří jak moc jsou si augmentovaná data podobná datům původním z hlediska klasifikace a metriku diversity, která měří variabilitu augmentovaných dat. Augmentovaná data se postupně procházejí a počítají se predikce. Jednak se spočítá přesnost predikce vůči skutečnému štítku pro affinity a poté se vypočítá rozdíl průměrných ztrát augmentovaných a původních dat pro index diversity. Funkce navrácí obě tyto hodnoty.

---

```
1 def affinity_diversity_index(base_model, augmented_loader,
2     aug_losses, base_losses):
3     base_model.eval()
4     preds = []
5     labels = []
6     with torch.no_grad():
7         for images, lbls in augmented_loader:
8             images = images.to(DEVICE)
9             outputs = base_model(images)
10            preds.append(outputs.argmax(dim=1).cpu().numpy())
11            labels.append(lbls.numpy())
12    preds = np.concatenate(preds)
13    labels = np.concatenate(labels)
14    affinity = accuracy_score(labels, preds)
15    diversity = np.mean(aug_losses) - np.mean(base_losses)
16    return affinity, diversity
```

---

#### Výpis 4.5 Implementace affinity a diversity indexu

### 4.5.3 Feature Space Coverage

Výpočet této metriky je realizován do dvou fází kdy prvně se použije PCA (principiální analýza komponentů) pro redukci dimenzionality dat a poté se spočítá konvexní obal, který kvantifikuje míru pokrytí prostoru příznaků. Dimenzionalitu zde redukuje, protože výpočet konvexního obalu pro více jak tři dimenze je nestabilní a výpočetně náročný. Nejprve se zkombinují data původních a augmentovaných příznaků a projdou redukcí pomocí PCA do 2D/3D prostoru. Výpočet konvexního obalu je následně vypočítán pomocí třídy ConvexHull z knihovny `scipy.spatial`.

---

```

1 def feature_space_coverage(original_features,
2   augmented_features, n_components=2):
3     if len(original_features.shape) > 2:
4         original_features = original_features.reshape(
5             original_features.shape[0], -1)
6         augmented_features = augmented_features.reshape(
7             augmented_features.shape[0], -1)
8     combined = np.concatenate([original_features,
9                               augmented_features], axis=0)
10    n_components = min(n_components, combined.shape[0] - 1,
11                      combined.shape[1])
12    if n_components > 3:
13        n_components = 3
14    pca = PCA(n_components=n_components)
15    reduced = pca.fit_transform(combined)
16    hull = ConvexHull(reduced)
17    return hull.volume

```

---

Výpis 4.6 Implementace feature space coverage

### 4.5.4 Domain Shift Score

---

```

1 def train_domain_classifier(original_loader, augmented_loader
2   ):
3     # Jednoduchý CNN na rozlišení originálních vs augmentovan
4     # ých dat
5     class DomainClassifier(nn.Module):
6         def __init__(self):
7             super().__init__()
8             self.net = nn.Sequential(
9                 nn.Flatten(),
10                nn.Linear(3*32*32, 256),
11                nn.ReLU(),
12                nn.Linear(256, 2)
13            )
14        def forward(self, x):
15            return self.net(x)
16    model = DomainClassifier().to(DEVICE)
17    criterion = nn.CrossEntropyLoss()
18    optimizer = optim.Adam(model.parameters(), lr=1e-3)
19    # Připravit dataset s labely 0 = originál, 1 = augmentace
20    def combined_loader():

```

```

21         for (x1, _), (x2, _) in zip(original_loader,
22                                     augmented_loader):
23             yield x1, torch.zeros(x1.size(0), dtype=torch.
24                                   long)
25             yield x2, torch.ones(x2.size(0), dtype=torch.long)
26     model.train()
27     for epoch in range(3):
28         for images, labels in combined_loader():
29             images, labels = images.to(DEVICE), labels.to(
30                 DEVICE)
31             optimizer.zero_grad()
32             outputs = model(images)
33             loss = criterion(outputs, labels)
34             loss.backward()
35             optimizer.step()
36     # Vyhodnocení accuracy (domain shift score)
37     model.eval()
38     all_preds = []
39     all_labels = []
40     with torch.no_grad():
41         for images, labels in combined_loader():
42             images, labels = images.to(DEVICE), labels.to(
43                 DEVICE)
44             outputs = model(images)
45             preds = outputs.argmax(dim=1)
46             all_preds.append(preds.cpu().numpy())
47             all_labels.append(labels.cpu().numpy())
48     all_preds = np.concatenate(all_preds)
49     all_labels = np.concatenate(all_labels)
50     acc = accuracy_score(all_labels, all_preds)
51     return acc

```

#### Výpis 4.7 Implementace domain shift score

### 4.5.5 Cluster Purity

Výpočet této metriky se skládá ze tří kroků:

- **Shlukování** pomocí algoritmu K-Means do 10 shluků. Výstupem je, že každý vzorek je přiřazen do jednoho ze shluků.
- **Kontingenční matice** je vytvořena pomocí funkce `confusion_matrix`, která porovná skutečné štítky vůči těm přiřazeným ke shlukům. Řádky odpovídají skutečným třídám a sloupce shlukům.
- **Výpočet purity** se děje tak, že pro každý shluk se najde maximální hodnota, ta představuje dominantní třídu daného shluku. Tyto hodnoty jsou sečteny napříč všemi shluky a jejich suma je dělena celkovým počtem všech vzorků a tím je získána hodnota purity, která se pohybuje v rozmezí 0.0 až 1.0.

---

```

1  def cluster_purity(features, true_labels, n_clusters=10):
2      kmeans = KMeans(n_clusters=n_clusters, random_state
3                      =42).fit(features)
4      cm = confusion_matrix(true_labels, kmeans.labels_)
5      purity = np.sum(np.max(cm, axis=0)) / np.sum(cm)
        return purity

```

---

Výpis 4.8 Implementace cluster purity

#### 4.5.6 KL Divergence

Funkce `kullback_leibler_divergence` kvantifikuje rozdíl mezi dvěma diskrétními pravděpodobnostními rozděleními  $p$  a  $q$ . Nejprve se obě rozdělení normalizují na pravděpodobnostní rozdělení, kde součet všech prvků v daném rozdělení je roven 1. Poté funkce `distance.rel_entr` vypočítá relativní entropii normalizovaných rozdělení a poté se nad ním provede suma což odpovídá KL divergenci. Ke každému rozdělení se přičítá malá hodnota  $1 \times 10^{-10}$ , která zabraňuje případu, kdyby se dělilo nulou.

---

```

1  def kullback_leibler_divergence(p, q):
2      p_norm = p / np.sum(p)
3      q_norm = q / np.sum(q)
4      return distance.rel_entr(p_norm + 1e-10, q_norm + 1e-10).
        sum()

```

---

Výpis 4.9 Implementace KL divergence

#### 4.5.7 Silhouette Score

Implementace siluetového skóre probíhá pomocí využití funkce `silhouette_score` z knihovny `sklearn.metrics`. V ukázce níže je příklad výpočtu této metriky kdy se formát příznaků přetvaruje do 2D matice s počtem vzorků a počtem příznaků. Funkce poté porovnává průměrné vzdálenosti bodů vůči ostatním ve stejném shluku. Návrátová hodnota se pohybuje mezi -1.0 a 1.0.

---

```

1  sil_score = silhouette_score(all_features.reshape(
        all_features.shape[0], -1), all_labels)

```

---

Výpis 4.10 Implementace silhouette score

## **ZÁVĚR**

Text ...





## SEZNAM POUŽITÉ LITERATURY

- [1] XU, Mingle; YOON, Sook; FUENTES, Alvaro a PARK, Dong Sun, 2023. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*. 2023. 137, s. 109347. ISSN 0031-3203. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0031320323000481>.
- [2] SHORTEN, Connor a KHOSHGOFTAAR, Taghi M, 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*. 2019.
- [3] MILUVS, 2025. What is the impact of brightness adjustment in data augmentation? Dostupné z: <https://milvus.io/ai-quick-reference/what-is-the-impact-of-brightness-adjustment-in-data-augmentation>.
- [4] UNIVERSITY, Northwestern, 2025. Analyzing and Enhancing Images (Image Processing Toolbox). Dostupné z: <http://www.ece.northwestern.edu/local-apps/matlabhelp/toolbox/images/enhanc12.html>.
- [5] DEVRIES, Terrance a TAYLOR, Graham W., 2017. Improved Regularization of Convolutional Neural Networks with Cutout. Dostupné z: <https://arxiv.org/abs/1708.04552>.
- [6] MILVUS, 2025. How does cutout augmentation work? Dostupné z: <https://blog.milvus.io/ai-quick-reference/how-does-cutout-augmentation-work>. [Accessed 08-05-2025].
- [7] CHOI, Juhwan a KIM, YoungBin, 2024. Colorful Cutout: Enhancing Image Data Augmentation with Curriculum Learning. Dostupné z: <https://arxiv.org/abs/2403.20012>.
- [8] GOODFELLOW, Ian J.; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDEFARLEY, David et al., 2014. Generative Adversarial Networks. Dostupné z: <https://arxiv.org/abs/1406.2661>.
- [9] GARCEA, Fabio; SERRA, Alessio; LAMBERTI, Fabrizio a MORRA, Lia, 2023. Data augmentation for medical imaging: A systematic literature review. *Computers in Bi-*

- ology and Medicine*. 2023. 152, s. 106391. ISSN 0010-4825. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S001048252201099X>.
- [10] MAATEN, Laurens a HINTON, Geoffrey E., 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*. 2008. 9, s. 2579–2605. Dostupné z: <https://api.semanticscholar.org/CorpusID:5855042>.
- [11] LOPES, Raphael Gontijo; SMULLIN, Sylvia J.; CUBUK, Ekin D. a DYER, Ethan, 2020. Affinity and Diversity: Quantifying Mechanisms of Data Augmentation. *CoRR*. 2020. abs/2002.08973. Dostupné z: <https://arxiv.org/abs/2002.08973>.
- [12] LIU, Zirui; JIN, Haifeng; WANG, Ting-Hsiang; ZHOU, Kaixiong a HU, Xia, 2021. Di-vAug: Plug-in Automated Data Augmentation with Explicit Diversity Maximization. Dostupné z: <https://arxiv.org/abs/2103.14545>.
- [13] STATLECT, 2025. Domain shift — statlect.com. Dostupné z: <https://www.statlect.com/machine-learning/domain-shift>. [Accessed 06-05-2025].
- [14] TENSORFLOW, 2025. cifar10 - TensorFlow Datasets — tensorflow.org. <https://www.tensorflow.org/datasets/catalog/cifar10>. [Accessed 19-05-2025].
- [15] KRIZHEVSKY, Alex, 2009. Learning multiple layers of features from tiny images.
- [16] AYYADEVARA, V. Kishore a REDDY, Yeshwanth, 2020. *Modern computer vision with PyTorch*. Packt. ISBN 978-1-83921-347-2.
- [17] Fleet, Pajdla, Schiele, Tuytelaars]computer-vision1 FLEET, David; PAJDLA, Tomáš; SCHIELE, Bernt a TUYTELAARS, Tinne, [2014]. *Computer vision - ECCV 2014*. Springer. Dostupné z: <https://doi.org/10.1007/978-3-319-10602-1>.
- [18] DYK, David A. a MENG, Xiao-Li, 2001. The Art of Data Augmentation. *Journal of Computational and Graphical Statistics*. 2001. 10, č. 1, s. 1–50. ISSN 10618600. Dostupné z: <http://www.jstor.org/stable/1391021>.
- [19] SHAPIRO, Linda G. a STOCKMAN, George C., c2001. *Computer vision*. Prentice Hall. ISBN 0-13-030796-3.
- [20] SHORTEN, Connor a KHOSHGOFTAAR, Taghi M, 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*. 2019. 6, č. 1, s. 1–48.

**SEZNAM OBRÁZKŮ**

Obrázek 4.1	Ukázka datasetu CIFAR-10 .....	30
-------------	--------------------------------	----

**SEZNAM TABULEK**

Tabulka 4.1	Tabulka použitých augmentací.....	32
Tabulka 4.2	Tabulka vybraných metrik .....	34

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GAN      Generative Adversarial Network

**SEZNAM PŘÍLOH**

PŘÍLOHA P I: První příloha

PŘÍLOHA P II: Druhá příloha

# **PŘÍLOHA P I: PRVNÍ PŘÍLOHA**

Obsah přílohy

## **PŘÍLOHA P II: DRUHÁ PŘÍLOHA**

Obsah přílohy