VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

MPC-STU

SROVNÁNÍ KNIHOVEN PRO ROZŠÍŘENÍ

OBRAZOVÉHO DATASETU

Pavel Dušek, Jiří Pražák, Stanislav Svědiroh,

Václav Večeřa, Václav Zmítko

Brno, 2021

Obsah

[Zadání 1](file:///D:\Download\Projekt_dokumentace.docx#_Toc68378635)

Seznam obrázků

**Nenalezena položka seznamu obrázků.**

# Zadání

Datasety jsou alfou a omegou strojového učení. Pro metody hlubokého učení je vhodné mít datasety dostatečně obsáhlé. Bohužel stává se, že vzorových dat je omezené množství. V tomto případě přichází na řadu metody rozšíření datasetu. Cíl projektu je nastudovat, zdokumentovat a vytvořit srovnání minimálně 5 knihoven vybraných po domluvě s vedoucím projektu pro rozšíření datasetu v jazyce Python. Součástí řešení je vytvoření časového srovnání jednotlivých implementací.

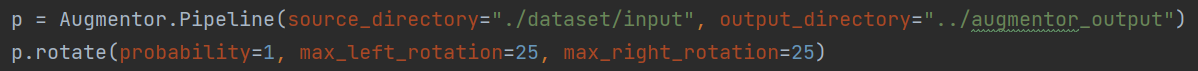
# Knihovny pro augmentaci dat

Augmentace dat je operace pro rozšíření datasetu, které jsou využity pro trénování modelů strojového učení. Důvodem jejího využití je zabránění overfitingu, což znamená případu, kdy je neuronová síť trénována na datech, která nejsou dostatečně rozmanitá. Existuje několik možností pro augmentaci dat, těmito možnostmi jsou různé translace, rotace, změny škálování, zašumění, překlápění obrazu, změna jasu, kontrastu atd… Bylo zadáno pět knihoven pro rozšíření datasetu (augmentaci dat). Těmito knihovnami jsou *Augmentor, Albumentations, Imgaug, PyTorch a AutoAugment.* Tyto knihovny jsou balíčkem v Pythonu navržených pro podporu rozšíření a umělého generování obrazových dat pro úlohy strojového učení.

## Augmentor

Tato knihovna nabízí augmentaci, a to několika způsoby, především zkosení (Perspective Skewing), elastické zkreslení (Elastic Distortions), rotaci (Rotating), roztažení (Shearing), oříznutí (Cropping), a zrcadlení (Mirroring). Do argumentu každé funkce v následujících podkapitolách je zadávána pravděpodobnost (Probability) v mezích 0 – 1 podle klasické pravděpodobnosti.

Práce s knihovnou probíhá následovně, je vytvořen objekt třídy Pipeline, kterému je v parametru zadána cesta k vstupním datům a cesta pro ukládání vygenerovaných dat. Dále je možné tomu objektu přidávat operace, viz podkapitoly níže.



Poté je nutné objektu *p* zadat počet dat, která se mají vygenerovat:

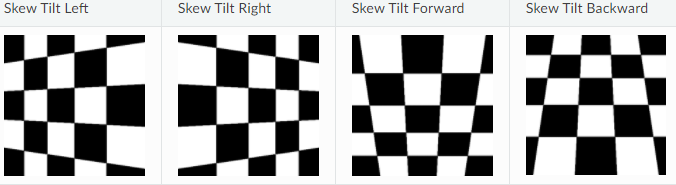


Pokud není požadováno uložení rozšířeného datasetu na disk, pak je možné tato data přímo importovat do PyTorch.

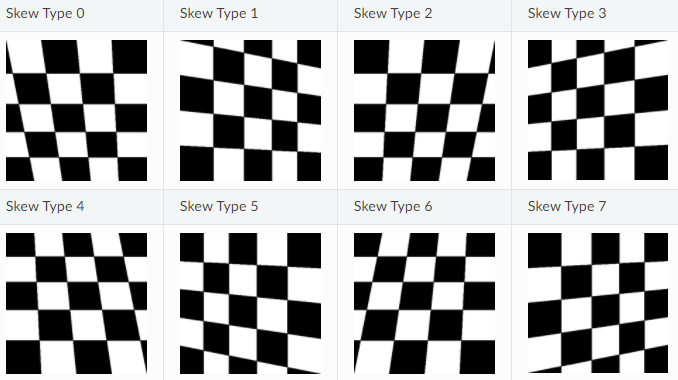
### Zkosení (Perspective Skewing)

Tento balíček funkcí zajišťuje transformaci vstupních dat tak, jako by na ně kamera nahlížela pod jiným úhlem. Všechny níže uvedené funkce mají parametr *magnitude*, kterým je *nastavováno maximální možné zkosení (v rozmezí 0.1 – 1.0).*:

* skew\_tilt() - podporované argumenty (left, right, forward, backward)



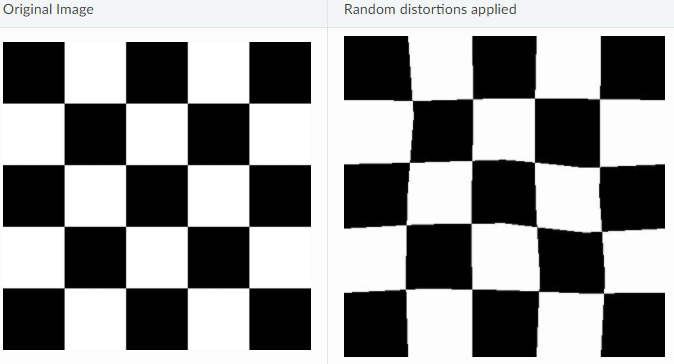
* skew\_left\_right()
* skew\_top\_bottom()
* skew\_corner() - výběr 8 možných bodů zkosení



* skew() -zkosí obrázek náhodně podle jednoho z 12 směrů zobrazených výše.

### Elastické zkreslení (Elastic Distortions)

Tento balíček funckí slouží k provedení elastického zkreslení obrazu, avšak je zachován poměr stran, jak lze vidět na obrázku níže.



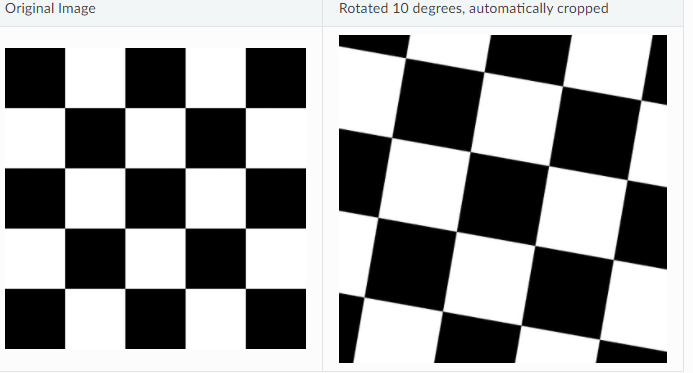
K tomuto zkreslení slouží funkce:

* *random\_distortion()* - jsou zadávány parametry *grid\_width, grid\_height, magnitude*. První dva parametry mohou nabývat hodnot 2 až 10 a určují velikost mřížky. Hodnota parametru *magnitude* by měla být v rozmezí 1 až 10.

### Rotace (Rotating)

Tento balíček funkcí slouží k augmentaci pomocí rotace, k tomu slouží funkce:

* *rotate()* - do argumentu této funkce je možné zadat parametry *max\_left\_rotation* a *max\_right\_rotation*, v těchto mezích dochází k náhodné rotaci. Pokud není obrázek rotován v násobcích 90°, pak jsou výsledná data oříznuta a je kladen důraz na co možná největší oříznutí, za účelem dodržení původního poměru stran, což vede k přiblížení objektu. Pokud by byla rotace provedena klasicky, aby byl vidět celý rotovaný objekt, pak vznikne kolem objektu pole, což může být nežádoucí. Tato problematika je vyobrazena na obrázku níže.

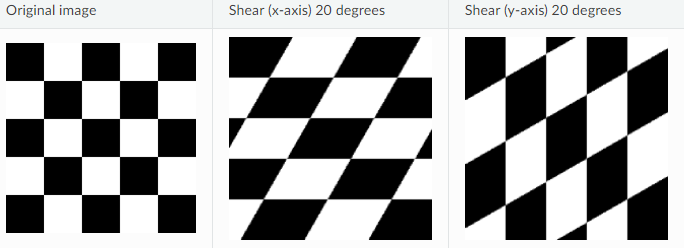


* *rotate90()* - rotace o 90°
* *rotate180()* - rotace o 180°
* *rotate270()* - rotace o 270°
* *rotate\_random\_90()* - funkce otočí objekt o náhodný násobek 90°

### Roztažení (Shearing)

Modifikuje objekt roztažením ve směru x nebo y. Opět tato knihovna klade důraz na zachování poměru stran, a proto se roztažený obraz ořízne (přiblíží). Jak lze vidět na obrázku níže. K této modifikaci slouží funkce:

* shear() - do argumentu této funkce je možné zadat parametry max\_shear\_left a max\_shear\_right, v těchto mezích dochází k náhodnému roztažení, přičemž není možno zadat hodnotu meze větší než 25°.



### Oříznutí (Cropping)

Oříznutí probíhá jako přiblížení určité náhodné části objektu, opět je zachován poměr stran, což je velká výhoda pro následující postprocessing. Oříznutí zařizují následující funkce:

* crop\_centre() - do argumentů této funkce je nutné zadat parametry percentage\_area, randomise\_percentage\_area. Pokud je druhá zmiňovaný parametr false, pak první parametr udává absolutní hodnotu oříznutí, v opačném případě udává rozmezí mezi 0 až percentage\_area.
* crop\_by\_size() - do argumentů této funkce je nutné zadat parametry width, height, centre. První dva tyto parametry zajišťují absolutní hodnotu oříznutí v daném směru. Centre, pokud je tento parametr true, pak je ořezáváno směrem do středu, pokud je to naopak, pak je ořezaný prostor náhodný.
* crop\_random() - do argumentů této funkce je nutné zadat parametry percentage\_area, randomise\_percentage\_area. Na rozdíl od funkce crop\_centre neprobíhá oříznutí směrem ze středu, ale v náhodném vybraném místě.

### Zrcadlení (Mirroring)

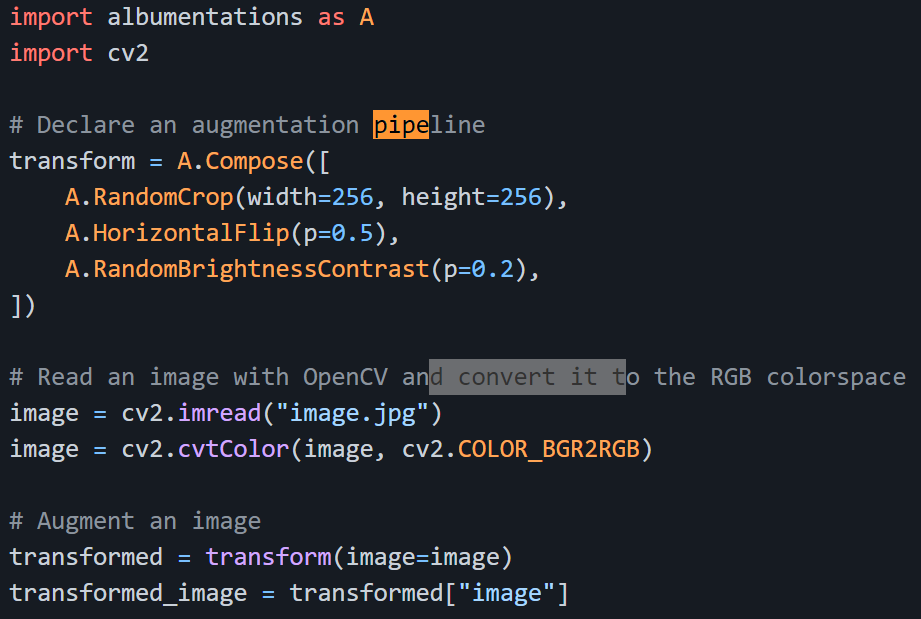
Tento balíček funkcí otáčí (zrcadlí) objekt pomocí následujících funkcí:

* flip\_left\_right() - přetočí objekt pomocí svislé osy zleva doprava
* flip\_top\_bottom() - přetočí objekt pomocí vertikální osy z vrchu dolu
* flip\_random() - přetočí objekt pomocí náhodně vybrané osy

## Albumentations

Tato knihovna nabízí více než 70 různých funkcí pro augmentaci dat. Kromě základních forem augmentace dat, které již byly popsány v knihovně Augmentor, se specializuje především na transformaci obrazu na úrovni pixelů. Díky tomu nabízí augmentaci pomocí změny barvy a jejich odstínů (*RGBShift, ChannelShuffle, ToGray, FancyPCA, HueSaturationValue, ToSepia*), změny sytosti (HueSaturationValue, *ColorJitter*), inverze barev (*InvertImg, Solarize*), změny kontrastu (*CLAHE, RandomContrast, RandomGamma,* *ColorJitter*), změna jasu (*RandomBrightness, ColorJitter*), rozmazání/rozostření (*Blur, MedianBlur, MotionBlur, GaussianBlur, GlassBlur, RandomFog*), zaostření (*Sharpen*), vyrovnání barev (*Equalize*), přidání šumu (*GaussNoise, ISONoise, MultiplicativeNoise, RandomRain, RandomShadow, RandomSnow, RandomSunFlare*), normalizace obrazu (*Normalize*) a kompresi obrazu (*JpegCompression, Downscale, ImageCompression, Posterize*). Velkou výhodou této knihovny je možnost neuvažovat (zahodit) části vstupních dat a zpracovávat pouze relevantní části, k tomu slouží funkce (*ChannelDropout, CoarseDropout, GridDropout, MaskDroupout*).

Vytvoření Pipeline probíhá velmi podobně, jako u předcházející knihovny, ukázka syntaxe je vyobrazena na následujícím obrázku:



Výsledky po provedení augmentace některými funkcemi jsou vyobrazeny na následujícím obrázku.

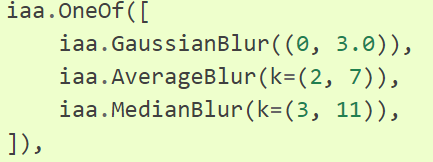


## Imgaug

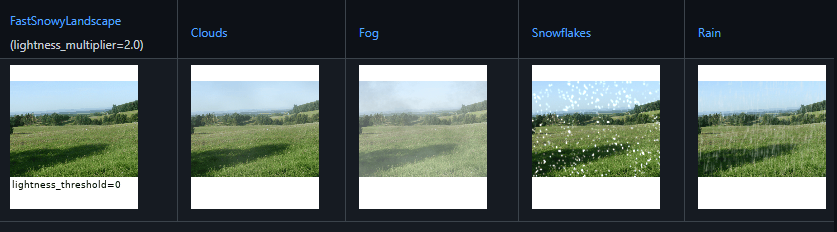
Tato knihovna kromě pokročilé augmentace dat dokáže také augmentovat na základě předem definovaného rozložení pravděpodobnosti (stochastické parametry) a dokáže pracovat s hloubkovými a segmentační mapami.

Oproti předchozím knihovnám lze nastavit aplikaci některých augmentačních funkcí jenom „občas“ (funkce *sometimes()*). Například *sometimes(0.5, GaussianBlur(0.3))* – takto bude *GaussianBlur()* aplikován pouze na 50% vstupních dat. Výhodou je, že těchto funkcí může *Somentimes()* obsahovat více a s touto pravděpodobností bude aplikován celý balíček funkcí na vstupní data.

Dalšími zajímavými funkcemi v této knihovně jsou SomeOf() a OneOf. Například SomeOf((0,5),….) vybere náhodně 0-5 pravidel, které jsou dále uvedeny v argumentu této funkce. Nedoporučuje se aplikace až příliš mnoha augmentačních funkcí najednou, protože mohou být vstupní data změněna k nepoznání. Toto se hodně projevuje u použití více než jednoho rozostření na každý objekt, této problematice zamezuje funkce OneOf(), která náhodně vybere pouze jedno pravidlo a to je aplikováno, dále je uveden příklad:



Zajímavou formu augmentace dat přináší změna počasí, a to především zasněžení objektů, jak je vyobrazeno na následujících obrázcích.

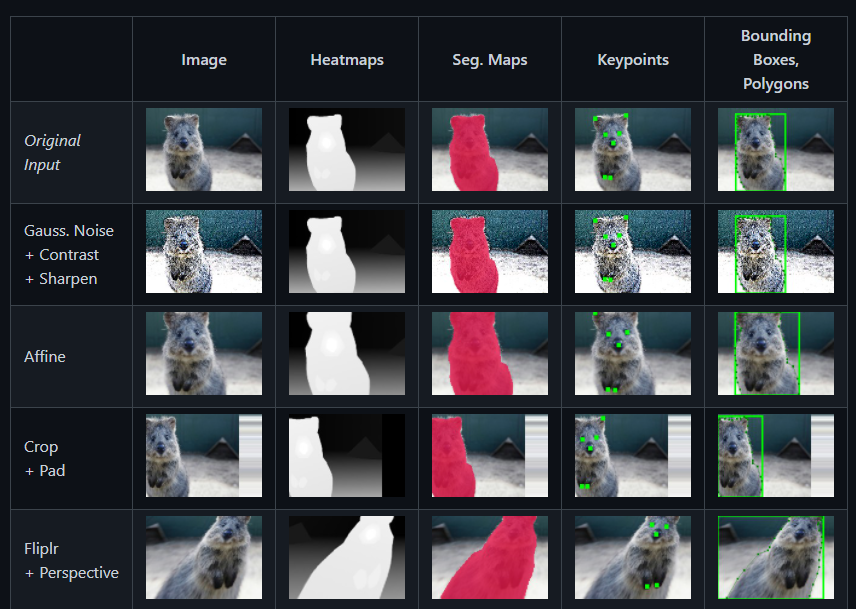




Vytvoření pipeline probíhá obdobně jako u předchozích knihoven viz následující obrázek:

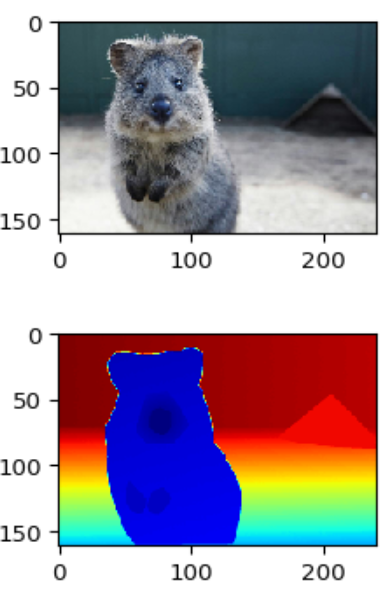


Knihovna umožňuje zobrazení Heatmap, Segmentačních map, klíčových bodů (Keypoints), ohraničujících rámečků a polygonů (Bounding Boxes):



### Heatmaps

Heatmapa je dalším rozměrem objektu, upřesňuje informace například o vzdálenosti objektu od kamery (*DepthMaps*).



### Segmentační mapy

Segmentační mapy pomáhají umělé inteligenci identifikovat objekty na vygenerovaných datech. Segmentační mapa je dalším rozměrem zpracovávaných dat. Při augmentačních operacích, měnících geometrii objektu na snímku (např. rotace, elastické deformace,…) se provede stejná operace i na segmentační mapě, aby objekty na něž segmentační vrstva odkazuje byly na shodné pozici. Segmentační mapu je možné vytvořit například pomocí polygonů.

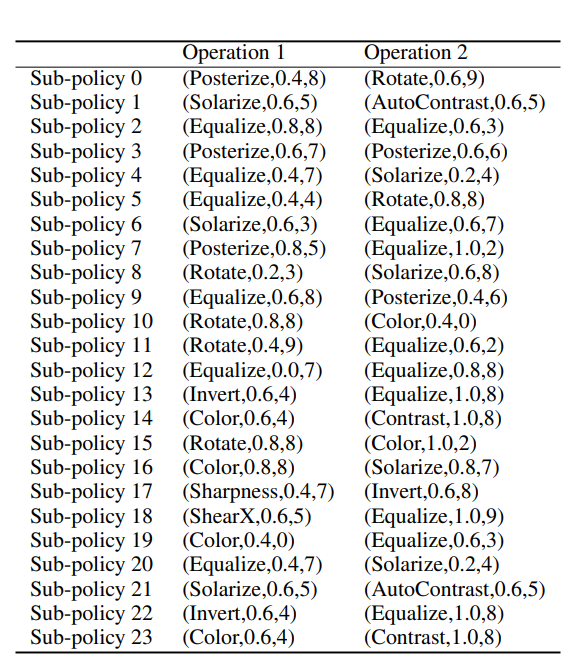


## AutoAugment

Narozdíl od předchozích knihoven, kde jsou programátorem předem definované funkce, určující, jakým způsobem budou vstupní data upravena, tak knihovna AutoAugment obsahuje sady pravidel (policy), jak obrázky upravovat, aby bylo dosaženo dobrých výsledků na testovacích datasetech. Tyto soubory pravidel byly vytvořeny umělou inteligencí na základě předem specifikovaných datasetů, proto každá z dílčí sady pravidel je naučena na práci se specifickými datasety a na nich dosahují nejlepších výsledků. Z toho důvodu nelze vytvořit pouze jednoduchou augmentaci dat, pomocí pouze jedné samostatné funkce (například rotace). Níže uvedené sady pravidel, provedou na vstupních datech všechny své předem definované úpravy podle toho, která sada pravidel je vybrána.

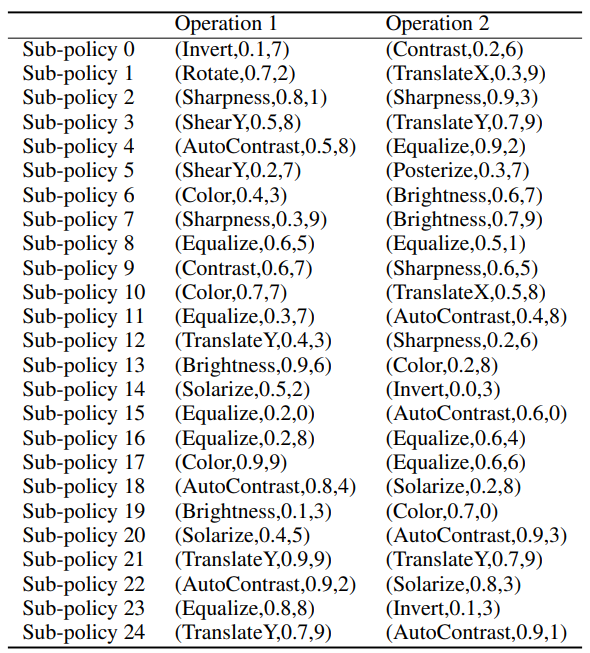
### ImageNetPolicy

Na vstupních datech jsou s určitou pravděpodobností prováděny funkce vypsané níže:



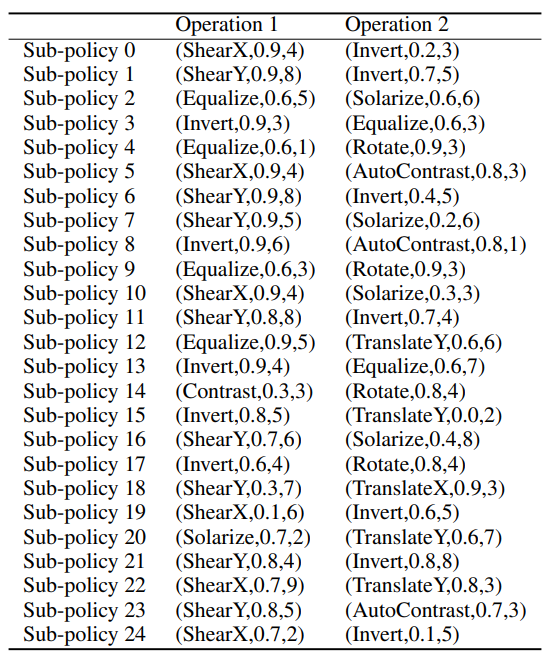
### CIFAR10Policy

Jedná se o pravidlo, které bylo umělou inteligencí vytvořeno na základě datasetu obsahujícího fotografie letadel, automobilů, kamionů, koček, psů, jelenů, koní, žab a ptáků. Na vstupních datech jsou s určitou pravděpodobností prováděny funkce vypsané níže:



### SVHNPolicy

Na vstupních datech jsou s určitou pravděpodobností prováděny funkce vypsané níže:



## PyTorch

Framework PyTorch je velmi obsáhlý, skládá se z několika knihoven, neobsahuje pouze knihovny pro augmentaci obrazových dat, ale také pokročilé metody deeplearningu nejenom obrazových dat, ale například i zvuku a textu. Předchozí zmíněné knihovny, generující rozšířené datasety mohou spolupracovat s frameworkem PyTorch, který je poté může použít pro strojové učení. Pro naši aplikaci je stěžejní balíček *torchvison* a jeho část *transforms*, která slouží k rozšíření datasetu.

### Torchvision.transforms

Tato část knihovny *torchvision* je zaměřena na augmentaci dat. Obsah jejích funkcí je podobný těm, které umí předchozí knihovny.