

Česká zemědělská univerzita v Praze

Technická fakulta

Katedra fyziky



Bakalářská práce

**Monitoring plevelů s využitím konvolučních
neuronových sítí**

Darya Iakhina

© 2023 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Darya Iakhina

Obchod a podnikání s technikou

Název práce

Monitoring plevelů s využitím konvolučních neuronových sítí

Název anglicky

Weed monitoring using convolutional neural networks

Cíle práce

- Vytvořit a otestovat softwarové řešení pro účely rozpoznávání a monitoringu plevelných rostlin s využitím konvoluční neuronové sítě.
- Ekonomicky zhodnotit náklady na vývoj softwarového řešení.

Metodika

- Vypracování literární rešerše s využitím vědeckých databází: Web of Science, Scopus, Google Scholar a dalších.
- Na základě literární rešerše zvolit vhodný typ konvoluční neuronové sítě.
- Implementovat a otestovat zvolenou neuronovou síť.
- Vypracovat ekonomické zhodnocení.

Doporučený rozsah práce

30-40

Klíčová slova

Deep learning, výpočetní technika, obrazová analýza

Doporučené zdroje informací

Asad, MH. & Bais, A. 2020. Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network, Information Processing in Agriculture, Volume 7, Issue 4, Pages 535-545.

Espejo-García, B., Mylonas, N., Athanasakos, L. & Fountas, S. 2020. Improving weeds identification with a repository of agricultural pre-trained deep neural networks. Computers and Electronics in Agriculture, Volume 175.

Chollet, F. Deep learning v jazyku Python. 2019. ISBN: 978-80-247-3100-1

Předběžný termín obhajoby

2022/2023 LS – TF

Vedoucí práce

Ing. Jakub Lev, Ph.D.

Garantující pracoviště

Katedra fyziky

Elektronicky schváleno dne 4. 1. 2022

prof. Ing. Martin Libra, CSc.

Vedoucí katedry

Elektronicky schváleno dne 9. 3. 2023

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 28. 03. 2023

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Monitoring plevelů s využitím konvolučních neuronových sítí“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.03.2023

Poděkování

Ráda bych touto cestou poděkovala Ing. Jakubu Lvovi, Ph.D., vedoucímu této bakalářské práce, za poskytnutí vyčerpávajících informací, zdrojů a dat pro tento výzkum.

Kromě toho bych chtěla poděkovat své rodině a blízkým za obrovskou podporu během celého studia.

Monitoring plevelů s využitím konvolučních neuronových sítí

Abstrakt

Cílem bakalářské práce bylo vyvinout softwarové řešení pro monitorování plevelů pomocí technik hlubokého učení. Ve studii byla použita architektura modelu VGG16 pro detekci a klasifikaci objektů, přičemž datová sada se skládala ze snímků plevelu na pozemcích.

Model VGG16 dosáhl během tréninkové fáze maximální přesnosti 95,5 %. Během predikční fáze však model vykazoval určitá omezení při rozpoznávání jednotlivých druhů plevelů. Toto omezení naznačilo potřebu zlepšit rozpoznávání jejich jednotlivých druhů plevelů v budoucí práci.

V rámci projektu byla také porovnána ekonomická efektivita použití aplikací Google Colab a AWS pro trénování a testování modelu. Byla provedena srovnávací analýza, jejímž cílem bylo určit jejich nákladovou efektivitu pro projekt. To zahrnovalo podrobné vyhodnocení jejich cenových modelů a strategií alokace zdrojů s cílem určit, která platforma poskytuje nejvhodnější rovnováhu mezi cenovou dostupností a výkonem. Z ekonomického hodnocení vyplynulo, že Google Colab byl pro tento projekt nákladově nejefektivnější variantou, která poskytovala dostatečné zdroje pro trénování modelu a provoz softwarového řešení při zachování nízkých nákladů. Testován byl také systém AWS, ale jeho cenový model nebyl pro tento konkrétní projekt tak výhodný.

Klíčová slova: Deep learning, výpočetní technika, obrazová analýza, konvoluční neuronové sítě, předzpracování snímků, cloudové platformy.

Weed monitoring using convolutional neural networks

Abstract

The bachelor thesis aimed to develop a software solution for weed monitoring using deep learning techniques. The study used the VGG16 model architecture for object detection and classification, with a dataset consisting of images of weeds on a parcels.

The VGG16 model achieved the highest accuracy 95.5%. However, during the prediction phase, the model showed some limitations in recognizing individual weed species. This limitation suggested the need for improved recognition of individual weed species in future work.

The project also compared the economic efficiency of using Google Colab and AWS for the training and testing of the model. A comparative analysis was conducted to determine their cost-effectiveness for the project. This involved a detailed evaluation of their pricing models and resource allocation strategies to determine which platform provided the most suitable balance between affordability and performance. The economic assessment revealed that Google Colab was the most cost-effective option for this project, providing sufficient resources to train the model and run the software solution while keeping costs low. AWS was also tested, but its pricing model was not as advantageous for this particular project.

Keywords: Deep learning, computing, image analysis, convolutional neural networks, image preprocessing, cloud platforms.

Obsah

| | |
|--|-----------|
| 1 Úvod | 1 |
| 2 Současný stav vědění | 2 |
| 2.1 Pojem konvoluční neuronové sítě a jejich použití | 2 |
| 2.2 Cloudové platformy | 7 |
| 3 Cíle práce | 9 |
| 4 Materiál a metody | 10 |
| 4.1 Metoda předzpracování snímků..... | 10 |
| 4.2 VGG16 model..... | 13 |
| 4.3 Trénování a ověřování | 13 |
| 4.4 Vývoj softwarového řešení | 15 |
| 4.5 Postup při realizaci programu | 17 |
| 5 Výsledky předzpracování snímků | 19 |
| 6 Výsledky trénování VGG16 | 21 |
| 7 Výsledky predikce | 24 |
| 8 Ekonomické posouzení | 27 |
| 9 Závěr | 30 |
| 10 Seznam použitých zdrojů | 31 |

Seznam obrázků

| | |
|--|----|
| OBR. 1 ARCHITEKTURA MODELŮ VGG16 (SHADEED ET AL, 2020) | 3 |
| OBR. 2 DVOUKROKOVÉ RUČNÍ OZNAČOVÁNÍ PLEVELNÝCH ROSTLIN (ASAD A BAISE, 2020) | 4 |
| OBR. 3 PŘÍTOMNOST PLEVELNÝCH ROSTLIN V PAPRICE PĚSTOVANÉ V POLYHOUSECH (SUBEESH ET AL., 2022) | 5 |
| OBR. 4 KONVERGENČNÍ KŘIVKY NATRÉNOVANÝCH MODELŮ PRO 100 EPOCH (ASAD ET AL., 2020) | 5 |
| OBR. 5 GRAFY PŘESNOSTI A ZTRÁTOVÉ FUNKCE (TRÉNOVACÍCH EPOCH = 30, BATCH SIZE = 16) A) ALEXNET, B) GOOGLENET, C) INCEPTIONV3, D) XCEPTION (SUBEESH ET AL., 2022) | 6 |
| OBR. 6 KAMERA JAI FS-3200D-10GE (JAI, 2023) | 10 |
| OBR. 7 ODSTRANĚNÍ POZADÍ A FUNGOVÁNÍ MASKY. PŮVODNÍ SNÍMEK (A), SNÍMEK S ODSTRANĚNÝM POZADÍM PRO VYTVOŘENÍ MASKY(B), IMPLEMENTACE MASKY NA PŮVODNÍ SNÍMEK S FILTREM NA ODSTRANĚNÍ PLEVELNÝCH ROSTLIN MENŠÍ VELIKOSTÍ (C). | 11 |
| OBR. 8 UKÁZKA SNÍMKŮ RŮZNÝCH DRUHŮ PLEVELŮ. HEŘMÁNKOVEC NEVONNÝ (A) VYKAZUJE VYSOKOU HUSTOTU ROSTLIN, MÁ VLČÍ (B) JE MENŠÍHO ROZMĚRU, A PROTO JE TĚŽKO ODLIŠITELNÝ OD MENŠÍCH POROSTŮ OSTATNÍCH DRUHŮ PLEVELŮ, JEŽATKA KUŘÍ NOHA (C) MÁ NEJVĚTŠÍ POROSTY A VYSOKOU HUSTOTU ROSTLIN, COŽ ZTĚŽUJE ZJIŠTĚNÍ MORFOLOGICKÝCH ROZDÍLŮ JEDNOTLIVÝCH POROSTŮ, ROZRÁZIL PERSKÝ (D) JE V POČÁTEČNÍM STADIU RŮSTU A LZE SI HO SPLÉST S KLÍČKY NAPŘÍKLAD HEŘMÁNKOVCE NEVONNÉHO. | 20 |
| OBR. 9 HISTORIE TRÉNOVÁNÍ | 22 |
| OBR. 10 TYPICKÉ CHYBY, KTERÉ LZE POZOROVAT U KONKRÉTNÍCH TŘÍD PLEVELŮ | 25 |

Seznam tabulek

| | |
|---|----|
| TAB. 1 PŘÍKLAD ZÁPISŮ ANOTACÍ DO CSV SOUBORU. KAŽDÝ ŘÁDEK PŘEDSTAVUJE ÚDAJE O KONKRÉTNÍ ROSTLINĚ NALEZENÉ NA SNÍMKU. | 12 |
| TAB. 2 HISTORIE UČENÍ MODELU. EPOCH JE CYKLUS TRÉNOVÁNÍ, „LOSS“ – HODNOTY ZTRÁTOVÉ FUNKCE PŘI TRÉNOVÁNÍ, „ACC“ – PŘESNOST MODELU PŘI TRÉNOVÁNÍ, „VAL_LOSS“ – HODNOTY ZTRÁTOVÉ FUNKCE PŘI VALIDACI, „VAL_ACC“ – PŘESNOST MODELU PŘI VALIDACI | 21 |
| TAB. 3 VÝSLEDKY PREDIKCE PRO KAŽDÝ DRUH PLEVELŮ | 24 |
| TAB. 4 CENY NA INSTANCE PLATFORMY AWS (AWS PRICING CALCULATOR, N, D,) | 27 |
| TAB. 5 CENY NA PŘEDPLATNÉ GOOGLE COLAB (GOOGLE, N, D,) | 27 |
| TAB. 6 VÝPOČTY NÁKLADŮ PRO JEDNOTLIVÉ MOŽNOSTI IMPLEMENTACE | 28 |

1 Úvod

Od počátku zemědělského hospodaření trápí lidstvo výskyt plevelných rostlin, které vážně narušují optimální vývoj rostlinných kultur. Především se projevuje negativní dopad na dostupnost vody a živin v půdě, což značně snižuje výnosy plodin. Plevelné rostliny také slouží jako ideální prostředí pro šíření škůdců a chorob, což zhoršuje zdravotní stav celého ekosystému. V neposlední řadě tyto druhy negativně ovlivňují plodiny svou konkurenční silou, zastíňují je a tím zabraňují jejich plnému rozvoji. Regulace plevelů je proto jedním z nejdůležitějších pěstitelských úkolů. Pokoušet se o absolutní a trvalé odplevelení je nicméně cíl, jehož lze jen těžko dosáhnout, dokonce je to nejspíše i nemožné.

Tato situace se ale nyní může změnit díky vývoji digitálních a výpočetních technologií. Ty hrají v moderním světě významnou roli. Tyto technologie s automatizací a strojovým učením pomáhají zvyšovat efektivitu, produktivitu, ziskovost a zároveň řeší některé environmentální a sociální problémy, kterým čelí mnoho odvětví včetně zemědělství. Neuronové sítě jsou typem strojového učení, které jsou inspirované strukturou lidského mozku, což jim umožňuje učit se z dat a provádět předpovědi nebo rozhodnutí. Tato technologie způsobila revoluci v oborech, jako jsou rozpoznávání obrazu a zpracování přirozeného jazyka, a umožňuje počítačům provádět úkoly, které pro ně byly dříve považovány za nemožné. V oblasti zemědělství se neuronové sítě již dlouho používají při analýze satelitních snímků polí, identifikaci rostlin, jejich chorob a tak dále. V případě plevelů mohou pomoci určit jejich typ a zajistit včasnou kontrolou výskyt, což v konečném důsledku zefektivní ochranu rostlin.

2 Současný stav vědění

Plevele představují pro zemědělství na celém světě závažný problém, neboť způsobují roční ztráty na výnosech v odhadované výši více než 100 bilionů dolarů (Esposito et al., 2021). Plevele soutěží s komerčními plodinami o živiny, světlo a vodu, což vede ke snížení výnosů, a jejich přítomnost může ovlivnit také kvalitu plodin (Khan et al., 2018). Mezi tradiční metody regulace plevelů patří aplikace herbicidů a mechanická kultivace, tyto metody však mají svá omezení. Chemické herbicidy mohou poškodit necílové rostliny, způsobit problémy v životním prostředí a vyvolat rezistenci vůči herbicidům (Rigon et al., 2020). Mechanické ošetření může také poškodit strukturu půdy a snížit její schopnost zadržovat vodu (Pulkrábek et al., 2019.). Proto vyvstala potřeba cílených, selektivních zásahů proti plevelům.

2.1 Pojem konvoluční neuronové sítě a jejich použití

Jedním z možných řešení je využití strojového učení a konvolučních neuronových sítí (CNN) k detekci a rozlišení plevelů od plodin. Tento přístup byl v posledních letech studován a bylo dosaženo dobrých výsledků (Barbedo, 2019). Použití algoritmů strojového učení může zajistit cílený zásah, protože plevely lze odhalit a zlikvidovat, aniž by byly ovlivněny okolní plodiny.

Algoritmy strojového učení fungují tak, že se učí ze souboru obrazových dat a používají tato data k identifikaci vzorů a znaků, které odlišují plevely od plodin. Konvoluční neuronové sítě jsou typem algoritmu strojového učení, který se obzvláště dobře osvědčil v úlohách rozpoznávání obrazu (Xin & Wang, 2019).

Sítě CNN jsou navrženy tak, aby zpracovávaly vstupní data s topologií podobnou mřížce, jako jsou obrázky, a trénovaly hierarchické reprezentace vstupních dat pomocí několika vrstev konvolučních a asociačních operací. Tato architektura se ukázala jako vysoce efektivní v různých úlohách strojového vidění, včetně klasifikace obrazu, detekce objektů a sémantické segmentace (Kamilaris & Prenafeta-Boldú, 2018).

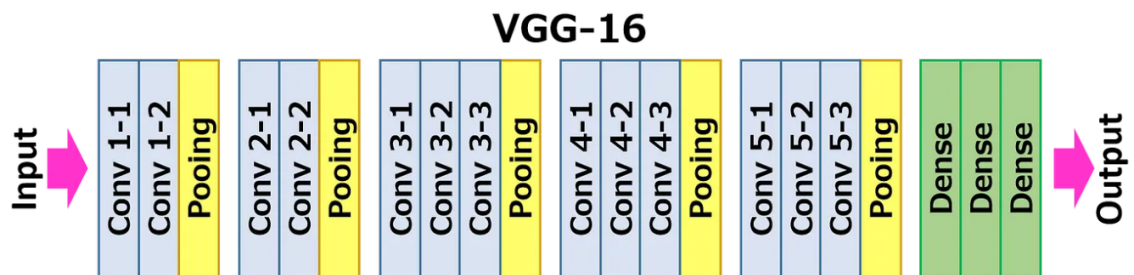
K dosažení vysokého výkonu CNN je často nutné vstupní obrazy různými způsoby předběžně zpracovat. Jedním z běžných kroků předzpracování je normalizace, která spočívá v uvedení hodnot pixelů obrazu do společného rozsahu. To může pomoci zlepšit konvergenci neuronové sítě během trénování a také zlepšit celkovou přesnost modelu (Russakovsky et al., 2015).

Dalším důležitým krokem předběžného zpracování je augmentace dat, jež spočívá v aplikaci různých transformací na vstupní obrazy s cílem zvýšit velikost a rozmanitost

trénovacího souboru. Mezi běžné techniky augmentace patří libovolné oříznutí, převrácení, otočení a změna barvy (Krizhevsky et al., 2012).

Kromě předzpracování může mít na výkonnost významný vliv také volba architektury neuronové sítě. Jedna z nejznámějších architektur CNN je VGG16, která byla představena výzkumníky z Oxfordské univerzity v roce 2014. Na obrázku 1 je schematicky znázorněna architektura systémů VGG16. Tato architektura se vyznačuje použitím malých konvolučních filtrů s pevnou velikostí 3×3 bodů a velkou hloubkou sítě až 16 vrstev. Model VGG16 dosáhl nejlepších výsledků na různých referenčních souborech dat pro klasifikaci obrázků (Simonyan a Zisserman, 2014).

Obr. 1 Architektura modelů VGG16 (Shadeed et al, 2020)



Mezi další populární architektury CNN patří ResNet, Inception a MobileNet, přičemž každá z nich má své silné a slabé stránky. ResNet, představená výzkumníky z Microsoft Research v roce 2015, využívá k přenosu informací z dřívějších vrstev do pozdějších vrstev pásmová spojení, což pomáhá zmírnit problém mizejících gradientů v hlubokých sítích (He et al., 2016). Inception, představená výzkumníky ze společnosti Google v roce 2014, využívá kombinaci konvolučních vrstev s různou velikostí filtrů a operací slučování ke zkoumání hierarchických charakteristik vstupních obrazů (Szegedy et al., 2015). MobileNet, představený výzkumníky ze společnosti Google v roce 2017, navržený jako lehký a efektivní nástroj pro použití na mobilních zařízeních s omezenými výpočetními zdroji, využívá hluboké separovatelné konvoluce k dosažení vysoké přesnosti s menším počtem parametrů než jiné architektury (Howard et al., 2017).

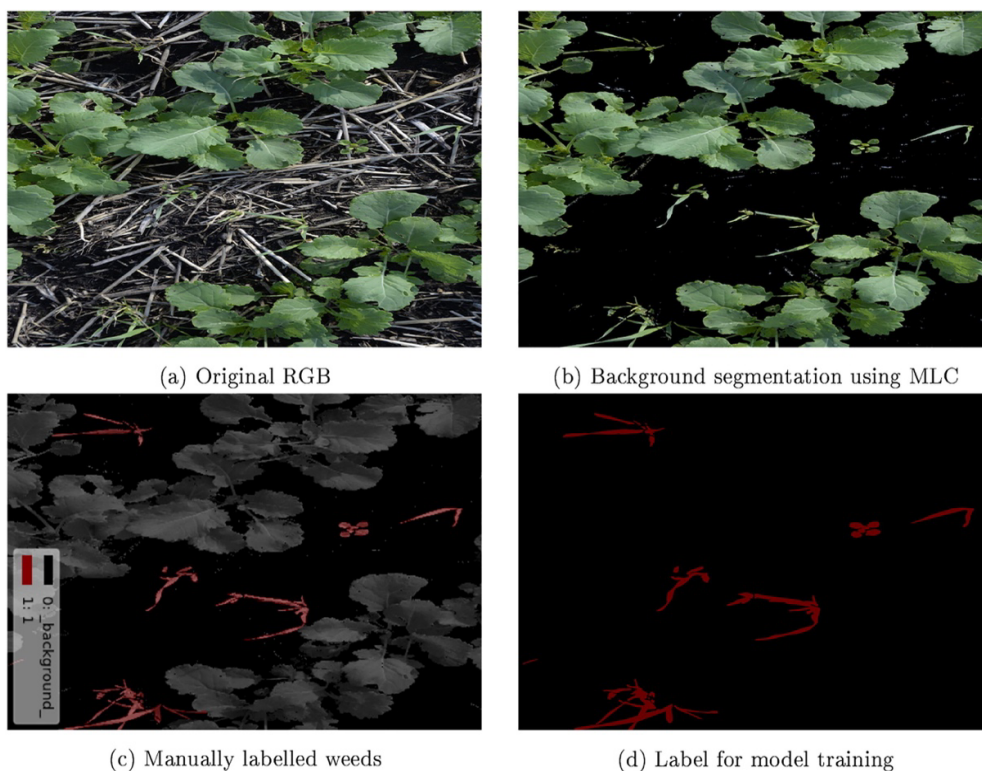
V současné době se detekce, rozlišování a likvidace plevelů provádí pomocí kombinace tradičních metod, jako jsou vizuální kontrola, mechanické odstraňování, použití herbicidů a použití algoritmů strojového učení. Například systém See & Spray vyvinutý společností Blue River Technology využívá algoritmy strojového učení k detekci a likvidaci plevelů pomocí postřikovací trysky, čímž snižuje množství použitých herbicidů (Chostner, 2017)

Konvoluční neuronové sítě prokázaly značné úspěchy v úlohách klasifikace obrazu, včetně detekce plevelu na zemědělských polích. Dvě nedávné studie autorů Asada a Baise

(2020) a Subeeshe et al. (2022) představují různé přístupy k využití CNN pro detekci plevelů na zemědělských polích.

V prvním studiu autoři Asad a Baise (2020) použili k oddělení rostlin řepky od pozadí dvoukrokové ruční označování a barevnou segmentaci. K extrakci texturních rysů ze snímků použili operátor LBP (Local Binary Pattern). Použili také klasifikační algoritmus Maximum Likelihood Classification (MLC) k rozřídění snímků do tříd plevelů a neplevelů. Tento dvoufázový proces výrazně zkracuje dobu potřebnou k ručnímu označování. Proces generování značek je znázorněn na obrázku 2. Výchozí surová fotografie (obr. 2(a)), převedení na snímek se segmentovaným pozadím (Obr. 2(b)), označování plevelů pomocí nástroje Labelme (Obr. 2(c)) a výsledné konečné labely (Obr. 2(d)).

Obr. 2 Dvoukrokové ruční označování plevelných rostlin (Asad a Baise, 2020)



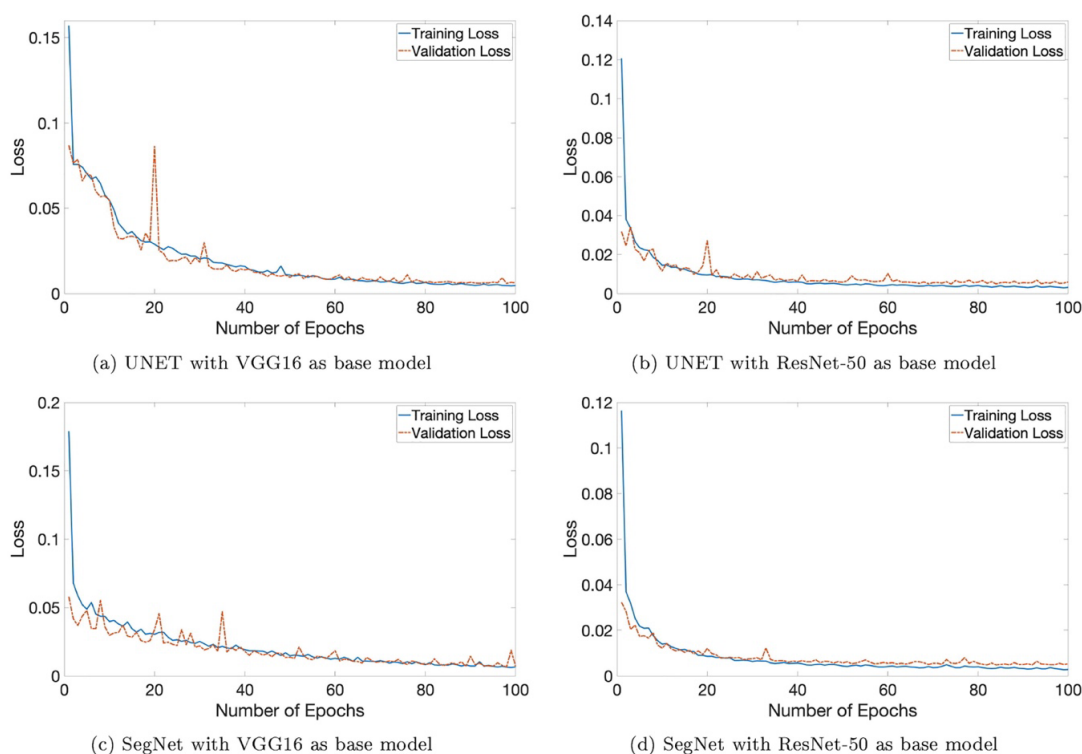
Autoři Subeeshe et al. (2022) použili k předzpracování obrazu kombinaci normalizace barev a morfologických operací. Obrázek 3 znázorňuje výskyt plevelů v porostu papriky pěstované v polyhousech. Techniky normalizace barev použili k redukci barevných variací způsobených osvětlením a faktory prostředí. Použili také morfologické operace k odstranění šumu a vyhlazení okrajů rostlin. Nakonec byla použita technika prahování k převodu předzpracovaných obrazů na binární obrazy.

Obr. 3 Přítomnost plevelných rostlin v paprice pěstované v polyhousech (Subeesh et al., 2022)



Asad a Baise (2020) použili hlubokou architekturu CNN sestávající ze šesti konvolučních vrstev a dvou plně propojených vrstev. Síť natrénovali na datové sadě 3 000 obrázků a dosáhli 97,2% přesnosti při klasifikaci obrázků do tříd plevelů a neplevelů. Průběh konvergence natrénovaných modelů je znázorněn na obrázku 4. Přesnost natrénovaných modelů je vyšší než přesnost klasifikátoru většinové třídy, ResNet-50 má nejvyšší přesnost 98,23 %, u VGG16 je to 98,39 %.

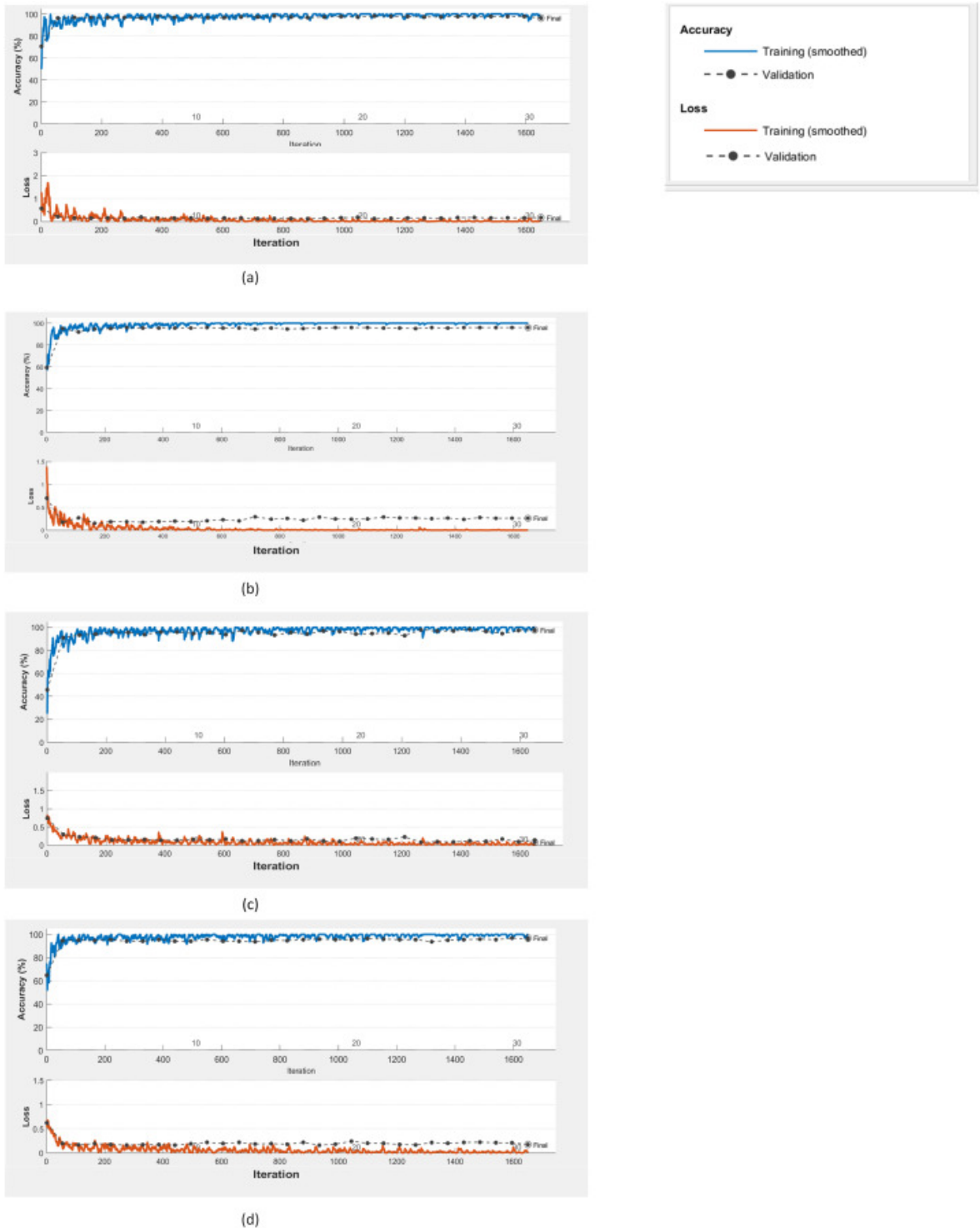
Obr. 4 Konvergenční křivky natrénovaných modelů pro 100 epoch (Asad et al., 2020)



V práci Subeesh et al. (2022) byly použity dvě různé architektury CNN, AlexNet a GoogLeNet. Síť trénovali na datové sadě 10 000 obrázků a dosáhli přesnosti 98,5 % při použití sítě AlexNet a 99,2 % při použití sítě GoogLeNet při klasifikaci obrázků do tříd

plevelů a neplevelů. Změny přesnosti a ztrátové funkce během tréninkové fáze s velikostí dávky 16 a 30 epochami jsou znázorněny na obrázku 5. Grafy ukazují, že ztrátová funkce všech modelů začala v počátečních epochách konvergovat a nedocházelo u ní k výrazným výkyvům.

Obr. 5 Grafy přesnosti a ztrátové funkce (trénovacích epoch = 30, batch size = 16) a) AlexNet, b) GoogLeNet, c) InceptionV3, d) Xception (Subeesh et al., 2022)



2.2 Cloudové platformy

Google Colab a AWS (Amazon Web Services) jsou dvě cloudové výpočetní platformy, které nabízejí různé výhody pro realizaci trénování a nasazení modelů strojového učení.

Google Colab je cloudová služba poskytovaná společností Google, jež umožňuje uživatelům spouštět a vykonávat kód na virtuálních strojích umístěných v infrastruktuře společnosti Google (Google, n. d.). Colab poskytuje rychlejší grafické procesory, delší dobu běhu a více paměti RAM ve srovnání s bezplatnou verzí služby Colab. Google Colab navíc v rozšířených verzích nabízí přednostní přístup k výpočetním zdrojům, což uživatelům umožňuje rychlejší práci (Google, n. d.).

Jednou z hlavních výhod aplikace Google Colab je její snadné používání (Google, n. d.). Uživatelé se mohou jednoduše přihlásit ke svému účtu Google, spustit Colab a začít pracovat na svých projektech v prostředí notebooku Jupyter, aniž by museli nastavovat jakoukoli infrastrukturu. Google Colab také poskytuje přístup k předinstalovaným knihovnám a frameworkům, jako jsou TensorFlow, PyTorch a OpenCV, které se běžně používají v projektech strojového učení (Google, n. d.).

Používání softwaru Google Colab má však také některá omezení (Google, n.d.). Zaprvé se jedná o sdílený nástroj, což znamená, že uživatelé soutěží o kapacity s ostatními uživateli. Za druhé, limit doby běhu 24 hodin znamená, že dlouho probíhající experimenty mohou být restartovány. A konečně, Colab Pro Plus nemůže poskytovat dostatek výpočetních zdrojů pro větší a složitější projekty.

Na druhou stranu AWS poskytuje komplexní sadu cloudových výpočetních služeb, které lze využít k realizaci projektů strojového učení, včetně Amazon EC2, Amazon SageMaker a Amazon S3 (AWS Documentation, n. d.). AWS nabízí řadu typů virtuálních strojů, včetně instancí GPU optimalizovaných pro pracovní zátěže strojového učení, a poskytuje flexibilní možnosti stanovení cen.

Jednou z hlavních výhod používání služby AWS je její škálovatelnost (AWS Documentation, n. d.). AWS umožňuje uživatelům snadno navyšovat, nebo snižovat výpočetní zdroje v závislosti na jejich aktuálních potřebách, což je užitečné zejména pro větší a složitější projekty. Kromě toho AWS poskytuje různé nástroje a služby pro správu a nasazení modelů strojového učení.

Používání AWS má však i některé potenciální nevýhody. Za prvé, AWS může být ve srovnání s Colab Pro Plus složitější na nastavení a konfiguraci (AWS Documentation, n. d.). Za druhé, pro efektivní používání systému AWS je zapotřebí určitá úroveň odborných znalostí,

zejména pokud jde o správu a nasazování modelů. A konečně, náklady na používání AWS mohou rychle narůst, zejména pokud si uživatelé nedávají pozor na využívání zdrojů (AWS Documentation, n. d.).

Pokud jde o realizaci projektu, jeví se jak Colab, tak AWS jako vhodné nástroje. Google Colab by mohl být dobrou volbou pro experimenty menšího rozsahu, a to díky snadnému použití a dostupnosti. AWS by mohl být dobrou volbou pro experimenty většího rozsahu nebo pro uživatele, kteří vyžadují větší kontrolu a flexibilitu nad svými výpočetními zdroji. Volba mezi Google Colab a AWS bude nakonec záviset na konkrétních potřebách a požadavcích uživatele (Google, n.d; AWS Documentation, n. d.).

3 Cíle práce

Prvním cílem této práce je vytvořit a otestovat softwarové řešení pro rozpoznávání a monitorování plevelů použitím konvoluční neuronové sítě, která prokázala svou účinnost v úlohách klasifikace obrazu a modelu. Pro tuto úlohu použijeme architekturu VGG16, která prokázala v předchozích studiích, například ve studii autorů Subeesh et al., (2022), vysokou přesnost. Modely budou trénovány na souboru dat se snímky plevelných rostlin, aby se naučily vzory a vlastnosti, které odlišují plevelné rostliny od ostatních rostlin. Poté budou natrénované modely testovány na samostatném souboru dat, aby se vyhodnotila jejich výkonnost.

Druhým cílem této práce je provést ekonomické posouzení nákladů na vývoj softwarového řešení s využitím cloudových služeb Amazon Web Services (AWS) a Google Colab. Budou odhadnuty náklady spojené s využíváním služeb AWS a Google Colab, včetně nákladů na výpočetní zdroje, úložiště a přenos dat.

Pro ekonomické vyhodnocení budou analyzovány cenové modely nabízené AWS a Google Colab, včetně instancí na vyžádání, rezervovaných instancí a bodových instancí. Náklady spojené s jednotlivými cenovými modely budou vyhodnoceny a porovnány s cílem určit nákladově nejefektivnější variantu vývoje softwarového řešení.

4 Materiál a metody

Data poskytnutá katedrou fyziky byla získána z měření provedených na pozemku v Suchdole o celkové ploše 144 m², který byl rozdělen na 12 menších částí, každá o rozměrech 2 × 6 m. V roce 2022 bylo z dostupných semen zaseto deset druhů plevelů, z nichž 5 druhů vyklíčilo v době pořizování záznamů.

K vytvoření kamerového systému byla využita multispektrální kamera JAI FS-3200D-10GE (Obrázek 6) s rozlišením 2048 × 1536 pixelů. Tato kamera rozděluje zachycené světlo pomocí optických hranolů na dva samostatné snímače CMOS 1/1,8 (úhlopříčka 8,89 mm). Jeden snímač zachycuje vlnové délky ve viditelné oblasti, zatímco druhý zachycuje NIR záření v rozsahu 720–1000 nm. Snímky se ukládají ve formátu TIFF do přidruženého počítače, který se používá také k ovládání kamery. K vytvoření snímků a správnému nastavení kamery byl použit software JAI eBUS Player.

V této studii byla použita sada dat obsahujících 12,9 GB párových infračervených a barevných snímků nafocených ve dnech 21. a 23. září roku 2022 s pěti různými druhy plevelů. Datová sada byla uspořádána do samostatných složek pro každý druh.

Obr. 6 Kamera JAI FS-3200D-10GE (JAI, 2023)

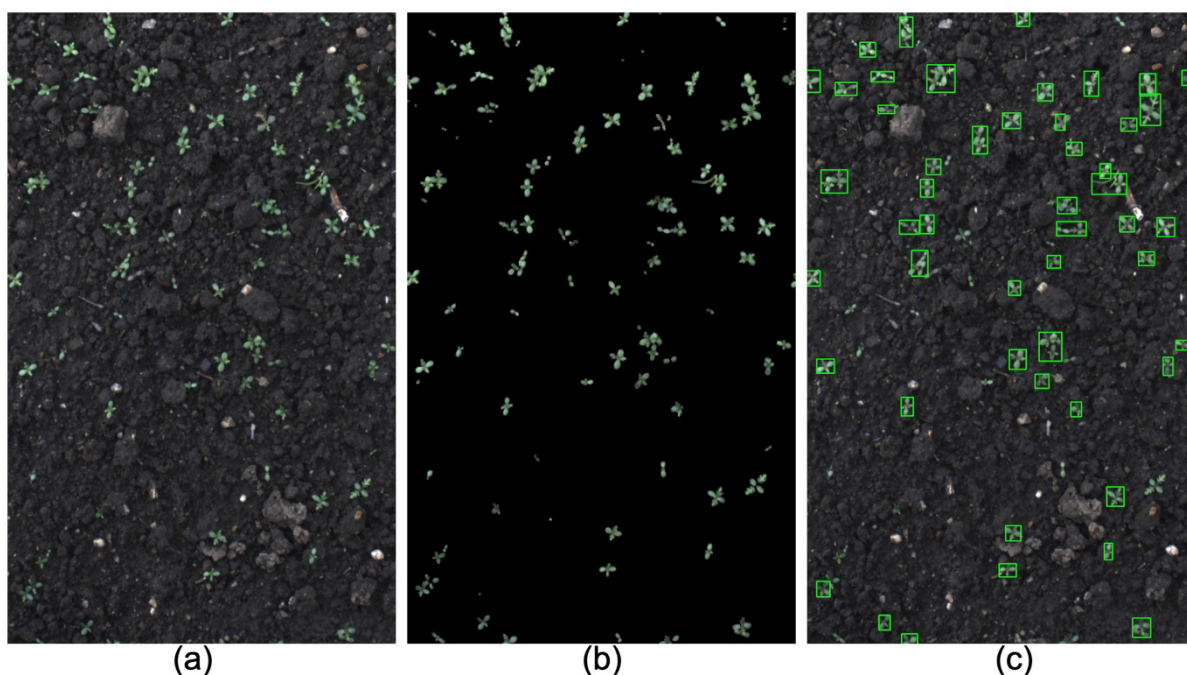


4.1 Metoda předzpracování snímků

Odstranění pozadí je obzvláště důležité v případech, kdy mají rostliny a pozadí podobnou barvu, texturu nebo tvar. Technika odstranění pozadí použitá v tomto projektu byla založena na přístupu barevné segmentace, který využívá barevný prostor HSV (hue, saturation, value of lightness). Barevný prostor HSV je upřednostňován před ostatními barevnými prostory díky své schopnosti oddělit barevné informace od jasu a sytosti. Prvním krokem techniky odstraňování pozadí je převod vstupního obrazu RGB do barevného prostoru HSV. Poté se použije barevná segmentace k oddělení zeleného pozadí od plevelů. Segmentace se provádí

prahováním kanálů odstínu a sytosti, po němž následují morfologické operace pro odstranění malých objektů a vyplnění mezer ve výsledné masce. Na Obr. 7 je vidět původní barevný snímek (a), předzpracovaný snímek s odstraněným pozadím (b) a převod masky zpět na barevný (c). Pro eliminaci zbytečných dat bylo rozhodnuto stanovit limit pro objekty menší než 30 pixelů, protože menší porosty jsou obtížně rozpoznatelné i pro zkušeného člověka.

Obr. 7 Odstranění pozadí a fungování masky. Původní snímek (a), snímek s odstraněným pozadím pro vytvoření masky(b), implementace masky na původní snímek s filtrem na odstranění plevelných rostlin menší velikosti (c).



Odstraněním pozadí a odfiltrováním menších objektů se neuronová síť může zaměřit pouze na cílové rostliny na snímku, čímž se sníží pravděpodobnost chybné klasifikace a zvýší se přesnost detekčního procesu. Použití masek dále zvyšuje přesnost neuronové sítě tím, že zvýrazňuje oblasti zájmu a potlačuje oblasti obrazu, které nejsou pro úlohu detekce a klasifikace relevantní.

Je důležité poznamenat, že krok předzpracování není spolehlivý a stále může vést k chybám v důsledku přítomnosti překrývajících se prvků nebo jiných rostlin v obraze. V takových případech může být k zajištění přesnosti výsledků nutná ruční korekce a kontrola snímků. Tento proces může být časově náročný, ale je nezbytný pro zajištění přesnosti označování.

Proces ruční kontroly zahrnující zkoumání snímků a identifikaci případných chyb v automatickém označování byl proveden pomocí nástroje LabelImg. Chyby byly poté opraveny v souboru formátu CSV, jehož část je vidět v Tab. 1 s označenými daty, která byla použita k ukládání výsledků procesu označování. Tento soubor se skládá z několika sloupců,

včetně „x“, „y“, „w“, „h“ a „plant_name“. Sloupce „x“ a „y“ představují souřadnice levého horního rohu ohraničujícího rámeček, který obklopuje objekt rostliny na snímku typu NumPy Array. Sloupce „w“ a „h“ představují šířku a výšku ohraničujícího rámečku. Tyto čtyři sloupce společně poskytují informace potřebné k přesnému určení polohy a identifikaci rostlinného objektu na snímku.

Tab. 1 Příklad zápisu anotací do CSV souboru. Každý řádek představuje údaje o konkrétní rostlině nalezené na snímku.

| Image_path | x | y | w | h | plant_name |
|-------------------------------|---------------|----------------------|---------------|----------------------|-------------------|
| ježatka kuří noha/01051C.tiff | 0.275390625 | 0.005208333333333333 | 0.02099609375 | 0.031901041666666664 | ježatka kuří noha |
| ježatka kuří noha/01051C.tiff | 0.927734375 | 0.0 | 0.0283203125 | 0.078125 | ježatka kuří noha |
| ježatka kuří noha/01051C.tiff | 0.4453125 | 0.0 | 0.111328125 | 0.15625 | ježatka kuří noha |
| ježatka kuří noha/01051C.tiff | 0.2392578125 | 0.0 | 0.02734375 | 0.055338541666666664 | ježatka kuří noha |
| rozrazil perský/01347C.tif | 0.70703125 | 0.9876302083333334 | 0.015625 | 0.012369791666666666 | rozrazil perský |
| rozrazil perský/01347C.tif | 0.59765625 | 0.9798177083333334 | 0.01513671875 | 0.020182291666666668 | rozrazil perský |
| rozrazil perský/01347C.tif | 0.94580078125 | 0.9680989583333334 | 0.01611328125 | 0.01171875 | rozrazil perský |
| rozrazil perský/01347C.tif | 0.7978515625 | 0.9602864583333334 | 0.02197265625 | 0.016927083333333332 | rozrazil perský |

Sloupec „plant_name“ slouží k uložení označení třídy přiřazené k objektu rostliny a sloupec „image_path“ naznačuje cestu do snímku. Cílem bylo zajistit, aby každý objekt na snímku byl správně označen a aby nedocházelo k falešně pozitivním, nebo falešně negativním výsledkům. Automatické označování výrazně urychlilo proces a snížilo pracovní zátěž, zatímco následná ruční kontrola zajistila přesnost a preciznost označování.

Normalizace obrazu je další technikou předzpracování použitou v této práci ke zlepšení výkonu systému detekce plevelů založeného na CNN. Normalizace byla provedena odečtením střední hodnoty pixelů datové sady ImageNet. Výsledný obraz se poté vydělí směrodatnou odchylkou datové sady ImageNet, aby se hodnoty pixelů škálovaly v rozmezí -1 až 1 . Proces normalizace pomáhá snížit vliv světelných podmínek, nastavení kamery a dalších parametrů pořízení obrazu na klasifikační výkonnost modelu CNN.

Techniky pro augmentaci obrazu použité v tomto projektu zahrnovaly horizontální převrácení, vertikální převrácení, náhodné otáčení a náhodné zvětšení.

Horizontální a vertikální převrácení zahrnuje převrácení vstupního obrazu podél horizontální, resp. vertikální osy. Náhodné otáčení zahrnuje otáčení vstupního obrazu o náhodný úhel mezi -30 a 30 stupni. Náhodné zvětšení zahrnuje zvětšení vstupního obrazu náhodným faktorem mezi $0,8$ a $1,2$. Proces rozšiřování obrazu byl proveden pomocí modulu

Keras ImageDataGenerator, který generuje nové obrázky během procesu trénování, čímž snižuje potřebu dalšího úložného prostoru.

4.2 VGG16 model

V této práci byl model VGG16 použit jako základní architektura pro vývoj modelu CNN pro detekci plevelů.

Model VGG16 se skládá ze 13 konvolučních vrstev, na něž navazují tři plně propojené vrstvy (Obr. 1). Konvoluční vrstvy jsou uspořádány do pěti bloků, přičemž každý blok se skládá ze dvou nebo tří konvolučních vrstev, po nichž následuje vrstva s maximálním spojením. Plně propojené vrstvy provádějí konečnou klasifikaci obrazu na základě extrahovaných rysů.

Během trénování byl model VGG16 inicializován pomocí předem natrénovaných vah ze souboru dat ImageNet, což mu umožnilo naučit se rysy, užitečné pro mnoho úloh rozpoznávání objektů. Poslední vrstvy modelu však byly upraveny pro specifickou úlohu detekce plevelů. Konkrétně byly závěrečné plně propojené vrstvy nahrazeny novými, jejichž výstupem bylo pět tříd odpovídajících pěti druhům plevelů.

4.3 Trénování a ověřování

Proces trénování a ověřování je klíčovým krokem v každém projektu strojového učení, včetně systému detekce plevelů vyvinutého v tomto projektu. Účelem trénování je naučit model strojového učení rozpoznávat a klasifikovat plevelné rostliny na základě předzpracovaných snímků. Proces validace slouží k posouzení výkonnosti natrénovaného modelu na samostatné sadě snímků, které nebyly použity ve fázi trénování.

Proces trénování byl proveden na souboru 1130 snímků pěti druhů plevelů na 50 epochách. Snímky byly rozděleny na trénovací a validační sadu v poměru 80/20. Tréninková sada byla použita k tréninku modelu a validační sada byla použita k vyhodnocení výkonu modelu na nezobrazených datech. Snímky v obou sadách byly před vložením do modelu předběžně zpracovány výše uvedenými metodami.

Ve fázi trénování je model VGG16 vyladěn pro konkrétní úlohu detekce plevelů. Proces začíná zmrazením předem natrénovaných vrstev modelu VGG16, což znamená, že váhy těchto vrstev nebudou během procesu trénování aktualizovány. To se provádí proto, aby se zachovala schopnost modelu extrahovat obecné rysy z obrázků, zatímco se soustředí na trénování nově přidaných vrstev pro rysy specifické pro plevel.

Po zmrazení natrénovaných vrstev se architektura modelu upraví přidáním dvou dalších vrstev. První přidaná vrstva se skládá z 28 jednotek s aktivační funkcí ReLU (Rectified Linear Unit), která vypadá takto:

$$f(x) = \max(0, x), \quad (1)$$

Funkce ReLU, kde x je vstupní hodnota, která počítá výstup jako maximum buď vstupní hodnoty, nebo nuly, čímž do modelu vnáší nelinearitu. ReLU je populární aktivační funkce v hlubokém učení především proto, že pomáhá zmírnit problém mizejícího gradientu a urychluje trénování (DeepAI, 2019).

Druhá přidaná vrstva se skládá z 5 jednotek, což odpovídá počtu tříd plevelů v problému. Tato vrstva využívá aktivační funkci „softmax“, která počítá pravděpodobnostní rozdělení vstupních hodnot. Vzorec této funkce je následující:

$$f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \text{ pro } i = 1, \dots, K \text{ a } x = (x_1, \dots, x_K) \in \mathbb{R}^K, \quad (2)$$

kde x je vstupní vektor s dimenzí K .

Funkce softmax pomáhá určit třídu s nejvyšší pravděpodobností a umožňuje tak modelu provést konečnou předpověď (Banerjee et al., 2020).

Nakonec je model sestaven pomocí kategoriální křížové entropie a optimalizátoru Adam. Kategoriální křížová entropie je vhodná pro problémy klasifikace více tříd, protože měří nepodobnost mezi předpovídaným rozdělením pravděpodobnosti a skutečnými označeními tříd (ztrátová funkce). Vzorec je následující:

$$L_{\text{cross-entropy}}(\hat{y}, y) = -\sum_i y_i \log(\hat{y}_i) \quad (3)$$

V uvedené rovnici vektor $y = y_1, \dots, y_n$, udává distribuci popisků 1 až n . Vektor $\hat{y} = \hat{y}_1, \dots, \hat{y}_n$ pak reprezentuje výstup klasifikátoru (Kodil, 2013). Adamův optimalizátor je adaptivní optimalizační algoritmus míry učení. Optimalizátor upravuje rychlost učení pro každý parametr zvlášť, což umožňuje rychlejší konvergenci a lepší výkon během trénování. (Tato & Nkambou, 2018)

Využitím kategoriální ztráty křížové entropie a Adamova optimalizátoru se model dokáže efektivně naučit specifické rysy plevelů při minimalizaci ztráty. Po sestavení je model připraven k tréninkové fázi, kde bude pomocí připravené sady dat doladěn tak, aby přesně klasifikoval a detekoval různé druhy plevelů.

Přesnost měří podíl správných předpovědí z celkového počtu predikcí provedených modelem. V klasifikačním modelu pro více tříd, jako je v našem případě, se přesnost vypočítá jako podíl počtu správně klasifikovaných objektů a celkového počtu objektů v souboru dat. Čím vyšší je přesnost, tím lepší je výkon modelu.

Tyto úpravy a konfigurace umožňují modelu naučit se specifické vlastnosti plevelných druhů a provádět přesnou klasifikaci a detekci. Proces validace byl proveden na samostatné sadě 336 obrázků s použitím stejných kroků předzpracování jako v tréninkové fázi. Účelem validace je posoudit výkonnost natrénovaného modelu na nové sadě snímků, které nebyly použity při trénování. Přesnost validace byla použita k měření výkonnosti modelu na této nové sadě snímků. Validací přesnost poskytuje odhad výkonu modelu na nezkoumaných datech, což je důležitým měřítkem jeho zobecnitelnosti.

4.4 Vývoj softwarového řešení

Softwarové řešení tohoto projektu je k dispozici na GitHubu na adrese https://github.com/darya-sea/CNN_monitoring. Jako programovací jazyk byl použit Python 3.10 vzhledem k jeho možnostem výkonu, flexibilitě a dostupnosti v projektu použitých knihoven. Projekt využívá několik knihoven Pythonu, včetně OpenCV-Python, NumPy, Matplotlib, Pandas, TensorFlow a Unidecode. Tyto knihovny byly vybrány pro jejich schopnost efektivně zpracovávat velké množství obrazových dat a manipulovat s nimi a také pro jejich kompatibilitu s modely hlubokého učení použitými v projektu.

OpenCV-Python (verze 4.7.0.72) je populární knihovna počítačového vidění, která poskytuje širokou škálu funkcí pro zpracování a analýzu obrazu. Knihovna také poskytuje funkce pro načítání obrázků v různých formátech, včetně JPEG, PNG a TIFF. Pomocí OpenCV-Python lze například měnit velikost obrázků, jejich barevné prostory, provádět prahování obrázků a aplikovat různé typy filtrů pro odstranění šumu a zlepšení jejich kvality (OpenCV, n. d.). V této bakalářské práci se OpenCV-Python používá k provádění segmentace, což je proces rozdělení obrazu na více segmentů nebo oblastí. Prahování bylo provedeno k převodu barevného obrazu plevelu na binární obraz, který je poté použit k vyčlenění jednotlivých rostlin z pozadí. Pomocí OpenCV-Python byly extrahovány vlastnosti rostlin, jako jsou výška, šířka a plocha a tyto funkce jsou pak použity k trénování modelů strojového učení. OpenCV-Python byl také použit k vizualizaci výsledků zpracování obrazu a algoritmů strojového učení.

NumPy (verze 1.24.2) je knihovna jazyka Python používaná pro numerické výpočty, která poskytuje podporu pro velká vícerozměrná pole a matice. Nabízí širokou škálu matematických funkcí pro práci s těmito poli, včetně základních aritmetických operací, lineární

algebry, Fourierových transformací a statistické analýzy (NumPy Documentation, n. d.). V této práci byl NumPy využíván ke zpracování a manipulaci s obrázky.

Matplotlib (verze 3.7.1) je knihovna pro vizualizaci dat v jazyce Python, jež uživatelům umožňuje vytvářet různé typy grafů, například čárové, sloupcové, rozptylové, histogramy a mnoho dalších (Matplotlib 3.7.1 documentation., n. d.). V projektu byl Matplotlib použit pro vizualizaci výsledků kroků zpracování obrazu a modelových předpovědí, také ve spojení s dalšími knihovnami, jako jsou NumPy a OpenCV-Python, pro manipulaci s obrázky a jejich vizualizaci. Například funkce „`pyplot.imshow()`“ slouží k zobrazení původních a předzpracovaných obrázků a funkce „`pyplot.plot()`“ slouží k vykreslení metrik výkonnosti modelu.

Pandas (verze 1.5.3) je knihovna jazyka Python, která se používá k manipulaci s daty a jejich analýze. Poskytuje datové struktury pro efektivní ukládání a manipulaci s velkými soubory dat a funkce pro transformaci, agregaci a vizualizaci dat (User Guide, n. d.). V projektu se Pandas používá ke čtení a manipulaci s daty v souborech CSV. Soubory CSV obsahují metadata o obrázcích, například názvy obrázků a k nim přiřazené popisky (Tab. 1). Jednou z důležitých funkcí programu Pandas použitých v tomto projektu je „`groupby`“. Tato funkce slouží k seskupování metadat obrázků podle přidružených štítků. Seskupená metadata pak byla použita k vytváření souhrnných statistik a vizualizací. Kromě toho byla knihovna Pandas použita k přípravě dat pro trénování a testování modelů strojového učení.

TensorFlow (verze 2.11) je populární open-source knihovna pro strojové učení, kterou vyvinul tým Google Brain. Je široce používána pro vytváření a trénování modelů hlubokého učení. TensorFlow umožňuje uživatelům vytvářet složité neuronové sítě s různými vrstvami, jako jsou konvoluční, rekurentní a husté vrstvy. Nabízí také různé optimalizační algoritmy a nástroje pro vyhodnocování modelů (API documentation: tensorflow V2.11.0., n. d.) byl použit pro vývoj a trénování modelu hlubokého učení použitého pro monitoring plevelle. V projektu bylo využito vysokoúrovňové rozhraní Keras API, které je postaveno nad TensorFlow. Keras zjednodušuje proces vytváření a trénování modelů hlubokého učení tím, že poskytuje uživatelsky přívětivé a intuitivní rozhraní. K předzpracování a rozšíření dat byla rovněž použita knihovna TensorFlow. Obrázky byly předzpracovány normalizací hodnot pixelů tak, aby byly v rozmezí 0 a 1. Pro zvětšení velikosti trénovacího souboru dat a zlepšení zobecnění modelu byly použity techniky rozšiřování dat, jako jsou rotace, převrácení a škálování. TensorFlow sehrál v této bakalářské práci klíčovou roli, neboť poskytl potřebné nástroje a infrastrukturu pro vývoj vysoce výkonného modelu detekce plevelle.

Unidecode (verze 1.3.6) je knihovna jazyka Python, která slouží k převodu textu Unicode na text ASCII. Tato knihovna byla v projektu použita pro standardizaci názvů rostlinných druhů. Názvy rostlin jsou získávány z názvů souborů s obrázky a tyto názvy obsahují speciální znaky nebo akcenty, které nejsou součástí znakové sady ASCII. Unicode převádí znaky na jejich nejbližší reprezentaci ASCII.

V projektu byla použita knihovna unidecode v souboru „visualisation.py“ k extrakci názvů rostlinných druhů z názvů souborů obrázků. Názvy souborů se nejprve rozdělí na jednotlivé složky a z názvu souboru se pomocí regulárních výrazů extrahuje název rostlinného druhu. Poté se použije knihovna unidecode k převodu znaků Unicode v názvu rostlinného druhu na znaky ASCII (*Unidecode 1.3.6. Project description, n. d.*)

Například název „heřmánkovec nevonný“ obsahuje znaky „ř“, „á“ a „ý“, které nejsou součástí znakové sady ASCII. Knihovna unidecode převede tyto znaky na „r“, „a“ a „y“ a vrátí řetězec ASCII „hermankovec nevonny“. Tento standardizovaný název pak lze použít k označování snímků nebo vytváření výkresů.

Knihovny s komentářem na konci, například „tensorflow-macos==2.10“ a „tensorflow-metal==0.6“, jsou alternativní verze knihovny TensorFlow, které lze využít, pokud na určitých platformách nebo systémech nelze pracovat s nejnovější verzí.

4.5 Postup při realizaci programu

Příprava dat:

1. Načte metadata ve formátu .json obsahujícím parametry, jako jsou minimální délka a šířka objektů v obrázcích (maska).
2. Vygeneruje seznam všech obrázků v podsložkách, zamíchá je a rozdělí v poměru 80/20 pro trénovací a validační sadu. Každý obrázek projde předzpracováním pomocí knihovny OpenCV-Python, včetně odstranění pozadí a identifikace objektů pomocí funkce „findContours“. Během fáze předzpracování objektů se ořezané objekty zmenší a uloží jako hodnoty typu „Array“ pomocí NumPy. Rozměry objektů jsou ověřeny podle metadat a objekty splňující určitá kritéria jsou uloženy do konečné složky „datasets“.

Trénování:

1. Vytvoření výstupní složky a složky pro zálohování, do nichž se ukládají tréninkové a validační obrázky. Obrázky se načítají pomocí nástroje „ImageDataGenerator“ systému Keras s režimem kategoričkových tříd a velikostí dávky 16. Indexy tříd jsou uloženy pro použití v kroku trénování.

2. Je vytvořen model VGG16 importovaný ze systému Keras. Vrstvy jsou zmrazeny a architektura modelu je upravena přidáním dvou vrstev s 28 jednotkami pomocí aktivační funkce „ReLU“ a 5 jednotek (kolik máme druhů plevelných rostlin) s aktivační funkcí „Softmax“. Model je sestaven pomocí měření ztráty kategoriální křížové entropie a optimalizátoru „Adam“, obojí importované z Keras. Funkce zpětného volání, jako jsou „backup_restore“, „checkpoint“ a „early_stopping“, jsou nakonfigurovány pomocí nástrojů zpětného volání systému Keras. Tyto funkce umožňují pokračovat v trénování od poslední epochy, uložit model nebo váhy v případě přerušení a zastavit proces trénování, pokud se ztráta po určitém počtu epoch nesníží.
3. Funkce „fit“ se volá s potřebnými parametry včetně iterátorů dat, funkcí zpětného volání a počtu epoch. Tím se zahájí proces trénování. Po dokončení je historie trénování vizualizována jako graf pomocí knihovny Matplotlib a data historie jsou uložena do nových souborů .json a .csv.

Predikce:

1. Funkce „get_best_model“ vybere nejpřesnější model ze složky obsahující natrénované modely. Třídy rostlin jsou načteny ze souboru .json.
2. Funkce „prediction“ se volá se složkou obsahující obrázky nebo s jedním obrázkem a cestou k souboru modelu s nejvyšší přesností. Model je načten do systému Keras.
3. Obrázky ve formátu .TIFF jsou zpracovány pomocí funkcí knihovny OpenCV-Python. Každý obrázek projde odstraněním pozadí, detekcí obrysů pomocí funkce 'findContours', extrakcí souřadnic a ověřením velikosti. Objekty jsou oříznuty pomocí masky a uloženy do pole s příslušnými souřadnicemi (get_bonding_boxes).
4. Objekty jsou zmenšeny na rozměr 150 × 150 pixelů a předzpracovány. Filtrování se provede u objektu s pravděpodobností předpovědi vyšší než 90 % a uloží se souřadnice a index třídy.
5. Indexy a výsledky jsou předány funkci „save_predicted_results“ (uložit předpovězené výsledky) z vizualizační třídy, která výsledky uloží do souboru .csv s podrobnými údaji o procentuálním zastoupení každého identifikovaného objektu. Tento krok je proveden pomocí knihovny Pandas pro manipulaci s daty a jejich ukládání.
6. Funkce „show_predicted_images“ zobrazí výsledky ve formě obrázků s maskou třídy a popiskem pro každý zjištěný objekt. Zobrazí se také procento pro detekované druhy plevelů na vybraném snímku. Vizualizace je dosaženo pomocí kombinace funkcí OpenCV-Python a Matplotlib.

5 Výsledky předzpracování snímků

Proces přípravy dat zahrnoval shromáždění a označení snímků a jejich následné rozdělení na trénovací a validační množinu. Výsledky tohoto procesu jsou prezentovány pro každý z pěti zkoumaných druhů rostlin (Tab. 2).

Tab. 2 Připravená data s rozdělením na trénovací a validační sady (sl. 3 a 4), které se skládají z objektů vytažených maskou ze vstupních obrázků (sl. 2)

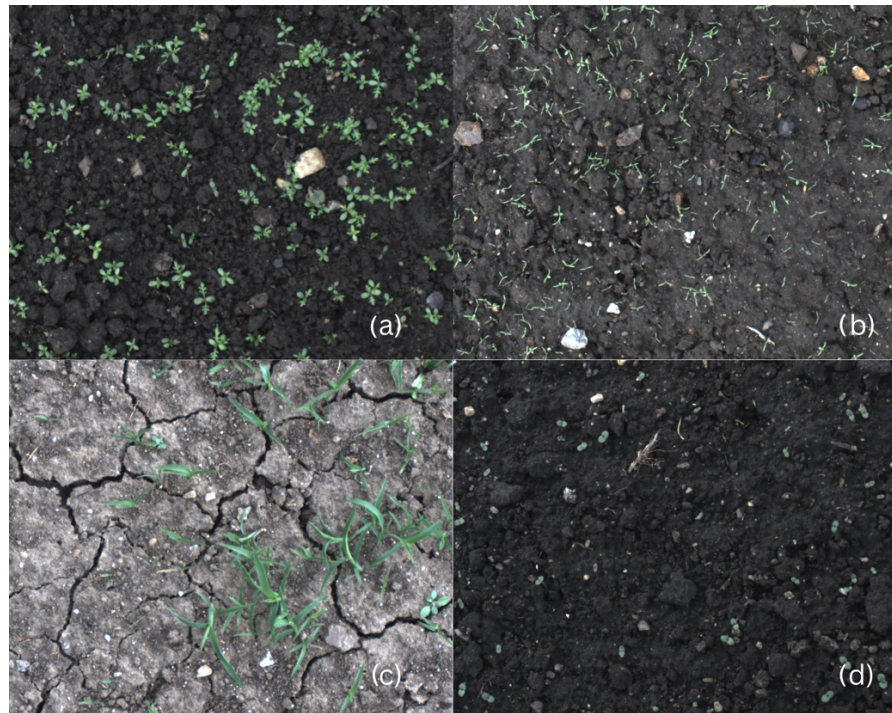
| Data pro třídu | Vstupní obrázky: | Trénovací sada: | Validační sada: |
|---------------------|------------------|-----------------|-----------------|
| ježatka kuří noha | 262 | 2930 | 722 |
| rozrazil perský | 274 | 11668 | 2755 |
| heřmánkovec nevonný | 145 | 11087 | 3142 |
| violka rolní | 287 | 3525 | 719 |
| mák vlčí | 162 | 3614 | 895 |

Pro rostlinu „ježatka kuří noha“ bylo k dispozici 262 vstupních snímků. Bylo z nich pomocí anotace vystřiženo a rozděleno na trénovací a validační sadu 2930 a 722 rostlin, respektive tohoto druhu plevelu. Podobně bylo pro „rozrazil perský“ k dispozici 274 vstupních snímků. Objekty z těchto snímků byly rozděleny na trénovací a validační množiny, které obsahovaly 11668 a 2755 snímků. Pro rostlinu „heřmánkovec nevonný“ bylo k dispozici 145 vstupních snímků, rostliny z nich byly rozděleny do trénovacích a validačních sad sestávajících z 11087 a 3142 snímků. Pro rostlinu „violka rolní“ bylo k dispozici 287 vstupních snímků, plevelné rostliny na nich byly rozděleny na trénovací a validační množiny sestávající z 3525, resp. 719 snímků. Pro rostlinu „mák vlčí“ bylo k dispozici 162 vstupních snímků a objekty byly rozděleny na trénovací a validační množiny sestávající z 3614 a 895 snímků.

Množství dat shromážděných v rámci tohoto projektu úzce souvisí s rychlostí růstu a hustotou různých druhů plevelů na zasetých plochách (Obr. 8). Heřmánkovec nevonný se vyznačuje vysokou hustotou rostlin, což vede ke kumulaci většího množství obrazových dat. Naproti tomu mák vlčí je relativně malý a může být obtížné jej odlišit od jiných menších plevelů. Ježatka kuří noha má ze všech druhů největší a nejhustší porosty, což ztěžuje identifikaci morfologických rozdílů mezi jednotlivými rostlinami. A konečně rozrazil perský je v počátečním stadiu růstu a může se podobat výhonkům jiných druhů, například heřmánkovce nevonného. Hustota a rychlost růstu různých druhů plevelů hraje významnou roli v množství shromážděných údajů a bylo třeba najít rovnováhu mezi sběrem dostatečného

množství údajů pro zajištění přesné identifikace pro každý druh a zabráněním přetížení systému příliš velkým množstvím dat.

Obr. 8 Ukázka snímků různých druhů plevelů. Heřmánkovec nevonný (a) vykazuje vysokou hustotu rostlin, mák vlčí (b) je menšího rozměru, a proto je těžko odlišitelný od menších porostů ostatních druhů plevelů, ježatka kuří noha (c) má největší porosty a vysokou hustotu rostlin, což ztěžuje zjištění morfologických rozdílů jednotlivých porostů, rozrazil perský (d) je v počátečním stadiu růstu a lze si ho splést s klíčky například heřmánkovce nevonného.



6 Výsledky trénování VGG16

Při pohledu na ztrátovou funkci (Tab. 2) můžeme pozorovat, že v průběhu trénování klesá, což znamená, že model zlepšuje svou schopnost předpovídat správné označení třídy pro každý objekt. Jak se model trénuje, stále lépe rozpoznává vzory v obrazových prvcích, které odpovídají určitým třídám objektů, což vede k přesnějším předpovědím.

Tab. 2 Historie učení modelu. Epoch je cyklus trénování, „loss“ – hodnoty ztrátové funkce při trénování, „acc“ – přesnost modelu při trénování, „val_loss“ – hodnoty ztrátové funkce při validaci, „val_acc“ – přesnost modelu při validaci

| epoch | „loss“ | „acc“ | „val_loss“ | „val_acc“ |
|-------|--------|-------|------------|-----------|
| 0 | 0.371 | 0.866 | 0.272 | 0.895 |
| 1 | 0.271 | 0.900 | 0.250 | 0.908 |
| 5 | 0.219 | 0.917 | 0.245 | 0.903 |
| 10 | 0.192 | 0.928 | 0.214 | 0.922 |
| 15 | 0.173 | 0.935 | 0.193 | 0.925 |
| 20 | 0.160 | 0.942 | 0.176 | 0.934 |
| 25 | 0.149 | 0.944 | 0.207 | 0.926 |
| 30 | 0.144 | 0.945 | 0.193 | 0.932 |
| 40 | 0.126 | 0.952 | 0.182 | 0.932 |
| 50 | 0.116 | 0.955 | 0.179 | 0.938 |

Během fáze trénování se model iterativně aktualizuje tak, aby se minimalizovala ztrátová funkce. Jak se model stává přesnějším v předpovídání správných štítků na trénovacích datech, zvyšuje se tréninková přesnost.

V projektu začínala tréninková přesnost modelu pro detekci a monitorování plevelů na hodnotě přibližně 86 % a v průběhu epoch se neustále zvyšovala. V poslední epoše byla tréninková přesnost přibližně 95 % (Tab. 2). To naznačuje, že model je schopen předpovídat správné značky na trénovacích datech s vysokou mírou přesnosti.

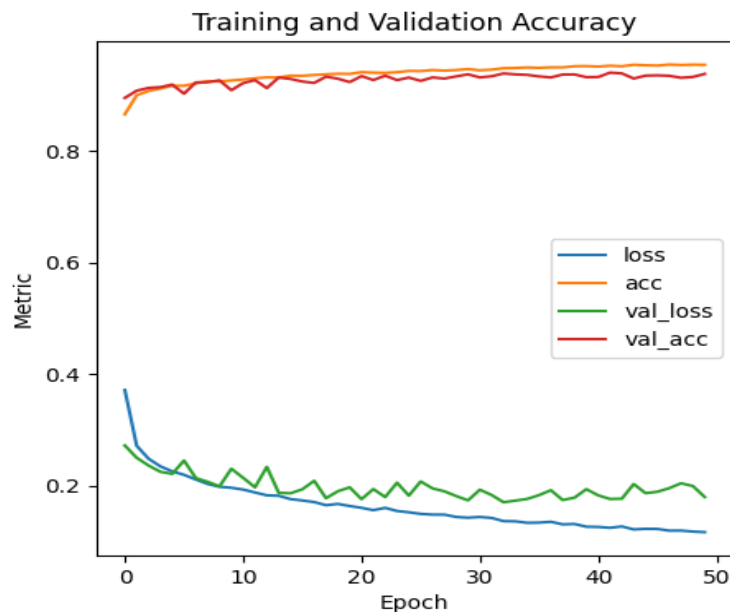
Je však důležité poznamenat, že tréninková přesnost nemusí vždy odpovídat výkonu modelu na nezobrazených datech. K přehnanému přizpůsobení může dojít, když se model příliš specializuje na tréninková data a na nových datech podává špatný výkon. Proto je velmi důležité vyhodnotit model na samostatné sadě validačních dat, aby bylo zajištěno, že model dobře zobecňuje a nezapamatovává si pouze trénovací data.

Hodnoty „val_loss“ (Tab. 2) představují ztrátu na validační datové sadě během procesu trénování. Účelem validační sady je vyhodnotit výkonnost modelu na neviděných datech a odhalit nadměrné přizpůsobení, k němuž dochází, když model funguje dobře na trénovacích datech, ale špatně na nových datech.

Během učení se validační ztráta snižuje, což naznačuje, že se zlepšuje schopnost modelu zobecňovat na nová data. Konkrétně se validační ztráta snižuje z počáteční hodnoty přibližně 0,27 na konečnou hodnotu přibližně 0,18, což naznačuje zlepšení výkonu modelu.

Hodnoty „val_acc“ v historii modelu představují přesnost validace, což je podíl správně předpovězených druhů plevelů na validačních datech. Přesnost validace je důležitou metrikou při hodnocení výkonnosti modelu strojového učení, protože měří, jak dobře je model schopen zobecnit se na nová, dosud neviděná data. Validační přesnost se během procesu trénování neustále zvyšovala a po 30 epochách dosáhla vrcholu kolem 93 %. To znamená, že model dokázal přesně předpovědět označení přibližně 93 % validačních obrázků.

Obr. 9 Historie trénování



Historie modelu (Tab. 2) a graf trénování (Obr. 9) ukazují vývoj výkonnosti modelu strojového učení v průběhu jeho trénování. Z toho vidíme, že ztráta a přesnost modelu se v průběhu všech epoch trénování pomalu zlepšují, což naznačuje, že se model učí a zlepšuje svou schopnost předpovídat. Ztráta při trénování začíná na hodnotě 0,37 a po 50 epochách se sníží na 0,11, zatímco přesnost při trénování začíná na hodnotě 0,86 a po 50 epochách se zvýší na 0,95. To naznačuje, že model je při svém trénování stále přesnější ve svých předpovědích.

V historii jsou uvedeny také metriky validační ztráty a přesnosti, které pomáhají vyhodnotit schopnost modelu zobecnit se na nová data. Na grafu (Obr. 9) vidíme, že validační ztráta a přesnost obecně sledují stejný trend jako metriky trénování, ale s určitými výkyvy. Validační ztráta začíná na hodnotě 0,27 a po 50 epochách se sníží na 0,10, zatímco validační

přesnost začíná na hodnotě 0,89 a po 50 epochách se zvýší na 0,94 (Tab. 2). To naznačuje, že se model příliš nepřizpůsobuje trénovacím datům a je schopen dobře zobecňovat na nová data. V průběhu procesu trénování dochází k určitým výkyvům metrik, jež mohou být způsobeny stochastickou povahou optimalizačního algoritmu použitého k trénování modelu nebo výkyvy v samotných trénovacích datech. Celkový trend metrik však naznačuje, že se model v průběhu času zlepšuje.

7 Výsledky predikce

Hlavním výsledkem trénování neuronové sítě je identifikace a kategorizace plevelů. Na snímcích jsou zobrazeny konkrétní druhy plevelů, včetně: heřmánkovec nevonný, ježatka kuří noha, mák vlčí, rozrazil perský a violka rolní. Uvedené výsledky v Tab. 3 pocházejí z předpovědí provedených modelem VGG16 pro jednotlivé typy plevelů, procenta představují přesnost modelu v jeho klasifikaci.

Tab. 3 Výsledky predikce pro každý druh plevelů

| Snímek třídy: | hermankovec nevonny (%) | ježatka kuri noha (%) | mak vlci (%) | rozrazil persky (%) | violka rolni (%) |
|---------------------|-------------------------|-----------------------|--------------|---------------------|------------------|
| violka rolní | 2,5 | 20 | 27,5 | 18,5 | 32 |
| mák vlčí | 4,5 | 22,5 | 49 | 0,5 | 29 |
| ježatka kuří noha | 3 | 69 | 17 | 3 | 8 |
| rozrazil perský | 5,5 | 23 | 8,5 | 46,5 | 23,5 |
| heřmánkovec nevonný | 39 | 5 | 19 | 3 | 35 |

Nejvyšší přesnost klasifikace vykazuje druh ježatka kuří noha, který dosahuje 69 % spolehlivosti (Tab. 3). To naznačuje, že model je při rozlišování tohoto konkrétního druhu plevele nejschopnější. Jak je patrné ze snímků, tato rostlina je mezi studovanými druhy největší a má výrazné morfologické charakteristiky, díky nimž ji neuronová síť snáze identifikuje. Přesnost je však poněkud snížena, pokud tyto rostliny rostou hustě.

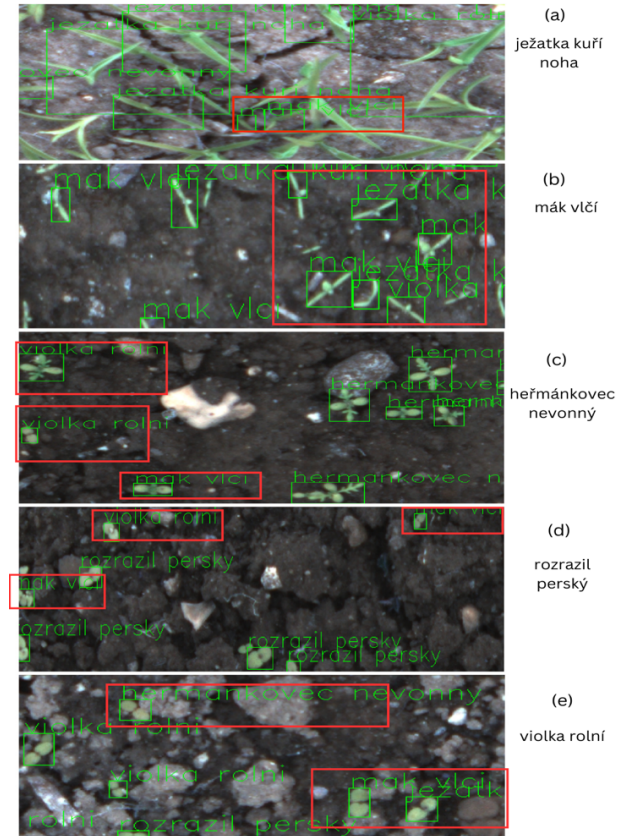
Rozrazil perský a mák vlčí vykazují střední přesnost klasifikace, přičemž skóre spolehlivosti činí 46,5 %, resp. 49 %. To naznačuje, že je model přiměřeně schopný detekovat tyto druhy plevelů, i když stále existuje potenciál pro zlepšení. Tyto dvě rostliny vykazují v určitých fázích růstu podobnou strukturu listů, což občas vede k nepřesným předpovědím neuronové sítě.

Heřmánkovec nevonný a violka rolní mají nejnižší přesnost klasifikace s příslušnými skóre spolehlivosti 39 % a 32 %. To znamená, že model naráží na potíže při přesné identifikaci těchto plevelných druhů, a to především kvůli jejich částečné podobnosti s jinými rostlinami, což vede k méně přesným předpovědím.

Hustota a velikost rostlin hrají významnou roli při tréninku a tato omezení jsou stejně důležitá i pro správné rozpoznání. Během analýzy výsledků této studie bylo zjištěno, že velikost a překrývání objektů na snímcích mohou mít významný vliv na přesnost neuronové sítě. Malé objekty a pozadí byly ze snímků odstraněny, aby se zlepšila přesnost, ale velikost zbývajících objektů stále hrála důležitou roli.

Větší objekty měly tendenci být lépe rozpoznány a snadněji klasifikovány, protože měly jasnější rysy, ale bylo dost pravděpodobné, že se budou překrývat s jinými porosty, což způsobovalo problémy při klasifikaci rostlin.

Obr. 10 Typické chyby, které lze pozorovat u konkrétních tříd plevelů



Na snímku výše (Obr. 10) jsou ukázané typické chyby, které se mohou vyskytnout při detekci a klasifikaci konkrétních tříd plevelů. Mezi faktory, jež k těmto chybám přispívají, patří překrývání rostlin, jak je vidět u ježatky kuří nohy (Obr. 10 (a)), slabé morfologické rozdíly mezi druhy, jako je to v případě máku vlčího (Obr. 10 (b)) a kukačky, a podobnosti v raných fázích růstu u heřmánkovce nevonného (c), rozrazilu perského (d) a violky rolní (e).

Tyto problémy podtrhují význam zdokonalování neuronové sítě a používání strategií pro zlepšení jejích detekčních a klasifikačních schopností. Možné přístupy zahrnují zvýšení počtu trénovacích dat, začlenění technik rozšiřování dat a využití znalostí o doméne k optimalizaci architektury modelu a extrakce příznaků. Řešením těchto problémů lze zvýšit výkonnost modelu, což pravděpodobně povede k přesnější detekci a monitorování plevelů v zemědělských podmínkách.

Potenciální řešení pro zlepšení přesnosti klasifikace spočívá v pořizování snímků rostlin ve zralejších stádiích a ve změně perspektivy, aby se zdůraznily morfologické rozdíly, což

bohužel odporuje agronomickému účelu odstraňování plevelů v počátečních stádiích růstu. Kromě toho by použití pokročilých neuronových sítí s alternativními metodikami trénování nebo zdokonalení technik předzpracování obrazu mohlo dále zmírnit problémy spojené s detekcí a monitoringem plevelů.

8 Ekonomické posouzení

Abychom mohli porovnat náklady platform AWS a Google Colab, musíme vzít v úvahu zdroje využívané modelem a ceny jednotlivých platform. Pokud jde o zdroje, nejdůležitější ukazatele pro platformy jsou: využití paměti GPU, systémové paměti RAM, čas na epochu a celkový čas pro trénování modelu. Během testování cloudové implementace bylo zjištěno, že náš model vyžaduje 5 hodin provozu a 6,7 GB paměti a 3 GB paměti GPU.

V případě platformy AWS bylo stanoveno, že cena závisí na typu použité instance, přičemž cena za hodinu na vyžádání se pohybuje od 0,42 do 0,752 USD (Tab. 4).

Tab. 4 Ceny na instance platformy AWS (AWS Pricing Calculator, n, d.)

| Instance | GPU Paměť (GB) | Paměť RAM (GB) | Cena na vyžádání/hod |
|----------------------|----------------|----------------|----------------------|
| g4dn.xlarge | 16 | 16 | \$0.526 |
| g4dn.2xlarge | 16 | 32 | \$0.752 |
| g5g.xlarge | 16 | 8 | \$0.42 |
| g5g.2xlarge | 16 | 16 | \$0.556 |
| Průměrná cena | | | \$0.516 |

Vzhledem k tomu, že využití instance na AWS závisí na její aktuální dostupnosti, mohou se náklady pohybovat v rámci cen uvedených v Tab. 4. Proto byl pro určení průměrné ceny použit vzorec aritmetického průměru:

$$\bar{x} = \frac{1}{n} (\sum_{i=0}^n x_i), \quad (4)$$

kde n je počet typů instancí a x_i je cena na vyžádání za hodinu pro každý typ instance. Pomocí tohoto vzorce byla získána průměrná cena 0,516 USD (Tab. 4) pro platformu AWS.

Pro Google Colab jsou dvě možnosti předplatného vhodného pro naše požadavky, jsou to Colab Pay-as-you-go a Colab Pro (Tab. 5).

Tab. 5 Ceny na předplatné Google Colab (Google, n, d.)

| Plán předplatného | GPU Paměť (GB) | Paměť RAM (GB) | Cena za 100 unitů |
|-------------------|----------------|----------------|-------------------|
| Pay as you go | 15.0 | 12.7 | \$9.99 |
| Pro | 40.0 | 83.5 | \$9.99 |

Základním prvkem systému Google Colab je výpočetní jednotka (unit), která se skládá z prvků pro zpracování dat. Odhady nákladů na plány předplatného vycházejí z jejich kvality a dostupnosti. Vzhledem k tomu, že je náš projekt relativně malý a není příliš závislý

na pokročilých funkcích souvisejících s využitím paměti a GPU, byl pro nás efektivnější volbou plán předplatného Pay-as-you-go.

Pro porovnání dvou použitých platforem byly spočítané celkové náklady u platformy AWS pro typ instance dle vzorce:

$$N_{AWS} = P_i t_i, \quad (5)$$

kde P_i je cena na vyžádání za hodinu a t_i je celkový čas práci.

Pak byly vypočteny průměrné náklady na instance (Tab. 6) dle vzorce aritmetického průměru pro možnost porovnání s produktem Google Colab, u něhož se výpočet nákladů liší. U produktu Pay-as-you-go záleží odhady nákladů celkově na počtu použitých „unitů“, kterých bylo spotřebováno 1,96 unitů za hodinu, a proto vzorec pro celkové náklady pro Google Colab je:

$$N_{GC} = \frac{P}{100} U_h t, \quad (6)$$

kde P je cena za 100 unitů, U_h jsou unity spotřebované za hodinu a t je celkový čas.

Tab. 6 Výpočty nákladů pro jednotlivé možnosti implementace

| Platforma | Plán předplatného | Použité GPU | Použitá paměť | Čas na 1 epochu (min.) | Celkový čas (hod.) | Celkové náklady |
|--------------|----------------------|-------------|---------------|------------------------|--------------------|-----------------|
| AWS | g4dn.xlarge | 6 | 3 | 6 | 5 | \$2.63 |
| | g4dn.2xlarge | 6 | 3 | 6 | 5 | \$3.76 |
| | g5g.xlarge | 6 | 3 | 6 | 5 | \$2.1 |
| | g5g.2xlarge | 6 | 3 | 6 | 5 | \$2.78 |
| | Průměrná cena | | | | | \$2.8175 |
| Google Colab | Pay-as-you-go | 6.3 | 3 | 6 | 5 | \$0,979 |

Co se týče ukládání dat, v případě služby Google Colab je úložiště poskytováno na Google Drive, za který nemusíme platit navíc. AWS používá pro ukládání dat S3 bucket, jehož cena se výrazně liší v závislosti na objemu dat a době ukládání. V našem případě cena za uložení 1 GB dat činila 0,023 USD. Náklady na úložiště jsou spočítané dle vzorce:

$$N_u = P_{s3} M, \quad (7)$$

kde P_{s3} jsou náklady na 1 GB paměti a M je potřebná kapacita úložiště, z čeho vychází hodnota 0,1541 USD.

Z toho lze snadno vypočítat celkové náklady pro tyto dvě možnosti softwarového řešení:

$$N_{c\ AWS} = N_{AWS} + N_u, \quad (8)$$

$$N_{cGC} = N_{GC}, \quad (9)$$

celkové náklady při použití platformy AWS pak činí 2.976 USD, u Pay-as-you-go se celkové náklady rovnají 0,979 USD.

Z toho vyplývá, že použití předplatného Pay-as-you-go společnosti Google Colab je pro tento projekt z hlediska nákladové efektivity vhodnější, s celkovými náklady 0,979 USD na základě skutečné spotřeby prostředků a jednotek za hodinu. Je však důležité poznamenat, že platforma AWS nabízí snížení cen na vyžádání, přičemž maximální sleva činí 62 %. Skutečnou výši slevy nelze předem určit, a proto jsme ji nezohlednili v našem srovnání.

9 Závěr

Model VGG16 dosáhl vysoké přesnosti při trénování, nicméně predikce ukazuje na potřebu vylepšení rozpoznávání jednotlivých druhů plevelů.

Do budoucna existuje několik oblastí, ve kterých by bylo možné tento projekt vylepšit, nebo rozšířit. Jednou z možností by bylo použití složitější architektury modelu, například YOLO nebo Faster R-CNN, což by mohlo zvýšit přesnost a rychlost detekce objektů. Další oblastí pro zlepšení by bylo použití většího a rozmanitějšího souboru dat, aby se dále zlepšil výkon modelu.

Ekonomické posouzení ukázalo, že využití předplatného Google Colab Pay-as-you-go bylo pro tento projekt nákladově nejefektivnější variantou. Poskytoval dostatečné zdroje pro trénování modelu a provoz softwarového řešení při zachování nízkých nákladů. Testována byla také platforma AWS, ale její cenový model nebyl pro tento konkrétní projekt tak výhodný. Ekonomické posouzení procesu vývoje softwaru navíc může pomoci při řízení budoucích projektů a při rozhodování o přidělování rozpočtu.

10 Seznam použitých zdrojů

- API documentation: tensorflow V2.11.0*. TensorFlow. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: https://www.tensorflow.org/api_docs
- Asad, M. H., & Bais, A. (2020). Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. *Information Processing in Agriculture*, 7(4), 535–545. <https://doi.org/10.1016/j.inpa.2019.12.002>
- AWS Documentation. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: https://docs.aws.amazon.com/?nc2=h_ql_doc_do
- AWS Pricing Calculator. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: <https://calculator.aws/#/>
- Barbedo, J. (2019). A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses. *Drones*, 3(2), 40. <https://doi.org/10.3390/drones3020040>
- Banerjee, K., Gupta, R. R., Vyas, K., & Mishra, B. (2020). Exploring alternatives to softmax function. *arXiv preprint arXiv:2011.11538*.
- Chostner, B. (2017). See & Spray: The next generation of weed control. *Resource Magazine*, 24(4), 4-5.
- Daly, D. J. (n.d.). *Economics 2: EC2*. Amazon. [online] Cit.: 28 03 2023. Dostupné z: <https://aws.amazon.com/ec2/instance-types/>
- DeepAI. (2019, May 17). *Relu*. DeepAI. [online] Cit.: 28 03 2023. Dostupné z: <https://deepai.org/machine-learning-glossary-and-terms/relu>
- Esposito, M., Crimaldi, M., Cirillo, V., Sarghini, F., & Maggio, A. (2021). Drone and sensor technology for sustainable weed management: a review. *Chem. Biol. Technol. Agric.* 8, 18 <https://doi.org/10.1186/s40538-021-00217-8>
- Fernandez-Gallego, J., Buchailot, M., Aparicio Gutiérrez, N., Nieto-Taladriz, M., Araus, J., & Kefauver, S. (2019). Automatic wheat ear counting using thermal imagery. *Remote Sensing*, 11(7), 751. <https://doi.org/10.3390/rs11070751>
- Google. (n.d.). *Choose the Colab plan that's right for you*. Google colab. [online] Cit.: 28 03 2023. Dostupné z: <https://colab.research.google.com/signup>
- Google. (n.d.). *Colaboratory*. Google colab. [online] Cit.: 28 03 2023. Dostupné z: <https://research.google.com/colaboratory/faq.html>
- Gowdra, N., Sinha, R., MacDonell, S., & Yan, W. (2020). *Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting*.

- He, H., Boyd-Graber, J., Kwok, K., & Daumé III, H. (2016, June). Opponent modeling in deep reinforcement learning. In International conference on machine learning (pp. 1804-1813). PMLR.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). A review of the use of convolutional neural networks in Agriculture. *The Journal of Agricultural Science*, 156(3), 312–322. <https://doi.org/10.1017/s0021859618000436>
- Kaur, S., Kaur, R., & Chauhan, B. S. (2018). Understanding crop-weed-fertilizer-water interactions and their implications for weed management in Agricultural Systems. *Crop Protection*, 103, 65–72. <https://doi.org/10.1016/j.cropro.2017.09.011>
- Khan, M. A., Ahmad, S., & Raza, A. (2019). Integrated Weed Management for Agronomic Crops. *Agronomic Crops*, 257–281. https://doi.org/10.1007/978-981-32-9783-8_14
- Koidl, K. (2013). Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional Neural Networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Matplotlib 3.7.1 documentation. (n.d.). [online] Získáno: Cit.: 28 03 2023. Dostupné z: <https://matplotlib.org/stable/index.html#learning-resources>
- NumPy documentation. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: <https://numpy.org/doc/>
- OpenCV. (n.d.). *OpenCV-python tutorials*. [online] Cit.: 28 03 2023. Dostupné z: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- Pulkrábek, J., Urban, J., Dvořák, P., & Bečková, L. (2019). *Vliv kypření půdy za vegetace na erozi půdy a produkci cukrové řepy*. *Listy Cukrovarnické a Reparské*, 135.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Rigon, C. A., Gaines, T. A., Küpper, A., & Dayan, F. E. (2020). Metabolism-based herbicide resistance, the major threat among the non-target site resistance mechanisms. *Outlooks on Pest Management*, 31(4), 162–168. https://doi.org/10.1564/v31_aug_04
- Sevi, M., & AYDIN, I. (2020). Covid-19 detection using Deep Learning Methods. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*. <https://doi.org/10.1109/icdabi51230.2020.9325626>

- Shadeed, G. A., Tawfeeq, M. A., & Mahmoud, S. M. (2020, June). Automatic Medical Images Segmentation Based on Deep Learning Networks. In IOP Conference Series: Materials Science and Engineering (Vol. 870, No. 1, p. 012117). IOP Publishing.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Subeesh, A., Bhole, S., Singh, K., Chandel, N. S., Rajwade, Y. A., Rao, K. V. R., Kumar, S. P., & Jat, D. (2022). Deep convolutional neural network models for weed detection in polyhouse grown bell peppers. *Artificial Intelligence in Agriculture*, 6, 47–54. <https://doi.org/10.1016/j.aiia.2022.01.002>
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2016.308>
- Tato, A., & Nkambou, R. (2018). Improving adam optimizer.
- Unidecode 1.3.6. Project description*. PyPI. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: <https://pypi.org/project/Unidecode/>
- User guide*. User Guide – pandas 1.5.3 documentation. (n.d.). [online] Cit.: 28 03 2023. Dostupné z: https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- Xin, M., & Wang, Y. (2019). Research on image classification model based on Deep Convolution Neural Network. *EURASIP Journal on Image and Video Processing*, 2019(1). <https://doi.org/10.1186/s13640-019-0417-8>