

Replikácia článku

A New Deep Learning Model for the Classification of Poisonous and Edible Mushrooms Based on Improved AlexNet Convolutional Neural Network

Ján Schmiedbauer, Valér Tóth

I. SUMARIZÁCIA ČLÁNKU A PROBLEMATIKY

NÁŠ článok [1] je od autorov Wacharaphol Ketwongsa [2], Sophon Boonlue a Urachart Kokaew [3], ktorí sú všetci členovia univerzity Khon Kaen [4], ktorá sa nachádza v Thajske. Hlavná problematika článku sa týka rozpoznávania jedlých a jedovatých húb pomocou CNN a tiež aj RCNN. Článok bol čiastočne motivovaný každoročnými smrťami, ktoré sa uskutočňujú ako dôsledok konzumácie jedovatých húb v Thajske a autori dôjedú, že výskum klasifikačných technológií môže pomôcť predísť takýmto incidentom. Tiež sú spomenuté rôzne tradičné spôsoby rozpoznávania jedovatých húb v Thajske a ich relatívna neefektivnosť. V článku "Toxíny v sebe nemajú iba muchotrávky: Lekárka radí, čo robiť, ak sa otrávite hubami" [5] od publikácie Nový čas je zverejnená štatistika, že približne 100 až 120 Slovákov je ročne hospitalizovaných kvôli otrave jedovatými hubami, čiže táto problematika je relevantná aj pre nás na Slovensku.

Hlavnou tému článku je návrh nového CNN modelu založeného na báze známeho modelu AlexNet. Tento nový model bol porovnaný s troma predtrénovanými architektúrami : GoogleNet (InceptionV1), ResNet50, a tiež aj so samotným AlexNet. Súčasťou porovnania bola tiež efektívnosť CNN ako súčasť RCNN. Všetky modely boli pomocou transfer learning metód pretrénované a testované na datasete 2000 obrázkov rôznych húb a bolo zistené, že navrhnutý model sice nepresahoval presnosť porovnávaných modelov, ale vedel získať podobné (vysoké) hodnoty za menší čas. Pre CNN 98.50% a pre RCNN 95.50%.

II. DATASET

PÔVODNÝ dataset obsahoval 623 obrázkov 5 rôznych druhov jedlých a jedovatých húb s veľkosťou 227 X 227 X 3 pixelov. Súčasťou jedlých druhov boli :

- A *Amanita citrina* (Muchotrávka citronožltá)
- B *Russula delica* (Plávka belavá)
- C *Phaeogyroporus portentosus* (Phlebopus marginatus)
- A jedovatých druhov :
- D *Inocybe rimosa* (Vláknica kuželovitá)
- E *Amanita phalloides* (Muchotrávka zelená)

TABLE I: Početnosť v pôvodnom datasete

Jedlé		Jedovaté		
Muchotrávka citronožltá	Plávka belavá	Phlebopus marginatus	Vláknica kuželovitá	Muchotrávka zelená
248	88	155	76	56

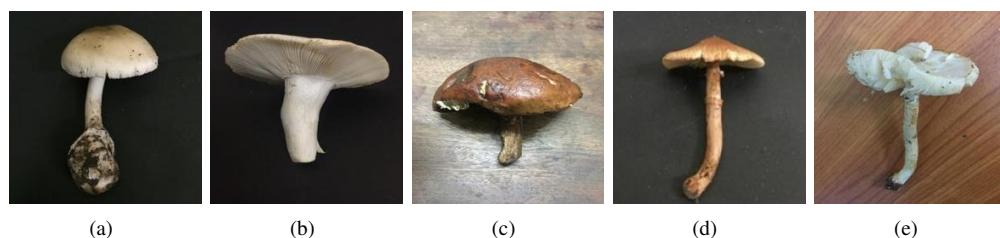


Fig. 1: (a) Muchotrávka citronožltá (b) Plávka belavá (c) Phlebopus marginatus (d) Vláknica kuželovitá (e) Muchotrávka zelená

Hlboké učenie si vyžaduje veľké množstvo trénovacích dát, z tohto dôvodu bol dataset augmentovaný (metódami nespomenutými v článku) na výsledný počet 2000 obrázkov (1473 jedlých a 527 jedovatých).

1) Poznámka o datasete: Muchotrávka citronožltá je sice formálne len nejedlá huba, ale je niekedy "preventívne" klasifikovaná ako jedovatá z dôvodu jej podobnosti k Muchotrávke zelenej. Obe huby sa nachádzajú v použitom datasete, čo zvýrazňuje schopnosť neurónových sietí rozpoznať aj predmety, ktoré sú si podobné.



Fig. 2: (a) Autor fotografia Igor Hlavatý [6], Malé karpaty (b) Miroslav Valent, Levočské vrchy [7]

III. POPIS POUŽITÝCH MODELOV

A. AlexNet

Hlboká konvolučná neurónová sieť, ktorá získala pozornosť po víťazstve v súťaži ILSVRC 2012 [9]. Architektúra pozostáva z ôsmich vrstiev, z toho päť konvolučných a tri plne prepojené. Využíva nelineárnu aktivačnú funkciu ReLU, max-pooling pre zníženie priestorovej dimenzie a dropout na redukciu pretrénovania. Pôvodný tréning bol rozdelený medzi dve GPU, čo umožnilo efektívne spracovanie rozsiahlych dát.

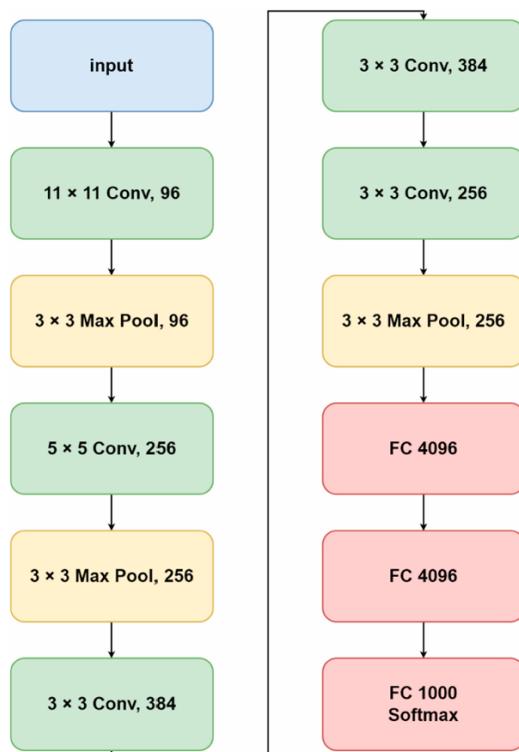


Fig. 3: Architektúra AlexNet [1]

B. GoogleNet (InceptionV1)

Známa aj ako Inception v1, je hlboká konvolučná neurónová sieť navrhnutá s cieľom znížiť počet parametrov a výpočtov bez straty presnosti. Jej kľúčovou súčasťou je tzv. *Inception modul*, ktorý umožňuje paralelné spracovanie obrazu rôznymi veľkosťami filtrov (1×1 , 3×3 , 5×5) a poolingom, čím sieť efektívne zachytáva viaceru mieru informácie.

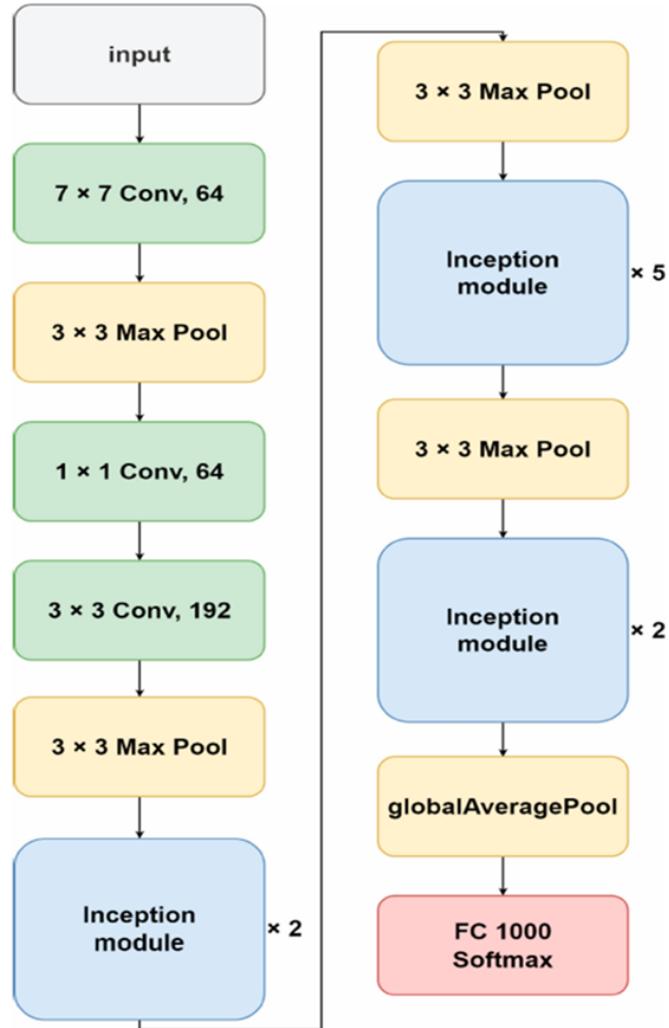


Fig. 4: Architektúra GoogleNet [1]

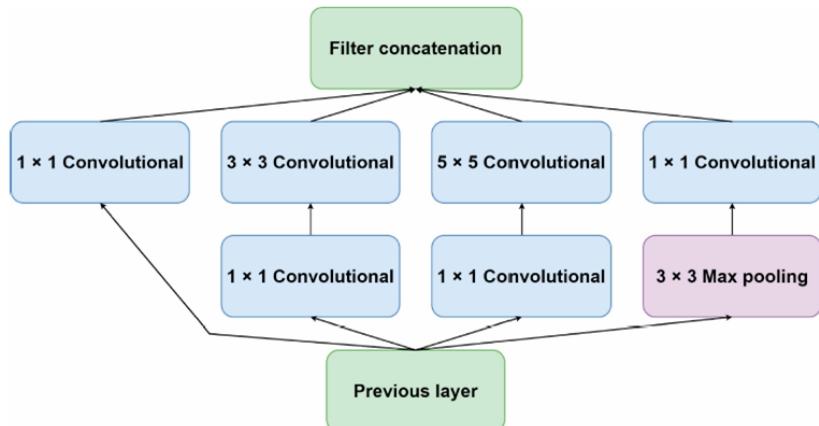


Fig. 5: Architektúra incepčného modulu [1]

C. ResNet-50

Hlboká konvolučná neurónová sieť, ktorá využíva architektúru zvanú *residual learning*. Architektúra ResNet-50 obsahuje 50 vrstiev a skladá sa z opakujúcich sa *bottleneck* blokov. Každý blok má tri vrstvy: 1×1 , 3×3 a opäť 1×1 konvolúciu, pričom prvá a posledná slúžia na redukciu a následné zväčšenie dimenzie. Použitím týchto blokov spolu s reziduálnymi spojeniami sa minimalizuje problém miziaceho gradientu a zlepšuje sa generalizácia modelu.

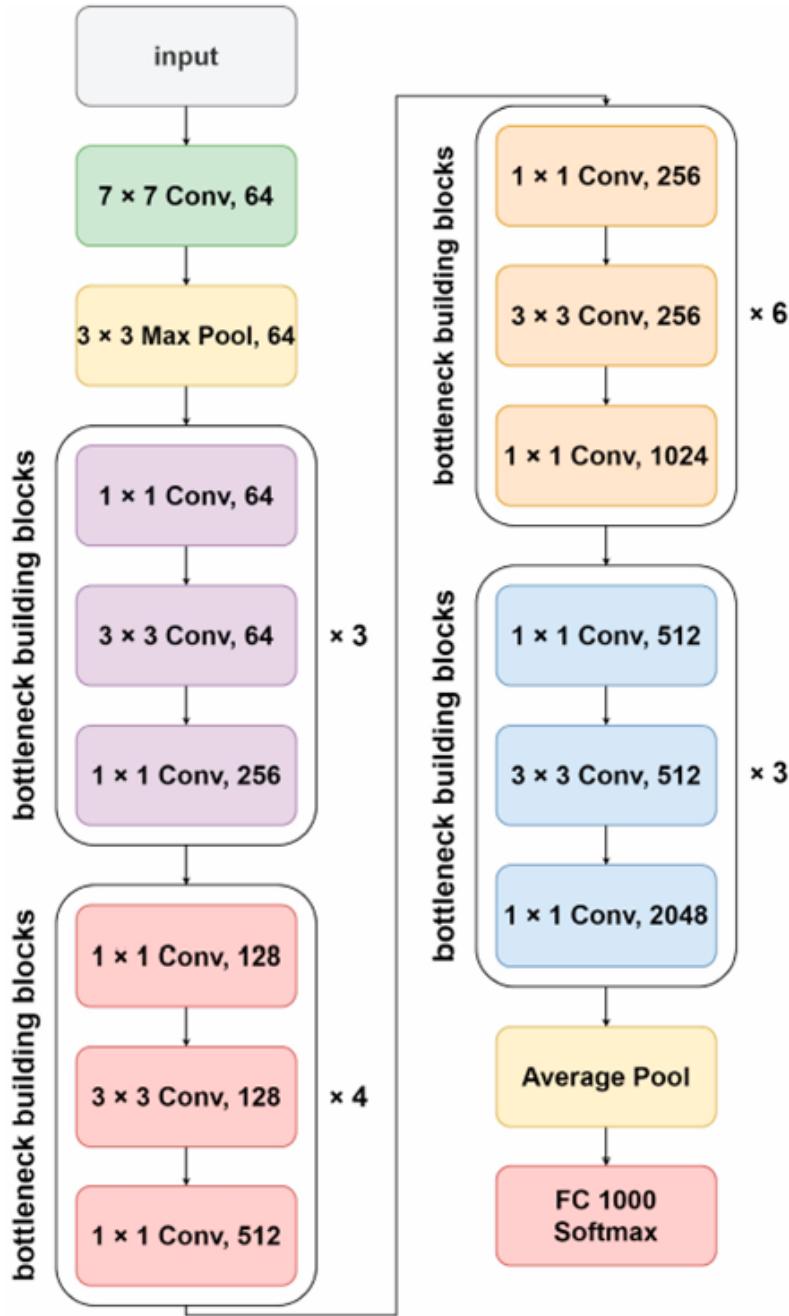


Fig. 6: Architektúra ResNet-50 [1]

D. Propozovaný model

Nový model vytvorený na báze AlexNet-u, 4. a 5. konvolučná vstava bola odstránená a nahradená incepčným modulom z GoogleNet čo by malo teoreticky zvýšiť jeho výkon.

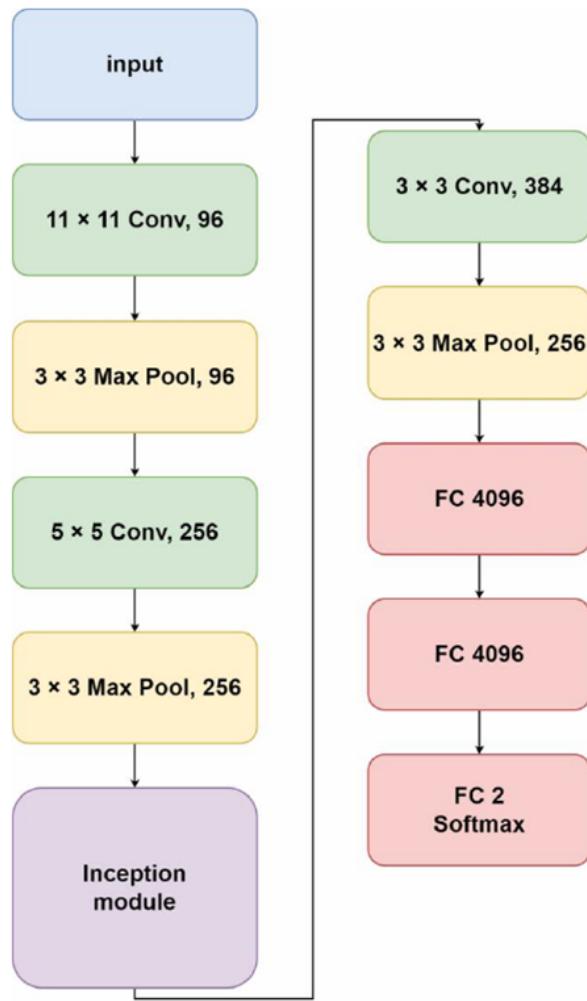


Fig. 7: Architektúra propozovaného modelu [1]

IV. METODOLÓGIA

A. CNN

Každý z modelov bol upravený, aby sa na nich mohlo uskutočniť pretrénovanie (transfer learning). Dataset 2000 obrázkov bol použitý dvoma spôsobmi :

- A Použitý metódou k-fold cross validation na získanie presnej hodnoty priemernej trénovacej presnosti.
- B Vytvorenie trénovacej a testovacej vzorky v pomere 9:1, tento train-test set bol rovnaký pre všetky CNN a bol použitý na získanie výsledných metrík modelov.

1) k-fold cross validation: Princíp fungovania validačnej metódy je nasledovný :

- 1) Dataset je rozdelený na k počet skupiniek (folds)
- 2) Jedna z k skupiniek je vybraná ako test set a zvyšok (k - 1) je použitý ako train set
- 3) Po trénovaní a testovaní modelu získame metriku, ktorá nás zaujíma pre iteráciu k
- 4) Proces sa opakuje dokiaľ sa všetky k skupinky nevystriedali byť test set-om
- 5) Výsledkom metódy je priemer metriky všetkých k iterácií

Táto metóda slúži na to, aby výsledná metrika, ktorú sa snažíme nájsť, nebola nijak ovplyvnená len variáciou v test set-e. V našom prípade bol dataset rozdelený na 10 k skupiniek (rôzne pre každý model), každá s počtom 200 obrázkov. Nami hľadaná metrika bola priemerná testovacia presnosť, kde pri trénovaní slúžil k-tý test set ako validation set. Každý model bol teda trénovaný 10-krát na rôznych variáciách train a validation set-u na získanie výslednej priemernej trénovacej presnosti.

B. RCNN

Každý z modelov bol upravený na to, aby mohol byť použitý ako súčasť RCNN. Pôvodný súbor dataset-u sice obsahoval priečinok rcnn s obrázkami pretvarovanými na veľkosť 300 X 300 X 3 pixelov, ale samotné anotácie obrázkov nutné pre trénovanie RCNN neboli poskytnuté. To znamenalo, že sme museli 2000 obrázkov anotovať sami, urobili sme to manuálne pomocou *MatLab Image Labeler*. Tento dataset bol potom rozdelený na train a test set v mierke 9:1, čiže 1800 obrázkov s anotáciami a 200 bez.

1) *RCNN (Region Convolutional Neural Network)*: Tradičný CNN sice vie klasifikovať objekt, ale nijak neukazuje v ktorej časti obrázku sa nachádza. RCNN vie nielen klasifikovať objekt ale aj nakreslí takzvaný "bounding box" okolo oblasti kde sa objekt (podľa modelu) nachádza. Funguje v 4 krokoch :

- 1) Z obrázka sa extrahuje 2000 "region proposals" ktoré sa potom pretvarujú na rovnakú veľkosť a použijú sa ako vstup do CNN
- 2) Pomocou CNN sú extraktované prvky pre každý region proposal
- 3) Extraktované prvky sú použité na klasifikáciu region proposals pomocou algoritmu SVM (Support Vector Machine)
- 4) Bounding box-y sú vytvorené okolo objektov

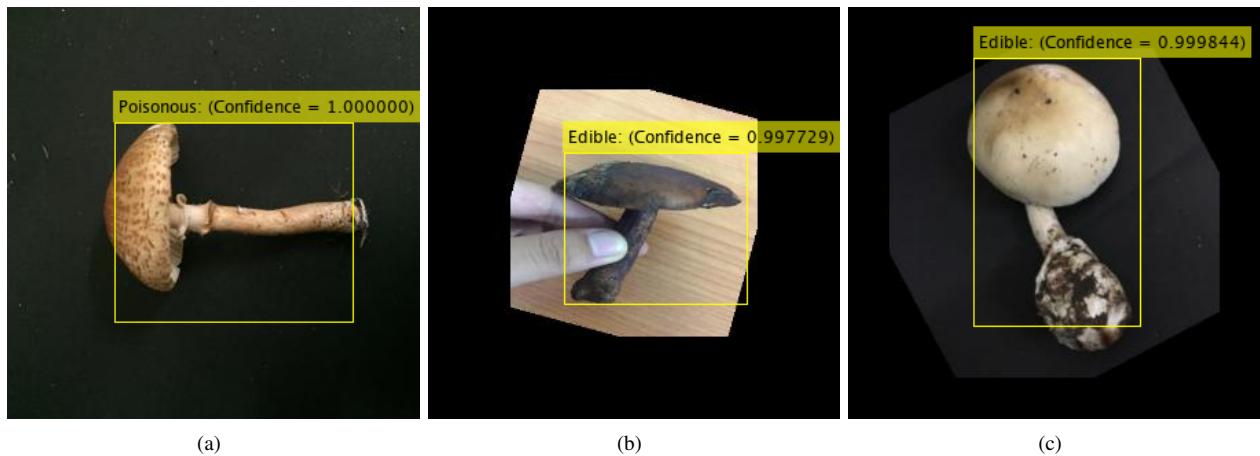


Fig. 8: Príklady klasifikácie pomocou RCNN

V. IMPLEMENTAČNÉ ROZDIELY

PÔVODNÝ článok sme sa snažili replikovať čo najlepšie sme mohli, nič sme úmyselne neurobili inak ako v pôvodnom vypracovaní, všetky implementačné rozdiely (ktoré sú nami známe) vznikli len základe nedostatku informácií z článku a technických dôvodov.

A. Vytvorenie propozovaného modelu

V článku je sice popísaná štruktúra propozovaného modelu ale není presne dané, z ktorej časti GoogleNet bol vybraný incepčný modul ktorý sa v ňom nachádza alebo ako boli vrstvy AlexNet zmenené na kompatibilitu s incepčným modulom. Mi sme propozovaný model vytvorili nasledovným spôsobom.

- 1) AlexNet bol rozdelený na dve časti odstránením 4. a 5. konvolučnej vrstvy
- 2) K prvej časti bol pripojený 2. incepčný modul 1. incepčného bloku GoogleNet
- 3) Druhá časť bola pripojená na koniec incepčného modulu
- 4) Prvé dve vrstvy druhej časti boli modifikované aby boli kompatibilné s incepčným modulom

Je možné že model v pôvodnom článku bol vytvorený iným spôsobom.

B. Neznámy optimalizátor

V článku není spomenuté aký optimalizátor bol použitý pri trénovaní modelov. Pomocou experimentácie sme došli k tomu že to bol s najväčšou pravdepodobnosťou SGDM.

C. Anotovanie dataset-u pre RCNN

V článku není nijak popísané akým spôsobom boli obrázky anotované pre trénovanie RCNN či manuálne alebo pomocou algoritmu. Mi sme dataset anotovali manuálne, čo je možno iný spôsob ako ktorí použili autori pri pôvodnom vypracovaní a môže to mať vplyv na výsledky.

D. MatLab verzia

V článku bol použitý MatLab R2021b, čo mi sme použili MatLab R2024b.

VI. DEEP DREAMS

DEEP dreams je technika vizualizácie, ktorú vyvinula spoločnosť Google na interpretáciu a pochopenie toho, čo sa neurónové siete naučili. Využíva spätnú propagáciu (backpropagation), ale namiesto úpravy váh siete upravuje samotný vstupný obrázok tak, aby maximalizoval aktiváciu určitých neurónov. Výsledkom sú "snové" obrázky s psychedelickými vzormi a opakujúcimi sa motívmi, ktoré odrážajú vnútorné reprezentácie siete.

Deep dream obrázky neboli súčasťou pôvodného článku, pridali sme ich len pre zaujímavosť.



Fig. 9: Príklad deep dream obrázka

VII. VÝSLEDKY

VŠETKY výsledky boli získané pomocou MatLab R2024b, na počítači s procesorom Intel Core i5 10400F, RAM pamäťou 16 GB a grafickou kartou NVIDIA RTX 3070. Pôvodný článok použil MatLab R2021b na počítači s Intel Core i5 12600, RAM pamäťou 12 GB a grafickou kartou NVIDIA RTX 3060. Čiže o 2 generácie a 2 rady lepší procesor ale o radu horšiu grafickú kartu.

A. CNN - Priemerná trénovacia presnosť

Pre každý model boli použité rovnaké hyper-parametre :

- 1) Optimalizátor = SGDM
- 2) Learning rate = 0.001
- 3) Počet epoch = 10
- 4) Batch size = 40

1) *AlexNet*: AlexNet dosiahol priemernú trénovaci presnosť s hodnotou 98.88 %, pričom najlepšia k iterácia (k=8) dosiahla presnosť až 99.50 %, trénovací graf tejto iterácie je zobrazený nasledovne.

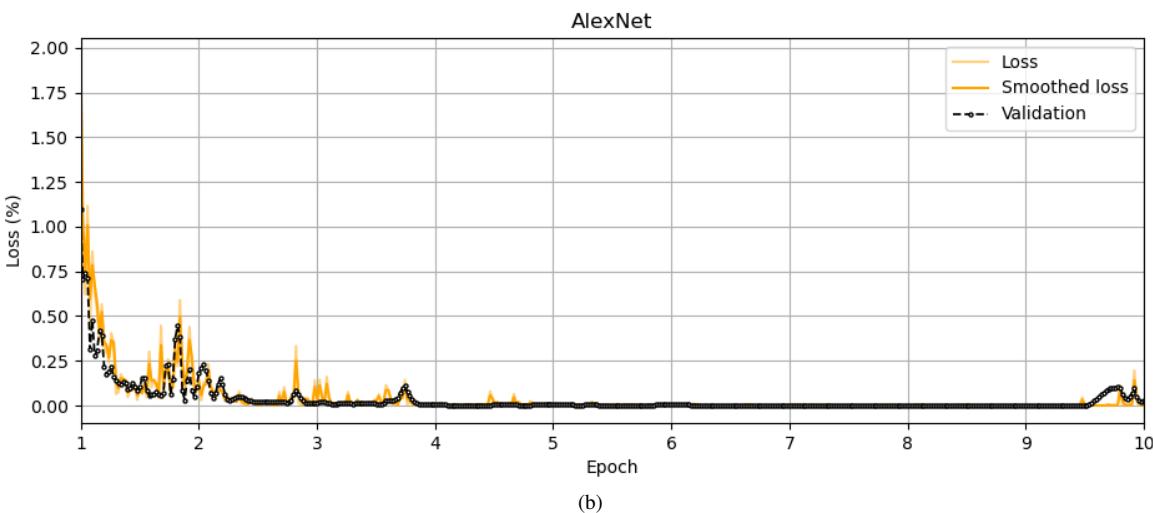
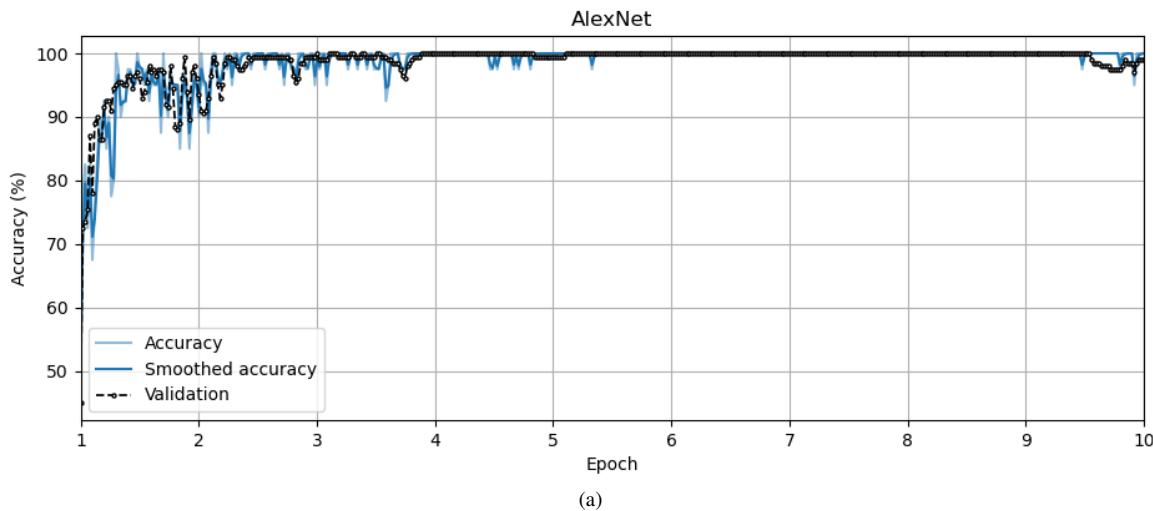


Fig. 10: (a) Trénovací graf presnosti AlexNet (b) Trénovací graf chyby AlexNet

2) *GoogleNet*: GoogleNet dosiahol priemernú trénovaci presnosť s hodnotou 99.70 %, pričom najlepšia k iterácia (k=3) dosiahla presnosť až 100 %, trénovací graf tejto iterácie je zobrazený nasledovne.

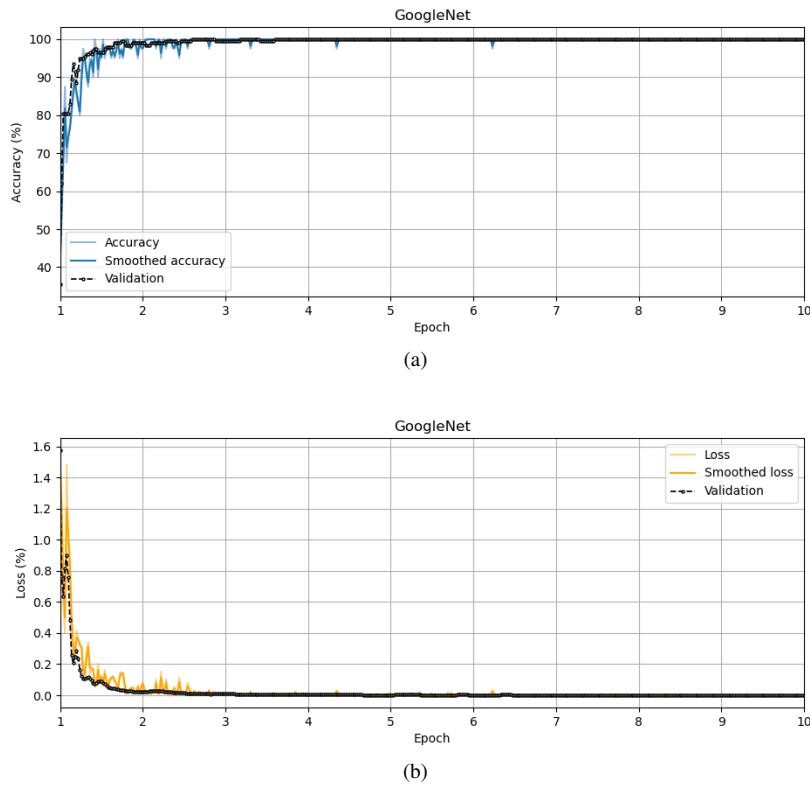


Fig. 11: (a) Trénovací graf presnosti GoogleNet (b) Trénovací graf chyby GoogleNet

3) *ResNet50*: ResNet50 dosiahol priemernú trénovaci presnosť s hodnotou 99.88 %, pričom najlepšia k iterácia ($k=1$) dosiahla presnosť až 100 %, trénovací graf tejto iterácie je zobrazený nasledovne.

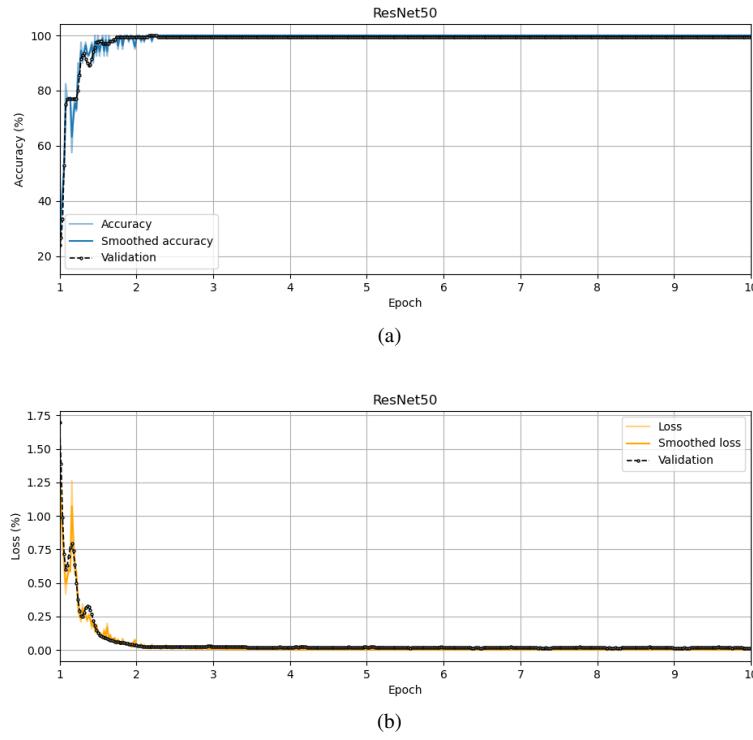


Fig. 12: (a) Trénovací graf presnosti ResNet50 (b) Trénovací graf chyby ResNet50

4) ArticleNet (Propozovaný model): Propozovaný model (nami nazvaný ArticleNet) dosiahol priemernú trénovaciu presnosť s hodnotou 92.53 %, pričom najlepšia k iterácia (k=4) dosiahla presnosť až 96.50 %, trénovací graf tejto iterácie je zobrazený nasledovne.

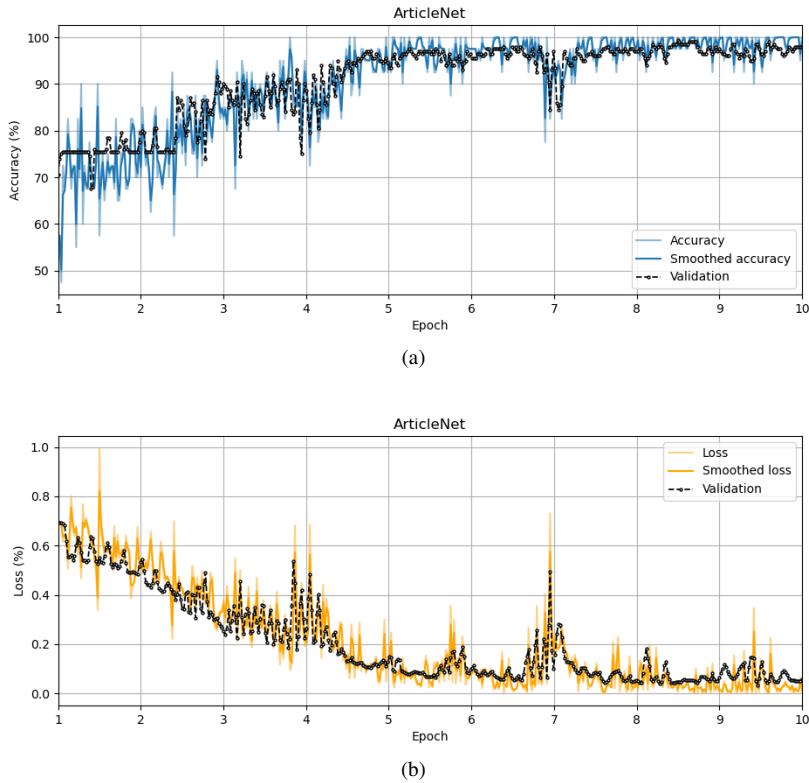


Fig. 13: (a) Trénovací graf presnosti propozovaného modelu (b) Trénovací graf chyby propozovaného modelu

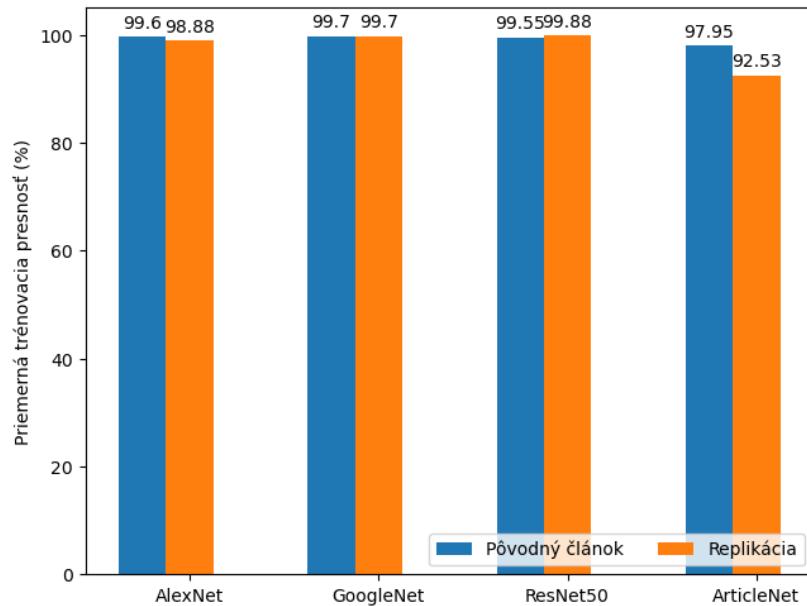


Fig. 14: Porovnanie priemernej trénovaciej presnosti z článku oproti výsledkom získaných z replikácie

S výnimkou propozovaného modelu (ArticleNet), výsledky replikácie sú v rozsahu $\pm 0.75\%$ od pôvodného článku. Rozdiel 5.42 % čo sa týka propozovaného modelu je s najväčšou pravdepodobnosťou dôsledok spomenutého implementačného rozdielu A, nedostatok informácií o jeho vytvorení.

B. CNN - Výsledné metriky

Pre každý model boli použité rovnaké hyper-parametre :

- 1) Optimalizátor = SGDM
- 2) Learning rate = 0.001
- 3) Počet epoch = 10
- 4) Batch size = 40

Nasledujúca tabuľka popisuje výsledky získané našou replikáciou, hodnoty v zátvorkách určujú o kolko sa hodnota zvýšila rešpektívne znížila oproti hodnote z pôvodného článku.

TABLE II: CNN Výsledky replikácie oproti výsledkom z článku

Model	Accuracy	Precision	Recall	F1	Time
GoogleNet	100.00 (+0.50)	100.00 (+0.65)	100.00 (0.00)	100.00 (+0.03)	1min 31s -(49.00s)
ResNet50	99.50 (0.00)	99.33 (0.00)	100.00 (0.00)	99.66 (0.00)	3min 13s (-2min 37.00s)
AlexNet	99.50 (+0.50)	99.33 (+0.72)	100.00 (0.00)	99.66 (+0.36)	45s -(57.00s)
ArticleNet	99.00 (+0.50)	99.33 (-0.06)	99.33 (+0.54)	99.33 (+0.24)	40s (-30.00s)

Jak môžeme vidieť v tabuľke, hodnoty replikácie sú všetky v rozsahu $\pm 1\%$ a v prípade F1 skóre dokonca až len $\pm 0.5\%$ od pôvodného článku čo indikuje že replikáciu sme urobili úspešne. Trénovacie časy modelov sú samozrejme nižšie z dôvodu použitia lepšej grafickej karty.

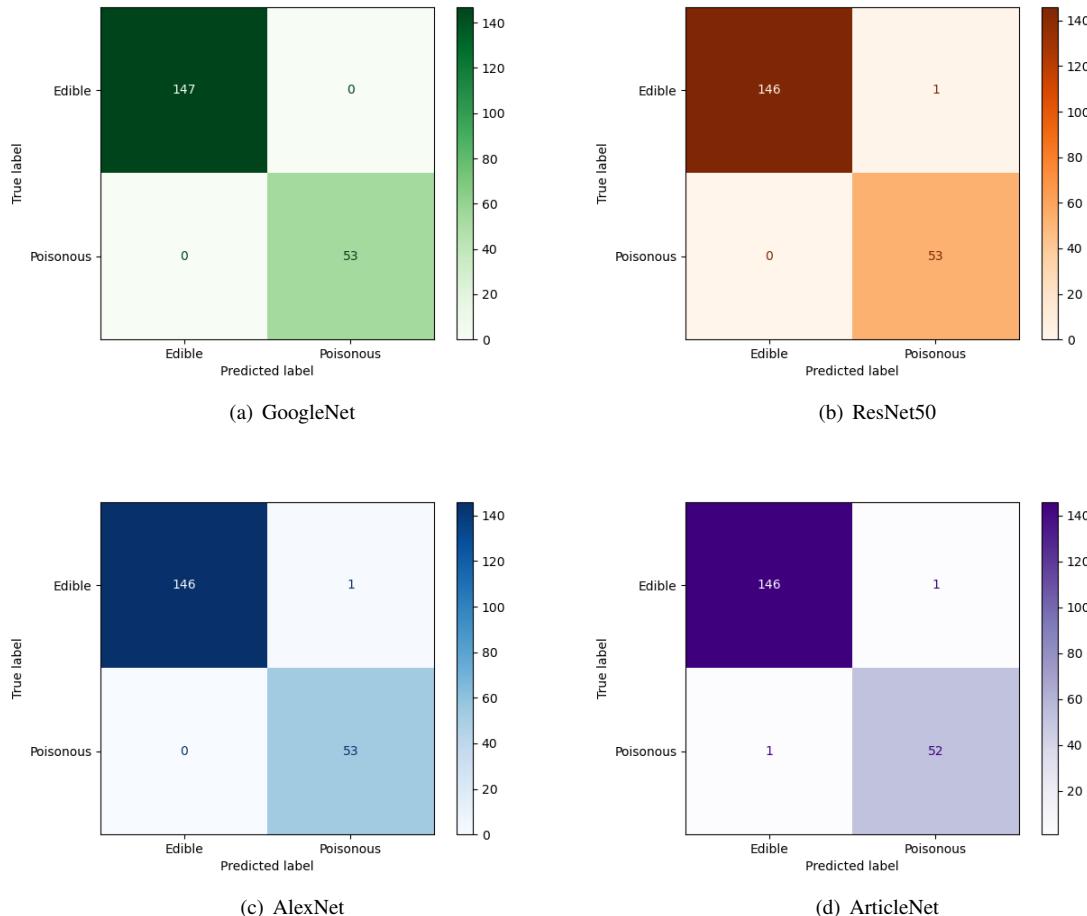


Fig. 15: CNN Confúzne matice modelov

C. CNN Deep Dreams

1) *AlexNet*: AlexNet dosahoval najvyššiu aktivačnú metriku, čiže obrázky sú veľmi detailné.

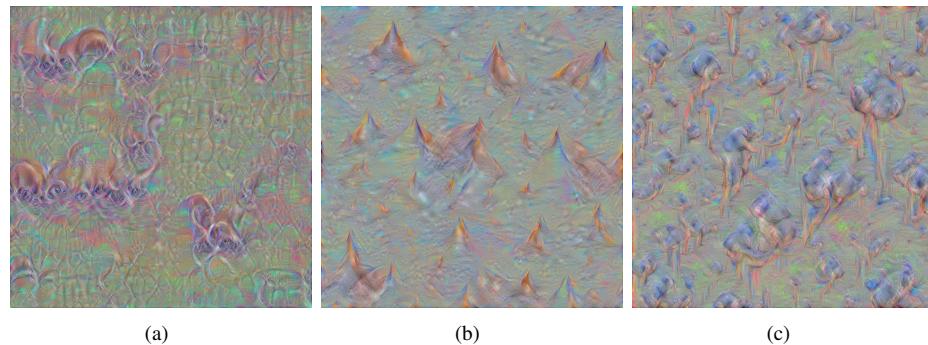


Fig. 16: Deep dream obrázky z fc7 vrstvy AlexNet-u

2) *GoogleNet*:

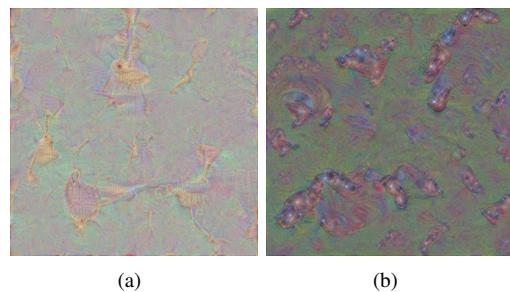


Fig. 17: Deep dream obrázky z loss3-classifier vrstvy GoogleNet-u

3) *ResNet50*:

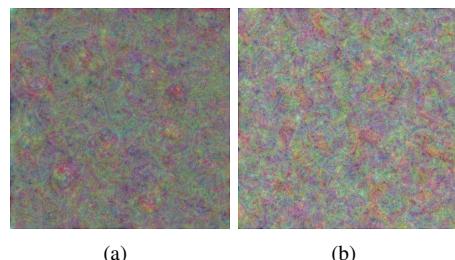


Fig. 18: Deep dream obrázky z fc1000 vrstvy ResNet50

4) *ArticleNet*:

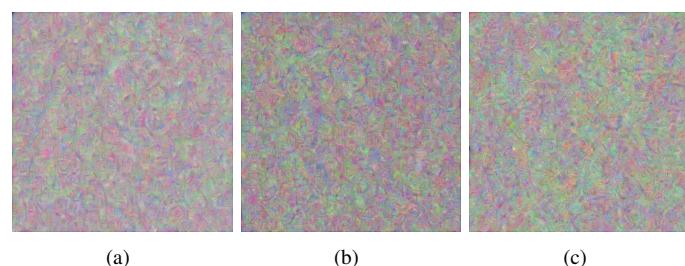


Fig. 19: Deep dream obrázky z fc7 vrstvy ArticleNet-u (propozovaného modelu)

D. RCNN

Pre každý model boli použité rovnaké hyper-parametre :

- 1) Optimalizátor = SGDM
- 2) Learning rate = 0.001
- 3) Počet epoch = 1
- 4) Batch size = 20

Nasledujúca tabuľka popisuje výsledky získané našou replikáciou, hodnoty v zátvorkách určujú o kolko sa hodnota zvýšila rešpektívne znížila oproti hodnote z pôvodného článku.

TABLE III: RCNN Výsledky replikácie oproti výsledkom z článku

Model	Accuracy	Precision	Recall	F1	Time
GoogleNet	97.50 (+1.00)	98.04 (-1.26)	98.68 (+2.76)	98.36 (+0.78)	5min 40s (-1min 48.00s)
ResNet50	96.00 (-0.50)	95.42 (-4.58)	99.32 (+4.08)	97.33 (-0.23)	11min 51s (-1min 53.00s)
ArticleNet	95.00 (-0.50)	94.77 (+0.18)	98.64 (-0.65)	96.67 (-0.22)	2min 39s (-1min 58.00s)
AlexNet	95.00 (0.00)	100.00 (+2.76)	93.87 (-2.05)	96.84 (+0.26)	2min 50s (-2min 17.00s)

Jak je vidno v tabuľke, hodnoty boli viac odlišné od pôvodného článku ako pri CNN, z dôvodu že pre RCNN sme museli dataset anotovať sami (implementačný rozdiel C). Napriek tomu v rámci presnosti a F1 skóre sa neodchyľujeme viac ako $\pm 1\%$ od hodnôt z pôvodného článku čo indikuje že replikáciu sme urobili správne aj keď bola trénovaná na inak anotovanom datasete.

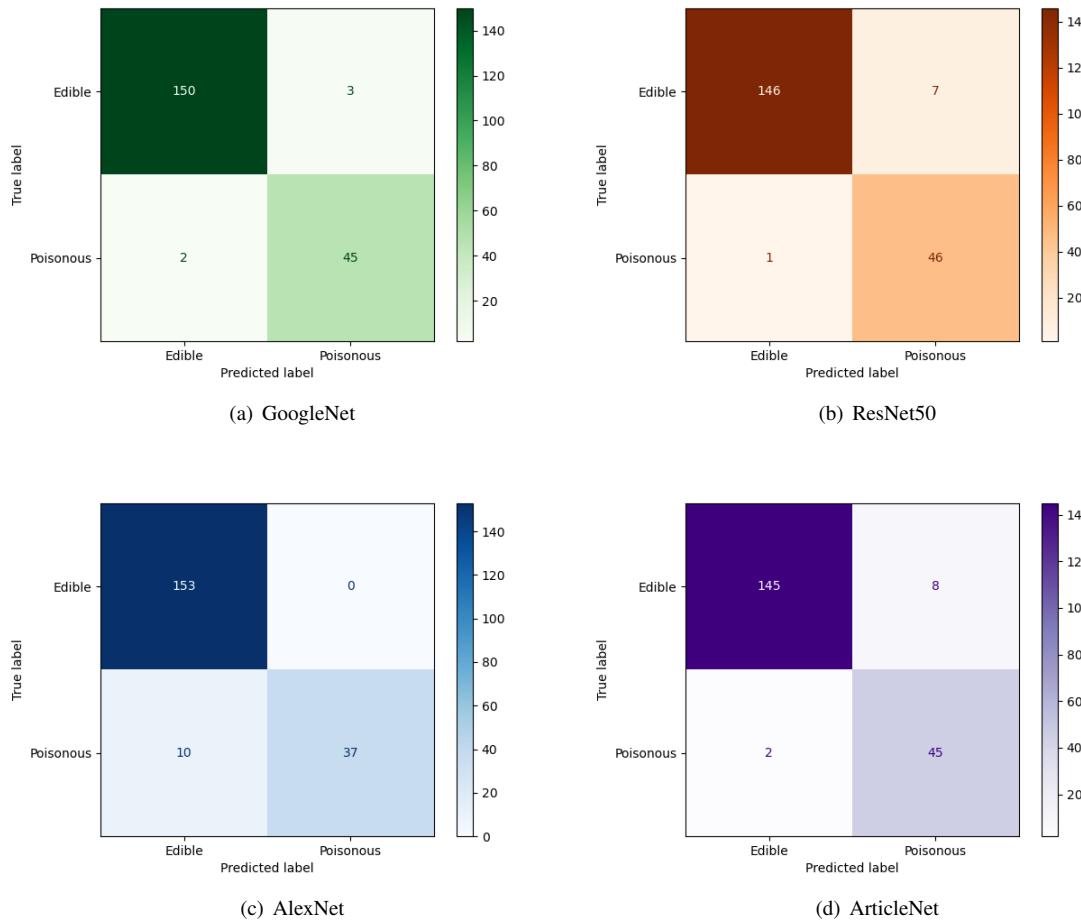


Fig. 20: RCNN Confúzne matice modelov

VIII. VYLEPŠENIE RIEŠENIA

PÔVODNÉ riešenie článku používa len aktivačnú funkciu ReLU, čo môže viesť k miznajúcim gradientom. Toto sme sa pokúsili vylepšiť vymenením funkcie ReLU za funkciu Swish. Urobili sme to tak v každej vrstve pôvodného modelu. Funkcia Swish je modifikácia Sigmoid funkcie, ktorá má výhodu oproti ReLU, že má viac vhodnejšiu deriváciu a tým pádom sa vie vyhnúť miznajúcim gradientom pri backpropagation. Toto by teoreticky malo viesť k zvýšenému výkonu pôvodného modelu.

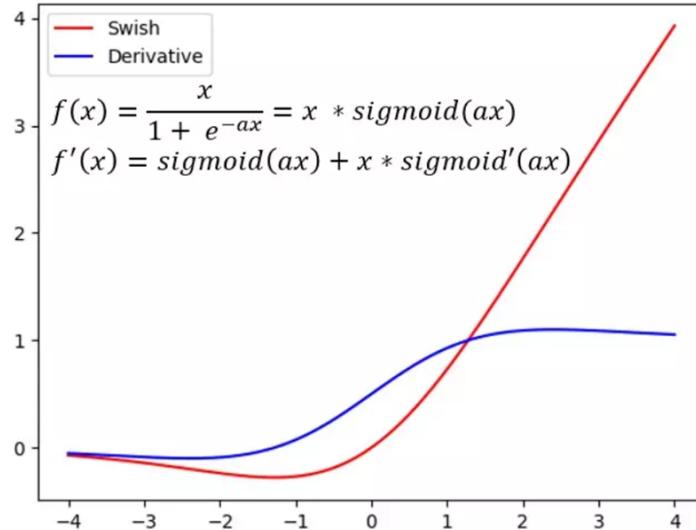


Fig. 21: Aktivačná funkcia Swish [8]

TABLE IV: Výsledky "vylepšeného" modelu (ImprovedNet) oproti pôvodnému modelu z článku (ArticleNet)

Model	Accuracy	Precision	Recall	F1	Time
ArticleNet	99.00	99.33	99.33	99.33	40s
ImprovedNet	98.00 (-1.00)	97.96 (-1.37)	99.31 (-0.02)	98.63 (-0.7)	54s (+14.00s)

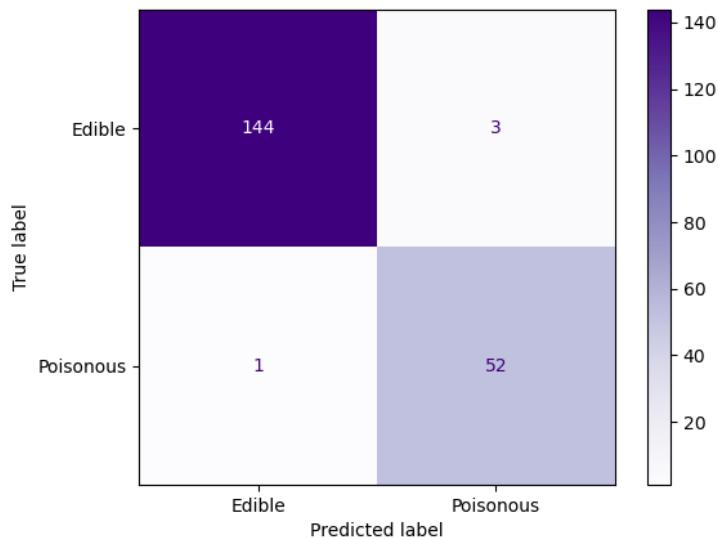


Fig. 22: Konfúzna matica ImprovedNet ("Vylepšeného" modelu)

Z výsledkov je vidno, že naša modifikácia nebola až taká prínosná ako sme si mysleli. Výsledky neindikujú žiadnu substačnú zmenu vo výkone modelu. V našom prípade boli dokonca všetky metriky horšie oproti pôvodnému modelu, s najväčšou zmenou v čase trénovalia, čo dáva zmysel pretože funkcia Swish je viac počítacovo námahová oproti ReLU. Tento výsledok nám však dáva informáciu že miznajúce gradienty niesu veľkým problémom pre nás model.

IX. ZÁVER

PROPOZOVANÝ model popísaný v článku sme implemenovali, trénovali a testovali na poskytnutom dataset-e. Jeho výsledky sme porovnali s 3 predtrénovaními architektúrami AlexNet, ResNet50, a GoogleNet (InceptionV1) ktoré boli všetky pretrénované na dataset pomocou transfer learning. Súčasťou výsledkov tiež bola aj priemerná trénovacia presnosť získaná pomocou k-fold cross validation datasetu a tiež aj efektívnosť modelov ako súčasť procesu RCNN. Napriek viaceru implemenačných rozdielov ktoré vznikli ako dôsledok nedostatku informácií z pôvodného článku, sme získali výsledky ktoré boli podobné pôvodnému článku a tým pádom sme aj došli k rovnakému záveru. Propozovaný model sice nevedel dosiahnuť väčšiu presnosť ako predtrénované architektúry ale vedel získať podobné hodnoty presnosti za najmenší čas, 99.00 % za 40s pre CNN, a 95.00 % za 2min 39s pre RCNN.

Nakoniec sme sa pokusili vylepšiť pôvodný model pomocou zmeny aktivačnej funkcie, čo nemalo žiadny vplyv na výkon modelu ale uistili sme sa tým že v modeli nedochádza k problému miznajúcich gradientov.

Tento replikáciou sme došli k tomu že architektúra propozovaného modelu je vhodná a replikovateľná, odporúčame ju na použitie v klasifikačných problémoch hlavne kde záleží na čase trénovalia modelu.

Kód, obrázky aj všetky použité dátá v našej replikácii sú dostupné na našom Github-e [10].

REFERENCES

- [1] A New Deep Learning Model for the Classification of Poisonous and Edible Mushrooms Based on Improved AlexNet Convolutional Neural Network [<https://www.mdpi.com/2076-3417/12/7/3409>]
- [2] Wacharaphol Ketwongsu [<https://orcid.org/0000-0002-5254-9869>]
- [3] Urachart Kokaew [<https://orcid.org/0000-0003-3900-6453>]
- [4] Khon kaen univerzita [<https://www.kku.ac.th/>]
- [5] Nový čas, Toxíny v sebe nemajú iba muchotrávky: Lekárka radí, čo robiť, ak sa otrávite hubami [https://www.cas.sk/clanok/2842865/toxiny-v-sebe-nemaju-iba-muchotravky-lekarka-radi-co-robit-ak-sa-otravite-hubami?orig_gallery_article=1]
- [6] Nahuby.sk, Igor Hlavatý [https://www.nahuby.sk/obrazok_detail.php?obrazok_id=720143&poradie=7&form_hash=15ae8dde5a27dfbc1b3e25a7f8234e08]
- [7] Nahuby.sk, Miroslav Valent [https://www.nahuby.sk/obrazok_detail.php?obrazok_id=805855&poradie=5&form_hash=bd6e91af4351fd68128857cc0969fbaf]
- [8] Graf aktivačnej funkcie Swish a jej derivácie [<https://www.superannotate.com/blog/activation-functions-in-neural-networks>]
- [9] ImageNet Large Scale Visual Recognition Challenge 2012 [<https://www.image-net.org/challenges/LSVRC/2012/>]
- [10] Nás Github [<https://github.com/PinkFloppa/NeuronoveSiete-Zadanie1>]