

# Klasifikacija Pasem Živali iz Oxford-IIIT Pet Dataset: Primerjava CNN Arhitektur in Vpliv Hiperparametrov

Janez Bučar  
Fakulteta za informacijske študije v  
Novem mestu  
Ljubljanska cesta 31a  
8000, Novo mesto  
gospod.bucarr@gmail.com

Ta raziskava se osredotoča na klasifikacijo pasem iz podatkovne zbirke slik Oxford-IIIT Pet Dataset, katera vsebuje skupno 7349 različnih slik in vključuje 37 pasem mačk in psov. Raziskali smo uporabo različnih CNN arhitektur kot so prilagojeni preprosti CNN, Le-Net-5, Alex-Net, ResNet-50, VGG16 in EfficientNet. Modele smo ocenili glede na njihovo kompleksnost in uspešnost klasifikacije na podlagi metrik kot so natančnost, preciznost, priklic, F1-rezultat in AIC.

Za optimizacijo modelov smo v raziskavo vključili predprocesiranje podatkov, ki zajema skaliranje velikosti slik, augmentacijo in normalizacijo posameznih slik. V eksperimentu smo pri vseh modelih uporabili kategorično krossentropijo kot funkcijo izgube ter optimizator Adam, skupaj z optimiziranimi hiperparametri ter uporabljenimi tehnikami regularizacije za boljšo generalizacijo modelov. Modeli so bili evaluirani na podlagi natančnosti, kompleksnosti ter velikosti parametrov. V raziskavi smo uporabili Google Colab z Nvidia L4 grafično kartico, ki ima 24GB VRAM pomnilnika.

Rezultati so pokazali, da preproste arhitekture CNN modelov, ob pravilni optimizaciji hiperparametrov in regularizaciji dosegajo višje natančnosti kot večji ter kompleksnejši modeli. Bolj specifično sta bila dva modela preprostih CNN arhitektur bolj uspešna od ostalih večjih modelov. Med novjšimi modeli je najboljše razmerje med kompleksnostjo in natančnostjo dosegel EfficientNet, medtem ko so AlexNet, VGG16 in ResNet50 dosegali slabše rezultate v vseh metrikah. Najslabše rezultate poleg VGG16 modela pa je pričakovano dosegal model LeNet5, kar lahko pripišemo starosti in neprilagojenosti na sodobnejše podatkovne zbirke.

*Keywords*—CNN, Klasifikacija slik, Oxford-IIIT Pet Dataset, Natančnost, AIC

## I. UVOD

Klasifikacija slik je temeljni problem v računalniškem vidu ter predstavlja osnovo pri reševanju ostalih problemov na tem področju. Ti problemi vključujejo segmentacijo slik, lokalizacijo objektov, itd. Klasifikacija slik se osredotoča predvsem na dodelitev kategorij vhodnim slikam glede na njihovo vsebino. Kot primer klasifikacije lahko vzamemo razvrščanje slik hišnih ljubljencev v kategorije kot so mačka, pes, itd. Klasifikacija slik ima širok nabor uporabe na različnih področjih kot so sistemi za nadzor prometa, detekcija prometnih znakov, prepoznavanje obraza, itd. Z napredkom v razvoju globokega učenja, predvsem konvolucionalnih nevronske mrež (ang. Convolution Neural Networks - CNN) je klasifikacija slik dosegla visoke

natančnosti. Pri nekaterih nalogah te natančnosti že presegajo človeško natančnost. Kljub velikemu napredku pa pri klasifikaciji še vedno naletimo na številne težave, kot so kompleksna ozadja, omejena količina označenih podatkov in variabilnosti v podatkih [8].

## II. TEORETIČNO OZADJE

V tej raziskavi smo se soočili z problemom klasifikacije pasem domačih živali (mačk in psov) glede na njihovo sliko. Cilj te raziskave pa je bil raziskovanje in evalvacija različnih arhitektur, posameznih komponent konvolucijskih nevronske mrež ter različnih hiperparametrov in njihov vpliv na končno natančnost modelov in ostale metrike.

### A. Konvolucijske nevronske mreže

Konvolucijske nevronske mreže (v nadaljevanju CNN) so vrsta večslojnih nevronske mrež, ki so zasnovane za obdelavo podatkov v matrični obliki (npr. slike). Zaradi tega so postale eden izmed najpomembnejših delov sodobnega računalniškega vida ter imajo pomembno vlogo pri aplikacijah kot so klasifikacija slik ali prepoznavanje objektov na slikah.

Prvi CNN model strojnega učenja Le-Net [16] je razvil Yan LeCun leta 1988, uporabljal pa se je za prepoznavanje ročno napisanih števil (podatkovna zbirka MNIST).

Leta 2012 je na tekmovanju (ILSVRC) globoki CNN model Alex-Net [3] dosegel izjemne rezultate na podatkovni zbirki Image-NET. Na področju je predstavljal velik preboj in s tem povzročil širšo uporabo CNN modelov ter posledično razvoj novih modelov kot so VGG-net [12], EfficientNet [14] ter mnogo drugih.

CNN modeli imajo ponavadi od 3 do 150 slojev ali pa še več. Globina CNN modela se navezuje na število slojev, ki jih ima.

Konvolucijske nevronske mreže vključujejo nekaj osnovnih konceptov kot so:

**Vhodni sloj:** Sloj, ki sprejme slike v obliki (širina, višina, kanali).

**Konvolucijska plast:** Konvolucijska plast je osrednji del CNN. Plast uporablja številne filtre (ang. Kernel) ter izvaja konvolucijo nad vhodno sliko. Kot izhod konvolucija ustvarja značilnostne mape (ang. Feature Maps), ki zajemajo specifične dele slike kot so robovi, teksture ter ostale specifične lastnosti slike in tako omogočajo, da se model nauči preprostih kot tudi kompleksnejših značilnosti slike.

**Aktivacijska plast:** Z aktivacijsko plastjo dodamo modelu nelinearnost izhodu prejšnje plasti. Modeli uporabljajo številne aktivacijske funkcije vključno z ReLU, ELU, Tahn, itd. Izbira ustrezne aktivacijske funkcije je odvisna od zadane naloge.

**Pooling Plast:** Uporabljamo jo za zmanjševanje dimenzij značilnostnih map. Predvsem uporabljamo Max-Pooling ali Average-Pooling.

**Polno povezani sloji:** Sloj z nevroni, kjer je vsak nevron povezan z vsemi nevroni v prejšnjem sloju. Uporabljamo jih za združevanje in obdelavo informacij prejšnjih slojev in pripravo za končne napovedi.

***Izhodna plast:*** Sloj, ki nam da končni rezultat. Uporablja aktivacijsko funkcijo, ki je odvisna od naše naloge (npr. Sigmoid za binarno klasifikacijo, SoftMax za večrazredno klasifikacijo).

Pri evalvaciji modelov v tej raziskavi bomo uporabljali naslednje metrike:

**Natančnost**, ki nam pove delež pravilno klasificiranih primerov glede na skupno število primerov.

**Priklic** (ang. *Recall*), ki nam pove koliko od vseh dejansko pozitivnih primerov je model pravilno klasificiral kot pozitivne.

**Preciznost** (ang. *Precision*), ki nam pove delež pravilno napovedanih pozitivnih primerov glede na vse pozitivno napovedane primere.

**F1-rezultat**, pa nam pove harmonično razmerje med priklicom in preciznostjo.

**AIC** (ang. Akaikejev informacijski kriterij), je statistična metrika, ki meri ravnovesje med prilaganjem modela ter njegovo kompleksnostjo [2].

### III. METADOLOGIJA

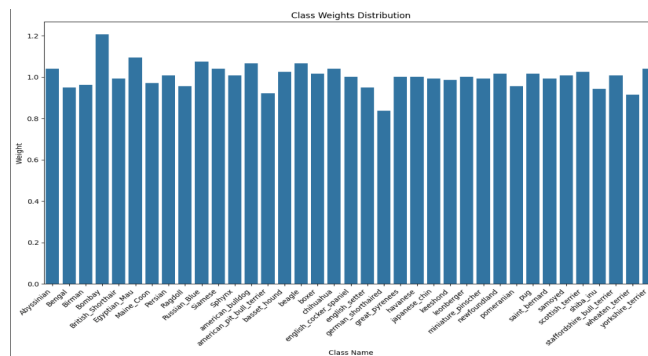
### A. Podatkovna zbirka

V tej raziskavi smo uporabili Oxford-IIT Pet Dataset [3], ki vsebuje 7349 slik, ki so kategorizirane v skupno 37 pasem mačk in psov. Vsaka pasma vsebuje okoli 200 slik.

Podatkovno zbirko smo razdelili v skupno 3 dele v naslednjih razmerjih:

- Trening nabor: 50% podatkov
- Validacijski nabor: 25% podatkov
- Testni nabor: 25% podatkov

Analiza razmerja med razredi je pokazala, da je v nekaterih razredih minimalno več instanc kot v drugih. Ta neuravnoteženost lahko negativno vpliva na učinkovitost moedlov, saj razredi z manj instancami prispevajo manj informacij pri učenju modelov. Na spodnji sliki 1, lahko opazimo distribucijo posameznih pasem v podatkovni zbirki.



### Slika 1: Razmerja med razredi

### B. Predprocesiranje podatkov

Podatke smo ustrezno pripravili na trening modelov. Tehnike uporabljene v fazi predprocesiranja podatkov so opisane spodaj:

**Augmentacija podatkov:** Augmentacija je tehnika, kjer podatkom umetno povečamo obseg in jim dodamo raznolikost v obliki manjšega naključnega šuma. Trening nabor podatkov smo povečali za petkrat ter slike augmentirali z naključnim obračanjem v levo in desno, obračanjem gor in dol, spremembo kontrasta, barvnega odtenka in svetlobe ter rotacije.

**Normalizacija:** Slike smo normalizirali v rang  $[0, 1]$ . To smo storili tako, da smo vsako sliko delili z 255, kar predstavlja maksimalno vrednost v RGB sliki.

**Sprememba velikosti:** Ključen korak pri pripravi podatkov za trening CNN modelov je zagotavljanje, da imajo vhodni podatki enake dimenzije, ki so prilagojene plastem v modelu. Vsem slikam smo spremenili velikost na enotno 224x224x3 z uporabo paddinga, za ohranjanje aspekta.

**Batchi:** Treniranje modela z eno sliko naenkrat je neučinkovit pristop. Zato smo se odločili za treniranje v Batchih. Ta pristop nam omogoča, da na enkrat obdelujemo več slik preden posodobimo uteži modela. Ta pritstop omogoča stabilno posodobitev gradientov ter boljšo konvergenco. Eksperimentiranje z različnimi velikostmi batchov (16, 32, 64 in 128) so pokazali, da je velikost batcha 32 najbolj učinkovita v kontekstu naše raziskave.

**One-Hot-encoding:** Ciljno spremenljivko smo pretvorili v binarno matriko, kjer vsako kategorijo predstavimo kot vektor z vrednostjo 1, če spremenljivka pripada razredu v nasprotnem primeru pa 0.

### C. Arhitecture CNN modelov

Sledijo opisi posameznih arhitektur CNN, katere so bile uporabljene v kontekstu te raziskave:

#### 1) Le-Net-5

- 2 konvolucijski plasti z 6 (7x7) in 16 (5x5) filtri
- Conv2D » Tahn » MaxPooling2D
- Dve polno povezani plasti z 256 in 84 nevroni
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 11,103,877

#### 2) Osnovni CNN 1

- 4 konvolucijske plasti z naraščajočim številom filtrov (16, 32, 64, 128) velikosti (3x3)
- Druga plast uporablja SeperableConv2D
- Conv2D » BatchNormalization » ReLU » MaxPooling2D
- GlobalAveragePooling2D
- Polno povezani sloj z 128 nevroni
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 248,085

#### 3) Osnovni CNN 2

- 6 konvolucijskih plasti z naraščajočim številom filtrov (8, 16, 32, 64, 128, 256) velikosti (3x3) in (5x5)
- Conv2D » BatchNormalization » LeakyReLU » MaxPooling2D
- GlobalAveragePooling2D
- Polno povezani sloj z 256 nevroni
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 470,917

#### 4) Alex-Net

- 5 konvolucijskih plasti z filtra (96, 256, 384, 256)
- Začetni sloji imajo večje filtre (11x11 in 5x5) v kasnejših pa (3x3)
- Conv2D » ReLU » MaxPooling2D
- 2 polno povezana sloja z 2048 nevroni
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 26,917,669

#### 5) VVG-16

- 13 konvolucijskih plasti z naraščajočim številom filtrov (64, 128, 256, 512, 512)
- Prve dve plasti imajo 2 konvolucije zadnje tri pa 3 konvolucije
- Conv2D » ReLU » MaxPooling2D
- Dve polno povezani plasti z 4096 nevroni
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 134,412,133

#### 6) ResNet-50

- Začetni konvolucijski sloj (64 filtrov (7x7)) » BatchNormalization » ReLU
- 5 slojev z residualni bloki z večanjem filtrov
  - o Sloj 2: 3 bloki (64 -256 filtrov)
  - o Sloj 3: 4 bloki (128-512 filtrov)
  - o Sloj 4: 6 bloki (256-1024 filtrov) Sloj 5: 3 bloki (512-2048 filtrov)
- Preskočna povezava (ang. Skip connection)
- GlobalAveragePooling2D
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 24,269,733

#### 7) EfficiantNet

- 7 initail konvolucijskih blokov (32 filtrov, 3x3, stride 2)
- 7 MBConv blokov z naraščajočimi filtri (16, 24, 40, 80, 112, 192, 320)
- Swish aktivacija, DepthWise konvolucije
- GlobalAveragePooling2D
- SoftMax izhodni sloj z 37 razredi
- **Število parametrov:** 2,066,629

### D. Trening modelov

Trening modelov je bil izveden z ciljem maksimizirati natančnost modelov in hkrati preprečevanje prenasíčenosti (Pojav, ko se model predobro prilega trening podatkom, vendar ima na testnih in novih podatkih zelo slabe rezultate). Ključne strategije, ki smo jih uporabili v raziskavi vključujejo regularizacije L1, L2, dropout, EarlyStopping, uravnoteženje razredov, dinamična stopnja učenja (ang. Learning rate) in primerni inicializerji uteži.

**Early Stopping:** Da smo preprečili prenasíčenost in nepotrebne trening epoche smo nastavili EarlyStopping, ki je ustavila trening modela, ko se validacijska natančnost ni izboljšala 10 zaporednih epochov.

**Learning Rate Scheduling:** Modelom smo dodali dinamično stopnjo učenja, da bi dosegli boljšo konvergenco. Če se validacijska natančnost ni izboljšala 5 zaporednih epochov smo zmanjšali stopnjo učenja za faktor 0.5, kar pomeni za polovico.

**Uteži razredov:** Kljub temu, da ni bilo velikih odstopanj neuraunoterenih razredov smo uteži razredov izračunali in jih uporabili pri treningu modelov.

**L1 regularizacija:** Dodali smo L1 regularizacijo, ki dodaja penale in sicer delež absolutne vsote uteži. Z tem dosežemo, da se nekatere uteži zmanjšajo na točno nullo in s tem dosežemo bolj preproste modele. Dodali smo jih konvolucijskim in polno povezanim slojem.

**L2 regularizacija:** Dodali smo tudi L2 regularizacijo, ki dodaja penale in sicer delež kvadratne vsote uteži, kar zmanjša uteži in z tem preprečuje prenasičenost. Dodali smo jih konvolucijskim in polno povezanim slojem.

**Dropout:** Uporabili smo dropout tehniko, kjer deaktiviramo določen procent nevronov med treningom in z tem preprečimo prenasičenost.

**Kernel Initializations:** Uporabljene v konvolucijskih slojih, kjer smo imeli ReLU aktivacijsko funkcijo, kjer smo imeli Tahn in SoftMax pa smo uporabili GlorotUniform.

Vsi parametri so bili optimizirani z eksperimentiranjem. Pri računsko manj zahtevnih modelih kot so preprosti CNN, Le-Net-5 in AlexNet smo uporabili optuno za iskanje najboljših parametrov.

Modeli so bili trenirani na največ 500 epochov in so uporabljali kategorično krossEntropijo kot funkcijo izgube. Za optimizator smo po več eksperimentih izbrali Adam optimizator. Trening nabor podatkov je bil augmentiran in povečan z čimer smo poskušali zagotoviti boljšo generalizacijo in validacijsko natančnost. Po treningu smo vse modele med seboj primerjali z uporabo različnih metrik.

## IV. REZULTATI IN RAZPRAVA

Evaluacija različnih arhitektur CNN modelov na testnih podatkih izvzetih iz treninga, kažejo na pomembne razlike v metrikah modelov kot so natančnost, preciznost, priklic, F1-rezultat in AIC. Pomembne ugotovitve vključujejo:

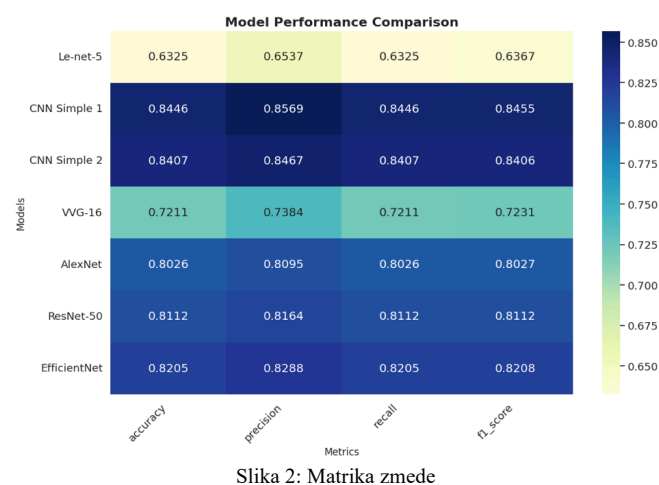
### 1) Natančnost

Presenetljivo sta oba **preprosta CNN** modela dosegla najvišje natančnosti (84,46 % CNN model 1 in 84,07 % CNN model 2). Ugotovitve kažejo, da preprostejši modeli lahko dosegajo primerljive natančnosti z pravilno optimizacijo in tuningom kot globlji modeli.

**EfficientNet** je tudi pokazal dobre rezultate in dosegel 82,05 % natančnost z veliko manj parametri, kot ostali večji modeli kot so Alex-Net in ResNet-50.

**Alex-net, ResNet-50 in VVG16** so dosegli malo nižje natančnosti (Res-Net-50 – 81,12 %, AlexNet - 80,26 % in VVG16 – 72,11%). Slabše rezultate lahko pripišemo malo slabšemu tuningu, zaradi omejitve računalniške moči uporabljene v eksperimentu.

**Le-Net-5** je imel pričakovano najnižjo natančnost glede na vse metrike z natančnostjo 63,25 %, kar kaže na nekonkurenčnost proti naprednejšim in kompleksnejšim podatkovnim zbirkam. Model je neprimeren za to podatkovno zbirko, vendar smo ga vključili za primerjavo zaradi svoje zgodovinske pomembnosti.



Slika 2: Matrika zmede

Evalvacija preciznosti, priklica in F1-rezultata nam dodatno potrjujejo, da so bolj preproste arhitekture z ustrezno regularizacijo in tuningom, bolj učinkovite kot starejši ali preveč kompleksni modeli. Od novejših arhitektur ima EfficientNet najboljše razmerje med priklicom in preciznostjo medtem ko sta optimizirana preprosta CNN še vedno dosegla boljše rezultate.

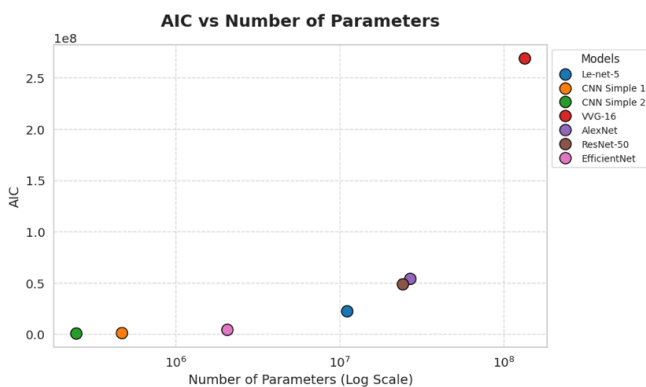
### 2) Kompleksnost modelov

Kompleksnost modelov, merjeno z AIC metriko je razkrila značilne razlike med arhitekturami ter vpliv števila parametrov model na uporabnost. Najmanj kompleksen in najboljši model je bil preprost CNN 2, ki je z najnižjo AIC vrednostjo in najnižjim številom parametrov dosegal najboljše razmerje med kompleksnostjo in uspešnostjo modela. Sledil mu je preprosti CNN 1, ki je tudi dosegel nizko AIC vrednost, vendar je imel več parametrov. Efficient net model, kot tretji po kompleksnosti je tudi dosegel podobne rezultate vendar z bistveno več parametri kot preprosta CNN modela. Ostali modeli so dosegli zelo slabe rezultate glede AIC metrike.

Presenetljivo je Le-Net-5 model kot četrti, kar kaže na boljše razmerje med kompleksnostjo in uporabnostjo. Najslabši model glede razmerja med kompleksnostjo in učinkovitostjo je VVG, ki ima najvišjo AIC vrednost in največje število

parametrov, sledita mu Alex-Net, Res-Net-5, katera imata podobne vrednosti AIC.

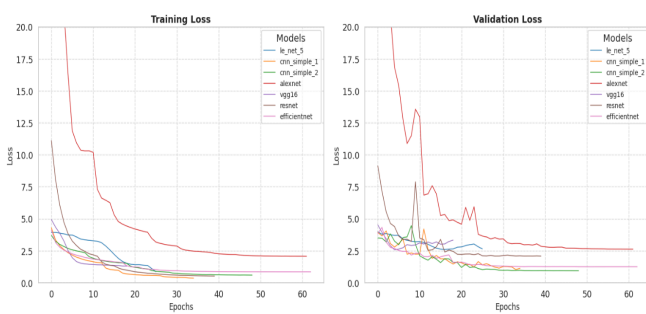
Raziskava nam je jasno pokazala, da je pri evaluacijah modelov poleg osnovnih metrik, dobro upoštevati tudi kompleksnost modela in število parametrov, saj tako najdemo najboljše razmerje med njima. Moderne arhitekture kot je EfficientNet nudijo dobro razmerje med kompleksnostjo in uspešnostjo, ampak lahko z dobro optimizacijo parametrov dosežemo podobne in boljše rezultate z bolj preprostimi modeli. Starejše arhitekture z več parametri pa ne upravičijo svoje kompleksnosti z številom parametrov glede na izbrane modele in podatkovni niz uporabljen v tej raziskavi.



Slika 3: AIC in število parametrov

### 3) Izguba in generalizacija

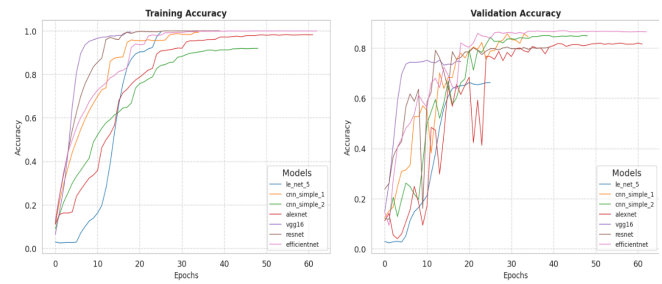
Najnižje vrednosti izgube sta dosegla preprosti CNN 2 (0.94) in EfficientNet (1.49), kar nakazuje na učinkovito optimizacijo ter generalizacijo na validacijskem naboru podatkov. Višje vrednosti pri modelih kot so VVG16 kažejo na prenasíčenost in omejitve preveč preprostih arhitektur kot je Le-Net-5. Najslabše rezultate in konvergenco kaže AlexNet, kar nakazuje na to, da je model preveč kompleksen in se ne prilaga dobro podatkom uporabljenim v tej raziskavi.



Slika 4: Trening in validacijska izguba

### 4) Trening in validacijska natančnost

Vsi modeli so hitro dosegli skoraj 100% trening natančnost, medtem ko pri validacijski natančnosti vidimo, da sta najbolj stabilna preprosta CNN modela in EfficientNet. Le-Net-5 je pričakovano dosegel slabo validacijsko natančnost in ni konvergirala nad približno 60 %. AlexNet in VVG16 pa kažeta na visoko nestabilnost.



Slika 5: Trening in validacijska natančnost

## V. ZAKLJUČEK

Raziskava je nedvoumno pokazala, da lahko z uporabo preprostih CNN arhitektur, kot sta preprosti CNN 1 in CNN 2, dosežemo zelo dobre natančnosti pri klasifikaciji slik in se lahko primerjajo z večjimi in bolj kompleksnimi modeli kot so AlexNet, ResNet-50 ali VVG16. EfficientNet se je pokazal kot dobra izbira med novejšimi modeli, saj ponuja dobro razmerje med kompleksnostjo in natančnostjo vendar sta ga preprosta CNN modela z manj kompleksno arhitekturo presegla v uspešnosti klasifikacije.

Analiza nam je pokazala, da večje število parametrov za zagotavlja boljše natančnosti. Dokazali smo ravno nasprotno in sicer, da preveč kompleksni modeli lahko dosegajo slabše rezultate, predvsem zaradi prenasíčenosti in težje optimizacije hiperparametrov.

Pomembno je poudariti, da igra ključno vlogo pravilna izbira hiperparametrov, ustreznih tehnik regularizacije ter tuning modelov pri doseganju visokih natančnosti in dobre generalizacije modelov.

Kot zaključek raziskave lahko izpostavimo, da je izmed modelov vključenih v eksperiment preprosti CNN 2, ob pravilni optimizaciji najbolj učinkovit model za klasifikacijo slik na podatkovni zbirki Oxford-IIIT Pet Dataset. V prihodnjih raziskavah priporočam vključitev več podatkov in augmentacije, vključitev novih naprednejših arhitektur ter boljšega tuninga kompleksnejših modelov.

- [1] „Logistic Regression in Machine Learning“, GeeksforGeeks. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://www.geeksforgeeks.org/understanding-logistic-regression/>
- [2] „Neural network AIC and BIC calculation (number of parameters?)“. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://www.mathworks.com/matlabcentral/answers/195674-neural-network-aic-and-bic-calculation-number-of-parameters>
- [3] „oxford\_iiit\_pet | TensorFlow Datasets“, TensorFlow. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: [https://www.tensorflow.org/datasets/catalog/oxford\\_iiit\\_pet](https://www.tensorflow.org/datasets/catalog/oxford_iiit_pet)
- [4] „Unveiling the Diversity: A Comprehensive Guide to Types of CNN Architectures | by Mohsen Nabil | Medium“. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://medium.com/@navarai/unveiling-the-diversity-a-comprehensive-guide-to-types-of-cnn-architectures-9d70da0b4521>
- [5] „What is Akaike Information Criterion (AIC)“. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://www.activeloop.ai/resources/glossary/akaike-information-criterion-aic/>
- [6] D. E. Lee, „Choosing the Right Colab Runtime: A Guide for Data Scientists and Analysts“, Medium. Pridobljeno: 22. december 2024.

- [Na spletu]. Dostopno na: <https://drlee.io/choosing-the-right-colab-runtime-a-guide-for-data-scientists-and-analysts-57ee7b7c9638>
- [7] F. Chollet, „Xception: Deep Learning with Depthwise Separable Convolutions“, 4. april 2017, *arXiv*: arXiv:1610.02357. doi: [10.48550/arXiv.1610.02357](https://doi.org/10.48550/arXiv.1610.02357).
- [8] G. Boesch, „A Complete Guide to Image Classification in 2025“, viso.ai. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://viso.ai/computer-vision/image-classification/>
- [9] I. Janiszewski, D. Slugin, in V. Arlazarov, *Training the Convolutional Neural Network with Statistical Dependence of the Response on the Input Data Distortion*. 2019. doi: [10.48550/arXiv.1912.00664](https://doi.org/10.48550/arXiv.1912.00664).
- [10] J. U. Rahman, F. Makhdoom, in D. Lu, „ASU-CNN: An Efficient Deep Architecture for Image Classification and Feature Visualizations“, 28. maj 2023, *arXiv*: arXiv:2305.19146. doi: [10.48550/arXiv.2305.19146](https://doi.org/10.48550/arXiv.2305.19146).
- [11] K. He, X. Zhang, S. Ren, in J. Sun, „Deep Residual Learning for Image Recognition“, 10. december 2015, *arXiv*: arXiv:1512.03385. doi: [10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385).
- [12] K. Simonyan in A. Zisserman, „Very Deep Convolutional Networks for Large-Scale Image Recognition“, 10. april 2015, *arXiv*: arXiv:1409.1556. doi: [10.48550/arXiv.1409.1556](https://doi.org/10.48550/arXiv.1409.1556).
- [13] M. S. Z. Rizvi, „Image Classification Using CNN“, Analytics Vidhya. Pridobljeno: 22. december 2024. [Na spletu]. Dostopno na: <https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/>
- [14] M. Tan in Q. V. Le, „EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks“, 11. september 2020, *arXiv*: arXiv:1905.11946. doi: [10.48550/arXiv.1905.11946](https://doi.org/10.48550/arXiv.1905.11946).
- [15] M. Z. Alom *idr.*, „The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches“, 12. september 2018, *arXiv*: arXiv:1803.01164. doi: [10.48550/arXiv.1803.01164](https://doi.org/10.48550/arXiv.1803.01164).
- [16] Y. Lecun, L. Bottou, Y. Bengio, in P. Haffner, „Gradient-Based Learning Applied to Document Recognition“, *Proceedings of the IEEE*, let. 86, str. 2278–2324, dec. 1998, doi: [10.1109/5.726791](https://doi.org/10.1109/5.726791).