

Klasifikacija životinja na osnovu crteža primenom konvolutivnih neuronskih mreža

Boško Kulušić^{#1}, Jelena Kovač^{#2}

[#]Fakultet tehničkih nauka, Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6, Srbija

¹kulusic.e224.2024@uns.ac.rs

²jelena.kovac@uns.ac.rs

Abstrakt— Korišćenje skica i crteža za izražavanje ideja je pojava sa kojom se susrećemo stalno. Od praistorijskog doba, čovek je imao potrebu da svoje ideje pretače u crteže. Stoga, prepoznavanje životinja na ručnim skicama je problem na koji su se autori fokusirali. U ovom radu, korištene su poznate arhitekture konvolucionih neuronskih mreža - DenseNet121, InceptionV1, ResNetV2, EfficientNet-B3. Najbolje rezultate pri rešavanju problema klasifikacije crteža/skica životinja ostvarili su modeli sa DenseNet i EfficientNet arhitekturama. Ovaj rad se bavi upoređivanjem dobijenih rezultata pomenutih arhitektura, izborom hiperparametara kao i rešavanjem problema *overfitting-a*.

Gljučne reči— CNN, ResNet, EfficientNet, GoogleNet, DenseNet, Klasifikacija crteža

I. UVOD

Ljudi koriste ručne crteže, poznatiji kao skice, od praistorijskog doba da bi izrazili svoje ideje i misli. Pored umetničkog izražaja, skice imaju i praktični aspekt koji ima zadatak da prenese informaciju. Skiciranje predstavlja jedan od osnovnih načina na koji ljudi prenose vizuelne informacije. Time, skice predstavljaju svojevrsan univerzalni jezik gde se osnovni predmet koji je od interesa može jasno identifikovati čime se prevazilaze ograničenja jezika, kulture i vremenskog perioda i starosne grupe [2].

Varijacija u stilovima crtanja predstavlja jedan od izazova ovog problema. Stil crtanja varira od autora do autora crteža. Dok se u fotografijama varijacije pronalaze u različitom osvetljenju, poziciji, uglovima i slično, varijacija u crtežima je jedinstvena.

Takođe, crteži su strukturalno jednostavniji od fotografija. dok fotografije uglavnom sadrže tri kanala(RGB), crteži su u crno-belom boji.

Klasifikacija skica se posmatra kao problem koji se rešava u okviru računarske vizije.

U ovom radu, fokusirali smo se na upoređivanje rezultata različitih arhitektura konvolutivnih neuronskih mreža kroz mehanizam koji je poznat kao *fine tuning*. Poznate arhitekture koje su korištene su *ResNet50V2*, *EfficientNetB3*, *DenseNet-121* i *InceptionV1(GoogLeNet)*. Tačnost(*accuracy*), preciznost(*precision*), odziv(*recall*) i F1 mera su korišteni kao metrike evaluacije ponašanja samog modela.

Rad je dalje organizovan na sledeći način. U drugom poglavlju predstavljamo relevantnu literaturu za naš problem. Dalje predstavljamo metodologiju zajedno sa arhitekturama koje koristimo. Sledi analiza rezultata i na kraju je predstavljen zaključak.

II. RELEVANTNA LITERATURA

U radu [1] predstavljena je klasifikacija crteža životinja korišćenjem *ResNet* arhitekture konvolutivnih mreža. Najbolje rezultate je davala kombinacija dropout-a od 20%, zajedno sa *learning rate*-om od 0.001 za prvih 15 epoha i 0.0001 za poslednjih 5 epoha. Model je ostavario ukupnu tačnost od 65%. Autori su imali problem sa varijacijama u crtežima. Model je najviše grešio u klasifikaciji ptica i pande zbog međuklasnog preklapanja sa drugim klasama.

U radu [2], autori predstavljaju klasifikaciju ručnih crteža korišćenjem poznatih arhitektura konvolutivnih neuronskih mreža: *ImageNet* i *LenNet*. Koriste konvolutivni sloj pomenutih arhitektura, dok su se odlučili za *SVM* kao

klasifikator. Skup podataka su proširivali augmentacijom podataka i to primenjujući sledeće metode: rotaciju, zumiranje, preslikavanje i sistematske kombinacije horizontalnih i vertikalnih pomeranja. Slike dobijene proširenjem su koristili tokom treniranja, dok se za testni skup koriste isključivo slike nad kojima nisu vršene transformacije. ImageNet se pokazao superiornijim od *LenNet* arhitekture. Autori smatraju da je to slučaj jer dublji konvolutivni slojevi pomažu u identifikaciji krucijalnih atributa na slici.

U radu [3], autori su rešavali problem klasifikacije departmana Utarančalsog univerziteta u Indiji na osnovu slika koje su prikupili. Korištene su duboke konvolutivne neuronske mreže i to: *ResNet50V2*, *ResNet152V2*, *MobileNet*, *InceptionV3* i *Xception* kroz primenu *transfer learning-a*. Radi jednostavnosti, slike su konvertovane u *grayscale*. Augmentacija podataka je primenjena u cilju rešavanja problema *overfitting-a* i radi proširivanja početnog skupa podataka radi poboljšanja generalizacije modela. Tačnost je korištena kao mera evaluacije modela, *ResNet50V2*, *ResNet152V2* i *InceptionV3* su postigli tačnost od 98.88%, 99.10% i 99.17% respektivno.

U radu [4] predstavljena je klasifikacija scena u daljinskom osmatranju (*Remote sensing*). Korištena je arhitektura duboke konvolutivne mreže *EfficientNetB3* uz dodavanje prilagođenog *attention* mehanizma u cilju poboljšavanja tačnosti modela. Eksperimentalna postavka uključuje korištenje Adam optimizatora, *batch size* od 32, *learning rate* 0.001 prvih 15 epoha i 0.0001 narednih 15 epoha. Trenirane su dvije mreže, Base *EfficientNetB3* i njena predložena, uapredena verzija sa novim *attention* mehanizmom (*EfficientNet-B3-Attn-2*). Ostvareni rezultati su upoređivani nad šest skupova podataka sa ostalim poznatim arhitekturama konvolutivnih neuronskih mreža. Iako je model uspio da ostvari bolje rezultate od većine modela, nije uspio da ostvari generalni najbolji rezultat. Autori smatraju da je *downsampling* slika na dimenziju 256x256 sa 400x400 značajno uticao na gubitak bitnih informacija na slikama.

U radu [10] autori su primenili transfer učenje sa 'zamrznutim' (*frozen*) ekstraktorima karakteristika na više dubokih konvolutivnih mreža (CNN) i transformera. Korišćenjem *DenseNet201*, *ResNet152*, *EfficientNetB4* i Vision Transformer-a, utvrđeno je da je *DenseNet201* postigao najbolje performanse među CNN mrežama (sa tačnošću od 67%), uz istovremeno zadržavanje male računarske kompleksnosti i lakoće implementacije. Rad potvrđuje efikasnost korišćenja pre-treniranih CNN arhitektura kao što je DenseNet u zadacima klasifikacije u domenu prepoznavanja životinja.

III. METODOLOGIJA

U ovom poglavlju detaljno je opisana metodologija primenjena za klasifikaciju skica životinja. Fokus je na opisu skupa podataka, strategiji podele, tehnikama pretprocesiranja i detaljima implementacije transfer učenja (finog podešavanja) za odabrane CNN arhitekture. Celokupan proces implementacije sproveden je korišćenjem Python programskog jezika i njegove PyTorch biblioteke za duboko učenje [9].

A. Skup podataka

Projektom je rešen problem klasifikacije gde se prepoznaju ručno nacrtane skice raznih životinja. Za ovaj zadatak korišćen je javno dostupan *Doodle Dataset* sa *Kaggle* platforme. [11]

Iako skup podataka sadrži 340 klasa, za potrebe ovog projekta odabrano je 20 klasa životinja. Odabrane klase su: mačka (*cat*), krava (*cow*), krokodil (*crocodile*), pas (*dog*), patka (*duck*), slon (*elephant*), riba (*fish*), jež (*hedgehog*), konj (*horse*), kengur (*kangaroo*), lav (*lion*), majmun (*monkey*), sova (*owl*), panda (*panda*), svinja (*pig*), ovca (*sheep*), puž (*snail*), zmija (*snake*), pauk (*spider*) i zebra (*zebra*).

Ukupan skup podataka obuhvata 60000 slika, sa po 3000 slika po klasi, čime je obezbeđena balansiranost skupa. Slike su *grayscale* (crno-bele), rezolucije 255x255 piksela i u PNG formatu.



Slika 1: Prikaz skica iz klase pas



Slika 2: Prikaz skica iz klase puž

B. Pretprocesiranje i podela skupa

Ulazne slike su zahtevale transformacije radi prilagođavanja *ImageNet* pretreniranim modelima. Slike su skalirane na 224x224 piksela. *Grayscale* slike su konvertovane u trokanalni (RGB) format dupliranjem kanala. Takođe, vrednosti piksela su normalizovane na osnovu proseka i standardne devijacije *ImageNet* skupa.

U cilju poboljšanja generalizacije modela i borbe protiv prekomernog prilagođavanja (*overfitting*), na trening skupu primenjene su sledeće augmentacije:

- slučajna rotacija do 20 stepeni,
- slučajna translacija do 15% po X i Y osi,
- slučajno horizontalno preslikavanje sa verovatnoćom 0.3.

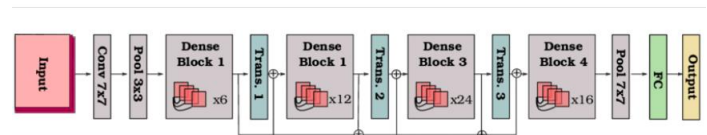
Skup podataka je podeljen na tri disjunktna podskupa: trening, validacioni i testni skup, u odnosu 70:15:15. Trening skup od 42000 slika (2100 po klasi) je korišten za obuku modela. Validacioni skup

od 9000 slika (450 po klasi) je korišten za praćenje performansi tokom obuke, podešavanje hiperparametra *learning rate* tokom treninga i implementaciju ranog zaustavljanja. Testni skup od 9000 slika (450 po klasi) je korišten za nepristrasnu, konačnu evaluaciju performansi modela.

C. Arhitekture konvolutivnih neuronskih mreža i strategija transfer učenja

Za ovaj projekat izabrane su četiri napredne CNN arhitekture, demonstrirajući različite pristupe dubokom učenju:

1. *DenseNet-121* (*Densely Connected Convolutional Networks*): Karakteriše se gustim povezivanjem, gde se izlaz svakog sloja povezuje sa ulazom svih narednih slojeva. Ovo omogućava efikasno ponovno korišćenje karakteristika i smanjuje problem nestajućeg gradijenta [5]. Ukupan broj parametara u *pytorch* implementaciji ove mreže iznosi oko 7 miliona.



Slika 3: Prikaz DenseNet121 arhitekture

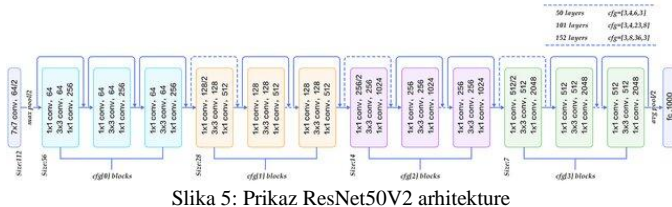
2. *InceptionV1* (*GoogleNet*): Koristi *Inception* module koji izvedu konvolucije različitih veličina istovremeno, čime se efikasno hvataju karakteristike različitih razmera i smanjuje broj parametara [6]. Ukupno postoji 9 *inception* modula u mreži. Ukupan broj parametara u *pytorch* implementaciji ove mreže iznosi manje od 6 miliona.



Slika 4: Prikaz GoogleNet arhitekture

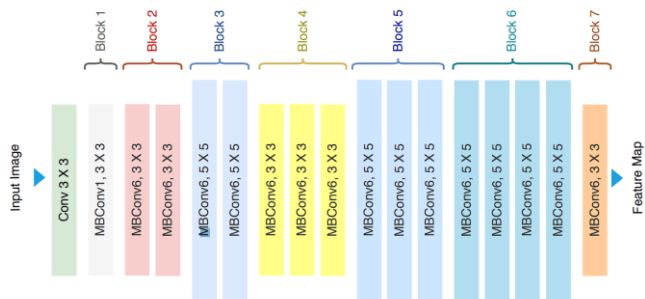
3. *ResNet50V2* (*Residual Networks*): Zasnovana je na konceptu rezidualnih konekcija (*skip connections*), koje omogućavaju direktan protok informacija zaobilazeći nekoliko slojeva. Verzija V2 optimizuje arhitekturu postavljanjem Batch

Normalization i aktivacione funkcije ReLU pre konvolucije[7]. Arhitektura sadrži ukupno 50 slojeva sa težinama zajedno sa 16 rezidualnih blokova i 16 rezidualnih konekcija. *Pytorch* implementacija ove arhitekture sadrži manje od 25 miliona parametara. Na slici 5, prikazana je arhitektura *ResNet50V2* arhitekture.



Slika 5: Prikaz ResNet50V2 arhitekture

4. *EfficientNet-B3*: Koristi strategiju kompaundnog skaliranja (*compound scaling*) za istovremeno optimizovanje dubine mreže, širine i rezolucije ulazne slike. Arhitektura B3 predstavlja balans između performansi i računarske efikasnosti [8]. Arhitektura sadrži oko 200 slojeva sa težinama, 22 *MBConv* bloka, 22 rezidualne konekcije i Swish aktivacionu funkciju. *Pytorch* implementacija sadrži manje od 11 miliona parametara. Na slici 6, prikazana je arhitektura *EfficientNetB3* arhitekture.



Slika 6: Prikaz EfficientNetB3 arhitekture

Korišten je princip finog podešavanja (*fine-tuning*) težina modela pretreniranih na *ImageNet* skupu. Težine prvih slojeva konvolutivne baze modela su zamrznute da bi se očuvale naučene generičke karakteristike. Poslednji duboki slojevi su odmrznuti i dotrenirani, a zamenjen je originalni potpuno povezani sloj klasifikatora novim linearnim slojem sa 20 izlaza.

U slučaju *DenseNet* modela, odmrznuti su i dotrenirani dva gusta bloka konvolutivne baze

denseblock3 i *denseblock4* i završni sloj normalizacije.

U slučaju *GoogleNet* modela, odmrznuti su i dotrenirani poslednji *inception* modul četvrte sekcije kao i čitava peta sekcija od dva *inception* modula.

U slučaju *ResNet* modela, odmrznut je i dotreniran poslednji rezidualni blok *layer4* kako bi se visoke konvolutivne osobine prilagodile novom skupu podataka.

U slučaju *EfficientNet* modela, odmrznuti su i dotrenirani sedmi i osmi konvolutivni blok koji se nalaze na kraju. Oba bloka predstavljaju *MBConv6* tip bloka arhitekture *EfficientNet*.

D. Hiperparametri i obuka

Kao funkcija gubitka (*Loss Function*) korišćena je kategorijska unakrsna entropija (*Categorical Cross-Entropy*) s obzirom na to da je reč o višeklasnom problemu klasifikacije.

Korišćen je Adam optimizator uz primenu različitih stopa učenja za segmente mreže. Hiperparametri su određeni empirijski. U cilju što efikasnije obuke i izbegavanja *overfitting*-a upotrebljeno je sledeće:

- Konvolutivni slojevi, to jest odmrznuti duboki slojevi su imali *learning rate* 0.0001 u slučaju *GoogleNet*, *ResNet* i *EfficientNet*, dok je *DenseNet* koristio 0.00005.
- Klasifikator je imao *learning rate* 0.001 u slučaju *GoogleNet*, *ResNet* i *EfficientNet*, dok je *DenseNet* koristio 0.0005..
- Primenjena je L2 regularizacija (*Weight Decay*): 0.4.
- Veličina serije (*Batch Size*): 64.
- *Learning Rate Scheduler*: Korišćen je *ReduceLROnPlateau* algoritam koji prati gubitak na validacionom skupu (*Validation Loss*) i smanjuje stopu učenja za faktor 0.5 ako se *loss* ne popravi u 2 epohe.
- Rano zaustavljanje (*Early Stopping*) je primenjeno na Validacioni gubitak, sa strpljenjem 4 epohe i minimalne razlike 0.01 za arhitekture *GoogLeNet*, *DenseNet* i *ResNetV2* u cilju sprečavanja prekomernog prilagođavanja. *EfficientNet* arhitektura je koristila minimalnu razliku 0.005. Maksimalan broj epoha postavljen je na 30.

E. Metrike evaluacije

Evaluacija na testnom skupu sprovedena je korišćenjem funkcije gubitka Kategorijska unakrsna entropija (*Cross-Entropy Loss*) i sledećih metrika:

1. Tačnost (*Accuracy*)
2. Preciznost (*Precision*)
3. Odziv (*Recall*)
4. F1-mera (*F1-Score*)

Osim ovih metrika, prikazani su i matrica konfuzije najboljeg modela, kao i tačnost po klasama.

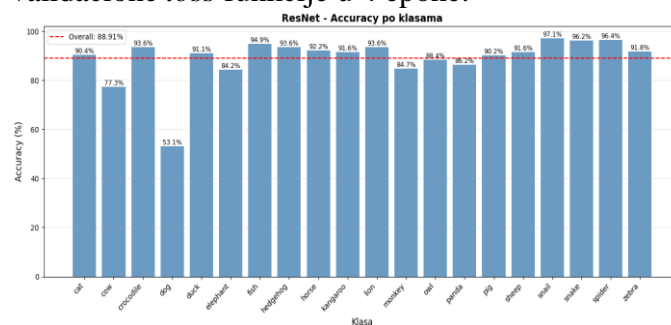
F1-mera, Preciznost i Odziv su izračunate kao ponderisani prosek (*weighted average*) preko svih 20 klasa, gde je težina svake klase jednaka njenom broju uzoraka u testnom skupu. S obzirom na balansiranost skupa, ovaj prosek daje rezultat ekvivalentan makro proseku.

IV. Rezultati

U radu je testirano više arhitektura dubokih konvolutivnih mreža kroz fine tuning pristup za prepoznavanje životinja na osnovu skica. Korištene su arhitekture: ResNet50V2, EfficientNetB3, DenseNet-121 i InceptionV1(GoogLeNet). Sve arhitekture su trenirane nad istim skupom podataka uz tačnost, preciznost, odziv i F1 meru kao metrike evaluacije. Praćenje loss funkcije se koristilo za prevenciju overfitting problema i za dalje podešavanje parametara. Takođe, augmentacija podataka je primenjena nad trening skopom, da bi se povećala generalizacija modela i preventovao overfitting. Slike u našem skupu podataka imaju jedan grayscale kanal, dok arhitekture očekuju tri kanala. Strategija koju smo koristili je sledeća: za preostala dva kanala smo iskoristili već postojeći.

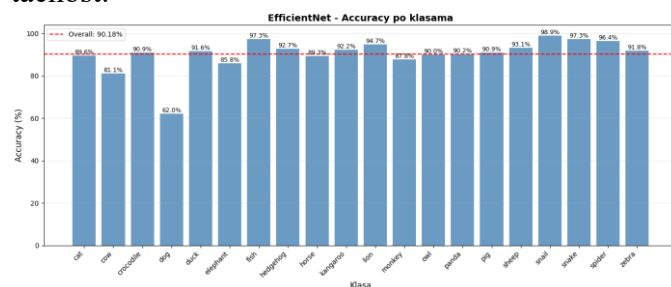
Pomenute arhitekture su trenirane na *NVIDIA Tesla T4* grafičkoj kartici. Eksperiment je izveden u *Google Colab* okruženju. Modeli koji su imali najduži trening su *EfficientNet* i *DenseNet* koji su trajali po 1 sat i 30 minuta. Ostale dve arhitekture su svoj trening uspešno kompletirale u vremenskom periodu od 1 sat +/- 10 minuta.

ResNet50V2 je postigao tačnost od 88.91%, preciznost od 88.92%, odziv od 88.91% i F1 meru od 88.77%. Tačnost po klasama je data na slici 7. Model je ostvario najmanju tačnost za psa, međutim to se može objasniti jer ta klasa ima dosta preklapanja sa ostalim i teško je izdvojiti jasne karakteristike koje ga određuju. Najveća tačnost je postignuta za puža. *Early stopping* je aktiviran nakon 20 epoha nakon što nije došlo do poboljšanja validacione *loss* funkcije u 4 epohe.



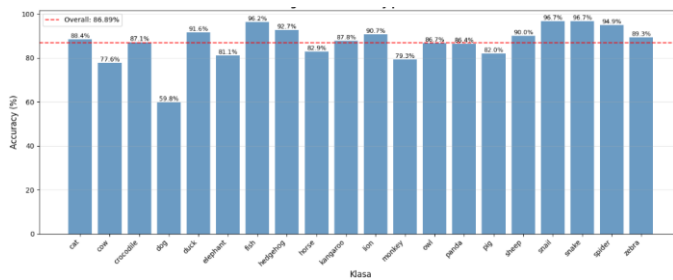
Slika 7: Prikaz tačnosti po klasama za ResNet50V2

EfficientNetB3 je postigao tačnost od 90.18%, preciznost od 90.28%, odziv od 90.18% i F1 mera 90.17%. Tačnost po klasama je prikazana na slici 8. Kao i za ostale modele, klasifikacija psa je ostvarila najmanju tačnost dok je klasa puža ostvarila najveću tačnost.



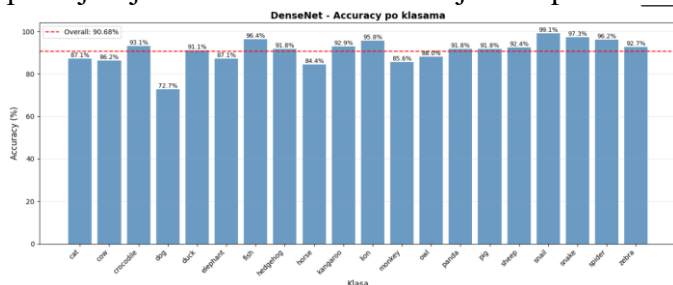
Slika 8: Prikaz tačnosti po klasama za EfficientNetB3

GoogLeNet je postigao tačnost od 86.89%, preciznost od 87.11%, odziv od 86.89% i F1 meru od 86.94%. Kao i za ostale modele, klasifikacija psa je ostvarila najmanju tačnost dok je klasa puža ostvarila najveću tačnost. Najveća tačnost je postignuta za puža i zmiju. *Early stopping* je aktiviran nakon 22 epohe nakon što nije došlo do poboljšanja validacione *loss* funkcije u 4 epohe i uprkos činjenici da je nakon 21. epohe aktivirano smanjenje *learning rate*-a za 50%.



Slika 9: Prikaz tačnosti po klasama za GoogleNet

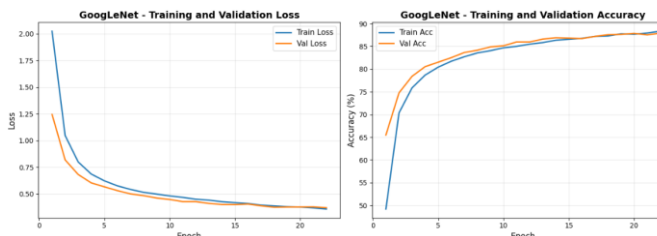
DenseNet121 je postigao tačnost od 90.68%, preciznost od 90.90%, odziv od 90.68% i F1 meru od 90.74%. Kao i za ostale modele, klasifikacija psa je ostvarila najmanju tačnost dok je klasa puža ostvarila najveću tačnost. Najveća tačnost je postignuta za puža i zmiju. *Early stopping* je aktiviran nakon 25 epoha nakon što nije došlo do poboljšanja validacione *loss* funkcije u 4 epohe.



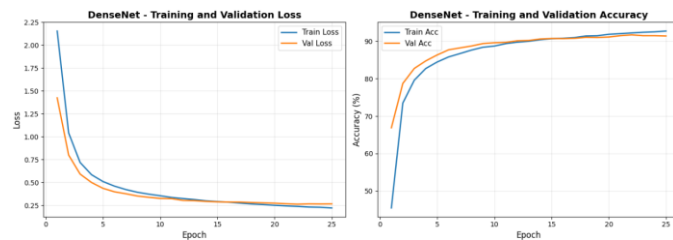
Slika 10: Prikaz tačnosti po klasama za DenseNet

Prevenција *overfitting* problema je rešena augmentacijom trening skupa kao i podešavanjem *learning rate* hiperparametra. Pored korištenja *Learning rate scheduler* mehanizma, početne vrednosti ovog hiperparametra su pronađene eksperimentalnim putem za svaku od arhitektura.

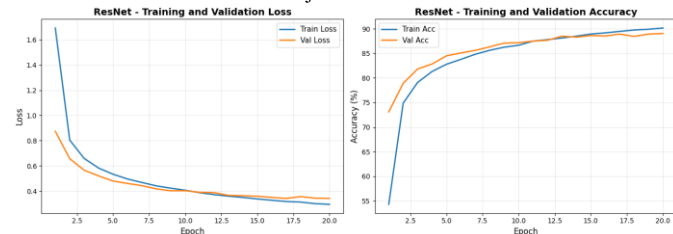
Na slikama 11-14 su prikazane *loss* funkcija i tačnost po pomenutim arhitekturama konvolutivnih neurosnkih mreža.



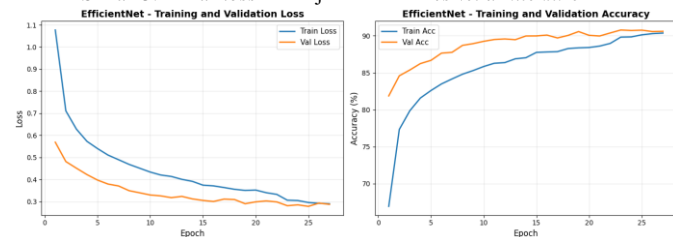
Slika 11: Prikaz *loss* funkcije i tačnosti GoogleNet arhitekture



Slika 12: Prikaz *loss* funkcije i tačnosti DenseNet arhitekture



Slika 13: Prikaz *loss* funkcije i tačnosti ResNet arhitekture

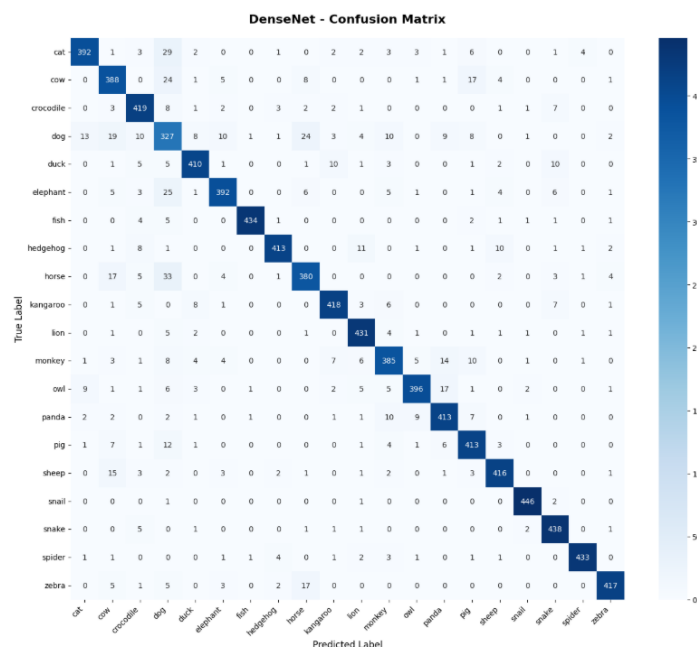


Slika 14: Prikaz *loss* funkcije i tačnosti EfficientNet arhitekture

Može se primetiti da je model koji je najbrže konvergirao ResNet koji je svoj trening završio u 20 epoha. Na osnovu prikazanih grafika, može se zaključiti da nije došlo do *overfitting-a* ni na jednom od modela, stoga je taj problem uspešno rešen pomenutim metodama.

Modeli DenseNet-121 i EfficientNet-B3 ostvarili su najvišu tačnost, što potvrđuje njihovu sposobnost efikasnog učenja kompleksnih apstraktnih karakteristika prisutnih u crtežima ili skicama. Ova visoka tačnost ukazuje na prednosti gusto povezanih slojeva DenseNet modela i protoka informacija i ponovnom korišćenju karakteristika, kao i na dobro postavljene parametre treninga. EfficientNet-B3 je najmodernija arhitektura i očekivano ima visoku tačnost, međutim zbog ne toliko velike složenosti problema nije pokazao svoju punu moć te ostvario malo slabiji rezultat od DenseNet-121 modela. Nasuprot tome, model GoogleNet (InceptionV1) je zabeležio najniže performanse, što se može pripisati njegovoj starijoj arhitekturi koja je manje optimizovana za hvatanje finih detalja i apstrakcija svojstvenih crtežima i skicama, u poređenju sa naprednijim modelima.

Matrice konfuzije (Confusion Matrices) za najbolji model (DenseNet) pokazuje mešanje između klase pas sa klasama konja, mačke i krave. Osim toga ističe se i mešanje među klase konja sa klasama zebra i krava. Pretpostavka je da je ovo posledica sličnog izgleda figure tih životinja na skicama iz profila, kao i velike raznovrsnosti klase pas.



Slika 15: Prikaz matrice konfuzije za DenseNet model

V. ZAKLJUČAK

U ovom radu sprovedeno je sistematsko poređenje četiri napredne konvolutivne neuronske mreže (DenseNet-121, InceptionV1, ResNetV2 i EfficientNet-B3) u zadatku višeklasne klasifikacije ručno nacrtanih skica životinja. Korišten je javno dostupan Doodle Dataset i primenjen je pristup transfer učenja (*fine-tuning*) na 20 odabranih klasa.

Model koji je postigao najveću tačnost na testnom skupu bio je DenseNet, ostvarivši tačnost od 90.68% i F1-meru od 90.74%, dok je GoogleNet (InceptionV1) imao najniže performanse. Analiza matrica konfuzije pokazala je mešanja između klase pas sa klasama poput konja, mačke i krave. To objašnjava najmanju ostvarenu tačnost u klasi pas kod svih modela.

Rezultati demonstriraju izuzetnu efikasnost transfera učenja i tehnike finog podešavanja u adaptaciji pretreniranih CNN modela na domen klasifikacije skica životinja. Detaljno poređenje četiri arhitekturne paradigme pruža vredan uvid istraživačima koji se bave klasifikacijom skica.

Buduća istraživanja mogu se fokusirati na primenu naprednijih tehnika augmentacije specifičnih za skice, testiranje na širem spektru klasa iz Doodle Dataset-a, i implementaciju *attention* mehanizama unutar postojećih arhitektura kako bi se model usmerio na ključne karakteristike skice (npr. oblik tela, dužina udova) i time rešiti problem sa mešanjem klase psa koji je uočen kod modela svih arhitektura.

REFERENCE

- [1] W. Lu and E. Tran, "Free-hand Sketch Recognition Classification," Stanford University, CS231n Course Project Report, 2017. [Online]. Available: <https://cs231n.stanford.edu/reports/2017/pdfs/420.pdf>
- [2] R. K. Sarvadevabhatla and R. Venkatesh Babu, "Freehand Sketch Recognition Using Deep Features," arXiv preprint arXiv:1502.00254, 2015. [Online]. Available: <https://arxiv.org/abs/1502.00254>
- [3] R. K. Sarvadevabhatla and R. Venkatesh Babu, "Freehand Sketch Recognition Using Deep Features," arXiv preprint arXiv:1502.00254, 2015. [Online]. Available: <https://arxiv.org/abs/1502.00254>
- [4] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention," IEEE Access, vol. 9, pp. 14078–14092, 2021. [Online]. Available: <https://scispace.com/pdf/classification-of-remote-sensing-images-using-efficientnet-1avtznhl.pdf>
- [5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.pdf
- [6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: <https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf>
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *European Conference on Computer Vision (ECCV) Workshops*, 2016. [Online]. Available: <https://arxiv.org/abs/1603.05027>
- [8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings / arXiv preprint*, 2019. [Online]. Available: <https://arxiv.org/pdf/1905.11946.pdf>
- [9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshine, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural

Information Processing Systems (NeurIPS), 2019, pp. 8024–8035.
[Online]. Available: <https://arxiv.org/abs/1912.01703>

- [10] Lukman Jibril Aliyu, Umar Sani Muhammad, Bilqisu Ismail, Nasiru Muhammad, Almustapha A Wakili, Seid Muhie Yimam, Shamsuddeen Hassan Muhammad, Mustapha Abdullahi, “Evaluating Deep Learning Models for African Wildlife Image Classification: From DenseNet to Vision Transformers”, 2025. Available: <https://arxiv.org/pdf/2507.213>
- [11] Doodle Dataset - Kaggle:
<https://www.kaggle.com/datasets/ashishjangra27/doodle-dataset/data>