

Klasifikacija tipova pirinča pomoću tehnika dubokog učenja

TEHNIČKI IZVEŠTAJ

Bojana Svilenković i Đorđe Nikolić | Duboko Učenje | Septembar 2022

Sadržaj

Uvod	2
Opis projekta.....	2
Korišćene tehnologije	2
Način validacije rezultata.....	2
Rad nad izdvojenih osobina	3
Analiza i preprocesiranje	3
Skaliranje	5
Optimizacija modela neuronske mreže	7
Kontrola pojave „overfitting“	7
Aktivacione funkcije.....	9
Konačni rezultati.....	10
Rad na slikama	11
Obrada slika	12
Kreiranje vanilla CNN modela.....	12
Korišćenje VGG-16 modela	14
Bibliografija	16

Uvod

Ovaj dokument predstavlja tehnički izveštaj i pregled projekta pod imenom „Klasifikacija tipova pirinča pomoću tehnika dubokog učenja“ i rađen je u okviru predmeta „Duboko učenje“ na Elektronskom fakultetu u Nišu.

OPIS PROJEKTA

Projekat obuhvata primenu algoritama dubokog učenja na dva povezana setova podataka koji se tiču 5 različite vrste pirinča. Oba seta podataka obuhvataju 75000 uzoraka, po 15000 za svaku vrstu pirinča. Jedan set podataka za uzorke ima slike pirinča, dok drugi sadrži po 106 izdvojenih osobina za svaku sliku. Tehnike pomoću kojih su ove osobine izdvojene nisu predmet ovog projekta, ali su opisane u već postojećem radu koji se tiče kreiranja ovih setova podataka [1]. Obzirom na različite tipove podataka u ova dva seta, biće primenjene adekvatne tehnike treniranja modela sa ciljem postizanja približnih rezultata u poređenju sa rezultatima koji su postignuti nad ovim podacima u okviru postojećih radova [2].

KORIŠĆENE TEHNOLOGIJE

Projekat je razvijan pomoću programskog jezika Python, verzije 3.10. Korišćene su sledeće biblioteke:

- Numpy (koristi se u ostalim bibliotekama kao matematička potpora)
- Pandas i Openpyxl (za učitavanje i upravljanje podacima)
- Seaborn i Matplotlib (za vizualizaciju podataka i rezultata)
- Sklearn (za preprocesiranje podataka i testiranje performansi modela)
- Keras i TensorFlow (za implementacije algoritama dubokog učenja)
- OpenCV (analizu i upravljanje slikama)
- Jupyter Notebook (za predstavljanje projekta u stilu dokumenta)

NAČIN VALIDACIJE REZULTATA

Obzirom na izbalansirane klase podataka, tačnost se nudi kao adekvatna metrika procene kvaliteta modela. Naravno, biće korišćene i druge metrike, naime preciznost, odziv, i F1 ocena, ali će se tačnost gledati kao glavna metrika ocenjivanja.

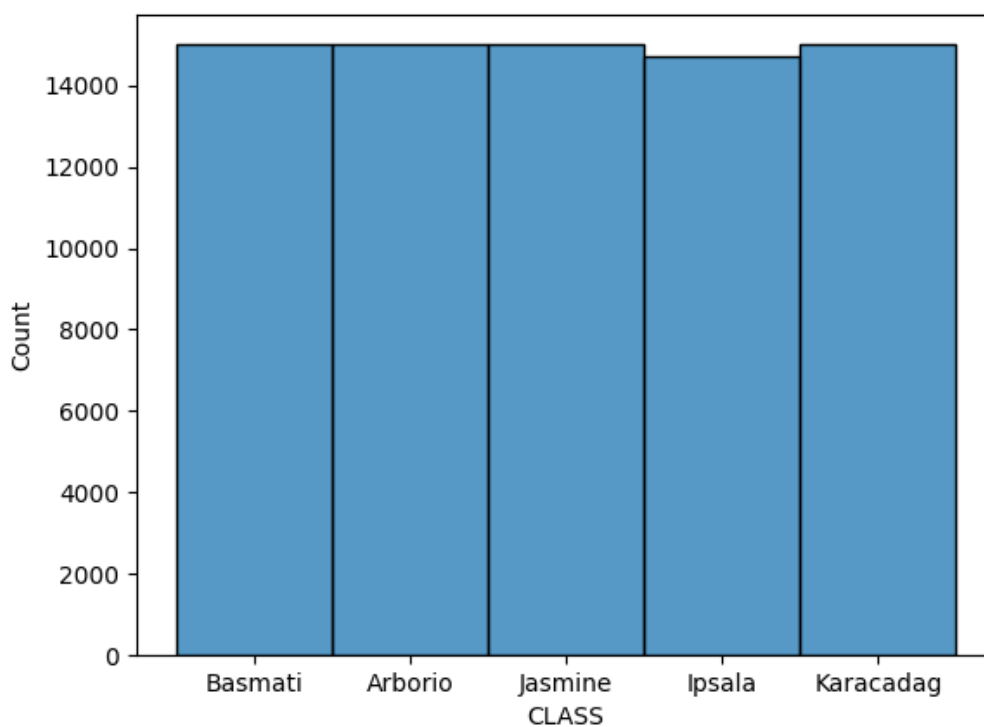
Gde je to moguće, tj. nije ometano predugim treniranjem modela, biće korišćena tehnika „Stratified K-Fold Cross Validation“ kako bi se dobile stabilne ocene kvaliteta modela.

Rad nad izdvojenih osobina

Kako bi stekli uvid u stanje podataka koji su nam na raspolaganju, biće urađene pojedine analize podataka, a zatim i čišćenje istih. Zatim, biće proverene performanse dobijene različitim metodama skaliranja podataka, korišćenjem prostih neuronskih mreža, ali i pomoću referentnih modela stabla odluka. Nakon što je najbolji način skaliranja utvrđen, prelazimo na optimizaciju modela neuronskih mreža. Na kraju, imaćemo uvid u to koji model daje najbolje performanse.

ANALIZA I PREPROCESIRANJE

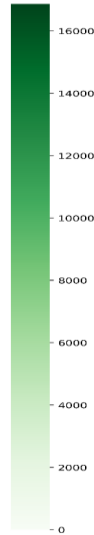
Kako je već pomenuto u uvodu, podaci obuhvataju 5 različitih klasa pirinča, i za svaku klasu postoje po 15000 uzoraka, dakle, klase su izbalansirane. Pronađeno je 8 uzoraka sa nepostojećim vrednostima pojedinih kolona, te su oni izbačeni. Zatim, utvrđeno je da u podacima postoje 296 duplikata uzoraka, te su i oni odbačeni. Na kraju ostaje set podataka sa 74696 uzoraka, čija je izbalansiranost klasa i dalje na veoma visokom nivou.



Slika 1. Broj uzoraka po klasama nakon čišćenja podataka

Nakon što su podaci očišćeni, dobro je proceniti količinu uzoraka koji su van ustaljenih granica, tj. „outliers“. To ćemo proceniti putem 3 različite metode: IQR, Z-Score i modifikovani Z-Score.

AREA -	12090	67	945.4
PERIMETER -	0	1	0
MAJOR_AXIS -	0	0	0
MINOR_AXIS -	0	0	0
ECCENTRICITY -	135	35	0
EQDIASO -	9554	8	3555
SOLIDITY -	722	534	276
CONVEX_AREA -	11692	68	8758
EXTENT -	39	0	0
ASPECT_RATIO -	107	25	312
ROUNDNESS -	0	0	0
COMPACTNESS -	0	0	0
SHAPEFACTOR_1 -	0	2	0
SHAPEFACTOR_2 -	0	0	0
SHAPEFACTOR_3 -	0	0	0
SHAPEFACTOR_4 -	1715	787	644
meanRR -	100	59	9
meanRG -	49	39	4
meanRB -	500	336	132
StdDevRR -	115	123	4
StdDevRG -	140	113	4
StdDevRB -	199	112	17
skewRR -	1091	502	387
skewRG -	2340	795	1212
skewRB -	423	327	51
kurtosisRR -	3598	1266	2815
kurtosisRG -	5389	1579	4632
kurtosisRB -	2307	1111	903
entropyRR -	13626	84	11847
entropyRG -	12108	75	10604
entropyRB -	6615	113	5081
meanH -	14847	0	16861
meanS -	110	149	40
meanV -	399	312	110
StdDevH -	2567	2043	4366
StdDevS -	1150	534	1029
StdDevV -	134	74	5
skewH -	2695	1081	1687
skewS -	729	725	742
skewV -	753	441	104
kurtosisH -	6663	1483	10691
kurtosisS -	2947	1273	2164
kurtosisV -	3372	1318	1651
entropyH -	1281	115	19
entropyS -	980	870	1570
entropyV -	1167	434	279
meanL -	74	61	7
meanA -	3332	2742	3056
meanB -	25	17	0
StdDevL -	150	106	5
StdDevA -	344	310	315
StdDevB -	851	414	2580
skewL -	1867	714	782
skewA -	815	372	126
skewB -	1933	425	517
kurtosisL -	4971	1487	3963
kurtosisA -	3640	1142	2780
kurtosisB -	4054	1423	3435
entropyL -	12334	72	10508
entropyA -	12779	93	9884
entropyB -	13414	46	11845
meanY -	80	66	8
meanCb -	34	22	0
meanCr -	7225	3351	5483
StdDevY -	159	120	5
StdDevCb -	856	430	2142
StdDevCr -	4058	1081	1751
skewY -	1775	698	754
skewCb -	2759	393	771
skewCr -	4273	1419	3275
kurtosisY -	4793	1450	3751
kurtosisCb -	4241	1403	3489
kurtosisCr -	5434	882	5309
entropyY -	12360	78	10473
entropyCb -	5822	79	3361
entropyCr -	13610	86	11421
meanXX -	65	50	3
meanYY -	38	26	1
meanZZ -	268	210	50
StdDevXX -	208	249	11
StdDevYY -	163	243	11
StdDevZZ -	272	163	13
skewXX -	2063	775	1054
skewYY -	2599	855	1481
skewZZ -	858	526	247
kurtosisXX -	4021	1382	3201
kurtosisYY -	4652	1514	4220
kurtosisZZ -	2434	1175	1005
entropyXX -	6780	837	4159
entropyYY -	5157	1333	2829
entropyZZ -	2768	719	709
ALLdaub4RR -	100	59	9
ALLdaub4RG -	49	39	4
ALLdaub4RB -	496	335	129
ALLdaub4H -	14853	0	16844
ALLdaub4S -	110	149	39
ALLdaub4V -	397	308	104
ALLdaub4L -	74	61	7
ALLdaub4a -	3334	2743	3058
ALLdaub4b -	25	17	0
ALLdaub4Y -	80	66	8
ALLdaub4Cb -	34	22	0
ALLdaub4Cr -	7223	3352	5486
ALLdaub4XX -	65	49	3
ALLdaub4YY -	38	26	1
ALLdaub4ZZ -	268	210	49
IQR		ZScore	ModZScore



Slika 2. Broj uzoraka van prosečnih vrednosti

U zavisnosti od korišćenog algoritma, dobijene su dosta različite količine *outlier-a*, gde su IQR i modifikovani Z-Score dali dosta visok broj *outlier-a*, a običan Z-Score daje dosta manji broj. Svakako, ustanovljeno je da broj *outlier-a* u podacima nije zanemarljiv.

SKALIRANJE

Obzirom na to da raspon vrednosti kolona nije standardizovan u datim podacima, potrebno je skalirati ih na neki način, kako bi neuronska mreža brže konvergirala. Isprobano je više metoda skaliranja, naime:

- Min Max normalization
- Standardization
- Robust Scaling
- Quantile (Uniform) transform
- Quantile (Normal) transform
- Power Transform

Svaka od ovih metoda skaliranja je isprobana sa referentnim modelom stabla odlučivanja, kao i prostim modelom neuronske mreže. Karakteristike datog modela su sledeće:

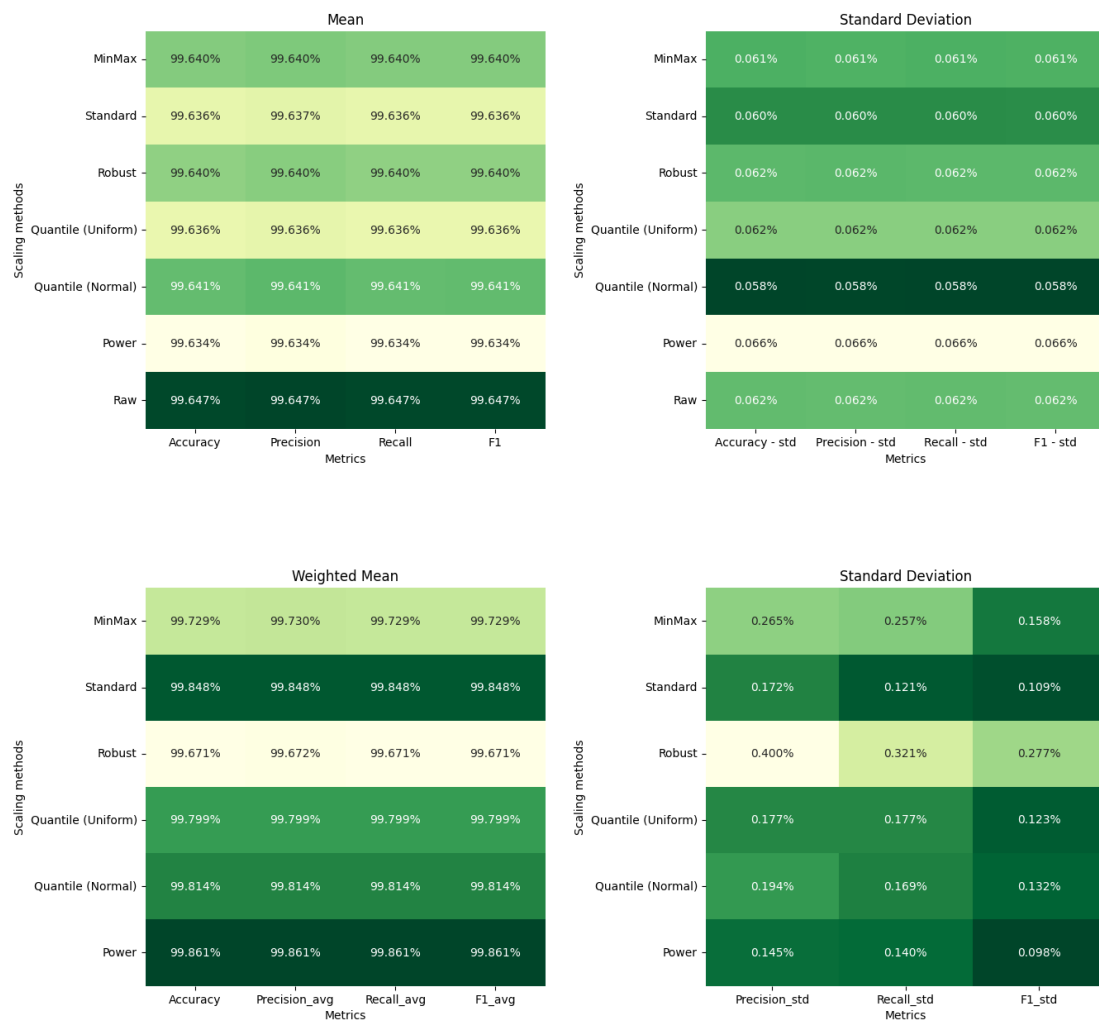
- 1 skriven sloj
- 80 potpuno povezanih neurona po sloju
- „RELU“ aktivaciona funkcija (sem izlaznog sloja gde je aktivaciona funkcija „softmax“)
- Optimizator ADAM
- Računanje gubitka putem *categorical crossentropy*
- Veličina serije jednaka 100
- 3 epohe

Za treniranje i testiranje referentnih modela stabla odlučivanja korišćena je metoda „Stratified 10-Fold Cross Validation“, proces je ponovljen 10 puta i konačni rezultati predstavljaju prosek. Za testiranje modela neuronske mreže, nije korišćena prethodno navedena metoda, ali je proces takođe ponovljen 10 puta.

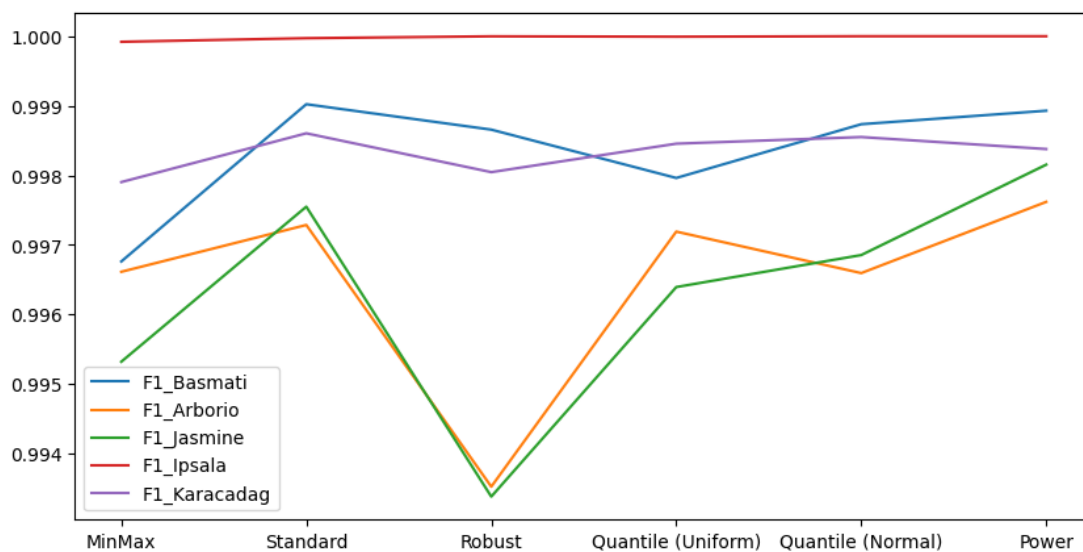
Na slici 3, moguće je videti rezultate testiranja, gde gornji red rezultata predstavlja ocene kvaliteta referentnog modela stabla odlučivanja, dok donji red predstavlja ocene kvaliteta modela neuronske mreže.

Rezultati su pokazali da najbolje performanse daju podaci skalirani putem standardizacije i *power transform-a*. Obzirom da će nam zbog testiranja određenih optimizacija biti potreban standardizovan set podataka, on će biti korišćen na dalje, ali na kraju rada, kada utvrdimo optimalne parametre, biće isproban i drugi set podataka dobijen putem *power transform* metode.

Comparing Scaling Methods



Slika 3. Rezultati testiranja različitih metoda skaliranja



Slika 4. Prikaz F1 ocena modela neuronske mreže po različitim metodama skaliranja

OPTIMIZACIJA MODELA NEURONSKE MREŽE

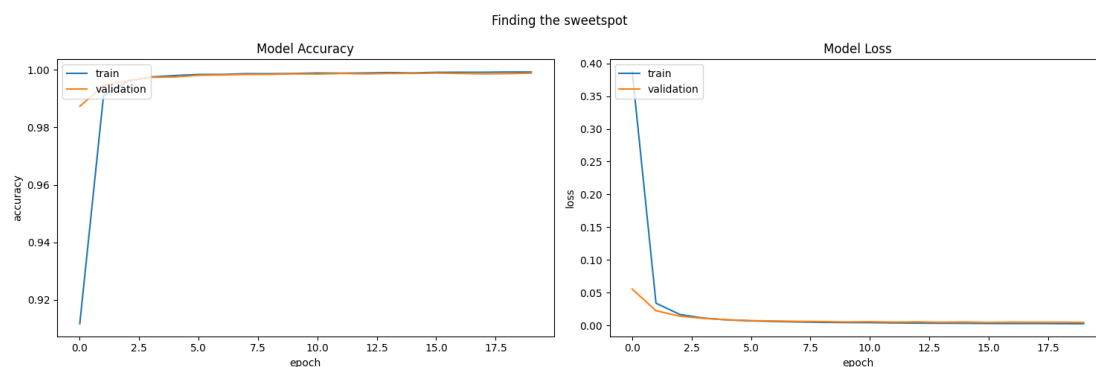
Kako bi postigli što bolje performanse model biće izvršeno više metoda optimizacija, i na kraju ćemo kombinovati one koje daju najbolje performanse.

Kontrola pojave „overfitting“

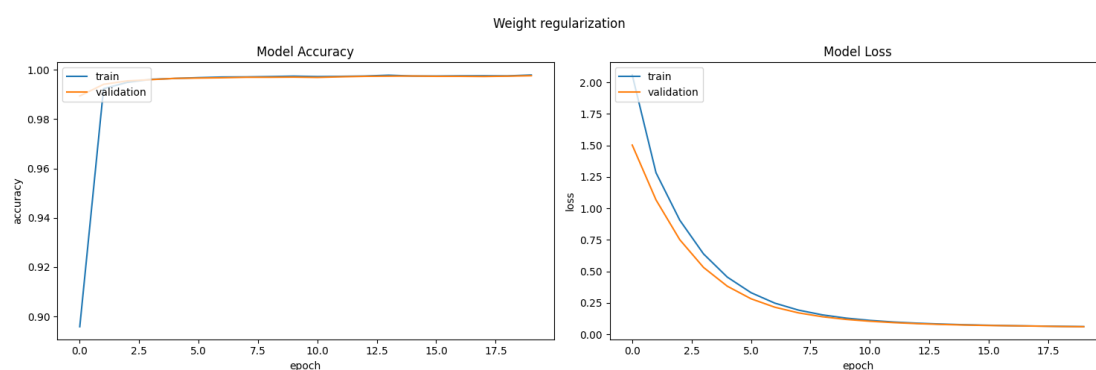
Sa željom smanjenja pojave *overfitting-a* (situacije kod koje model daje dobre rezultate za trening podatke ali ne i za realne podatke), biće isprobane nekoliko metoda, uključujući:

- Regularizacija težina (L2, faktor 0.01)
- Rano zaustavljanje prema gubitku
- „Dropout“ slojevi

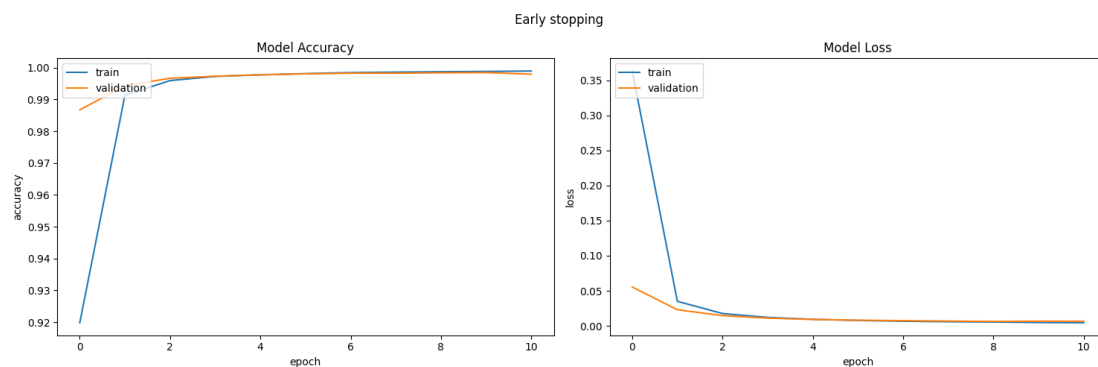
Osobine modela koji se treniraju i testiraju ostaju iste, sem toga što će u ovom slučaju veličina serije biti 1000 i broj epoha biti 20. Podaci se jednom dele po pravilu da 30% podataka ostaje za testiranje, ali se proces ne ponavlja više puta.



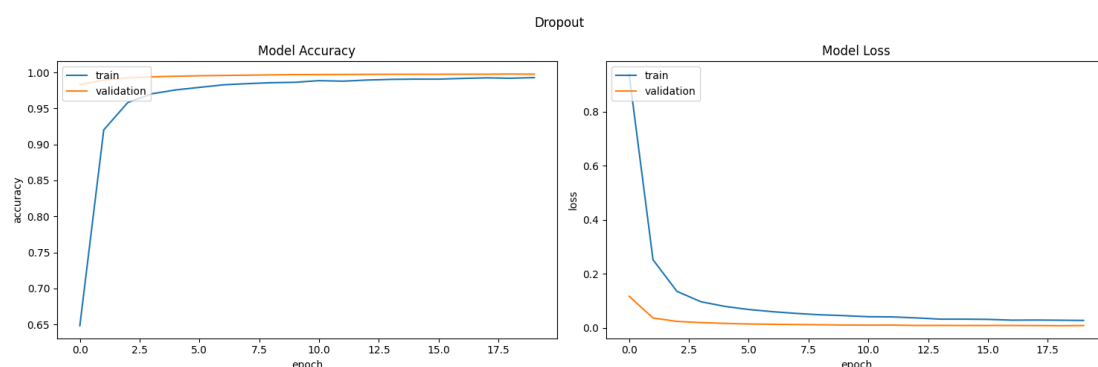
Slika 5. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela bez optimizacije



Slika 6. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa regularizacijom težina



Slika 7. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa tehnikom ranog zaustavljanja



Slika 7. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa tehnikom *dropout* slojeva

Weighted Mean					Standard Deviation					
Methods		Accuracy					Precision_std			
		Metrics					Metrics			
		Unmodified	Weighted	EarlyStopping	Dropout		Unmodified	Weighted	EarlyStopping	Dropout
		Accuracy	Precision_avg	Recall_avg	F1_avg		Precision_std	Recall_std	F1_std	
		99.848%	99.848%	99.848%	99.848%		0.172%	0.121%	0.109%	
		99.741%	99.742%	99.741%	99.741%		0.348%	0.198%	0.188%	
		99.844%	99.844%	99.844%	99.844%		0.157%	0.108%	0.092%	
		99.808%	99.808%	99.808%	99.808%		0.187%	0.113%	0.121%	

Slika 8. Rezultati testiranja performansi modela dobijenih kroz različite metode optimizacije

Gledajući rezultate, može se zaključiti da korišćenjem regularizacije težina i *dropout* slojeva modeli gube na tačnosti, dok tehnika ranog zaustavljanja ne utiče na tačnost u velikoj meri. Što je i logično, obzirom na to da ta tehnika samo zaustavlja treniranje kada u optimalnom momentu. Doduše, gledajući krivu gubitka kod modela koji ne koriste regularizaciju težina, čini se da model prebrzo konvergira, i to posle samo 1 epohe, dok je kod modela sa regularizacijom težina to dosta sporije, tek oko desete epohe. Tokom dalje optimizacije, biće korišćena tehnika ranog zaustavljanja, a na kraju će biti isprobana i kombinacija sa regularizacijom težina.

Aktivacione funkcije

Kako bi se utvrdilo koje aktivacione funkcije daju najbolje rezultate, biće isprobane čak 11 različite funkcije za unutrašnje slojeve, i dve različite funkcije za izlazni sloj. Razlike u odnosu na osnovni model koji je ranije opisan su male:

- Koristi se tehnika ranog zaustavljanja
- Veličina serija je 1000
- 20 epoha

Functions	Different activation functions testing results						
	F1_avg	Accuracy	F1_Basmati	F1_Arborio Metrics	F1_Jasmine	F1_Ipsala	F1_Karacadag
ReLU	99.900%	99.900%	99.917%	99.840%	99.823%	100.000%	99.920%
Parametric ReLU	99.900%	99.900%	99.920%	99.833%	99.813%	100.000%	99.933%
Leaky ReLU (Sigmoid)	99.897%	99.897%	99.917%	99.830%	99.817%	100.000%	99.923%
SELU (Sigmoid)	99.894%	99.894%	99.910%	99.830%	99.803%	99.997%	99.933%
Tanh	99.892%	99.892%	99.910%	99.827%	99.817%	99.997%	99.910%
ELU (Sigmoid)	99.890%	99.890%	99.910%	99.820%	99.810%	100.000%	99.913%
ELU	99.889%	99.889%	99.903%	99.827%	99.800%	100.000%	99.917%
GeLU	99.889%	99.889%	99.913%	99.810%	99.807%	100.000%	99.917%
Tanh (Sigmoid)	99.888%	99.888%	99.907%	99.817%	99.797%	99.997%	99.923%
ReLU (Sigmoid)	99.886%	99.886%	99.920%	99.800%	99.793%	99.997%	99.923%
SELU	99.881%	99.881%	99.893%	99.817%	99.804%	99.997%	99.897%
GeLU (Sigmoid)	99.881%	99.881%	99.913%	99.790%	99.790%	100.000%	99.913%
Softsign (Sigmoid)	99.877%	99.877%	99.883%	99.817%	99.773%	100.000%	99.913%
Softsign	99.877%	99.877%	99.873%	99.827%	99.763%	99.997%	99.927%
Leaky ReLU	99.876%	99.875%	99.903%	99.793%	99.794%	100.000%	99.890%
Parametric ReLU (Sigmoid)	99.874%	99.874%	99.890%	99.803%	99.790%	99.997%	99.893%
Softplus (Sigmoid)	99.869%	99.866%	99.873%	99.803%	99.757%	99.997%	99.917%
Softplus	99.861%	99.861%	99.887%	99.773%	99.753%	99.997%	99.897%
Softmax (Sigmoid)	99.837%	99.837%	99.863%	99.737%	99.717%	100.000%	99.870%
Softmax	99.821%	99.821%	99.847%	99.716%	99.674%	100.000%	99.870%
Sigmoid (Sigmoid)	99.819%	99.819%	99.793%	99.763%	99.647%	99.997%	99.900%
Sigmoid	99.811%	99.811%	99.786%	99.756%	99.634%	99.997%	99.887%

Slika 9. Rezultati testiranja različitih aktivacionih funkcija

Kako je ustanovljeno da većina aktivacionih funkcija daje slične rezultate, kako bi se dobili stabilni rezultati korišćena je tehnika „Stratified 10-Fold Cross Validation“. Rezultati su pokazali da *ReLU* i *PReLU* daju identične rezultate, kada se primene na unutrašnje slojeve, a *softmax* funkcija daje najbolje rezultate kada se koristi na izlazni sloj.

KONAČNI REZULTATI

Sa rezultatima koji su do sada dobijeni, mogu se doneti određeni zaključci koji nalažu kako je najbolje pripremiti model za date podatke. Ostalo je još nekoliko kombinacija koje se mogu proveriti, a to su:

- Sa ili bez regularizacije težina
- Standardizovani podaci, ili podaci skalirani putem *power transform* metode

Izgled konačne neuronske mreže pomoću koje će gorepomenute kombinacije biti testirane:

- 1 skriven sloj
- 80 potpuno povezanih neurona po sloju
- „PReLU“ aktivaciona funkcija (sem izlaznog sloja gde je aktivaciona funkcija „softmax“)
- Optimizator ADAM
- Računanje gubitka putem *categorical crossentropy*
- Veličina serije jednaka 1000
- 20 epohe
- Koristi se tehnika ranog zaustavljanja

Final testing results						
Standardized data + NO weight reg	99.886%	99.886%	99.903%	99.817%	99.800%	100.000%
Power transformed data + NO weight reg	99.882%	99.882%	99.917%	99.790%	99.790%	99.997%
Standardized data + weight reg	99.787%	99.787%	99.813%	99.667%	99.614%	99.997%
Power transformed data + weight reg	99.802%	99.802%	99.836%	99.666%	99.634%	100.000%
	F1_avg	Accuracy	F1_Basmati	F1_Arborio Metrics	F1_Jasmine	F1_Ipsala
						F1_Karacadag

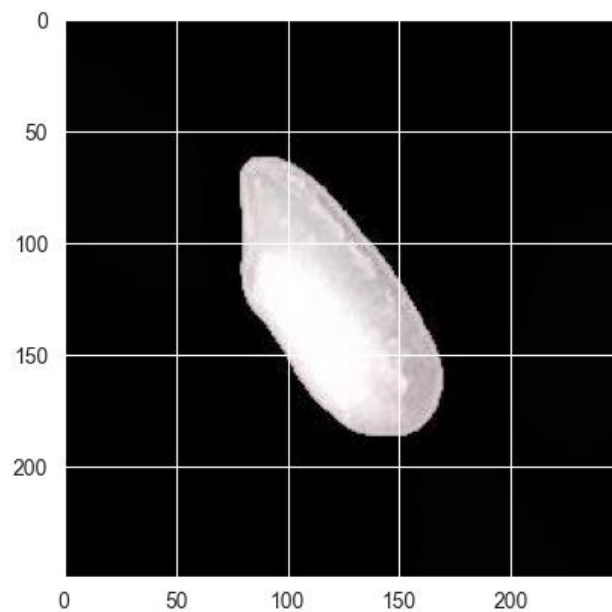
Slika 10. Konačni rezultati

Kao referentni rezultati uzimaju se vrednosti koje su dobijene u radu koji se pominje na početku ovog izveštaja [2], a ti rezultati su tačnost od 99.87%, koristeći običnu veštačku neuronsku mrežu, a 99.95% koristeći duboku veštačku neuronsku mrežu sa čak 19 skrivenih slojeva. Obzirom na to da su u okviru ovog projekta korišćene samo obične neuronske mreže sa samo jednim skrivenim slojem, dobijeni rezultati su i više nego adekvatni. Još jedan faktor koji nije modifikovan u okviru ovog projekta je sam optimizator neuronske mreže. Korišćen je samo ADAM algoritam, a SGD algoritam (engl. stochastic gradient descent) sa stepenom učenja kao parametrom nije isprobavan jer je sam ADAM davao dobre rezultate. Sigurno je da bi korišćenjem dubljih neuronskih mreža kao i optimizovanjem stepena učenja, u potencijalnom nastavku projekta moguće dostići, a možda i nadmašiti referentne performanse modela.

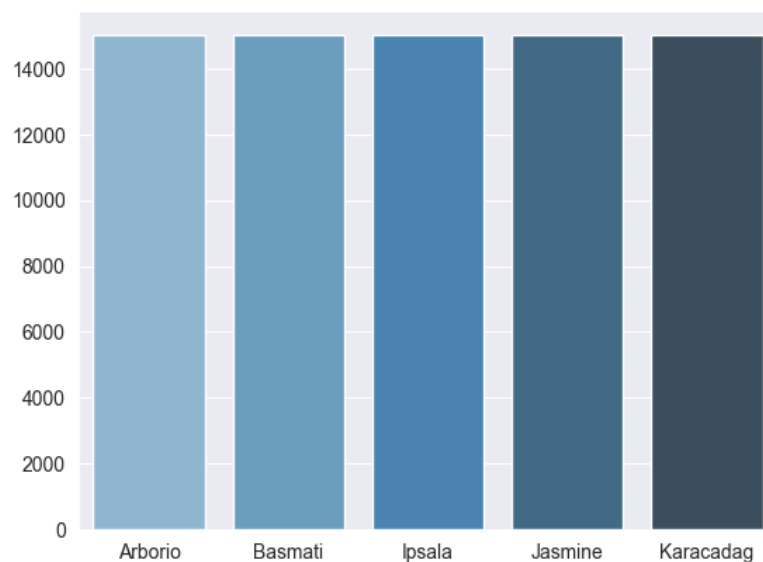
Kod pomoću kog je izvršeno prethodno istraživanje se nalazi u fajlu *FeatureAnalysis.ipynb*. U folderu *images/feature_analysis* se mogu naći sve slike i generisane figure.

Rad na slikama

Originalni podaci su sačinjeni od slika dimenzija 250 sa 250 piksela, od kojih postoje po 15000 za svaku klasu pirinča (Arborio, Basmati, Ipsala, Jasmine i Karacadag).



Slika 11. Primer slike pirinča



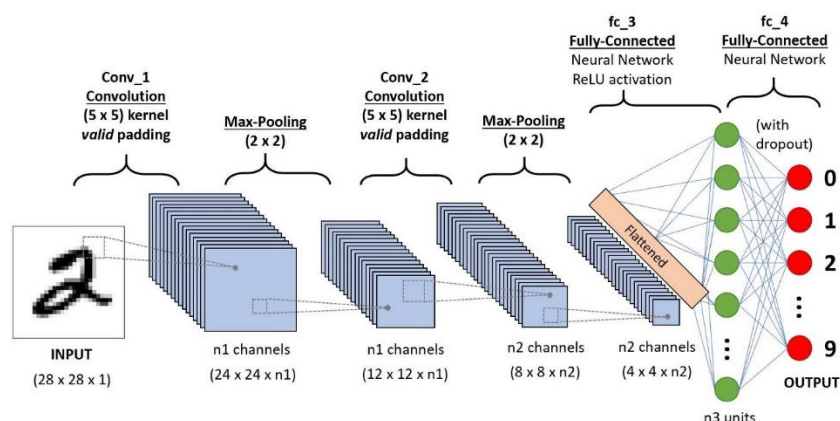
Slika 12. Broj slika po svakoj kategoriji

OBRADA SLIKA

Slike su pre treniranja modela obrađene tako da njihove nove dimenzije budu 175 sa 175 piksela, određena je veličina serije (engl. batch size) od 128 i podeljene su slike u tri grupe. Slike za treniranje, slike za testiranje, i slike za validaciju, od kojih slika za treniranje ima 52500 (70% originalnog skupa), a za testiranje i validaciju po 11250 slika (15% originalnog skupa).

KREIRANJE VANILLA CNN MODELA

Kako bi se trenirao model neuronske mreže preko same slike, biće korišćen poseban tip neuronske mreže pod imenom konvolucijske neuronske mreže.



Slika 12. Primer konvolucijske neuronske mreže

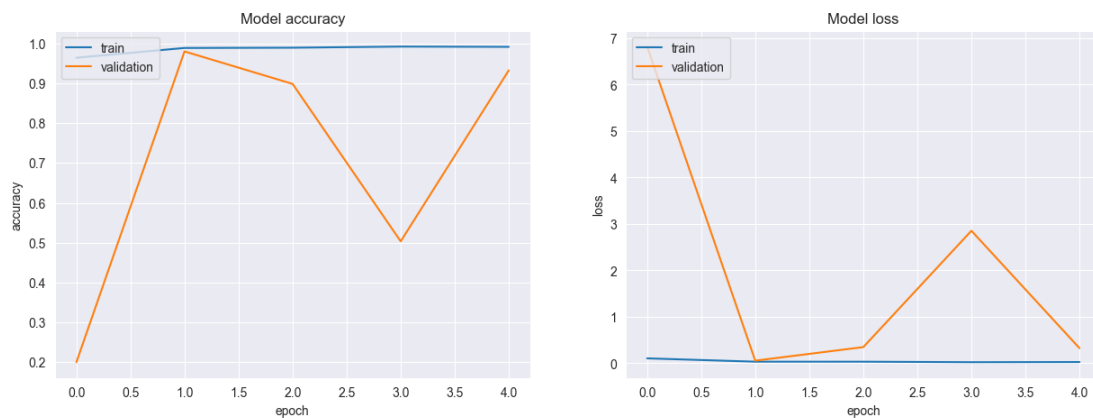
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 173, 173, 32)	896
batch_normalization (BatchNormalization)	(None, 173, 173, 32)	128
conv2d_1 (Conv2D)	(None, 173, 173, 32)	9248
batch_normalization_1 (BatchNormalization)	(None, 173, 173, 32)	128
max_pooling2d (MaxPooling2D)	(None, 87, 87, 32)	0
dropout (Dropout)	(None, 87, 87, 32)	0
conv2d_2 (Conv2D)	(None, 87, 87, 64)	18496
batch_normalization_2 (BatchNormalization)	(None, 87, 87, 64)	256
conv2d_3 (Conv2D)	(None, 87, 87, 64)	36928
batch_normalization_3 (BatchNormalization)	(None, 87, 87, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 44, 44, 64)	0
dropout_1 (Dropout)	(None, 44, 44, 64)	0
conv2d_4 (Conv2D)	(None, 44, 44, 128)	73856
batch_normalization_4 (BatchNormalization)	(None, 44, 44, 128)	512
conv2d_5 (Conv2D)	(None, 44, 44, 128)	147584
batch_normalization_5 (BatchNormalization)	(None, 44, 44, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 22, 22, 128)	0
dropout_2 (Dropout)	(None, 22, 22, 128)	0
flatten (Flatten)	(None, 61952)	0
dense (Dense)	(None, 512)	31719936
batch_normalization_6 (BatchNormalization)	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 128)	65664
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 5)	645
Total params: 32,077,093		
Trainable params: 32,075,173		
Non-trainable params: 1,920		

Slika 13. Slojevi kreirane konvolucijske neuronske mreže

Kreirana je mreža sa 24 skrivena sloja, korišćenjem:

- konvolucijskih slojeva
- slojeva normalizacije serija (batch normalization)
- *dropout* slojeva
- potpuno povezanih slojeva.

Takođe se koristi tehnika ranog zaustavljanja (sa pogledom na vrednost gubitka), kao i tehnika smanjivanja stope učenja kada vrednost gubitka stagnira. Model je treniran kroz 8 epoha, ali je treniranje zaustavljeno na petoj epohi i postignuta tačnost je bila 99.17%.



Slika 14. Rast i odnos tačnosti i gubitka kroz epohe treniranja vanilla CNN modela

	precision	recall	f1-score	support
Arborio	0.955250	0.986667	0.970704	2250.000000
Basmati	1.000000	0.965778	0.982591	2250.000000
Ipsala	1.000000	0.990667	0.995311	2250.000000
Jasmine	0.951777	1.000000	0.975293	2250.000000
Karacadag	0.998148	0.958222	0.977778	2250.000000
accuracy	0.980267	0.980267	0.980267	0.980267
macro avg	0.981035	0.980267	0.980335	11250.000000
weighted avg	0.981035	0.980267	0.980335	11250.000000

Slika 15. Ocene performansi vanilla CNN modela

KORIŠĆENJE VGG-16 MODELA

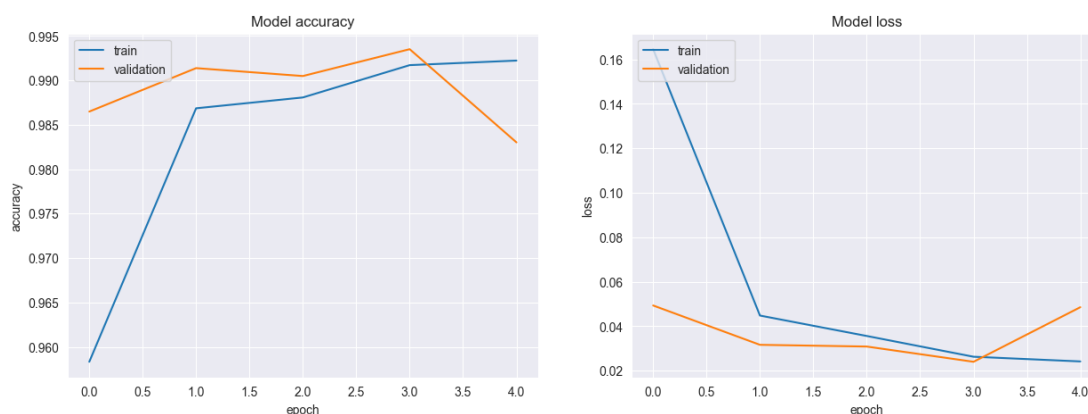
Nakon što je istestiran vanilla CNN model, isproban je već treniran VGG-16 model, koji je zatim dodatno treniran na slikama pirinča, zajedno sa tehnikama ranog zaustavljanja i smanjenja stope učenja prilikom stagniranja gubitka. Izvršeno je 5 epoha treniranja, i opet se na poslednjoj aktiviralo rano zaustavljanje, a maksimalna tačnost je bila 99.22%.

Da se primetiti da korišćenjem CNN modela na slikama nije dostignuta bolja tačnost od prostih modela koji su radili sa osobinama. Obzirom da su rezultati koji su postignuti CNN modelima u radu koji je pomenut na početku dokumenta, nesumnjivo je da su bolje performanse moguće. Shodno tome, postoji dosta prostora za dalju diskusiju i unapređenje modela koji su razvijeni u okviru ovog projekta.

Kod pomoću kog je izvršeno prethodno istraživanje se nalazi u fajlu *ImagesAnalysis.ipynb*.

Model: "model"		
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 175, 175, 3)]	0
vgg16 (Functional)	(None, 5, 5, 512)	14714688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense_3 (Dense)	(None, 1024)	525312
dense_4 (Dense)	(None, 5)	5125
Total params: 15,245,125		
Trainable params: 530,437		
Non-trainable params: 14,714,688		

Slika 16. Prošireni VGG-16 model



Slika 17. Rast i odnos tačnosti i gubitka kroz epohe treniranja VGG-16 modela

	precision	recall	f1-score	support
Arborio	0.955250	0.986667	0.970704	2250.000000
Basmati	1.000000	0.965778	0.982591	2250.000000
Ipsala	1.000000	0.990667	0.995311	2250.000000
Jasmine	0.951777	1.000000	0.975293	2250.000000
Karacadag	0.998148	0.958222	0.977778	2250.000000
accuracy	0.980267	0.980267	0.980267	0.980267
macro avg	0.981035	0.980267	0.980335	11250.000000
weighted avg	0.981035	0.980267	0.980335	11250.000000

Slika 18. Ocene performansi treniranog VGG-16 modela

Bibliografija

- [1] I. Cinar i M. Koklu, „Determination of Effective and Specific Physical Features of Rice Varieties by Computer Vision In Exterior Quality Inspection,“ *Selcuk Journal of Agriculture and Food Sciences*, t. 36, br. 2, 2022.
- [2] M. Koklu, I. Cinar i Y. Selim Taspinar, „Classification of rice varieties with deep learning methods,“ *Computers and Electronics in Agriculture*, t. 187, 2021.