

NEURALNE MREŽE
PRVI PROJEKTNI ZADATAK-DUBOKO UČENJE

Tanja Kvaščev 0031/21

Sanja Drobnjak 0492/21

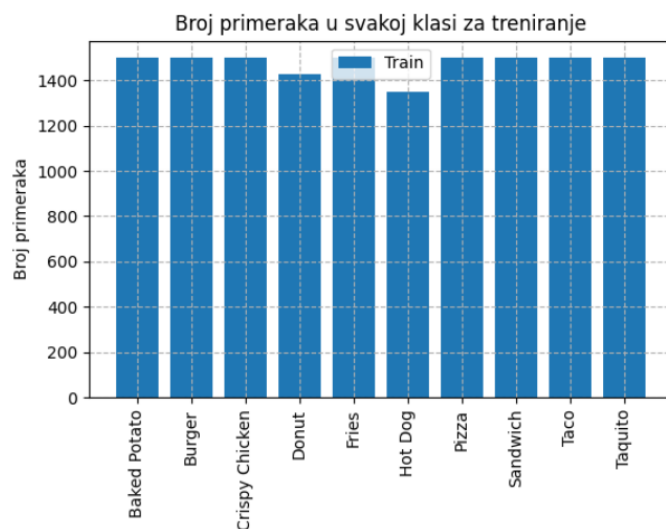
Kratak opis problema koji se rešava za odabran dataset

U ovom projektu obrađen je problem klasifikacije brze hrane. Klase predstavljaju vrste brze hrane. Postoji deset klasa i one su: „Baked Potato“, „Burger“, „Crispy Chicken“, „Donut“, „Fries“, „Hot Dog“, „Pizza“, „Sandwich“, „Taco“ i „Taquito“. Ulazni podaci su slike i podeljeni su na tri skupa: Train, Valid i Test. Broj odbiraka svake od klasa u svakom od pomenutih podskupova su prikazani dalje u izveštaju.

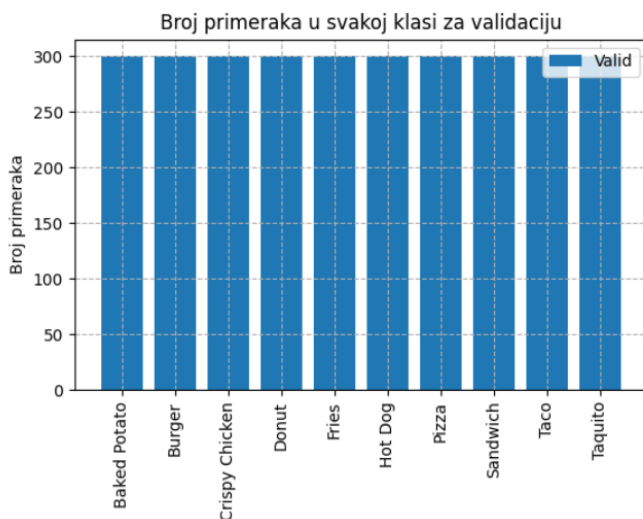
Cilj zadatka nam je da obučimo neuralnu mrežu na taj način da zadata nepoznata slika bude ispravno klasifikovana u odgovarajuću grupu.

Prikaz grafika-broj odbiraka svake klase

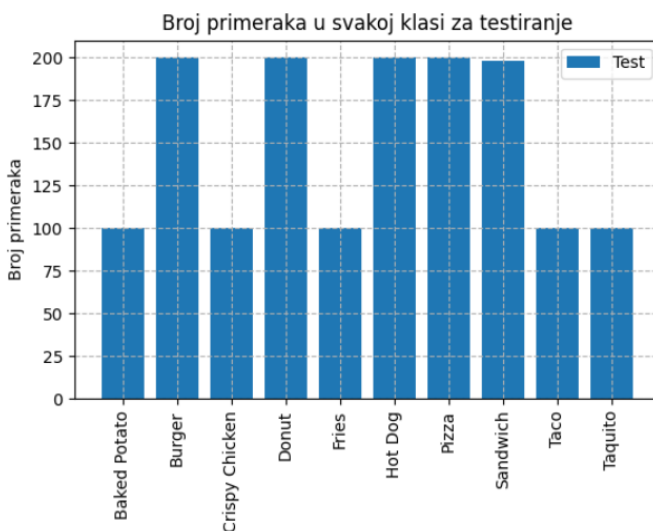
Podaci za treniranje su balansirani, jer svaka klasa za treniranje sadrži isti broj primeraka, što se vidi i na grafiku ispod.



Takođe, podaci za validaciju su isto balansirani, što se može i videti na grafiku:

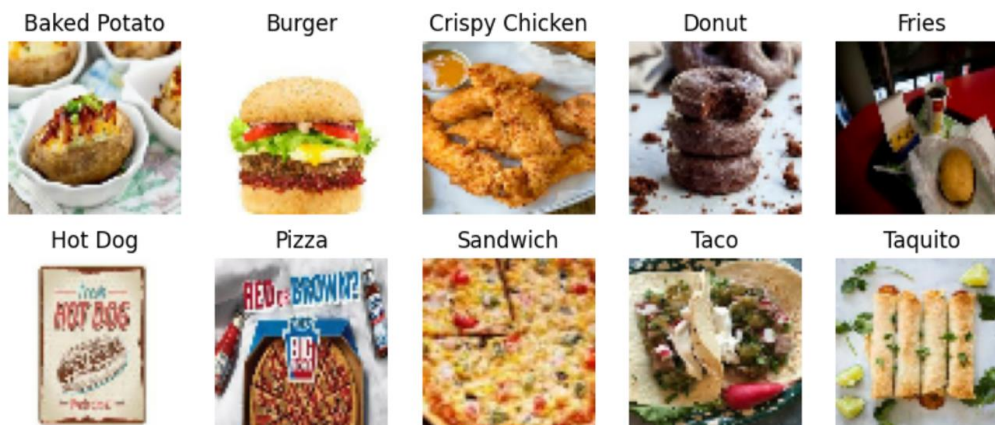


Sa druge strane, podaci za testiranje nisu balansirani jer neke klase imaju duplo više primeraka u odnosu na druge, kao što je prikazano na grafiku:



Balansiranje skupa za testiranje ovde nije neophodno, jer skup za testiranje služi za procenu performansi modela na realnim podacima, a ne za obučavanje modela.

Po jedan primer podataka iz svake klase



Podela podataka na odgovarajuće skupove

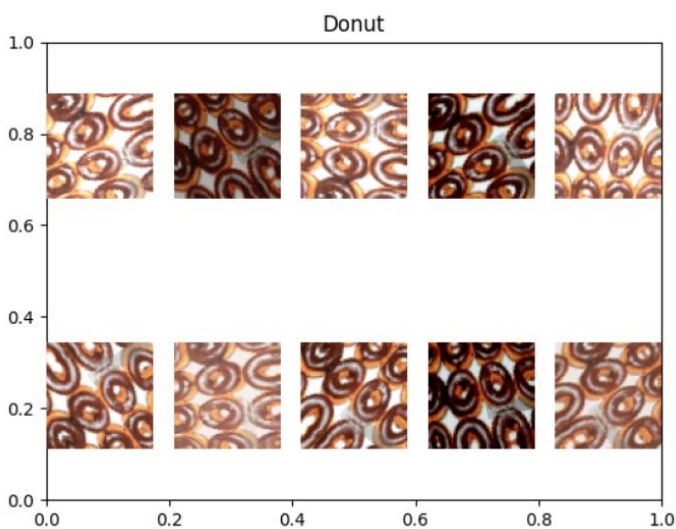
U našem dataset-u, podaci su već podeljeni na tri skupa: podaci za treniranje, podaci za validaciju i podaci za testiranje. Imajući u vidu da su naši podaci već podeljeni, nije bilo neophodno da koristimo parametar *validation_split* i *subset* unutar funkcije *image_dataset_from_directory* (oni omogućavaju automatsko razdvajanje podataka na podskupove unutar funkcije).

Podela podataka na odgovarajuće skupove-za treniranje, validaciju i testiranje, je ključna u procesu obučavanja modela. Na taj način se sprečava preobučavanje, omogućava generalizaciju na novim podacima i pomaže se u proceni performansi modela. Pomoću skupa za treniranje model uči tako

što prilagođava svoje parametre kako bi greška na trening uzorcima bila minimalna. Skup za validaciju sprečava modelu da dođe do preobučavanja tako što omogućava validaciju modela na nezavisnom skupu podataka i omogućava generalizaciju modela (da znamo da li je naš model primenljiv na novom skupu podataka). Pomoću skupa za testiranje možemo da obavimo konačno procenjivanje performansi modela nakon obučavanja, što postizemo skupom podataka koje model nije video u fazi obučavanja.

Izvršavanje predprocesiranja podataka

Augmentacija je u našem slučaju postignuta pomoću različitih operacija nad slikama, odnosno pomoću operacije nasumičnog obrtanja slike po vertikalnoj osi, operacije nasumične rotacije (od maksimalno 0,25 radijana u smeru kazaljke na satu ili obrnuto), operacije nasumične procene osvetljenja slika (maksimalno 25%) i operacije nasumične promene kontrasta slika (maksimalno 25%). Augmentacija omogućava stvaranje varijacija u skupu za treniranje, sa ciljem poboljšanja performansi modela. Ona se primenjuje nasumično na svaku sliku tokom svake epohe generalizacije i treniranja kako bi se modelu neuralne mreže omogućilo učenje raznovrsnim podacima. Augmentacija doprinosi poboljšanju i generalizaciji modela, odnosno da model može da radi dobro i na sebi nepoznatom novom skupu podataka.



Skaliranje se takođe primenjuje na sve slike koje prolaze kroz model tokom obučavanja, validacije i testiranja, kako bi se osigurali da svi pikseli imaju vrednosti u opsegu koji model može efikasno da obradi tokom optimizacije i predviđanja.

Formiranje i obučavanje neuralne mreže

Kriterijumska funkcija (loss function) je funkcija koja meri razliku između stvarnih i predviđenih vrednosti modela. Cilj nam je da što manje smanjimo ovu razliku, odnosno da nam predviđanja budu što tačnija. Odabrale smo *Sparse Categorical Crossentropy* koja se često koristi kod problema klasifikacije sa više klasa. Ova funkcija je korisna kada treba da radimo sa ciljnim promenljivima koje nisu predstavljene kao one-hot kodirani vektor, već kao niz celih brojeva. Na primer, umesto

da imamo vektor $[0,1,0]$ za klasu dva, u našem slučaju klasu možemo jednostavno predstaviti brojem 2. Zbog ovoga je implementacija brža i efikasnija jer nema potrebe za konverzijom ciljnih vektora u one-hot kodiranje pre treniranja.

Funkcija aktivacije neurona je matematička operacija koja se primenjuje na ulaznim podacima kako bi se dobio određeni izlaz. *ReLU* je jedna od najčešće korišćenih funkcija aktivacije. Definiše se kao $f(x)=\max(0,x)$, odnosno izlaz je nula za sve nepozitivne vrednosti ulaza, a za pozitivne vrednosti izlaz je upravo ta pozitivna vrednost. Odabrale smo ReLU funkciju jer je brza i efikasna, a i rešava probleme nestajućeg i eksplodirajućeg gradijenta koji se javlja kod nekih drugih aktivacionih funkcija, npr. sigmoida. ReLU doduše ima problem pojavljivanja mrtve zone, koji dovodi do toga da neki neuroni postaju neaktivni tokom treniranja. Ovo se može rešiti upotrebom npr. Leaky ReLU-a ($f(x)=\max(0.01*x,x)$).

Metoda optimizacije kriterijumske funkcije je od velikog značaja, jer igra važnu ulogu u prilagođavanju težina mreže kako bi se minimizovala vrednost kriterijumske funkcije. Odabrale smo metodu *Adam* jer se ona često koristi u praksi, jer daje dobre performanse i brzo konvergira tokom treniranja.

Naš model ima arhitekturu koja uključuje konvolucione slojeve (*Conv2D*), slojeve za pooliranje (*MaxPooling2D*), dropout sloj (*Dropout*), sloj za ravnanje (*Flatten*) i potpuno povezane slojeve (*Dense*).

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 64, 64, 3)	0
rescaling (Rescaling)	(None, 64, 64, 3)	0
conv2d (Conv2D)	(None, 64, 64, 16)	448
max_pooling2d (MaxPooling2D)	(None, 32, 32, 16)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
dropout (Dropout)	(None, 16, 16, 64)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 128)	2097280
dense_1 (Dense)	(None, 10)	1290

```
=====
Total params: 2122154 (8.10 MB)
Trainable params: 2122154 (8.10 MB)
Non-trainable params: 0 (0.00 Byte)
=====
```

Definisanje preobučavanja neuralne mreže

Preobučavanje je pojava kada se mreža za date podatke obučila dobro, ali se ne može primeniti na nekom drugom skupu podataka. Drugim rečima, desilo se da se mreža previše prilagodila datom skupu podataka. Još jedna karakteristika je da je za malu promenu ulaznih podataka stepen greške dosta veliki.

U ovom zadatku primenjen je *dropout*. Dropout je metoda zaštite od preobučavanja pri kojoj tokom treniranja nasumično isključujemo određene neurone i veze između njih. Na ovaj način se sprečava da se mreža previše osloni na određene putanje. Tokom svake iteracije, svaki neuron ima verovatnoću p da se isključi (tu veličinu ćemo zadati kao hiperparametar prilikom obučavanja mreže). Isključen neuron ne doprinosi obučavanju mreže.

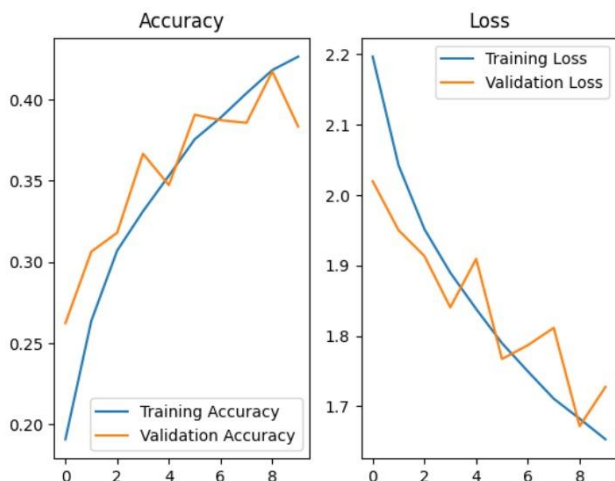
Konkretno, u našoj mreži smo dodale Dropout sloj sa parametrom 0.3, što znači da će tokom treniranja biti isključeno 30% neurona.

Prikaz finalno obučenog grafika

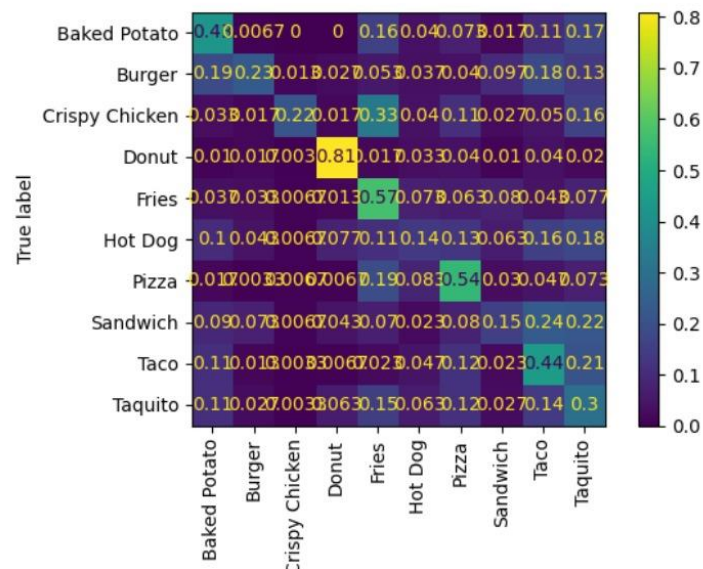
Proces obučavanja neuralnih mreža se obično sastoji od iterativnog prolaska kroz skup podataka, odnosno od epoha, gde model prilagođava svoje parametre kako bi minimizirao grešku na skupu za treniranje. Zato smo upravo i podešavale broj epoha (prolazile smo kroz isti dataset za deset, trideset i pedeset epoha), kako bi videle njegov uticaj na performanse modela.

Deset epoha

Grafik performanse neuralne mreže kroz 10 epoha obučavanja nad trening i validacionom skupom:



Matrica konfuzije na validacionom skupu:



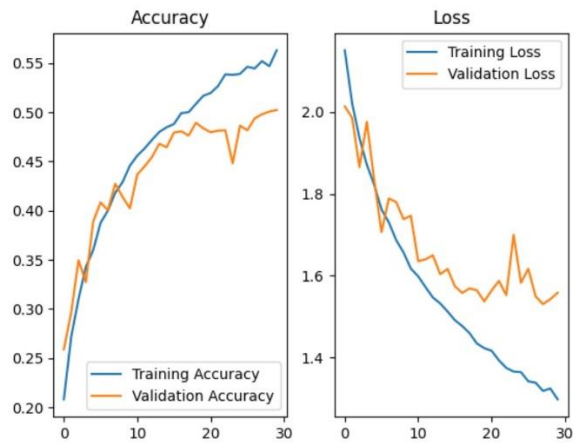
Primeri dobro i loše kvalifikovanih primera dataset-a:



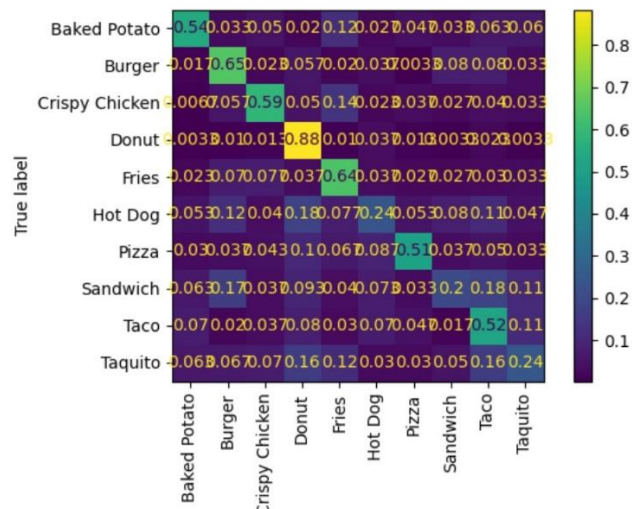
Tačnost modela na validacionom skupu je: **38.333333333333336%**

Trideset epoha

Grafik performanse neuralne mreže kroz 30 epoha obučavanja nad trening i validacionom skupu:



Matrica konfuzije na validacionom skupu:



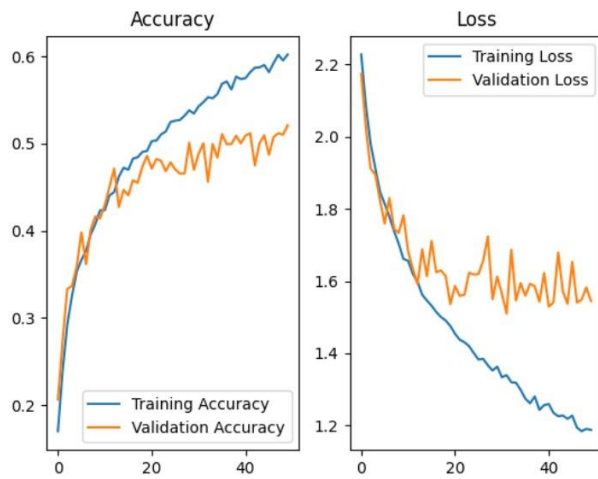
Primeri dobro i loše kvalifikovanih primera dataset-a:



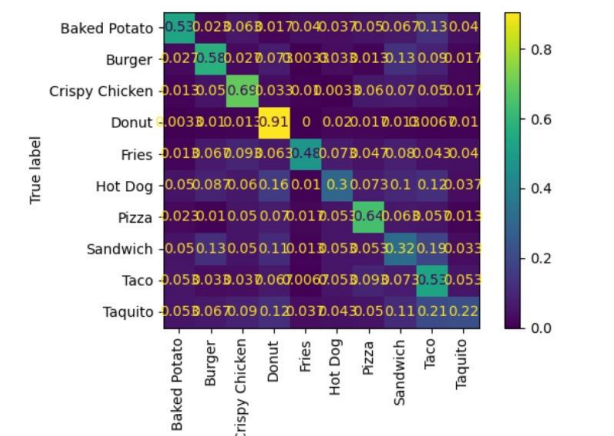
Tačnost modela na validacionom skupu je: **50.23333333333333%**

Pedeset epoha

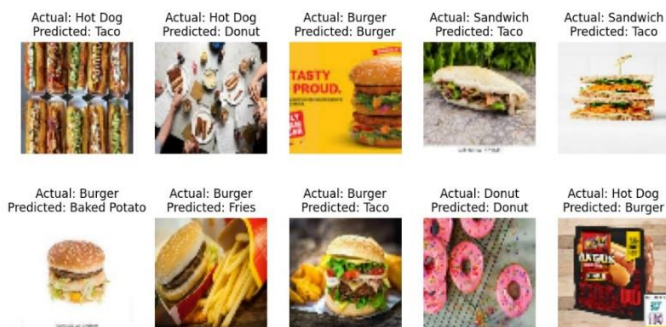
Grafik performanse neuralne mreže kroz 50 epoha obučavanja nad trening i validacionom skupu:



Matrica konfuzije na validacionom skupu:



Primeri dobro i loše kvalifikovanih primera dataset-a:



Tačnost modela na validacionom skupu je: **52.1%**