



Katedra za Signale i sisteme  
Elektrotehnički fakultet  
Univerzitet u Beogradu



13E054NM - Neuralne mreže

## Prvi projektni zadatak

*Deep Learning*

Studenti:  
Aleksa Janjić 2019/0021  
Aleksandar Đordžević 2019/0086

Mentori:  
prof. dr Goran Kvaščev  
dipl. ing Marija Novičić

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Konvolucione neuralne mreže . . . . .	1
1.2	Zadatak . . . . .	1
<b>2</b>	<b>Dataset</b>	<b>2</b>
<b>3</b>	<b>Projektovanje modela</b>	<b>4</b>
3.1	Struktura mreže . . . . .	4
3.2	Podela podataka i preobučavanje . . . . .	4
3.3	Treniranje mreže . . . . .	5
3.4	Zaštita od preobučavanja . . . . .	5
<b>4</b>	<b>Rezultati</b>	<b>6</b>
4.1	Bez ranog zaustavljanja . . . . .	6
4.2	Sa ranim zaustavljanjem . . . . .	7
<b>5</b>	<b>Zaključak</b>	<b>9</b>

## 1 Uvod

### 1.1 Konvolucione neuralne mreže

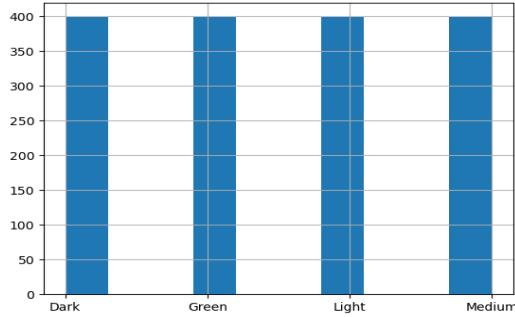
Konvolucione neuralne mreže su primer dubokih neuralnih mreža. Za duboke neuralne mreže je karakteristično da se izbor obeležja vrši automatski. U konvolucionim neuralnim mrežama za ovo su zaslužni algoritmi bazirani na konvoluciji - *filtriranje* i *pooling*. Filtriranje predstavlja ekstrakciju obeležja odnosno ono postojeća obeležja kombinuje u nova obeležja najčešće smanjujući njihov ukupan broj. Pooling predstavlja selekciju obeležja odnosno od svih obeležja biraju se ona najbolja po nekom kriterijumu. Ovakav tip mreža je veoma pogodan za klasifikaciju slika što je upravo potrebno u ovom projektnom zadatku.

### 1.2 Zadatak

Cilj ovog projektnog zadatka je da se primenom konvolucionih neuralnih mreža izvrši klasifikacija zrna kafe. Na raspolaganju je *dataset* sačinjen od 1600 slika zrna kafe četiri različite vrste *Dark, Medium, Light, Green*.

## 2 Dataset

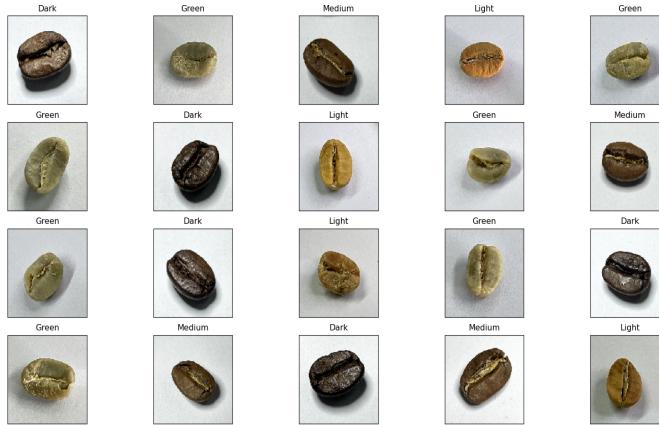
Pre nego što se upustimo u projektovanje modela zgodno je sagledati *dataset* koji nam je na raspolaganju. Na sledećem histogramu možemo videti kako su primerci raspoređeni po klasama (vrstama kafe).



Slika 1: Raspored podataka po klasama

Kao što možemo videti na slici klase su idealno balansirane (svaka klasa sadrži po 400 primeraka) te nije potrebno primenjavati metode za balansiranje klasa.

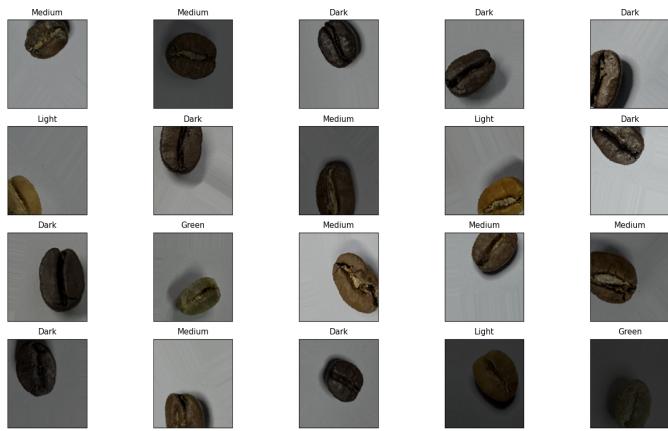
Kako primerci izgledaju možemo videti na sledećoj slici:



Slika 2: Primeri podataka

Primetimo da su slike veoma kvalitetne odnosno zrna su slikana uvek odgore sa dobrom osvetljenjem. Na prvi pogled ovo deluje veoma povoljno, međutim model obučen na ovakvom skupu će biti u stanju da prepozna samo zrna koja su slikana u veoma dobrim uslovima. Kako bi naš model učinili generalnijim odnosno sposobili ga da prepoznae zrna slikana i u lošijim uslovima uradili smo **augmentaciju** podataka. Slike su zarotirane, pomerene i promenjena je osvetljenost.

Nakon augmentacije dataset izgleda ovako:



Slika 3: Primeri podataka nakon augmentacije

## 3 Projektovanje modela

### 3.1 Struktura mreže

Kao što smo već naveli koristićemo konvolucionu neuralnu mrežu. U delu za izbor mreže korišćeni su filtri sa identičnom (*same*) konvolucijom i *max pooling*. Nakon njih postoji *fully connected* sloj sa *relu* aktivacionim funkcijama i 4 izlaz (onoliko koliko ima klase). Takođe na kraju se nalazi *softmax* aktivaciona funkcija koja kao izlaz daje verovatnoće pripadanja određenim klasama. Dodatno su uvedeni *dropout* slojevi kao zaštita od preobučavanja, ali o tome će biti više reči kasnije.

Struktura modela izgleda ovako:

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 128, 128, 32)	896
conv2d_9 (Conv2D)	(None, 126, 126, 32)	9248
max_pooling2d_4 (MaxPooling 2D)	(None, 63, 63, 32)	0
dropout_5 (Dropout)	(None, 63, 63, 32)	0
conv2d_10 (Conv2D)	(None, 63, 63, 64)	18496
conv2d_11 (Conv2D)	(None, 61, 61, 64)	36928
max_pooling2d_5 (MaxPooling 2D)	(None, 30, 30, 64)	0
dropout_6 (Dropout)	(None, 30, 30, 64)	0
conv2d_12 (Conv2D)	(None, 30, 30, 128)	73856
conv2d_13 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_6 (MaxPooling 2D)	(None, 14, 14, 128)	0
dropout_7 (Dropout)	(None, 14, 14, 128)	0
conv2d_14 (Conv2D)	(None, 14, 14, 512)	1638912
conv2d_15 (Conv2D)	(None, 10, 10, 512)	6554112
max_pooling2d_7 (MaxPooling 2D)	(None, 2, 2, 512)	0
dropout_8 (Dropout)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_2 (Dense)	(None, 1024)	2098176
dropout_9 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 4)	4100

Total params: 10,582,308  
Trainable params: 10,582,308  
Non-trainable params: 0

Slika 4: Struktura mreže

### 3.2 Podela podataka i preobučavanje

Pre samog treniranja neuralne mreže neophodno je izvršiti podelu podataka na trening i test skup. Ako ovaj deo preskočimo ne možemo dobiti realan uvid u to koliko je naša mreža valjana jer postoji opasnost od preobučavanja odnosno mreža se previše navikne na podatke na kojima se obučava i na njima radi odlično, ali ne uspeva da generalizuje odnosno radi značajno lošije za podatke na kojima nije obučavana. Dakle, ideja je da se mreža obučava na jednom skupu (trening), a testira na drugom skupu (test) na podacima koje nije "videla" u toku obučavanja. U našem slučaju podela je izvršena u odnosu 75:25 u korist trening skupa, ali kako smo koristili rano zaustavljanje (detaljnije kasnije) trening skup je dodatno podeljen

na validacioni i to u odnosu 80:20 u korist trening skupa. Na kraju je balans sledeći: 960 primeraka u trening skupu, 240 u validacionom i 400 u test skupu.

### 3.3 Treniranje mreže

Sada kada smo napravili strukturu neuralne mreže i podelili podatke na određene skupove preostaje nam samo da definišemo način treniranja mreže. Koristili smo optimizator *Adam*, a za *loss* funkciju izabrali smo kategoričku krosentropiju (*categorical crossentropy*). Za njih smo se odlučili jer su oni najpopularniji izbor za probleme klasifikacije. Mrežu smo trenirali kroz 100 epoha prvo bez ranog zaustavljanja, a nakon toga i sa njim.

### 3.4 Zaštita od preobučavanja

Kao što smo već napomenuli preobučavanje je veliki problem prilikom treniranja neuralnih mreža i kako bi ga sprečili uveli smo *dropout* i rano zaustavljanje.

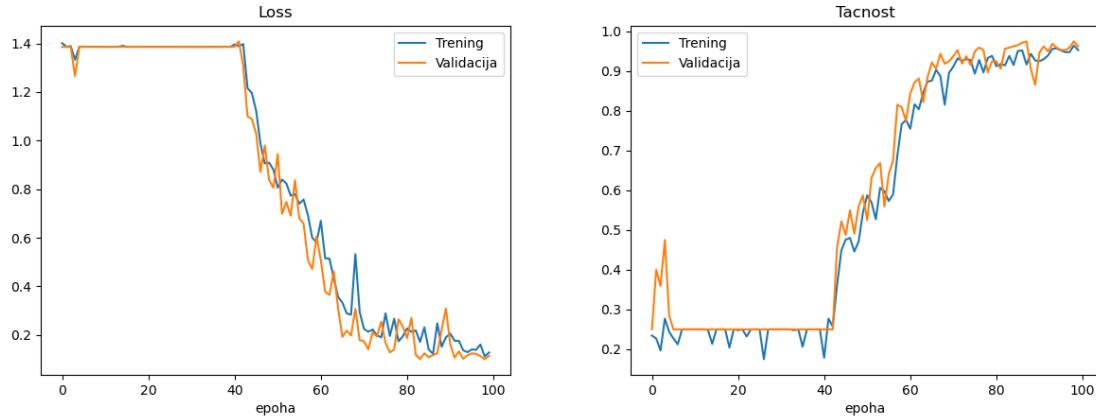
*Dropout* radi tako što uklanja neke neurone tako što njihove težine postavlja na nulu. To se dešava u svakom koraku sa određenom verovatnoćom (20% u našem slučaju).

*Rano zaustavljanje* funkcioniše tako što tokom treniranja računa tačnost modela na validacionom skupu i treniranje zaustavlja ako tačnost na validacionom skupu nije dostigla novi maksimum u određenom broju epoha (u našem slučaju 10).

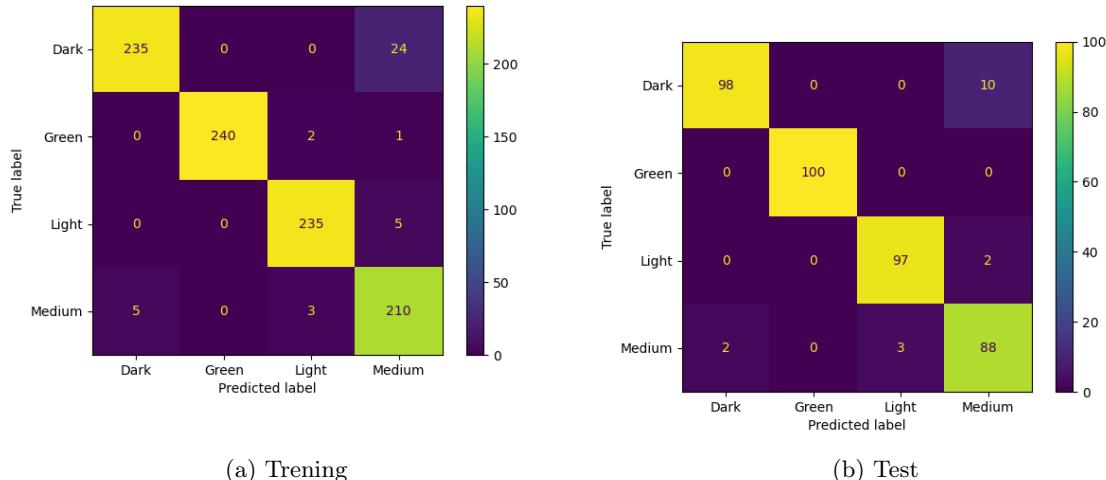
## 4 Rezultati

### 4.1 Bez ranog zaustavljanja

Grafički tačnosti i *loss* funkcije kroz epohe izgledaju ovako:

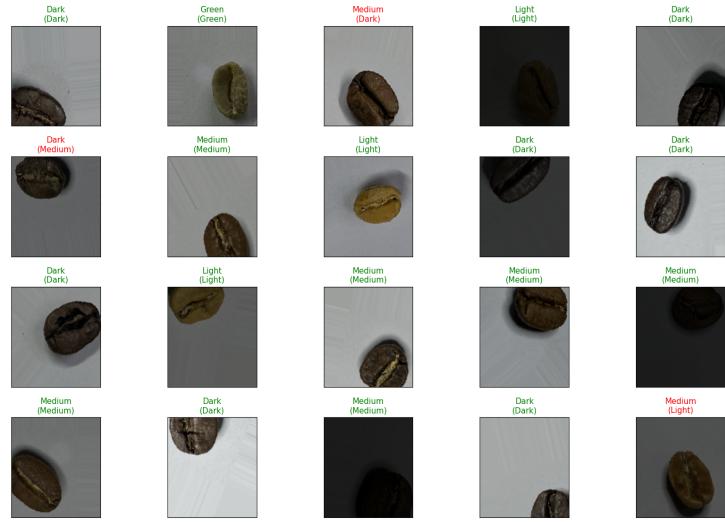


Kao što vidimo nakon određenog broja epoha i tačnost i *loss* počinju da stagniraju i pored *dropout-a*. Zbog ovoga je kasnije uvedeno i rano zaustavljanje.  
Konfuzione matrice na trening i test skupu izgledaju ovako:



Tačnost na test skupu iznosi 95.75%, a na trening skupu 95,83%.  
Primetimo da najviše grešaka model pravi kod klasifikacije između klasa *Dark* i *Medium* što je i očekivano jer su one i na oko najsličnije.

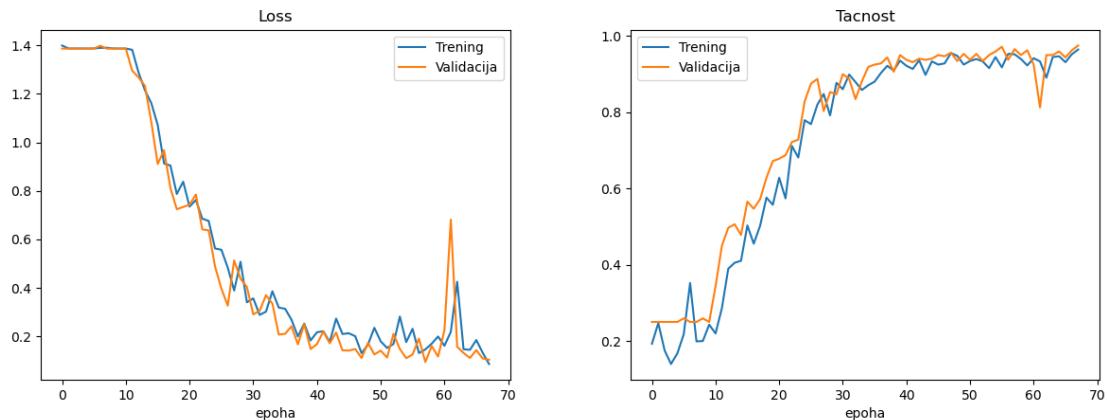
Evo i primera dobrih i loših klasifikacija:



Slika 7: Primeri klasifikacije

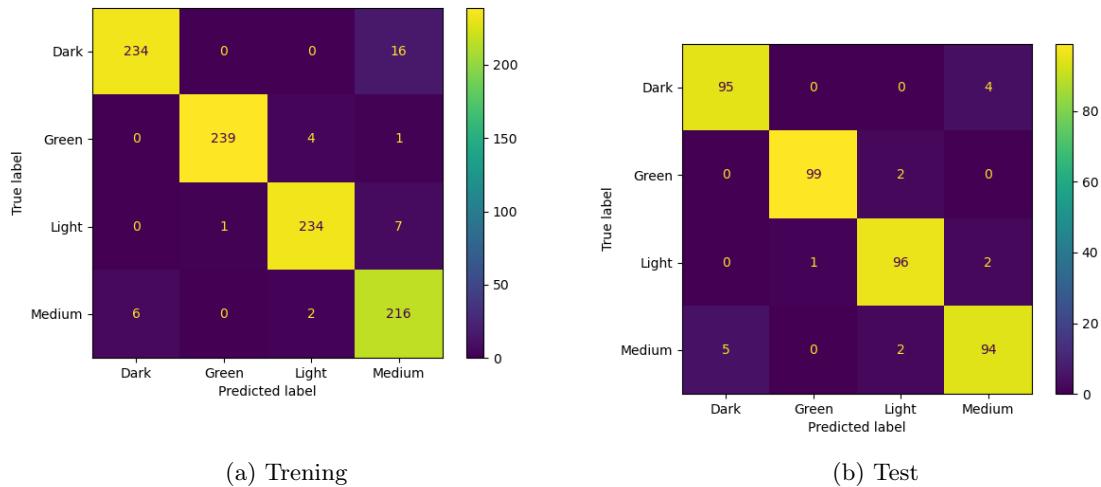
## 4.2 Sa ranim zaustavljanjem

Grafici tačnosti i *loss* funkcije kroz epohe izgledaju ovako:



Primetimo da je sada treniranje zaista zaustavljeno ranije i na ovaj način smo ubrzali treniranje, a kasnije ćemo videti čak i dobili veću tačnost.

Konfuzione matrice na trening i test skupu izgledaju ovako:

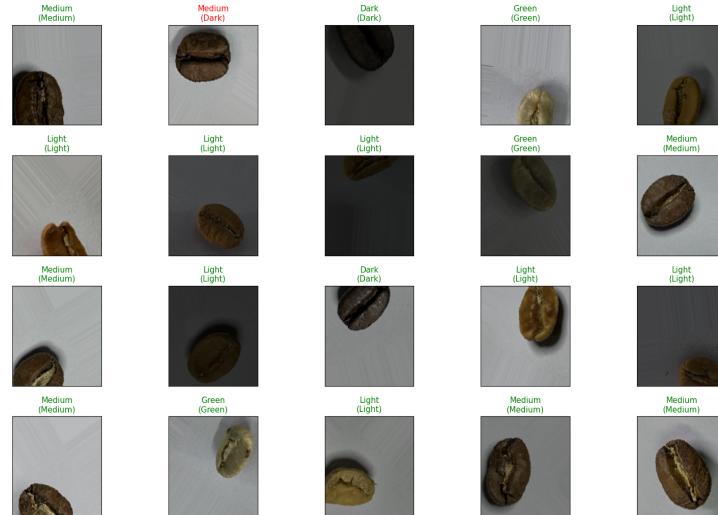


(a) Trening

(b) Test

Tacnost na test skupu iznosi 96.00%, a na trening skupu 96,15%. Sada je dobijena veća tačnost te je uvođenje ranog zaustavljanja opravdano

Evo i primera dobrih i loših klasifikacija:



Slika 10: Primeri klasifikacije

## 5 Zaključak

Na osnovu rezultata možemo zaključiti da je klasifikacija zrna kafe u 4 klase na osnovu slike uspešno odrađena. Projektovana je konvolucionna neuralna mreža sa filtriranjem identičnom konvolucijom i *max pooling*-om. U *fully connected* sloju su korišćene *relu* aktivacione funkcije, a na izlazu *softmax*. Kao optimizator korišćen je *Adam*, za *loss* uzeta je kategorička krosentropija. Kao zaštita od preobučavanja uveden je *dropout*, a testiran je i model sa još dodatim ranim zaustavljanjem koji se i pokazao kao bolji.