

# Image classification using a transfer learning model

Projekat iz Računarske inteligencije  
Matematički fakultet  
Univerzitet u Beogradu

Dorđe Petrović  
[mi19113@alas.matf.bg.ac.rs](mailto:mi19113@alas.matf.bg.ac.rs)  
Nikola Subotić  
[mi18169@alas.matf.bg.ac.rs](mailto:mi18169@alas.matf.bg.ac.rs)

Jun 2023

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Opis problema</b>	<b>3</b>
<b>3</b>	<b>Naše rešenje</b>	<b>4</b>
3.1	Dataset . . . . .	4
3.2	Pretprocesiranje . . . . .	5
3.3	Metrike . . . . .	5
3.4	Konture i predviđanje . . . . .	6
3.5	Transfer Model . . . . .	7
3.6	Čist model . . . . .	8
3.7	Klasičan model . . . . .	9
<b>4</b>	<b>Zaključak</b>	<b>10</b>
<b>5</b>	<b>Literatura</b>	<b>11</b>

## 1 Uvod

Treniranje dubokih neuronskih mreža je zahtevan proces koji koristi mnogo resursa, a često je potrebna i velika količina podataka za obuku neuronske mreže koji nisu uvek lako dostupni. Stoga, ove probleme možemo izbeći koristeći težine modela iz prethodno obučениh modela koji su razvijeni zarad testiranja standardnih skupova podataka računarske grafike. U transfer learningu, šušinski pokušavamo da iskoristimo ono što je naučeno u jednom zadatku kako bismo poboljšali generalizaciju u drugom. Prebacujemo težine koje je mreža naučila u zadatku 1 sa velikim brojem dostupnih podataka na novi zadatak 2 koji nema mnogo podataka. Znanje već obučenog modela se prenosi na drugi, ali blisko povezan problem kroz transfer learning. To znanje može biti različitih oblika u zavisnosti od problema i podataka. Najbolje modele možemo preuzeti i koristiti direktno ili ih integrisati u novi model za naše sopstvene probleme. Na taj način ubrzavamo fazu obuke i poboljšavamo performanse našeg modela mašinskog učenja čak i sa malim skupom podataka.

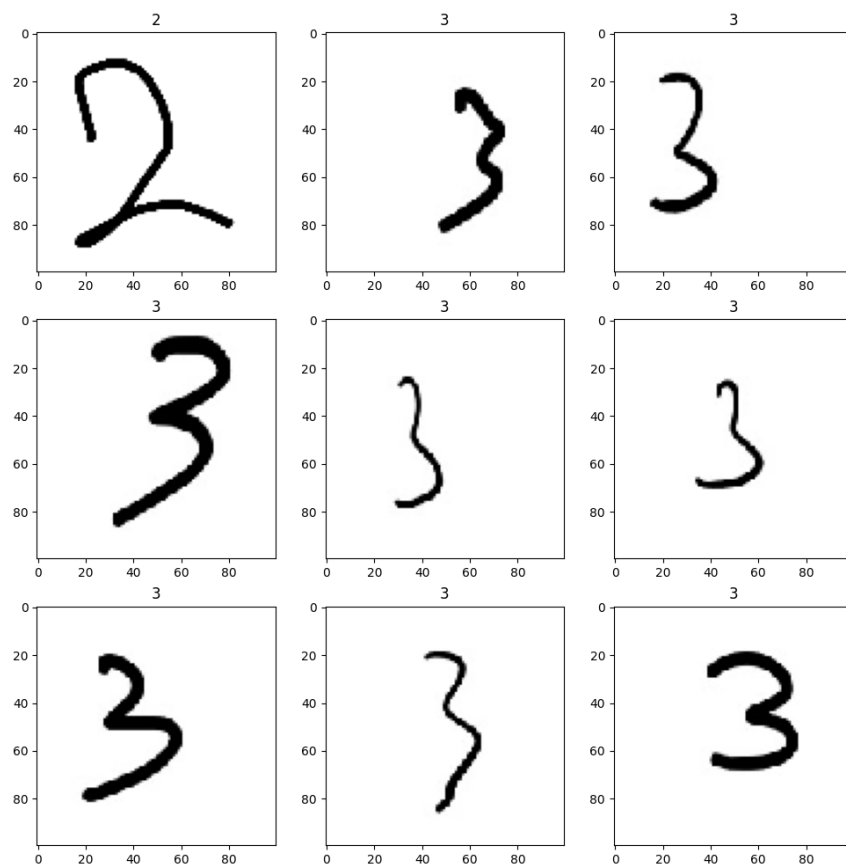
## 2 Opis problema

Odlučili smo da transfer learning iskoristimo za stvari za koje nije originalni model bio namenjen, a to je prepoznavanje rukom pisanih cifara i osnovnih matematičkih operatora i evaluiranje zadatog izraza. Ovo je zahtevan problem koji može koristiti učenicima za proveru rešenja nekog zadatka.

## 3 Naše rešenje

### 3.1 Dataset

Koristili smo "Handwritten Math Symbols" ([Kaggle.com](https://www.kaggle.com/datasets/ahmedhussein/handwritten-math-symbols)) uz dodatne modifikacije. Ovaj skup podataka je bio najbolji javni koji smo pronašli. Zbog ograničenja u dataset-u i resursima mi smo obradili najosnovnije operatore koji podrazumevaju sabiranje, oduzimanje, množenje, deljenje, jednakost i decimalna tačka. Nakon što se programu prosledi slika izraza on će prepoznati individualne cifre i operatore i evaluirati ih davajući konačan rezultat u vidu broja ili boolovske vrednosti (ako je u pitanju jednakost). Model najbolje prepoznaje slike napisane u digitalnom formatu sa belom pozadinom i crnim tekstom, ali ovo se može poboljšati korišćenjem većeg skupa podataka i naprednijih algoritama za prepoznavanje karaktera na proizvoljnoj slici. Originalni dataset je podeljen na 3 grupe, za treniranje (80 %), valuaciju (10 %) i testiranje (10 %). Takođe je očišćen od suvišnih klasa i urađena je bolja organizacija podataka. Postoji 16 klasa, cifre od 0 do 9 kao i operatori (+, -, \*, /, ., ==).



Slika 1: Primer elemenata dataseta

## 3.2 Pretprocesiranje

Korak pretprocesiranja je bitan zbog transformacije slika u format pogodan za model. Koristili smo proces augmentacije podataka sa Keras ImageDataGenerator klasom kako bismo izmenili naš skupa podataka i tako poboljšali treniranje modela. Slike su učitane kao RGB(3 kanala), zatim su prebačene u crno-bele i izvršeno je invertovanje boje, tj sa slike koja ima belu pozadinu i crn tekst prešli smo na crnu pozadinu sa belim tekstom. Primenjeno je i skaliranje slika na veličinu 100x100 piksela, takođe ostajemo na 3, sada crno bela kanala, zbog kompatibilnosti sa prethodno obučanim modelima.

## 3.3 Metrike

Postoji nekoliko mera koje se koriste za prikazivanje performansi rezultata klasifikacije, mi smo uzeli u obzir sledeće četiri:

$$Tačnost = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Preciznost = \frac{TP}{TP + FP}$$

$$Odziv = \frac{TP}{TP + FN}$$

$$f1\text{-skor} = \frac{2 * (Preciznost * Odziv)}{(Preciznost + Odziv)}$$

*TP: Tačno Pozitivne, TN: Tačno Negativne*

*FP: Lažno Pozitivne i FN: Lažno Negativne.*

**Tačnost** - (Accuracy) je osnovna metrika koja se koristi za evaluaciju modela, opisuje broj tačnih predviđanja u odnosu na ukupan broj predviđanja. Poželjno postići tačnost veću od 90%.

**Preciznost** - (Precision) je mera koliko su tačna pozitivna predviđanja koja su napravljena.

**Odziv** - (Recall) je mera koliko je klasifikator tačno predvideo pozitivne slučajeve u odnosu na ukupan broj pozitivnih slučajeva u podacima.

**f1-skor** - (f1-score) je mera koja ravnotežno vrednuje dva odnosa: Preciznost i Odziv. f1-skor može imati vrednost od 0 do 1, pri čemu je 0 najgora moguća vrednost, a 1 najbolja. Vrednost 1 predstavlja da je model savršeno klasifikovao svako posmatranje u ispravnu klasu.

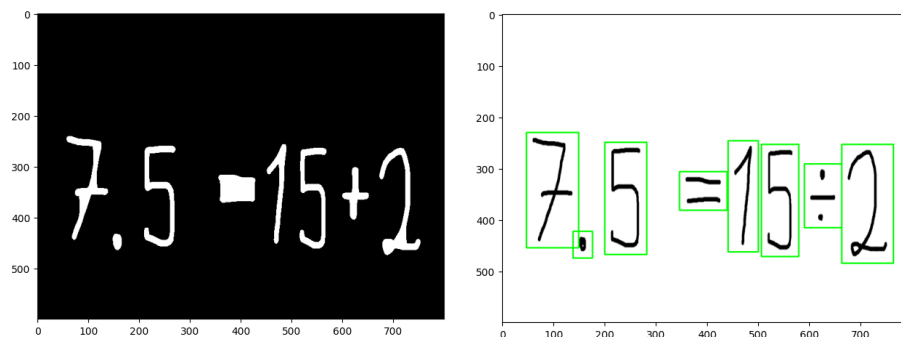
### 3.4 Konture i predviđanje

Model je treniran nad pojedinačnim ciframa i operatorima, a na ulazu prima sliku na kojoj se nalazi ceo izraz. Problem koji se javlja je kako prepoznati svaki od elemenata koji se nalazi na slici i klasifikovati ga. Korišćenjem cv2 biblioteke i korišćenjem funkcije `findContours` pronalazimo konture na prethodno pretprocesiranoj slici. Funkcija pronalazi konture spojenih piksela, na primeru dec-a (podeljeno) bice prepoznata 3 elementa, a to su tačka, linija, tačka. Javlja se novi problem, sada imamo 3 nezavisna elementa koja treba da čine celinu. Rešenje se zasniva na tome da korišćenjem funkcije `morphologyEx` primenimo `MORPH_CLOSE` operaciju koja spaja bliske elemente u jednu celinu. Sada, ponovo pozivamo funkciju `findContours` ali na rezultatu prethodnog koraka. Time dobijamo ispravne konačne konture elemenata.

Prethodnim rezultatom dobili smo pozicije svih kontura na slici. Sortiramo ih i za svaku od njih, sečemo sliku na odgovarajućim pozicijama i šaljemo ih modelu na predviđanje. Rezultati svih predviđanja čuvamo u nizu i vrednost izraza evaluiramo na osnovu njih.

$$7.5 = 15 \div 2$$

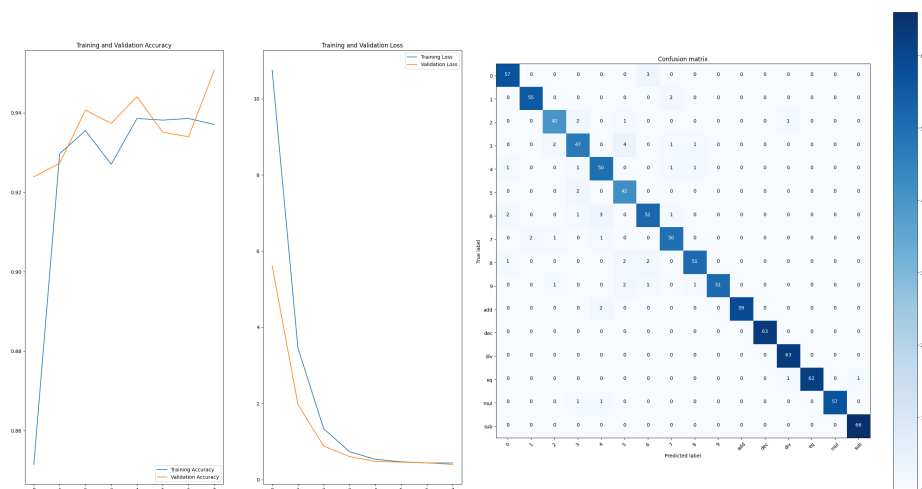
Slika 2: Originalna slika



Slika 3: Rezultat primene morfologije i prepoznavanje konačnih kontura

### 3.5 Transfer Model

Za potrebe transfer learning-a smo koristili MobileNetV2 koji dolazi sa bibliotekom Keras i koristi imagenet težine. MobileNetV2 je model konvolucionih neuronskih mreža (CNN) koji je razvijen kao efikasno rešenje za mobilne i resursno ograničene uređaje. Koristi mnoga poboljšanja kao što su Bottleneck strukture i Inverzna rezidualna veza i ima 53 sloja sa oko 3.4 miliona parametara. Izbacili smo gornji sloj i isključili treniranje baznom modelu, a potom dodali BatchNormalization, Dense i Dropout slojeve, kao i finalni Dense koji za svaku klasu vrši predviđanje. Model takođe koristi i sistem za rano zaustavljanje koji prati val\_loss i potencijalno može zaustaviti treniranje ranije ako misli da se dodatnim treniranjem neće postići bolji rezultat.

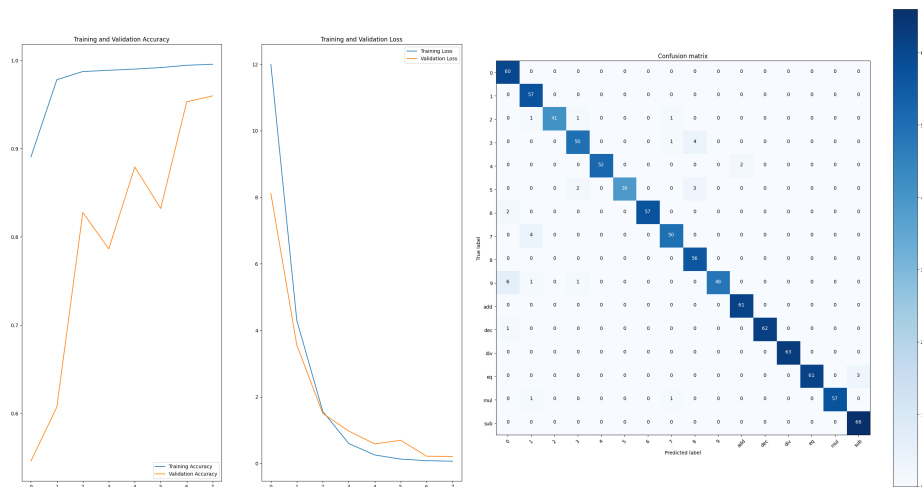


Slika 4: Rezultati treniranja transfer modela

	precision	recall	f1-score
0	0.93	0.95	0.94
1	0.96	0.96	0.96
2	0.91	0.91	0.91
3	0.87	0.85	0.86
4	0.88	0.93	0.9
5	0.82	0.95	0.88
6	0.9	0.88	0.89
7	0.91	0.93	0.92
8	0.94	0.91	0.93
9	1	0.91	0.95
add	1	0.97	0.98
dec	1	1	1
div	0.97	1	0.98
eq	1	0.97	0.98
mul	1	0.97	0.98
sub	0.99	1	0.99
accuracy	0.95		

### 3.6 Čist model

Radi poređenja smo trenirali istu arhitekturu kao u predhodnom primeru ali ovog puta model je od nule trenirao sve parametre i oni nisu bili zaključani. Dodati su isti slojevi nakon uklanjanja gornjeg sloja. Ovaj model je bio treniran na Google Colab platformi zbog velikog broja parametara i slojeva.



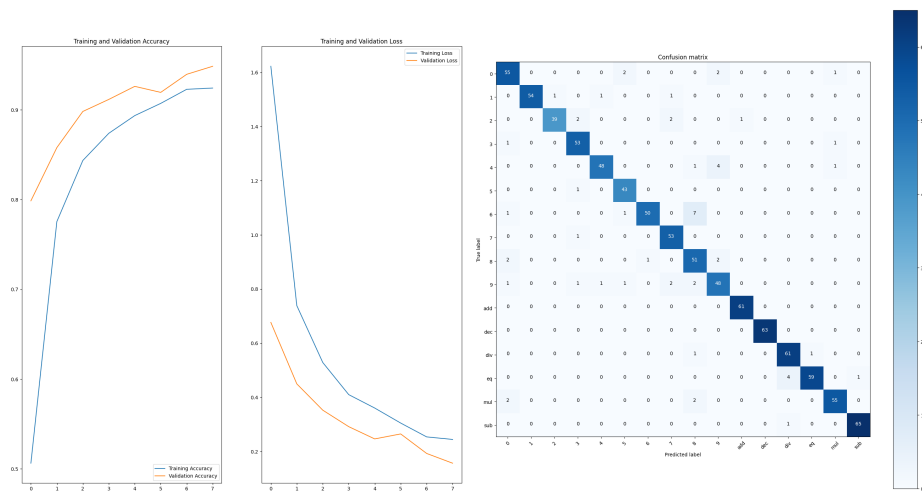
Slika 5: Rezultati treniranja čistog transfer modela

	precision	recall	f1-score
0	0.87	1	0.93
1	0.89	1	0.94
2	1	0.93	0.96
3	0.93	0.91	0.92
4	1	0.96	0.98
5	1	0.89	0.94
6	1	0.97	0.98
7	0.94	0.93	0.93
8	0.89	1	0.94
9	1	0.86	0.92
add	0.97	1	0.98
dec	1	0.98	0.99
div	1	1	1
eq	1	0.95	0.98
mul	1	0.97	0.98
sub	0.96	1	0.98
accuracy	0.96		



### 3.7 Klasičan model

Na kraju radi poređenja dodat je model koji ne koristi transfer learning već ima strukturu koja je klasičan primer modela za prepoznavanje slika. Sastoji se iz Conv2D, MaxPooling, Flatten i Dense slojeva i ima ukupno 7 slojeva i oko 4.35 miliona parametara.



Slika 6: Rezultati treniranja našeg modela

	precision	recall	f1-score
0	0.89	0.92	0.9
1	1	0.95	0.97
2	0.97	0.89	0.93
3	0.91	0.96	0.94
4	0.96	0.89	0.92
5	0.91	0.98	0.95
6	0.98	0.85	0.91
7	0.91	0.98	0.95
8	0.8	0.91	0.85
9	0.86	0.86	0.86
add	0.98	1	0.99
dec	1	1	1
div	0.92	0.97	0.95
eq	0.98	0.92	0.95
mul	0.95	0.93	0.94
sub	0.98	0.98	0.98
accuracy	0.94		

## 4 Zaključak

Iako je za najbolje rešavanje ovog konkretnog problema bolje bilo koristiti neke drugačije tehnike od transfer learninga želeli smo da pokažemo da je to moguće i da se uz manje resursa mogu postići slični ili bolji rezultati, što bi bilo još evidentnije na nekom boljem skupu podataka.

## 5 Literatura

- [1] Materijali iz Računarske inteligencije
- [2] Image Classification with Transfer Learning Using a Custom Dataset: Comparative Study