

# Poređenje tehnika mašinskog učenja korišćenih za klasifikaciju slika na osnovu umetnika

Katarina Dimitrijević 1075/2018  
Septembar, 2020.

# Uvod

Predmet ovog rada je prepoznavanje umetnika na osnovu slika - zadatak koji je za ljude netrivialan i zahteva veliki stepen poznavanja materije. Kada su tehnike mašinskog učenja u pitanju, izazovi modela su uočavanje svojstava koje ljudi ne mogu da primete.

Konkretna tehnika klasifikacije obrađene u ovom radu su SVM i par različitih modela CNN-a sa osvrtom na međusobno upoređivanje ovih tehnika.

# Skup podataka

Skup podataka korišćen za potrebe ovog rada je preuzet sa sajta Kaggle (Painters by numbers) gde je korišćen podskup koji čini 7462 slike u boji, dimenzija 256x256 grupe od 15 umetnika. Slike svakog od njih su približno slično zastupljene u ovom skupu podataka, tako da su klase izbalansirane. Skup podataka je u razmeri 80%, 10%, 10% podeljen na trening, test i skup za validaciju, redom.



# SVM pristup problemu

## izdvajanje atributa slike

Kao najkorisniji deskriptori slike pokazali su se:

- **GIST deskriptori** - globalni deskriptori tekstone, definisani tako da opiše prostorna struktura prirodnih scena.
- **Hu momenti** - skup 7 polinomijalnih kombinacija momenata slike invarijantnih na skaliranje, translaciju i rotaciju
- **Histogrami boja** - predstavljaju raspodelu crvene, zelene i plave boje na slici

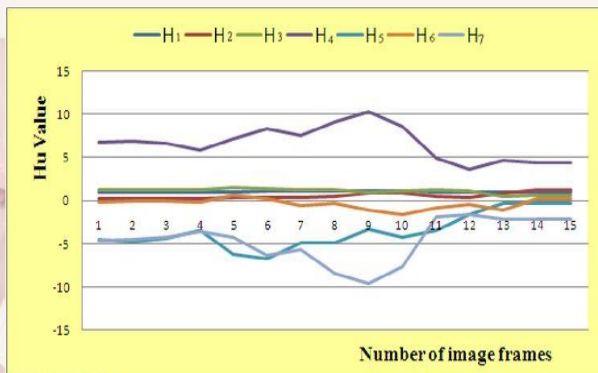


## Example: Hu Moments

crying



rizmanid

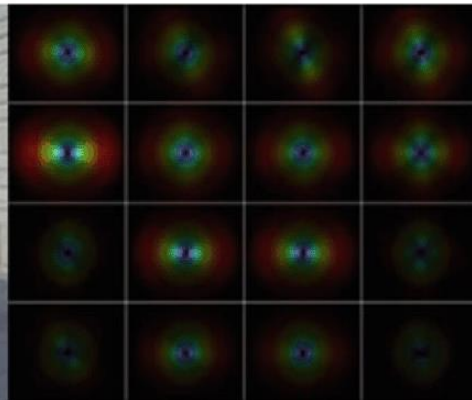


9

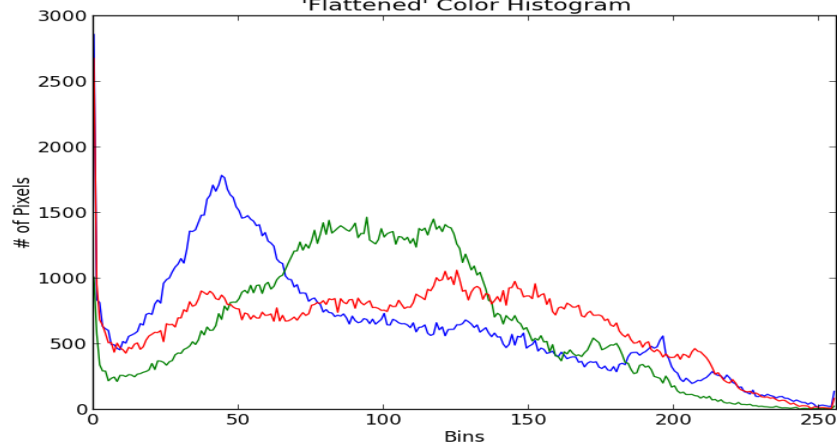
Input Image



GIST Descriptor



'Flattened' Color Histogram



# SVM pristup problemu

Jedna od tehnika koje su korišćene je analiza glavnih komponenti (PCA) sa ciljem dobijanja linearno nekorelisanih atributa i smanjenja dimenzionalnosti prostora

Konkretni model koji je korišćen je višeklasni SVM klasifikator sa linearnim kernelom, pri čemu je najveća tačnost dobijena za parametre  $C = 10$  i  $\gamma = 0.1$ , dobijene pomoću grid search-a.

U ovom konkretnom slučaju se linearni kernel pokazao boljim od RBF kernela, sa približnom tačnošću 50% u odnosu na tačnost od približno 48% koja je postignuta korišćenjem RBF kernela.

# SVM pristup problemu

Matrica konfuzije:

[	33	1	5	0	1	0	1	0	0	3	0	4	3	2	2]
[	2	26	2	4	1	2	2	3	1	0	1	3	3	4	1]
[	0	2	27	4	0	0	2	5	0	0	0	1	5	3	5]
[	1	2	4	18	2	6	4	8	0	0	2	0	2	4	2]
[	0	3	1	3	20	11	5	0	0	0	2	0	3	2	4]
[	0	0	1	1	9	28	2	6	0	0	2	1	5	0	0]
[	1	2	4	6	3	1	17	0	0	0	1	1	2	3	13]
[	2	2	3	5	3	0	4	29	0	0	1	0	5	0	1]
[	2	0	0	0	0	0	0	0	47	6	0	0	0	0	0]
[	2	1	1	0	0	0	0	1	3	44	0	0	1	2	0]
[	3	4	0	1	5	6	4	3	0	0	22	1	2	1	2]
[	1	9	1	3	2	0	2	2	0	2	0	26	3	2	2]
[	2	5	1	2	2	3	3	6	0	1	0	2	22	2	4]
[	6	1	4	3	2	3	2	2	1	0	0	1	4	25	1]
[	3	2	8	6	6	0	5	1	3	1	2	0	1	1	16]]

Classification report:

	precision	recall	f1-score	support
0	0.57	0.60	0.58	55
1	0.43	0.47	0.45	55
2	0.44	0.50	0.47	54
3	0.32	0.33	0.32	55
4	0.36	0.37	0.36	54
5	0.47	0.51	0.49	55
6	0.32	0.31	0.32	54
7	0.44	0.53	0.48	55
8	0.85	0.85	0.85	55
9	0.77	0.80	0.79	55
10	0.67	0.41	0.51	54
11	0.65	0.47	0.55	55
12	0.36	0.40	0.38	55
13	0.49	0.45	0.47	55
14	0.30	0.29	0.30	55
accuracy			0.49	821
macro avg	0.50	0.49	0.49	821
weighted avg	0.50	0.49	0.49	821

# CNN kao pristup problemu

Jedna od ideja u radu bila je i korišćenje transfera učenja.

**Transfer učenja** u slučaju deep learning-a predstavlja tehniku gde se neuronska mreža najpre trenira na problemu koji je srodan problemu koji se rešava, zatim su tako pretrenirani slojevi mreže dostupni za potrebe problema koji se rešava. Na ovaj način se drastično može skratiti vreme potrebno za treniranje mreže.



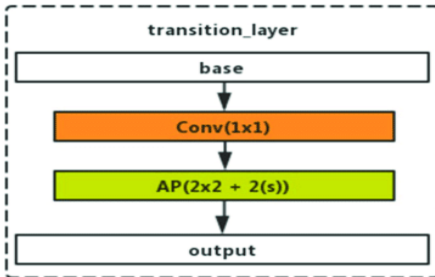
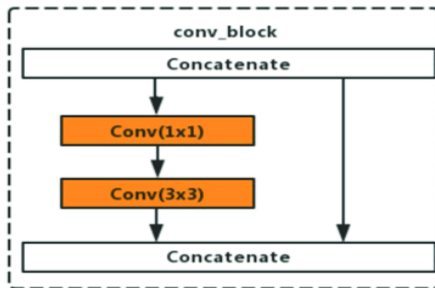
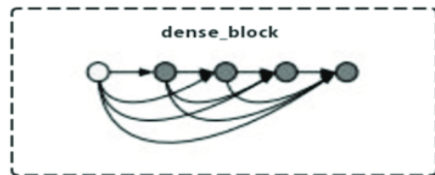
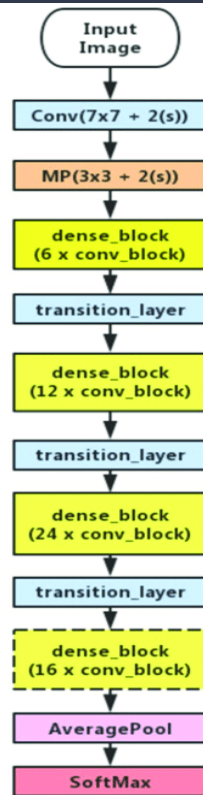
# CNN kao pristup problemu

Usled tehničkih poteškoća i nedostatka resursa za treniranje inicijalne konvolutivne neuronske mreže, kao rešenje problema su korišćeni sledeći Keras modeli:

1. DenseNet121
2. ResNet50v2

# DenseNet121

DenseNet121 je jedan od modela Keras biblioteke koji se bazira na ideji da duboke neuronske mreže mogu biti efikasnije i dati tačnije rezultate ukoliko postoje krace veze između slojeva bliskih ulazu i onih bliskih izlazu. DenseNet modeli su se dobro pokazali u slučaju rešavanja problema nestajućih gradijenata, ponovnog iskorišćavanja atributa, zatim smanjenja broja parametara mreže. Na slici desno je prikazana DenseNet121 arhitektura.

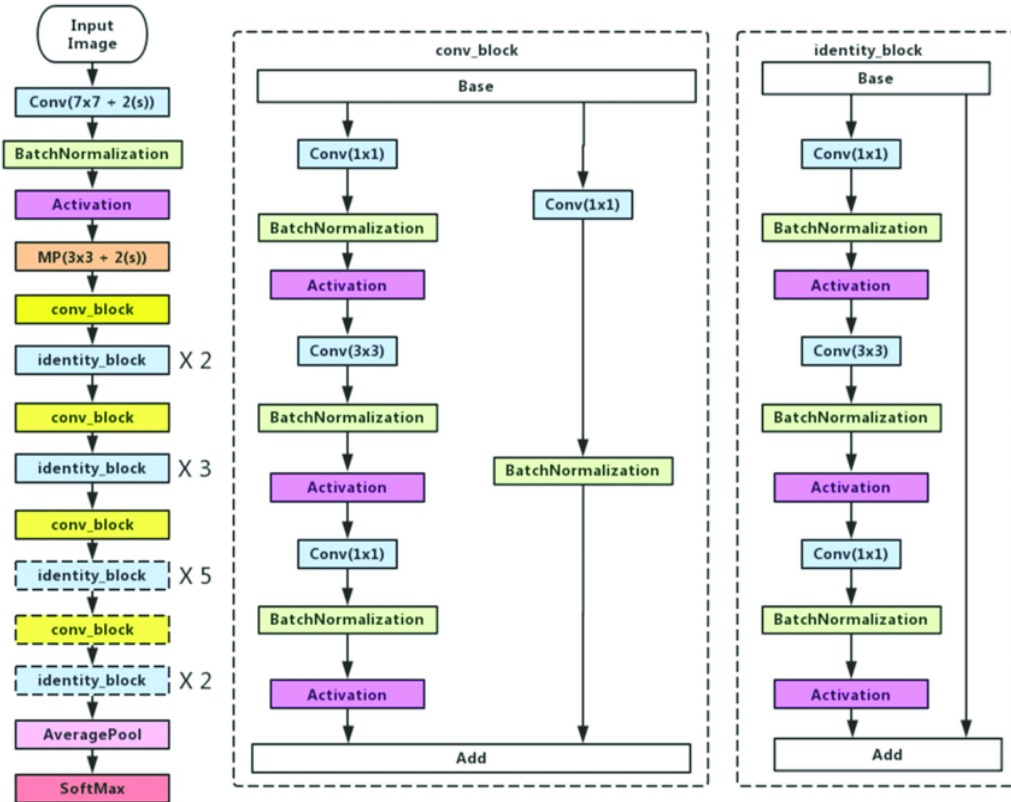


# ResNet50v2

ResNet je arhitektura koja je omogućila drastično povećanje broja slojeva mreže i uz to eliminaciju problema nestajućih gradijenata. Ideja se zasniva na traženju identičkog mapiranja između slojeva i preskakanja tih slojeva.

Osnovna razlike između ResNet v1 i v2 je korišćenje tehnike unutrašnje standardizacije u slučaju v2.

Na slici desno je prikazana ResNet50v2 arhitektura (50 slojeva)



# DenseNet121

Zbog korišćenja pretreniranog DenseNet121 modela (koji je dao najbolje rezultate), za potrebe pretprocesiranja podataka korišćena je Keras funkcija **application.densenet.preprocess\_input**, pri čemu je bilo i eksperimenata sa DenseNet121 mrežom koja nije unapred trenirana i, u tom slučaju, funkcijom **ImageDataGenerator** za pretprocesiranje. Ulaz u mrežu cine slike dimenzija 224x224x3.

Takođe, kako su pretrenirani DenseNet modeli namenjeni za znatno veće skupove podataka, na postojeću arhitekturu dodato je još dva sloja Flatten i Softmax koji pridružuje klase instancama (jednu od 15).

Korišćena je i tehnika “ranog zaustavljanja” sa patience parametrom jednakim 3 na parametru loss. Takođe, uzet je snapshot modela sa najvećom vrednošću ‘val\_accuracy’ parametra kao najbolji.

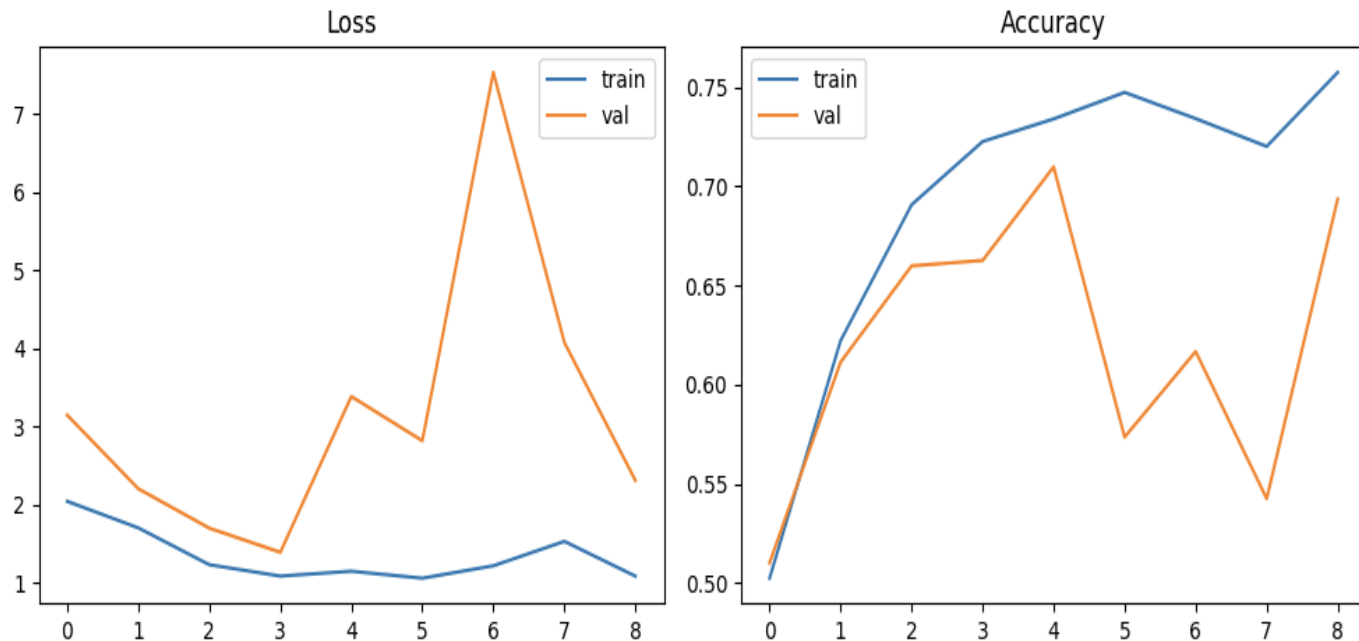
# DenseNet121

**Test loss:**

3.8096697330474854

**Test accuracy:**

0.688622772693634



# DenseNet121

Matrica konfuzije:

[	23	10	0	0	1	0	1	0	7	1	0	0	2	0	0]
[	3	30	0	0	0	1	1	0	2	0	3	2	0	1	1]
[	3	3	22	6	0	2	7	0	0	0	0	1	0	0	0]
[	0	2	0	32	0	4	5	0	0	0	0	2	0	0	0]
[	1	2	0	3	25	9	1	0	1	0	2	0	0	0	0]
[	0	0	0	5	1	34	0	0	2	0	1	1	0	0	0]
[	2	0	0	3	0	2	36	0	0	0	1	0	0	0	0]
[	0	2	0	4	0	3	0	34	1	0	0	0	0	0	0]
[	0	1	0	0	0	8	0	0	28	7	0	0	1	0	0]
[	0	1	0	2	0	0	1	0	2	36	0	1	0	0	2]
[	0	1	0	1	0	0	2	0	0	0	39	0	0	0	1]
[	4	6	0	0	0	3	1	0	2	0	2	24	1	1	1]
[	0	1	0	0	0	1	0	0	0	0	0	1	41	0	1]
[	0	2	1	3	0	1	0	0	3	1	3	4	1	26	0]
[	2	3	0	1	1	0	7	0	0	0	0	0	0	1	30]]

Classification report:

	precision	recall	f1-score	support
0	0.61	0.51	0.55	45
1	0.47	0.68	0.56	44
2	0.96	0.50	0.66	44
3	0.53	0.71	0.61	45
4	0.89	0.57	0.69	44
5	0.50	0.77	0.61	44
6	0.58	0.82	0.68	44
7	1.00	0.77	0.87	44
8	0.58	0.62	0.60	45
9	0.80	0.80	0.80	45
10	0.76	0.89	0.82	44
11	0.67	0.53	0.59	45
12	0.89	0.91	0.90	45
13	0.90	0.58	0.70	45
14	0.83	0.67	0.74	45
accuracy			0.69	668
macro avg	0.73	0.69	0.69	668
weighted avg	0.73	0.69	0.69	668

# ResNet50v2

U slučaju ovog modela, dimenzije ulaza, odnosno svake slike su postavljene na 299x299x3.

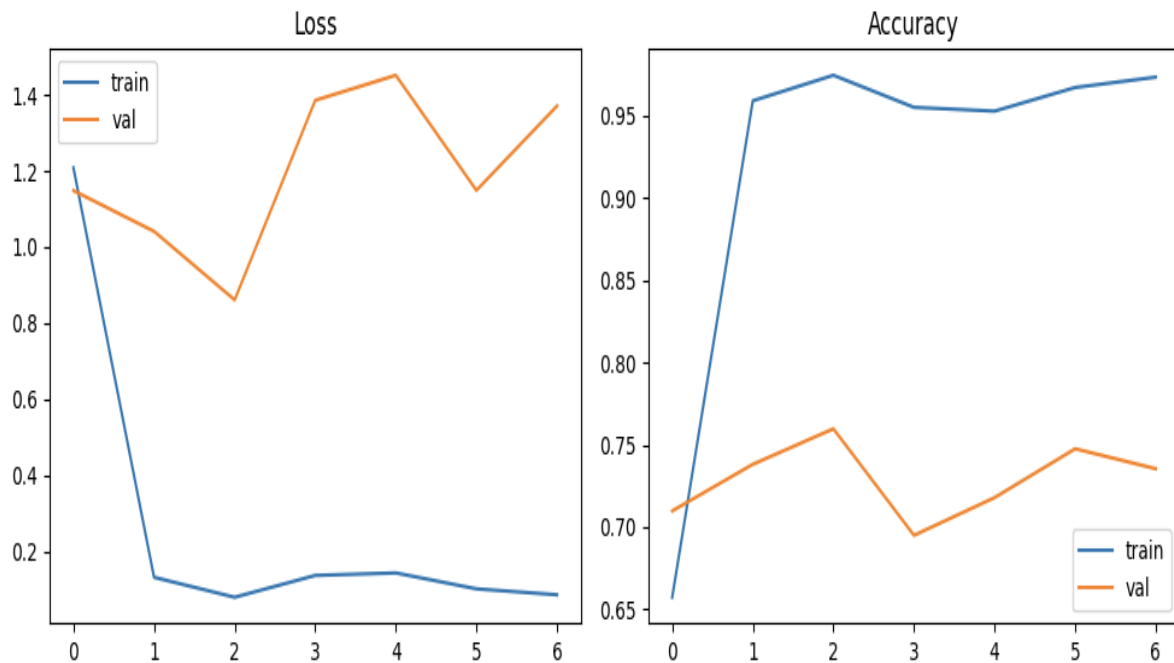
Analogno je korišćena Keras funkcija `application.resnet.preprocess_input` za potrebe preprocesiranja. Generalno, princip treniranja modela i donošenja odluka je identičan kao i u prethodno navedenom slučaju DenseNet121.

Ipak, ResNet50v2 je dao nešto bolje rezultate prilikom testiranja, ali se i više prilagodio podacima:

Test loss: 0.926967203617096

Test accuracy:

0.7679640650749207



# ResNet50v2

Matrica konfuzije:

[	34	2	0	0	0	0	3	0	0	1	2	0	0	0	3]
[	0	32	0	1	0	0	0	0	0	0	3	4	0	3	1]
[	1	0	27	0	0	0	1	2	0	0	1	1	3	1	7]
[	1	0	1	31	2	0	2	0	0	0	0	2	1	1	4]
[	0	1	0	2	30	7	1	0	0	0	1	0	0	1	1]
[	1	4	0	4	1	33	0	0	0	0	1	0	0	0	0]
[	1	1	1	0	1	0	31	0	0	0	4	2	0	0	3]
[	0	0	0	1	0	1	0	42	0	0	0	0	0	0	0]
[	1	0	0	0	0	0	0	0	42	0	2	0	0	0	0]
[	1	0	0	0	0	0	2	0	0	38	0	1	1	1	1]
[	0	3	0	1	1	0	0	0	0	0	37	2	0	0	0]
[	4	3	0	0	1	1	2	0	0	0	2	24	3	2	3]
[	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0]
[	1	0	0	1	0	0	0	0	0	0	0	2	2	38	1]
[	1	0	2	0	1	1	6	0	0	0	2	1	1	1	29]]]

Classification report:

	precision	recall	f1-score	support
0	0.74	0.76	0.75	45
1	0.70	0.73	0.71	44
2	0.87	0.61	0.72	44
3	0.76	0.69	0.72	45
4	0.81	0.68	0.74	44
5	0.77	0.75	0.76	44
6	0.65	0.70	0.67	44
7	0.95	0.95	0.95	44
8	1.00	0.93	0.97	45
9	0.97	0.84	0.90	45
10	0.67	0.84	0.75	44
11	0.62	0.53	0.57	45
12	0.80	1.00	0.89	45
13	0.79	0.84	0.82	45
14	0.55	0.64	0.59	45
accuracy			0.77	668
macro avg	0.78	0.77	0.77	668
weighted avg	0.78	0.77	0.77	668



# SVM vs CNN

Postoji par kategorija po kojima možemo diskutovati prednosti i mane ovih pristupa rešavanju problema:

- Osnovna metrika koju koristimo za procenjivanje modela je tačnost na skupu za testiranje  
U tom smislu su se oba CNN modela pokazala boljim od SVM pristupa
- U pogledu jednostavnosti implementacije, budući da SVM zahteva prethodno izdvajanje deskriptora, a da se u slučaju CNN-a mogu samo primeniti gotove funkcije, prednost ima CNN
- Prednost SVM-a je u tome što je za obradu podataka i postavljanje modela potrebno znatno manje resursa nego u slučaju CNN-a, međutim korišćenje pretreniranih modela ublažava ovo u velikoj meri

# Izazovi i (neuspeli) eksperimenti

Izdvajanje svojstava u slučaju SVM-a, kao i treniranje CNN modela zahtevaju izvesne resurse (npr. jači procesor i grafiku), što može dovesti do toga da proces pripreme podataka, odnosno obučavanja bude vremenski zahtevan. Za potrebe ovog rada je, zbog tih problema, korišćena Google Cloud Compute VM instanca sa 2 NVidia P100 graficke (300\$ “trial” kredita ~ 150 sati), dedicated instancom sa “screen-om”. Uprkos tome treniranje inicijalnog CNN modela (<http://cs229.stanford.edu/proj2018/report/41.pdf>) je bilo neuspešno, jer se epoha trenirala 6 minuta a bilo je potrebno 500 epoha.

# Izazovi i (neuspeli) eksperimenti

Takodje je isprobano i treniranje od nule, DenseNet121 mreže, kojoj je bilo potrebno mnogo vremena, zbog čega je usled tehničkih uslova pređeno na korišćenje pomenutih pretreniranih “imagenet” modela.

Jedan od eksperimenata kada su u pitanju CNN modeli, bilo je i korišćenje ResNet152v2 mreže (priložen history file), međutim, ona je dala inferiornije rezultate u odnosu na prethodne modele, kao i EfficientNetB5.

Neupešan eksperiment je i izdvajanje SIFT atributa slike u slučaju SVM-a i njegovo kombinovanje sa ostalim atributima iako mu se u referentnom radu ne pridaje preveliki značaj.

# Poređenja rezultata sa referentnim radom

Model	Tačnost (trening skup %)	Tačnost (test skup %)	Vreme treniranja	Atributi slike
SVM - ref. rad	97.9	68.1	6.78s	GIST, Hu momenti
SVM - ref. rad	97.1	61.2	5.12s	GIST, hist. boja, Hu momenti
SVM	0.57	49.3	5.8s	GIST, hist. boja, Hu momenti
CNN - ref. rad	81.3	74.7	28921s	
DenseNet121	73.4	68.9	~15min (pretrenirani slučaj)	
ResNet50v2	97.4	77	~15min (pretrenirani slučaj)	

# ZAKLJUČAK

Prilikom rešavanja problema klasifikacije slika na osnovu umetnika koji ih je naslikao (skup od 7462 slike i 15 klasa) parametri uspešnosti pristupa koji su razmatrani bili su tačnost, zahtevnost implementacije i resursa potrebnih prilikom treniranja modela.

Ukupno gledano, CNN odnosi prednost, sa posebnim osvrtom na korišćenje već istreniranih modela Keras biblioteke čime se proces znatno optimizuje.

U slučaju SVM-a postoji potencijal za poboljšanje tačnosti ispitivanjem različitih kombinacija atributa slike, ali za razliku od rezultata referentnog rada ne primećuje se problem preprilagođavanja.

# KORISNA LITERATURA

- Referentni rad:  
<http://cs229.stanford.edu/proj2018/report/41.pdf>
- <https://github.com/matf-ml/materijali-sa-vezbi/>
- <https://keras.io/api/applications/>
- <http://cs231n.stanford.edu/reports/2017/pdfs/406.pdf>

# Dopune i poboljšanja – SVM sa GridSearch – om

Matrica konfuzije:

[	31	3	2	1	2	0	3	2	1	3	1	1	2	1	2]
[	5	19	0	1	0	6	1	0	1	0	2	10	5	1	4]
[	1	3	24	4	1	0	9	4	0	0	0	0	3	2	3]
[	1	1	5	20	5	3	7	6	0	0	2	0	1	1	3]
[	2	2	0	6	14	9	6	2	0	0	5	0	6	1	1]
[	0	1	1	5	7	23	6	5	0	1	1	0	4	1	0]
[	0	1	8	2	3	7	17	5	0	0	0	0	3	1	7]
[	1	1	3	5	3	1	6	28	0	0	1	1	3	1	1]
[	5	0	1	0	0	0	0	0	44	5	0	0	0	0	0]
[	3	0	0	0	0	0	0	0	2	50	0	0	0	0	0]
[	2	2	0	1	13	4	3	1	0	0	24	0	2	1	1]
[	2	7	2	0	3	1	2	1	1	0	2	26	2	5	1]
[	3	6	2	1	3	9	3	4	0	2	2	2	16	2	0]
[	5	2	5	5	0	3	5	2	1	1	0	1	3	22	0]
[	4	0	8	2	6	0	4	1	0	0	3	5	1	2	19]]

Classification report:

	precision	recall	f1-score	support
0	0.48	0.56	0.52	55
1	0.40	0.35	0.37	55
2	0.39	0.44	0.42	54
3	0.38	0.36	0.37	55
4	0.23	0.26	0.25	54
5	0.35	0.42	0.38	55
6	0.24	0.31	0.27	54
7	0.46	0.51	0.48	55
8	0.88	0.80	0.84	55
9	0.81	0.91	0.85	55
10	0.56	0.44	0.49	54
11	0.57	0.47	0.51	55
12	0.31	0.29	0.30	55
13	0.54	0.40	0.46	55
14	0.45	0.35	0.39	55
accuracy			0.46	821
macro avg	0.47	0.46	0.46	821
weighted avg	0.47	0.46	0.46	821

# ResNet50v2 – regularizacija

Kao što je sugerisano isprobani su metodi regularizacije l2 i weight decay na nivou optimizatora modela.

Patience je povećan na 6 epoha za EarlyStopping.

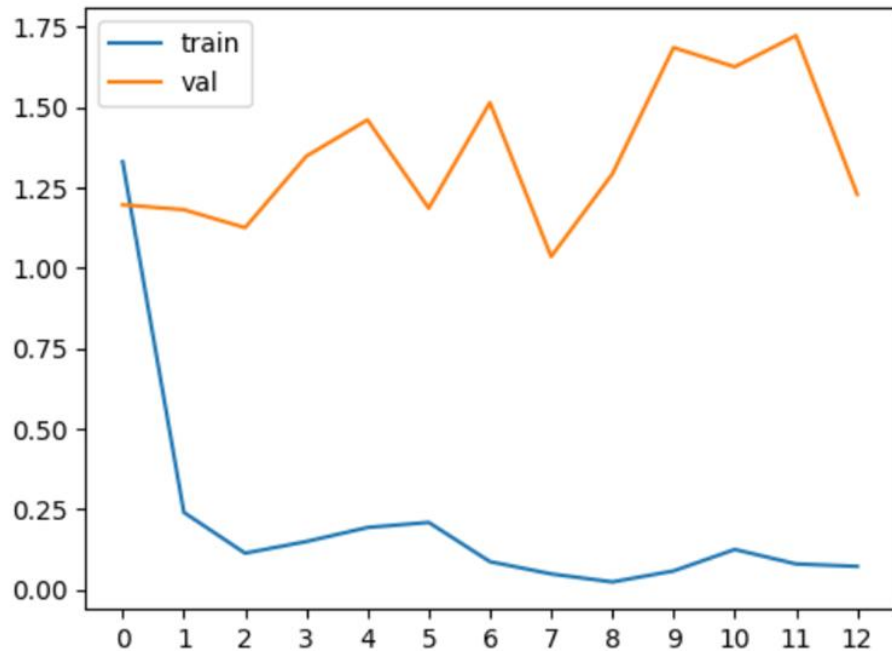
Model se i dalje prilagođava, doduše nešto sporije nego pre. Rezultati približni prethodnim su postignuti i sa weight decay-em. Najbolje rezultate na test setu je dao model sa l2 regularizacijom (history file: `resnet_l2_adam.json`) gde je dobijena neznatno veća tačnost na test skupu (78%, prethodno 77%).

Regularizacija l2 je imala parametar 0.0015, a veća vrednost ovog parametra je davala lošije vrednosti (npr. 0.004 je već u velikoj meri uticalo na ukupnu tačnost i prilagođavanje podacima).

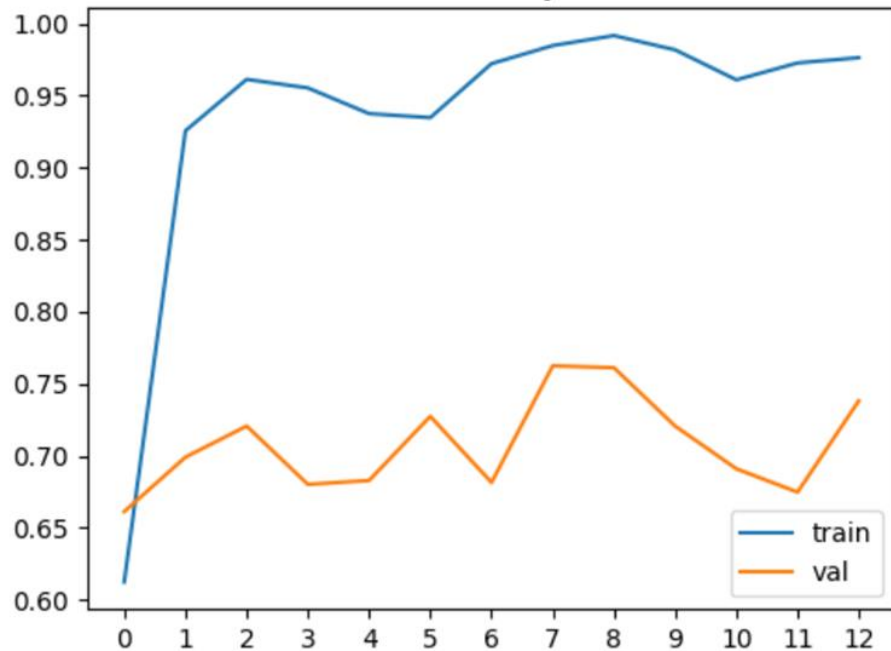


# ResNet50v2 – regularizacija

Loss



Accuracy



# Prikaz broja slika po autoru – izbalansiranost klasa

