Univerzitet u Beogradu
Matematički fakultet

Projekat iz predmeta Mašinsko učenje 2023/2024.

Klasifikacija slika odeće

Slobodan Jovanović 1050/2023
Jovan Rumenić 1047/2023

1 Uvod

U današnjem svetu, klasifikacija slika predstavlja ključnu komponentu u mnogim aplikacijama koje se oslanjaju na prepoznavanje vizuelnih obrazaca. Jedna od oblasti primene ove tehnologije je klasifikacija odeće, koja se koristi u e-trgovini, pretragama slika i preporukama proizvoda. Korišćenjem metoda mašinskog učenja, posebno dubokih neuronskih mreža, moguće je automatski identifikovati različite kategorije odeće na osnovu slika. Ovaj rad istražuje primenu tehnika mašinskog učenja u procesu klasifikacije odeće, sa posebnim fokusom na upotrebu konvolucionih neuronskih mreža (CNN) kao jednog od najefikasnijih alata za prepoznavanje vizuelnih sadržaja.

2 Ciljevi projekta

Cilj projekta je učenje modela da prepozna koji tip odeće je na slici. Koristićemo tehnike preprocesiranja podataka, optimizacije modela i evaluacije kako bi dobili što bolji model.

3 Skup podataka

Koristićemo skup podataka sa sajta *Kaggle*. Skup podataka sadrži 3781 slika koje su podeljene u 10 kategorija. Kategorije su:

1. dress
2. hat
3. logsleeve
4. outwear
5. pants
6. shirt
7. shoes
8. shorts
9. skirt
10. t-shirt

4 Preprocesiranje podataka

Preprocesirali smo podatke tako što smo:

- Skalirali sve slike na dimenziju 224x224 piksela.
- Konvertovali smo sve slike u tenzore.
- Normalizovali smo tenzore uz pomoć srednih vrednosti [0.485, 0.456, 0.406] i standardnie devijacije [0.229, 0.224, 0.225].

- Korišćenjem augmentacije podataka kako bi sprečili overfitting transformisali smo svaku 20-tu sliku iz skupa tako što smo je translirali, prevrnuli horizontalno, rotirali i zumirali. Tako ćemo dobiti generalniji model za korišćenje.

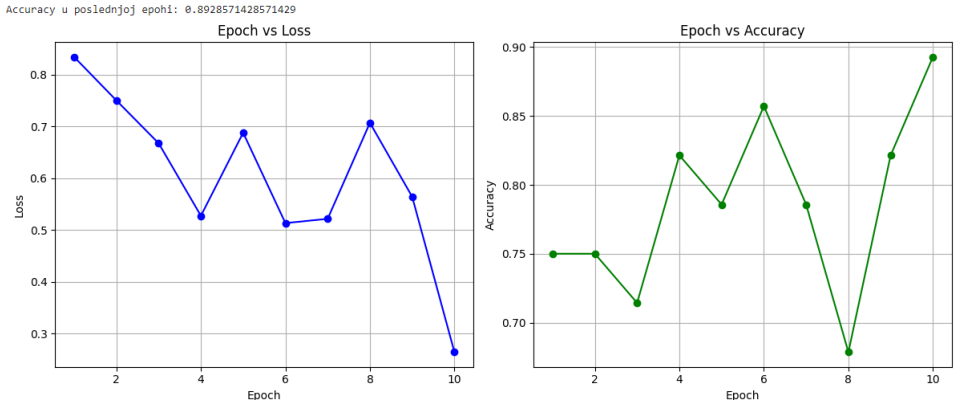
5 Modeliranje

5.1 ResNet50

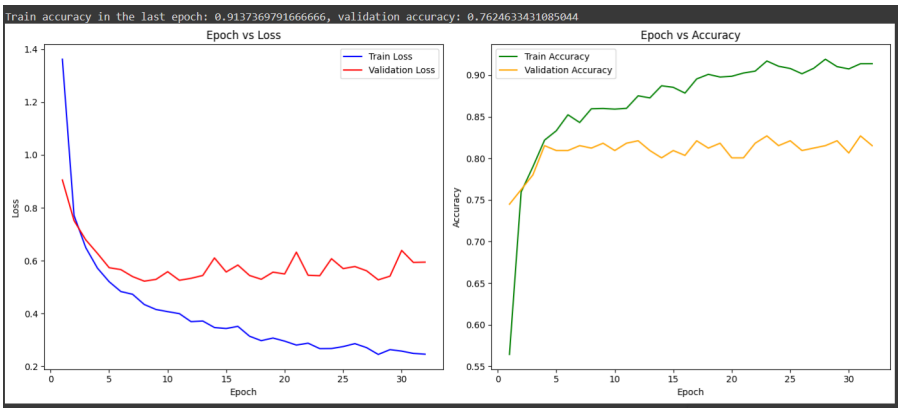
ResNet50, skraćeno od Residual Network sa 50 slojeva, je duboka konvoluciona neuronska mreža (CNN) arhitektura koja je uvedena kako bi rešila problem nestajanja gradijenata u veoma dubokim mrežama. Ključna inovacija u ResNet-u je uvođenje rezidualnih veza. Ove veze omogućavaju mreži da zaobiđe jedan ili više slojeva, što olakšava optimizaciju i omogućava efikasnije učenje. ResNet50 konkretno ima 50 slojeva, uključujući konvolucione slojeve, slojeve za normalizaciju i potpuno povezane slojeve. Široko se koristi u zadacima klasifikacije slika zbog svog balansa između dubine i računarske efikasnosti.

Koristićemo i Adam optimizator kao i unakrsnu entropiju kao funkciju greške.

Korišćenjem osnovnog ResNet50 modela na trening skupu smo tokom treniranja dobili sledeće statistike:



Slika 1: Preciznost i tačnost base ResNet50 modela na 10 epoha.

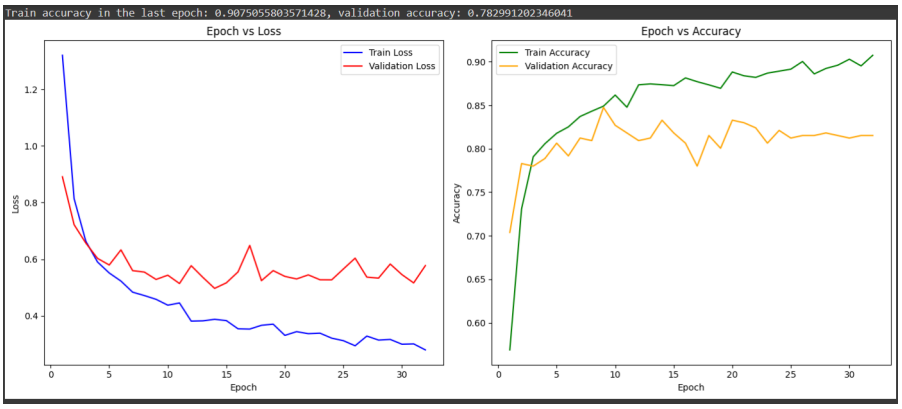


Slika 2: Preciznost i tačnost base ResNet50 modela na 32 epohe na trening i validacionom skupu.

Kako bi izbegli potencijalno overfittovanje iskoristićemo tehinku **augmentacije podataka**. Transformicali smo podatke narednim transformacijama:

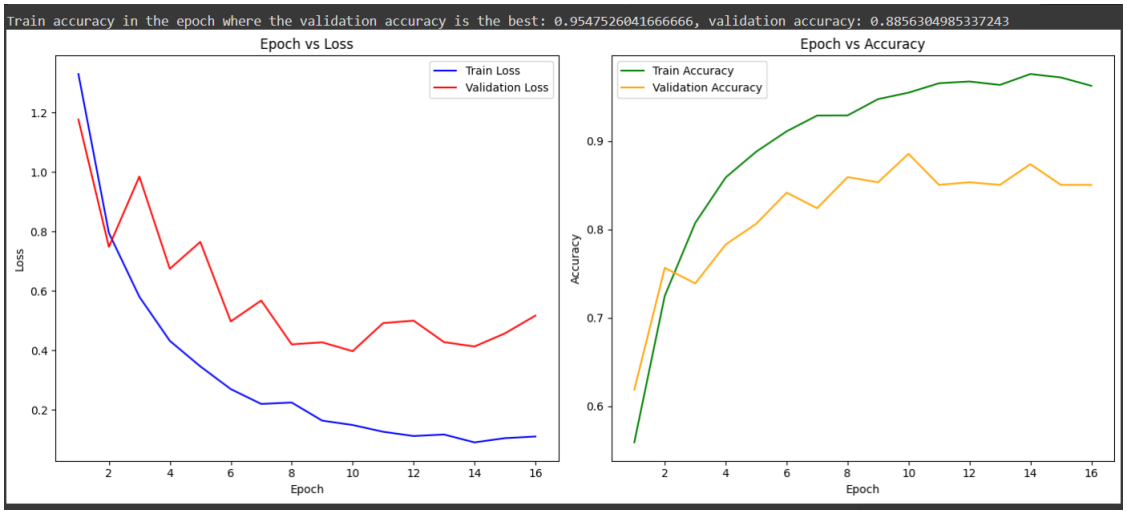
- Rotacijom koja nasumično rotira sliku za ugao iz opsega [-20, +20].
- Nasumičnom translacijom pomeramo sliku horizontalno ili vertikalno za do 10% veličine slke.
- Nasumično izrezivanje dela slike.
- Nasumično prevrtanje slike horizontalno sa verotnoćom od 50%

Ove transformacije se često koriste u augmentaciji podataka kako bi se veštački povećala raznovrsnost skupa za treniranje. Primena nasumičnih rotacija, pomeranja, izrezivanja i horizontalnog prevratanja pomaže modelu da prepozna objekte u različitim uslovima, što poboljšava njegovu sposobnost da generalizuje na nepoznate podatke.



Slika 3: Preciznost i tačnost ResNet50 modela korišćenjem augmentacije podataka.

Iskoristićemo na kraju i tehniku **finetuning**. Finetuning je tehnika u mašinskom učenju, gde se unapred trenirani model dodatno trenira na novom skupu podataka koji je specifičan za zadatak koji želite da rešite. Umesto da trenirate model od nule, koristite već postojeće znanje modela, prilagođavajući ga za vaš konkretan problem. ResNet50 model je unapred trenirani na ImageNet skupu podataka. Korišćenjem tehnike finetuning na poslednje konvolutivne slojeve dobijamo:



Slika 4: Preciznost i tačnost ResNet50 modela korišćenjem tehnike finetuning.

Koristili smo ovde i rano zaustavljanje ukoliko se tačnost na validacionom skupu ne poveća za više od 6 epoha. Na ovom primeru je izvršeno 16 epoha.

5.2 Evaluacija modela

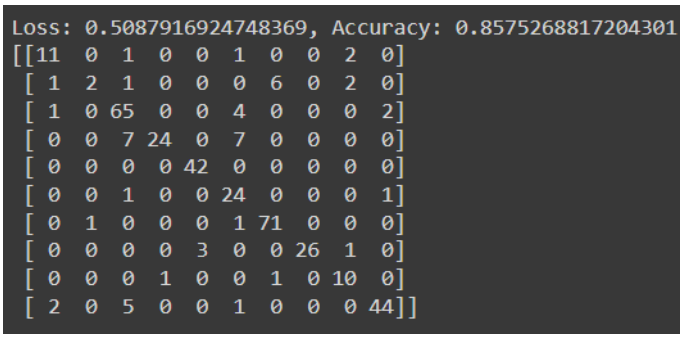
Evaluirali smo na test skupu sva 3 modela i dobili naredne rezultate:

```
Loss: 0.4596870057685401, Accuracy: 0.8387096774193549
[[12  0  0  0  0  1  0  2  0  0]
 [ 0 10  0  1  0  0  1  0  0  0]
 [ 1  1 53  4  0  4  0  1  2  6]
 [ 0  0  4 31  0  2  0  0  1  0]
 [ 0  0  0  0 39  0  0  2  1  0]
 [ 0  0  1  2  0 21  0  1  0  1]
 [ 0  1  0  0  0  1 71  0  0  0]
 [ 0  0  0  0  1  0  0 29  0  0]
 [ 1  1  0  0  0  0  0  2  8  0]
 [ 3  0  6  0  0  5  0  0  0 38]]
```

Slika 5: Gubitak, preciznost i matrica konfuzije osnovnog modela.

```
Loss: 0.4803991975483074, Accuracy: 0.8413978494623656
[[13  0  0  0  0  1  0  1  0  0]
 [ 0  9  0  1  0  0  2  0  0  0]
 [ 3  2 45  8  0  5  0  0  1  8]
 [ 0  0  1 35  0  1  0  0  1  0]
 [ 0  0  0  0 42  0  0  0  0  0]
 [ 1  0  0  8  0 14  0  1  0  2]
 [ 0  0  0  0  0  0 73  0  0  0]
 [ 0  0  0  1  2  0  0 27  0  0]
 [ 1  0  0  1  0  0  0  2  8  0]
 [ 1  0  1  0  0  2  0  1  0 47]]
```

Slika 6: Gubitak, preciznost i matrica konfuzije modela gde se koristila augmentacija podataka..



Slika 7: Gubitak, preciznost i matrica konfuzije modela gde se koristila tehinka finetuning..

Na osnovu preciznosti modela na test podacima se može videti da je predviđanje modela bolje sa povećanjem broja tehnika koje koristimo. Model u kom se koristi tehnika finetuning u kom se koristi i augmentacija podataka se pokazao kao najbolji, međutim nije velika razlika u odnosu na osnovni dataset. Dobar rezultat osnovnog je i posledica činjenice da je već treniran na velikom skupu podata pre našeg korišćenja.

5.3 VGG

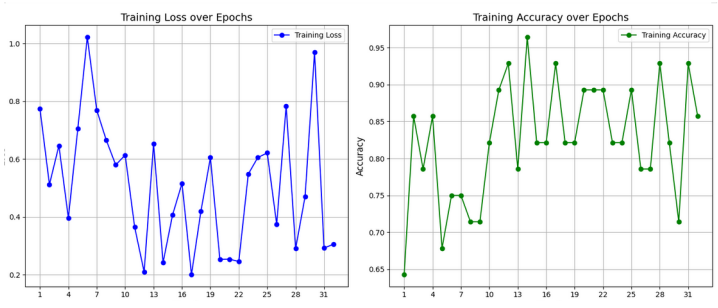
VGG16 je duboka konvolucijska neuronska mreža koja je postala poznata zbog svoje jednostavne i efektivne arhitekture. Razvijena je od strane Visual Geometry Group (VGG) sa Univerziteta u Oxfordu. Ima 16 slojeva koji uče i koriste parametre tokom obuke, uključujući 13 konvolucijskih slojeva i 3 sloja potpuno povezana (fully connected). Svi konvolucijski slojevi koriste male 3x3 filtre, što omogućava mreži da uči složene obrasce sa relativno malim brojem parametara. Aktivacijska funkcija korišćena u mreži je ReLU (Rectified Linear Unit), koja poboljšava sposobnost mreže da modeluje nelinearne odnose. VGG16 koristi maksimalno smanjenje (max pooling) za smanjenje dimenzija između konvolucijskih slojeva. Mreža je trenirana na ImageNet datasetu koji sadrži veliki broj slika i oznaka. Njena struktura omogućava prepoznavanje složenih obrazaca i objekata u slikama. VGG16 je postao popularan zbog svoje sposobnosti da postigne visoku tačnost u prepoznavanju slika. Upotrebljava se u različitim primenama uključujući detekciju objekata, segmentaciju slika i prepoznavanje lica. Njena arhitektura se često koristi kao osnovna mreža u transfer učenju. Zbog svoje jednostavnosti, VGG16 se lako razume i implementira. Iako je efikasna, ima veliki broj parametara, što može povećati zahteve za memorijom i računarstvom. Mreža je prvotno predstavljena u istraživačkom radu "Very Deep Convolutional Networks for Large-Scale Image Recognition". VGG16 je postavila nove standarde u performansama na benchmark testovima prepoznavanja slika kada je predstavljena. Sa velikim brojem slojeva, omogućava modelima da uče hijerarhiju složenih karakteristika. Njena struktura se često koristi kao referenca za dizajn drugih dubokih mreža. Mreža je veoma korisna u eksperimentalnim istraživanjima i komercijalnim aplikacijama zbog svoje fleksibilnosti i efikasnosti.

Koristićemo i Adam optimizator kao i unakrsnu entropiju kao funkciju greške.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=10, bias=True)
  )
)
```

Slika 8: Izgled VGG16 modela

Korišćenjem osnovnog VGG16 modela na trening skupu smo tokom treniranja dobili sledeće statistike:

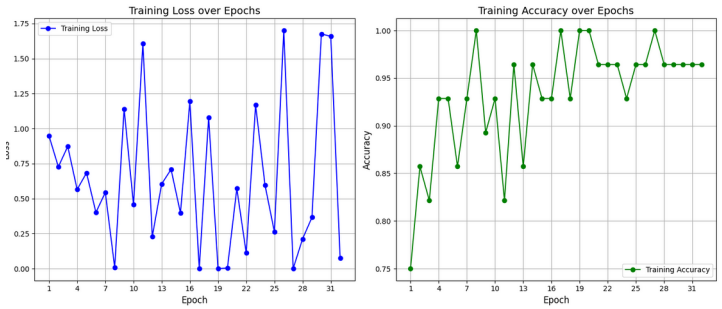


Slika 9: Preciznost i gubitak base VGG16 modela na 32 epoha.

Preciznost i gubitak u poslednjoj epohi: Epoch 32: [Loss: 0.3922, Accuracy: 0.8572]
Preciznost i gubitak na test skupu: Test Loss: 0.5679, Test Accuracy: 0.8010.

Kako bi poboljšali učenje i rezultate modela prilagodili smo ga našim potrebama tako što smo u sam model dodali nekoliko slojeva(Relu, Dropout, Linear sa 4096 neurona).

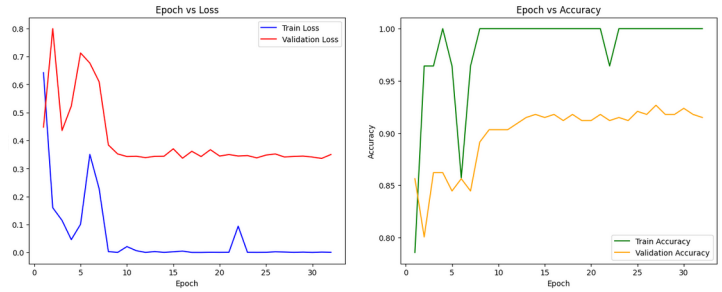
Preciznost i gubitak u poslednjoj epohi: Epoch 32: [Loss: 0.5709, Accuracy: 0.9788]



Slika 10: Preciznost i gubitak custom VGG16 modela na 32 epoha.

Preciznost i gubitak na test skupu: Test Loss: 5.8660, Test Accuracy: 0.8646.

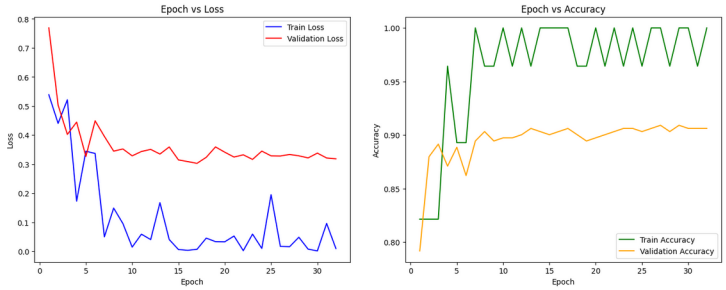
Kako bi dalje poboljšali ponašanje našeg modela dodali smo skup za validaciju i Fine-tuning koji omogućava bržu obuku i bolje performanse na specifičnim zadacima jer model već ima osnovno znanje koje se može dodatno prilagoditi.



Slika 11: Preciznost i gubitak custom VGG16 modela na 32 epoha i odnos sa validacionim skupom.

Preciznost i gubitak u poslednjoj epohi(za trening i validacioni skup):
Epoch 32: [Train Loss: [0.0039], Train Accuracy: [0.9987], Valid Loss: [0.3499], Valid Accuracy: [0.9150]]
Preciznost i gubitak na test skupu: Test Loss: 0.3445, Test Accuracy: 0.9141

I poslednje unapredjenje modela je data augmentation. Data augmentation (proširenje podataka) je tehnika učenja mašine koja se koristi za povećanje raznolikosti skupa podataka proširivanjem postojećih podataka. Cilj je poboljšanje generalizacije modela i sprečavanje prenaučenosti (overfitting) tako što se modelu pruža više varijacija u podacima.



Slika 12: Preciznost i gubitak custom VGG16 modela na 32 epoha i odnos sa validacionim skupom.

Preciznost i gubitak u poslednjoj epohi(za trening i validacioni skup):
Epoch 32: [Train Loss: [0.0360], Train Accuracy: [0.9883], Valid Loss: [0.3185], Valid Accuracy: [0.9062]]
Statistika na test skupu: Test Loss: 0.2647, Test Accuracy: 0.9375 Precision: 0.9369, Recall: 0.9355, F1 Score: 0.9352

Ovi rezultati pokazuju da model ima vrlo dobre performanse. Test Loss od 0.2647 ukazuje na to da model relativno dobro aproksimira stvarne vrednosti, dok testna tačnost od 0.9375 znači da model ispravno klasifikuje oko 93.75% slučajeva na testnom skupu.

Precision od 0.9369 znači da je model tačan u 93.69% slučajeva kada predviđa pozitivne klase. Recall od 0.9355 pokazuje da model prepoznaje 93.55% stvarnih pozitivnih slučajeva. F1 Score od 0.9352 kombinuje precision i recall, dajući uravnoteženu meru ukupne tačnosti modela. Ovi rezultati sugerišu da je model pouzdan i efikasan u predviđanju, s minimalnim greškama.