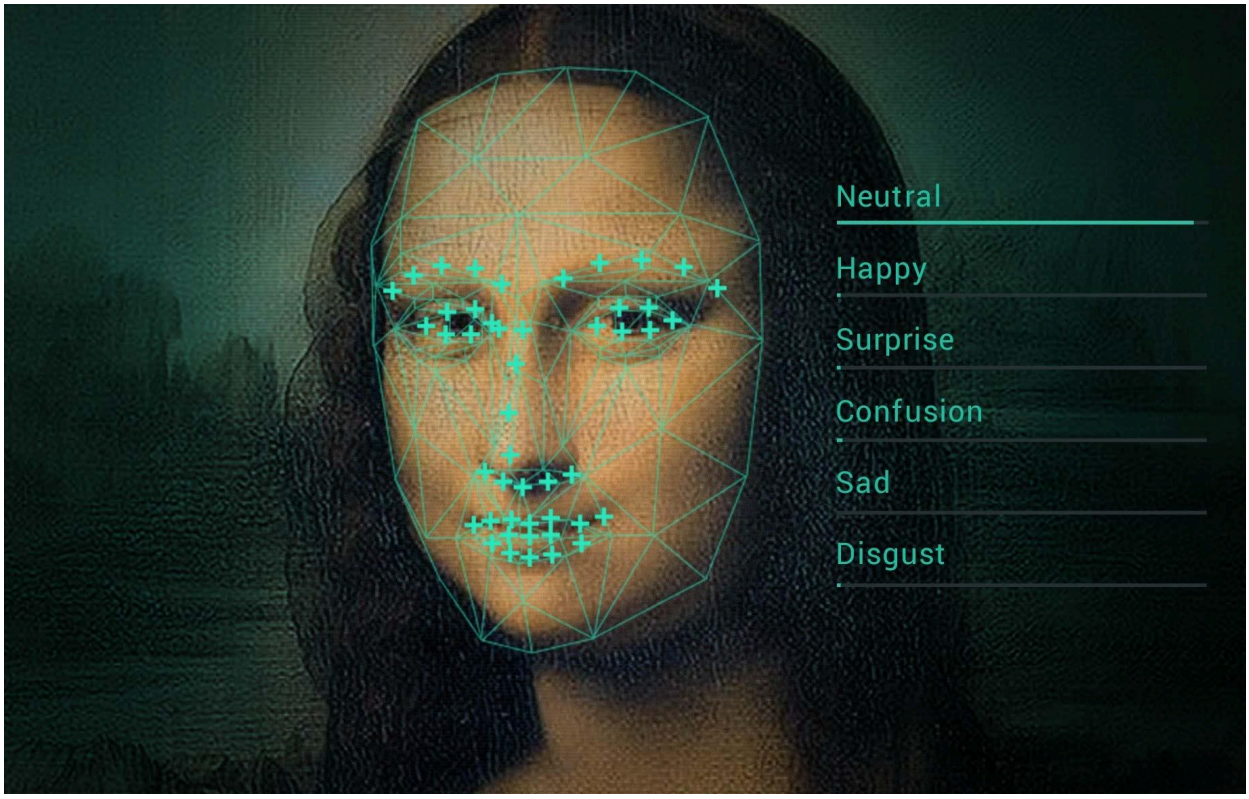


Prepoznavanje facijalnih ekspresija korišćenjem dubokih neuronskih mreža

27.09.2024

Mihailo Simić

Matematički fakultet, Univerzitet u Beogradu

Uvod

Duboke neuronske mreže (DNN) revolucionisale su oblast veštačke inteligencije, omogućavajući programima da uče iz ogromnih količina podataka i prave tačne predikcije. Ove mreže, inspirisane strukturom ljudskog mozga, sastoje se od više slojeva koji obrađuju i izvlače karakteristike iz ulaznih podataka. U poslednjim godinama, duboke neuronske mreže, a posebno konvolucione neuronske mreže (CNN), pokazale su izvanredne performanse u zadacima obrade slika, kao što su detekcija objekata, klasifikacija i segmentacija.

Među različitim primenama dubokih neuronskih mreža, prepoznavanje slika se ističe kao značajna oblast istraživanja i razvoja. Sistemi za prepoznavanje slika dizajnirani su da analiziraju vizuelne informacije, interpretiraju obrasce i identifikuju objekte unutar slika. Ovi sistemi se široko koriste u mnogim industrijama, uključujući zdravstvenu zaštitu, bezbednost i zabavu. Ključni podskup prepoznavanja slika je prepoznavanje lica, koje se fokusira na identifikaciju i verifikaciju ljudskih lica. Tehnologija prepoznavanja lica danas se koristi u različitim primenama, od autentifikacije pametnih telefona do sistema nadzora, pružajući sigurne i efikasne metode za verifikaciju identiteta.

Projekat predstavljen u ovom radu fokusira se na razvoj sistema za prepoznavanje izraza lica korišćenjem tehnika dubokog učenja. Korišćenjem unapred definisanih skupova podataka o licima, sistem koristi arhitekturu konvolucione neuronske mreže (CNN) kako bi klasifikovao slike u različite emocionalne kategorije. Ovaj rad predstavlja jednu ideju rešenja za Kaggle challege, sa koga su takođe preuzeti i resursi za treniranje modela.

U okviru ovog projekta, sprovedene su **analiza skupa podataka**, **kreiranje različitih arhitektura neuronskih mreža**, kao i **prilagođena demonstracija**. Za izradu projekta korišćen je programski jezik Python, a za sve vezano za kreiranje, treniranje i evaluaciju neuronskih mreža, modela, optimizatora i slično, korišćena python biblioteka PyTorch.

Ciljevi

I. Analiza skupa podataka
II. Duboke neuronske mreže
III. Modeli
IV. Demonstracija upotrebe

Analiza skupa podataka

U ovom projektu korišćen je skup podataka koji sadrži ukupno **35.887 slika** različitih lica, pri čemu svaka slika prikazuje jedan od sedam kategorisanih izraza lica: **"srećan" (happy)**, **"tužan" (sad)**, **"strah" (fear)**, **"gađenje" (disgust)**, **"neutralan" (neutral)**, **"ljut" (angry)** i **"iznenađenje" (surprise)**. Skup podataka je podeljen u tri dela: **Trening skup**, **Test skup** i **Validacioni skup**.

- **Trening skup** sadrži **28.709 slika**, koje se koriste za učenje modela i prilagođavanje njegovih parametara.
- **Test skup** i **Validacioni skup** sadrže po **3.589 slika** svaki. Validacioni skup služi za proveru performansi modela tokom treniranja, dok se Test skup koristi za konačnu evaluaciju modela nakon završenog treniranja.

Sve slike su u formatu **48x48 piksela** i crno-bele (grayscale), što znači da svaka slika sadrži samo informacije o intenzitetu svetlosti, bez boje. Ovaj format olakšava obradu i analizu podataka, dok smanjena rezolucija omogućava efikasnije treniranje modela, smanjujući potrebu za velikom količinom računarskih resursa (i na ovom skupu je trajalo satima).

Podaci su mogli biti korišćeni u dva oblika:

1. Kao .csv fajl u kome jedan unos je jedna slika, gde niz brojeva predstavlja vrednosti osvetljenosti svakog piksela.
2. Kao prava slika(ovde je .png formata)

Iako resursi predstavljeni kao slke zauzimaju više prostora, bolje su za vizualizaciju skupa podataka, kako zapravo izlgedaju, šta se na njima nalazi i koliko je evidentno koja facijalna ekspresija se nalazi na slici i nama kada je posmatramo okom, i samim tim može da nam olakša ili eventualno ukaže na strukturu neuronske mreže koje ćemo iskoristiti.

Možemo ispod pogledati neke od slika iz trening skupa podataka:

Možemo primetiti da su neke od slika skro pa karikature(druga slika), kao i da nisu slikane iz istih uglova, čak na nekim slikama je deo lica pokriven. Ovo omogućava detaljniju analizu i bolje rezultate jer se sprečava da se modeli preprilagode i da recimo određeni kadar ili ugao slikanja kategorišu kao jednu emociju, već da se zaista nauče karakteristike lica prilikom različitih facijalnih ekspresija.

Duboke neuronske mreže

Uvod

Ime duboke neuronske mreže potiče od njihove strukture. Svaka neuronska mreža ima ulazni sloj, izlazni sloj koji u zavisnosti od problema (klasifikacije ili regresije) mogu imati jedan ili više čvorova. Čvor u neuronskoj mreži zovemo neuronom. Između ova dva sloka može da se nađe proizvoljan broj slojeva, gde se svaki sloj, takođe, sastoji od proizvoljnog broja neurona. U zavisnosti od tipa neuronske mreže ovi slojevi mogu biti potpuno ili delimično povezani. Ako je svaki neuron jednog sloja mreže povezan sa svakim neuronom susednog sloja mreže, onda ovu mrežu (ili bar ove slojeve) nazivamo potpuno povezanim. Neuronska mreža koja se sastoji od potpuno povezanih slojeva neurona se naziva potpuno povezana neuronska mreža. Dubokim mrežama se nazivaju one koje sadrže više od jednog sloja, takozvanog skrivenog sloja, između ulaznog i izlaznog.

Konvolutivne neuronske mreže (CNN) predstavljaju specifičnu vrstu dubokih neuronskih mreža koje su posebno uspešne u zadacima obrade slika i prepoznavanja obrazaca. Osnovni koncept konvolutivnih mreža zasniva se na procesu **konvolucije**, matematičke operacije koja omogućava ekstrakciju značajnih karakteristika iz podataka, poput ivica, oblika i tekstura, sa slike. Tokom konvolucije, mala matrica poznata kao **filter** (ili kernel) prolazi kroz sliku i računa vrednosti po unapred definisanom uzorku, stvarajući novu sliku koja sadrži bitne informacije, dok se nebitni podaci filtriraju.

Konvolutivne mreže se sastoje od više slojeva, od kojih su najvažniji **konvolucioni slojevi**, koji primenjuju konvoluciju na ulazne podatke. Ovi slojevi automatski uče odgovarajuće filtere tokom procesa treniranja, što omogućava mreži da identifikuje složene obrasce, kao što su oblici lica i karakteristične tačke na slikama. Pored konvolucionih slojeva, CNN-ovi često sadrže **slojeve za smanjenje dimenzija** (eng. pooling layers), koji služe za redukciju veličine slike, čime se smanjuje broj parametara i računski zahtevi, dok se zadržavaju najvažnije informacije.

Konvolutivne neuronske mreže postale su standard u zadacima prepoznavanja slika, detekcije objekata i klasifikacije, zahvaljujući svojoj sposobnosti da automatski i efikasno izvlače karakteristike iz slika, bez potrebe za ručnim inženjeringom. Ove neuronske mreže se koriste u širokom spektru aplikacija, uključujući prepoznavanje lica, autonomna vozila, medicinsku dijagnostiku i mnoge druge oblasti.

Za potrebe ovog projekta su napravljene 4 duboke neuronske mreže. Jedna potpuno povezana neuronska mreža, i tri konvolutivne neuronske mreže različite arhitekture, broja slojeva i broja neurona, pa samim tim i različitim brojem trenirabilnih parametara.

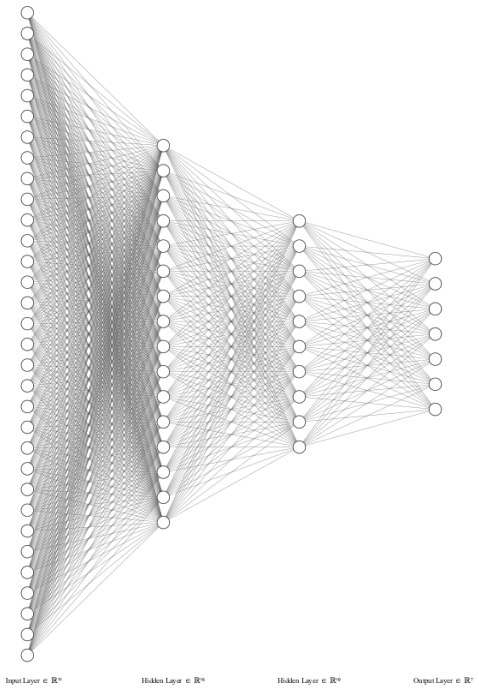
Potpuno povezana neuronska mreža (**FullyConnectedNN**)

Potpuno povezane neuronske mreže (Fully Connected Neural Networks - FCNN), poznate i kao višeslojne perceptronske mreže (MLP), predstavljaju osnovni tip neuronske mreže u kojoj su svi neuroni iz jednog sloja povezani sa svim neuronima iz sledećeg sloja.

Jedno rešenje koje smo napravili je neuronska mreža sa 3 skrivena potpuno povezana sloja. Ulazni sloj je takođe potpuno povezan sa prvim skrivenim slojem, i on ima 128 neruona. Pošto naše mreže treba da obrađuju slike koje su veličine 48x48 piksela, ako koristimo potpuno povezane mreže, ne možemo da ubacimo sliku bez ikakve modifikacije u formatu. Slike smo učitali i prebacili u tenzore (eng. tensor), ali da bi ulazni sloj naše potpuno povezane mreže to mogao da obradi mora da se uradi takozvano izravnjavanje (eng. flatten).

Ovo možemo da ubacimo u našu strukturu mreže kao fiktivan “flatten” sloj, a možemo i manuelno da uradimo. Za koju god opciju da se odlučimo bitno je da Ulazni broj u našu mrežu bude tačan, a to je 48*48=2304. Pošto smo rekli da prvi skriveni sloj ima 128 neurona, svaki neuron ulaznog sloja je povezan sa svakim od 128 neurona prvog skrivenog sloja. Drugi skriveni sloj smo odabrali da bude duplo manji i on ima 64 neurona, dok treći skriveni sloj ima 32 neurona koij su potpuno povezani i sa prethodnim skrivenim slojem, ali i sa izlaznim slojem naše mreže. Pošto imamo 7 klasa, odnosno 7 različitih facijalnih ekspresija koje želimo da razlikujemo ovde, izlazni sloj mreže ima tačno 7 neurona.

Za nelinearnu aktivacionu funkciju odlučili smo se za ReLU, kao standardno dobru opciju prilikom treniranja neuronskih mreža za probleme višeznačne klasifikacije. Bitno je naglasiti da se funkcija aktivacije primenjuje nakon svakog sloja, odnosno nakon što dobijemo izlaze kada provučemo neki podatak kroz taj sloj mreže. Na slici ispod se može otprilike steći utisak kako bi ova mreža izgledala (Naravno, nije crtano u pravim dimenzijama jer se ne bi videlo 4 hiljade neurona i veze između njih).



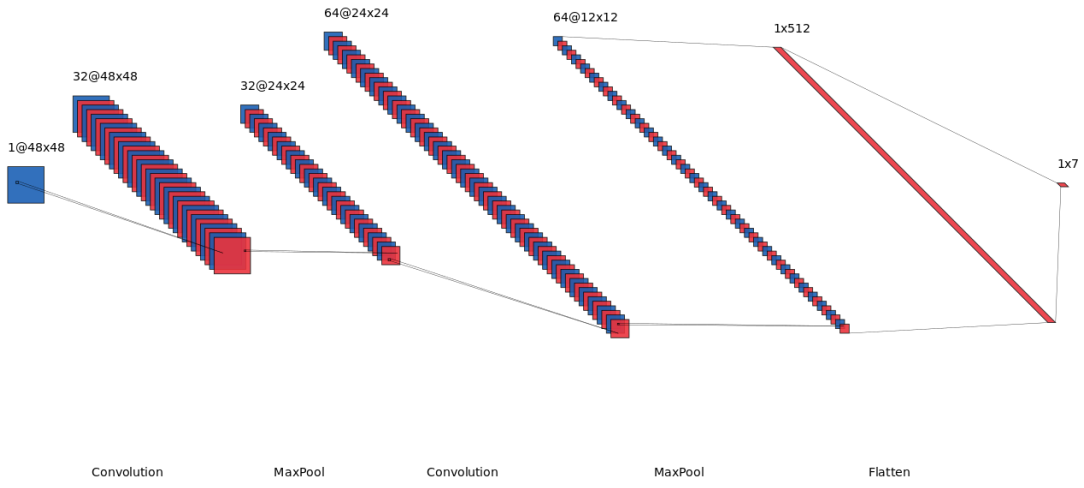
Iako ovo jeste duboka neuronska mreža, nismo očekivali da da preterano dobre (a pogotovo ne bolje od konvolutivnih), jer kao što smo već rekli, mi ovde kao ulaze imamo slike. Slike su složeni ulazi jer boja, količina svetlosti i ostale karakteristike, na samom jednom pikselu nam ne daje previše informacija o tome šta se na slici nalazi. Potpuno povezanu neuronsku mrežu smo napravili i iskoristili da bismo mogli da vidimo benefit koje će nam doneti konvolucija.

SimpleConvNN - konvolutivna neuronska mreža

Ovo je prva konvolutivna mreža koju smo napravili u cilju iskorišćavanja konvolucije radi bolje obrade ulaza, koje su nam slike, nego što je to radila potpuno povezana neuronska mreža. Ovo je slična arhitektura mreže koju smo korisitli kada smo radili prepoznavanje rukom pisanih cifara “MNIST”. Ono što je drugačije je da smo ovde imali drugačiju dimenziju slike, pa samim tim i drugačiji ulazni sloj mreže. Ova mreža ima sledeću arhitekturu:

- Ulazni sloj
- 2 konvolutivna sloja sa dimenzijom filtera 3x3
- 1 potpuno povezani sloj
- Izlazni sloj
- ReLU aktivaciona funkcija

Na slici ispod se može videti vizualizacija arhitekture ove neuronske mreže (imati na umu da vektor nije dimenzije 512 ali je stavljeno da bi moglo da se vizualizuje)



Ova mreža je dobar početak za naše potrebe jer koristi benefit konvolucije jer radimo sa slikama i želimo da izvlačimo složenije informacije koje se ne mogu lako (ili uopšte) videti kada to radimo sa potpuno povezanim neuronskim mrežama. Poređenje ove jednostavnije konvolutivne neuronske mreže i potpuno povezane mreže ćemo obraditi kasnije u poglalju Modeli.

EmotionNet - konvolutivna neuronska mreža

Napravili smo jednu konvolutivnu mrežu sa idejom da probamo da postignemo bolje tačnosti i što manje gubitke. Pošto projektujemo duboku neuronsku mrežu, odlučili smo se za sledeću arhitekturu:

- Ulazni sloj
- 3 konvolutivna sloja sa kernelom dimenzije 3x3 i padding-om 1 piksel
- 3 sloja smanjenja dimenzije (MaxPool) sa kernelom 2x2
- 1 potpuno povezani sloj sa 512 neurona
- 1 dropout sloj sa faktorom 0.5 radi regularizacije
- Izlazni sloj sa 7 neurona koliko ima i klasa
- ReLU nelinearna funkcija aktivacije

Ulazni sloj sada nema onoliko neurona koliko ima piksela slika, jer se ne radi o potpuno povezanom sloju, već se ulaznom sloju prosleđuje cela slika a ulazna dimenzija je broj kanala koje slika ima. Pošto su sve slike crno bele (greyscale), postoji samo jedan ulazni kanal.

Za veličinu kernel-a za konvoluciju odlučili smo se za 3x3 dimenziju jer bi veća dimenzija kernela brže smanjila dimenziju ulaza, a pošto su nam slike 48x48 piksela, bolje je da bude 3x3. Padding kod konvolutivnih slojeva nam služi da bismo u procesu konvolucije izvukli informacije i iz krajnjih piksela koji se nalaze na samoj ivici slike i samim tim bi u normalnim uslovima uvek bili po ivici kernel-a koji vrši konvoluciju. Padding od 1 piksela dodaje fiktivan “okvir” oko slike tako da kernel koji prolazi sliku kada radi konvoluciju “izađe van slike” a da u centru kernel-a budu ivični pikseli.

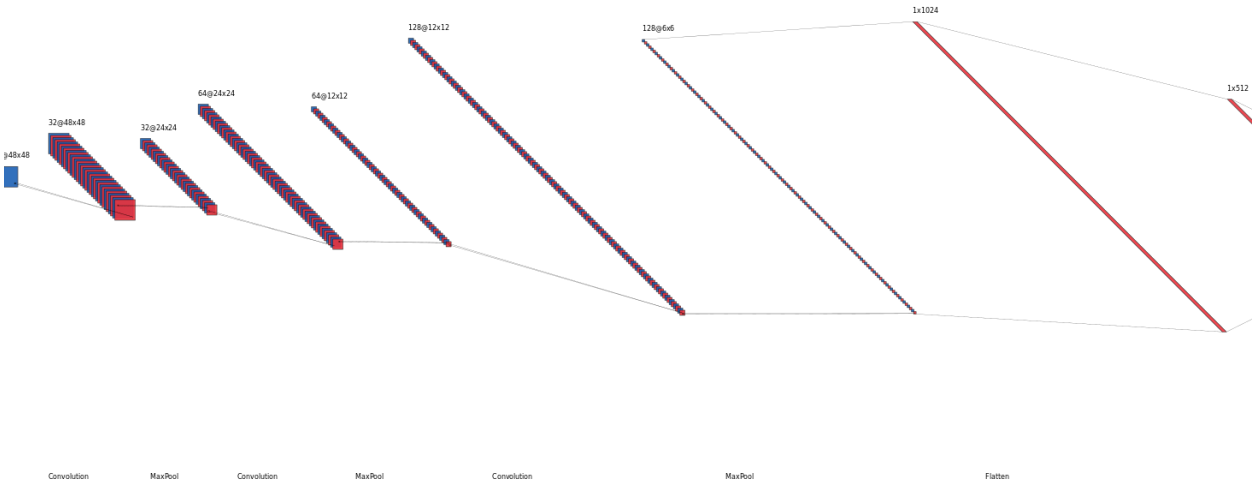
Nakon svake konvolucije se vrši *pooling*, a pošto je dimenzija *pooling* kernel-a 2x2, znači da se efektivno smanjuje dimenzija izlaza prethodnog sloja na pola po svakoj osi, jer se za svaki blok od 2x2 dimenzije uzima maksimum i time se čuvaju najbitnije informacije.

Često je praksa da se posle određenog broja konvolutivnih slojeva, ubace potpuno povezani slojevi pre nego što se poveže izlazni sloj(ili se opet uposli konvolutivni sloj ako je to potrebno). Ovde je stavljen jedan takav potpuno povezani sloj sa 512 neurona, koji je povezan sa svakim iz prethodnog i sledećeg sloja. Naravno, da bismo to postigli moramo da pre ovog sloja izlaze izravnamo (flatten), pa tek onda ubacimo u ovaj sloj.

Pre nego što ubacimo u poslednji, izlazni sloj neuronske mreže, dodali smo jedan dropout sloj koji će sa nekom verovatnoćom da odbaci određene neurone, odnosno izlaze iz tih neurona. Ovo smo uradili jer imamo dosta slojeva, 3 konvolutivna, pooling, jedan potpuno povezan, i radi regularizacije je ovde dobro da imamo jedan dropout sloj.

Funkcija aktivacije je i ovde ReLU, i nju pozivamo posle primena određenog sloja(bez obzira da li je konvolutivni ili potpuno povezani sloj), ali pre nego što uradimo smanjene dimenzije pooling-om.

Na slici ispod možemo videti otprilike kako bi ova mreža izgledala (Isto kao i kod drugih, obratiti pažnju da dimenzije krajnjeg vektora nisu ove jer bi bilo preveliko za vizualizaciju).



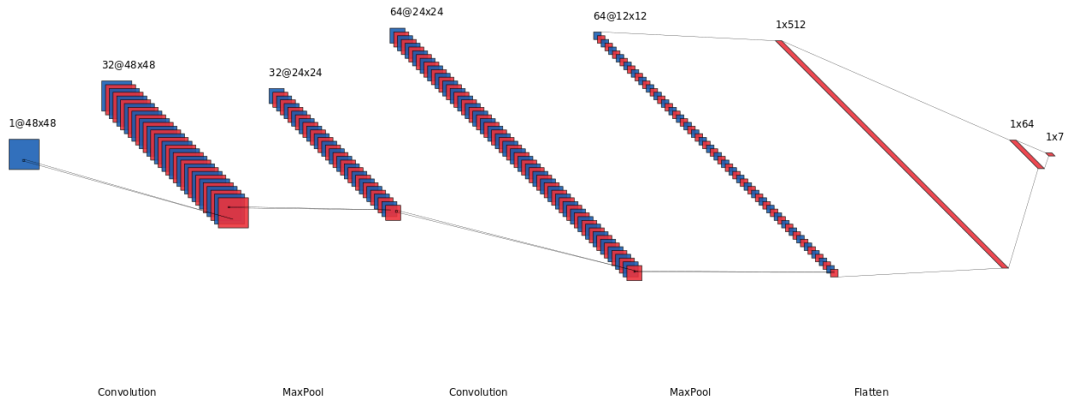
EDA_CNN - konvolutivna neuronska mreža

Ovo je jedna konvolutivna neuronska mreža koju smo našli upravo na Kaggle challenge-u, kao jedan on rešenja koje je postavljeno, i odlučili smo da radi poređenja uljkučimo i ovu arhitekturu, ali da sami istreniramo ovu mrežu kako bismo imali iste uslove kao i sa ostale tri mreže koje smo isprojektovali. Arhitekturu ove mreže smo našli na jednom od thread-ova na Kaggle platformi. Može se pogledati ovde: EDA_CNN.

Arhitektura EDA_CNN neuronske mreže je sledeća:

- Ulazni sloj
- 2 konvolutivna sloja sa filterom dimenzije 3x3
- 1 potpuno povezani sloj sa 64 neurona
- Izlazni sloj
- ReLU i Softmax aktivacione funkcije

Slika ispod prikazuje arhitekturu EDA_CNN konvolutivne neuronske mreže:



Ključna razlika ove mreže u odnosu na naše je što ova mreža pored ReLU aktivacione funkcije koja se primenjuje posle konvolutivnih slojeva, ima i Softmax aktivacionu funkciju koja se primenjuje na izlaznom sloju. Nije neophodno primeniti softmax aktivacionu funkciju na izlaz poslednjeg sloja ako koristiš **CrossEntropyLoss** u PyTorch-u. Razlog je taj što **CrossEntropyLoss** u PyTorch-u kombinuje i **softmax aktivaciju** i **negativni logaritam verovatnoće (NLLLoss)** u jednoj funkciji. Ipak smo ostavili ovako radi konzistentnosti.

Modeli

Svaka neuronska mreža je trenirana pojedinačno, i to po 10 epoha na skupu za obuku, nakon čega je evaluirana na validacionom skupu podataka. Za funkciju gubitka korišćena je **CrossEntropyLoss**, koja je odgovarala prirodi klasifikacionog problema, jer omogućava efikasno učenje razlika između različitih kategorija izraza lica.

Za optimizaciju modela izabrali smo **Adam** optimizer, koji je poznat po svojoj efikasnosti u različitim vrstama dubokih neuronskih mreža. Početna vrednost parametra učenja (learning rate) bila je postavljena na 0.001. Tokom treniranja, parametar učenja je korigovan i smanjivan u momentima kada je mreža prestala da konvergira, kako bi se poboljšala stabilnost učenja i izbeglo prekomerno osciliranje u vrednostima težina modela.

Ovaj proces omogućio je da svaki model bude bolje optimizovan, uz prilagođavanje parametara tokom faze treniranja, što je doprinelo postizanju boljih rezultata na validacionom skupu.

Takođe je vredno napomenuti da, iako nismo imali vremena da ih isprobamo, u implementaciju smo dodali i mogućnost korišćenja **schedulers** za dinamičko prilagođavanje stope učenja, kao što su **Cyclic Learning Rate (CLR)** i **StepLR**. Ove tehnike omogućavaju automatsko podešavanje stope učenja tokom treniranja, što bi potencijalno moglo da doprinese efikasnijem učenju i postizanju boljih rezultata.

Modele smo posle treniranja i evaluacije izvezli putem PyTorch ugradjene opcije za izvoz modela. Za svaki model su izvezene težine grana i parametri koji se posle mogu učitati i koristiti bilo za dalje treniranje, a to smo omogućili u projektu putem skripte, bilo za učitavanje i korišćenje za predikciju, što ćemo na kraju i demonstrirati. Bitno je napomenuti da smo se odlučili na čuvanje samo težina grana i parametara pre nego izvoz i čuvanje eclokupnog modela iz više razloga:

1. Izvoz celokupnog modela koristi više resursa, što utiče kako na samo skladištenje, tako i na brzinu čuvanja i učitavanje prilikom dalje upotrebe.
2. Čuvanjem samo težina i parametara modela smo sprečili potencijalne probleme oko učitavanja i ponovnog treniranja modela na drugim mašinama koje imaju drugačiju arhitekturu, verzije okruženja, biblioteka, pa čak i operativnog sistema. Na ovaj način smo osigurali da korisnik na svojoj mašini može učitati trenirane parametre modela, a da ne brine da li su modeli trenirani na istim verzijama.

Sada ćemo ukratko prikazati trenirane modele bazirane na neuronskim mrežama koje smo u prethodnom poglavlju objasnili.

FCNN

Model dobijen treniranjem potpuno povezane neuronske mreže je od svih koje smo trenirali je, očekivano, imao najlošije rezultate. Ovo je očekivano jer potpuno povezane mreže imaju ograničene mogućnosti kada su ulazi slike na kojima je potrebno pronaći šablone i karakteristike. Ovaj model ima 4 potpuno povezana sloja, i 305.607 trenirabilnih parametara.

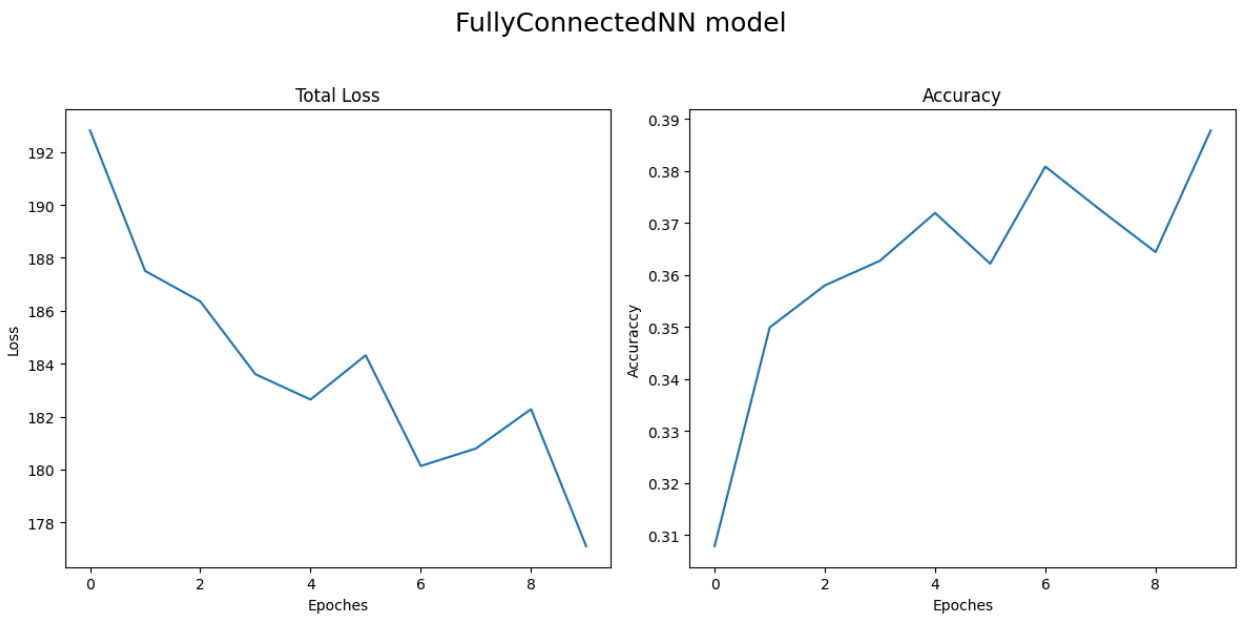
Treniranje ovog modela 10 epoha je na našoj mašini trajalo oko 10 minuta. Na slici ispod možemo videti kako su se kretali ***total loss*** kao i ***accuracy*** prilikom treniranja i evaluacije nad trening i validacionom skupu.

```
Training epoch: 1...
Total loss: 192.8188375234604
Accuracy: 0.30788520479242126
Training epoch: 2...
Total loss: 187.5046433210373
Accuracy: 0.3499582056283087
Training epoch: 3...
Total loss: 186.35391747951508
Accuracy: 0.35803845082195596
Training epoch: 4...
Total loss: 183.6011198759079
Accuracy: 0.36277514628030094
Training epoch: 5...
Total loss: 182.64474427700043
Accuracy: 0.3719699080523823
Training epoch: 6...
Total loss: 184.31645572185516
Accuracy: 0.36221788799108384
Training epoch: 7...
Total loss: 180.13262498378754
Accuracy: 0.3808860406798551
Training epoch: 8...
Total loss: 180.79382026195526
Accuracy: 0.3725271663415993
Training epoch: 9...
...
Accuracy: 0.38785176929506826
```

Možemo videti da kroz 10 epoha, gubitak je smanjen za oko 12, a tačnost je povećana za oko 7%. Obzirom na to da su ovo početne iteracije, možda bismo očekivali veći porast

tačnosti, ali zapravo je ovo u redu, ne zaboravimo da je ovo model potpuno povezane neuronske mreže i samim tim sporije kovergira jer sporije uočava pravila kad su u pitanju slike.

Na slici ispod možemo videti grafike kretanja gubitaka i tačnosti kroz iteracije.



Grafici nisu konstantno rastući i opadajući, čak i kada smo povećali brzinu učenja (**learning rate**). Tačnost se poboljšava kada smo pustili model da nastavi sa treningom novih 10 epoha, ali ni tu nismo imali neke preterano značajne razlike, svega do na nekoliko procenata. Ovo nam ukazuje na to da potpuno povezane neuronske mreže nisu najbolji izbor modela za rad sa slikama, prepoznavanjem šablona na slikama ili facijalnih karakteristika na licima. Poslednje vrednosti do kojih smo uspeli da dođemo je negde oko 42-43%.

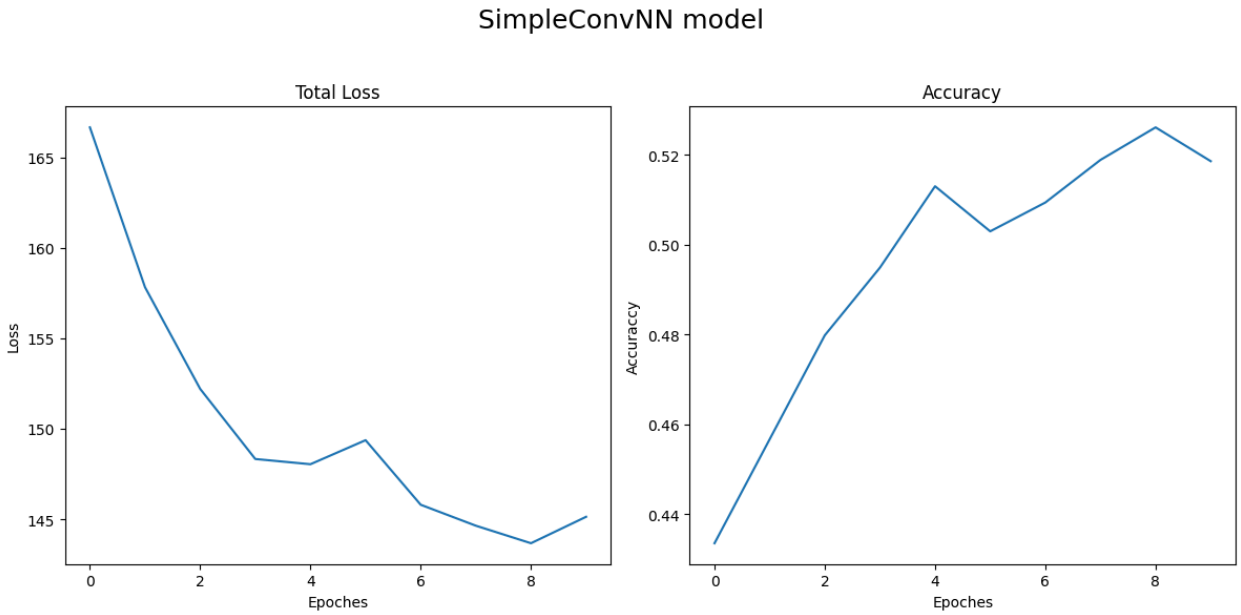
SimpleConvNN

Model koji smo dobili treniranjem jednostavne konvolutivne neuronske mreže. Ova mreža iako na pogled izgleda komplikovanija jer imamo sada slojeve konvolucije i pooling-a, ima znatno manji broj trenirabilnih parametara od modela potpuno povezane neuronske mreže. Ovaj model ima 63.623 trenirabilnih parametara. Sa druge strane, iako ima znatno manje trenirabilnih parametara, vreme treniranja 10 epoha traje duže upravo zbog pomenute operacije konvolucije i pooling-a. Imajući u vidu na naša mašina na kojoj smo trenirali nema grafičku karticu, sva izračunavanja se izvršavaju na procesoru (**CPU**). Pošto operacija konvolucije praktično uključuje matrice i šetanje matrice filtera (**kernel**), ova operacija je zahtevna, a pošto nemamo na raspolaganju grafičke processor koji ove operacije drastično olakšavaju, samim tim se ovaj model trenirao duže. Preciznije skoro duplo duže, negde oko 17-18 minuta za 10 epoha. Na slici ispod se mogu videti gubici i tačnosti tokom treniranja.

```
Training epoch: 1...
Total loss: 166.65481662750244
Accuracy: 0.43354694901086654
Training epoch: 2...
Total loss: 157.81692349910736
Accuracy: 0.4566731680133742
Training epoch: 3...
Total loss: 152.19933354854584
Accuracy: 0.4797993870158819
Training epoch: 4...
Total loss: 148.32393598556519
Accuracy: 0.4948453608247423
Training epoch: 5...
Total loss: 148.03550165891647
Accuracy: 0.5129562552242964
Training epoch: 6...
Total loss: 149.36358952522278
Accuracy: 0.5029256060183895
Training epoch: 7...
Total loss: 145.79551243782043
Accuracy: 0.5093340763443857
Training epoch: 8...
Total loss: 144.63767856359482
Accuracy: 0.5188074672610755
Training epoch: 9...
...
Accuracy: 0.518528838116467
```

VIdimo da je od potpuno povezanog modela u istih prvih 10 epoha bolji za više od 10%.

Na slici ispod možemo ispratiti kovergenciju vrednosti gubitka modela, kao i tačnost.



VIdimo da u odnosu na model potpuno povezane neuronske mreže, ima strmiji pad ukupnog gubitka u prvim epohama, kao i stabilniji rast tačnosti. Model u prvih 10 epoha treniranja je ostvario oko 51% tačnosti. Ovo je naizgled prilično loše jer je pogođena klasa u samo pola slučajeva, ali moramo se setiti da je ovo problem gde imamo 7 klasa, i da je ovo samo prvih 10 iteracija. Naknadnim pokretanjem treninga i evaluacije ovog modela smo došli do nekih 55-56% tačnosti posle otprilike 30 epoha.

EmotionNet

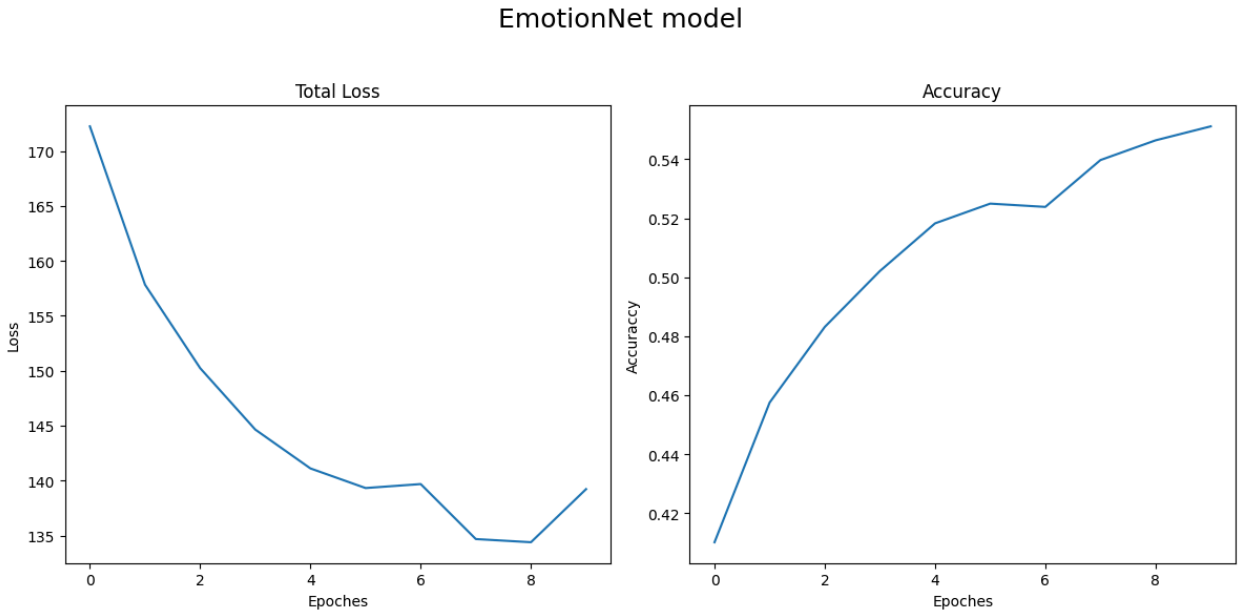
Model koji je od svih ovde navedenih najkompleksniji po strukturi, a i po broju trenirabilnih parametara. Pošto ovaj model ima 3 konvolutivna sloja, 2 potpuno povezana sloja, pooling-e i dropout sloj, broj trenirabilnih parametara je drastično porastao u odnostu na jednostavnu konvolutivnu mrežu koju smo prethodno naveli, a čak i u odnosu na potpuno povezanu neuronsku mrežu. Ovaj model ima 2.456.071 trenirabilnih parametara. To skoro 35 hiljada puta više nego jednostavniji model konvolutivne mreže. Ovaj model, kao što se može pretpostaviti je najduže morao da se trenira zbog velike količine parametara. Trening 10 epoha je trajao negde oko 30-35 minuta. Obzirom da je jako velika razlka u broju parametara, vreme treninga je bilo svega oko duplo više (što je za nas ispalo povoljno :)).

Slika ispod prikazuje proces treniranja od prvih 10 epoha EmotionNet kovolutivne mreže.

```
Training epoch: 1...
Total loss: 172.24362111091614
Accuracy: 0.41014210086375036
Training epoch: 2...
Total loss: 157.8295851945877
Accuracy: 0.4575090554471998
Training epoch: 3...
Total loss: 150.235227227211
Accuracy: 0.4831429367511842
Training epoch: 4...
Total loss: 144.64310282468796
Accuracy: 0.502089718584564
Training epoch: 5...
Total loss: 141.10841244459152
Accuracy: 0.5182502089718585
Training epoch: 6...
Total loss: 139.3230301141739
Accuracy: 0.5249373084424631
Training epoch: 7...
Total loss: 139.68401992321014
Accuracy: 0.523822791864029
Training epoch: 8...
Total loss: 134.6783720254898
Accuracy: 0.539704653106715
Training epoch: 9...
...
Accuracy: 0.5511284480356645
```

Vidimo da je u prvih 10 epoha EmotionNet model bolji za skoro 5%. Takođe vidimo da se u prvih 10 epoha poboljšao za skoro 15%.

Na slici ispod možemo posmatrati konvergenciju vrednosti za gubitak, kao i za tačnost modela.



Vidimo da je kriva i greške, i tačnosti glatkija nego što je to slučaj kod jednostavnijeg modela konvolutivne mreže. Iako je u prvoj iteraciji jednostavniji model bio bolji, EmotionNet model se pokazao kao bolji, odnostno uspeo je da napravi veći pomak u tačnosti za istih 10 epoha, što je i očekivano obzirom da ima složeniju strukturu koja mu dozvoljava veći stepen prilagođavanja prilikom treninga.

Naknadnim ponovnim puštanjem treninga nad ovim modelom se postigla čak i najbolja tačnost od sva 4 koje smo implementirali. Model je dostigao tačnost oko 62-63%, za oko 30-35 epoha treniranja. Moguće su neke ispravke modela, kao i načina treniranja koje bi potencijalno dovele do poboljšanja modela, a o njima će biti reči u zaključku.

EDA_CNN

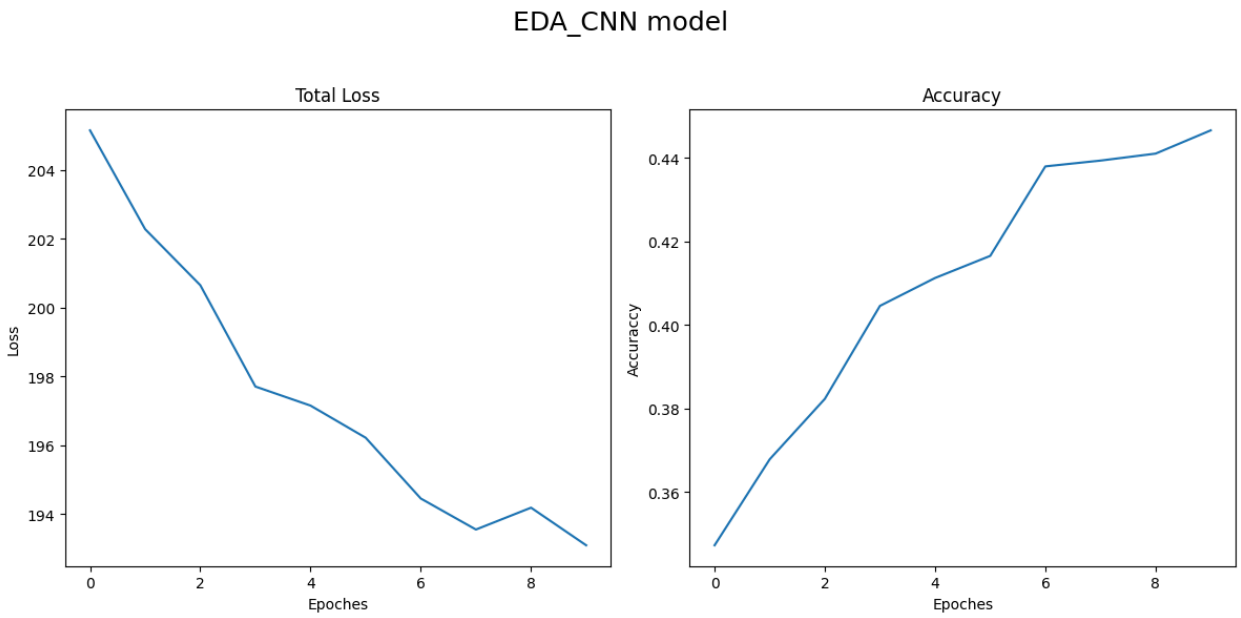
EDA_CNN je poslednja duboka konvolutivna mreža koju smo testirali, i nju smo uzeli kao model koji je već postojao kao jedno rešenje ovog problema analize facijalnih ekspresija i prepoznavanja emocija na licu. Model je po složnosti strukture između našeg jednostavnijeg konvolutivnog modela, i našeg kompleksnijeg EmotionNet modela. EDA_CNN model ima 428.935 trenirabilnih parametara, i kao takav je za 10 epoha treninga zahtevao oko 20 minuta, nešto duže od **SimpleConvNN** modela.

Na slici ispod se može videti proces treniranja od 10 epoha, ukupan gubitak kao i tačnost u svakoj od prvih 10 epoha.

```
Total loss: 205.1426694393158
Accuracy: 0.3471719141822235
Training epoch: 2...
Total loss: 202.27469968795776
Accuracy: 0.3677904708832544
Training epoch: 3...
Total loss: 200.64971089363098
Accuracy: 0.38227918640289776
Training epoch: 4...
Total loss: 197.70747900009155
Accuracy: 0.4045695179715798
Training epoch: 5...
Total loss: 197.15452790260315
Accuracy: 0.41125661744218445
Training epoch: 6...
Total loss: 196.22252714633942
Accuracy: 0.4165505711897464
Training epoch: 7...
Total loss: 194.4582952260971
Accuracy: 0.43800501532460295
Training epoch: 8...
Total loss: 193.55785644054413
Accuracy: 0.43939816104764556
Training epoch: 9...
Total loss: 194.19205260276794
...
Accuracy: 0.44664251880746725
```

EDA_CNN model je u početku bio lošiji od jednostavnijeg modela, čak i nakon 10 epoha treninga je imao skoro 10% manju tačnost od **SimpleConvNN** modela, što je bilo u početku čudno jer je stabilnije konvergirao.

Slika ispod prikazuje konvergencije ovih vrednosti u prvih 10 epoha treniranja **EDA_CNN** modela.



Ono što je ključna stvar je da se na ovom grafiku u prvih 10 epoha ne vidi skoro uopšte da se dostigla konvergencija modela. Upravo to je razlog zašto je ovaj model pokazao lošije rezultate u prvih 10 epoha od jednostavnijeg konvolutivnog modela **SimpleConvNN.** Model nije počeo da konvergira, odnosno sporije konvergira. Dovoljno je bilo da pustimo opet dodatnih 10 epoha gde se videlo da je **EDA_CNN** model dostigao veću tačnost od **SimpleConvNN** modela. Model je dostigao oko 60% tačnosti, više nego **SimpleConvNN.** Čak i tad je kriva konvergencije tačnosti bila strmija čak i od **EmotionNet** modela koji ima bolju tačnost, ali postoji šansa da bi **EDA_CNN** model kada bi se pustio da se trenira dodatnih recimo 100-200 iteracija, prestigao **EmotionNet** model.

Ono što je sigurno bez obzira na model je da se proces treniranja može unaprediti. Na primer pomenuli smo podešavanje koraka učenja (**learning rate**), i primene **CLR-a.** Takođe je moguće i korišćenje druge funkcije greške koja bi mogla da dovede do brže oknvergencije modela. Naravno, treniranje modela na specijalizovanom hardveru sa grafičkim karticama, možda ne bi doveo do poboljšanja modela, ali svakako do bržeg treniranja i evaluacije modela.

Demonstracija upotrebe

Kada smo isprojektovali, istrenirali,koliko je to bilo moguće ograničenim resursima pa samim vremenom, i evaluirali modele, napravili smo demonstraciju neke realne primene ovoga.

Napravili smo demo program gde korisnik prilikom pokretanja programa, pokreće proces koji otvara kameru i daje nekoliko sekundi korisniku da se namesti i uslika svoje lice. Poželjno je da se korisnik namesti tako da mu je lice skroz vidljivo i u centru kadra, ali to nije obavezno jer su modeli trenirani na najrazličitijim slikama, kadrovima, pa čak i preprekama koje zaklanjaju lice. Svakako da se očekuje da model najbolje prepozna kada mu je lice u krupnom planu, i naravno da je kvalitet same slike što je moguće bolji.

Nakon slikanja korisnikovog lica, slika se transformiše tako da može da bude prosleđena nekom od modela. To znači da se pre svega mora transformisati u format 48x48 piksela, a nakon toga, moramo osigurati da slika ima samo jedan kanal, odnosno da se tranformiše u greyscale format. Ovako transformisana slika je sada spremna da se pretvori u **tensor** i da se koristi u kao ulaz u model.

Mi smo u demonstraciji koristili našu **EmotionNet** model, jer je dao najbolje rezultate prilikom treninga. Pošto smo modele prethodno sačuvali kao težine i parametre, veoma lako učitavamo željeni model.

Sve što sada preostaje je da učitanom modelu kažemo da želimo da evaluira ulaz, i da transformisanu sliku, sada već **tensor**, prosledimo modelu. Izlaz iz modela su zapravo težine svake od mogućih izlaznih klasa, a filtriranjem onog koji ima maksimalnu vrednost, zapravo dobijamo šta je naš najverovatnije da je model predvideo kao facijalnu ekspresiju koju je pronašao na našem licu.

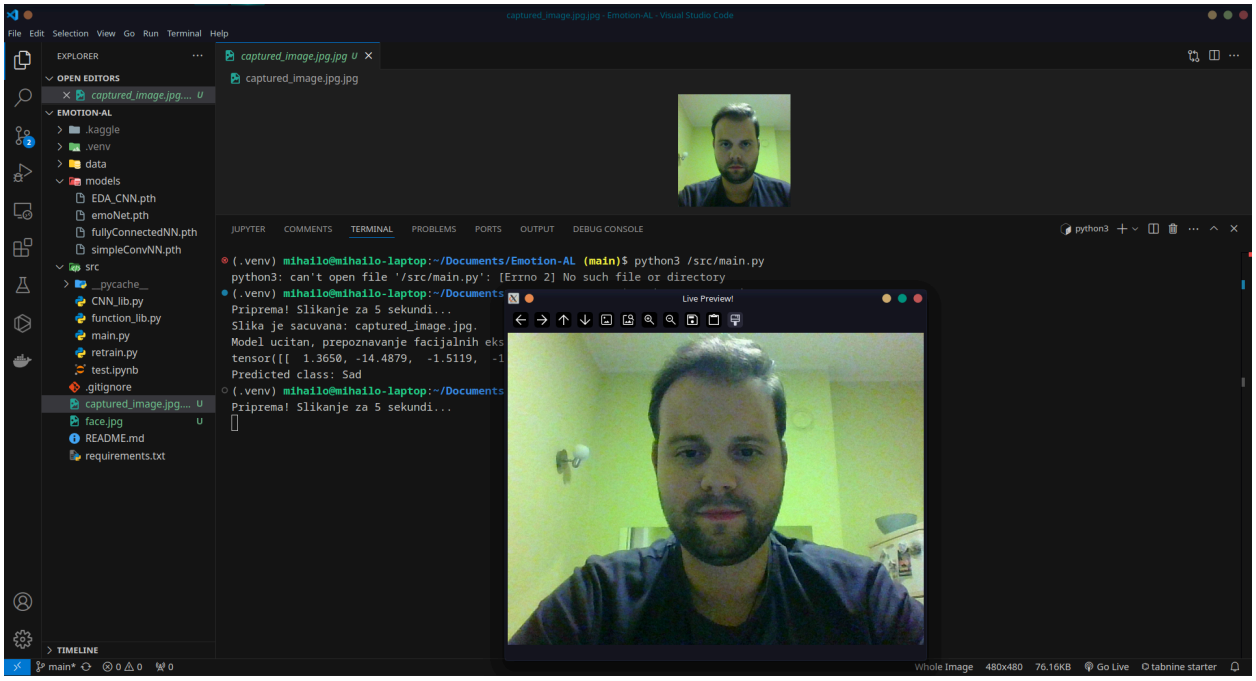
Ono što smo primetili je da model uglavnom može da prepozna emocije: “sreća”, “tuga”, “neutralnost”, dok emocije kao što su “gađenje” i “bes”, može potencijalno da pogrešno klasifikuje iz dva razloga:

1. Gađenje i bes uglavnom imaju veoma izražene promene na licu, pa je nekad teško uočiti da li se osoba “mršti iz besa”, ili izražava “gađenje” pa joj se pojavljuju bore oko čela.
2. Kada smo pogledali slike koje su korišćene za treniranje modela, primetili smo da su na primer, neke facijalne ekspresije koje bi i mi kao ljudi ponekad rekli da su tužne, zapravo klasifikovane kao “neutralne”, ili nešto što nama izgleda kao “bes”, u skupu podataka klasifikovano kao “strah” i slično

Imajući ovo u vidu, zaista i prilikom našeg testiranja gde smo pravili facijalne ekspresije gde smo želeli da budemo “tužni”, nas je model klasifikovao kao da smo “neutralni”.

Ovo svakako ne znači da je naš model savršen, ali se mogu uvideti mesta gde je očiglednije da model greši, a gde je to manje verovatno da će da se desi. Na primer, ako se odaljimo od kamere, i naša faca bude mala, ima drastično veće šanse da se pogrešno klasifikuje nego ako se približimo tako da nam je jedan deo glave odsečen. To je zato što uglavnom instance nad kojima se trenira su upravo ovakve, bliže kameri i faca zauzima najveći deo slike.

Na slici ispod se može videti demonstracija, otvorena kamera i slikanje.



Zaključak

U ovom projektu uspešno smo implementirali i istražili različite arhitekture neuronskih mreža za prepoznavanje izraza lica. Korišćenjem unapred definisanih skupova podataka, svaka mreža je individualno trenirana i evaluirana, pri čemu smo koristili funkciju gubitka **CrossEntropyLoss** i **Adam** optimizer. Tokom procesa treniranja, pažljivo smo pratili vrednosti parametra učenja, a prilagođavanjem ovog parametra kada je konvergencija postala spora, postigli smo stabilnije rezultate.

Pored toga, iako nismo imali vremena da testiramo, dodali smo mogućnosti dinamičkog prilagođavanja stope učenja koristeći schedulers kao što su **Cyclic Learning Rate (CLR)** i **StepLR**. Ovaj dodatak bi omogućio mrežama da bolje reaguju na promene u učenju tokom treniranja, čime bi se potencijalno povećala efikasnost modela. Na kraju, kao demonstraciju, razvili smo prilagođeni program koji koristi kameru za hvatanje slike korisnika u realnom vremenu i njenu obradu uz pomoć treniranih modela. Ovaj projekat pokazuje praktične primene dubokih neuronskih mreža u prepoznavanju slika i izraza lica.

Ovo je samo mali deo mogućnosti koje duboke neuronske mreže, imaju, što se svakodnevno može videti u svetu, jer mogućnosti veštačke inteligencije rastu, kao i njena popularnost, a samimm tim i njene primene. Oblast prepoznavanja lica, emocija, autorizacija i autentikacija putem skeniranja lica, otisaka prsta, skeniranja rožnjače, sve više je zastupljena i zalazi u sve sfere, od bezbednosti, preko implementacija u mobilnim uređajima pa sve do primena u medicini, prepoznavanju bolesti za skenera, slika, rendgena i slično, primene su nebrojene.