

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1115

**DUBOKE KONVOLUCIJSKE
NEURONSKE MREŽE ZA
RASPOZNAVANJE ZNAKOVA**

Matija Ilijaš

Zagreb, lipanj 2015.

Zagreb, 6. ožujka 2015.

DIPLOMSKI ZADATAK br. 1115

Pristupnik: **Matija Ilijaš (0036449582)**
Studij: Računarstvo
Profil: Računarska znanost

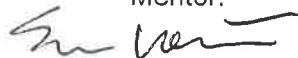
Zadatak: **Duboke konvolucijske neuronske mreže za raspoznavanje znakova**

Opis zadatka:

Optičko raspoznavanje znakova ima brojne primjene u raznim područjima ljudske djelatnosti. U okviru diplomskog rada potrebno je proučiti metode za obradu slika dokumenata, detekciju teksta i raspoznavanje znakova te istražiti primjenjivost dubokih konvolucijskih neuronskih mreža za problem klasifikacije tiskanih znakova uključujući slova, brojeke i posebne znakove. Potrebno je iskoristiti postojeće programske okvire za treniranje mreža (torch, caffe), implementirati algoritam evaluacije mreže u jeziku C++, te usporediti performanse dobivenih modela sa drugim metodama klasifikacije kao što su višeslojne neuronske mreže i metoda najbližeg susjeda. Eksperimentalnu validaciju razvijenih metoda potrebno je provesti na dva skupa podataka: umjetno generiranom skupu, te bazi slika dobivenih aplikacijom PhotoMath. Za detaljne instrukcije obratiti se mentoru.

Zadatak uručen pristupniku: 13. ožujka 2015.
Rok za predaju rada: 30. lipnja 2015.

Mentor:



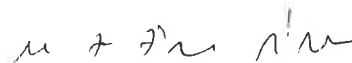
Prof. dr. sc. Sven Lončarić

Djelovođa:



Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
diplomski rad profila:



Prof. dr. sc. Siniša Srbljić

SADRŽAJ

1. Uvod	1
2. Optičko prepoznavanje znakova	3
2.1. Predprocesiranje slike znaka	3
2.1.1. Binariziranje slike	3
2.1.2. Morfološke operacije	5
2.2. Klasifikacija slike znaka	6
3. Umjetne neuronske mreže	7
3.1. Arhitektura neuronske mreže	7
3.2. Učenje neuronskih mreža	8
4. Duboko učenje	10
4.1. Hijerarhijski prikaz podataka	10
4.2. Učenje dubokih mreža	12
5. Duboke konvolucijske neuronske mreže	15
5.1. Lokalni prostorni filteri	16
5.2. Dijeljenje težina	17
5.3. Lokalno udruživanje	18
5.4. Ispravljena linearna jedinica	19
5.5. Konačni model	20
6. Implementacija	21
6.1. Sustav za učenje	21
6.1.1. Torch razvojni alat	22
6.1.2. Definiranje arhitekture	22
6.2. Sustav za testiranje	24
6.2.1. Matrica greške klasifikacije	24

6.2.2. Prosječno trajanje klasifikacije	25
6.2.3. Simultano analiziranje više modela	25
6.3. Prilagodba Microblink OCR sustavu	26
7. Rezultati	27
7.1. Utjecaj predprocesiranja podataka	27
7.2. Pronalaženje optimalne arhitekture mreže	29
7.2.1. Standarne neuronske mreže	29
7.2.2. Konvolucijske neuronske mreže	31
7.3. Usporedba optimalnih arhitektura	32
7.4. Usporedba sa Microblink klasifikatorom	34
7.5. Moguća poboljšanja	35
8. Zaključak	37
Literatura	39

1. Uvod

Optičko prepoznavanje znakova (engl. *Optical Character Recognition – OCR*) područje je istraživanja u računalnom vidu koje se bavi segmentacijom i klasifikacijom znakova sa slike. Prepoznavanje teksta sa slike ima vrlo široku primjenu u mnogim područjima ljudske djelatnosti. Jedna od trenutno najvećih primjena je arhiviranje i pretvorba dokumenata iz papirnatog oblika u digitalni oblik kako bi se mogli jednostavnije i brže pohranjivati i pretraživati. Razvijanjem mobilne tehnologije i njene primjene u svakodnevnom životu pojavljuju se nove izazovnije primjene optičkog prepoznavanja znakova koje zahtijevaju prepoznavanje teksta u prirodnim scenama. Tako se danas OCR algoritmi koriste za prepoznavanje podataka sa osobnih dokumenata i računa, prepoznavanje teksta za simultano prevođenje teksta, rješavanje matematičkih jednadžbi i mnoge druge primjene.

U strojnom učenju se u zadnjih nekoliko godina pojavilo novo obećavajuće područje istraživanja pod nazivom duboko učenje. Koncept dubokog učenja zasniva se na učenju korisnih prezentacija podataka povezanih u obliku hijerarhije, te je djelomično inspiriran vizualnim korteksom mozga sisavca. Zbog velike kompleksnosti modela dubokog učenja, te velike količine podataka potrebnih za učenje, istraživanja su u ovom području postala tek nedavno moguća uvođenjem specijaliziranih alata za brzo učenje na grafičkim karticama. Algoritmi dubokog učenja pokazali su jako dobre rezultate na problemima raspoznavanja uzoraka, posebice u području računalnog vida. Za problem klasifikacije na temelju slika razvijene su duboke konvolucijske neuronske mreže, također inspirirane vizualnim korteksom, koje su ostvarile najbolje rezultate na gotovo svim javno dostupnim skupovima za testiranje. Dodatni dokaz da duboko učenje u nekoj mjeri oponaša vizualni korteks su i nedavna istraživanja gdje su se značajke slike naučene dubokim učenjem pokazale sličnim onima u korteksu.

U ovom radu provedeno je istraživanje primjene dubokih konvolucijskih neuronskih mreža na problemu raspoznavanja znakova. U tu svrhu razvijen je sustav za učenje i testiranje različitih arhitektura standarnih neuronskih mreža

te dubokih konvolucijskih neuronskih mreža. Podsustav za učenje implementiran je pomoću razvojnog alata za paralelno računanje na grafičkoj kartici kako bi postupak bio što brži. Dio sustava za testiranje razvijen je s ciljem detaljne analize rada naučenih modela, te jednostavne i pregledne usporedbe više različitih modela.

Istraživanje je ostvareno uz dodatnu podršku tvrtke Microblink specijalizirane za razvoj tehnologija računalnog vida za mobilne uređaje. Za učenje i testiranje modela koriste se od tvrtke interni skupovi podataka, a prilikom testiranja provodi se analiza i njihovog internog klasifikatora. Rezultati su prikazani i komentirani sa ciljem pronalaska optimalne arhitekture modela za primjenu na mobilnom uređaju. Pritom je osim točnosti naučenih klasifikatora uzeta u obzir i njihova kompleksnost te prosječno trajanje klasifikacije.

2. Optičko prepoznavanje znakova

Proces optičkog prepoznavanja znakova provodi se u dva glavna koraka, segmentacije znakova iz slike i njihovog klasificiranja. U ovom radu provodi se istraživanje metoda klasifikacije, te se koristi već segmentirani skup znakova za učenje i testiranje klasifikatora. Stoga je u nastavku ovog poglavlja opisan drugi korak optičkog prepoznavanja znakova, njihovo klasificiranje.

Kao i kod svakog problema raspoznavanje uzoraka u računalnom vidu, transformacije ulaznih slika znakova nastale promjenjivim uvjetima okoline znatno otežavaju klasifikaciju. Promjenjiva pozadina znakova, moguća oštećenja i okluzije, te različita osvjetljenja prilikom slikanja mogu znatno promijeniti ulaznu sliku. Geometrijske transformacije posljedica su i različitih parametara kamere s kojom se slika, kao i pozicije kamere u odnosu na sliku prilikom slikanja. Kako bi se smanjio utjecaj transformacija ulazne slike, uobičajeni je postupak korištenje neke metode predprocesiranja prije slanja slike klasifikatoru.

2.1. Predprocesiranje slike znaka

Cilj predprocesiranja je proslijediti klasifikatoru sliku sa jasno definiranim područjem koje predstavlja znak. Često korišten postupak za dobivanje takve slike je metoda binarizacije, odnosno ograničavanje vrijednosti svih piksela slike na vrijednosti 0 ili 1. Tako dobivene binarne slike često imaju šum koji može otežati klasifikaciju, stoga se nakon binarizacije slika dodatno obrađuje morfološkim operacijama.

2.1.1. Binariziranje slike

Proces binarizacije uspoređuje svaku vrijednost piksela ulazne slike sa definiranim pragom, te ukoliko je njegova vrijednost veća dodjeljuje pikselu vrijednost 1, a u suprotnom vrijednost 0 (Izraz 2.1). Postoji više metoda definiranja praga

za pojedinačni piksel. Najjednostavnija metoda binarizacije koristi ručno određeni prag fiksne vrijednosti jednak za sve piksele. Ovako definiran prag je teško odrediti zbog promjenjivog intenziteta piksela pozadine i znaka, stoga naprednije metode računaju prag na temelju ulazne slike.

$$y(i, j) = \begin{cases} 1 & \text{ako } x(i, j) > \text{prag} \\ 0 & \text{inače} \end{cases} \quad (2.1)$$

Jednostavniji pristup računanja praga na temelju slike koristi se svim vrijednostima piksela, te provodi binarizaciju piksela sa istim globalnim pragom. Ovako određen prag daje bolje rezultate od ručno određenog praga zbog bolje prilagođenosti pojedinačnoj slici, no zbog svoje fiksne vrijednosti za sve piksele nije robusan na lokalne promjene u kontrastu slike.

Kako bi binarizacija bila robusna na promjene u osvjetljenju potrebno je određivati prag pojedinačno za svaki piksel u slici. Takav prag se zove adaptivni, te se računa na temelju lokalnog susjedstva piksela. Sama operacija računanja praga može biti jednostavni prosjek vrijednosti svih piksela u susjedstvu ili težinski prosjek koristeći na primjer 2D Gaussovu funkciju sa centrom na lokaciji piksela (i, j) :

$$w_{gauss}(i, j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\left(\frac{(i - i_0)^2}{2\sigma^2} + \frac{(j - j_0)^2}{2\sigma^2}\right)\right) \quad (2.2)$$

Usporedba binariziranih slika globalnim i adaptivnim pragom može se vidjeti na Slici 2.1. Na područjima gdje razlika između piksela pozadine i znaka postaje vrlo mala, binarizacija globalnim pragom dala je lošije rezultate što je rezultiralo uklanjanjem ključne razlike između znaka 'G' i '6'.



Slika 2.1: Usporedba binarizacije sa globalnim i adaptivnim pragom

2.1.2. Morfološke operacije

Digitalnom morfologijom analiziraju se oblici digitalnih objekata. Slika se promatra u kontekstu teorije skupova te se predstavlja kao skup binarnih elemenata koji su grupirani u određene dvodimenzionalne strukture. Za obrađivanje binarnih slika tipično se koriste morfološke operacije dilatacije i erozije, te njihovih kombinacija, operacija zatvaranja i otvaranja.

Dilatacija prolazi sa strukturnim elementom po slici i traži maksimalnu vrijednost piksela zahvaćenog dijela slike. Pronađenu maksimalnu vrijednost postavlja kao vrijednost svojeg središnjeg piksela, te na taj način proširuje svijetle regije piksela na slici. Erozija, s druge strane, traži minimalnu vrijednost piksela te vrši isti postupak i proširuje tamne regije piksela. Binarna slika sadrži izdvojen znak u crnoj boji na bijeloj podlozi, odnosno predstavljena je tamnom regijom piksela okruženom svijetlim pikselima. Stoga su efekti ovih operatora obrnuti za potrebe obrade slika znakova.

S obzirom da je u praksi teško znati kada je potrebna dilatacija a kada erozija, češće se koriste njihove kombinacije, otvaranje i zatvaranje.

$$\begin{aligned}y(i, j) &= \text{otvaranje}(x(i, j)) = \text{dilatacija}(\text{erozija}(x(i, j))) \\y(i, j) &= \text{zatvaranje}(x(i, j)) = \text{erozija}(\text{dilatacija}(x(i, j)))\end{aligned}\tag{2.3}$$

Operacija otvaranja primjenjuje prvo eroziju pa dilataciju na slici, čime uklanja mala oštećenja na znakovima. Zatvaranje provodi prvo dilataciju slike a zatim eroziju, te time uklanja nepotrebni šum na slici.



Slika 2.2: Primjer uklanjanja šuma operacijom zatvaranja

2.2. Klasifikacija slike znaka

Kako bi odredili ASCII vrijednost znaka, potrebno je sliku znaka provesti kroz postupak klasifikacije. Standardni postupak razvijanja rješenja klasifikacije sastoji se od dva dijela, razvijanja kvalitetnih značajki koje olakšavaju klasifikaciju te razvijanja klasifikatora.

Kvalitetne značajke koje olakšavaju klasifikaciju moraju zadovoljiti više uvjeta. Značajka prije svega treba biti diskriminantna, odnosno njena vrijednost mora biti različita za slike različitih znakova. Drugi uvjet je da vrijednost značajke ostaje ista za različite primjere slika istog znaka, odnosno da je robusna na moguće varijacije unutar jedne klase. Varijacije nastaju uslijed promjenjivih uvjeta okoline, a manifestiraju se najčešće kao geometrijske transformacije i lokalne promjene kontrasta. Stoga značajke trebaju biti invarijantne na promjene osvijetljenja i pomake kao što su translacija i rotacija. Značajke se mogu odrediti ručno na temelju promatranja podataka ili izravno iz skupa podataka metodama koje su opisane u kasnijim poglavljima ovog rada.

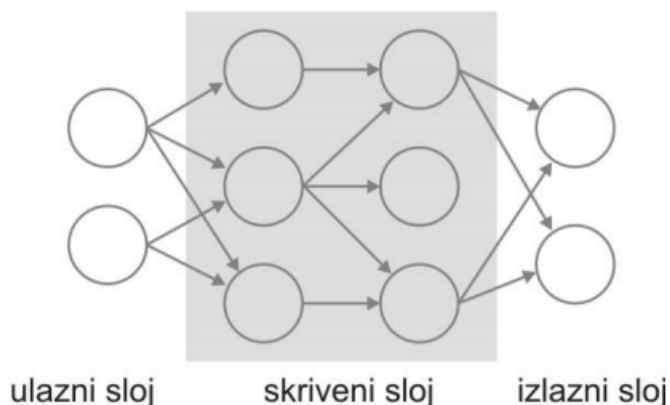
Općeniti problem učenja klasifikatora veliko je područje istraživanja, te će se opisati u kontekstu neuronskih mreža u kasnijim poglavljima. Važno je spomenuti svojstvo generalizacije kao glavno mjerilo dobre klasifikacije koje definira sposobnost klasifikatora da točno klasificira dotad neviđene primjere. Veliku ulogu u konačnim rezultatima klasifikatora imaju prethodno navedene metode određivanja značajki podataka, kao i sam skup podataka na kojima se klasifikator uči.

3. Umjetne neuronske mreže

Umjetna neuronska mreža (engl. *Artificial Neural Network* - *ANN*) algoritam je strojnog učenja inspiriran strukturom i funkcionalnošću ljudskog mozga. Zasniva se na paralelnoj obradi podataka, te vrlo dobro rješava probleme kod kojih postoji složena nelinearna veza ulaza i izlaza. Radi s velikim brojem parametara i varijabli, te može raditi s nejasnim podacima što ju čini robusnom na pogreške. Jedna od najčešćih primjena neuronskih mreža je u području raspoznavanja uzoraka, te se vrlo često koristi upravo za raspoznavanje znakova.

3.1. Arhitektura neuronske mreže

Neuronska mreža se sastoji od tri vrste sloja: ulaznog, izlaznog, te skrivenog koji povezuje prethodna dva. Čvorišta, tzv. neuroni, skrivenog sloja povezani su težinskim vezama s ulaznim i izlaznim neuronima. Standardna arhitektura neuronske mreže je potpuno povezana i aciklička što znači da su svi neuroni iz susjednih slojeva međusobno povezani te da nema ciklusa u mreži između neurona.



Slika 3.1: Primjer jednostavne arhitekture neuronske mreže

Umjetni neuron je matematička funkcija inspirirana funkcionalnošću biološkog

neurona koja radi težinsku sumu svojih ulaza a zatim primjenjuje svoju prijenosnu funkciju na izlazu [14]. Ako ulaze neurona definiramo sa vektorom x duljine m i njegove težine jednim redom matrice težina w , tada se izlaz neurona k može definirati izrazom:

$$y_k = f \left(\sum_{j=0}^m w_{kj} x_j \right) \quad (3.1)$$

gdje je f prijenosna funkcija neurona. Kako bi mreža mogla predstaviti visoko nelinearni odnos između njenog ulaza i izlaza, prijenosna funkcija njenih neurona mora biti nelinearna funkcija. Najčešće korištena prijenosna funkcija je sigmoidalna funkcija koja se koristi zbog svojeg blažeg prijenosa od obične funkcije s fiksnim pragom i svoje derivabilnosti, te je poseban slučaj logističke funkcije definirane izrazom:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

3.2. Učenje neuronskih mreža

Dvije su faze rada umjetnih neuronskih mreža, učenje i eksploatacija. Učenje je iterativan postupak predočavanja ulaznih primjera i očekivanog izlaza pri čemu dolazi do postupnog prilagođavanja težina veza neurona. Nakon što se težine skrivenog sloja prilagode postupkom učenja, eksploatacijom neuronske mreže može se za do tad neviđeni ulazni primjer dobiti pripadajući izlaz.

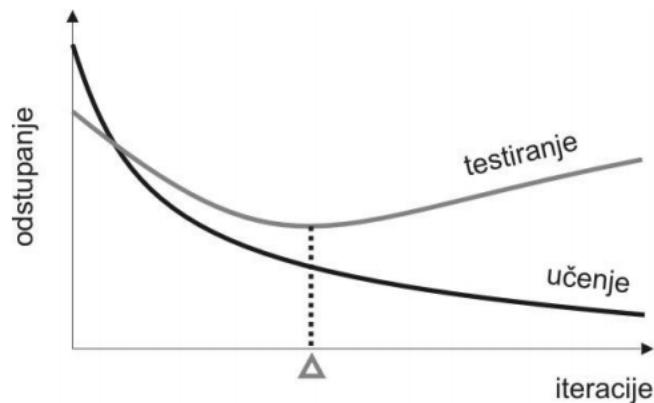
Jedna neuronska mreža sa svojim težinama može se gledati kao nelinearna funkcija sa više varijabli, te se njeno učenje može opisati kao traženje minimuma te funkcije s obzirom na grešku klasifikacije. Ako se uzme u obzir da izlazni sloj može sadržavati više neurona, pogreška se može definirati izrazom:

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{izlazni}} (t_{kd} - o_{kd})^2 \quad (3.3)$$

pri čemu je t_{kd} tražena izlazna vrijednost i o_{kd} izlazna vrijednost dobivena mrežom za k -ti neuron na primjeru podatka za učenje d [19]. Metoda minimiziranja ove pogreške zove se gradijentni spust (engl. *Gradient descent*), a algoritam

koji implementira tu metodu za učenje neuronske mreže zove se algoritam propagacije greške unatrag (engl. *Backpropagation algorithm*) [15]. Radi gradijentne metode prijenosna funkcija neurona mora biti derivabilna kako bi se mogli računati gradijenti, što vrijedi za ranije spomenutu sigmoidalnu funkciju.

Glavna mjera uspješnosti neuronske mreže u klasifikaciji je njena sposobnost generalizacije, odnosno koliko dobro može klasificirati primjere koji nisu korišteni prilikom njenog učenja. Stoga se za potrebe učenja neuronskih mreža podatci dijele na dva (a nekad i tri) skupa: skup za učenje, skup za testiranje i ponekad skup za dodatnu evaluaciju. Kada je greška klasifikacije velika na testnom skupu a mala na skupu za učenje kažemo da je mreža pretjerano prilagođena skupu za učenje (engl. *Overfitting*). Kako bi se dobila mreža sa najboljom sposobnošću generalizacije, postupak učenja se zaustavlja kada se greška mreže na testnom skupu prestane smanjivati. Pojednostavljeni prikaz jednog takvog postupka može se vidjeti na Slici 3.2.



Slika 3.2: Prikaz greške klasifikacije prilikom učenja mreže

4. Duboko učenje

Duboko učenje se u zadnjih nekoliko godina pojavilo kao novo obećavajuće područje strojnog učenja. Iako je sama ideja prezentirana u radovima Kunihika Fukushima [6] i Yanna Lecuna [12] prije gotovo 30 godina, duboko učenje je postalo predmetom istraživanja relativno nedavno. Razlog tome su bila tada nedovoljno snažna računala za učenje velikih modela karakterističnih za taj pristup. Koncept dubokog učenja zasniva se na učenju korisnih prezentacija podataka povezanih u obliku hijerarhije. Ideja je djelomično inspirirana vizualnim korteksom mozga sisavaca koji se sastoji od niza procesirajućih elemenata koji obrađuju vizualni podražaj. Dokaz da duboko učenje u nekoj mjeri oponaša vizualni korteks su i nedavna istraživanja gdje su se naučene značajke pokazale slične onima u korteksu. Glavna karakteristika algoritama dubokog učenja je velik broj slojeva kroz koje ulazni podatak mora proći do izlaza, gdje se prvi dio prolaza može zamisliti kao formiranje hijerarhijskog prikaza podatka, a drugi kao proces klasifikacije. Time jedan algoritam dubokog učenja ustvari preuzima zadatak izvlačenja korisnih značajki iz podataka uz sami zadatak klasifikacije. Kod učenja takvog algoritma cijeli se postupak uči na temelju podataka, te se time gubi potreba za ručnim definiranjem značajki podataka.

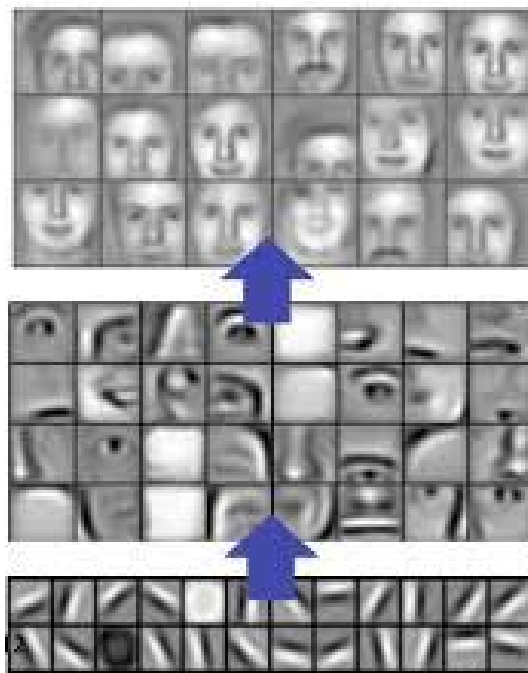
4.1. Hijerarhijski prikaz podataka

Slika se može klasifikatoru prikazati na mnogo načina, od kojih je najjednostavniji vektor vrijednosti intenziteta piksela. Apstraktniji prikaz bio bi recimo skup rubova na slici, a još apstraktniji skup različitih oblika. Neki prikazi olakšavaju klasifikatoru proces učenja, a različiti problemi klasifikacije zahtijevaju razvijanje različitih prikaza odnosno značajki ulaznih podataka. Stoga se tradicionalno razvijanje rješenja problema klasifikacije sastoji od dva dijela, razvijanja kvalitetnih značajki koje olakšavaju učenje te razvijanja klasifikatora.

Uobičajeni proces razvoja kvalitetnih značajki podrazumijeva promatranje

svojstva ulaznih podataka te naglašavanje onih koja su maksimalno diskriminantna. Napredak u točnosti klasifikacije većim dijelom se ostvaruje razvijanjem boljih značajki, tako je na primjer značajni napredak u prepoznavanju čovjeka ostvaren razvijanjem značajki pomoću histograma orijentiranih gradijenata (engl. *Histogram of Oriented Gradients - HoG*) [5]. Histogrami su dobro opisivali lokalne oblike te u isto vrijeme bili u velikoj mjeri invarijantni na promjene osvjetljenja i geometrijske transformacije.

No negativna strana postupka razvijanja značajki koje se baziraju na ljudskom promatranju podataka je gubitak informacija koje čovjek nije u stanju vidjeti. Kako bi se koristile sve informacije iz podataka, potrebne su metode koje određuju diskriminantne značajke izravno iz skupa podataka. Među popularnijim metodama tog tipa su metoda linearne diskriminantne analize (engl. *Linear Discriminant Analysis - LDA*) te metoda glavnih komponenta (engl. *Principal Component Analysis - PCA*). Linearna diskriminantna analiza pronalazi prikaz podataka koji maksimalno diskriminira podatke različitih klasa, dok metoda glavnih komponenta prikazuje podatke kao linearnu kombinaciju linearno nekoreliranih varijabli koje imaju najveću varijancu.



Slika 4.1: Hijerarhija značajki ljudskog lica naučena dubokim učenjem

Istraživanja vizualnog korteksa mozga sisavaca pokazala su da vizualni podražaj prolazi kroz hijerarhiju više procesirajućih slojeva gdje svaki dodatni sloj predstavlja dodatnu razinu apstrakcije značajki. Značajke dobivene prethodno

navedenim metodama predstavljaju jednu razinu apstrakcije, te se mogu zamisliti kao imitacija najnižeg sloja vizualnog korteksa. Ideja dubokog učenja je naučiti cijelu hijerarhiju značajki, odnosno imitirati i više slojeve vizualnog korteksa. Primjer jedne takve hijerarhije značajki ljudskog lica naučene dubokim učenjem prikazan je na Slici 4.1.

4.2. Učenje dubokih mreža

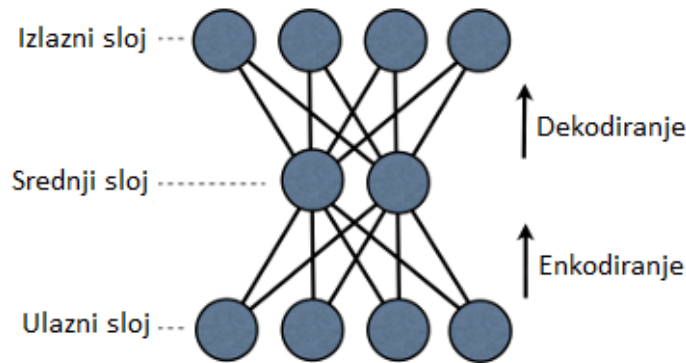
Kao što je ranije spomenuto ideja dubokog učenja je naučiti hijerarhiju značajki iz podataka. S obzirom da to zahtijeva nekoliko dodatnih slojeva mreže, kompleksnost dubokih modela značajno je veća od standardnih. Poznato je da količina potrebnih podataka za kvalitetno učenje modela raste proporcionalno sa brojem njegovih slobodnih parametara. Stoga je za učenje dubokih mreža potrebno značajno više podataka nego kod učenja standardnih klasifikatora. Nedovoljna količina podataka za učenje, te nedovoljno jaka računala za učenje tako velikih modela, razlog su zašto duboko učenje nije zaživjelo 1980-ih godina kada je prvi put predstavljeno kao ideja. Dolazak specijaliziranih razvojnih alata za brzo učenje mreža na grafičkim karticama, kao i sve veća količina dostupnih podataka za učenje, omogućili su da duboko učenje u zadnjih nekoliko godina pokaže svoj potencijal na raznim problemima strojnog učenja.

Jedna od velikih prednosti dubokog učenja je što uz metode nadziranog učenja ima razvijene i metode nenadziranog učenja. U ovom radu koristi se isključivo nadzirano učenje zbog dostupnosti velike količine označenih podataka. U nastavku ovog poglavlja opisana je metoda nenadziranog učenja, a metoda nadziranog učenja opisana je u idućem poglavlju kao uvod u duboke konvolucijske mreže.

Sve veća količina podataka postaje dostupna za učenje, no velikim dijelom radi se o neoznačenim podacima. Kako bi se i ti podatci iskoristili potrebne su metode nenadziranog učenja. Kao što je ranije spomenuto, duboki modeli predstavljaju hijerarhiju značajki podataka, a jednu razinu apstrakcije moguće je postići metodama poput metode glavnih komponenata. Stoga se u teoriji postavlja pitanje da li uzastopnim primjenjivanjem metode glavnih komponenata možemo dobiti traženu hijerarhiju. Odgovor na to pitanje je da ne možemo, a razlog tome je linearnost te metode. Uzastopnom linearnom transformacijom konačno se uvijek dobiva linearna transformacija, što onemogućuje stvaranje željene hijerarhije. Iz tog razloga koriste se dvije metode sa dodanom nelinearnošću, autoenkoderi

(engl. *Autoencoders*) i ograničeni Boltzmann strojevi (engl. *Restricted Boltzmann Machines - RBM*) [1][9]. U sklopu ovog rada opisati će se autoenkoderi, jednostavnija i intuitivnija metoda od dvije navedene.

Autoenkoder je jednostavna troslojna neuronska mreža kod koje su izlazni neuroni izravno povezani sa ulaznim neuronima. Ulazni i izlazni sloj imaju jednak broj neurona, a u srednjem sloju ima ih manje. Kao posljedica toga prolaskom podataka kroz mrežu događaju se dvije radnje, enkodiranje i dekodiranje.



Slika 4.2: Prikaz rada autoenkodera

Enkodiranje se može definirati kao postupak preslikavanja ulaznog vektora x u vektor elemenata skrivenog sloja y izrazom:

$$y = f(Wx + b) \quad (4.1)$$

gdje je f nelinearna prijenosna funkcija (npr. sigmoidalna), a W matrica težina autoenkodera. Zato što je broj neurona srednjeg sloja manji od broja elemenata ulaznog sloja, ulazni vektor se kompresira, odnosno podatak se u srednjem sloju prikazuje pomoću apstraktnijih značajki. Nakon toga vrši se dekompresija, odnosno dekodiranje, kako bi se na temelju značajki iz srednjeg sloja rekonstruirao ulazni vektor. Postupak dekodiranja vektora značajki srednjeg sloja y u vektor rekonstrukcijskog izlaznog sloja z definiran je izrazom:

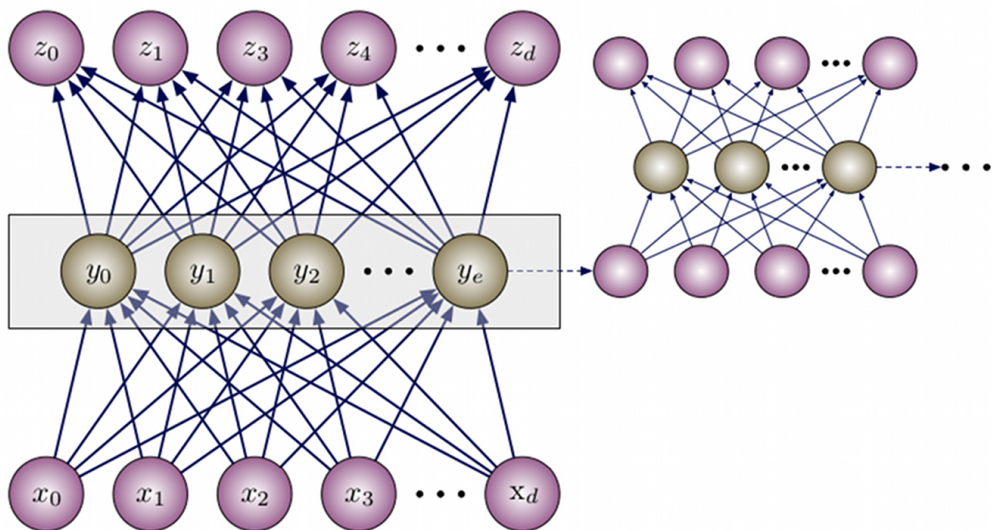
$$z = f(W'y + b') \quad (4.2)$$

gdje se W' može definirati kao W^T , transpozicija matrice težina W . U tom slučaju autoenkoder ima takozvano svojstvo povezanih težina (engl. *Tied weights*), što smanjuje konačan broj parametara koje je potrebno naučiti. Vrijednosti izlaznog vektora zatim se uspoređuju sa onima ulaznog vektora kako bi se dobila informacija o kvaliteti rekonstrukcije, odnosno koliko dobro značajke iz srednjeg

sloja predstavljaju ulazni podatak. Rekonstrukcijska greška može se računati na mnogo načina, od kojih je jedan od jednostavnijih korištenje funkcije kvadratne greške:

$$E(xz) = (x - z)^2 \quad (4.3)$$

Proces učenja autoenkodera sastoji se od iterativnog enkodiranja i dekodiranja ulaznih podataka za učenje te korigiranja njegovih težina na temelju greške u rekonstrukciji. Rezultat je autoenkoder sa težinama koje čine elemente njegovog srednjeg sloja apstraktnim značajkama koje dobro opisuju prezentirane podatke. S obzirom da autoenkoder radi nelinearnu transformaciju nad ulaznim podacima, možemo ostvariti hijerarhiju apstraktnih značajki slijednim povezivanjem više autoenkodera. Učenje se provodi iterativno, odnosno prvi autoenkoder uči se na temelju ulaznih podataka, a svaki idući na temelju naučenih značajki svojeg prethodnika.



Slika 4.3: Prikaz iterativnog učenja slijeda autoenkodera

Ovako dobivena mreža naučena je isključivo na neoznačenim podacima te predstavlja hijerarhiju značajki podataka bez mogućnosti klasifikacije. Konačni klasifikator dobiva se dodavanjem jedne manje mreže koja na ulaz prima značajke zadnjeg autoenkodera a na izlaz vraća klase zadanog problema. Tako povezana mreža autoenkodera i mreža za klasifikaciju zatim se uče na označenim podacima kao jedna velika mreža.

5. Duboke konvolucijske neuronske mreže

U prethodnom poglavlju opisana je metoda nenadziranog učenja koja može naučiti hijerarhiju značajki koristeći se neoznačenim podacima, a zatim naučiti klasificirati pomoću manjeg skupa označenih podataka. Ovaj pristup je praktičan u većini problemskih domena zbog male količine dostupnih označenih podataka za učenje. No u praksi se pokazalo da za određene probleme, gdje ima dovoljno označenih podataka, najbolje rezultate daju metode nadziranog učenja. Problemska domena ovog rada ima dostupnu veliku količinu označenih podataka, te su svi modeli ućeni metodama nadziranog ućenja.

Kao i kod nenadziranog ućenja, konaćni model koji se ući sastoji se od dijela za formiranje hijerarhije znaćajki i dijela za klasifikaciju, no nadzirano ućenje je proces koji se provodi u jednom koraku i na jednom skupu podataka. Sam postupak ućenja ne razlikuje se od standarnog ućenja neuronske mreže metodom gradijentnog spusta.

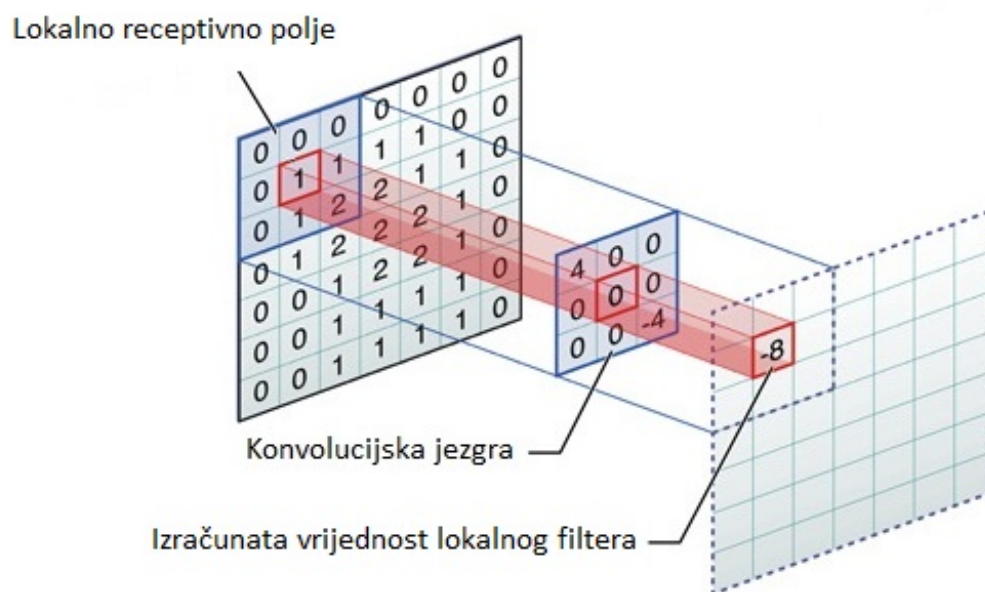
IstraŹivanja vizualnog korteksa pokazala su da njegovi procesirajući elementi reagiraju na male lokalne podraŹaje unutar vizualnog polja, funkcionirajući kao lokalni receptori. Glavna karakteristika prirodnih slika je njihova lokalna prostorna korelacija, pa je logićno da se vizualni korteks prilagodio traŹenju lokalnih znaćajki. Klasićne neuronske mreŹe sa potpuno povezanim slojevima ne iskorišćavaju ovo korisno svojstvo, Źto je potaknulo Yanna Lecuna 1989. godine da razvije model konvolucijske neuronske mreŹe [13]. Razvijena mreŹa oponaŹa lokalne receptore vizualnog korteksa sa svojim lokalnim prostornim filterima ostvarenim konvolucijom (Izraz 5.1).

$$y(m, n) = x(m, n) * h(m, n) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x(i, j)h(m - i, n - j) \quad (5.1)$$

5.1. Lokalni prostorni filteri

Konvolucijom ostvareni filteri omogućuju provođenje kompleksnih operacija jednostavnom promjenom konvolucijske jezgre. Tako bi na primjer upotrebom Gaussovne jezgre dobili za rezultat zamućenje slike, a upotrebom Sobelove jezgre [16] naglašene rubove na slici. Ručno definiranje konvolucijskih jezgri upravo odgovara ranije spomenutom ručnom određivanju značajki u standardnom postupku razvijanja klasifikatora. Cilj konvolucijske neuronske mreže je naučiti konvolucijske jezgre specifične za dane podatke, što odgovara ranije spomenutom učenju značajki iz podataka.

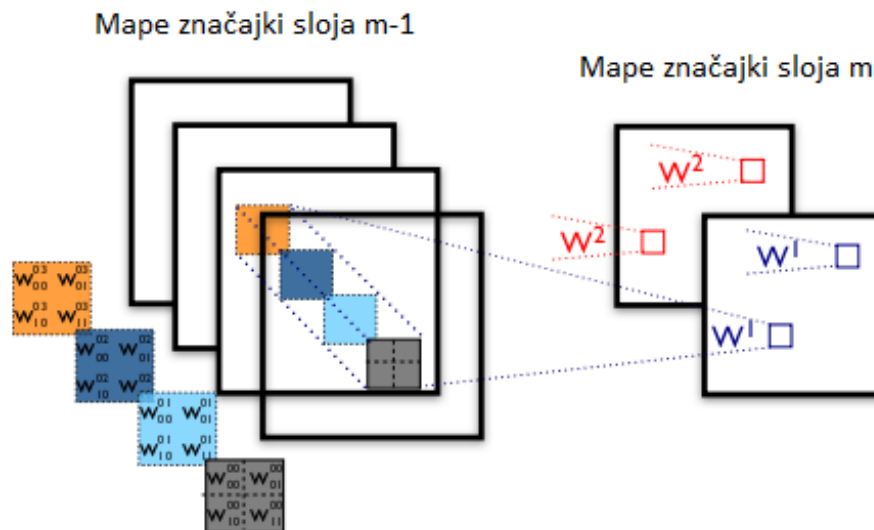
U dubokoj konvolucijskoj mreži prvih nekoliko slojeva zaduženih za izgradnju hijerarhije značajki je konvolucijsko. Uloga prvog sloja konvolucije je jasna s obzirom da odgovara standardnom filtriranju slike sa nekom konvolucijskom jezgrom. Viši slojevi konvolucije nemaju doticaja sa ulaznom slikom, već formiraju apstraktnije lokalne značajke od značajki prethodnog sloja. Na primjer u problemu detekcije čovjeka prvi sloj konvolucije razvio bi lokalne detektore rubova na slici, sloj iznad njega formirao bi lokalne oblike koristeći se sa rubovima iz prvog sloja, a sloj za razinu više bi koristio lokalne oblike za formiranje detektora dijelova tijela čovjeka. Jednom kada mreža ima detektore značajki koje opisuju prisutnost glavnih dijelova objekta, u ovom slučaju čovjeka, problem klasifikacije postaje znatno jednostavniji.



Slika 5.1: Primjer računanja vrijednosti lokalnog prostornog filtera

5.2. Dijeljenje težina

Jednom naučeni lokalni filteri mogu prepoznati neku značajku podataka, no nepoznato je gdje se ta značajka nalazi u prostoru. Kako bi razvili lokalne filtere invarijantne na poziciju u prostoru, potrebno je za svaki filter imati skup njegovih inačica koje pokrivaju sve moguće podprostore. Takav skup zovemo mapa značajki (engl. *Feature maps*), a glavna karakteristika takve mape je dijeljenje težina između svih elemenata skupa. Svaki element konvolucijskog sloja prikazuje se tada kao matrica koja sadrži vrijednosti odziva pripadajućeg filtera za svaku poziciju u prostoru. Dijeljenje težina na višoj razini apstrakcije prikazano je na Slici 5.2, gdje je važno primjetiti da konvolucija nakon prvog sloja postaje trodimenzionalna zbog uvođenja mape značajki koja je dvodimenzionalni prikaz elementa konvolucijskog sloja.



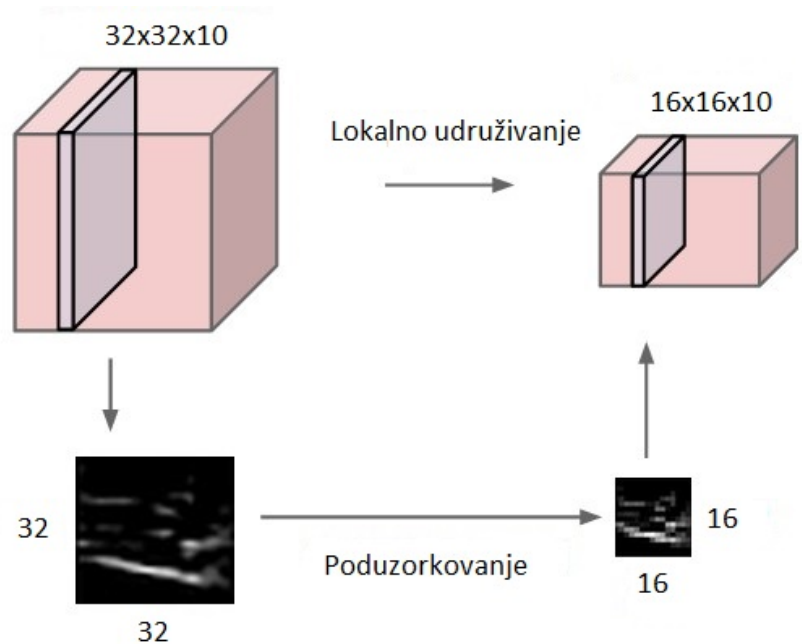
Slika 5.2: Prikaz dijeljenja težina na višoj razini apstrakcije

Za učenje mreža sa dijeljenim težinama još uvijek se može koristiti metoda gradijentnog spusta, ali uz malu preinaku. Gradijent dijeljene težine računa se kao suma gradijenata svih parametara koji se dijele. Osim što ostvaruje prostornu invarijantnost lokalnih filtera, dijeljenje težina znatno doprinosi i procesu učenja konvolucijske mreže. Dijeljenjem težina između parametara mreže smanjujemo ukupan broj slobodnih parametara koje je potrebno naučiti, što smanjuje trajanje učenja. No ono što je još važnije, smanjenjem broja slobodnih parametara poboljšava se sposobnost generalizacije mreže stavljanjem dodatnog ograničenja

na njenu kompleksnost.

5.3. Lokalno udruživanje

Još jedan važan koncept dubokih konvolucijskih mreža je lokalno udruživanje (engl. *Pooling*), koje se može definirati kao oblik nelinearnog poduzorkovanja. U praksi se najviše koristi inačica maksimalnog lokalnog udruživanja (engl. *Max pooling*) koja za poduzorkovanje koristi operaciju traženja maksimuma. Postupak maksimalnog lokalnog udruživanja na slici sastoji se od dijeljenja slike na nepreklapajuće podprozore fiksne veličine, te uzimanja maksimalne vrijednosti svakog podprozora za rezultat. Dobivena slika za podprozore veličine 2x2 bila bi u tom slučaju dvostruko manja od originalne slike, sa vrijednostima svih maksimalnih elemenata podprozora. Maksimalno lokalno udruživanje se u dubokim konvolucijskim mrežama ne primjenjuje na ulaznoj slici, već kao dodatni sloj nakon svakog sloja konvolucije. Time ova metoda postaje korisna na dva načina: eliminira vrijednosti koje nisu maksimalne što smanjuje količinu računanja u višim slojevima, te ostvaruje jedan oblik invarijantnosti na translaciju. Invarijantnost proizlazi iz činjenice da se uzimanjem maksimalne vrijednosti nekog podprozora prepoznaje odziv lokalnog filtera na bilo kojoj lokaciji unutar podprozora, odnosno dopušta se mali pomak naučenih značajki u prostoru.



Slika 5.3: Primjer lokalnog udruživanja na 10 mapa značajki

5.4. Ispravljena linearna jedinica

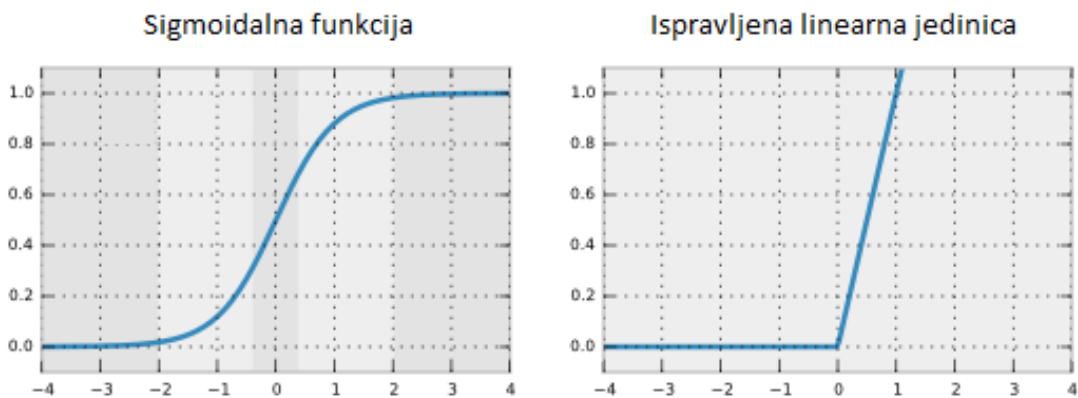
Ono što razlikuje neuronske mreže od linearnih klasifikatora su dodane prijenosne funkcije za postizanje nelinearnosti. U standardnoj potpuno povezanoj neuronskoj mreži svaki element sadrži prijenosnu funkciju, najčešće sigmoidalnog tipa. Dodavanje nelinearnosti u dubokoj konvolucijskoj mreži postiže se dodavanjem sloja koji na svaku vrijednost prijašnjeg sloja primjenjuje prijenosnu funkciju.

Primjenom sigmoidalne prijenosne funkcije (Izraz 5.2) kod učenja duboke arhitekture pojavljuje se problem nestajućeg gradijenta (engl. *Vanishing gradient problem*) [10]. Zbog svojstva sigmoidalne funkcije da ograničava vrijednosti na mali izlazni interval, vrijednosti gradijenata postaju izrazito male ukoliko se prosjeđuju kroz veliki broj slojeva, što onemogućuje kvalitetno učenje. Rješenje ovog problema prvi je osmislio Yoshua Bengio primjenom ispravljene linearne jedinice (engl. *Rectified Linear Unit - ReLU*) koja ne ograničava vrijednosti veće od nule [7]. Ispravljena prijenosna funkcija definirana je operacijom maksimuma, što ju čini izrazito jednostavnom za računanje (Izraz 5.3).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.2)$$

$$f(x) = \max(x, 0) \quad (5.3)$$

Osim što se rješava problem nestajućeg gradijenta, postiže se i efekt raspršene aktivacije, odnosno u prosjeku se samo pola elemenata aktivira svakim prolazom kroz mrežu što pospješuje sposobnost generalizacije.

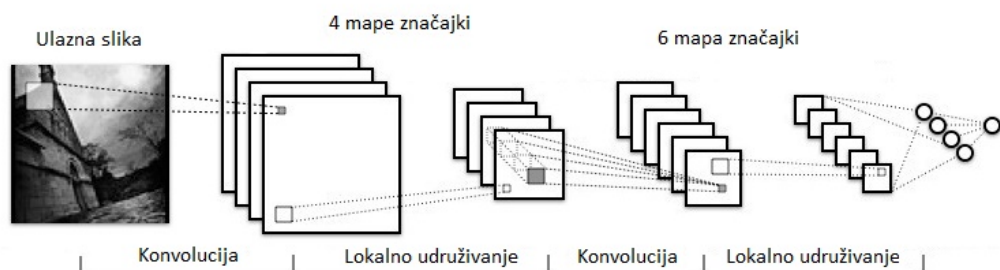


Slika 5.4: Usporedba sigmoidalne funkcije i ispravljene linearne jedinice

5.5. Konačni model

Kao što je ranije spomenuto, konačni model duboke konvolucijske mreže sastoji se od dijela za formiranje hijerarhije značajki i dijela za klasifikaciju. Prvi dio modela ostvaren je slojevima koji implementiraju ranije spomenute koncepte dubokih konvolucijskih mreža, a drugi standardnim potpuno povezanim slojevima.

Nakon svakog sloja konvolucije dodaje se sloj maksimalnog lokalnog udruživanja, a zatim sloj ispravljenih linearnih jedinica za postizanje nelinearnosti. Primjer jednog takvog konačnog modela prikazan je na Slici 5.5, uz pretpostavku da se nakon svakog sloja lokalnog udruživanja primjenjuje prijenosna funkcija.



Slika 5.5: Primjer modela duboke konvolucijske neuronske mreže

Odabir arhitekture dubokih konvolucijskih mreža dodatno je otežan u odnosu na standarne neuronske mreže. Osim standardnih hiperparametara koji određuju broj slojeva i broj elemenata po sloju mreže, za duboku konvolucijsku arhitekturu potrebno je odrediti veličine lokalnih prostornih filtera i filtera maksimalnog lokalnog udruživanja. Veličine lokalnih filtera ponajprije ovise o veličine ulazne slike, te je cilj odrediti veličinu kojom će se kvalitetno formirati apstrakcije značajki po slojevima mreže. Veličine filtera maksimalnog lokalnog udruživanja uobičajeno su male zbog gubljenja velike količine podataka kod većih filtera. Također je važno uzeti u obzir da se veličine mapa značajki smanjuju nakon svakog sloja konvolucije i lokalnog udruživanja, što se u praksi kompenzira dodavanjem većeg broj mapa u višim slojevima.

6. Implementacija

U sklopu ovog rada razvijen je sustav za učenje i testiranje različitih arhitektura neuronskih mreža. Prilikom razvijanja takvog sustava bilo je važno zadovoljiti dva glavna uvjeta: brzo učenje kompleksnih arhitektura na velikoj količini podataka te detaljna analiza rada naučenih mreža. Iz tog razloga sustav je podijeljen na dva podsustava razvijena sa zasebnim alatima. Podsustav zadužen za učenje implementiran je u razvojnom alatu Torch koji omogućuje paralelno izvođenje procesa učenja na grafičkoj kartici [4]. Podsustav koji provodi analizu rada naučenih mreža implementiran je u jeziku C++ korištenjem OpenCV biblioteke [3], a za pokretanje Torch modela u tom okruženju koristi se C++ biblioteka JitTorch koju je razvio Jonathan Tompson [18].

Kako bi se dobio dodatni uvid u rad naučenih neuronskih mreža napravljena je prilagodba C++ implementacije Torch modela OCR sustavu tvrtke Microblink. Time je omogućeno korištenje naučenih modela kao klasifikatora u Microblink mobilnim aplikacijama te dodatna analiza primjenom na mobilnom uređaju.

6.1. Sustav za učenje

Kako bi cijeli proces istraživanja različitih arhitektura mreža i podataka bio što efikasniji, potrebno je prije svega imati brz sustav za učenje. Učenje velikih arhitektura na velikoj količini podataka može trajati tjednima, pa i mjesecima, sekvencijalnim računanjem na standardnom procesoru računala. Stoga je tek pojavom specijaliziranih biblioteka koje omogućuju optimizirano paralelno računanje na grafičkoj kartici postalo moguće provoditi istraživanje u području dubokog učenja. Osim brzine učenja, važna je i jednostavnost definiranja parametara mreže koja se uči, kao i parametara samog procesa učenja. Kod implementacije ovog rada odabran je razvojni alat Torch zbog svoje jednostavnosti i brzine, ali i podrške velikih kompanija kao što su Facebook i Google, koje nerijetko dijele korisne implementacije sa zajednicom.

6.1.1. Torch razvojni alat

Prilikom odabira razvojnog alata uzeti su u obzir i alati Caffe i Theano [11][2]. Možda najpopularniji alat za duboko učenje Caffe ima izrazito jednostavno sučelje za odabir parametara mreže i procesa učenja, no kao posljedica toga teško je raditi promjene u postojećoj implementaciji u svrhu istraživanja ili razvoja. Theano s druge strane pruža puno manju razinu apstrakcije te ostavlja mogućnost lake izmjene većine implementacije, no postaje nepraktičan kod česte upotrebe standardnih algoritama. Torch je po razini apstrakcije negdje između prethodno navedenih alata, dovoljno je niske apstrakcije za laku izmjenu implementacije kod razvoja i istraživanja, ali ima i sučelje za standardne algoritme koje ga čini praktičnim.

Brzina samog izvođenja procesa učenja na grafičkim karticama velikim dijelom ovisi o pozadinskoj implementaciji paralelnog računanja koja je za sve navedene alate otprilike podjednako brza, te se toliko često mijenja da nema smisla uspoređivati njihove brzine. S druge strane, važan faktor kod odabira alata je podrška akademske zajednice i industrije s obzirom da o tome ovisi koliko brzo se razvijaju implementacije novih koncepata. Torch ima implementaciju za većinu novih koncepata u dubokom učenju zbog podrške kompanija kao što su Google, Facebook i Twitter. Tako je primjerice implementirana relativno nova metoda izbacivanja (engl. *Dropout*)[17] za regularizaciju prilikom učenja mreže te ispravljena linearna jedinica (engl. *Rectified Linear Unit - ReLU*), funkcija prijelaza koja je pokazale bolje rezultate od standardne sigmoidalne funkcije u nedavnim istraživanjima.

Sama implementacija Torcha podijeljena je na dva dijela. Sve računske operacije na grafičkoj kartici napisane su kao C/CUDA implementacija za maksimalnu brzinu, dok je ostatak funkcionalnosti implementiran u skriptnom jeziku LuaJIT za lakše korištenje alata.

6.1.2. Definiranje arhitekture

Neuronska mreža je u Torch alatu definirana kao model koji se sastoji od sekvencijalno povezanih slojeva koji vrše različite računske operacije. Tako je na primjer standardna dvoslojna neuronska mreža definirana kao sekvencijalni niz koji prvo linearno množi ulazni vektor sa težinama prvog sloja, zatim primjenjuje prijenosnu sigmoidalnu funkciju na svim elementima prvog sloja, te konačno linearno množi vektor elemenata prvog sloja sa težinama drugog sloja kako bi se dobile izlazne vrijednosti mreže. Jedan takav Torch model prikazan je na Slici 6.1.

```
-- Simple 2-layer neural network
model = nn.Sequential()
model:add(nn.Reshape(ninputs))
model:add(nn.Linear(ninputs,nhiddens))
model:add(nn.Tanh())
model:add(nn.Linear(nhiddens,noutputs))
```

Slika 6.1: Definiranje Torch modela dvoslojne neuronske mreže

Za definiranje duboke konvolucijske mreže u Torchu potrebno je dodatno koristiti slojeve za prostornu konvoluciju, maksimalno lokalno udruživanje i ispravljene linearne jedinice. Primjer definiranja jednog dubokog konvolucijskog modela prikazan je na Slici 6.2.

```
-- hidden units, filter sizes
nstates = {20,40,180}
filtsize = 3
poolsize = 2

model = nn.Sequential()

-- stage 1 : filter bank -> squashing -> max pooling
model:add(nn.SpatialConvolutionMM(1, nstates[1], filtsize))
model:add(nn.ReLU())
model:add(nn.SpatialMaxPooling(poolsize))

-- stage 2 : filter bank -> squashing -> max pooling
model:add(nn.SpatialConvolutionMM(nstates[1], nstates[2], filtsize))
model:add(nn.ReLU())
model:add(nn.SpatialMaxPooling(poolsize))

-- stage 3 : standard 2-layer neural network
model:add(nn.View(nstates[2]*filtsize*filtsize))
model:add(nn.Linear(nstates[2]*filtsize*filtsize, nstates[3]))
model:add(nn.ReLU())
model:add(nn.Linear(nstates[3], noutputs))
```

Slika 6.2: Definiranje Torch modela duboke konvolucijske neuronske mreže

6.2. Sustav za testiranje

Detaljna analiza rada naučenih mreža ključna je za pronalaženje optimalne arhitekture. Mjera optimalnosti nekog modela općenito se može definirati kao omjer njegove točnosti klasifikacije i njegove kompleksnosti. Ovisno o području primjene postoje određena ograničenja i prioriteti za različite karakteristike modela. U ovom radu razvijen je sustav koji analizira naučene mreže u dovoljno detalja da bi se donosili kvalitetni zaključci o njihovoj primjenjivosti. Osim analize pojedinih modela omogućeno je i simultano analiziranje više modela za što lakšu usporedbu njihovih pojedinih karakteristika.

6.2.1. Matrica greške klasifikacije

Testiranjem klasifikatora na skupu podataka u konačnici se dobivaju dvije oznake klasa za svaki podatak. Jedna oznaka je ručno određena te uvijek predstavlja točnu klasu podatka, dok drugu dobivamo kao rezultat klasifikatora koji se testira. Standardna mjera točnosti klasifikatora koristi se dobivenim oznakama samo za informaciju o tome da li oznaka klasifikatora odgovara označenoj oznaci, te vraća kao informaciju ukupni postotak točno klasificiranih podataka. Promatranjem klasa oznaka koje klasifikator vraća u slučaju krive klasifikacije može se dobiti bolji uvid u rad analiziranog modela.

		Oznaka dobivena klasifikatorom						Greška klasifikacije
		1	2	3	4	5	H	
Tražena oznaka	1	18	0	0	0	0	2	0.100
	2	0	19	1	0	0	0	0.050
	3	0	2	18	0	0	0	0.100
	4	0	0	0	20	0	0	0.000
	5	0	0	0	1	19	0	0.050
	H	3	0	0	0	0	17	0.150

Slika 6.3: Primjer matrice greške klasifikacije

Matrica greške klasifikacije definirana je redovima koji predstavljaju ručne oznake klasa, te stupcima koji predstavljaju oznake dobivene klasifikatorom. Matrica se generira brojanjem uređenih parova ručne oznake i oznake klasifikatora, odnosno tražene i dobivene oznake. Tako bi na primjer potpuno točan klasifikator generirao matricu sa vrijednostima nula na svim mjestima osim na glavnoj

dijagonali matrice gdje su tražene i dobivene oznake jednake. Ovakav prikaz greške klasifikacije omogućuje uvid u točnost klasifikatora za pojedine klase, kao i informaciju o klasama koje se često međusobno zamjenjuju.

6.2.2. Prosječno trajanje klasifikacije

Kako bi se donio zaključak o primjenjivosti nekog modela potrebno je poznavati i mjeru njegove kompleksnosti. Sustav za testiranje računa za svaku arhitekturu ukupan broj računskih operacija potrebnih za klasifikaciju jedne slike. Kod standardnih potpuno povezanih mreža taj broj odgovara ukupnom broju veza između neurona, dok se kod kovolucijskih arhitektura prilikom računanja tog broja moraju uzeti u obzir veličine filtera i mjere njihovog preklapanja.

Kompleksnost modela se može definirati i brojem slobodnih parametara, koji se koristi i za procjenu sposobnosti generalizacije klasifikatora. Kod potpuno povezanih mreža broj slobodnih parametara jednak je broju računskih operacija. Kod dubokih konvolucijskih arhitektura taj je broj manji od broja računskih operacija zbog ranije opisane metode dijeljenja težina lokalnih filtera.

Iako se pomoću broja računskih operacija može aproksimirati prosječno trajanje klasifikacije, sustav za testiranje provodi i empirijsko mjerenje. Testiranjem klasifikatora na skupu podataka bilježi se vrijeme potrebno da klasifikator odredi ulaznoj slici oznaku, što nam omogućuje računanje prosječnog trajanja klasifikacije.

6.2.3. Simultano analiziranje više modela

Osim analize pojedinih modela, sustav za testiranje omogućuje i simultano analiziranje više modela s ciljem njihove lakše usporedbe. Kod simultanog analiziranja prikazuju se sve ranije navedene informacije za svaki pojedini model, uz dodatak tabličnih prikaza pojedinih karakteristika svih modela.

Kako bi rezultati bili što pregledniji, nakon analize svih modela generiraju se dva tablična prikaza. Prva tablica omogućuje lako uspoređivanje točnosti klasifikatora po pojedinim klasama i njihove ukupne točnosti. Druga tablica prikazuje prosječno trajanje klasifikacije za svaki analizirani model.

Kod problema klasifikacije slika analizu znatno olakšava mogućnost vizualne provjere podataka. Vrlo često se može donijeti neki zaključak o slabostima klasifikatora promatranjem slika krivo klasificiranih primjera. Sustav za testiranje

sprema za svaki analizirani model krivo klasificirane slike u zasebne datoteke ovisno o krivo određenoj klasi.

6.3. Prilagodba Microblink OCR sustavu

Istraživanje u ovom radu ostvareno je uz dodatnu podršku tvrtke Microblink koja razvija rješenja za mobilne uređaje u području računalnog vida. Njihovo rješenje sastoji se od vlastitog OCR sustava optimiziranog za rad na mobilnom uređaju, te implementacija raznih mobilnih aplikacija koje primjenjuju taj sustav. Neke od implementacija su aplikacije za plaćanje računa, izvlačenja informacija iz osobnih dokumenata, pa čak i rješavanje matematičkih jednažbi.

Microblink OCR sustav sastoji se od dva glavna dijela, podsustava za segmentaciju znakova sa slike, te podsustava za klasifikaciju tih znakova. U ovom radu analizirani su modeli neuronskih mreža za klasifikaciju znakova, te su ulazni skupovi podataka već segmentirani znakovi iz slike. Stoga kako bi se naučeni modeli mogli pokretati na mobilnom uređaju napravljena je prilagodba C++ implementacije Torch modela sučelju Microblink podsustava za klasifikaciju.

Sa implementiranom prilagodbom Microblink sustavu moguće je koristiti bilo koji naučeni Torch model kao klasifikator u bilo kojoj Microblink mobilnoj aplikaciji. Time je omogućena dodatna analiza rada modela iz njegove izravne primjene na mobilnom uređaju. Korištenjem aplikacija može se dobiti uvid u točnost i brzinu klasifikacije u raznim uvjetima i na različitim mobilnim uređajima.

7. Rezultati

Cilj ovog rada bio je analizirati rad dubokih konvolucijskih mreža na problemu raspoznavanja znakova te ih usporediti sa standarnim neuronskim mrežama. Modeli su učeni i testirani na internim skupovima podataka tvrtke Microblink, a prilikom testiranja provedena je analiza i njihovog internog klasifikatora.

Istraživanje je provedeno na problemu raspoznavanja znakova na osobnim iskaznicama, takozvanim strojno čitljivim područjima (engl. *Machine Readable Zone - MRZ*). MRZ podatci imaju 37 klasa znakova fonta ocrb, te obuhvaćaju znamenke, velika slova i znak manje. U ovom radu koriste se dva skupa takvih podataka za učenje i testiranje klasifikatora. Prvi skup zove se sintetizirani skup, a dobiven je slikanjem printanih imitacija osobnih iskaznica. Sastoji se od 450 tisuća označenih slika znakova te je slikan sa više različitih modela mobitela u raznim uvjetima osvjetljenja. Drugi skup dobiven je slikanjem pravih osobnih iskaznica u različitim uvjetima, te sadrži oko 12 tisuća ručno označenih slika znakova. Za potrebe učenja i testiranja sintetizirani skup je podijeljen na 300 tisuća slika za učenje i 150 tisuća za testiranje modela, a ručno označeni skup koristi se za dodatno testiranje sposobnosti generalizacije modela.

7.1. Utjecaj predprocesiranja podataka

U ranijim poglavljima spomenuta je uobičajena praksa predprocesiranja slika znakova prije njihovog klasificiranja. Promjenjivi uvjeti okoline rezultiraju raznim transformacijama originalne slike koje otežavaju njenu klasifikaciju, stoga je logično primjenjivati metode ublažavanja posljedica tih transformacija. Takve metode najčešće su neki oblik operacije binarizacije, različite po načinu na koji se definira njihov prag. Jedna takva metoda koja uzima u obzir lokalne promjene osvjetljenja je metoda adaptivnog praga opisana ranije u radu.

Metode predprocesiranja razvijene su sa ciljem naglašavanja korisnih informacija, te odbacivanjem onih nepotrebnih. Mana ovog pristupa je što se odluka

o korisnosti informacija donosi pomoću ručno razvijenih metoda zasnovanih na ljudskom promatranju podataka, zbog čega dolazi do gubitka korisnih informacija koje čovjek nije u stanju vidjeti. Kako bi se koristile sve informacije iz podataka, potrebne su metode koje pronalaze korisne informacije izravno iz skupa podataka. Takve metode razvijaju se u području dubokog učenja gdje se na temelju skupa podataka uči cijela hijerarhija značajki i klasifikator u jednom procesu učenja. Posljedice promjenjivih uvjeta okoline se tada rješavaju na način da se procesom učenja na velikoj količini različitih oblika podataka razvije kvalitetna hijerarhija značajki otporna na nastale transformacije.

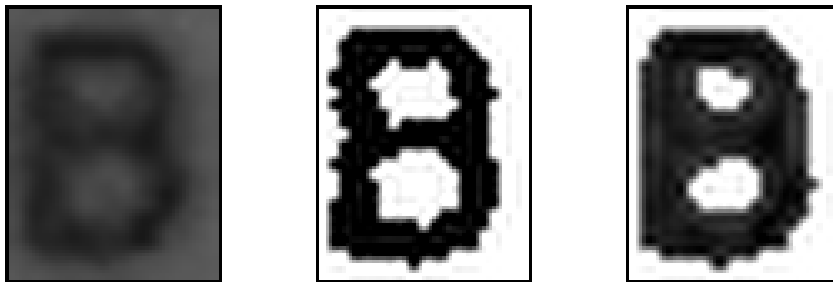
U ovom radu provedeno je istraživanje kako bi se analizirao utjecaj predprocesiranja podataka na točnost klasifikatora. Korištena je duboka konvolucijska mreža sa 10 mapa značajki u prvom i 20 mapa značajki u drugom konvolucijskom sloju. Testiranje je provedeno na slikama dimenzija 18 piksela u 3 različita oblika: predprocesiranim metodom adaptivnog praga, predprocesiranim internom metodom tvrtke Microblink, te originalnim slikama bez predprocesiranja.

Skup podataka	Adaptivni prag	Microblink interni	Bez predprocesiranja
Sintetizirani	98.29%	97.87%	99.57%
Ručno označeni	96.35%	97.66%	99.21%

Tablica 7.1: Rezultati istraživanja utjecaja predprocesiranja podataka

Dobiveni rezultati su općenito lošiji za ručno označeni testni skup, što je logično s obzirom da su mreže učene na sintetiziranom skupu koji je dobiven slikanjem printanih imitacija osobnih iskaznica. Rezultati mreže učene na originalnim podacima bez predprocesiranja pokazuju da mreža dobro generalizira s obzirom da su postotci točnosti klasifikacija na oba testna skupa vrlo visoki. Interna metoda tvrtke Microblink pokazala je lošije rezultate na oba testna skupa, no bolje od metode adaptivnog praga koja je vrlo loše generalizirala na ručnom testnom skupu.

Analizom krivo klasificiranih primjera za mreže koje se koriste predprocesiranim podacima može se dobiti bolji uvid u probleme ovog pristupa. Jedan takav primjer može se vidjeti na Slici 7.1. Prikazana je originalna slika slova 'B' lijevo, slika dobivena predprocesiranjem adaptivnim pragom u sredini, te rezultat primjene Microblink metode predprocesiranja desno. Veliki broj krivo klasificiranih



Slika 7.1: Primjer predprocesiranja slike adaptivnim pragom i Microblink metodom

slika za mreže sa predprocesiranim podacima rezultat je zamjene slova 'B' i znamenke '8'. Za razlikovanje ove dvije klase jedna dobro naučena mreža koristiti će se razlikom u zaobljenosti njihovih znakova. Na Slici 7.1 može se vidjeti da metode predprocesiranja binariziranjem krivo označavaju neke dijelove slike, čime znak poprima zaobljenost karakterističnu za znamenku '8'.

7.2. Pronalaženje optimalne arhitekture mreže

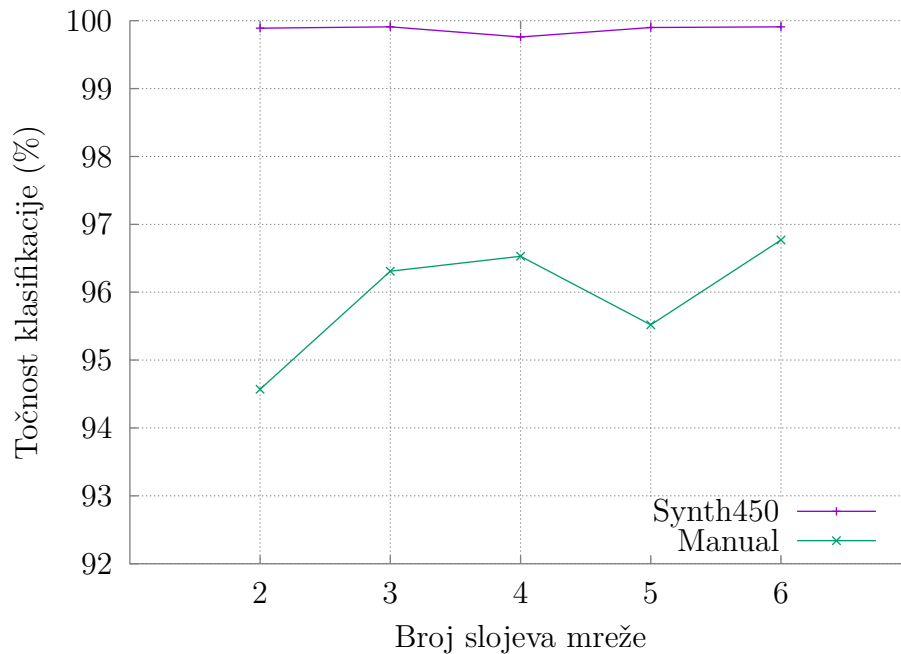
Glavna mjera kvalitete naučenog modela je njegova sposobnost generalizacije, odnosno koliko dobro klasificira dotad neviđene primjere. Odabir prave arhitekture modela ključan je za dobru generalizaciju, stoga je u ovom radu proveden postupak pronalaženja optimalne arhitekture. Ovaj postupak se još naziva i selekcija modela, a zasniva se na testiranju više arhitektura različitih kompleksnosti. Prilikom odabira optimalne arhitekture postupno se povećava kompleksnost modela, te se promatra njegova točnost na skupovima podataka za testiranje.

Selekcija modela napravljena je za standarne neuronske mreže i duboke konvolucijske neuronske mreže kako bi se u konačnici mogli usporediti rezultati njihovih najboljih arhitektura.

7.2.1. Standarne neuronske mreže

Dodavanje slojeva neuronskoj mreži u teoriji omogućuje razvijanje hijerarhije značajki koja robusnije predstavlja objekt klasifikacije. U svrhu što boljeg razumijevanja utjecaja dubine arhitekture i njenog tipa (standarni ili konvolucijski) na uspješnost klasifikacije, provedeno je istraživanje i nad dubokim standarnim ne-

uronskim mrežama. Analiziran je rad 5 arhitektura različitih dubina, od plitke dvoslojne mreže do duboke mreže sa šest slojeva. Rezultati su prikazani uzlazno s obzirom na njihovu kompleksnost na Slici 7.2.



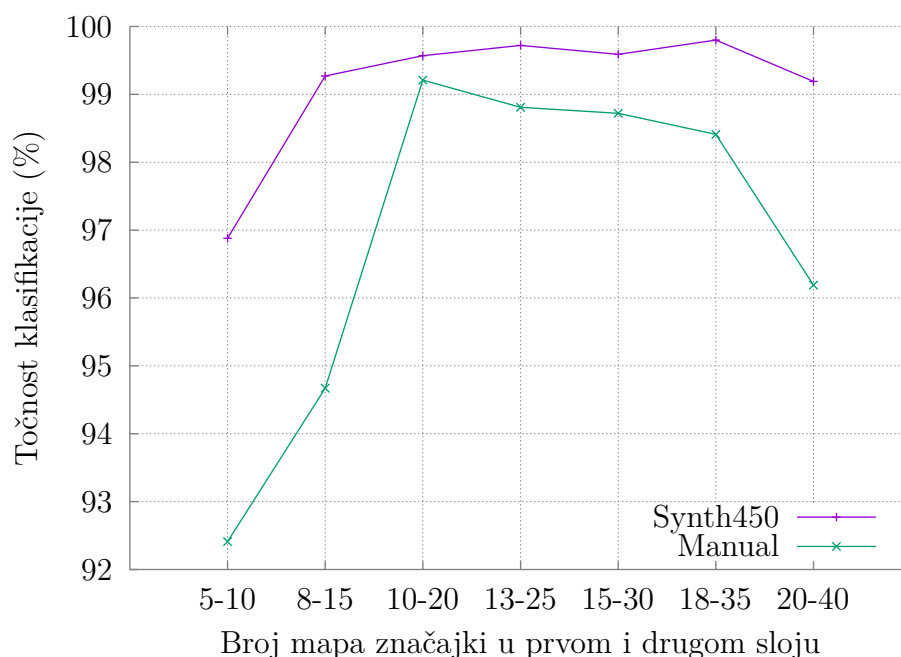
Slika 7.2: Usporedba 5 arhitektura neuronskih mreža različitih dubina

Dobiveni rezultati pokazuju značajnu razliku u točnosti klasifikacije između dva testna skupa za sve arhitekture. Prilikom testiranja ustanovljeno je da svi klasifikatori vrlo dobro generaliziraju na testnom sintetiziranom skupu iako on nije korišten za njihovo učenje. Razlog tome mogu biti kontrolirani uvjeti u kojima je sintetizirani skup generiran, te jednolikost printanih imitacija kartica. Rezultati na ručno označenom setu lošiji su nego na sintetiziranom za sve arhitekture te nam daju bolji uvid u sposobnost generalizacije standardnih neuronskih mreža.

Utjecaj dubine arhitekture, odnosno broja slojeva mreže, nije vidljiv na rezultatima na sintetiziranom testnom skupu zbog već vrlo visokog postotka točnosti klasifikacije dvoslojne mreže. Na ručno označenom skupu vidljivi su bolji rezultati prilikom dodavanja jednog sloja dubine dvoslojnoj arhitekturi, no dodatno dodavanje slojeva ne pokazuje značajno i konzistentno bolje rezultate. Stoga je optimalna arhitektura standardnih neuronskih mreža ona sa tri sloja neurona.

7.2.2. Konvolucijske neuronske mreže

Kod definiranja arhitekture konvolucijskih mreža postoje određena ograničenja prilikom odabira njenih parametara. S obzirom da se ulazna slika postepeno smanjuje u dimenzijama zbog primjene konvolucijske jezgre i lokalnog udruživanja, broj slojeva mreže je ograničen. U ovom radu koriste se relativno male ulazne slike dimenzija 18 piksela te lokalni filteri dimenzija 3 piksela, stoga mreže mogu imati najviše dva konvolucijska sloja. Analizirano je 7 arhitektura sa različitim brojem mapa značajki u svakom konvolucijskom sloju, te su prikazani rezultati s obzirom na njihovu kompleksnost na Slici 7.3.



Slika 7.3: Usporedba 7 dubokih konvolucijskih arhitektura različitih kompleksnosti

Kao i kod analize standardnih neuronskih mreža rezultati na sintetiziranom testnom skupu bolji su od onih na ručno označenom skupu, no za određene arhitekture ta je razlika vrlo mala. Na prikazu rezultata vidljiva je standardna krivulja generalizacije za oba testna skupa. Točnost klasifikacije je mala za nedovoljno kompleksne modele jer njihova arhitektura nema dovoljno slobodnih parametara da bi kvalitetno opisala podatke hijerarhijom značajki. Za pretjerano kompleksne modele je točnost na testnim skupovima također manja zbog previše slobodnih parametara koji omogućuju pretjeranu prilagođenost modela podatcima za učenje. Analizom je pronađena optimalna arhitektura sa 10 mapa značajki u prvom konvolucijskom sloju te 20 mapa značajki u drugom konvolucijskom sloju. Re-

zultati modela sa optimalnom arhitekturom pokazuju dobru sposobnost generalizacije na oba testna skupa.

7.3. Usporedba optimalnih arhitektura

Usporedbom najboljih predstavnika standardnih i konvolucijskih neuronskih mreža moguće je dobiti uvid u utjecaje različitih tipova arhitektura. Analizom standardnih mreža pokazano je da se dodavanjem dubine plitkoj dvoslojnoj mreži povećava njena sposobnost generalizacije. Optimalna konvolucijska arhitektura također je duboka, stoga dobiveni rezultati upućuju na prednost dubokih arhitektura nad plitkima.

U Tablici 7.2 prikazana je usporedba rezultata dva optimalna modela na oba testna skupa. Točnost klasifikacije na sintetiziranom skupu približno je jednaka za oba modela, no na ručno označenom skupu rezultati su bolji kod konvolucijskog modela. Rezultati upućuju na to da konvolucijska arhitektura mreže poboljšava sposobnost generalizacije modela.

Skup podataka	Standardna arhitektura	Konvolucijska arhitektura
Sintetizirani	99.91%	99.57%
Ručno označeni	96.31%	99.21%

Tablica 7.2: Usporedba standardnih i konvolucijskih arhitektura

Analizom točnosti klasifikacije za svaku zasebnu klasu te promatranjem krivo klasificiranih primjera mogu se točno vidjeti slučajevi koji standardnoj neuronskoj mreži predstavljaju problem, a koje konvolucijska neuronska mreža uspijeva točno klasificirati. U Tablici 7.3 prikazana je usporedba postotka točnosti klasifikacije za takve slučajeve. Jedan takav slučaj je sličnost između znamenke '0' i slova 'O' koji se razlikuju samo po zaobljenosti konture. Rezultati upućuju na to da lokalni filteri konvolucijskih mreža mogu robusnije predstaviti zaobljene konture i time doprinjeti robusnosti algoritma prilikom razlikovanja ovih klasa. Također problematičan slučaj predstavlja razlikovanje slova 'B' od znamenke '8'. Optimalni model standardne neuronske mreže velik broj slova 'B' krivo je klasificirao kao znamenku '8', dok je konvolucijska mreža imala vrlo visok postotak točnosti klasifikacije za obadvije klase.

Arhitektura	Znamenka '0'	Slovo 'O'	Znamenka '8'	Slovo 'B'
Standardna	74.15%	91.97%	98.69%	38.57%
Konvolucijska	97.55%	95.35%	99.81%	99.29%

Tablica 7.3: Usporedba arhitektura na teškim slučajevima

Osim točnosti naučenih klasifikatora prilikom analize uzeta je u obzir i njihova kompleksnost te prosječno trajanje klasifikacije. Kao što je ranije spomenuto, kompleksnost modela može se aproksimirati preko broja računskih operacija potrebnih za klasifikaciju jedne slike, ili preko broja slobodnih parametara, odnosno broja težina koje se mogu mijenjati prilikom učenja mreže.

Svojstvo	Standardna arhitektura	Konvolucijska arhitektura
Broj računskih operacija	170550	97780
Broj slobodnih parametara	170550	6930
Prosječno trajanje klasifikacije	0.202 ms	0.223 ms

Tablica 7.4: Usporedba svojstava kompleksnosti arhitektura

U tablici 7.4 prikazana je usporedba navedenih svojstava za oba optimalna modela. Po broju računskih operacija standardni model gotovo je dvostruko kompleksniji, no prosječno trajanje klasifikacije približno je jednako za oba modela. Razlog tome je što se prosječno trajanje klasifikacije računa empirijski te ovisi o implementaciji svih računskih operacija, koje u ovom slučaju traju dulje kod konvolucijskih mreža.

Važno je primjetiti da konvolucijska arhitektura ima za red veličine manji broj slobodnih parametara od standardne arhitekture. Manji broj slobodnih parametara izravna je posljedica dodatnih ograničenja koja mehanizmi dubokih konvolucijskih mreža postavljaju na kompleksnost modela. Ovo svojstvo konvolucijskog modela doprinosi njegovoj sposobnosti generalizacije, te omogućuje u konačnici bolje rezultate na ručno označenom skupu podataka.

7.4. Usporedba sa Microblink klasifikatorom

Osim analize raznih arhitektura neuronskih mreža, u sklopu ovog rada provedena je i analiza internog klasifikatora tvrtke Microblink. Radi se o klasifikatoru razvijenom po klasičnim principima rješavanja problema raspoznavanja znakova gdje se klasifikacija provodi u tri glavna koraka. Prvi korak je predprocesiranje ulazne slike gdje se nastoji ublažiti transformacije nastale promjenjivim uvjetima okoline. Idući korak je računanje diskriminantnih značajki ulazne slike kako bi se dodatno olakšao problem klasifikacije, te je konačni korak primjena algoritma klasifikacije na značajkama slike.

Klasifikator	Sintetizirani skup	Ručno označeni skup
Microblink	86.93%	91.69%
Standarna mreža	99.91%	96.31%
Konvolucijska mreža	99.57%	99.21%

Tablica 7.5: Usporedba Microblink klasifikatora sa optimalnim neuronskim mrežama

Usporedbom rezultata u Tablici 7.5 može se vidjeti da interni klasifikator tvrtke Microblink daje lošije rezultate od standardnih i konvolucijskih neuronskih mreža. Postoji više faktora koji u teoriji objašnjavaju ovakve rezultate. Jedan problem je negativni utjecaj predprocesiranja podataka zbog nesavršenosti ručno razvijenih metoda obrade slike. Zbog gubitka informacija neke od slika postaju neprepoznatljive nakon algoritma za predprocesiranje, što čini točnu klasifikaciju nemogućom. Još jedan faktor koji otežava klasifikaciju težih primjera je nesavršenost ručno definiranih značajki slike koje se predaju klasifikatoru. Umjesto da se značajke uče na temelju skupa podataka kako bi se maksimizirala njihova diskriminantnost, one se ručno definiraju na temelju ljudskog promatranja podataka što dovodi do gubitka informacija. Te konačno, ovako razvijeni klasifikator ima samo jedan sloj asptrakcije što onemogućuje stvaranje hijerarhije značajki koje robusnije predstavljaju objekt klasifikacije i poboljšavaju točnost klasifikatora.

Prednost Microblink klasifikatora u odnosu na odabrane modele neuronskih mreža je njegova brzina. U Tablici 7.6 može se vidjeti da je njegovo prosječno trajanje klasifikacije za red veličine manje od onog od neuronskih mreža. Brzine neuronskih modela još uvijek su dovoljno velike za primjenu u stvarnom vremenu

Klasifikator	Prosječno trajanje klasifikacije
Microblink	0.025 ms
Standarna mreža	0.202 ms
Konvolucijska mreža	0.223 ms

Tablica 7.6: Usporedba prosječnog trajanja klasifikacije

na mobilnom uređaju, no konačna brzina klasifikatora može se i dodatno smanjiti ukoliko se koriste u kombinaciji sa bržim Microblink klasifikatorom.

7.5. Moguća poboljšanja

Kao i kod svakog pristupa, moguća su poboljšanja u točnosti i brzini izvođenja klasifikacije. U području dubokog učenja veliki je naglasak na učenju svakog koraka procesa klasifikacije na temelju podataka, stoga je dodatni naglasak na važnosti podataka za učenje. Predstavljene duboke konvolucijske arhitekture pokazale su se sposobnima graditi kvalitetne hijerarhije značajki na temelju podataka, no za to je prije svega potrebna velika količina kvalitetnih podataka koji dobro predstavljaju domenu problema. Stoga je za budući napredak prije svega potrebno prikupljanje što veće količine kvalitetnih podataka za učenje.

Najbolji rezultati u ovom radu postignuti su sa konvolucijskom arhitekturom vrlo male kompleksnosti, svega nekoliko tisuća slobodnih parametara. Dodatnim povećavanjem kompleksnosti smanjivala se sposobnost generalizacije modela, što upućuje na pretjeranu prilagodbu podacima za učenje. Ograničavanje kompleksnosti klasifikatora kako bi se povećala sposobnost generalizacije standardni je pristup prilikom selekcije modela, no pretjerana potreba za ovim metodama upućuje na nedovoljno kvalitetne podatke za učenje. Za učenje se u ovom radu koristi sintetizirani skup podataka prikupljen nad imitacijama osobnih iskaznica i u kontroliranim uvjetima. Stoga je jedno od mogućih poboljšanja razvijanje kvalitetnijih imitacija kartica, te porađivanje na raznolikosti uvjeta prilikom njihovog slikanja.

Naučeni duboki konvolucijski modeli tipično imaju vrlo malu brzinu izvođenja klasifikacije zbog njihove velike dubine i kompleksnosti, što ih čini teže primjenjivima na problemima u stvarnom vremenu. U ovom radu optimalni modeli vrlo su male kompleksnosti zbog ranije spomenutih razloga, te imaju dovoljno veliku

brzinu izvođenja klasifikacije za primjenu u stvarnom vremenu. No poboljšavanjem kvalitete podataka za učenje moći će se iskoristiti potencijal kompleksnijih modela, što će za posljedicu imati problem izvođenja tih modela u stvarnom vremenu. Jedna metoda koja bi mogla poboljšati brzinu ovakvih modela je metoda Geoffreya Hintona pod nazivom distilacija (engl. *Distillation*)[8]. Ideja je koristiti izlaze naučenog velikog modela kao takozvane mekane oznake za svaku sliku prilikom učenja manjeg modela. Istraživanje je pokazalo da se sa mekanim oznakama prenosi dio informacija o podacima koje je prikupio veliki model, što poboljšava konačnu točnost naučenog manjeg modela. Tako naučeni manji model može se koristiti u stvarnom vremenu zbog svoje manje kompleksnosti i time veće brzine izvođenja klasifikacije.

8. Zaključak

Prepoznavanje teksta sa slike ima vrlo široku primjenu u mnogim područjima ljudske djelatnosti. Razvijanjem mobilne tehnologije pojavljuju se nove izazovnije primjene optičkog prepoznavanja znakova. Algoritam za prepoznavanje znakova mora biti robustan na promjenjive uvjete okoline, ali i dovoljno brz za primjenu u stvarnom vremenu na mobilnim uređajima.

Duboko učenje je novo obećavajuće područje istraživanja inspirirano vizualnim korteksom mozga sisavca. Zasniva se na učenju korisnih prezentacija podataka povezanih u obliku hijerarhije, te je pokazalo jako dobre rezultate na problemima raspoznavanja uzoraka. Za problem klasifikacije u računalnom vidu razvijene su duboke konvolucijske neuronske mreže koje traže lokalne značajke na slici pomoću lokalnih filtera ostvarenih konvolucijom.

U ovom radu razvijen je sustav za učenje i testiranje različitih arhitektura neuronskih mreža, kao i prikaz rezultata u svrhu usporedbe različitih modela. Korištenjem sustava provedeno je istraživanje primjene standardnih neuronskih mreža i dubokih konvolucijskih neuronskih mreža na problemu raspoznavanja znakova. Rezultati su prikazani i komentirani sa ciljem pronalaska optimalne arhitekture modela za primjenu na mobilnom uređaju, a osim točnosti naučenih klasifikatora uzeta je u obzir i njihova kompleksnost te prosječno trajanje klasifikacije. Također je razvijeno sučelje za korištenje naučenih modela mreža kao klasifikatora u Microblink OCR sustavu čime je omogućena dodatna analiza primjenom modela na mobilnom uređaju.

Dobiveni rezultati potvrdili su prednosti primjene dubokih konvolucijskih mreža na problemu raspoznavanja znakova.

Analiziranjem rada klasifikatora sa i bez predprocesiranja podataka pokazana je prednost prosljeđivanja neobrađenih slika klasifikatoru u svrhu očuvanja informacija slike. Posljedice promjenjivih uvjeta okoline se tada rješavaju na način da se procesom učenja na velikoj količini različitih oblika podataka razvije kvalitetna hijerarhija značajki otporna na nastale transformacije.

Postupkom selekcije modela nad standardnim neuronskim arhitekturama ustanovljeno je da se dodavanjem dubine neuronskoj mreži pospješuje njena sposobnost generalizacije. Ovaj rezultat može se protumačiti kao posljedica razvijanja hijerarhije značajki koja robusnije predstavlja objekt detekcije.

Usporedbom rezultata optimalnih modela standardnih i konvolucijskih neuronskih mreža pokazana je prednost korištenja konvolucijske arhitekture na problemu raspoznavanja znakova. Konvolucijska arhitektura ima za red veličine manji broj slobodnih parametara od standardne arhitekture, što je izravna posljedica dodatnih ograničenja koje mehanizmi dubokih konvolucijskih mreža postavljaju na kompleksnost modela. To svojstvo konvolucijskog modela doprinosi njegovoj sposobnosti generalizacije, te omogućuje u konačnici bolje rezultate na ručno označenom skupu podataka.

Rezultati klasifikatora tvrtke Microblink lošiji su od rezultata standardnih i konvolucijskih neuronskih mreža, što se može objasniti negativnim utjecajem predprocesiranja podataka i ručnog definiranja značajki samo jedne razine apstrakcije. Prednost Microblink klasifikatora u odnosu na odabrane modele neuronskih mreža je njegova za red veličine veća brzina, no brzine neuronskih modela još uvijek su dovoljno velike za primjenu u stvarnom vremenu na mobilnom uređaju.

LITERATURA

- [1] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.
- [2] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, i Yoshua Bengio. Theano: a CPU and GPU math expression compiler. U *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Lipanj 2010. Oral Presentation.
- [3] G. Bradski. Open source computer vision library - opencv. *Dr. Dobb's Journal of Software Tools*, 2000.
- [4] R. Collobert, K. Kavukcuoglu, i C. Farabet. Torch7: A matlab-like environment for machine learning. U *BigLearn, NIPS Workshop*, 2011.
- [5] Navneet Dalal i Bill Triggs. Histograms of oriented gradients for human detection. U *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, stranice 886–893, Washington, DC, USA, 2005. IEEE Computer Society. URL <http://dx.doi.org/10.1109/CVPR.2005.177>.
- [6] Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.
- [7] Xavier Glorot, Antoine Bordes, i Yoshua Bengio. Deep sparse rectifier neural networks. U Geoffrey J. Gordon i David B. Dunson, urednici, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, svezak 15, stranice 315–323. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011. URL <http://www.jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf>.

- [8] G. Hinton, O. Vinyals, i J. Dean. Distilling the Knowledge in a Neural Network. *ArXiv e-prints*, Ožujak 2015. URL <http://adsabs.harvard.edu/abs/2015arXiv150302531H>.
- [9] G E Hinton i R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, Srpanj 2006. URL <http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed&uid=16873662&cmd=showdetailview&indexed=google>.
- [10] S. Hochreiter, Y. Bengio, P. Frasconi, i J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. U Kremer i Kolen, urednici, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
- [11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, i Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, i L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Prosinac 1989. ISSN 0899-7667. URL <http://dx.doi.org/10.1162/neco.1989.1.4.541>.
- [13] Yann Lecun, Léon Bottou, Yoshua Bengio, i Patrick Haffner. Gradient-based learning applied to document recognition. U *Proceedings of the IEEE*, stranice 2278–2324, 1998.
- [14] WarrenS. McCulloch i Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. URL <http://dx.doi.org/10.1007/BF02478259>.
- [15] David E. Rumelhart, Geoffrey E. Hinton, i Ronald J. Williams. Neuro-computing: Foundations of research. poglavlje Learning Representations by Back-propagating Errors, stranice 696–699. MIT Press, Cambridge, MA, USA, 1988. URL <http://dl.acm.org/citation.cfm?id=65669.104451>.
- [16] I. Sobel i G. Feldman. A 3x3 isotropic gradient operator for image processing. 1968.

- [17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, i Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Siječanj 2014. URL <http://dl.acm.org/citation.cfm?id=2627435.2670313>.
- [18] J. Tompson. jtorch - jonathan's torch7 utility library (c++, opencl). <https://github.com/jonathantompson/jtorch>, 2013.
- [19] Šnajder, Dalbello Bašić, i Čupić. Umjetne neuronske mreže. Skripta predmeta, Fakultet elektrotehnike i računarstva Zagreb, 2008.

DUBOKE KONVOLUCIJSKE NEURONSKE MREŽE ZA RASPOZNAVANJE ZNAKOVA

Sažetak

U ovom radu provedeno je istraživanje primjene dubokih konvolucijskih neuronskih mreža na problemu raspoznavanja znakova. U tu svrhu razvijen je sustav za učenje i testiranje različitih arhitektura neuronskih mreža. Podsustav za učenje implementiran je pomoću razvojnog alata za paralelno računanje na grafičkoj kartici kako bi postupak bio što brži. Dio sustava za testiranje razvijen je s ciljem detaljne analize rada naučenih modela, te jednostavne i pregledne usporedbe više različitih modela.

Istraživanje je ostvareno uz dodatnu podršku tvrtke Microblink specijalizirane za razvoj tehnologija računalnog vida za mobilne uređaje. Za učenje i testiranje modela koriste se od tvrtke interni skupovi podataka, a prilikom testiranja provodi se analiza i njihovog internog klasifikatora. Rezultati su prikazani i komentirani sa ciljem pronalaska optimalne arhitekture modela za primjenu na mobilnom uređaju. Pritom je osim točnosti naučenih klasifikatora uzeta u obzir i njihova kompleksnost te prosječno trajanje klasifikacije.

Ključne riječi: računalni vid, raspoznavanje znakova, neuronske mreže, duboko učenje, konvolucijske mreže

Deep convolutional neural networks for character recognition

Abstract

In this thesis, deep convolutional neural networks for optical character recognition have been researched. For this purpose a system for learning and testing of different architectures of neural networks has been developed. Subsystem for learning was implemented with development tools for parallel computing on graphics cards to make the procedure as fast as possible. Part of the system for testing was developed for detailed analysis of the learned models, as well as simple and transparent comparison of different models.

The research had additional support from the company Microblink specialized in the development of computer vision technology for mobile devices. Internal datasets from the company are used for the learning and testing of models, and additional analysis was conducted on the companies internal classifier. The results are presented and discussed with the aim of finding the optimal model architecture for the application on a mobile device. In addition to the accuracy of the learned classifiers, their complexity and average classification duration was taken into account as well.

Keywords: computer vision, character recognition, neural networks, deep learning, convolutional networks