

UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

MASTER RAD

**Autofokusiranje elektronskih mikroskopa
primenom dubokih neuronskih mreža**

Autor:
Lazar RANKOVIĆ

Mentor:
dr Mladen NIKOLIĆ



Beograd,
2018.

Sadržaj

1	Uvod	1
1.1	Izazovi, nedostaci dosadašnjih rešenja i prostor za poboljšanje	1
1.2	Doprinosi	2
2	Mikroskopija i obrada digitalnih fotografija	3
2.1	Uvod u mikroskopiju	3
2.2	Obrada digitalnih fotografija	4
	Predstavljanje i obrada slika	5
3	Uvod u autofokusiranje	7
3.1	Dosadašnja rešenja	7
3.2	Fokusiranje	7
3.3	Mere fokusiranosti	8
3.4	Tradicionalni pristupi autofokusiranja	10
	3.4.1 Brentov metod	11
	3.4.2 Algoritam penjanja uzbrdo	11
4	Mašinsko učenje	15
4.1	Neuronske mreže	16
	4.1.1 Neuronske mreže sa propagacijom unapred	16
4.2	Regularizacija	17
4.3	Mere kvaliteta modela	18
5	Predloženi pristup	19
5.1	Priprema atributa	19
	5.1.1 Procesiranje slika	19
	5.1.2 Izračunavanje mera fokusiranosti	20
	5.1.3 Generisanje atributa	20
5.2	Modeli	21
	Model koji koristi jednu sliku (M_1)	22
	Model koji koristi dve slike (M_2)	23
5.3	Optimizacioni metod	25
5.4	Podaci, treniranje i evaluacija modela	26
	5.4.1 Skup podataka	27
	5.4.2 Priprema skupa podataka	27
	5.4.3 Treniranje i evaluacija modela	29
5.5	Rezultati i poređenje sa ostalim pristupima	31
6	Softver	35
6.1	Simulator mikroskopa	35
6.2	Radni okvir za razvoj modela mašinskog učenja	36
7	Diskusija	41

8 Zaključak	43
Bibliografija	45

Glava 1

Uvod

Problem fokusiranih slika je široko rasprostranjen i sastavni je deo svih optičkih elektronskih uređaja novijeg datuma. Potreba za fokusiranjem slike se može naći u optičkim elektronskim uređajima kao što su: pametni telefoni, digitalni fotoaparati, teleskopi, mikroskopi i sl. U ovom radu će akcenat isključivo biti na mikroskopima. Ljudi gledanjem slike jednostavno mogu da odrede da li je slika fokusirana ili ne. Ta veština dolazi sa iskustvom i nakon pregledanih nekoliko desetina slika čovek je u stanju da prilikom gledanja jedne slike klasifikuje slike na fokusirane i nefokusirane. Ova klasifikacija za čoveka je vrlo jednostavna i čini se da problem nije težak. Kako svet teži sve većoj automatizaciji jednostavnih poslova tako i problem klasifikacije fokusiranih i nefokusiranih slika treba da bude automatizovan. Pojam automatizovanog fokusiranja slike se naziva *autofokusiranje*. Autofokusiranje predstavlja autonomno (bez pomoći čoveka) fokusiranje slike na nekom uređaju. Postoje dve grupe metoda autofokusiranja: aktivno i pasivno autofokusiranje [1]. Aktivno autofokusiranje koristi senzore da odredi udaljenost kamere od željenog objekta, dok pasivan način autofokusa može biti podeljen u dve grupe:

- metodi koji koriste dodatne uređaje

Zahteva dodatni hardver koji služi za autofokus kao npr. podeljena prizma.

- metodi koji koriste samo sliku

Ne zahteva dodatni uređaj već koristi samo sliku dobijenu pomoću mikroskopa. Primer je metod detekcije kontrasta.

U ovom radu biće predstavljen jedan nov pristup autofokusiranja koji pripada grupi pasivnog autofokusa i to koristi samo sliku kao jedinu informaciju potrebnu za dobijanje fokusirane slike. Pristup odstupa od tradicionalnog pristupa rešavanju ovog problema, a razlozi će detaljnije biti opisani u narednim glavama. Metod koji je predstavljen zasnovan je na primeni veštačke inteligencije, preciznije, primeni algoritma mašinskog učenja koji pokušava da iz iskustva nauči zakonitosti pomoću kojih je moguće predvideti konfiguraciju uređaja u kojoj je slika idealno fokusirana. Zato je važno izvršiti fokusiranje u minimalnom broju koraka, i u ovom aspektu se nalazi potencijal za značajna unapređenja postojećih postupaka.

1.1 Izazovi, nedostaci dosadašnjih rešenja i prostor za poboljšanje

Proces automatizacije procesa nije uvek jednostavan. U slučaju autofokusiranja potrebno je odrediti nekakvu metriku po kojoj bi bilo jasno da li je neka slika fokusirana ili ne. Ta metrika će nadalje biti nazivana *mera fokusiranosti*. Za sada nije pronađena mera fokusiranosti koja u opštem slučaju može tačno da opiše stepen fokusiranosti slike. To je posledica osobine mera fokusiranosti da nisu invarijantne na

sadržaj slike, već direktno zavise od sadržaja na slici. Štaviše, neke mere zavise i od osobina slike kao što je nivo osvetljenosti. Mere fokusiranosti su u početku bile zasnovane na gradijentima slike, kasnije su dodavane mere bazirane na statistici dok se sada uvode mere koje zahtevaju viši nivo apstrakcije kao što je ugao u određenom Banahovom prostoru. Iako ove mere mogu u nekoj meri da opišu nivo fokusiranosti slika i dalje ne postoji rešenje koje u globalnom smislu može rešiti fokusiranje. Poučeni ovim iskustvom za sada jedini način poređenja fokusiranosti slika jeste poređenje slike sa istim sadržajem samo sa različitim nivoom fokusiranosti. Kada god se vrši fokusiranje ili pak autofokusiranje poređenje se odvija relativno (između dve slike sa istim sadržajem), ne postoji način globalnog poređenja. Slika koja ima najvišu vrednost mere fokusiranosti se uzima za fokusiranu sliku. Iz ovih razloga sadašnji metodi autofokusiranja zasnivaju se na primeni numeričkih algoritama optimizacija - algoritmi koji imaju za cilj da pronađu globalni maksimum određene mere fokusiranosti.

Sve što je u radu vezano za mikroskope odnosi se na elektronske mikroskope koji su povezani na računar. Kada se koriste ti mikroskopi najveći deo vremena odlazi na proces *akvizicije* (eng. *acquisition*). Akvizicija je celokupan proces dobijanja slike pomoću mikroskopa, obuhvata zahtev računara za slikom iz mikroskopa, uzimanje slike i povratak slike iz uređaja do računara. Očigledno je da je akvizicija vremenski zahtevan proces te je jako bitan broj akvizicija koje se izvrše tokom autofokusiranja. Broj akvizicija tokom autofokusa takođe je bitan iz još jednog razloga, taj razlog je primena autofokusa u slučaju fluorescentne mikroskopije. U ovoj grani mikroskopije sa svakom akvizicijom preparat koji se posmatra oštećuje se pod uticajem svetlosti, tako da porastom broja akvizicija opada kvalitet uzorka koji će biti izučavan.

1.2 Doprinosi

U ovom radu predložen je originalan metod autofokusiranja, čija se glavna prednost u odnosu na postojeće ogleda u istovremenom smanjenju greške fokusiranja i broja akvizicija. Dodatna prednost pristupa koji će biti predstavljen u ovom radu je taj da ne zahteva dodatni hardver, veliku računarsku moć, već može da se izvršava i na prosečnim desktop računarima. Metod ne pripada optimizacijama lokalne pretrage kao većina pristupa već je baziran na globalnim stanjima pa je i robustan na razne oblike šuma. Model ne koristi samo jednu meru fokusiranosti u toku fokusiranja već informaciju dobija iz više različitih mera fokusiranosti. Više o ovome biće u narednim glavama.

Glava 2

Mikroskopija i obrada digitalnih fotografija

Budući da se u radu akcenat stavlja na autofokusiranje slika dobijenih pomoću mikroskopa, u ovoj glavi će ukratko, bez zalaska u fizičke teorije, biti objašnjeni osnovni principi rada mikroskopa. Biće predstavljeno osnovno znanje vezano za mikroskope koje je značajno za razumevanje rada.

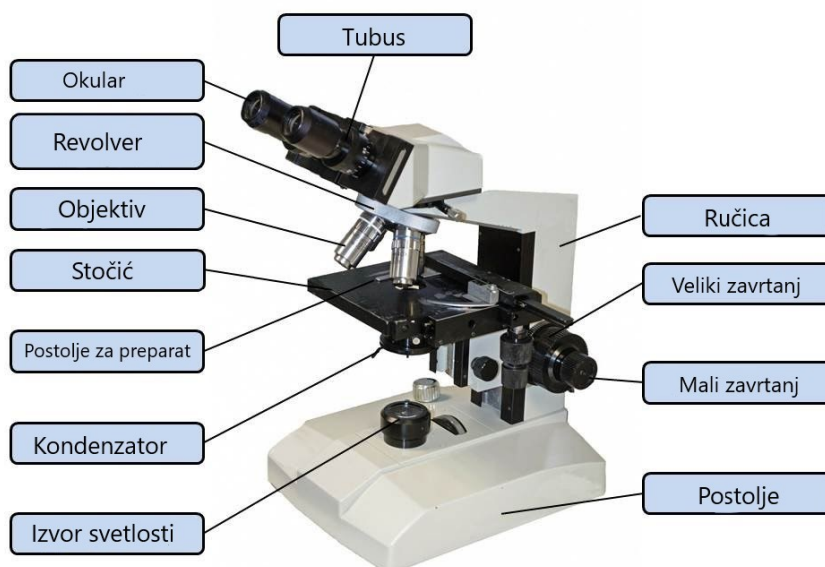
2.1 Uvod u mikroskopiju

Radovoljnost ljudi da gledaju sitne objekte, teško ili nemoguće vidljive golim okom prirodno je dala za posledicu razvoj alata koji to omogućavaju. Prvi takav alat je lupa. Može se reći da je lupa preteča mikroskopa. Budući da je ta radoznalost rasla ka sve manjim i manjim objektima prirodno se težilo razvoju alata koji imaju veću moć uvećanja. Tada nastaje mikroskop.

Kroz istoriju mikroskopi su evoluirali tako da se moć uvećanja postepeno povećavala. Pored ove osobine polako su postajali sve manje zavisni od čoveka. Prvi svetlosni mikroskopi, koje je otkrio Luj Paster 1958. godine, koristili su prirodno svetlo, a za fino podešavanje pozicije sočiva/stočića su koristili zavrtnje koji su ručno pomerani. Na slici 2.1 su prikazani delovi svetlosnog mikroskopa. Pomoću malog zavrtnja (eng. *fine focus*) i velikog zavrtnja (eng. *coarse focus*) se fokusira slika, odnosno pomeranjem stočića (eng. *stage*). Stočić se tim točkicama podešava vertikalno i u matematičkom duhu se taj smer pomeranja stočića naziva z-pozicija (eng. *z-stage*). Stočić može da se pomera i horizontalno u dva pravca, ali taj deo nije bitan te detaljnije neće biti razmatran.

Kasnije su došli elektronski mikroskopi koji kao izvor osvetljenja koriste snop elektrona. Poboljšanja su u smislu većeg uvećanja ali je suštinski princip rada ostao isti kao i kod svetlosnih mikroskopa. Moderni mikroskopi kakvi se danas uglavnom koriste, imaju motorizovane delove koji služe za fokusiranje. Točkici koji su ručno pomerani radi fokusiranja sada su prikačeni na električne motore te je taj deo automatizovan. Više nije potrebno da čovek direktno gleda u uzorak kroz okular, već je tu postavljena kamera te je mikroskop povezan sa računarom i slika se direktno vidi na ekranu računara. Primer modernog motorizovanog elektronskog mikroskopa se može videti na slici 2.2. Na slici se mogu videti i koji su delovi motorizovani. U ovom trenutku razvoja mikroskopa cilj je da oni rade autonomno bez nadzora čoveka.

Očigledno je da čovek više nije potreban da ručno podešava poziciju stočića, već se to radi pomoću motora. Takođe, čovek više ne mora da posmatra uzorak kroz

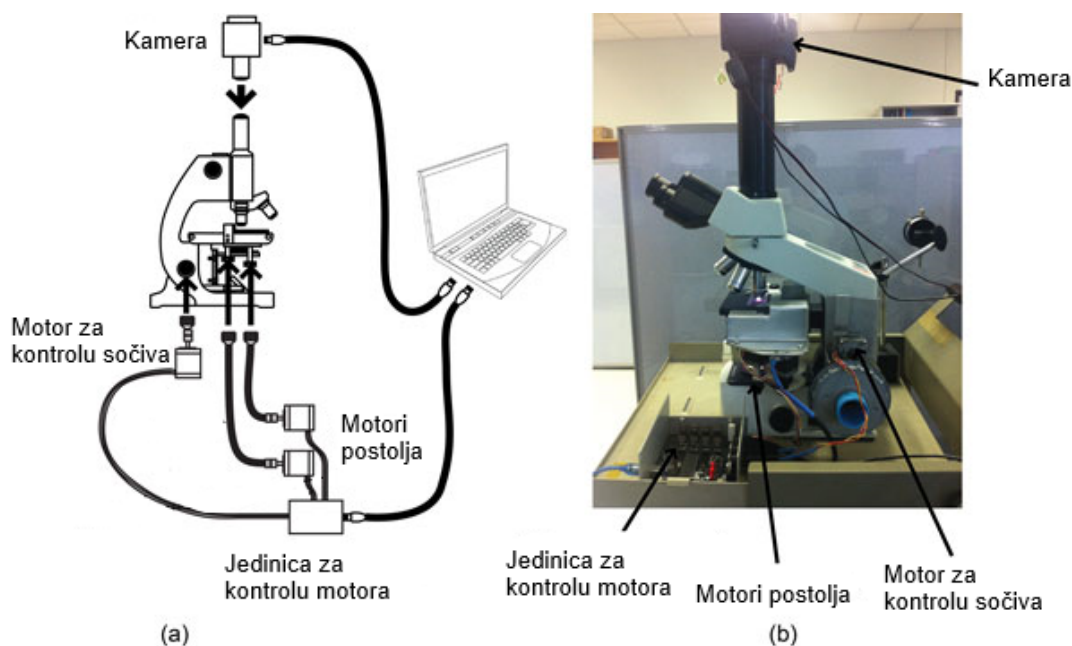


SLIKA 2.1: Primer svetlosnog mikroskopa.

okular već je slika dostupna u računar, te je pored toga slika pristupačna i za računarsku obradu. Sada postoje svi uslovi da se potpuno automatizuje korišćenje mikroskopa što omogućava razne pogodnosti kao što su veći protokog uzoraka, detekcija objekata na slici, smanjenje ljudskih napora prilikom obrade uzoraka, omogućavanje posmatranja uzoraka na daljinu itd. Poslednji doprinos je možda jedan od najvažnijih u današnjem vremenu, jednostavnim slanjem uzorka na željenu adresu istraživač bi bio u mogućnosti da gleda svoj uzorak pod sofisticiranim mikroskopima i sve to bez putovanja, smeštaja i dodatnih troškova. Uz prednosti koje donosi automatizacija mikroskopa očigledno se nameće i dosta problema, sve to treba računarski rešiti da bude efikasno. Obrada slika (eng. *image processing*) je centralna tema koja se vezuje za rešavanje problema u ovom domenu. Stoga će u nastavku biti više reči o tome.

2.2 Obrada digitalnih fotografija

Krajem dvadesetog i početkom dvadesetprvog veka obrada digitalnih fotografija doživljava svoju ekspanziju. Obrada digitalnih fotografija datira jos od 1960-ih godina. Tih godina započinje obrada fotografija na raznim institutima i fakultetima kao što su: Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland i ostali. Oni su se bavili obradom fotografija dobijenih pomoću satelita. Razvijanje ove grane računarstva veoma je bitno zbog primena u medicini, automobilske industriji itd. U sadašnjem vremenu spoj obrade fotografije i veštačke inteligencije ima široku primenu kako u akademiji tako i u industriji. Primeri su: obrada fotografija u medicinske svrhe (prepoznavanje ćelija raka, pravljanje referentnog skeleta, prepoznavanje anomalija, itd.), klasifikacija fotografija na online prodavnicama kao što su Ebay, Amazon itd. Nakon što je ukazano na široku primenu obrade digitalnih fotografija u nastavku će biti prikazane osnovni pojmovi bitni za ovaj rad.



SLIKA 2.2: Primer motoritovanog mikroskopa.

Predstavljanje i obrada slika

Slike se u računarstvu mogu predstaviti koristeći *vektorski zapis*, *rasterski zapis* ili *kombinovani zapis*.

U vektorskom obliku, slike se predstavljaju pomoću konačnog broja geometrijskih figura (tačkica, linija, krivih i poligona). Svaka figura na slici se predstavlja svojim koordinatama ili jednačinom u Dekartovoj ravni.

U rasterskom obliku, slika je predstavljena pravougaonom matricom komponenti koje se nazivaju pikseli (eng. *PICTure ELeMent*). Svaki piksel je individualan i opisan je jednom bojom.

Za predstavljanje crno belih slika (eng. *grayscale*) boju piksela dovoljno je predstaviti intenzitetom svetlosti. Slike u boji mogu se predstavljati u modelu boja RGB (eng. *Red Green Blue*), CMYK (eng. *Cyan Magenta Yellow*) itd. U ovom radu vršiće se obrada samo crno belih slika tako da će akcenat biti na ovom tipu slika.

Dinamički raspon slike predstavlja dubinu slike, tj. opseg nijansi koje može sadržati jedan piksel. Pošto piksel uzima vrednosti brojeva od $[0, 2^{b*8} - 1]$, gde je b broj bajtova onda broj nijansi zavisi od broja bajtova po kanalu i to određuje kvalitet slike. Jasno je da što je opseg veći to će slika biti kvalitetnija.

Naredna definicija će posebno biti korišćena u narednim glavama radi unapređenja efikasnosti prilikom obrade slika.

Definicija 1. Dvodimenzionalna konvolucija u slučaju primene filtera (f) na početnu sliku (I) može definisati na sledeći način:

$$(I * f)_{ij} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} I_{k,l} f_{i-k,j-l} \quad (2.1)$$

Konvolucija je bitan pojam u obradi fotografija i često se koristi. Štaviše praktično svaka obrada fotografija može biti predstavljena kao konvolucija slike I sa filterom f .

U ovom radu ima elemenata i obrade slike kao i računarskog vida (*engl. computer vision*). Obrada slike se generalno svodi na primenu različitih filtera na sliku, kao što su detekcija ivica, detekcija uglova, detekcija karakterističnih tačaka itd. Obradom slike se bavi prvi deo rada i to se odnosi na izračunavanje atributa slike 3.3. Sa druge strane računarski vid se bavi razvojem modela koji mogu da razumeju neki sadržaj sa slike, tj. mogu da zamene čoveka i obavljaju zadatke umesto njega. Na primer, to su detekcija objekata, autonomna vožnja automobila itd. Zapravo, pristup rešavanja autofokusa predstavljen u ovom radu bi se mogao svrstati u računarski vid.

Glava 3

Uvod u autofokusiranje

U prvom delu ove glave biće predstavljene mere koje opisuju nivo fokusiranosti slike. Drugi deo se bavi tradicionalnim načinima autofokusiranja.

3.1 Dosadašnja rešenja

Kao što je pomenuto, zbog prirode mera fokusiranosti tradicionalni metodi autofokusa koriste optimizacione algoritme. Dva optimizaciona algoritma koji se najčešće koriste su Brentov metod [2] i algoritam penjanja uzbrdo (eng. *hill climbing*) [1], koji će biti diskutovani u nastavku.

Većina radova iz oblasti mikroskopije se bavi pronalaženjem novih mera fokusiranosti ili poboljšanjima već razvijenih mera, manji deo radova se bavi razvojem novih algoritama. U ovoj glavi biće predstavljeni neki od najboljih algoritama autofokusiranja.

Dobar pregled mera fokusiranosti predstavljen je u radu Hashim Mir et al. [3]. U ovom radu su predstavljene sve relevantne mere fokusiranosti strukturane po uspešnosti na određenim vrstama uređaja. Jedan od tradicionalnih metoda autofokusa je predstavljen u radu Hea i koautora [4]. Autori predstavljaju optimizacioni algoritam dobijen modifikacijom algoritma penjanja uzbrdo sa adaptivnom veličinom koraka. Ove dve modifikacije značajno ubrzavaju rad algoritma a samim tim i brzinu fokusiranja. U vreme razvoja novog metoda autofokusiranja pronađena su dva relevantna rada koji primenom algoritama mašinskog učenja rešavaju ovaj problem. Rudi Čen i Peter van Bek [5] predlažu korišćenje stabala odlučivanja. Predložen je kontrolni algoritam koji je implementiran kao konačni automat. Mašinsko učenje je primenjeno na odlučivanje vezano za tranzicije između stanja dok traži najbolju poziciju stočića. Hašim Mir i ostali koautori [6] predlažu da se mašinsko učenje primeni na izbor veličine koraka. Ukratko je ovo pregled najvažnijih pronađenih radova na temu autofokusiranja.

3.2 Fokusiranje

Postavlja se pitanje: šta je to fokus? Kada je neka slika u fokusu? Iako ljudi mogu jasno i jednostavno da daju odgovor na to pitanje, odgovor na to pitanje nije sasvim jednostavno izraziti u nekom formalnom obliku.

Budući da se autofokusiranje obavlja međusobnim poređenjem slike sa istim sadržajem na različitim pozicijama stočića onda neophodno je uvesti metriku koja opisuje nivo fokusiranosti trenutno dobijene slike. Metrike koje nam omogućavaju međusobno poređenje slika nazivaćemo mere fokusiranosti i neke od njih će biti predstavljene u nastavku.

3.3 Mere fokusiranosti

Postoje različite mere kojima se matematički može opisati koliko je neka slika dobro fokusirana. Između mnoštva različitih mera fokusiranosti izabrane su one koje su najbolje za slike dobijene pomoću mikroskopa. U ovoj glavi prikazaćemo mere fokusiranosti koje su među najboljima iz domena mikroskopije. Definišimo sada funkciju fokusiranosti.

Definicija 2. Funkcija ϕ je mera fokusiranosti ako $\phi : \mathcal{I} \rightarrow \mathbb{R}$ gde je domen skup slika a kodomen je realan broj. Slika je predstavljena kao matrica brojeva. Funkcija dostize globalni maksimum kada je slika najbolje fokusirana.

U nastavku će biti predstavljene neke od najboljih mera za domen mikroskopije [3]. Važno je napomenuti da je slika I predstavljena matricom.

- Mere fokusiranosti bazirane na prvom izvodu

- Kvadratni gradijent

$$\phi(I) = \sum_{x=0}^{n-1} \sum_{y=0}^{m-2} (I_{x,y+1} - I_{x,y})^2 \quad (3.1)$$

- Brenerova mera

$$\phi(I) = \sum_{x=0}^{n-1} \sum_{y=0}^{m-3} (I_{x,y+2} - I_{x,y})^2 \quad (3.2)$$

- Vertikalna Brenerova mera

$$\phi(I) = \sum_{x=0}^{n-3} \sum_{y=0}^{m-2} (I_{x+2,y} - I_{x,y})^2 \quad (3.3)$$

- Sobelov operator

Matrica S predstavlja Sobelov operator.

$$S = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

Konvolucijom Sobelovog operatora sa slikom I dobija se nova matrica R dimenzija $N \times M$ i to se računa po formuli $R = I * S$. Sobelova mera fokusiranosti slike se izračunava prema formuli 3.4.

$$\phi(I) = \sum_{i=1}^N \sum_{j=1}^M (R_{i,j})^2 \quad (3.4)$$

- Šarov operator

Matrica S predstavlja Šarov operator.

$$S = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}$$

Konvolucijom Šarovog operatora sa slikom I dobija se nova matrica R dimenzija NxM i to se računa po formuli $R = I * S$. Šarova mera fokusiranosti slike se izračunava prema formuli 3.5.

$$\phi(I) = \sum_{i=1}^N \sum_{j=1}^M (R_{i,j})^2 \quad (3.5)$$

- Mere fokusiranosti bazirane na drugom izvodu
 - Kvadratni Laplasov operator

Matrica KL predstavlja kvadratni Laplasov operator.

$$KL = \begin{bmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & +6 & 0 & -3 \\ -4 & +6 & +20 & +6 & -4 \\ -3 & 0 & +6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{bmatrix}$$

Konvolucijom kvadratnog Laplasovog operatora sa slikom I dobija se nova matrica R dimenzija NxM i to se računa po formuli $R = I * KL$. kvadratna Laplasova mera fokusiranosti slike se izračunava prema formuli 3.6.

$$\phi(I) = \sum_{i=1}^N \sum_{j=1}^M (R_{i,j})^2 \quad (3.6)$$

- Mere fokusiranosti bazirane na statistici
 - Varijansa

$$\frac{1}{nm} \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} (I_{x,y} - \mu)^2 \quad (3.7)$$

- Mere fokusiranosti bazirane na histogramima
 - Histogram entropije

$$- \sum_{x=0}^k p(x) \log_2 p(x) \quad (3.8)$$

- Mere fokusiranosti bazirane na korelaciji
 - Volatova F4 mera

$$\phi(I) = \sum_{x=0}^{n-2} \sum_{y=0}^{m-1} I_{x,y} I_{x+1,y} - \sum_{x=0}^{n-3} \sum_{y=0}^{m-1} I_{x,y} I_{x+2,y} \quad (3.9)$$

- Volatova F5 mera

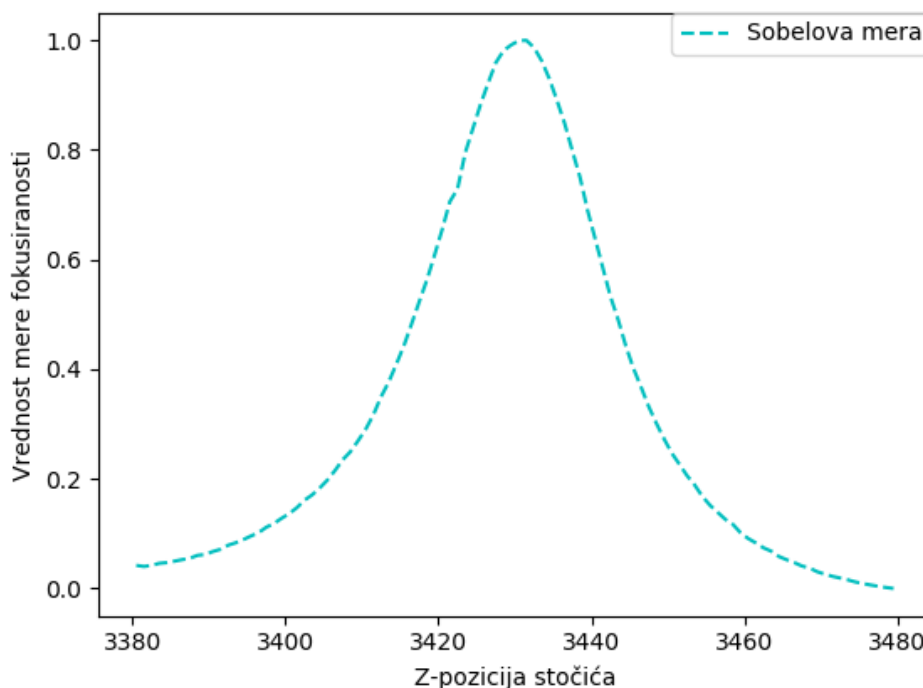
$$\phi(I) = \sum_{x=0}^{n-2} \sum_{y=0}^{m-1} I_{x,y} I_{x+1,y} - n * m \mu^2 \quad (3.10)$$

- Spektralne mere fokusiranosti

- Furijeova transformacija

$$\hat{I}_m = \frac{1}{PQ} \sum_{j=0}^{P-1} \sum_{k=0}^{Q-1} I_{jk} e^{2\pi i(jl/P + km/Q)} \quad (3.11)$$

Primer grafika jedne mere fokusiranosti može se videti na slici 3.1. U ovom slučaju mera fokusiranosti je dobijena primenom Sobelovog operatora 3.4 duž vertikalnih pozicija stočića. Na apscisi se nalaze pozicije stočića mikroskopa a na ordinati se nalaze vrednosti mere fokusiranosti skalirane na opseg [0, 1].



SLIKA 3.1: Primer mere fokusiranosti.

Gledajući sliku 3.4 mogu se uočiti određeni problemi. Maksimum funkcije fokusiranosti dostiže se u fokusiranim slikama. S tim u vezi prirodno se nameće rešavanje ovog problema nekim metodama koje pronalaze globalni maksimum funkcije, što ne mora biti lako. Pored toga, ne postoji tačno jedna slika koja se može uzeti kao fokusirana slika. To je posledica nepostojanja privilegovane mere fokusiranosti.

3.4 Tradicionalni pristupi autofokusiranja

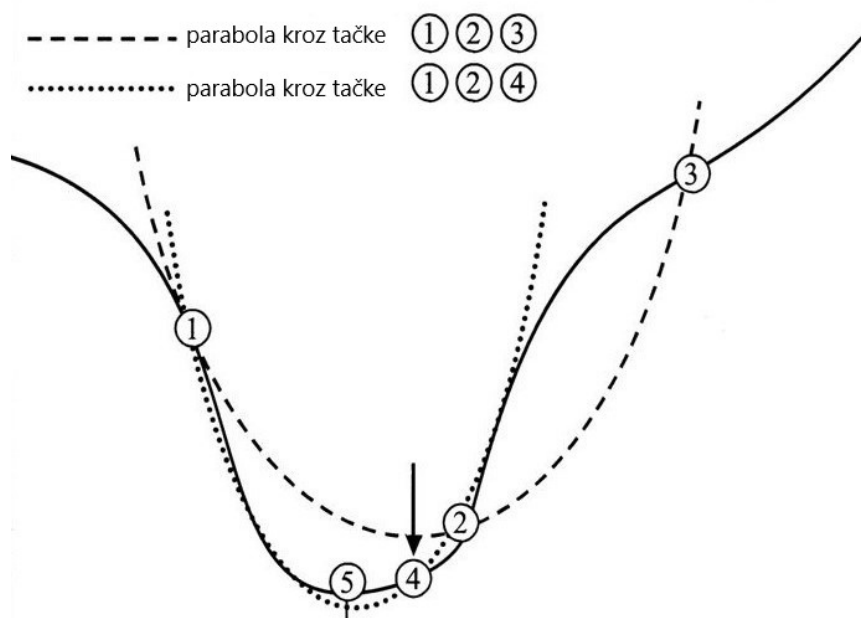
Budući da ne postoji analitički zapis mere fokusiranosti, analitičke metode nalaženja ekstremuma funkcije se odmah odbacuju, tako da jedino što ostaje je primena

numeričkih metoda. U nastavku će biti prikazan način rešavanja autofokusiranja numeričkim algoritmima optimizacije. Na ovom mestu će biti prikazana dva najčešća algoritma numeričke optimizacije koji se koriste pri autofokusiranju.

3.4.1 Brentov metod

Brentov metod je optimizacioni algoritam koji ne koristi izvode. Predstavljen je u knjizi Richard P. Brenta [2]. Ovaj metod polazi od tri tačke koje moraju zadovoljavati uslov: ukoliko imamo funkciju f , i tri tačke na apscisi a, b, c mora važiti: $a < b < c$ i $f(b) < f(a)$ i $f(b) < f(c)$. U svakom koraku se na osnovu tri tačke, konstruiše parabola kojom se aproksimira funkcija i izračunava se nova tačka koja je minimum te parabole i nova aproksimacija minimuma funkcije. Nakon toga se biraju sledeće tri tačke (od kojih je jedna uvek novokonstruisana tačka), na osnovu kojih se vrši sledeći korak aproksimacije, ali detalji ovog postupka ovde neće biti opisani.

Primer upotrebe Brentovog algoritma je prikazan na slici 3.2.



SLIKA 3.2: Korak izvršavanja Brentovog algoritma.

3.4.2 Algoritam penjanja uzbrdo

Algoritam penjanja uzbrdo (eng. *hill-climbing*) [7] je algoritam lokalne pretrage koji počinje u pseudo-slučajno izabranoj tački i kreće se ka većim vrednostima funkcije sve dok ne dostigne maksimum. Drugim rečima počinje se od neke vrednosti funkcije i poboljšava se vrednost sve dok se ne dostigne optimalna vrednost. Algoritam ima jedan metaparametar α koji određuje veličinu koraka u određenom smeru. Metaparametar α se bira na osnovu smera povećanja vrednosti funkcije, dakle uvek prati smer u kom funkcija raste. Ako se algoritam nalazi na poziciji x_n , sledeća pozicija x_{n+1} se računa $x_{n+1} = x_n + \alpha$. Očigledno je da malo α usporava konvergenciju, isto tako veliko α ubrzava konvergenciju ali optimalna vrednost može biti preskočena, preciznije neće se preskočiti u bukvalnom smislu već će krajnja vrednost biti

previše udaljena od optimalne vrednosti. Zbog svega toga metaparametar α se stavlja da bude adaptibilan, tj. menja veličinu kroz vreme što daje dobre rezultate. Algoritam 1 prikazuje pseudo-kod algoritma penjanja uzbrdo.

Algoritam 1: Algoritam penjanja uzbrdo.

Input: p_c - trenutna pozicija,
 α - veličina koraka,
 $f(x)$ - funkcija koja se optimizuje,
MAX_AKVIZICIJE - maksimalni dozvoljeni broj akvizicija
Output: predviđena pozicija idealne pozicije stočića

```

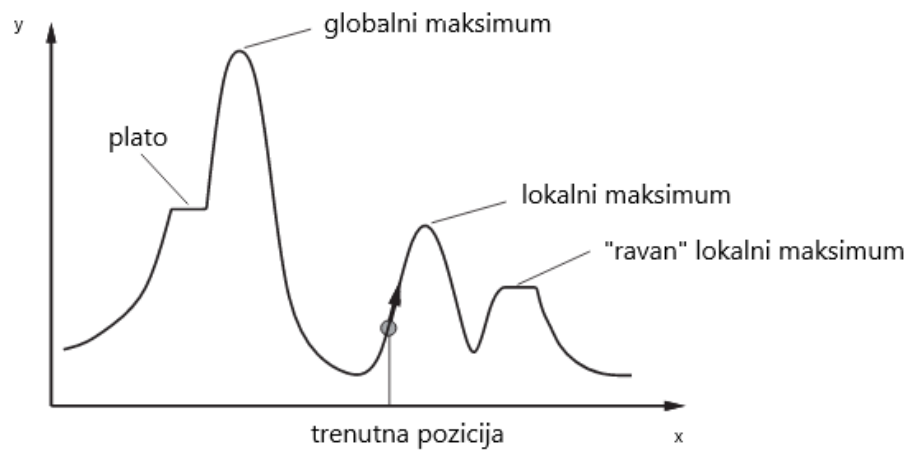
1 iteracije = 0
2 levo =  $p_c - \alpha$ 
3 desno =  $p_c + \alpha$ 
4  $fc = f(p_c)$ 
5  $f1 = f(\text{levo})$ 
6  $f2 = f(\text{desno})$ 
7 smer = 1 if  $f1 < f2$  else -1
8  $p_n = p_c + \text{smer} * \alpha$ 
9  $fn = f(p_n)$ 
10 while iteracije < MAX_AKVIZICIJE do
11     if  $fn < fc$  then
12         return  $p_c$ 
13      $p_c = p_n$   $fc = fn$ 
14      $p_n = p_c + \text{smer} * \alpha$ 
15      $fn = f(p_n)$ 
16     iteracije += 1

```

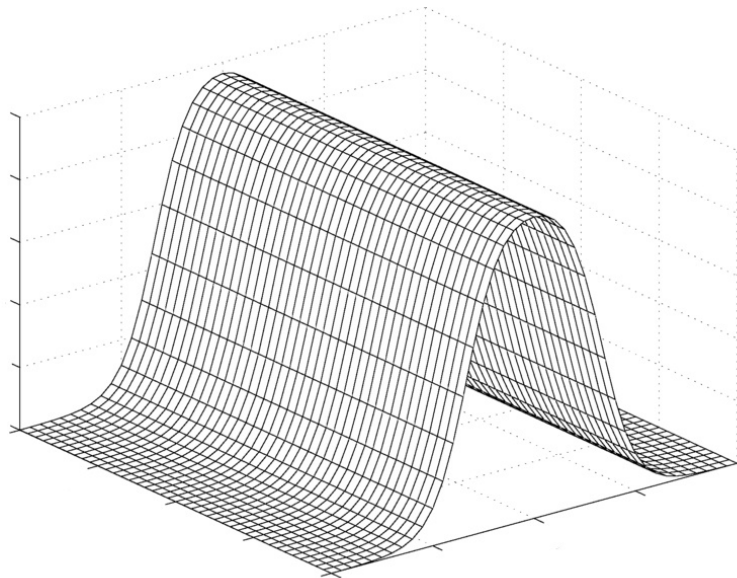
Hill climbing ne može da dosegne optimalnu vrednost (globalni maksimum) funkcije ako naiđe na neki od sledećih regiona [8]:

- Lokalni maksimum: na poziciji lokalnog maksimuma sve tačke u dovoljno maloj okolini imaju vrednost funkcije manju od trenutne tačke, stoga po uslovu zaustavljanja, algoritam prestaje sa pretragom i prijavljuje da je našao optimalnu vrednost, što je ilustrovano slikom 3.3. Ovo je posledica činjenice da je algoritam zasnovan na pohlepnoj pretrazi i razmatra samo lokalne okoline.
- Plato: platoi predstavljaju oblasti prostora pretrage u kojima funkcija ima konstantnu vrednost, pogledati sliku 3.3. Zbog toga je nemoguće odrediti koji potez je najbolji, a samim tim može se desiti da pohlepna pretraga ne nađe izlaz sa platoa.
- Greben: grebeni predstavljaju uske funkcije koje opadaju ili rastu duž nekog pravca, kao što ilustruje slika 3.4. U takvim problemima, penjanje uzbrdo ne vodi u pravcu rasta funkcije, već je potrebno da napravi mnogo cik-cak koraka da se popenje uz greben (ili da se spusti niz greben)

U ovom poglavlju su prikazani najčešći algoritmi koji se koriste za autofokusiranje. Takođe izložene su i mane tih algoritama koji u većoj ili manjoj meri utiču na performanse fokusiranja. Čak i kada bi te mane bile zanemarene jedan od glavnih zahteva i dalje ostaje, a to je da se ni jedan algoritam ne bavi brojem slika koje se



SLIKA 3.3: Algoritam penjanja uzbrdo.



SLIKA 3.4: Greben.

koriste u optimizaciji, što bi bila jedna od osnovnih mana tradicionalnih algoritama.

Glava 4

Mašinsko učenje

Mašinsko učenje je jedna od oblasti veštačke inteligencije koju možemo da posmatramo sa dva stanovišta. Prvi je da se mašinsko učenje bavi izgradnjom prilagodljivih sistema koji mogu da poboljšavaju svoje performanse kroz vreme. Poboljšavanje performansi se postiže korišćenjem informacija iz iskustva. Ovaj pogled je više sa praktičnog stanovišta, dok sa druge strane ako bismo posmatrali iz teorijske perspektive mašinsko učenje bi moglo biti definisano kao disciplina koja se bavi izučavanjem generalizacije i konstrukcijom algoritama koji generalizuju.

Obično se mašinsko učenje deli na tri grupe problema:

- nadgledano (eng. *supervised learning*)
- nenadgledano (eng. učenje *unsupervised learning*)
- učenje potkrepljivanjem (eng. *reinforcement learning*)

Nadgledano učenje je verovatno najraspotranjeniji vid učenja. To je učenje kod kog se pored podataka iz kojih se uči, algoritmu proslede i vrednosti ciljne promenljive. Podaci koji se dovode na ulaz algoritma mašinskog učenja se nazivaju atributi (eng. *features*). Ciljna promenljiva je vrednost koju model treba da predviđa. Procesom treninga se dobija model mašinskog učenja koji je zapravo funkcija koja modeluje zavisnost između atributa i ciljne promenljive. Postoje dva tipa nadgledanog učenja:

- klasterovanje: odnosi se na kontekst u kojem je ciljna promenljiva kategorička vrednost
- regresija: odnosi se na kontekst u kojem je ciljna promenljiva neprekidna vrednost

Nenadgledano učenje se odnosi na problem u kojem algoritam dobije samo vrednosti atributa, ali ne i vrednosti ciljne promenljive. Od algoritma se očekuje da sam nauči zakonitosti u podacima. Primer nenadgledanog učenja je klasterovanje. Nenadgledano učenje se nekada primenjuje za pretprocesiranje podataka u slučaju primene nekog od algoritama nadgledanog učenja.

Učenje potkrepljivanjem je pristup između prethodna dva. Osnovne komponente su agent, akcija, nagrada i politika. Primenjuje se na probleme u kojima agent opaža tekuća stanja i preduzima određene akcije koje su nagrađuju. Cilj učenja je optimalna politika koja preslikava stanja u akcije tako da se na kraju dobije optimalna (što je moguće viša) ukupna nagrada. Potrebno je naglasiti da nije poznato koje akcije su optimalne u određenim stanjima. Da je tako nešto poznato radilo bi se o problemu nadgledanog učenja.

U ovom radu se koriste algoritmi nadgledanog učenja, stoga će u narednom odeljku ukratko biti objašnjen jedan od najznačajnijih algoritama mašinskog učenja.

4.1 Neuronske mreže

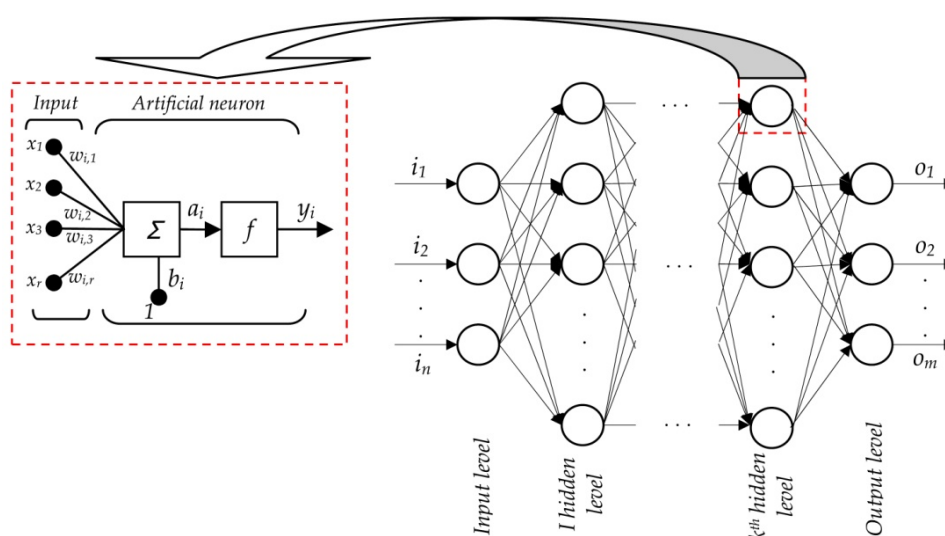
U ovom radu korišćene su neuronske mreže sa propagacijom unapred (eng. *feed-forward neural networks*) te će one biti detaljnije objašnjene. Pored neuronskih mreža sa propagacijom unapred postoje i konvolutivne neuronske mreže, rekurentne neuronske mreže i mnoge druge.

Konvolutivne neuronske mreže se uglavnom koriste za obradu signala. U vreme pisanja rada konvolutivne mreže su apsolutno najkorišćenije u problemima obrade slika. Slike koje su dobijene pomoću mikroskopa su jako velike, dimenzije su im npr.: 1280x1080px, 2048x2048px, dalje, imaju 16b po kanalu tako da njihovo procesiranje i treniranje konvolutivnim neuronskim mrežama zahteva dosta vremena. Neke do standardnih dimenzija koje se koriste kao ulaz u konvolutivnu neuronsku mrežu su 100x100px ili 24x24px. U slučaju autofokusiranja nije sasvim razrešeno i ispitano koliko će skaliranje slika na manje dimenzije doprineti gubitku podataka relevantnih za određivanje fokusa te bi ovaj pristup mogao biti ispitano u nekom od narednih radova. Danas su neuronske mreže [9] jedan od najpopularnijih algoritama mašinskog učenja. Ta popularnost upravo dolazi zbog rezultata konvolutivnih neuronskih mreža.

U nastavku će nešto detaljnije biti objašnjene mreže koje su korišćene u ovom radu.

4.1.1 Neuronske mreže sa propagacijom unapred

U ovom odeljku ukratko će biti objašnjene neuronske mreže sa propagacijom unapred. Da bi se bolje razumeo algoritam korisno je objasniti značenje celog naziva algoritma. Deo *neuronske mreže* proističe iz toga što su razvijane po modelu neuronskih mreža u čovekovom mozgu. Drugi deo *unapred* objašnjava kretanje informacije kroz mrežu. Na slici 4.1 je prikazana arhitektura jedne mreže.



SLIKA 4.1: Primer neuronske mreže.

Svaka neuronska mreža sa propagacijom unapred sastoji se iz neurona i slojeva. Neuroni su matematičke funkcije koje na ulazu primaju linearnu kombinaciju atributa i primenjuju neku aktivacionu funkciju na tu linearnu kombinaciju. Na slici 4.1 neuroni su prikazani krugovima, a sa leve strane postoji uvećanje koje dodatno objašnjava arhitekturu svakog neurona. Aktivacione funkcije su nelinearne funkcije i načešće se koriste:

- Tanh $g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Sigmoid $g(x) = \frac{1}{1 + e^{-x}}$
- ReLU $g(x) = \max(0, x)$

Neuroni se grupišu u slojeve. Svaka neuronska mreža sa propagacijom unapred sastoji se od izlaznog i ulaznog sloja. Ulazni sloj prima atribute, a izlazni se koristi za klasifikaciju ili regresiju (sve zavisi o kom problemu se radi). Ukoliko sem ova dva sloja postoji još jedan ili više skrivenih slojeva tada se radi o *dubokim neuronskim mrežama* (eng. *deep neural network*).

Formulacija modela :

$$h_0 = x$$

$$h_i = g(W_i h_{i-1} + w_{i0}) \quad i = 1, 2, \dots, L$$

Gde je:

x - ulazni vektor (atributi)

L - broj slojeva

W_i - matrica gde red j predstavlja parametar vektora jedinice j u sloju i

w_{i0} - vektor slobodnih članova

g - nelinearna aktivaciona funkcija

Takođe, uspešnost neuronskih mreža je teorijski potkrepljena *teoremom o univerzalnoj aproksimaciji* [10]. Ukratko, ona izražava važno svojstvo neuronskih mreža – da se svaka neprekidna funkcija može proizvoljno dobro aproksimirati neuronskom mrežom sa jednim skrivenim slojem i konačnim brojem neurona. Iako je ova teorema ohrabrujuća za korišćenje neuronskih mreža to nikako ne znači da je lako naći adekvatan model, čak može biti i preteško.

4.2 Regularizacija

Prilikom razmatranja modela bitna osobina je fleksibilnost modela. Za model se smatra da je fleksibilan ako može dobro da se prilagodi podacima. Primer modela koji se obično ne smatra fleksibilnim je linearna regresija. Ako se ovaj model koristi za regresiju njegova mana je to što ne može da opiše nelinearne zavisnosti, pa se u tom slučaju koriste fleksibilniji modeli kao što su neuronske mreže. Preprilagođavanje (eng. *overfitting*) modela se dešava kada se model previše prilagodi podacima. Kada se to desi sposobnost modela da dobro generalizuje je loša. Stoga se fleksibilnost modela mora kontrolisati da ne bi došlo do neželjenih rezultata. Kontrola prilagođavanja modela se vrši regularizacijom. Postoje razni načini regularizacije modela kao što su L1 regularizacija, L2 regularizacija, unutrašnja standardizacija (eng. *batch*

normalization) i mnogi drugi. U ovom radu korišćen je vid regularizacije koji se naziva izostavljanje (eng. *dropout*) [11]. Izostavljanje se odnosi na anuliranje neurona. Pseudo-slučajno izabrani neuroni se izostavljaju tokom treninga i tako se aproksimira ansambl svih neuronskih mreža koje se mogu dobiti od polazne. Verovatnoća anuliranja neurona je metaparametar koji se postavlja za ovaj vid regularizacije.

4.3 Mere kvaliteta modela

U praksi je uvek potrebno kvantifikovati moć generalizacije modela. Kvantifikacija kvaliteta modela se vrši pomoću mera kvaliteta modela. Takođe u slučaju da postoji više modela koji rešavaju isti problem potrebno je na osnovu mera kvaliteta modela izabrati najbolji od njih. U nastavku će biti predstavljene mere koje se koriste za problem regresije. U slučaju da se radi o problemu klasifikacije koriste se druge mere kvaliteta koje neće biti predstavljene u ovom radu.

Definicija 3. *Srednja kvadratna greška* (eng. *mean squared error*) se izračunava po formuli

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2 \quad (4.1)$$

Predstavlja grešku regresije. Kvadriranjem se sprečava potiranje vrednosti različitog znaka i dodatno se penalizuju velike greške.

Definicija 4. *Koren srednje kvadratne greške* (eng. *root mean squared error*) se izračunava po formuli

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2} \quad (4.2)$$

Osnovna prednost je što se može dobiti bolji osećaj o veličini greške u odnosu na srednju kvadratnu grešku.

Definicija 5. *Varijansa (disperzija)* se aproksimira se sledećom formulom:

$$VAR = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_n)^2 \quad (4.3)$$

To je mera raspršenosti podataka, predstavlja prosečno odstupanje od proseka.

Definicija 6. *Koeficijen determinacije* se izračunava sledećom formulom

$$R^2 = 1 - \frac{MSE}{VAR[y]} \quad (4.4)$$

objašnjava udeo objašnjene varijanse. Uzima vrednosti iz intervala $(-\infty, 1]$. Neformalno rečeno poredi se sa najjednostavnijim regresorom a to je prosek. Ako regresor radi jednako dobro kao prosek onda je vrednost koju uzima koeficijent determinacije 0, ako je sva varijansa objašnjena onda uzima vrednost 1. Moguće je i da uzme vrednost manju od 0, to znači da regresor predviđa lošije nego prosek.

Glava 5

Predloženi pristup

Nakon predstavljanja bitnih pojmova koji su korišćeni u daljem radu u ovoj glavi biće predstavljen nov metod autofokusiranja primenom dubokih neuronskih mreža. Biće predstavljen i ceo postupak koji uključuje procesiranje slike, generisanje atributa, treniranje modela i konačnog optimizacionog algoritma kao i tehnologije kojima je to urađeno.

5.1 Priprema atributa

Ovo je vrlo važan i verovatno najzahtevniji deo rada. Postupak razvoja atributa (eng. *feature engineering*) je neizostavni deo prilikom razvoja novih modela primenom algoritama mašinskog učenja (sem u slučaju kada se koriste sirovi podaci, kao što je preporuka u slučaju konvolutivnih neuronskih mreža). Uslov da se atributi kvalitetno razviju je dobro domensko znanje o problemu koji se rešava. U slučaju autofokusa potrebno je na osnovu slike i poznatih rezultata do sada, zaključiti koji su to atributi slike relevantni za problem koji se rešava. Pažljivim odabirom i velikim brojem testova izabrani su određeni atributi za učenje. Empirijski se došlo do zaključka da je za uspešnu pripremu atributa potrebno proći kroz nekoliko faza. Te faze su:

- Procesiranje slika
- Izračunavanje mera fokusiranosti
- Generisanje atributa

Redosled i sadržaj faza je od velike važnosti. Ako bi se neka od celina polovično izvršila ili pak preskočila, u najboljem slučaju rezultati bi bili nešto lošiji od očekivanih. Ipak je verovatnije da pristup ne bi radio, tako da će u nastavku svaka od celina biti detaljnije objašnjena.

5.1.1 Procesiranje slika

Primećeno je da mere fokusiranosti nisu invarijantne na različite nivoe osvetljenosti (eng. *brightness*) slike. Prvi korak procesiranja slika je stavljanje svih slika na isti nivo osvetljenosti.

Budući da se osvetljenost slike može modelovati očekivanom vrednošću piksela, teži se da očekivana vrednost piksela svih slika bude jednaka na svim slikama sa malim odstupanjem. Postavljanje osvetljenosti slike je urađeno tako što se svaki piksel slike pomnoži sa koeficijentom ν koji se dobija iz jednačine 5.1

$$\nu = \frac{N}{E(I)}, N \in [0, b] \quad (5.1)$$

gde je N željena očekivana vrednost, a $E(I)$ očekivana vrednost slike I , a b predstavlja dubinu kanala.

5.1.2 Izračunavanje mera fokusiranosti

U prethodnom odeljku prikazan je pristup procesiranja slike. Tako procesirana slika pripremljena je za dalju obradu. Sledeći korak pri generisanju atributa je izračunavanje mera fokusiranosti koje su predstavljene u odeljku 3.3.

Za sliku za koju se izračunavaju mere fokusiranosti se pravi vektor mera fokusiranosti. Vektor mera fokusiranosti sadrži sledeće mere:

- Sobelov operator (formula 3.4)
- Brennerov operator (formula 3.2)
- Vertikalni Brennerov operator (formula 3.3)
- Kvadrirani Laplasov operator (formula 3.6)
- Furijeova transformacija (formula 3.11)

U jednom od prethodnih odeljaka data je definicija 1, o konvoluciji, koja će biti korišćena u nastavku. Konvolucija se koristi zbog povećanja efikasnosti izračunavanja mera fokusiranosti i to na sledeći način. Svaka od mera fokusiranosti može se predstaviti filterom F npr. za kvadrirani gradijent filter dimenzija 3×1 izgleda:

$$F = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Primenom konvolucije filtera F na sliku I dobija se rezultat R . Rezultat R je nova matrica (slika). Potrebno je kvadrirati piksele matrice R a nakon toga ih sumirati. Suma svih piksela predstavlja meru fokusiranosti. Ovakav pristup je efikasniji jer postoje optimizovani načini konvolucije koji su implementirani u Python [12] biblioteci OpenCV [13]. Analogno ovom primeru sve ostale mere mogu se predstaviti filterima i konvolucijom se dobiti efikasniji pristup njihovom izračunavanju.

Rezultat ove faze je vektor brojeva koji opisuju fokusiranost slike. Svaki element tog vektora je vrednost razičite mere fokusiranosti. Pored mera fokusiranosti vektor sadrži polje koje opisuje relativnu poziciju z -pozicije mikroskopa. Pozicija je bitna informacija kako bi se tokom treninga odredila udaljenost trenutne pozicije z -pozicije od idealne pozicije z -pozicije.

5.1.3 Generisanje atributa

Dobijeni vektor mera fokusiranosti iz prethodne faze sadrži šest elemenata (pet elemenata su mere fokusiranosti i šesti element je z -pozicija). Na početku je bitno napomenuti da trenutna pozicija z -pozicije nikako ne sme da se koristi prilikom učenja. To je uzrokovano strukturom skupa podataka. Sasvim slučajno se desilo da na svim podacima iz skupa podataka pozicija idealne fokusiranosti odstupa vrlo malo. Zbog ove činjenice moglo bi se desiti da model stvori lažnu sliku da je moguće samo u određenom opsegu dostići željeni cilj. To bi dovelo do loše generalizacije, te se ova naizgled korisna informacija odbacuje.

Broj predstavljenih mera fokusiranosti pogodnih za mikroskopiju je 12. Ako bi bile korišćene sve moguće mere kao i svi mogući odnosi mera fokusiranosti postojalo bi 12^2 atributa a to je 144 atributa. Važna težnja prilikom razvoja bilo kog od

modela mašinskog učenja je briga o dimenzionalnosti podataka. Važan fenomen koji jako dobro opisuje bitnost smanjenja dimenzionalnosti podataka je prokletstvo dimenzionalnosti [14] (eng. *curse of dimensionality*). Postavlja se pitanje da li je moguće smanjiti broj atributa? Lako je uočiti da se na početku odmah eliminišu atributi koji su dobijeni količnikom iste mere fokusiranosti. Kada se to eliminiše ostaju 132 atributa. Dodatno smanjenje dimenzionalnosti atributa je urađeno primenom alata WEKA [15]. To je alat koji na osnovu međusobne korelacije atributa i korelacije atributa sa ciljnom promenljivom bira neophodne atribute. U ovom slučaju alat je izabrao deset odnosa mera fokusiranosti.

U ovom radu je korišćeno 10 atributa. Pet atributa su sirove mere fokusiranosti, a ostalih 5 su kombinacije ovih mera fokusiranosti.

Atributi:

- Sobelov operator (formula 3.4)
- Brenerova mera (formula 3.2)
- Vertikalna Brenerova mera (formula 3.3)
- Kvadratni Laplasov operator (formula 3.6)
- Furijeova transformacija (formula 3.11)

- $\frac{\text{Brenerova mera}}{\text{Kvadratni Laplasov operator}}$

- $\frac{\text{Vertikalna Brenerova mera}}{\text{Kvadratni Laplasov operator}}$

- $\frac{\text{Kvadratni Laplasov operator}}{\text{Vertikalna Brenerova mera}}$

- $\frac{\text{Kvadratni Laplasov operator}}{\text{Furijeova transformacija}}$

- $\frac{\text{Furijeova transformacija}}{\text{Sobelov operator}}$

- $\frac{\text{Furijeova transformacija}}{\text{Kvadratni Laplasov operator}}$

Ovaj skup atributa predstavljaće osnovu za treniranje modela.

5.2 Modeli

Na ovom mestu biće predstavljeni modeli koji su razvijeni u radu. Modeli će biti predstavljeni samostalno kao celine. Moguće je da razlozi za neke izbore trenutno neće biti potpuno jasni ali zbog dekompozicije rada sve nejasnoće biće potpuno razjašnjene u odeljku 5.3.

Radi boljeg razumevanja sada će biti predstavljeni pojmovi koji će biti korišćeni u daljem radu.

- *z-pozicija* - deo mikroskopa koji se pomera pri fokusiranju. Obično se ovaj deo mikroskopa pomera vertikalno.
- *trenutna pozicija mikroskopa* (z_c) - trenutna pozicija z-pozicije izražena u mikrometrima (μm).

- *idealna pozicija z-pozicije* (z_i) - je željena pozicija z-pozicije. Kada se dostigne ova pozicija mikroskop je fokusiran i slika koja se dobija pomoću mikroskopa je najbolje fokusirana slika.
- *strana fokusa* - može biti leva ili desna u odnosu na idealnu poziciju mikroskopa. Na slici 3.1 z_i je oko $3432\mu m$. Leva strana fokusa (s_l) je svaka vrednost z-pozicije manja od z_i , dok je desna strana (s_r) svaka pozicija veća od z_i .

U sekciji 4.1 su predstavljene neuronske mreže. Nadalje će biti predstavljena dva modela neuronskih mreža koja se primenjuju tokom autofokusiranja.

Modeli:

- Model koji koristi isključivo jednu sliku (M_1) za predviđanje idealne pozicije z_i
- Model koji koristi dve slike (M_2) za predviđanje idealne pozicije z_i

U nastavku ce prvo biti objašnjen M_1 model, a nakon toga i model M_2 .

Model koji koristi jednu sliku (M_1)

Kao što je navedeno M_1 koristi jednu sliku, preciznije govoreći koristi attribute dobijene iz te slike.

Atributi koje koristi ovaj model su predstavljeni u odeljku 5.1.3, tako da M_1 na ulazu ima 10 atributa.

Ciljna promenljiva (y) je apsolutna razlika između trenutne pozicije mikroskopa i idealne pozicije mikroskopa.

$$y = |z_c - z_i| \quad (5.2)$$

Arhitektura modela je predstavljena u tabeli 5.1.

Sloj	Broj neurona	Aktivaciona funkcija	Regularizacija
1	10	ReLU	0.2
2	20	ReLU	0.2
3	1	Linear	/
Σ	3	50	/

TABELA 5.1: Arhitektura modela koji koristi jednu sliku.

Sada će biti predstavljen primer upotrebe ovog modela. Pretpostaviti da je z-pozicija mikroskopa na poziciji z_c . Prvi korak je predikcija rastojanja, $d = M_1(z_c)$. Poznate su dve stvari z_c i d što je apsolutno rastojanje do z_i . Postavlja se pitanje, koja je idealna vrednost z-pozicije? Da li je to $z_c + d$ ili je to $z_c - d$? Ovaj nedeterminizam je osnovni nedostatak modela koji koristi jednu sliku za predviđanje pozicije idealne fokusiranosti. Generalno nije poznato na koji način odrediti u kom smeru se nalazi idealna pozicija. Zbog ovog problema pribegava se pseudo-slučajnom biranju da li je $z_i = z_c + d$ ili $z_i = z_c - d$. Stoga, da bi postojala informacija o smeru idealne pozicije z-pozicije moraju se uzeti najmanje dve slike.

Jedan od centralnih problema je što nema kriterijuma za proveru fokusiranosti slike. Potrebno je razviti klasifikator koji klasifikuje slike na fokusirane i nefokusirane. No, razvoj ovakvog modela je van domašaja ovog rada.

U tabeli 5.2 su prikazane vrednosti evaluacije modela sa jednom slikom.

Metrika	Vrednosti
RMSE	2.48 μm
R^2	0.89
MaAE ¹	9.12 μm
MiAE ²	0.00 μm

TABELA 5.2: Evaluacija modela koji koristi jednu sliku.

Budući da su rezultati u nekom smislu dovoljno dobri postavlja se pitanje zašto stvarno nisu dobri. Prvi problem je nemogućnost određivanja smera kretanja. Ako bi se uzele dve slike onda automatski postoji informacija o smeru ispravnog kretanja. Problem u ovom slučaju je nešto veći broj akvizicija. Drugi problem je ne tako dobra aproksimacija u okolini ciljne promenljive, to je prikazano na slici 5.1. Na slici su prikazana predviđanja na četiri skupa podataka. Predviđanja nikako nisu stvarno označena već je levo od nule veštački dodat predznak i sve to u cilju jasnijeg prikaza dijagrama. Crvenom bojom je predstavljeno očekivano predviđanje, tj stvarna vrednost, a plavom bojom predviđena vrednost. Očigledno je da predviđanje kada priđe idealnoj poziciji nije dobro. Baš u trenutku kada se dostigne blizina idealne pozicije model značajno poveća grešku, samim tim ne postoji konvergencija. Bitno je razjasniti da ne postoji konvergencija samo u slučaju modela koji koristi jednu sliku. U toku istraživanja nije pronađen dovoljno dobar odgovor na ovakvo ponašanje.

Model koji koristi dve slike (M_2)

Ovaj model na ulazu uzima dve slike na pozicijama z_{c_0} i z_{c_1} za koje važi uslov $z_{c_0} < z_{c_1}$. Jedna važna osobina ovog modela je da rastojanje između slika može biti proizvoljno.

Atributi koje koristi ovaj model su predstavljeni u odeljku 5.1.3. Model na ulazu dobija atribute iz obe slike, tako da je to dvadeset atributa. Međutim, ovaj model ima još jedan dodatni atribut - dodatni atribut je rastojanje po z osi između ove dve slike, tako da sve ukupno iznosi dvadesetjedan atribut.

Ciljna promenljiva (y) je rastojanje između trenutne slike koja ima manje z rastojanje z_{c_0} u odnosu na idealnu poziciju mikroskopa z_i .

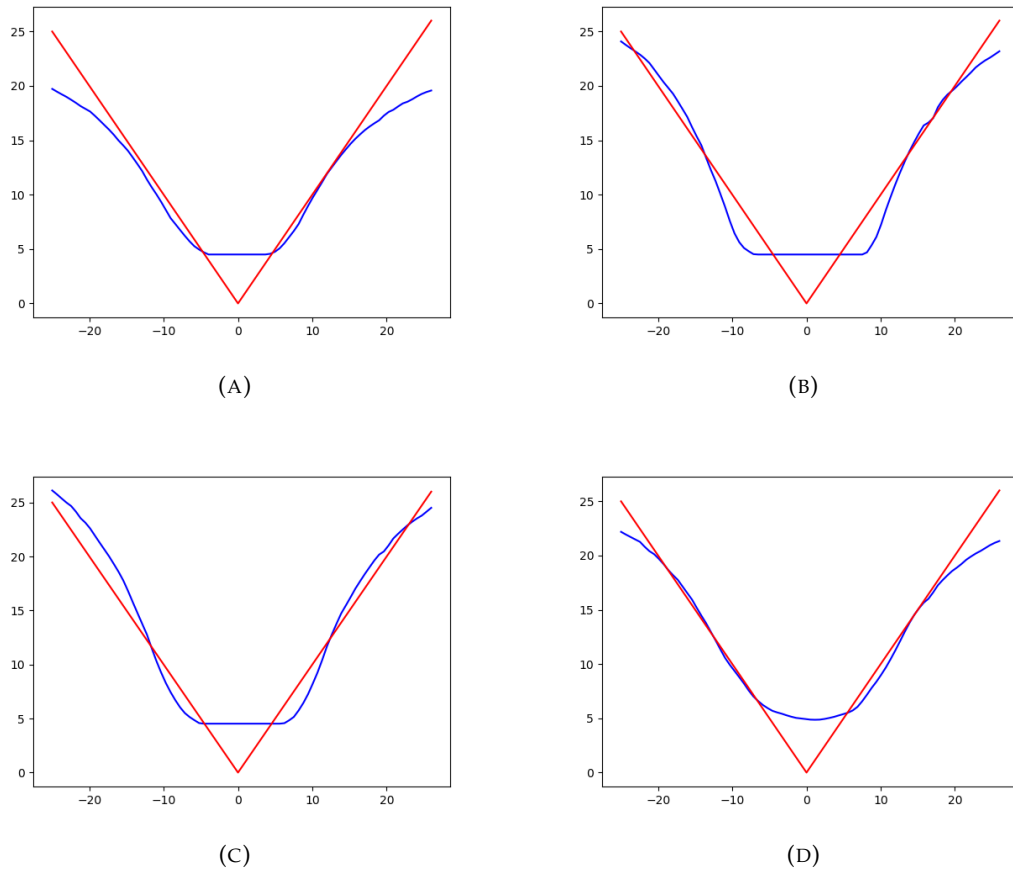
$$y = z_{c_0} - z_i \quad (5.3)$$

Arhitektura neuronske mreže za M_2 model predstavljena je u tabeli 5.3.

Postavlja se pitanje, da li je redosled slika bitan? Naravno, model je treniran tako da redosled slika uvek bude ovakav, kao i da uslov koji je gore pomenut bude uvek

¹Maksimalna apsolutna greška.

²Minimalna apsolutna greška.



SLIKA 5.1: Primeri predviđanja sa modelom koji koristi jednu sliku.

ispunjen. Treniranje modela tako da nauči simetriju i da mu ne bude bitan redosled slika nije uspelo. Rezultati su bili znatno lošiji od rezultata dobijenih na predstavljenom modelu.

U tabeli 5.4 prikazana je evaluacija modela koji koristi dve slike. Iako na prvi pogled rezultati izgledaju lošije u odnosu na prethodni model, treba uzeti u obzir da je ciljna promenljiva značajno drugačija od ciljne promenljive modela sa jednom slikom. Dodatno, model sa dve slike nema unapred određeno rastojanje koje mora biti između slika da bi mogao da bude primenjen. Ulaz u model mogu biti slike na proizvoljnom rastojanju. To je značajan uslov za korišćenje ovog modela jer je primenljiv uvek, ali isto tako to dosta otežava njegovo treniranje i očekivano je da rezultati budu malo lošiji. Sve ovo će biti detaljnije diskutovano u nastavku. Detaljnije razmatranje rezultata M_2 modela i generisanja grafika poput grafika 5.1 nije moguće. To nije lako zbog korišćenja dve slike. Slabost ovog modela je ne tako mala greška fokusiranja.

Budući da oba modela imaju svoje slabosti koje utiču na performanse autofokusiranja potrebno je na neki način ublažiti nedostatke ili ih u najboljem slučaju eliminisati. U sledećem odeljku biće predstavljen optimizacioni algoritam koji ima za cilj da od oba modela izvuče ono najbolje, a primenom algoritamskog znanja autofokus dostigne najbolje moguće performanse.

Slojevi	Broj neurona	Aktivacije	Regularizacija
1	21	Relu	0.2
2	80	Relu	0.2
3	80	Relu	0.2
4	1	Linear	
Σ	/	182	/

TABELA 5.3: Arhitektura M_2 modela.

Metrika	Vrednosti
RMSE	$2.82 \mu m$
R^2	0.95
MaAE	$16.17 \mu m$
MiAE	$0.00 \mu m$

TABELA 5.4: Evaluacija modela koji koristi dve slike.

5.3 Optimizacioni metod

U ovoj glavi biće detaljno predstavljen optimizacioni metod. Početak istraživanja ove teme bio je ambiciozan. Prvobitni cilj je bio da se razvije model koji koristeći samo jednu sliku predviđa poziciju idealnog fokusa. To za sada nije uspelo zbog razloga navedenih u glavi 5.2. Nakon tog neuspeha sledeći korak je da se razvije model koji koristeći dve slike predviđa idealnu poziciju z-pozicije. To je pošlo za rukom i dobili su se bolji rezultati. Međutim, zašto odbaciti model koji koristi jednu sliku? U nekim slučajevima može da doprinese tačnošću predviđanja. Glavna ideja je da se na sve slike primene oba modela i to na način koji će biti objašnjen u nastavku.

Početna pozicija (z_1) sa koje se primenjuje algoritam je pozicija na kojoj se mikroskop zatekao prilikom primene, znači ne postoji nikakva informacija o početnoj poziciji. Može se smatrati da je ona pseudo-slučajno generisana. Sa početne pozicije se uzima slika koja se propušta kroz model koji koristi jednu sliku i predviđa udaljenost (d) trenutne pozicije mikroskopa od idealne pozicije mikroskopa. Budući da je već predstavljeno da je nemoguće znati smer u kom se treba kretati na ovom mestu se "baca novčić" i pseudo-slučajno se bira smer kretanja sa verovatnoćom 0.5 za oba smera. Sledeća pozicija mikroskopa je $z_2 = z_1 \pm p$ (smer je određen znakom). Sada na red dolazi upotreba modela koji koristi dve slike. Na ulaz modela sa dve slike dovode se slike z_1 i z_2 i predviđa se z_3 . U svakom narednom koraku algoritma izračunava se nova tačka uprosečavanjem svih tačaka koje se dobijaju kada se model sa dve slike primeni na sve parove do tada izračunatih tačaka, a model sa jednom slikom primeni na sve pojedinačne tačke, ali birajući tada već poznati smer pozicije fokusa. Uslov zaustavljanja jeste uslov da razdaljina poslednja dva kandidata bude manja od unapred zadate tolerancije. I tako se algoritam primenjuje iterativno dok uslovi zaustavljanja nisu ispunjeni. Ideja je da svaka slika koja se uzme bude maksimalno iskorišćena, tako da model sa jednom slikom predviđa udaljenost od z_i , tako da i svaka slika bude uparena sa ostalim slikama i dovedena na ulaz modela koji predviđa sa dve slike. Kao što je navedeno uslov zaustavljanja je tačnost, tj. razlika

između susednih pozicija. Još jedan od uslova je broj akvizicija. Ako se stavi prevelika tačnost može se ograničiti broj akvizicija.

Pseudo kod optimizacionog metoda, tj algoritam predstavljen je na algoritam 2.

Algoritam 2: Optimizacioni algoritam autofokusiranja.

```

Input:  $c_p$  - trenutna pozicija,
          tol - tolerancija,
          MAX_AKVIZICIJE - broj dozvoljenih akvizicija,
           $M_1$  - model koji koristi jednu sliku,
           $M_2$  - model koji koristi dve slike
Output:  $\hat{f}_i$  predviđena pozicija idealne vrednosti z-pozicije
           koraci - broj akvizicija

1 predikcije = [ ]          /* sve predviđene pozicije tokom optimizacije */
2 prva_predikcija =  $M_1(c_p)$ 
3 if bacanje_novčića( ) > 0.5 then
4 |   smer = -1
5 else
6 |   smer = 1
7 predikcije.append( $c_p + smer * prva\_predikcija$ )
8 while len(predikcije) < MAX_AKVIZICIJE do
9 |   pozicije = [ ]          /* sve pozicije dobijene od oba modela */
10 |   n = length(predikcije)
11 |   foreach  $p1, p2$  from predikcije do
12 |   |   pozicije.append( $M_2(p1, p2)$ )
13 |   trenutna_predikcija = average(pozicije)
14 |   foreach  $p$  from predikcije do
15 |   |   rastojanje =  $M_1(p)$ 
16 |   |   if abs(trenutna_predikcija - (p + rastojanje)) < abs(trenutna_predikcija - (p -
17 |   |   |   rastojanje)) then
18 |   |   |   pozicije.append( $p + rastojanje$ )
19 |   |   |   else
20 |   |   |   pozicije.append( $p - rastojanje$ )
21 |   |   predikcije.append(average(pozicije))
22 |   |   if razlika između poslednje dve predikcije < tol then
23 |   |   |   koraci = len(predikcije)
24 |   |   |   return pop(predikcije), koraci

```

5.4 Podaci, treniranje i evaluacija modela

U ovom poglavlju biće predstavljen skup podataka kao i način treniranja modela. Softver je programiran u programskom jeziku *Python* [12]. Neuronske mreže su implementirane koristeći biblioteku *Keras* [16]. Trening je izvršavan na računaru sledećih karakteristika: CPU: Intel I7 7700HQ, GPU: GeForce GTX 950M, RAM: 8GB. Vreme potrebno za treniranje modela koji koristi jednu sliku je oko 1 minut, a vreme

koje je potrebno za treniranje modela koji koristi dve slike je oko 10 minuta. Ovako kratko vreme treninga je možda pomalo neočekivano, ali postoji razumno objašnjenje za to. Kratko vreme treniranja je posledica procesiranja slika i pripreme atributa pre treniranja. Naime iz svih slika su izračunati atributi i zapisani u tekstualnu datoteku, pa se prilikom treninga atributi čitaju iz tekstualnih datoteka i na taj način se značajno skraćuje vreme treninga.

5.4.1 Skup podataka

Skup podataka je skup slika različitog nivoa fokusiranosti. U skupu podataka postoji tridesetšest *stekova slika*. Stek slika predstavlja jednu sliku sa istim sadržajem ali različitim nivoom fokusiranosti. Svaki stek slika sadrži sto različitih slika. U daljem tekstu pod različitim slikama biće podrazumevane slike sa istim sadržajem ali različitim nivoom fokusiranosti. O svakom steku postoje i metapodaci npr.: x , y ili z -pozicija stočića, brzina uzimanja slika, jačina svetla itd. Od svih metapodataka jedini podatak koji se koristi je podatak o poziciji z -pozicije mikroskopa. Ispostavlja se da u podacima nad kojim su trenirani modeli u ovom radu vrednost idealne pozicije z -pozicije odstupa sa standardnom devijacijom od 5 mikrona, tako da ne postoji raznovrsnost u opsegu vrednosti z -pozicije. Zbog toga je izbegnuto da kao jedan od atributa učenja bude pozicija stočića. Model bi mogao da iz toga nauči da uvek predviđa neki broj iz tog opsega što u opštem slučaju ne bi dalo smislene rezultate.

5.4.2 Priprema skupa podataka

Prethodno je pomenuta konfiguracija skupa podataka. Radi podsećanja, skup podataka se sastoji iz 17 stekova slika. Svaki stek slika sadrži 100 slika različitog nivoa fokusiranosti na ekvidistantnim pozicijama. Distanca između svake slike je $0.5 \mu\text{m}$. Distanca između slika se ogleda u distanci između vrednosti koje uzima z -pozicija mikroskopa dok se kreće vertikalno.

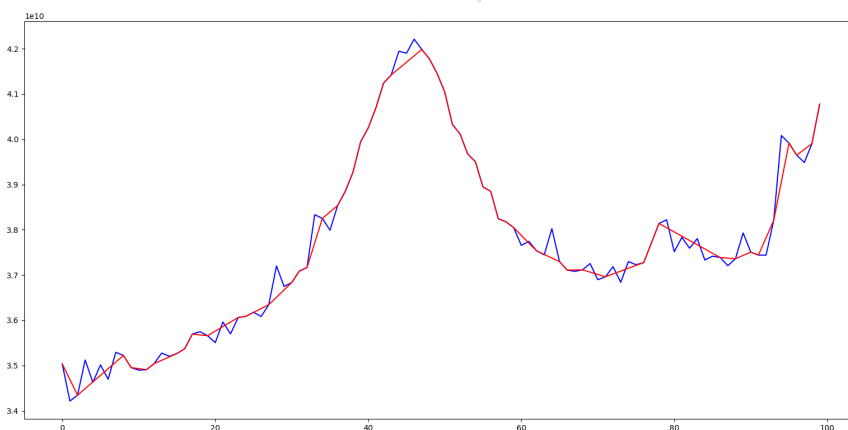
Priprema podataka zahteva 4 koraka. Ovi koraci su objašnjeni u odeljku 5.1, te će ovde biti samo nabrojani:

1. Procesiranje slika
2. Izračunavanje mera fokusiranosti
3. Uglašavanje mera fokusiranosti

Jedna od mera fokusiranosti je predstavljena na slici 3.1. U određenim stekovima slika postoji šum. Ustanovljeno je da postoji greška na hardveru te se pristupilo sređivanju podataka uglašavanjem mera fokusiranosti. Da bi se izbacio šum ili bar deo šuma korišćena je Python funkcija [17] za detekciju lokalnih ekstremuma. Detektovani lokalni ekstremumi su kasnije izbačeni a na tim mestima je mera fokusiranosti interpolirana. Na slici 5.2 prikazan je efekat uglašavanja.

4. Generisanje atributa

Sledeći korak je izračunavanje idealne pozicije fokusiranosti. Zanimljivo je da različite mere fokusiranosti daju različite idealne pozicije fokusiranosti, dakle postoji odstupanje u poziciji maksimuma za različite mere fokusiranosti. Na slici 5.3 se može videti ovaj fenomen. Odstupanje je do $10 \mu\text{m}$, što u nekim slučajevima može predstavljati značajnu razliku u kvalitetu fokusiranosti slike. Nije sasvim jasno da li



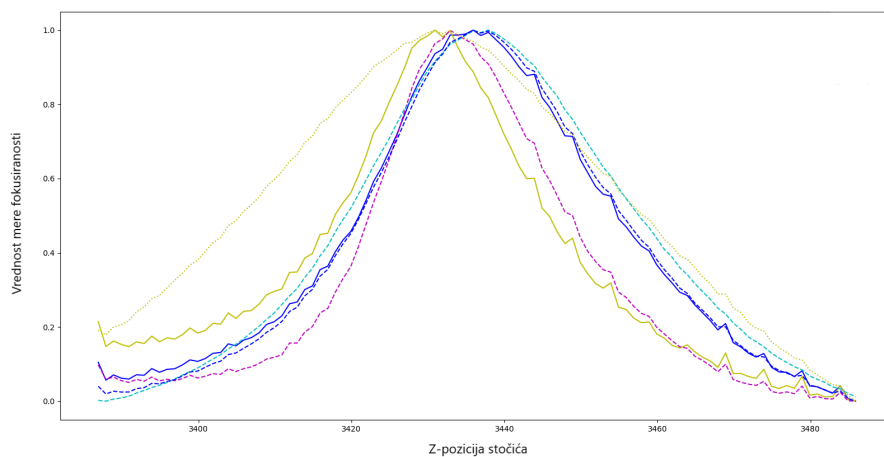
SLIKA 5.2: Uglučana mera fokusiranosti.

je neka mera bolja od druge, ne postoji nikakav rad o tome, te se u ovom slučaju pristupa uprosečavanju svih mera fokusiranosti. Tako da se idealna pozicija z-pozicije računa po formuli 5.4

$$z_i = \frac{1}{n} \sum_{k=1}^n \operatorname{argmax}_x \phi_k(x) \quad (5.4)$$

Gde je:

- n - broj mera fokusiranosti
- m - broj slika po steku
- $\phi_k(x)$ - mera fokusiranosti



SLIKA 5.3: Primer različitih globalnih ekstremuma kod različitih mera fokusiranosti.

5.4.3 Treniranje i evaluacija modela

Kada želimo da primenimo algoritme mašinskog učenja za rešavanje nekog problema ključni deo jeste način treniranja i evaluacija modela. Na ovom mestu će biti definisana neka od bitnih načela prilikom treniranja modela:

1. **Trening i test skup prilikom treniranja modela moraju biti disjunktни.**
Ovo je jedno od osnovnih pravila da bi evaluacija bila uspešna.
2. **Varijansa ciljnih promenljivih na trening i test skupu mora biti ista.**
Ovo je ispunjeno tako što su slike izbalansirane tako da postoji jednak broj svih različitih udaljenosti koje se predviđaju. Ovo je postignuto tako što je iz svakog steka slika uzet jednak broj slika simetrično od pozicije idealnog fokusa.
3. **Svi očekivani slučajevi upotrebe modela bi trebalo da se nađu u skupu podataka.**
Ovo načelo znači da se od modela ne mogu očekivati dobri rezultati ako u skupu podataka ne postoje slučajevi upotrebe koji značajno odstupaju od svega viđenog u skupu podataka.
4. **Trening i test skup moraju biti organizovani na isti način na koji se podaci nalaze u stvarnoj upotrebi.**
Prirodne jedinice organizacije podataka u ovom problemu nisu pojedinačne slike, već uzorci, odnosno skupovi slika koji odgovaraju jednom uzorku. Stoga je prilikom evaluacije modela deljenje podataka na trening i test skup potrebno izvršiti nasumičnim razvrstavanjem takvih skupova slika, a ne nasumičnim razdvajanjem pojedinačnih slika. Na ovaj način se simulira primena autofokuseriranja na potpuno nepoznatom uzorku.

Treniranje oba predstavljena modela se značajno razlikuje u načinu korišćenja slika iz stekova. Ovde će biti objašnjena oba pristupa.

Model koji koristi jednu sliku je nešto jednostavniji u odnosu na model koji koristi dve slike. U toku treninga ovaj model se kreće duž opsega z-pozicija i za svaku uzetu sliku predviđa apsolutno rastojanje od idealne pozicije z-pozicije. Kada se trenira ovaj model ulaz je niz uređenih parova slika i apsolutnog rastojanja od idealne pozicije z-pozicije.

Model koji koristi dve slike je za nijansu komplikovaniji. Budući da je data velika fleksibilnost (rastojanje između slika je proizvoljno) prilikom treninga zahteva se učenje i zavisnosti između rastojanja i atributa slika. Ulaz prilikom treninga u ovom slučaju je uređena trojka dve slike i rastojanje između njih. Skup slika se dobija generisanjem skupa svih parova slika duž intervala z-pozicije, pod uslovom sa početka odeljka 5.2 u kom je naveden neophodan uslov za ovaj model.

Isprobana su tri tipa izbora modela:

1. *Trening i test skup* - tako što su svi stekovi slika spojeni u jedan veliki skup slika, pa se taj skup pseudo-slučajno izmeša i podeli na trening i test skup u odnosu 70% - 30%. Veći deo podataka je skup za treniranje a manji deo podataka je skup za testiranje, odnosno procenu generalizacije modela. Model se trenira na skupu za treniranje, a evaluira na skupu za testiranje.

2. *Unakrsna validacija* - tako što su svi stekovi slika spojeni u jedan veliki skup slika. Dalje se taj skup pseudo-slučajno izmeša i tako dobijeni skup se posle koristi u evaluaciji modela na sledeći način. Skup podataka se izdela na n podskupova, jedan skup instanci se izvuče iz celog skupa d_g , treniranje se odvija na ostatku skupa d_t . Kada se treniranje završi evaluacija se vrši na izdvojenom skupu d_g . Ovaj proces se ponavlja sve dok se ne prođe kroz ceo skup podataka.
3. *Unakrsna validacija po steku slika* - iz skupa stekova izdvoji se jedan stek slika s_g , na ostalim stekovima s_t se model trenira. Kada se proces treniranja završi evaluacija modela se vrši na odvojenom steku. I tako za sve stekove iz skupa podataka.

Prva dva pristupa nisu dala dobru ocenu generalizacije, u toku evaluacije postignuti su optimistični rezultati. Model koji predviđa koristeći jednu sliku imao je rezultate na test skupu $RMSE \approx 1.5 \mu m$ a $R^2 \approx 97$. Rezultati za model koji koristi dve slike bili su još bolji. Kao što se može pretpostaviti ti rezultati su previše optimistični. Nije sasvim objašnjeno odakle dolazi ta optimističnost, ali postoji intuitivno objašnjenje. Objašnjenje je da model ima neku informaciju o skupu podataka koji će biti na test skupu. Ovo nije nikako preklapanje trening i test skupa ali intuitivno se tako može protumačiti. U trenutku kada se stekovi svih slika izmešaju u jedan skup, dolazi do implicitnog rasipanja informacija o slikama iz test skupa u trening skup. Slike iz svakog steka slika imaju neku zajedničku karakteristiku koja je karakteristična baš za taj skup. Ako je bar jedna slika viđena na trening skupu model ima tu informaciju i unapred zna šta ima u test skupu. Ovo je upravo deo koji je objašnjen načelom 4.

Poslednji, treći pristup daje najveći nivo poverenja jer je njegova formulacija najopreznija po pitanju preklapanja trening i test skupa. Ovaj način upotrebe realistično odgovara budućoj primeni jer će prilikom primene imati nov uzorak kakav nije imao u podacima za treniranje. Naravno, ni u prva dva pristupa nije se desilo da se preklape trening i test skup, ali gore navedeni fenomen opisuje u čemu je problem. Primer bi mogao biti u tome da novi skup podataka na kom se evaluira ima potpuno različitu teksturu ili različitu količinu sadržaja na slici. Detaljnije, to se može shvatiti da će slika biti nekakav ekstrem koji odstupa od većine već viđenih slika. Ovim načinom treniranja navedeni problem će biti uključen u evaluaciju.

Optimizacija neuronskih mreža vršena je algoritmom ADAM [18]. Isprobano je nekoliko arhitektura neuronskih mreža. U tabeli 5.1 je prikazana arhitektura modela sa jednom slikom. Povećanjem broja slojeva performanse modela se nisu značajno povećavale. Sa dodavanjem jednog sloja od 20 neurona i istim aktivacionim funkcijama greška fokusa je smanjena za oko $0.02 \mu m$. Vrlo malo smanjenje greške nije dovoljan razlog da se značajno poveća broj parametara modela a samim tim i fleksibilnost modela. Zbog prethodnog razloga je odlučeno da se ostane na jednostavnijoj arhitekturi.

U tabeli 5.3 je prikazana arhitektura modela koji koristi dve slike. Smanjenjem broja skrivenih slojeva, npr. ako se izbacij sloj 3 greška fokusiranja je $3.04 \mu m$, što je značajno lošije nego u izabranom modelu. Pri dodavanju još 2 sloja, prvu veličine 80 neurona, drugi od 40 neurona sa istim aktivacionim funkcijama greška fokusiranja se smanji na $2.69 \mu m$. Međutim tokom evaluacije metode optimizacije autofokusiranja ovaj model nije doprineo poboljšanju rezultata, te je izabran model sa manje skrivenih slojeva iz istog razloga kao kod modela sa jednom slikom.

5.5 Rezultati i poređenje sa ostalim pristupima

U ovom odeljku biće predstavljeni rezultati koji su postignuti kao i poređenja sa tradicionalnim pristupima predstavljenim u glavi 3.4.1.

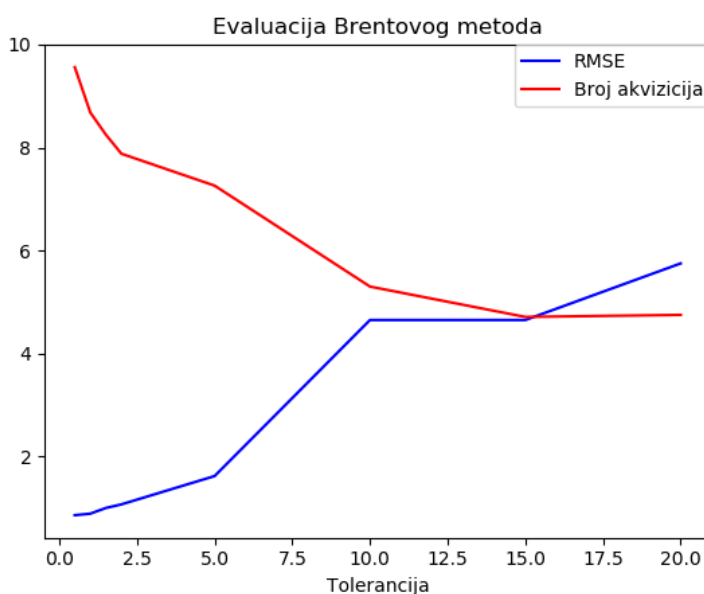
Da bi rezultati poređenja i metodi autofokusiranja mogli da se porede, moraju se postaviti određena pravila, tako da se svaki pristup predstavi u najboljem svetlu. Cilj ovog rada je da se istovremeno minimizuje greška autofokusa kao i broj akvizicija. Budući da tradicionalne metode ne rade to, već se oslanjaju na minimizaciju jednog ili na minimizaciju drugog, fer bi bilo prikazati ih u svakom svetlu i tek posle toga doneti sud o uspešnosti novog modela.

Brentov metod za autofokus ima veoma jake preduslove za primenu. Prema tome rezultati ovog algoritma moraju se predstaviti u dva oblika:

1. **nije obezbeđen uslov primene** - algoritam se pušta da radi, s tim da se svaki put kada se desi da uslov nije ispunjen računa kao neuspešno autofokusiranje
2. **obezbeđen uslov primene** - algoritam se pušta da radi samo u slučaju kada je sigurno da neće doći do neuspeha

U praksi se ne dopušta prvi slučaj već se ulaže dodatni trud da se stanje mikroskopa dovede do pogodnog za primenu Brentovog algoritma. Ovde će biti razmatran samo drugi oblik, kada je sigurno da će do uspeha doći. Novi metod autofokusa nema ovih problema. On nema nikakve ulazne uslove.

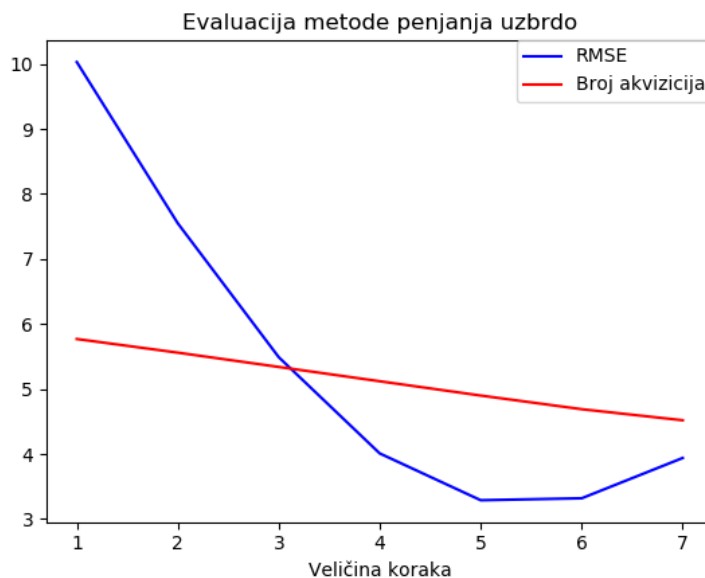
Radi podsećanja, jedan od kriterijuma zaustavljanja Brentovog algoritma je razlika između susednih tačaka. Na slici 5.4 je prikazano kako variranje tačnosti ovog kriterijuma zaustavljanja utiče na broj akvizicija i na odstupanje od idealne vrednosti z-pozicije.



SLIKA 5.4: Evaluacija Brentovog metoda.

Algoritam penjanja uzbrdo ima metaparametar α koji može da bude fiksiran ali može i da bude adaptibilan. Zbog obima posla koji je potreban da bi se kvalitetno

isprogramirao oblik adaptibilnog načina taj način neće biti prikazan. Drugi, mnogo bitniji razlog je što taj pristup zahteva izračunavanje gradijenata u tačkama a to je u ovom slučaju nedopustivo skupo. Na slici 5.5 je prikazano kako variranje α parametra utiče na broj akvizicija i na odstupanje od idealne z-pozicije.



SLIKA 5.5: Evaluacija metode penjanja uzbrdo.

Iz priloženog se može uočiti da proizvoljnim variranjem vrednosti kriterijuma zaustavljanja ili pak vrednosti metaparametara metodi ne mogu da dostignu željene rezultate optimizacije greške autofokusiranja i broja akvizicija. U slučaju Brentovog metoda smanjenjem greške autofokusiranja povećava se broj akvizicija i obratno, dok u slučaju metoda penjanja uzbrdo greška autofokusiranja nikad ne dostiže dovoljno dobre rezultate u poređenju sa novim predloženim metodom.

U tabeli 5.5 je prikazano poređenje tradicionalnih pristupa sa novim metodom. Za tradicionalne metode su izabrani najbolji mogući rezultati koje oni mogu da postignu uzimajući u obzir oba parametra (RMSE i broj akvizicija). Ovde je uzeto u obzir da Brentov metod može da ima problem i da ne dođe do pozicije fokusiranosti. U praksi se ovaj problem rešava dovodenjem mikroskopa u pogodno stanje za primenu metoda ali to dodatno povećava broj akvizicija te je ovo način da realnije uporedimo pristupe. U tabeli pored metode stoji i numerička vrednost izražena u μm koja predstavlja opseg u kom metod vrši pretragu. Algoritam penjanja uzbrdo je evaluiran sa optimalnim α , na ovom skupu podataka, tako da dobije optimalne željene rezultate.

Iz tabele se može videti da je predstavljeni metod postigao manju grešku autofokusa (RMSE) u značajno manjem broju koraka i bez neuspelih pokušaja.

Metod	RMSE	Akvizicije	Neuspesi
Brentov metod $\pm 25\mu m$	1.34	6	64 %
Predloženi metod	1.69	3.5	0 %
Brentov metod $\pm 35\mu m$	1.86	6	17 %
Brentov metod $\pm 45\mu m$	2.37	6	0 %
Penjanje uzbrdo	2.87	5.11	0 %

TABELA 5.5: Poređenje metoda.

Glava 6

Softver

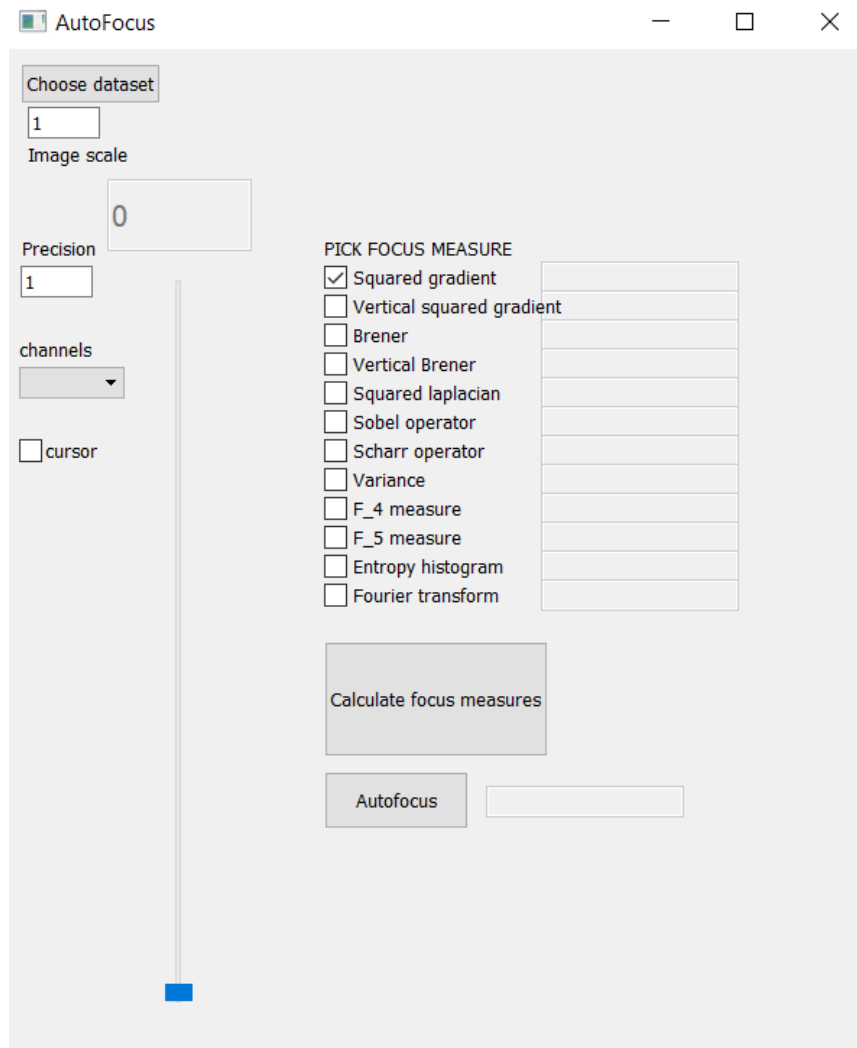
U ovoj glavi će ukratko biti predstavljen softver koji je razvijen za potrebe ovog istraživanja. Interna implementacija komponenti neće biti predstavljena zbog obima rada jer bi se znatno povećao broj strana. Celokupan softver je napisano u programskom jeziku Python [12] i ima oko 6500 linija koda. U taj kod se računa sav prpratni softver koji je posredno vezan sa razvojem modela. Jedan od prpratnih alata je *simulator mikroskopa* i biće predstavljen u narednom odeljku. Za potrebe razvoja modela mašinskog učenja je razvijen radni okvir (eng. *framework*) i biće ukratko predstavljen u odeljku 6.2.

6.1 Simulator mikroskopa

U toku istraživanja napravljen je prpratni softver koji ima za cilj olakšavanje istraživanja. Jedan od tih softvera je simulator mikroskopa koji je implementiran po uzoru na prave elektronske mikroskope. Na slici 6.1 je prikazan početni ekran simulatora mikroskopa.

Dugme *choose dataset* služi za učitavanje skupa podataka, odnosno jednog steka slika. Polje *image scale* služi da se naznači da li učitane slike treba skalirati sa nekom konstantom. Polje *precision* predstavlja sa kojim preciznošću, tj. pomerajem korisnik želi da pomera z-pozicije mikroskopa. Slider omogućava pomeranje z-pozicije mikroskopa izraženo u mikrometrima. Na vrhu slajdera se nalazi polje u kojoj se nalazi trenutna vrednost z-pozicije mikroskopa, ako nema ništa učitano nalazi se 0. Takođe, moguće je označiti sve mere fokusiranosti koje treba izračunati na steku slika. To je omogućeno poljima za izbor (eng. *check box*). Sve mere koje su odabrane se izračunavaju prilikom klika na dugme *calculate focus measures*. Na slici 6.2 prikazano je kako izgleda kada se učitava stek slika. Sa leve strane se nalazi prozor koji prikazuje slike iz steka na određenim pozicijama.

Klikom na dugme *calculate focus measures* se pojavljuje prozor koji prikazuje grafik iscrtanih mera fokusiranosti. Slika 6.3 prikazuje kako izgleda iscrtan grafik odabranih mera fokusiranosti (desno). Dodatno, ako je odabrano *cursor* pojavljuje se vertikalna crna linija na grafiku koja omogućava da pomeranjem slajdera korisnik prati menjanje fokusiranosti slike simultano sa vrednostima mere fokusiranosti. Na slici (A) z pozicija mikroskopa je $-2965.5 \mu\text{m}$, slika očigledno nije fokusirana što se može videti i po slici golim okom, ali još važnije detektuje se da mere fokusiranosti nemaju globalni maksimum na toj poziciji. Na slici (B) se vidi da je slika fokusirana kao i da mere fokusiranosti dostižu maksimum kad je z-pozicija na toj poziciji. Na slici (C) se može videti da je pomeranjem z-pozicije na vrednost -2949.5 slika potpuno otišla iz fokusa, a to nam takođe potvrđuje činjenica da su vrednosti mera fokusiranosti opale u odnosu na sliku (B). Prethodni primer prikazuje i osnovnu namenu kursora i sinhronizaciju kursora i slajdera. To je omogućilo esencijalna zapažanja u daljem



SLIKA 6.1: Simulator mikroskopa.

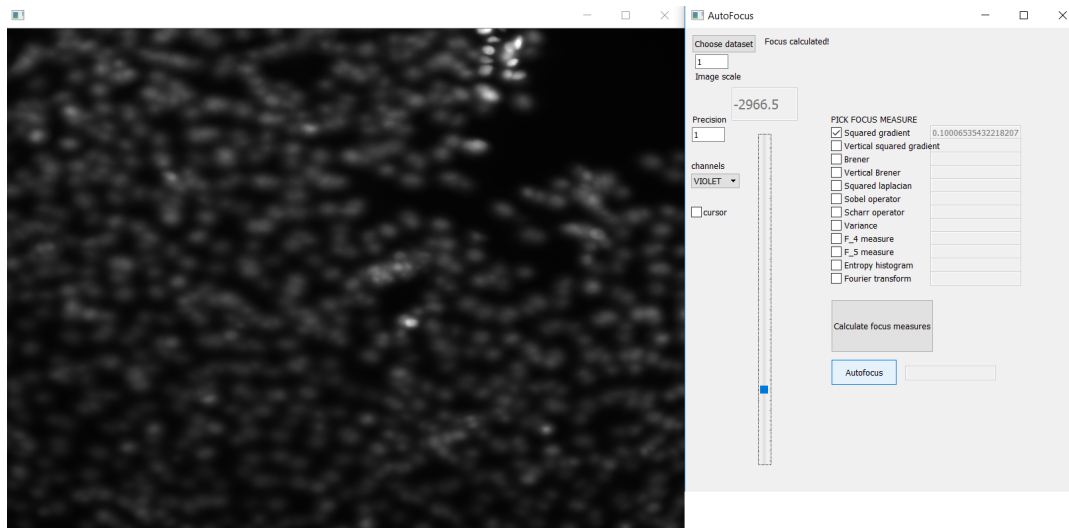
istraživanju jer je fenomen fokusiranja mogao sinhronizovano da se prati po više dimenzija.

6.2 Radni okvir za razvoj modela mašinskog učenja

Istraživanje i razvoj modela mašinskog učenja je zahtevan po više dimenzija. Istraživanje, odabir arhitekture i parametara modela je posebna celina za sebe i o tome je bilo reči u prethodnim poglavljima. Razvoj modela mašinskog učenja može se podeliti u dve faze koje nisu sasvim razdvojene. Prva faza je implementacioni deo u kom programer mora da implementira celo okruženje u kom će vršiti treniranje i izbor modela. Sam trening modela mašinskog učenja nije programerski zahtevan. Računski jeste zahtevan pa ako nisu obezbeđeni značajni resursi (grafičke kartice) za trening, može da traje vremenski duže, dok ako su obezbeđeni značajni resursi trening će trajati nešto kraće, sve zavisi od problema do problema.

U ovom delu će biti ukratko predstavljen implementacioni deo radnog okvira. Samo jezgro radnog okruženja sastoji se iz više modula. Neki od modula su:

- modul za pripremu atributa



SLIKA 6.2: Simulator mikroskopa sa učitanim stekom slika.

- modul za regresiju
- modul za autofokusiranje
- modul za evaluaciju

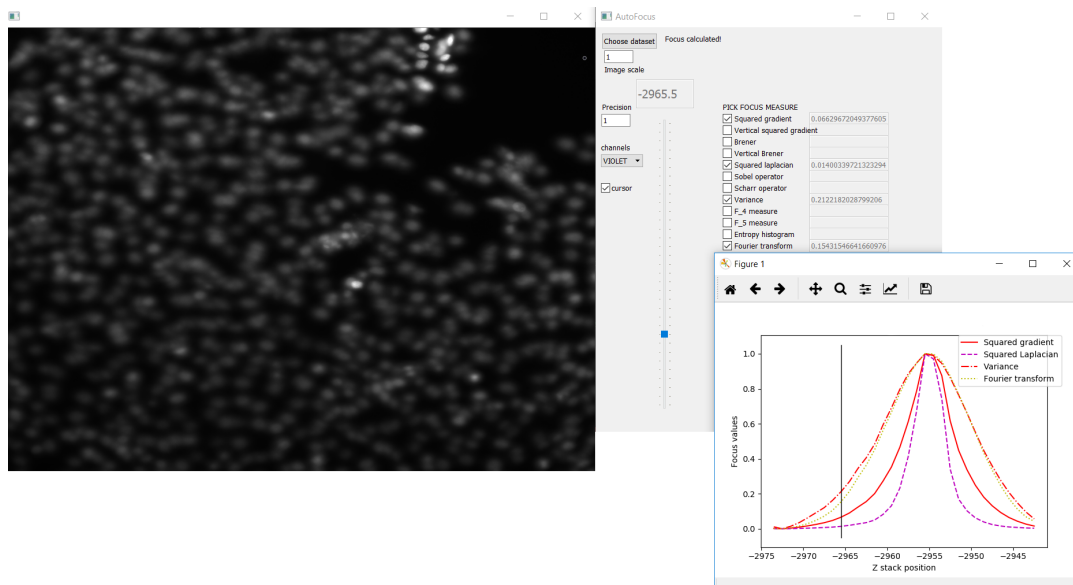
Modul za pripremu atributa potpuno je posvećen radu sa atributima. Učitava sirove attribute celog steka slika i kombinuje ih na razne načine. Najbitniji metodi su metodi za generisanje atributa modela. Sadrži i metode za rad sa sirovim slikama, koje izračunavaju attribute i dalje kombinuju po potrebi modela.

Modul za regresiju sadrži sve neophodne elemente za razvoj i trening modela mašinskog učenja. U sklopu ovog modula implementirane su arhitekture neuronskih mreža kako za model sa jednom slikom tako i za model koji koristi dve slike. Pored implementacije samih arhitektura sadrži metod za unakrsnu validaciju opisanu u odeljku 5.4.3, metodu za treniranje nad celim skupom podataka, metode za čuvanje i učitavanje modela u hdf5 formatu [19], metod za predikciju sa već učitanim modelom itd. U suštini, ovo je modul koji omogućava razvoj modela mašinskog učenja. Jednostavnom zamenom metode koja implementira arhitekturu neuronske mreže nekim drugim algoritmom mašinskog učenja omogućava brz prelazak na neki drugi algoritam. Ostatak koda nije potrebno menjati.

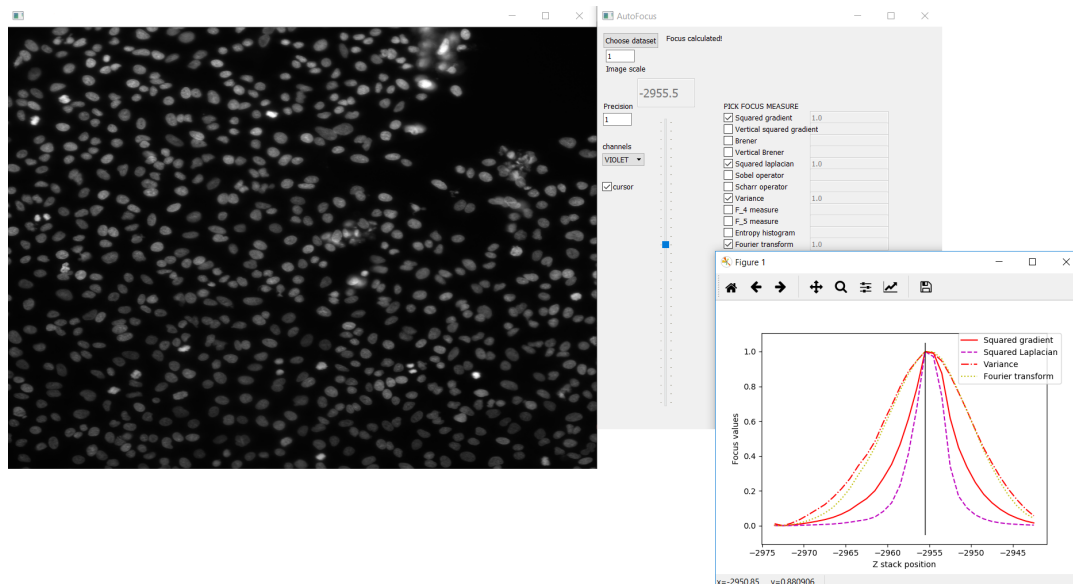
Modul za autofokusiranje sadrži različite pristupe autofokusiranja. Implementirano je autofokusiranje optimizacionim metodama (Brentov metod i metod penjanja uzbrdo) i optimizacioni metod koji je predstavljen u ovom radu. Jednostavno se koriste, pozove se metoda koju korisnik želi, a kao argument se prosledi pozicija steka slika i dodatni parametri optimizacije. Prilagođen je i za rad na realnim mikroskopima a ne samo na simulatoru.

Modul za evaluaciju služi da koristeći modul za autofokusiranje poredi pristupe autofokusa i da da ocenu performansi.

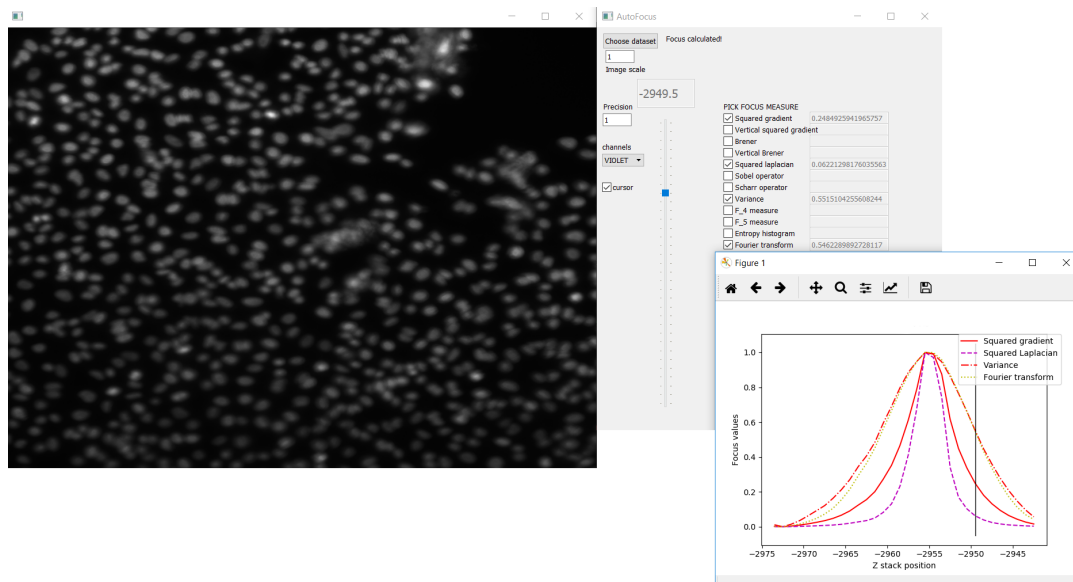
Kod koji je napisan u službi autofokusa je prilagodljiv stvarnim mikroskopima. Budući da je simulator mikroskopa pravljen po uzoru na realan mikroskop, jednostavnim pisanjem interfejsa za komunikaciju sa hardverom mikroskopa može se prilagoditi upotrebi u realnim uslovima.



(A)



(B)



(C)

SLIKA 6.3: Primer upotrebe softvera.

Glava 7

Diskusija

U ovom radu predstavljen je nov metod autofokusa baziranog na algoritmu mašinskog učenja. Osnovna prednost ovog metoda je što istovremeno uzima u obzir broj akvizicija i grešku fokusiranja. Ovo rešenje je primenjeno na mikroskope i rezultati su predstavljeni u odeljku 5.5. Sa dovoljno velikim skupom podataka mogli bi se istrenirati i modeli za ostale uređaje kojima je potrebno fokusiranje kao što su mobilni telefoni, teleskopi itd.

Postavlja se pitanje da li su neuronske mreže sa propagacijom unapred najbolji algoritam mašinskog učenja za ovaj problem? Verovatno i nisu uzimajući u obzir da su konvolutivne neuronske najpopularniji pristup u primenama modela mašinskog učenja nad signalima. Trening konvolutivnih neuronskih mreža je računski vrlo zahtevan. Korišćenje neuronske mreže sa propagacijom unapred je omogućilo da zahtevi za performansama računara nisu veliki, te se ovaj metod autofokusiranja može koristiti i na slabijim mašinama.

Prostora za dalji razvoj definitivno ima. Imajući u vidu sekvencijalnu prirodu procesa optimizacije, jedna alternativa neuronskoj mreži sa propagacijom unapred je rekurentna neuronska mreža. Drugi put istraživanja je predstavljen u pasusu iznad i to je slučaj primene konvolutivnih neuronskih mreža. U slučaju da se ovaj pristup pokaže kao uspešan na digitalnim fotoaparatom, telefonima, kamerama, mogao bi da bude primenjen na probleme autonomne vožnje.

Već je pomenuto da bi bilo dobro da postoji adekvatan klasifikator koji klasifikuje slike, dobijene pomoću mikroskopa, na fokusirane i nefokusirane. Tada bi se on koristio kao kriterijum zaustavljanja, tako da ne bi bilo potrebe da se proces fokusiranja nastavi ako je već slika fokusirana.

Na ovom mestu biće uveden jedan drugačiji pogled na fokusiranje. Ako imamo sliku p mozemo je predstaviti kao konvoluciju polazne (fokusirane slike) I i filtera f sa nekim dodatnim šumom n .

$$p = I * f + n \quad (7.1)$$

Ako je poznat filter f onda je jednostavno naći polaznu sliku kao i udaljenost trenutne slike od idealno fokusirane slike. No, na žalost to nije jednostavno, uglavnom se filter f ne zna. Postoje načini aproksimacije filtera f . Jedan od načina aproksimacije može se naći u radu [20].

Bitno je razumeti da postoje dve vrste uzroka nefokusiranosti. To su:

- nefokusirane slike usled pomeranja kamere

- nefokusirane slike zbog pozicije izvan fokusa

Prva vrsta nefokusiranosti nastaje usled pomeranja kamere, sočiva ili objekta koji se fotografiše. Druga vrsta nefokusiranosti nastaje usled udaljenosti od idealne pozicije fokusiranosti. Oba načina nefokusiranosti mogu se predstaviti kao konvolucija početne slike i nekog filtera. Ispostavlja se da ti filteri izgledaju vrlo slično *funkciji širenja tačke* (eng. *point spread function*) [21], ali im se osobine razlikuju. Koristeći metod potpornih vektora (engl. *support vector machine*) [22], algoritam mašinskog učenja, moguće je klasifikovati slike na fokusirane i nefokusirane. U pomenutom radu je čak moguće i klasifikovati vrste nefokusiranih slika. Ostaje da se ovaj pristup proba na tim slikama jer je u tom radu eksperimentisano na standardnim slikama. Zbog malog skupa podataka fokusiranih i nefokusiranih slika ovaj metod nije testiran u ovom radu.

Glava 8

Zaključak

Autofokusiranje, iako to možda nije očigledno, je bitan problem koji utiče na kvalitet života ljudi. Trivijalan primer potrebe za autofokusiranjem je prilikom korišćenja fotoaparata i mobilnih uređaja. Ozbiljnije primene uzima u medicini, astronomiji i drugim naučnim oblastima. Skoro svi optičkim uređaji su značajno upotrebljiviji u praksi ukoliko postoji kvalitetan algoritam autofokusiranja.

Prilikom autofokusiranja cilj je minimizovati grešku autofokusiranja. Rešavanjem ovog problema se bavi većina dosadašnjih metoda autofokusiranja. Budući da svaka akvizicija zahteva resurse i vreme, poželjno je optimizovati i ovaj segment korišćenja mikroskopa. Centralni problem koji se rešava u radu je uporedna minimizacija greške autofokusiranja i broja akvizicija.

U radu su predstavljena dva modela mašinskog učenja. Model koji koristi jednu sliku pri predviđanju neozačene vrednosti rastojanja trenutne pozicije i pozicije idealne fokusiranosti, dok model sa dve slike predviđa označenu vrednost rastojanja trenutne pozicije i pozicije idealne fokusiranosti. To je nov način upotrebe algoritama mašinskog učenja za potrebe autofokusiranja.

Predložen je optimizacioni algoritam koji koristeći pomenute modele konvergira ka tački idealne fokusiranosti mikroskopa. Modeli navode algoritam ka idealnoj z-poziciji u kojoj je slika fokusirana. Predstavljeni algoritam za vrednost ciljne promenljive kombinuje više mera fokusiranosti, ne postoji privilegovana mera koja se smatra povlašćenom već se uprosečavanjem više mera fokusiranosti dobija optimalna mera. Glavna prednost ovog algoritma je rešavanje problema koji je naveden kao najznačajniji, a to je istovremena minimizacija greške autofokusiranja i broja akvizicija. Navedene prednosti su značajne i prave razliku između predstavljenog algoritma i postojećih optimizacionih algoritama autofokusiranja koji se ne bave ovim problemima.

Tokom istraživanja radi boljeg razumevanje procesa fokusiranja implementiran je i softver za simulaciju mikroskopa. Pored namene da simulira rad mikroskopa, softver poseduje i dodatak za vizualizaciju mera fokusiranosti i za njihovo poređenje. Pored simulatora mikroskopa u radu je predstavljena i implementacija optimizacionog algoritma koja je napisana u programskom jeziku Python.

Dalji razvoj autofokusiranja može se sprovesti u smeru predstavljenom u ovom radu. Pomenuta je mogućnost korišćenja novog kriterijuma zaustavljanja koji omogućava značajne informacije u toku ali i na kraju autofokusiranja. Ovaj kriterijum je ostvariv razvojem klasifikatora koji klasifikuje slike na fokusirane i nefokusirane. Razvoj nove mere fokusiranosti koja bi omogućila generalno korišćenje bez potrebe za poređenjem slika je ideal kome se teži. Dalje, može se pokušati sa drugim modelima neuronskih mreža kao što su rekurentne neuronske mreže ili konvolutivne neuronske mreže koje postižu značajne rezultate u primenama na probleme računarskog vida.

Bibliografija

- [1] Yi Yao, Besma Abidi, Narjes Doggaz, and Mongi Abidi. Evaluation of sharpness measures and search algorithms for the auto-focusing of high magnification images. *Proceedings of SPIE - The International Society for Optical Engineering*, 6246, June 2006.
- [2] Richard P. Brent. *Algorithms for Minimization Without Derivates*. Prentice-Hall, 1973.
- [3] Hashim Mir, Peter Xu, and Peter van Beek. An extensive empirical evaluation of focus measures for digital photography, journal =.
- [4] Jie He, Rongzhen Zhou, and Zhiliang Hong. Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera. *IEEE Transactions on Consumer Electronics*, 49, May 2003.
- [5] Rudi Chen and Peter van Beek. Improving the accuracy and low-light performance of contrast-based autofocus using supervised machine learning. *Pattern Recognition Letters*, 56:30–37, 15 April 2015.
- [6] Hashim Mir, Peter Xu, and Peter van Beek. An autofocus algorithm for digital cameras based on supervised machine learning. *Journal of Heuristics*, 21:599–616, October 2015.
- [7] Narjes Doggaz Yi Yao, Besma Abidi and Mongi Abidi. Evaluation of sharpness measures and search algorithms for the auto-focusing of high magnification images. *Proceedings of SPIE - The International Society for Optical Engineering*, 6246, June 2006.
- [8] Predrag Janičić and Mladen Nikolić. *Veštačka inteligencija*. Matematički fakultet.
- [9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.
- [10] G. Cybenkot. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, 2:03–314, 1989.
- [11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [12] Guido Rossum. Python reference manual. 1995.
- [13] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [14] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:306 – 307, July 1979.
- [15] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

-
- [16] François Chollet et al. Keras. <https://keras.io>, 2015.
- [17] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed <today>].
- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Machine Learning*, 20:273–297, September 1995.
- [19] Itseez Intel Corporation, Willow Garage. Hdf5 for python. <http://docs.h5py.org/en/latest/index.html>.
- [20] Ping Hsu and Bing-Yu Chen. Blurred image detection and classification. *Advances in Multimedia Modeling, Springer*, 4903, 2008.
- [21] Rudi Rottenfusser, Erin E. Wilson, and Michael W. Davidson. Point spread function. <http://zeiss-campus.magnet.fsu.edu/articles/basics/psf.html>.
- [22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *CoRR*, abs/1412.6980:273–297, 2014.