



Univerzitet u Nišu  
Elektronski fakultet



Predmet: Prikupljanje i predobrada podataka za mašinsko učenje

## Augmentacija podataka kod slika

Seminarski rad

Smer: Veštačka inteligencija i mašinsko učenje

Student:

Anđelija Mladenović, br. ind. 1625

Profesor:

Doc. dr Aleksandar Stanimirović

Niš, april 2024. godine

# Sadržaj

1. Uvod.....	3
2. Motivacija za korišćenje augmentacije podataka kod slika.....	4
3. Tradicionalne metode augmentacije podataka kod slika i njihova podela .....	5
4. Pregled tradicionalnih metoda augmentacije podataka kod slika .....	6
4.1 Geometrijske transformacije .....	6
4.1.1 Rotacija .....	6
4.1.2 Izvrtanje .....	7
4.1.3 Isecanje i zumiranje .....	8
4.1.4 Pomeranje .....	8
4.2 Fotometrijske transformacije.....	9
4.3 Metode bazirane na dodavanju šuma .....	11
4.3.1 Dodavanje šuma.....	12
4.3.2 Nasumično brisanje.....	13
4.4 Metode bazirane na primeni filtera .....	15
4.4.1 <i>Sobel</i> filter .....	15
4.4.2 <i>Gaussian</i> filter.....	17
4.4.3 <i>Canny</i> filter .....	19
5. Zaključak.....	23
Literatura.....	24

# 1. Uvod

Modeli dubokog učenja stekli su tokom godina veliku popularnost i uznapredovali za primenu za rešavanje zadataka vezanih za računarski vid, kao što su klasifikacija slika, detekcija objekata, segmentacija objekata sa slike itd. Ovo je podstaknuto pojavom različitih novih arhitektura neuronskih mreža, dostupnošću računarskih resursa za njihovo pokretanje i dostupnošću sve veće količine podataka [1].

Ipak, algoritmi korišćeni kod dubokog učenja zahtevaju ogromne količine podataka, i često imaju problema sa *overfitting*-om – model radi jako dobro na trening podacima, ali i izuzetno loše na test, odnosno „neviđenim“ podacima. Ovaj problem se obično javlja u slučajevima kada nam nije dostupna dovoljno velika količina podataka za treniranje modela – ovo se može dogoditi zbog zaštite privatnosti ljudi koji se na slikama nalaze (čest problem sa medicinskim slikama), ili u slučajevima kada je labeliranje slika jako zahtevno i skupo [1].

Jedno od potencijalnih rešenja ovog problema predstavlja augmentacija podataka. Augmentacija podataka obuhvata širok spektar tehnika koje se primenjuju na same podatke, a ne na model sa kojim radimo. Ove tehnike mogu poboljšati performanse modela tako što veštački proizvode različite i raznolike uzorke na kojima će model moći da uči. Kada je skup podataka sa kojim radimo dovoljno kvalitetan i poseduje dovoljno uzoraka, performanse modela koji se primenjuje za neki zadatak su znatno bolje. S toga, podaci s kojima radimo moraju ispuniti dva uslova – raznolikost i brojnost, a i jedno i drugo može se postići augmentacijom podataka [2].

Iako postoje razne podele metoda za augmentaciju podataka kod slika, najčešća je ona koja ih deli na tradicionalne metode i metode zasnovane na dubokom učenju. Tradicionalne metode odnose se na one koje izvršavaju neku vrstu transformacije na slici, bile to geometrijske, ili fotometrijske transformacije, dodavanje šuma, primena filtera i slično. Sa druge strane, metode zasnovane na dubokom učenju najčešće koriste *GAN*-ove (*Generative Adversarial Network*) da na osnovu trening slika generišu potpuno nove, sintetičke slike koje će se koristiti za treniranje našeg modela[2]. U ovom radu, fokus će biti na tradicionalnim metodama, i one će biti obrađene u nastavku.

## 2. Motivacija za korišćenje augmentacije podataka kod slika

Modeli dubokog učenja uspešno su primenjeni za rešavanje zadataka iz oblasti računarskog vida kao što su klasifikacija slika i segmentacija slika, detekcija objekata i slično, a sve to zahvaljujući razvoju konvolucionih neuronskih mreža (engl. *Convolutional Neural Network* - CNN). Ove neuronske mreže koriste parametrizovane, retko povezane kernele koji očuvavaju prostorne karakteristike slike. Konvolucionni slojevi sekvencijalno *downsample*-uju prostornu rezoluciju slike dok u isto vreme povećavaju dubinu mapa karakteristika (engl. *feature map*). Niz konvolucionih transformacija može napraviti reprezentaciju slike koja je znatno manjih dimenzija od originala, ali koja predstavlja znatno korisniji oblik slike [3].

Povećanje sposobnosti generalizacije ovih modela pokazalo se kao značajan problem. Generalizacija predstavlja sposobnost modela da radi podjednako dobro na već viđenim primerima (trening podaci) i na podacima sa kojima se nije susreo u prošlosti (test podaci). Ukoliko model ne generalizuje dobro, odnosno radi lošije na test podacima, to obično znači da je došlo do *overfitting*-a – model je naučio da radi sa trening slikama, ali nije imao dovoljno podataka da podesi svoje parametre tako da može da radi i sa drugim podacima [3].

Razvijen je veliki broj metoda za suočavanje s ovim problemom, a jedna od njih je i augmentacija podataka. Augmentacija podataka predstavlja veoma moćan metod za poboljšanje performansi modela. Njenom primenom proširujemo trening skup podataka modifikacijom već postojećih slika, čime dobijamo opširniji i temeljniji skup mogućih podataka, čime se smanjuje moguća razlika između trening podataka i podataka na kojima će se model primenjivati u budućnosti [3].

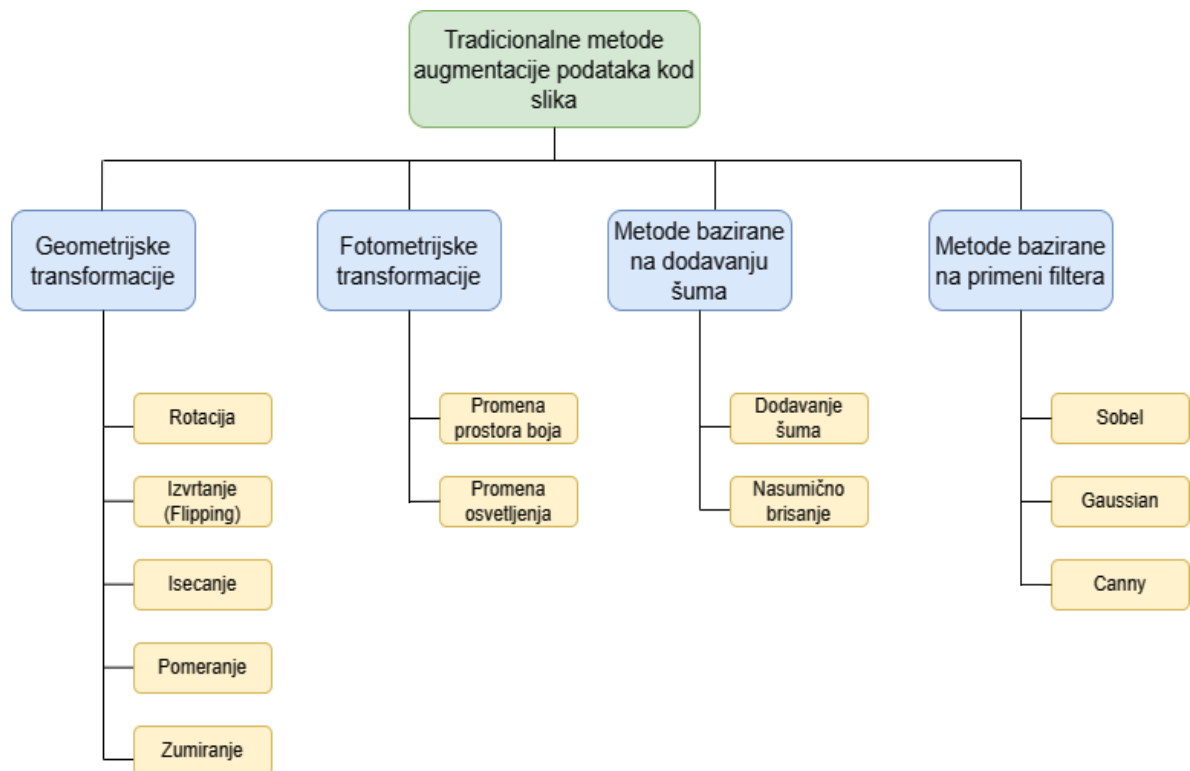
Za razliku od većine drugih tehnika, augmentacija podataka prisupa problemu *overfitting*-a od njegovog korena, odnosno od samog trening skupa podataka. Ovo se radi pod pretpostavkom da se dodatne informacije mogu dobiti iz originalnog skupa podataka primenom različitih augmentacija. Ove augmentacije veštački povećavaju veličinu trening skupa podataka. Augmentacije mogu predstavljati različite stvari – geometrijske transformacije slike, njihovo zamućenje, promenu njihove reprezentacije, odnosno prostora boja, zatim brisanje određenih delova slike, i druge [3].

Činjenica da augmentacija podataka povećava broj trening primera jeste najočiglednija, ali ne i jedina prednost ove metode. Cilj pri pravljenju modela za rešavanje zadataka iz oblasti računarskog vida jeste da ti modeli budu što bliži čoveku. Transformacije koje se najčešće primenjuju prilikom augmentacije su percepciono moguće – ljudsko oko radi gotovo podjednako dobro bez obzira na osvetljenje, vid može da se zamuti, odnosno – augmentacijom podataka zapravo simuliramo realne uslove pod kojima ljudsko oko može da radi. Augmentacijom podataka sposobni smo da generišemo veliki broj novih primera kojima ćemo „predviđati“ potencijalne realne primere na kojima će model biti korišćen [4].

### 3. Tradicionalne metode augmentacije podataka kod slika i njihova podela

Kao što je ranije pomenuto, metode augmentacije podataka mogu se podeliti na tradicionalne metode, koje se baziraju na modifikaciji slika iz trening seta, i na metode zasnovane na dubokom učenju, koje koriste određene modele dubokog učenja da na osnovu trening primera generišu potpuno nove primere [2]. Fokus u ovom radu biće stavljen na tradicionalne metode augmentacije podataka.

Tradicionalne metode mogu se podeliti u četiri grupe: geometrijske transformacije, fotometrijske transformacije, metode bazirane na dodavanju šuma i metode bazirane na primeni nekog filtera [2]. Taksonomija tradicionalnih metoda augmentacije podataka na slikama data je na slici 3.1. Važno je napomenuti da ovo nije jedina podela metoda augmentacije podataka kod slika, ali jeste ona koju ćemo usvojiti u ovom radu. Svaka od metoda biće obrađena pojedinačno u nastavku rada.



Slika 3.1 Taksonomija tradicionalnih metoda augmentacije podataka [2]

## 4. Pregled tradicionalnih metoda augmentacije podataka kod slika

U nastavku će biti dat pregled tradicionalnih metoda za augmentaciju podataka kod slika.

### 4.1 Geometrijske transformacije

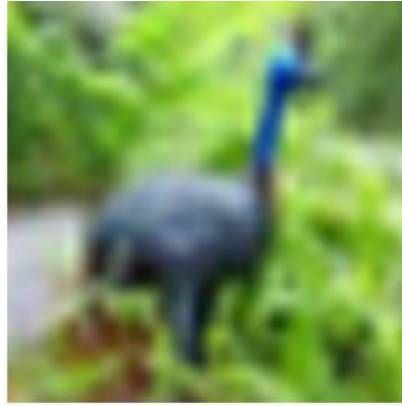
Osnovne geometrijske transformacije, kao što su izvrtanje (engl. *flipping*), isecanje ili rotacija su i dalje popularni metodi za augmentaciju podataka. Ove tehnike generalno povećavaju količinu dostupnih podataka za treniranje, i doprinose diverzitetu podataka. Takođe, njihova primena je jako jednostavna [2].

Ovaj odeljak baviće se različitim geometrijskim transformacijama slike koje se mogu iskoristiti za augmentaciju podataka. Takođe, biće razmotrena i „bezbednost“ primene ovih transformacija. Bezbednost augmentacije podataka odnosi se na verovatnoću očuvanja labele slike nakon transformacije. Na primer, rotacija i izvrtanje su generalno bezbedne kada je u pitanju klasifikacija slika na slike sa psima i slike sa mačkama, ali neće biti bezbedne ukoliko je neophodno da razvrstamo slike koje sadrže broj 9 ili broj 6. Transformacija koja ne očuvava labele u potpunosti može potencijalno pojačati sposobnost modela da da odgovor čak i onda kada nije potpuno siguran u svoju predikciju. Međutim, da bi se ovo postiglo neophodne su dodatne modifikacije modela, što može biti skup proces. Zbog toga, važno je razmotriti „bezbednost“ augmentacije podataka. Ona je donekle zavisna od domena primene, zbog čega je teško razviti neka generalna pravila o augmentaciji podataka. Ne postoji ni jedna metoda obrade slike koja nikada ne može rezultovati izmenom labele. Sve ovo neophodno je razmotriti pre primene metoda augmentacije navedenih u nastavku [3].

#### 4.1.1 Rotacija

Rotacija je jednostavna geometrijska transformacija često korišćena za augmentaciju podataka. Njenom primenom, postojeće slike se rotiraju za određeni ugao, i zatim se novodobijene slike koriste kao trening primeri [2].

Rotacija se može vršiti u levo ili u desno za ugao od  $1^\circ$  do  $359^\circ$ . Bezbednost rotacije je određena vrednošću ugla rotacije. Blaga rotacija za ugao između  $1^\circ$  i  $20^\circ$  (odnosno  $-1^\circ$  do  $-20^\circ$ ) može biti korisna za treniranje modela koji prepoznaje cifre, ali što se ugao više povećava, sve su veće šanse da labela podataka neće biti očuvana nakon transformacije [3]. Primer slike na kojoj je primenjena rotacija može se videti na slici 4.1.



a)



b)

Slika 4.1 a) Originalna slika; b) Slika nakon rotacije [2]

#### 4.1.2 Izvrtanje

Termin izvrtanje (engl. *flipping*) odnosi se na proces izvrtanja slike horizontalno, vertikalno ili u oba pravca, kao što je prikazano na slici 4.2 [2]. U praksi, horizontalno izvrtanje je mnogo češće korišćeno od vertikalnog, zbog toga što je obično realističnije [2, 3]. Ova augmentacija jedna je od najlakših za implementaciju, i pokazala se kao izuzetno korisna na brojnim skupovima podataka [3].

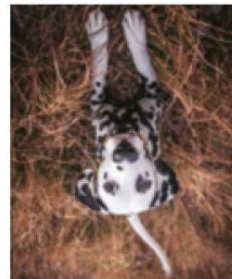
Izvrtanje je jedna od najintuitivnijih metoda za povećanje diverziteta i broja podataka. Ipak, nije pogodna za primenu na podacima koji imaju jedinstvena svojstva. Na primer, izvrtanje nije pogodno onda kada su objekti koje segmentiramo asimetrični ili su osetljivi na smer, kao što su to slova ili brojevi. U ovakvim slučajevima, primena izvrtanja rezultovaće promenom labele podataka, što će prouzrokovati brojne probleme [2].



a)



b)



c)



d)

Slika 4.2 a) Originalna slika; b) Horizontalno izvrtanje; c) Vertikalno izvrtanje; d) Oba izvrtanja [2]

#### 4.1.3 Isecanje i zumiranje

Ove dve metode najčešće idu ruku pod ruku. Isecanje je često korišćena metoda augmentacije koja može biti primenjena na nasumični deo slike ili na sam centar slike. Ova metoda podrazumeva isecanje dela slike i zatim njegovo reskaliranje (zumiranje) na način da veličina novodobijene slike odgovara veličini inicijalne slike [1, 2]. Zbog toga što trening skup podataka može imati slike različite veličine, isecanje se često koristi za preprocesiranje podataka kako bi se sve slike dovele na iste dimenzije [2].

Bitno je pomenuti da isecanje može proizvesti primere sa pogrešnom labelom. Na primer, slika koja sadrži više od jednog objekta, koja je klasifikovana na osnovu objekta koji je na slici dominantan po veličini, može biti pogrešno labelirana ukoliko se taj objekat većinski iseče sa slike. U tom slučaju, moguće je iseći deo slike koji sadrži više detalja tih pratećih objekata, umesto onog koji sadrži dominantni objekat, kao što je prikazano na slici 4.3. Ova slika inicijalno je bila označena labelom „Pas“, ali se nakon isecanja pas uopšte ne nalazi na slici, što znači da je labela postala netačna. Zbog toga, neophodno je obratiti pažnju na primenu ove metode.



a)



b)

Slika 4.3 a) Originalna slika, označena labelom „Pas“; b) Slika nakon isecanja i reskaliranja (zumiranja), i dalje, ali sada netačno, označena labelom „Pas“ [2]

#### 4.1.4 Pomeranje

Pomeranje, odnosno translacija, je metoda augmentacija podataka na slici koja podrazumeva pomeranje slike gore, dole, levo ili desno kako bi se dobila raznolika reprezentacija podataka [1]. Ovakvo pomeranje može biti značajno za sprečavanje tendencije modela (engl. *bias*) da povezuje objekte sa njihovom prostornom lokacijom. Na primer, ukoliko su sve slike u skupu podataka

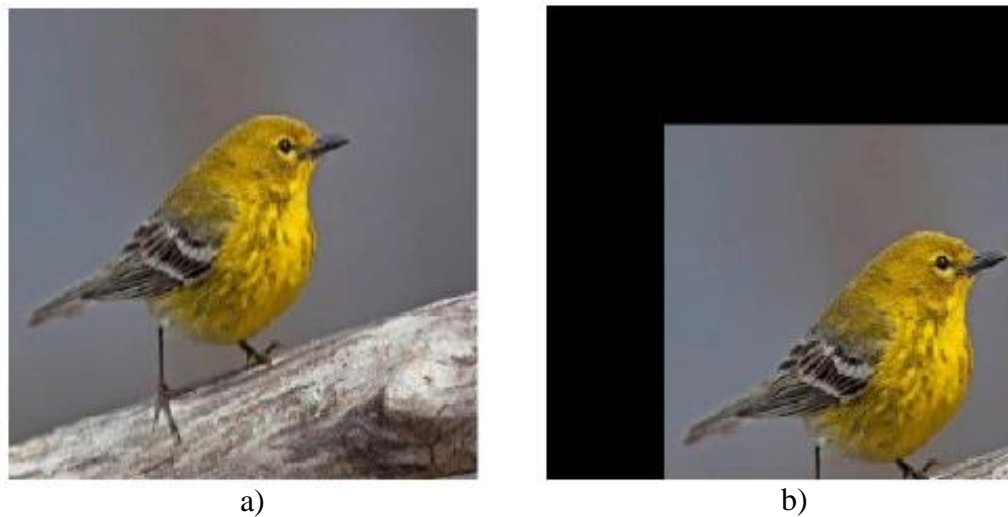


centrirane, što je često kod skupova podataka vezanih za prepoznavanje lica, to znači da će i sve test slike morati da budu centrirane da bi model radio kako treba, što gotovo nikad nije slučaj. Zbog toga bi pomeranje bila pogodna transformacija za augmentaciju ovakvog skupa podataka [3].

Nakon pomeranja, delovi slike koji ostanu „prazni“, odnosno iz kojih je originalna slika izmeštena, mogu se dopuniti nekom konstantnom vrednošću kao što su 0 ili 255, ili se pak mogu popuniti nekim Gausovim šumom. Ova dopuna služi da sačuva prostorne dimenzije slike i nakon augmentacije [3].

Magnituda pomeranja mora biti pažljivo izabrana u zavisnosti od slučaja primene. Veliko pomeranje slike može rezultovati značajnom promenom izgleda slike. Na primer, pomeranje broja 8 u levo za polovinu širine slike može rezultovati slikom koja je približnija broju 3 nego originalnoj labeli. Zbog toga, jako je bitno razmotriti prirodu skupa podataka sa kojima radimo pri određivanju magnitude translacije kako bi se osigurala njena efikasnost [1].

Primer pomeranja dat je na slici 4.4.



Slika 4.4 a) Originalna slika; b) Slika nakon pomeranja na dole i u desno [1]

## 4.2 Fotometrijske transformacije

Za razliku od geometrijskih transformacija koje menjaju poziciju piksela na slici, fotometrijske transformacije utiču na same vrednosti kojima su pikseli predstavljeni. Ovaj pristup podrazumeva različite metode, kao što su promena osvetljenja i/ili boja [2].

Baziranje odluke na osnovu osvetljenja slike (engl. *lighting bias*) je među najčešćim problemima kod modela za prepoznavanje slika. Zbog toga su fotometrijske transformacije jako značajne. Na primer, slike koje su previše svetle ili tamne se na jednostavan način mogu „popraviti“ iteriranjem kroz piksele i povećanjem, odnosno smanjenjem vrednosti kanala piksela za konstantnu vrednost. Način predstavljanja boja digitalnih slika daje brojne mogućnosti za augmentaciju podataka [3].

Tradicionalne fotometrijske transformacije mogu imati svoja ograničenja – na primer, mogu zahtevati veliku količinu računarskih resursa i memorije. Pored toga, mogu rezultovati krucijalnim gubitkom podataka sa slike, naročito u slučajevima kada je boja kolorimetrijska karakteristika na osnovu koje se mogu diferencirati različite kategorije u skupu podataka sa kojim radimo [2]. I pored toga, ove tehnike su i dalje često korišćene za augmentaciju podataka kod slika.

Slike su najčešće enkodirane kao trodimenzionalni tensor, dimenzija  $visina \times širina \times broj\ kanala\ boje$ . Razlika između različitih reprezentacija boja leži u delu tenzora koji predstavlja kanale boja. Na primer, RGB reprezentacija, odnosno prostor boja koristi kombinaciju tri kanala boja (crveni, zeleni i plavi) da predstavi svaki pojedinačni piksel. Manipulacija, odnosno promena ovih kanala boja predstavlja jednu od osnovnih tehnika u augmentaciji podataka vezanih za boju kod slika. Na primer, slika će se lako prevesti u svoju reprezentaciju pomoću samo jedne od boja ukoliko se svi drugi kanali postave na nulu [2].

Pored već pomenutog RGB prostora boja, postoje i brojne druge reprezentacije boja. Na primer, HSL prostor boja kombinuje nijansu, saturaciju i osvetljenost boje. Nijansa je jedinstveni pigment koji nema primesa (crnu ili belu boju). Saturacija označava jačinu boje, a osvetljenost označava koliko je boja svetla, odnosno tamna [2]. Postoje i brojni drugi prostori boja, o kojima ovde neće biti reči. Primer slike koja je iz RGB prostora boja prevedena u nijanse sive (engl. *grayscale*) može se videti na slici 4.5.



a)



b)

Slika 4.5 a) Originalna slika predstavljena u RGB prostoru boja; b) Slika predstavljena kao *grayscale* slika [1]

Promenom vrednosti svakog od kanala slike pojedinačno, možemo sprečiti model da se „navikne“ na određeni tip osvetljenja. Jedan od najprostijih metoda promene prostora boja i izmene osvetljenja podrazumeva zamenu jednog od kanala slike nasumično generisanim vrednostima, ili postavljanje celog kanala na konstantnu vrednost (najčešće 0 ili 255) [1]. Primer primene različitih tehnika za izmenu osvetljenja i prostora boja dat je na slici 4.6.



Slika 4.6 Primer primene različitih fotometrijskih transformacija na slici [3]

### 4.3 Metode bazirane na dodavanju šuma

U ovom delu razmotrićemo dva načina za augmentaciju podataka kod slika koja spadaju u ovu grupu: dodavanje šuma i nasumično brisanje delova slike.

#### 4.3.1 Dodavanje šuma

Šum se često definiše kao nasumična varijacija u boji slike. Ova pojava obično je prouzrokovana tehničkim ograničenjima senzora kojima se slika beleži ili lošim uslovima za slikanje. Na žalost, ovi problemi su u realnim situacijama najčešće neizbežni, te je pojava šuma čest problem kod akvizicije slika [2].

Iako šum predstavlja problem, poželjno je da model koji treniramo dobije i trening primere koji sadrže šum. Na ovaj način pripremamo model da radi u realnim uslovima. Pokazalo se da dodavanje šuma na trening slike povećava robustnost modela [1].

Gausov, „so i biber“ (engl. *Salt and Pepper*) i tačkasti (engl. *speckle*) šum su tri poznata oblika šuma koji se često koriste za augmentaciju podataka kod slika. Gausov šum je statistički šum čija je funkcija gustine verovatnoće jednaka normalnoj distribuciji. Distribucija Gausovog šuma je uniformna kroz signal. Pošto je ovo aditivni šum, pikseli rezultujuće slike predstavljaju sumu originalnih vrednosti piksela i vrednosti Gausovog šuma u toj tački. Ono što je dobro je da je vrednost Gausovog šuma nezavisna od magnitude signala na kom se primenjuje [2].

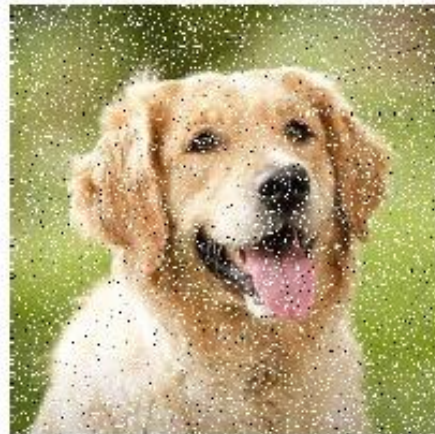
*Salt and Pepper* šum prouzrokuje pojavu crnih i belih piksela na nasumičnim mestima na slici. Ovakav tip šuma je u realnim situacijama najčešće prouzrokovao greškama u prenosu podataka [2].

Tačkasti šum je multiplikativan. Ovo znači da se rezultujuća slika dobija tako što se vrednosti piksela originalne slike pojedinačno množe nasumično dobijenim vrednostima [2].

Primer dodavanja šuma prikazan je na slici 4.7.



a)



b)

Slika 4.7 a) Originalna slika; b) Slika nakon dodavanja *Salt and Pepper* šuma [1]



#### 4.3.2 Nasumično brisanje

Augmentacije podataka iz ove grupe uključuju brisanje određenog dela (ili delova) slike i njihovu zamenu vrednošću 0, 255 ili srednom vrednošću piksela iz celog skupa podataka. Postoji više različitih metoda za brisanje delova slika, a najčešće korišćeni su isecanje i brisanje (engl. *cutout*), sakrivanje (engl. *hide-and-seek*) i primena grid maske [1].

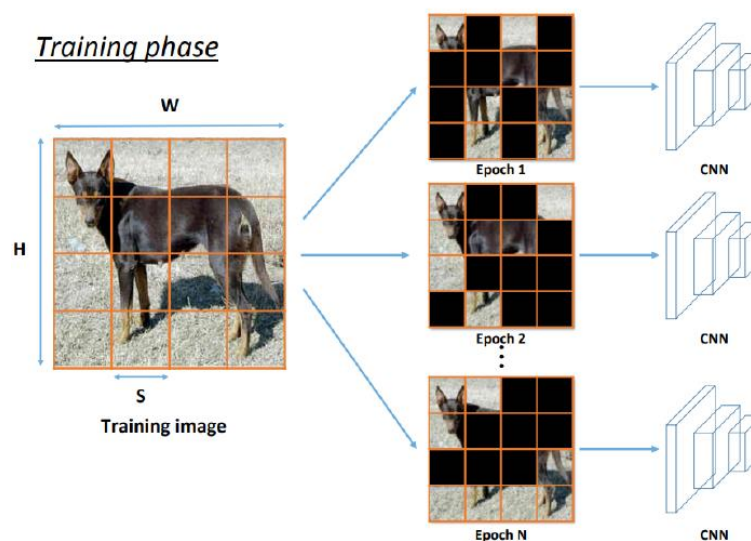
Isecanje podrazumeva uklanjanje podregiona slike, koji se zatim popunjava vrednostima 0 ili 255 [2]. Primer primene ovog metoda može se videti na slici 4.8.



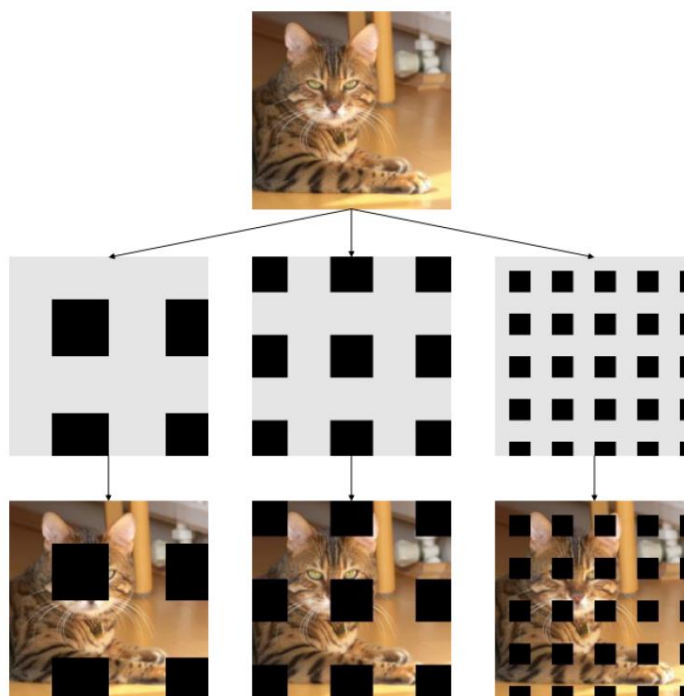
Slika 4.8 Gore – originalne slike; dole – slike sa isečenim nasumičnim delovima [2]

Sakrivanje uključuje podelu slike na uniformne kvadrate nasumične veličine i zatim nasumično uklanjanje određenog broja ovih kvadrata. Ova tehnika ima za cilj da prisili neuronsku mrežu da nauči da prepozna relevantne karakteristike objekata čak i onda kada joj informacije nedostaju. Ova tehnika se najčešće primenjuje tako da se u svakoj epohi treniranja modelu dostavlja ista slika, ali sa prikazanim različitim njenim delovima [1]. Primer sakrivanja može se videti na slici 4.9.

Primena grid maske na neki način kombinuje prethodne dve tehnike. U ovom slučaju kreira se maska delova slike koji će biti uklonjeni, i zatim se ta maska preklapa sa slikom kako bi se dobio željeni rezultat. Primer primene grid maske dat je na slici 4.10.



Slika 4.9 Primer sakrivanja delova slike [1]



Slika 4.10 Primena grid maske [1]

Zbog činjenice da se deo slike koji se briše najčešće bira nasumično, lako može doći do toga da se obrišu gotovo sve informacije od interesa sa slike. Zbog toga, ove tehnike nisu preporučljive u slučajevima kada se radi sa osetljivim podacima koji ne mogu podneti brisanje

nekog njihovog dela, kao što je to u slučaju prepoznavanja registarskih tablica vozila ili slova i brojeva [2].

#### 4.4 Metode bazirane na primeni filtera

Kernel filteri su veoma popularna tehnika u oblasti obrade slika. Ovi filteri funkcionišu tako što prevlače  $n \times n$  matricu preko slika, pri čemu matrice sadrže reprezentaciju nekog od filtara. U zavisnosti od vrste filtera koje koristimo, rezultujuća slika može biti zamućena verzija inicijalne slike, izoštrena verzija originala, može sadržati samo ivice originalne slike itd. Intuitivno, korišćenje i zamućenih slika prilikom treniranja modela može rezultovati većom otpornošću na zamućenje nastalo pokretom (engl. *motion blur*). Pored toga, korišćenje izoštrenih slika može pomoći modelu da bolje nauči detalje objekata od interesa [3].

Filteri su relativno neistražena oblast kada je u pitanju augmentacija podataka. Njihov nedostatak leži u činjenici da su veoma slični unutrašnjem mehanizmu konvolucionih neuronskih mreža. Konvolucione neuronske mreže koriste parametrizovane kernele da nauče da na najoptimalniji mogući način predstave sliku sloj po sloj. Kernel filteri se često mogu bolje implementirati kao sloj same mreže, a ne samo kao dodatak skupu podataka kroz augmentacije [3].

##### 4.4.1 Sobel filter

*Sobel* filter predstavlja jedan od najčešće korišćenih filtara za detekciju ivica na slici. Rezultat ovog filtera predstavlja magnitudu gradijenta slike (koju je neophodno prevesti u nijanse sive (engl. *grayscale*)). Najčešće se za računanje parcijalnih izvoda slike koriste kerneli dimenzija  $3 \times 3$  prikazani na slici 4.11. Vrednost magnitude gradijenta u pikselu na poziciji  $[i, j]$  može se dobiti pomoću sledeće formule:

$$M = \sqrt{s_x^2 + s_y^2} \quad (4.1)$$

gde se parcijalni izvodi  $s_x$  i  $s_y$  računaju na sledeći način:

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (4.2)$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \quad (4.3)$$

gde je  $c = 2$ , a vrednosti  $a_0 - a_7$  predstavljaju odgovarajuće elemente okoline piksela koji se posmatra [5]. Raspored piksela unutar okoline piksela koji se posmatra dat je na slici 4.12.

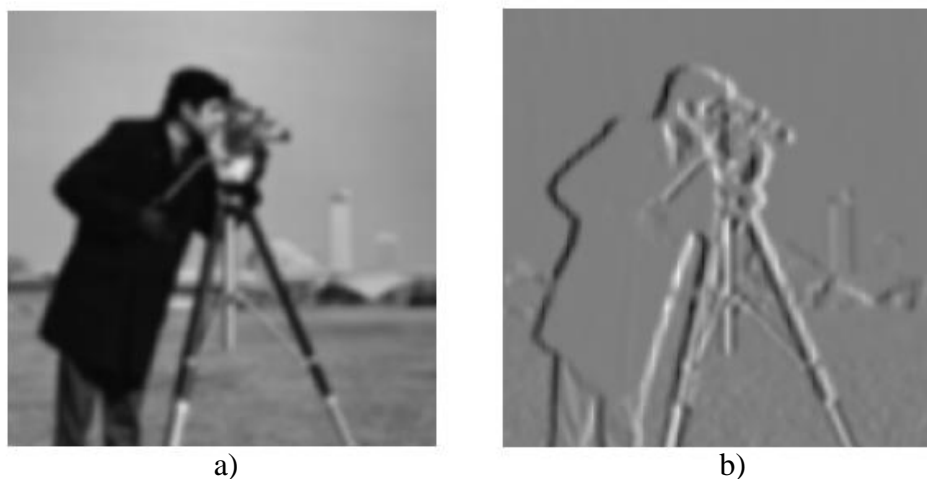
$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Slika 4.11 *Sobel* kerneli korišćeni za računanje parcijalnih izvoda slike [5]

$a_0$	$a_1$	$a_2$
$a_7$	$[i, j]$	$a_3$
$a_6$	$a_5$	$a_4$

Slika 4.12 Raspored piksela unutar okoline piksela  $[i, j]$  [5]

Ovi kerneli mogu se koristiti i pojedinačno, kada će nalaziti vertikalne, odnosno horizontalne ivice na slici [1]. Primena *Sobel* filtera za augmentaciju slika može pomoći u prepoznavanju oblika objekata od interesa [2]. Primer primene *Sobel* filtera može se videti na slici 4.13.



Slika 4.13 a) Originalna slika; b) Slika nakon primene *Sobel* filtera [2]



#### 4.4.2 Gaussian filter

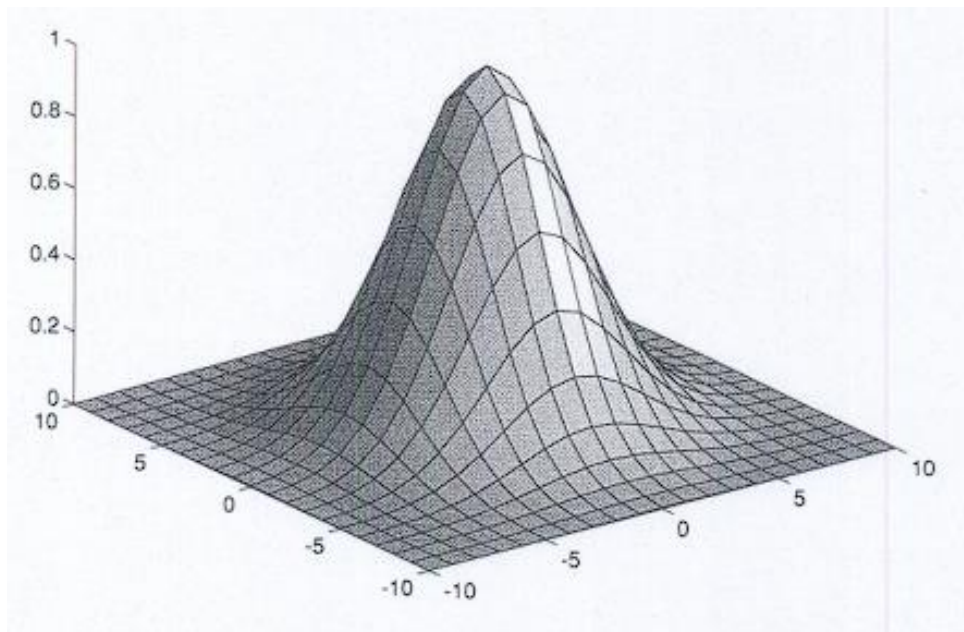
*Gaussian* filteri predstavljaju klasu linearnog „zaglađivanja“ (engl. *smoothing*) sa težinama unutar kernela izabranim tako da njihove vrednosti reprezentuju oblik Gausove funkcije. Ovaj filter jako je dobar za uklanjanje šuma sa slike. Gausova funkcija sa srednjom vrednošću 0 data je u jednoj dimenziji formulom:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (4.4)$$

gde parametar  $\sigma$  određuje širinu Gausove funkcije. Za potrebe obrade slika, Gausova funkcija sa srednjom vrednošću 0 se predstavlja sledećom formulom:

$$g[i, j] = e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \quad (4.5)$$

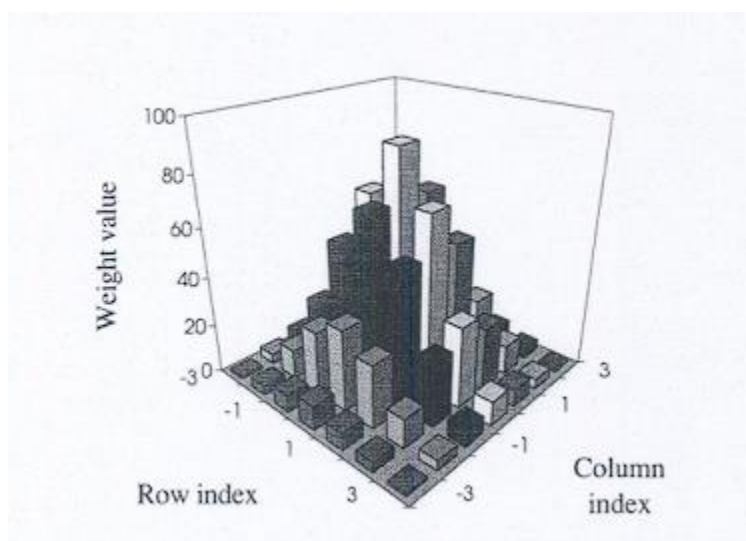
Ova verzija funkcije koristi se za „zaglađivanje“ slika, što u stvari nije ništa drugo do njihovo zamućenje [5]. Vizuelni prikaz ove funkcije dat je na slici 4.14, dok je primer kernela filtera dat na slici 4.15. Na slici 4.16 dat je prikaz kernela u 3 dimenzije, na kom se jasno vidi da je kernel ništa drugo do diksretizacija Gausove funkcije.



Slika 4.14 Dvodimenzionalna Gausova funkcija sa srednjom vrednošću 0 [5]

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

Slika 4.15 Primer  $7 \times 7$  *Gaussian* kernela [5]



Slika 4.16 3D prikaz  $7 \times 7$  kernela [5]

Neka od bitnih svojstava *Gaussian* filtera su:

- U dve dimenzije, ovaj filter je rotaciono simetričan. Ovo znači da je se slika zamućuje podjednako u svim pravcima. Generalno, ivice na slici neće biti orijentisane u nekom unapred određenom smeru, pa samim tim ne postoji neko *a priori* znanje u kom smeru bi trebalo zamutiti sliku. Svojstvo rotacione simetričnosti implicira da ovaj filter neće uticati na potencijalnu detekciju ivica koja može uslediti posle njega [5].
- Ovaj filter zamućuje sliku tako što svaki piksel slike menja težinskom srednjom vrednošću okolnih piksela, tako da težine date susednim pikselima monotono opadaju sa povećanjem njihove distance od centralnog piksela. Ovo svojstvo je važno zbog toga što je ivica lokalna osobina slike, i zamućenje slike takvo da

udaljeni pikseli imaju veliki uticaj na konačni rezultat može rezultovati distorzijom karakteristika slike [5].

- Širina, a samim tim i stepen zamućenja, *Gaussian* filtera određen je parametrom  $\sigma$ , a veza između parametra i zamućenja je jako jednostavna – veće  $\sigma$  znači da je Gausova funkcija koju filter predstavlja šira, a samim tim je i stepen zamućenja veći [5].

Augmentacija slika primenom *Gaussian* filtera rezultovaće trening primerima koji su zamućeni i koji će u našem skupu podataka simulirati zamućenje nastalo naglim pokretima. Ovo se može iskoristiti za poboljšanje otpornosti modela na zamućenje slika prilikom prepoznavanja objekata [2]. Primer slike na kojoj je primenjen *Gaussian* filter data je na slici 4.17.



Slika 4.17 a) Originalna slika; b) Slika nakon primene *Gaussian* filtera [2]

#### 4.4.3 Canny filter

*Canny* filter, poznat i kao *Canny* detektor ivica (engl. *Canny edge detector*) predstavlja prvi izvod Gausove funkcije. Neka je sa  $I[i, j]$  označena slika. Rezultat primene *Gaussian* filtera na slici može se zapisati u obliku:

$$S[i, j] = G[i, j; \sigma] * I[i, j] \quad (4.6)$$

gde je  $\sigma$  parameter koji određuje širinu Gausove funkcije i kontroliše nivo zamućenja slike [5].

Gradijent zamućene slike  $S[i, j]$  može se izračunati korišćenjem  $2 \times 2$  prvom diferencom aproksimacijom (engl. *first-difference approximation*) (slika 4.18), pri čemu se dobijaju dve slike  $P[i, j]$  i  $Q[i, j]$ , koje odgovaraju parcijalnim izvodima po  $x$  i  $y$  koordinati [5]:

$$P[i, j] \approx \frac{S[i, j + 1] - S[i, j] + S[i + 1, j + 1] - S[i + 1, j]}{2} \quad (4.7)$$

$$Q[i, j] = \frac{S[i, j] - S[i + 1, j] + S[i, j + 1] - S[i + 1, j + 1]}{2} \quad (4.8)$$

$$G_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Slika 4.18  $2 \times 2$  prve difference aproksimacije po  $x$  (levo) i po  $y$  koordinati (desno) [5]

U obe formule je deljenje sa 2 uvedeno kako bi se računanje parcijalnih izvoda po  $x$  i  $y$  koordinati vršilo u istoj tački. Magnituda i orijentacija gradijenta mogu se izračunati standardnim formulama:

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (4.9)$$

$$\theta[i, j] = \arctan\left(\frac{Q[i, j]}{P[i, j]}\right) \quad (4.10)$$

Magnituda slike  $M[i, j]$  će imati visoke vrednosti tamo gde je gradijent slike veliki, ali ovo samo po sebi nije dovoljno za detekciju ivica, jer je inicijalni problem iz traženja tačaka gde postoji promena u intenzitetu piksela samo transformisan u problem traženja lokalnih maksimuma magnitude gradijenta. Kako bi se odredile ivice, „izbočine“ dobijene računanjem magnitude gradijenta (ovde se misli na izgled magnitude gradijenta ukoliko se dobijena slika posmatra kao reljef, gde je visina reljefa u odgovarajućoj tački predstavljena magnitudom gradijenta u toj tački) moraju da se istanje kako bi ostale samo tačke od interesa, odnosno lokalni maksimumi. Ovaj proces naziva se potiskivanje nemaksimuma (engl. *nonmaxima suppression*), i rezultat procesa su istanjene ivice [5].

Potiskivanje nemaksimuma istanjuje „izbočine“ magnitude gradijenta  $M[i, j]$  tako što potiskuje sve vrednosti duž linije gradijenta koje nisu dovoljno visoke. Algoritam prvo redukuje sve uglove gradijenta  $\theta[i, j]$  na jedan od četiri sektora prikazanih na slici 4.19,

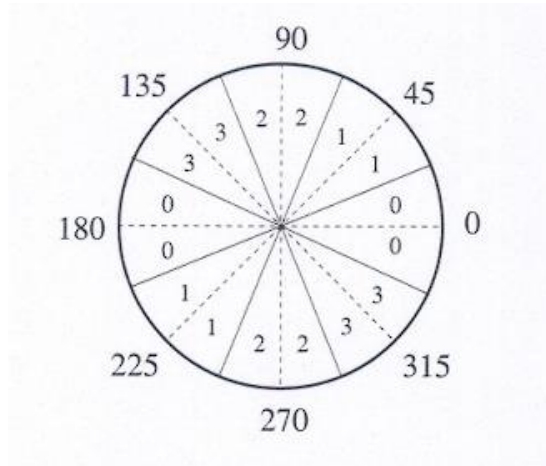
$$\zeta[i, j] = \text{Sector}(\theta[i, j]) \quad (4.11)$$

Algoritam prevlači  $3 \times 3$  prozor preko slike magnitude gradijenta  $M[i, j]$ . U svakoj tački, element  $M[i, j]$  koji se nalazi u sredini prozora poredi se sa svoja dva suseda duž linije gradijenta koji su određeni vrednošću sektora  $\zeta[i, j]$ . Ukoliko vrednost  $M[i, j]$  koja se nalazi u centru prozora nije veća od vrednosti magnitude oba suseda koja se posmatraju, onda se vrednost  $M[i, j]$  postavlja na 0. Ovaj proces dovodi do toga da su ivice slike širine jednog piksela [5].

Neka je sa

$$N[i, j] = nms(M[i, j], \zeta[i, j]) \quad (4.12)$$

označen process potiskivanja nemaksimuma. Nenula vrednosti  $N[i, j]$  odgovaraju količini kontrasta na slici u odgovarajućim tačkama, kada se porede trenutni i susedni piksel. Uprkos zamućenju slike u inicijalnom koraku primenom *Gaussian* filtera, slika dobijena potiskivanjem nemaksimuma sadržaće veliki broj lažnih ivica koji su posledica šuma i finijih tekstura na slici. Kontrast lažnih ivica trebalo bi da bude manji u odnosu na kontrast pravih ivica [5].



Slika 4.19 Sektori u koje se klasifikuju gradijenti slike na osnovu njihovog ugla [5]

Tipična procedura za smanjenje broja lažnih ivica na rezultatu potiskivanja nemaksimuma je primena granične vrednosti. Sve vrednosti koje su ispod zadate granične vrednosti (engl. *threshold*) postavljaju se na nulu. Rezultat ove procedure je slika koja sadrži samo ivice originalne slike  $I[i, j]$ . Još uvek će biti nekih lažnih ivica zbog toga što je granična vrednost  $\tau$  previše niska

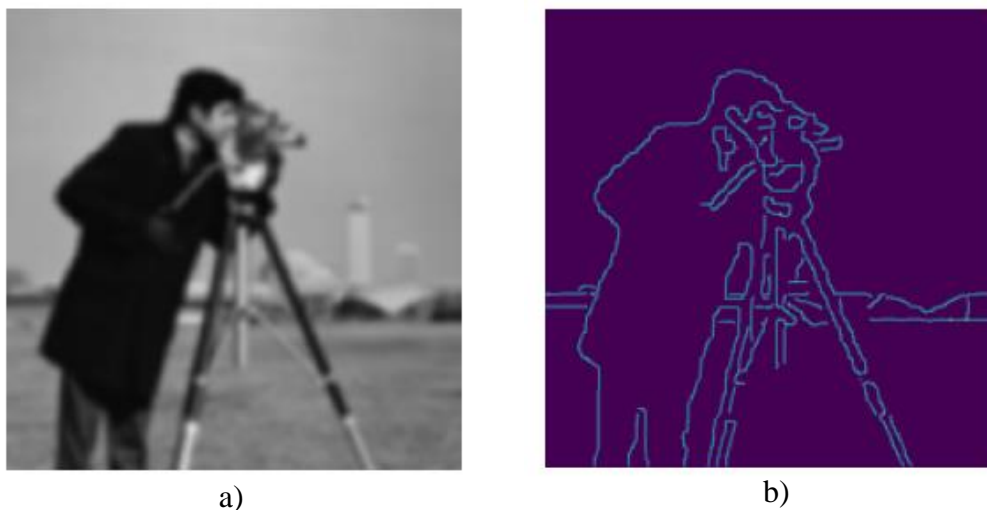
(lažne pozitivne ivice – engl. *false positives*), i delovi nekih ivica mogu nedostajati (lažne negativne ivice – engl. *false negatives*) što zbog manjeg kontrasta na tom delu ivice nastalom kao posledica postojanja senki ili pak zbog činjenice da je granična vrednost  $\tau$  previše visoka. Selektovanje odgovarajuće granične vrednosti može biti teško i obično zahteva dosta isprobavanja. Zbog toga se obično umesto jedne koriste dve granične vrednosti, što predstavlja efektivniji način odbacivanja lažnih ivica [5].

Ovaj algoritam na rezultate dobijene potiskivanjem nemaksimuma primenjuje dve granične vrednosti  $\tau_1$  i  $\tau_2$ , pri čemu je najčešće  $\tau_2 \approx 2\tau_1$ , čime se dobijaju dve nove slike  $T_1[i, j]$  i  $T_2[i, j]$ . Pošto je slika  $T_2$  formirana korišćenjem veće granične vrednosti, sadržaće manji broj lažnih ivica, ali isto tako može sadržati i veliki broj rupa u ivicama, odnosno lažno negativnih piksela. Tu nastupa slika nastala primenom manje granične vrednosti. Algoritam povezuje ivice sa slike  $T_2$  u konture. Kada dođe do kraja ivice, algoritam posmatra sliku  $T_1$  i 8-okolinu piksela na slici  $T_1$  koji odgovara pikselu sa slike  $T_2$  na kom se nalazi kraj ivice. Ukoliko u toj okolini postoji deo ivice koji ne postoji na slici  $T_2$ , ivica se povezuje. Ovakav algoritam znatno olakšava biranje graničnih vrednosti [5].

Koraci za primenu *Canny* detektora ivica su [5]:

- Zamutiti sliku korišćenjem *Gaussian* filtera;
- Izačunati magnitudu gradijenta i njegovu orijentaciju korišćenjem konačnih diferencnih aproksimacija za računanje parcijalnih izvoda;
- Priminiti potiskivanje nemaksimuma na magnitudu gradijenta;
- Priminiti dve granične vrednosti za pronalaženje i povezivanje ivica.

Što se tiče primene ovog algoritma za augmentaciju podataka kod slika, slično kao i sa *Sobel* filterom, treniranje modela na ovakvim slikama može pomoći mreži da bolje nauči da odredi ivice objekata [2]. Primer primene *Canny* algoritma može se videti na slici 4.20.



Slika 4.20 a) Originalna slika; b) Slika nakon primene *Canny* filtera [2]

## 5. Zaključak

U situacijama kada je skup podataka sa kojim radimo ograničen, ili kada je kreiranje potpuno novih test primera skupo i vremenski zahtevno, augmentacija podataka je definitivno nešto što treba razmotriti. Augmentacijom podataka izvući ćemo maksimum iz podataka koje posedujemo, bez potrebe za dodatnim troškovima koji bi bili neophodni za akviziciju dodatnih primera.

Naravno, neophodno je biti oprezan sa načinom na koji se podaci augmentuju – potrebno je razmotriti postojeće slike, problem na koje se te slike odnose i odlučiti koje tehnike i metode augmentacije podataka su pogodne za datu primenu. Ovo može biti izazovno, ali dobro izvedena augmentacija podataka gotovo sigurno garantuje veću uspešnost modela koji treniramo.

## Literatura

- [1] T. Kumar, A. Mileo, R. Brennan, M. Bendeache, „Image Data Augmentation Approaches: A Comprehensive Survey and Future directions“, *arXiv preprint*, 2023.
- [2] K. Alomar, H. I. Aysel, X. Cai, „Data Augmentation in Classification and Segmentation: A Survey and New Strategies“, *Journal of Imaging*, vol. 9, 2023.
- [3] C. Shorten, T. M. Khoshgoftaar, „A survey on Image Data Augmentation for Deep Learning“ *Journal of Big Data*, vol. 6:60, 2019.
- [4] A. Hernández-García, „Data augmentation and image understanding“, PhD thesis, Institute of Cognitive Science, University of Osnabrück, 2020.
- [5] R. Jain, R. Kasturi, B. G. Schunck, „Machine Vision“, *McGraw-Hill*, pp. 123-137, 147-148, 169-173, 1995.