

Predavanje 7

Kolegij: Osnove strojnog učenja

Sadržaj

- Digitalna slika i 2D konvolucija
- Konvolucijske neuronske mreže
 - Motivacija
 - Struktura
 - Konvolucijski sloj
 - Sloj sažimanja
 - Primjena na klasifikaciju CIFAR-10 podatkovnog skupa
- Popularne arhitekture konvolucijskih neuronskih mreža
- Poboljšanje procesa učenja (konvolucijskih) neuronskih mreža
 - Sloj s nasumičnim izbacivanjem neurona
 - Augmentacija skupa podataka
 - Učenje prijenosom

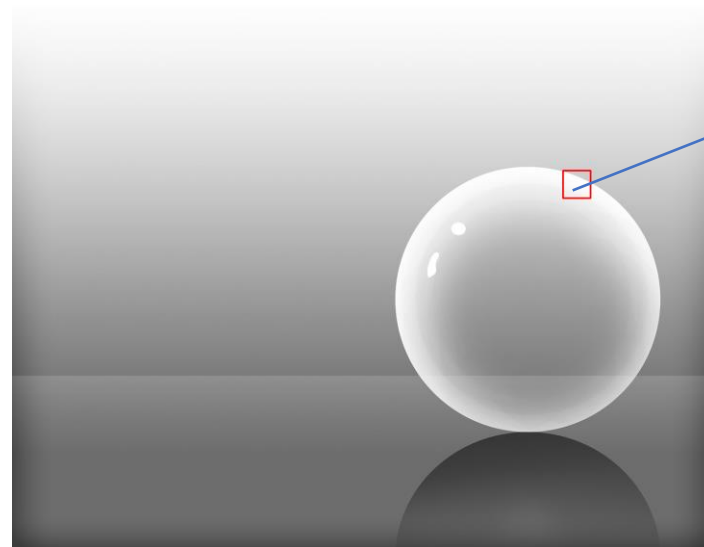
Digitalna slika i 2D konvolucija

Digitalna slika

- Prisjetimo se kako je pohranjena digitalna slika na računalu
- Slika u sivim tonovima može se prikazati kao matrica dimenzija $h \times w$ koja sadrži vrijednosti u intervalu od $[0,255]$

		w				
h		0-255	0-255	0-255	0-255	0-255
		0-255	0-255	0-255	0-255	0-255
		0-255	0-255	0-255	0-255	0-255
		0-255	0-255	0-255	0-255	0-255
		0-255	0-255	0-255	0-255	0-255

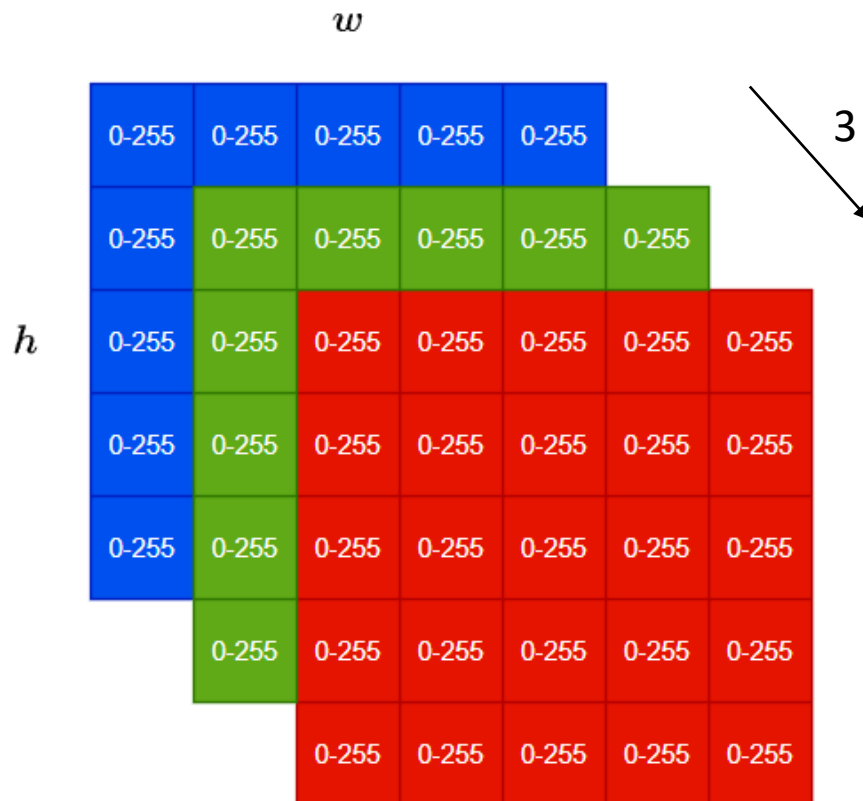
Primjer



```
[ [208 207 207 ... 207 207 207]
  [202 203 205 ... 207 207 207]
  [242 233 220 ... 207 207 207]
  ...
  [236 237 237 ... 247 248 248]
  [236 236 237 ... 248 248 248]
  [236 236 237 ... 248 248 248] ]
```

Digitalna slika

- Slika u boji najčešće se prikazuje u RGB zapisu (R – crveni kanal, G – zeleni kanal, B – plavi kanal)
- Slika se predstavlja kao matrica dimenzija $h \times w \times 3$ s vrijednostima u intervalu [0-255]



2D konvolucija

- **Filtriranje** slike odnosi se na primjenu matematičkog operatora na piksele slike s ciljem poboljšavanja slike, izoštravanja slike, uklanjanja neželjenih šumova iz slike i sl.
- **2D konvolucija** je matematička operacija koja se koristi u obradi slika, a temelji se na konceptu konvolucije → to je specijalan slučaj filtriranja slike
- Digitalnu sliku tj. njezine elemente (piksele) predstavljamo ulaznom matricom
- Konvolucija koristi matematički operator poznat kao "**kernel**" ili "**maska**" koja se primjenjuje na određeni piksel slike uzimajući u obzir njegovo susjedstvo (**prostorno filtriranje**)
- Maska se također predstavlja matricom unaprijed definirane (fiksne) veličine (npr. 3x3) koja se pozicionira na ulaznu matricu te se primjenjuju operacija zbrajanja i množenja na odgovarajuće piksele koji se preklapaju s maskom
- „Kretanjem” maske preko cijele ulazne matrice stvara se nova matrica tj. filtrirana slika
- U području neuronskih mreža ova matrica se naziva "**mapa značajki**" jer sadrži informacije o odgovarajućim oblicima, rubovima i drugim značajkama prisutnim u originalnoj slici → ovisno kakva maska je primijenjena
- Primjena 2D konvolucije u dubokim neuronskim mrežama omogućuje **automatsko učenje značajki**, što može biti vrlo korisno u zadacima kao što su klasifikacija slika, detekcija objekata i segmentacija slika

2D konvolucija

Ulazna slika

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}

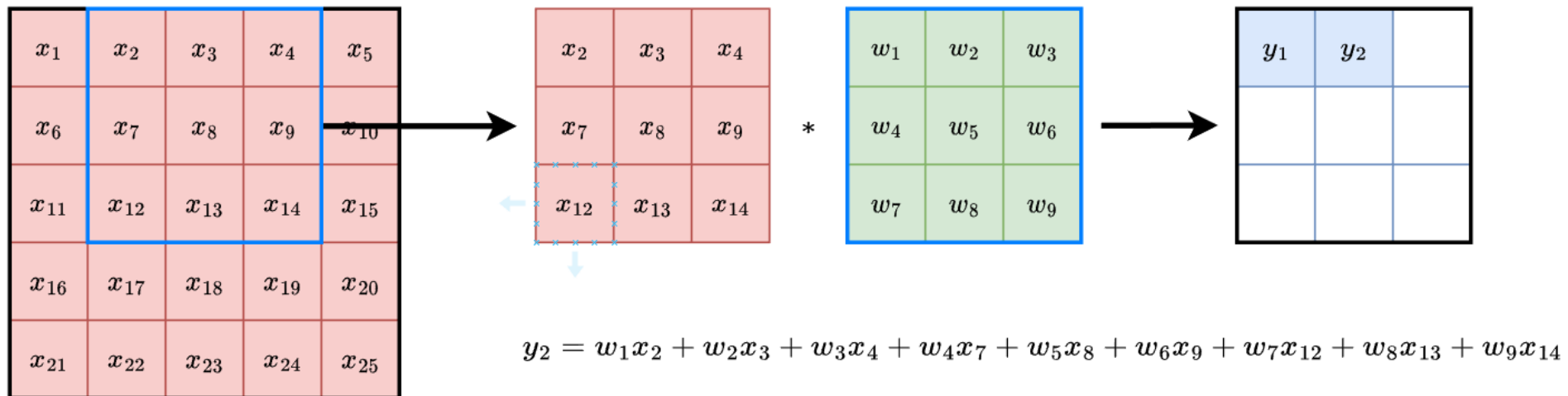
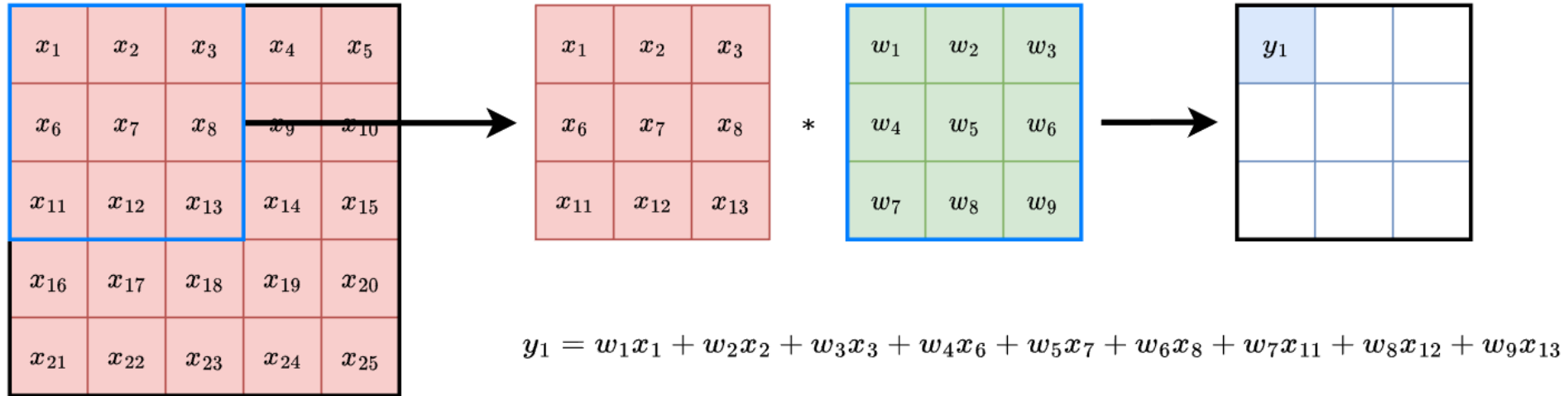
Maska 3x3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

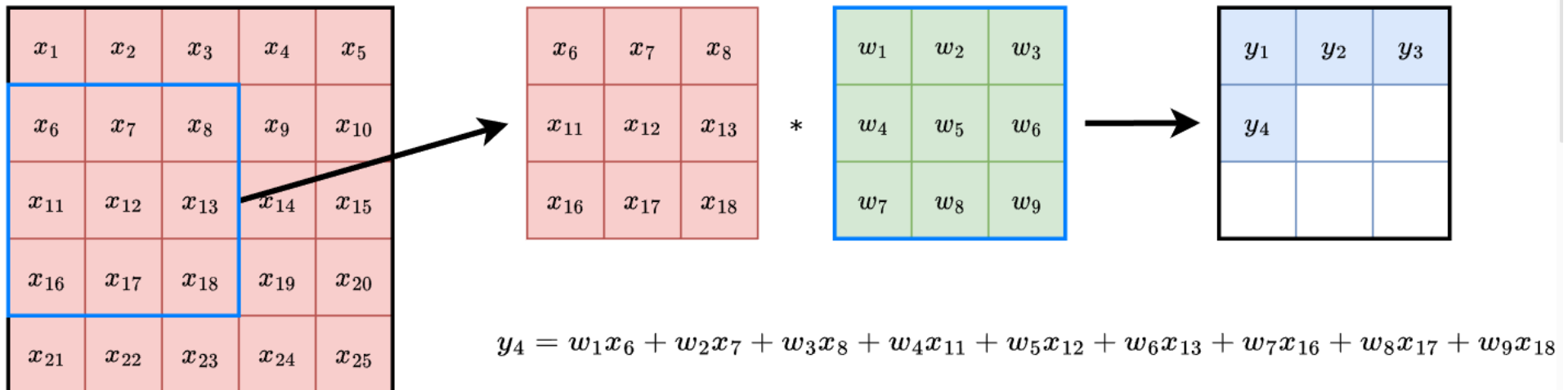
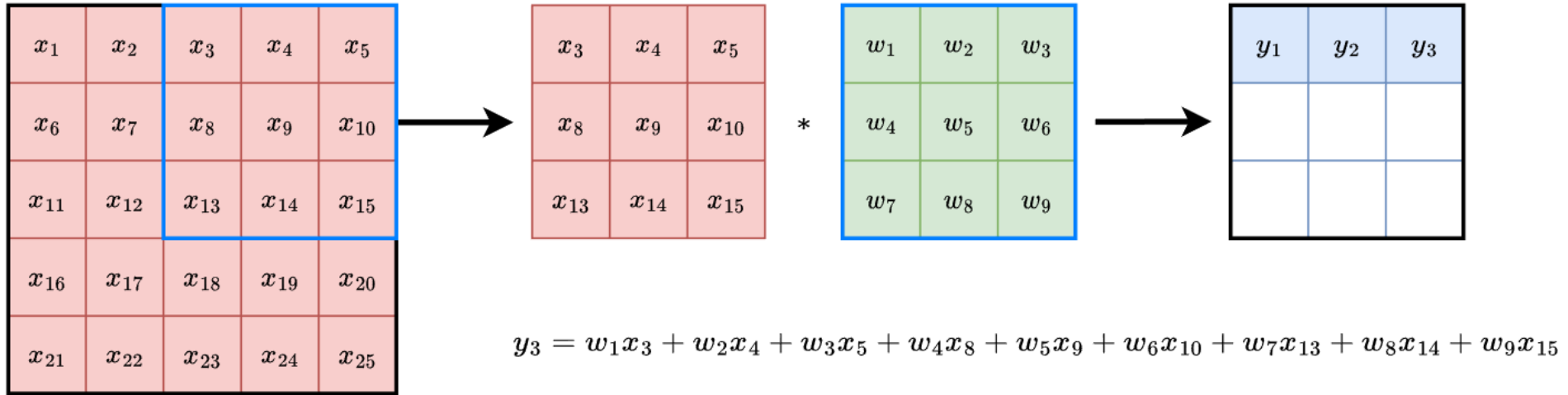
Filtrirana slika

y_1	y_2	y_3
y_4	y_5	y_6
y_7	y_8	y_9

2D konvolucija



2D konvolucija



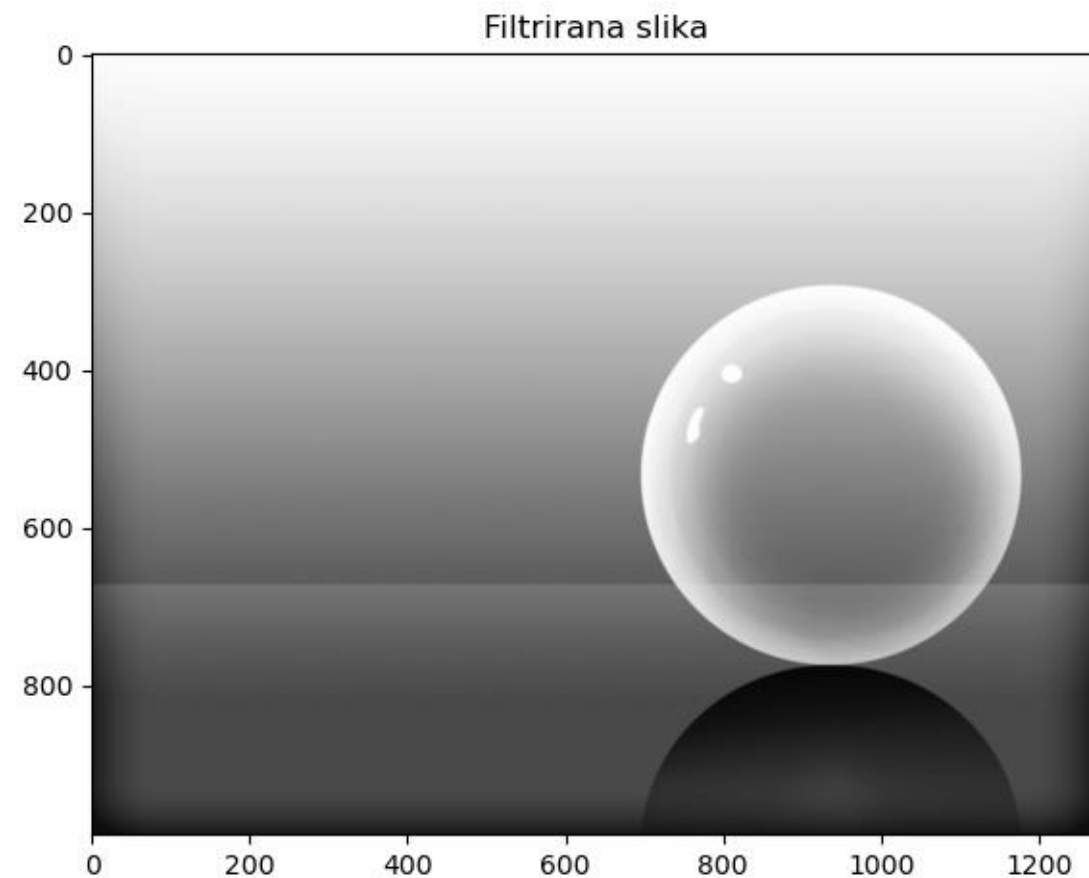
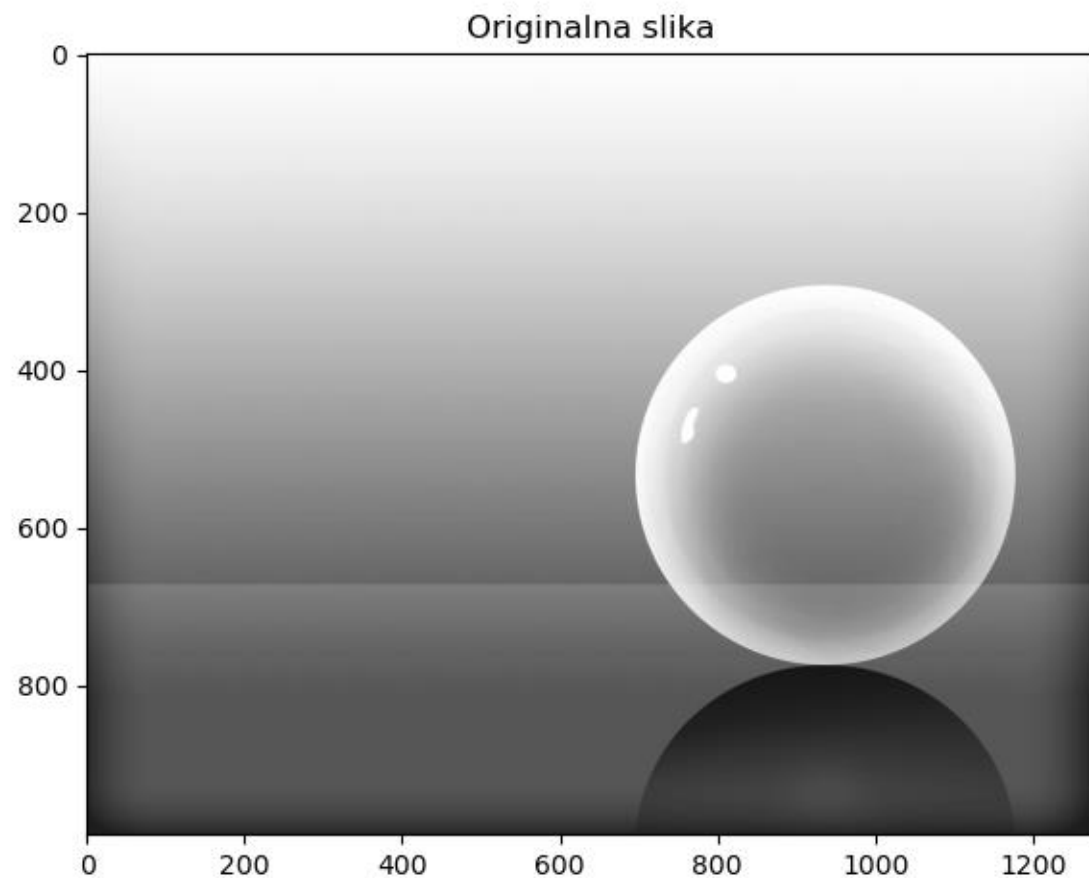
2D konvolucija primjer - zamućivanje

- Zamućivanje (engl. *blurring*) slike može se postići s ovakvom 3x3 maskom:

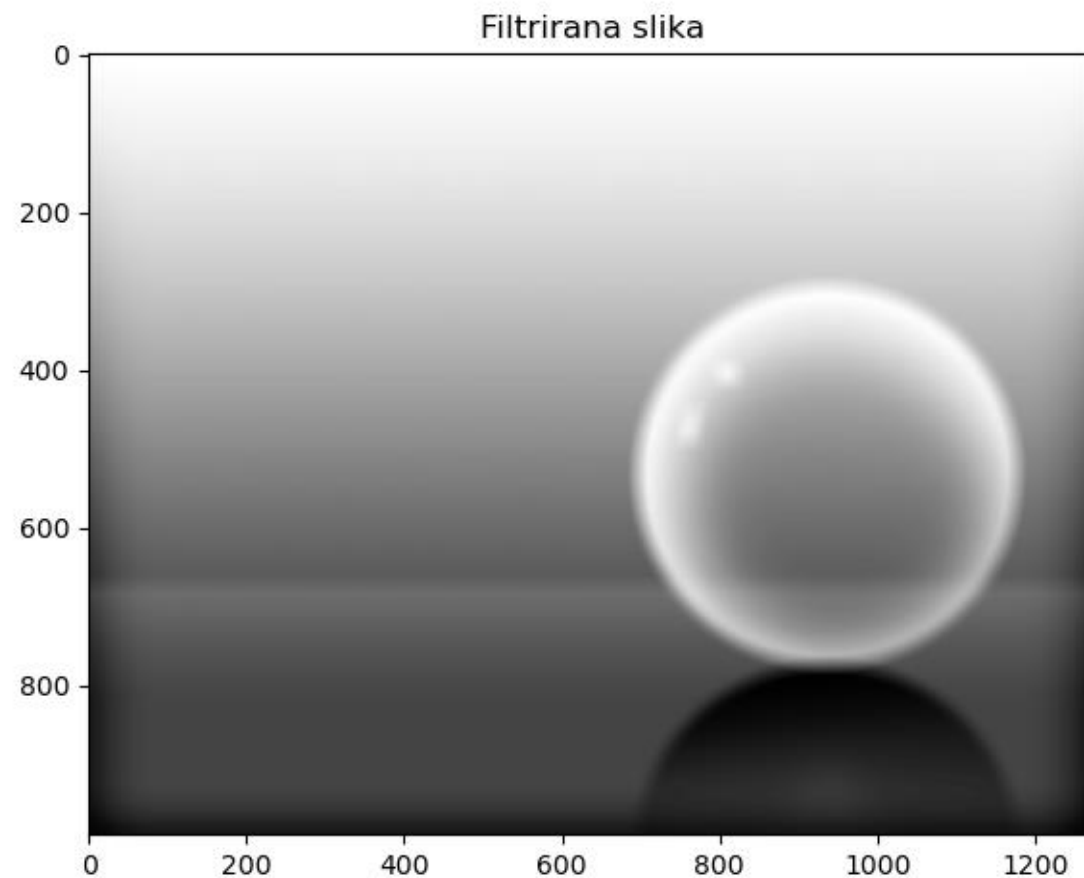
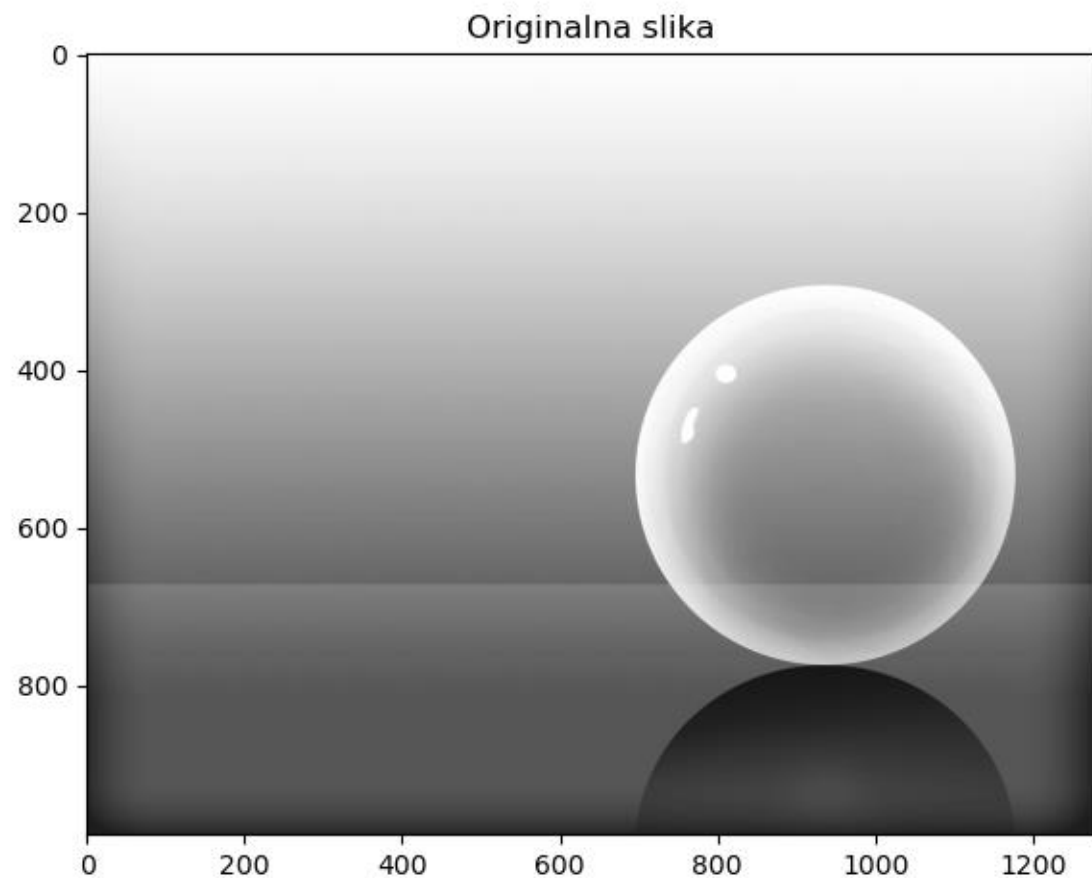
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Što se događa tijekom filtriranja na svakoj lokaciji ulazne slike?
- Što se događa s povećanjem dimenzije maske (npr. 25x25) i kako bi izgledala takva maska?
- Za što se u praksi koristi ovakvo filtriranje? Gdje biste ga primijenili?

2D konvolucija primjer – zamućivanje (3x3)



2D konvolucija primjer – zamućivanje (25x25)



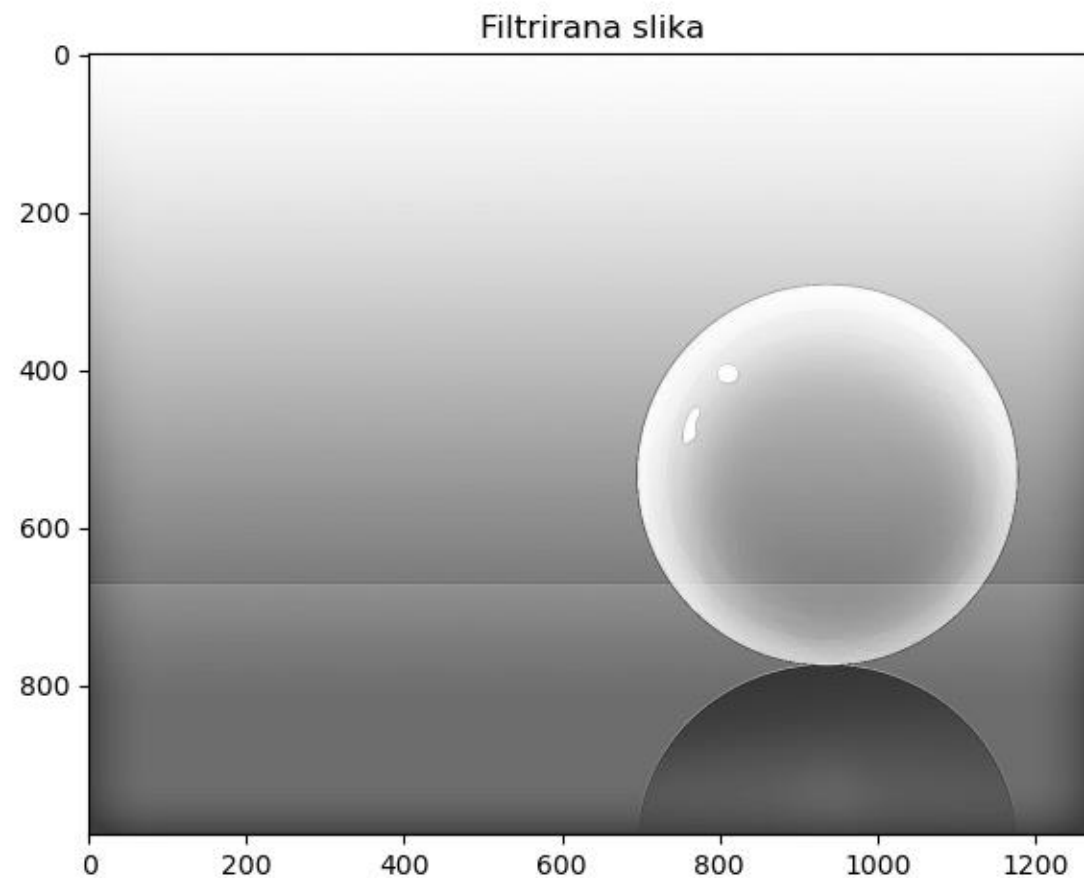
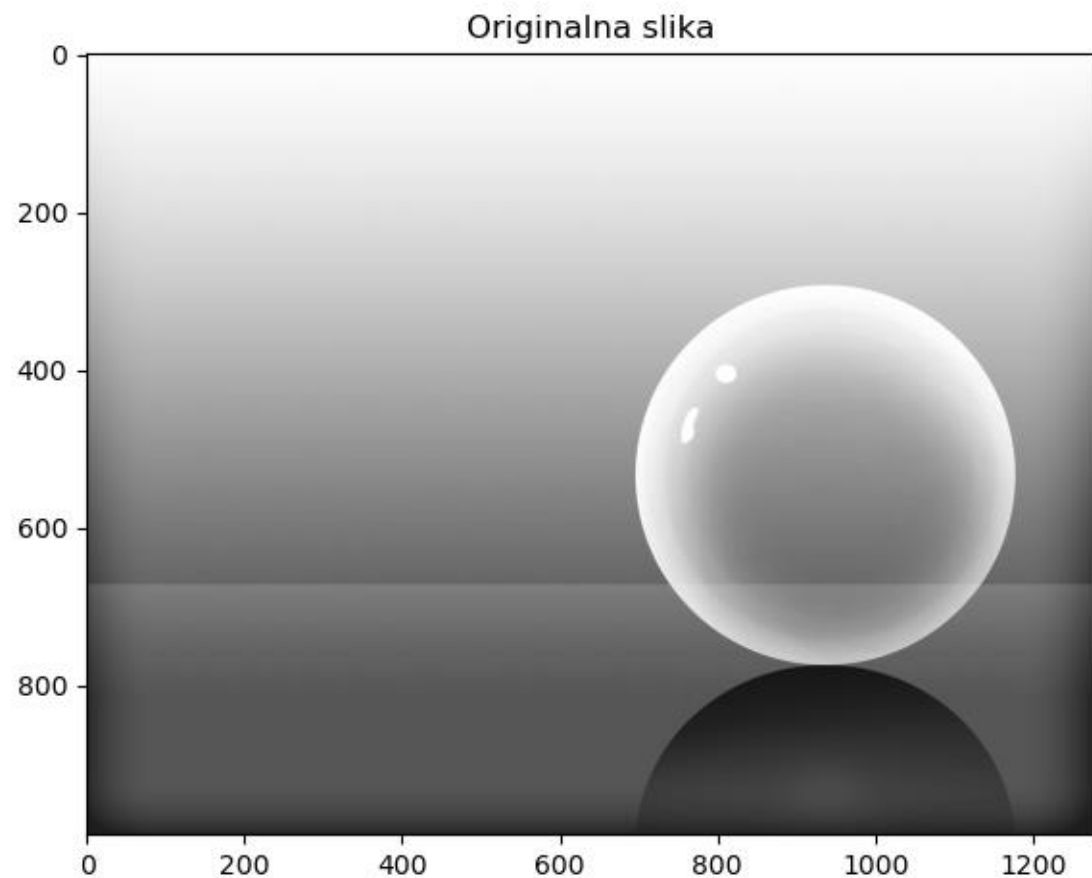
2D konvolucija primjer – izoštravanje

- Izoštravanje slike može se postići s ovakvom 3x3 maskom:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Što se događa tijekom filtriranja na svakoj lokaciji ulazne slike?

2D konvolucija primjer – izoštravanje



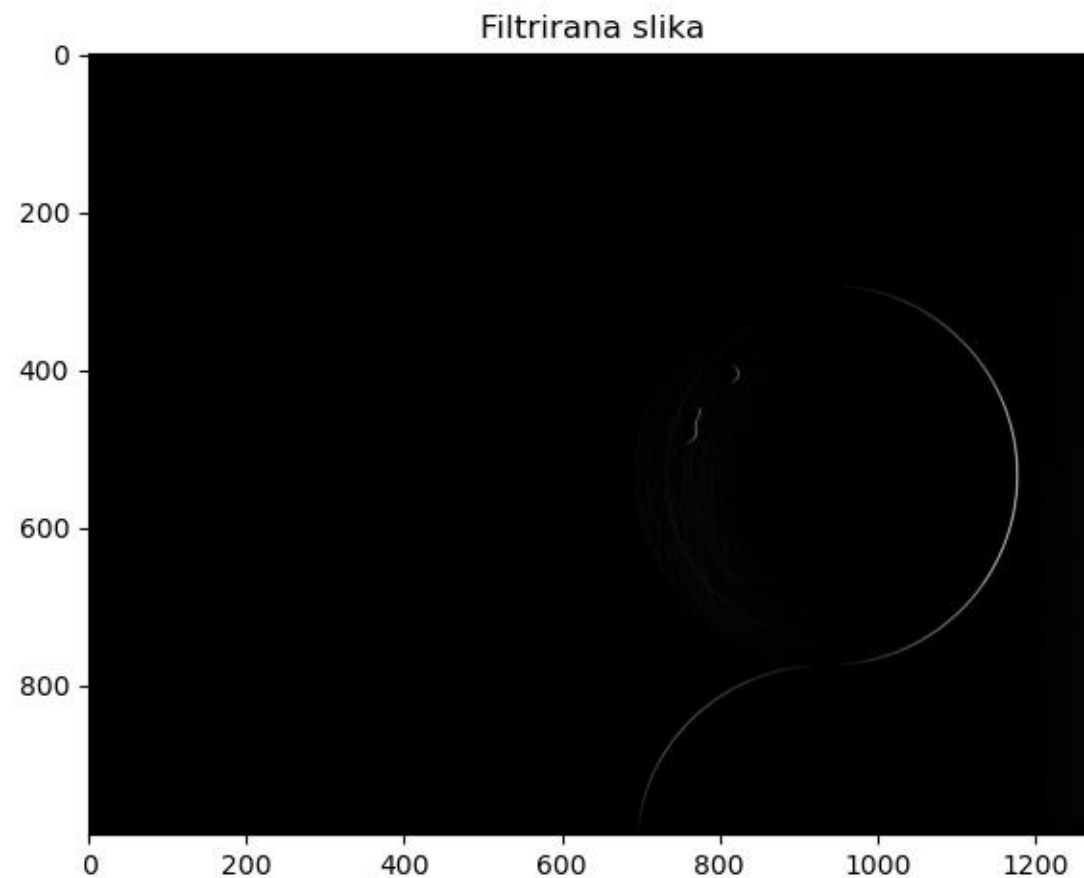
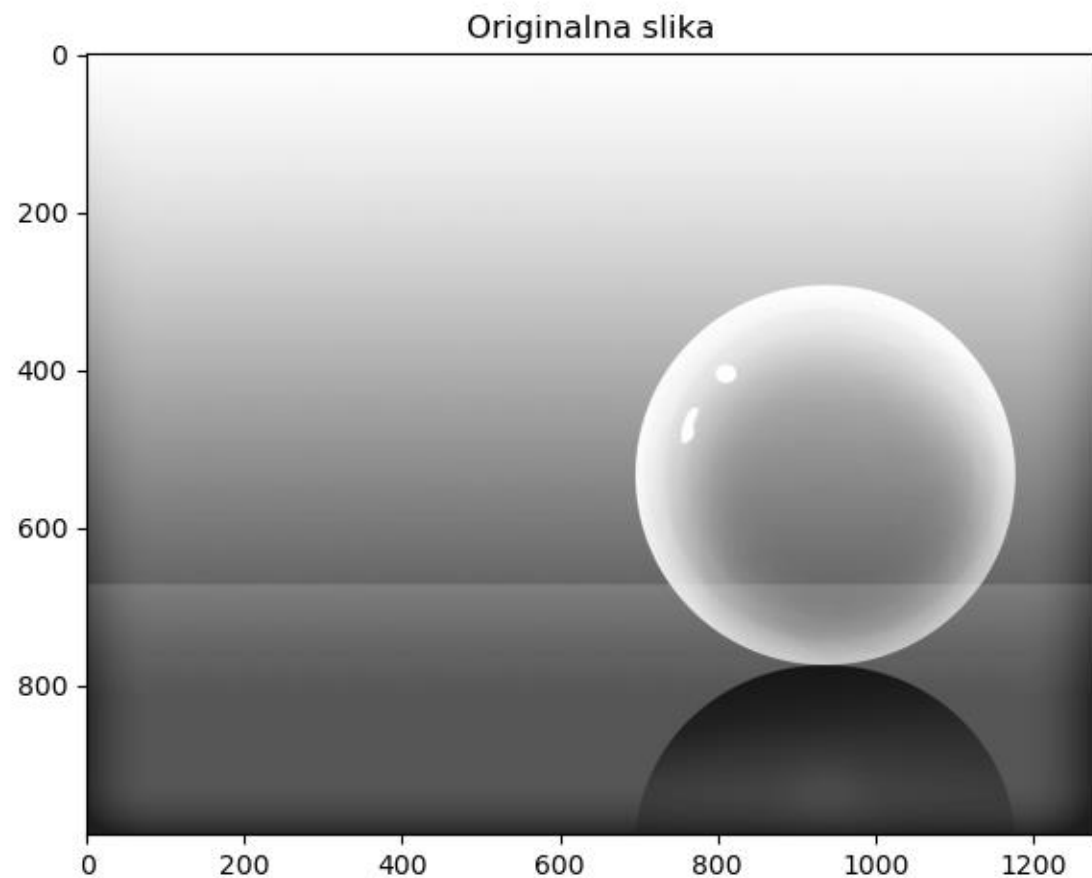
2D konvolucija primjer – vertikalni rubovi

- Naglašavanje vertikalnih rubova na slici može se postići s ovakvom 3x3 maskom:

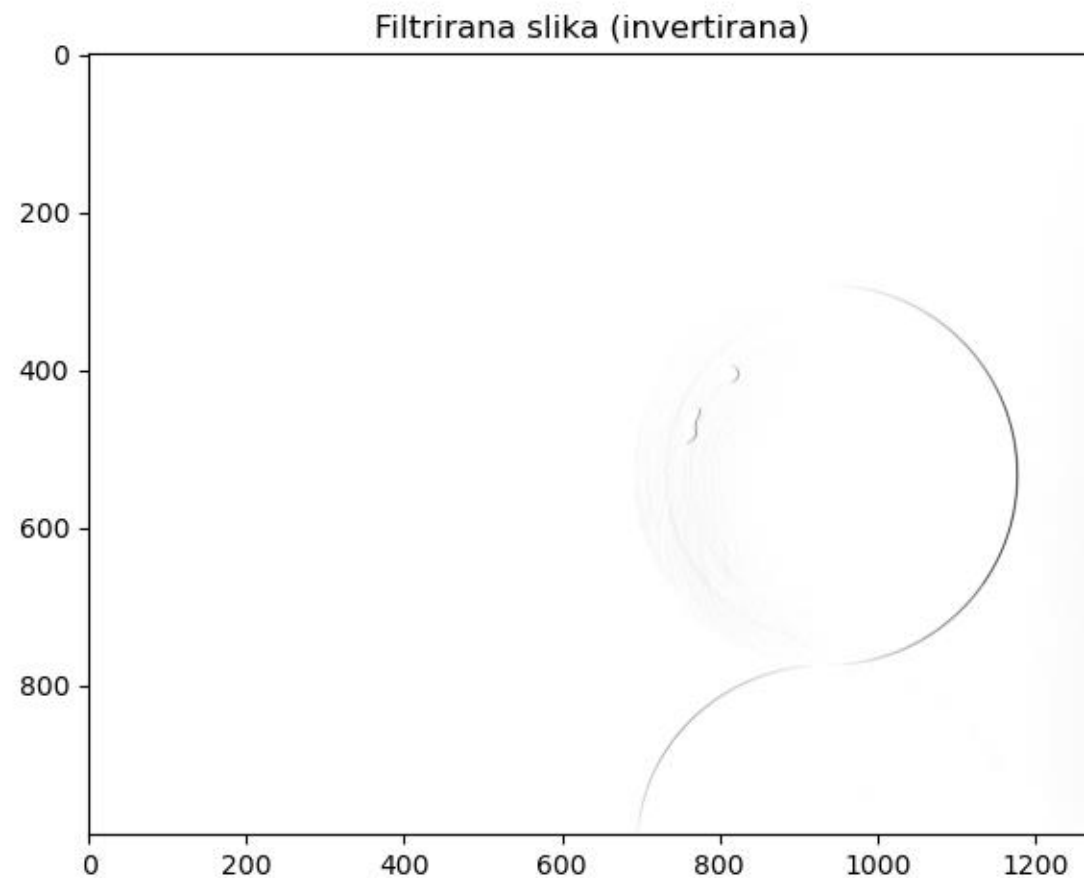
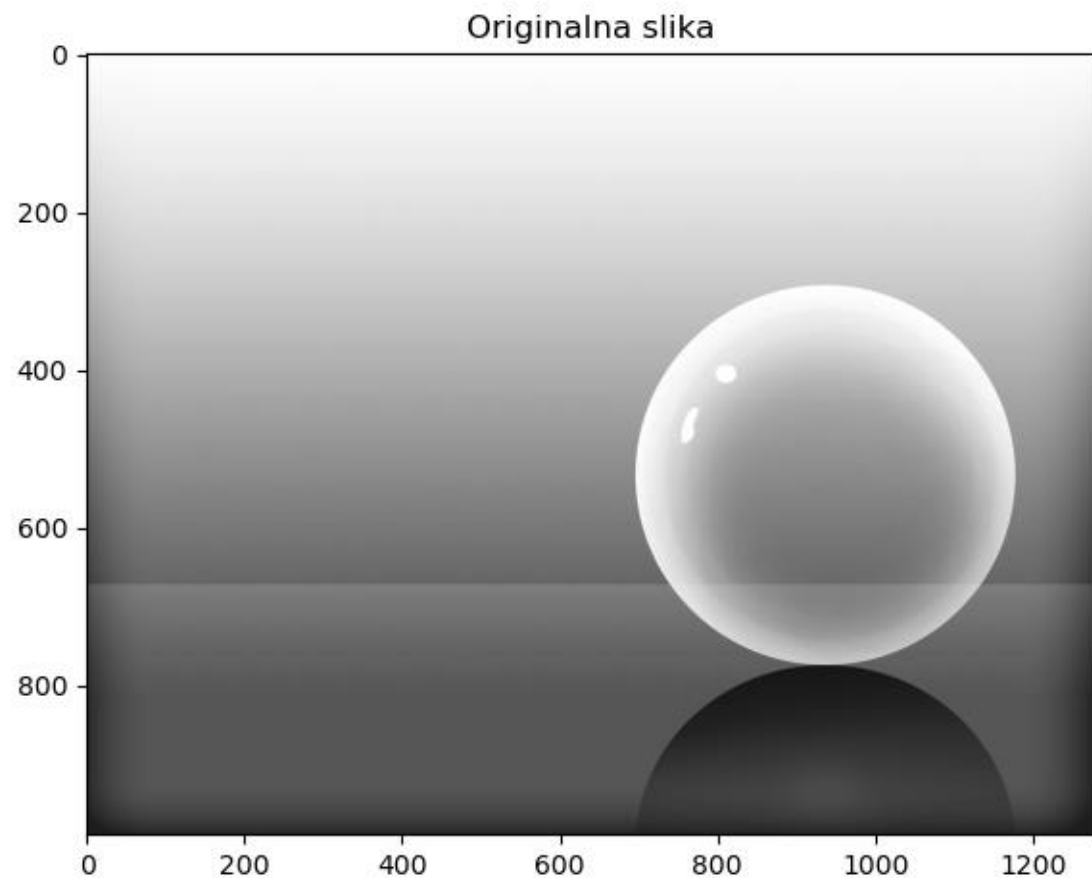
$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

- Imate li ideju kako izdvojiti horizontalne rubove na slici?

2D konvolucija primjer – vertikalni rubovi



2D konvolucija primjer – vertikalni rubovi



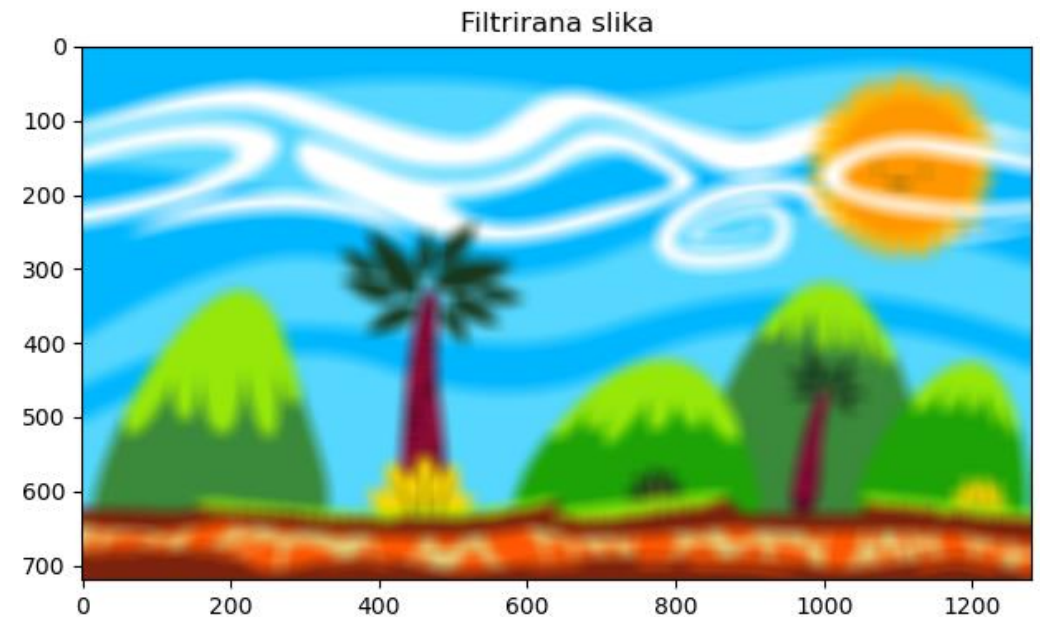
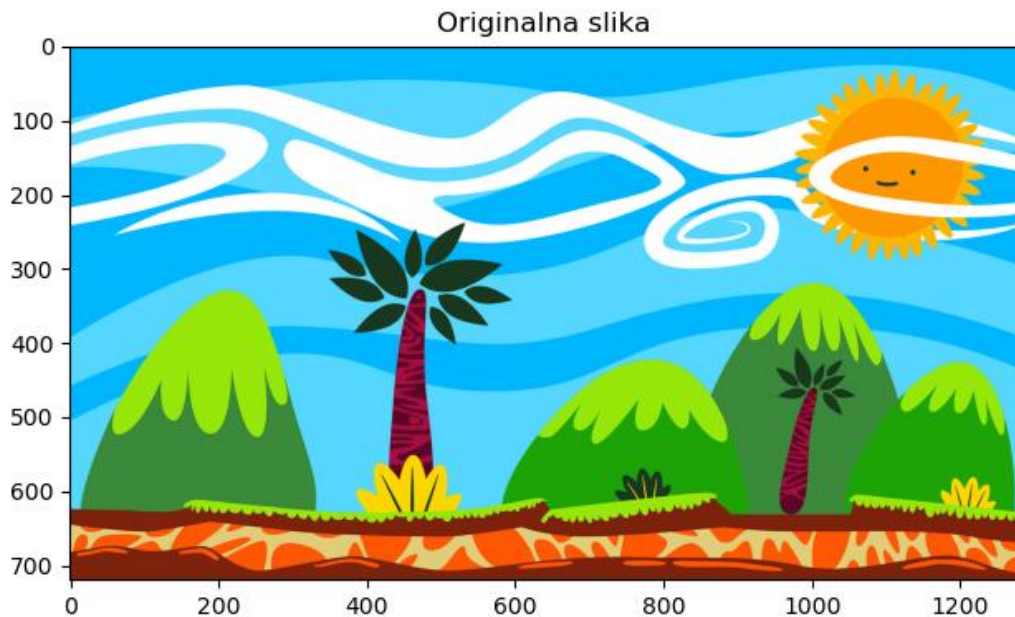
2D konvolucija primjer

- Što je rezultat 2D konvolucije neke digitalne slike s ovakvom maskom ?

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2D konvolucija primjer

- Na sličan način moguće je primijeniti 2D konvoluciju na RGB sliku – pri tome se maska primjenjuje na svaki kanal zasebno te se rezultat ponovo spaja u RGB zapis

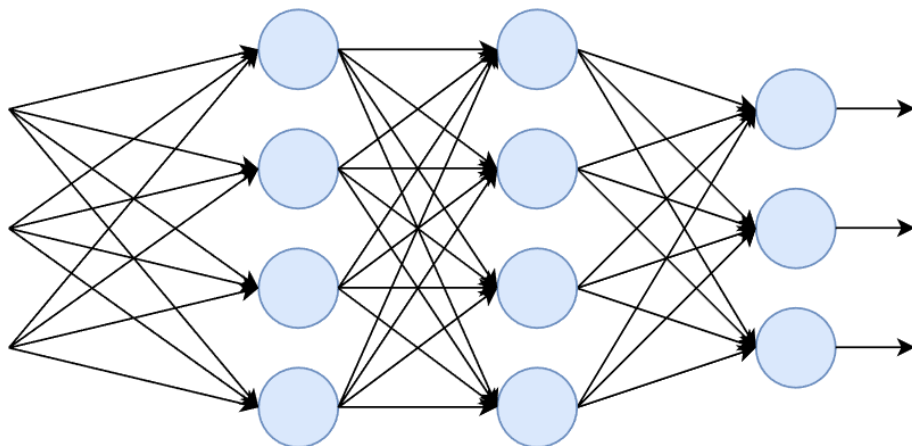


Konvolucijske neuronske mreže

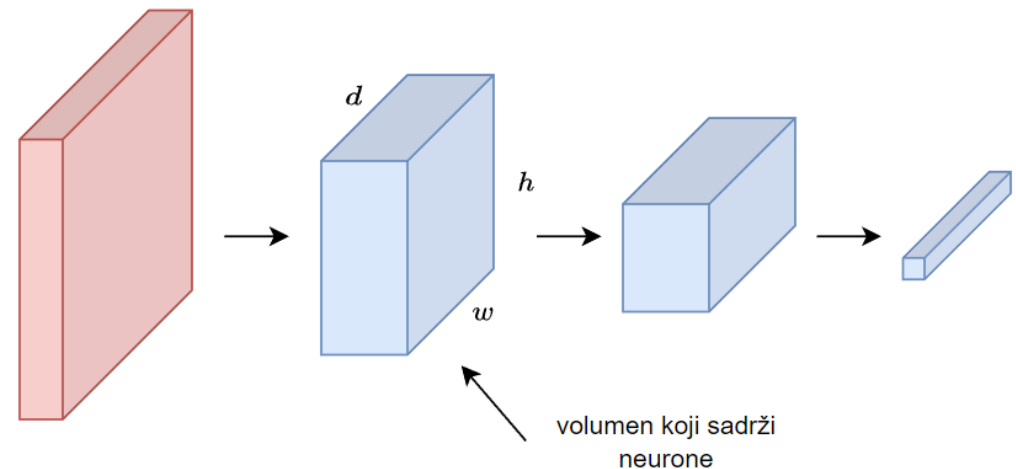
Konvolucijska neuronska mreža (CNN)

- **Konvolucijske neuronske mreže** (engl. *Convolutional Neural Networks* - CNN) su vrsta neuronskih mreža koja se koristi za obradu podataka s prostornim rasporedom, poput primjerice digitalnih slika
- One su osmišljene s ciljem oponašanja vizualnog sustava sisavaca gdje se informacije o objektima iz okoline obrađuju pomoću niza slojeva neurona
- Ti slojevi zadržavaju prostorni raspored informacija (npr. gdje se određena značajka nalazi na slici) i omogućuju mreži da prepozna obrasce poput rubova, oblika i složenijih struktura unutar podataka.

Potpuno povezana neuronska mreža



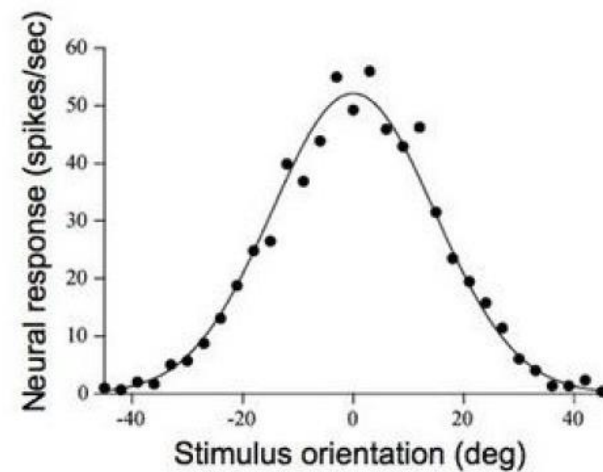
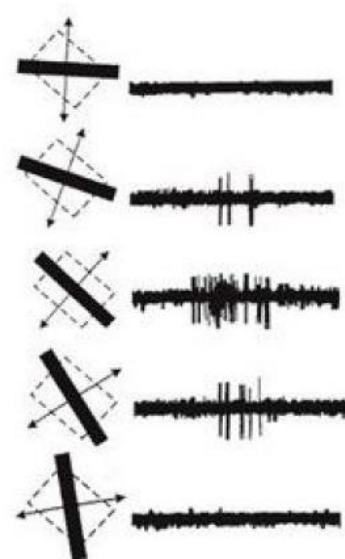
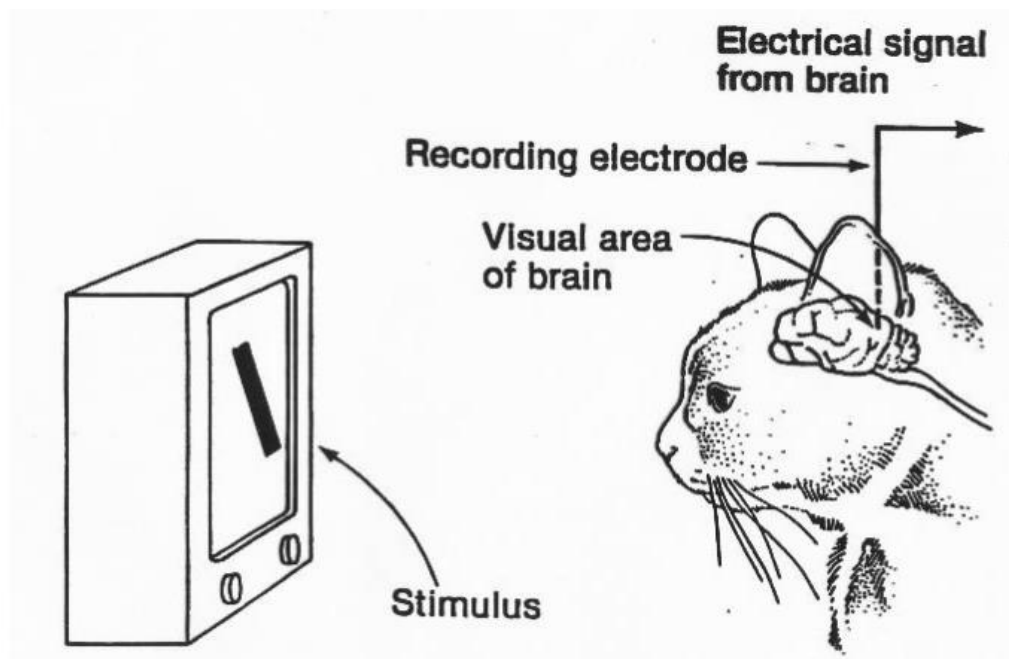
Konvolucijska neuronska mreža



D.H. Hubel i T.N. Wiesel (1959.)

- Hubel i Wiesel su dvojica neuroznanstvenika koji su 1960-ih godina provodili eksperimente na mačkama kako bi istražili vizualni sustav sisavaca
- Njihova istraživanja bila su ključna za razumijevanje načina na koji mozak obrađuje vizualne informacije
- Naime, otkrili su da postoji specijalizacija neurona u vizualnom korteksu u smislu da se aktiviraju na neke određene vizualne značajke poput orijentacije, duljine i sl.
- Na primjer, neki neuroni u vizualnom korteksu aktiviraju se samo kada se prikaže linija koja se kreće u određenom smjeru, dok drugi neuroni reagiraju samo na određenu duljinu linije
- Nadalje, otkrili su da se informacije o vizualnim značajkama obrađuju hijerarhijski gdje se jednostavnije značajke (npr. linije i rubovi) obrađuju u nižim razinama vizualnog korteksa, dok se složenije značajke (npr. objekti i lica), obrađuju u višim razinama

D.H. Hubel i T.N. Wisel (1959.)

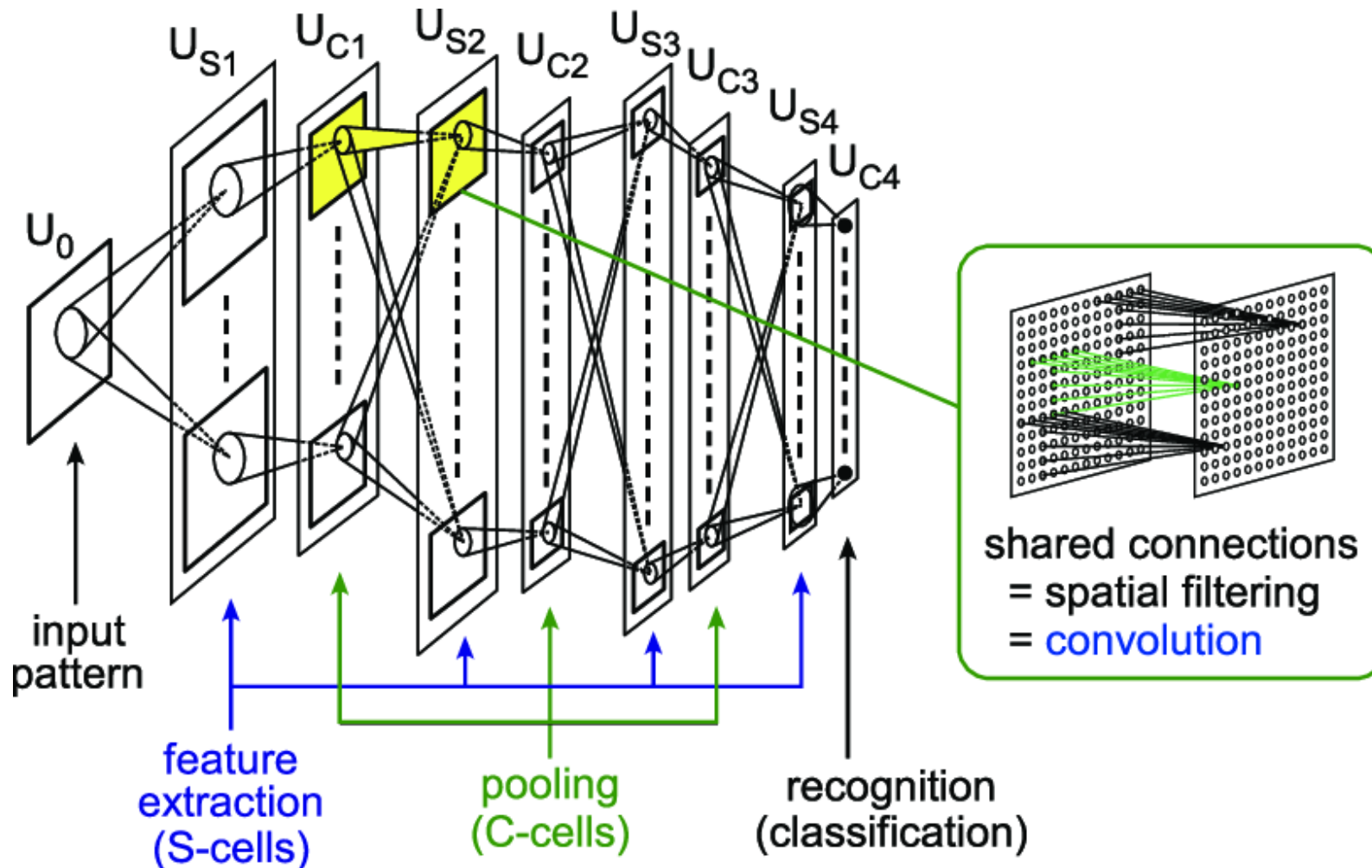


[Izvor](#)

Neocognitron (1979.)

- Autor Fukushima je bio inspiriran biološkim istraživanjima Hubela i Wiselate je kreirao **Neocognitron**
- To je jedan od prvih pokušaja simuliranja vizualnog korteksa pomoću umjetnih neuronskih mreža
- Neocognitron je bio koncipiran kao neuronska mreža s više slojeva koja uči prepoznavati složenije vizualne značajke u višim slojevima mreže, slično kao što se vjeruje da vizualni korteks obrađuje vizualne informacije
- Sjetite se da tada još uvijek ne postoji efikasan algoritam za učenje mreža; ovdje je primijenjen određeni oblik nenadziranog učenja
- Primjena je bila na prepoznavanje rukom pisanih brojeva i ostale probleme raspoznavanja uzoraka
- Inspiracija za konvolucijske neuronske mreže

Neocognitron (1979.)



Arhitektura mreže

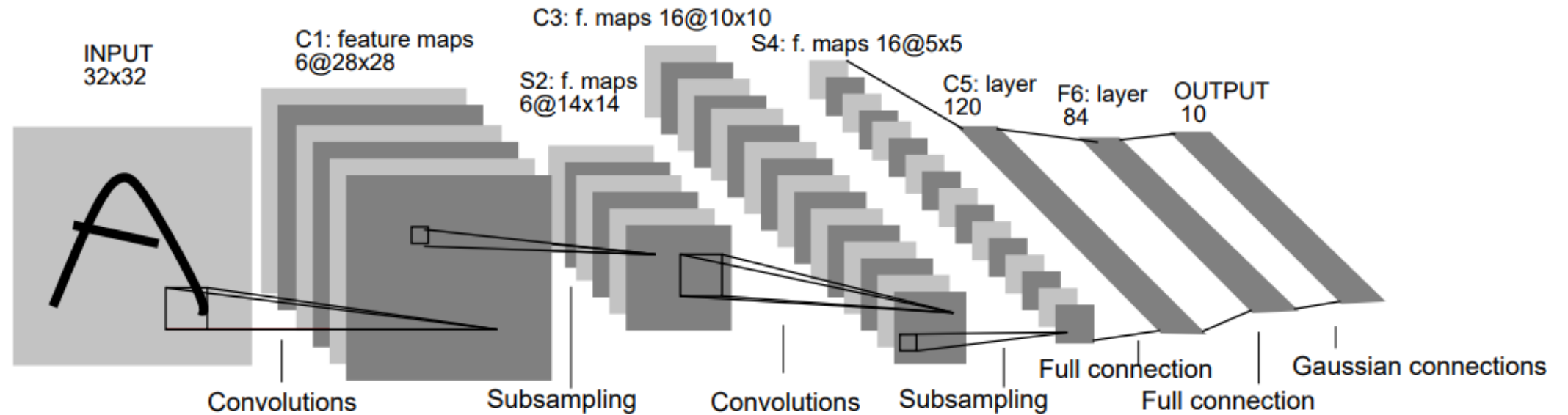
S i C slojevi

Izvor: Recent advances in the deep CNN neocognitron, 2019.

LeNet-5 (1998.)

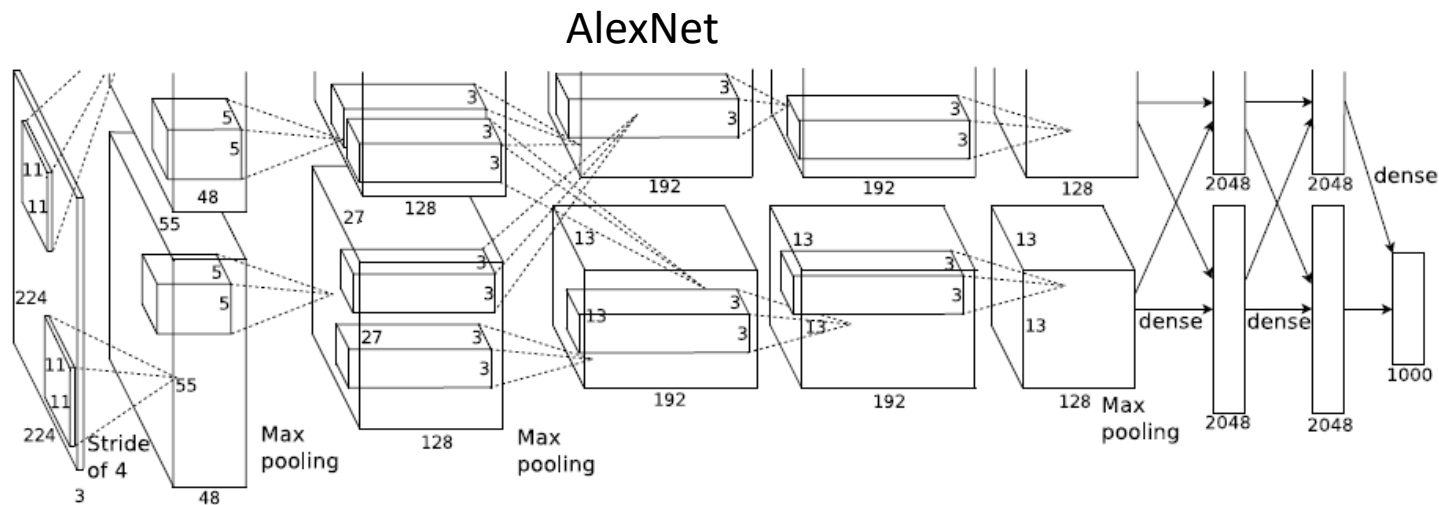
- Znanstveni rad: LeCun, Bottou, Bengio, Haffner. *Gradient-based learning applied to document recognition*, 1998.
- Predložena mreža naziva **LeNet-5** i smatra se prvom uspješnom primjenom konvolucijskih neuronskih mreža u praksi
- Inspiracija je otkriće lokalno osjetljivih i orijentacijski osjetljivih neurona kod vizualnog sustava mačke (Hubel and Wiesel) i Neocognitron (Fukushima)
- Konvolucijska neuronska mreža koja je naučena pomoću algoritma unazadne propagacije (engl. *backpropagation*)
- Primjena na prepoznavanje rukom pisanih brojeva u poštanskim pošiljkama
- U idućih nekoliko godina konvolucijske neuronske mreže su se koristile u raznim primjenama, ali ne intenzivno
- Uglavnom se koristio „klasični računalni vid”
- Problem s ograničenim računalnim resursima i s količinom označenih podataka

LeNet-5 (1998.)



Značajniji rezultat CNN (2012.)

- Prva uspješna primjena na RGB slike veće rezolucije
- Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks*, 2012.



IMAGENET



„We trained the network for roughly 90 cycles through the training set of 1.2 million images, which took five to six days on two NVIDIA GTX 580 3GB GPUs”

Konvolucijska neuronska mreža - detaljno

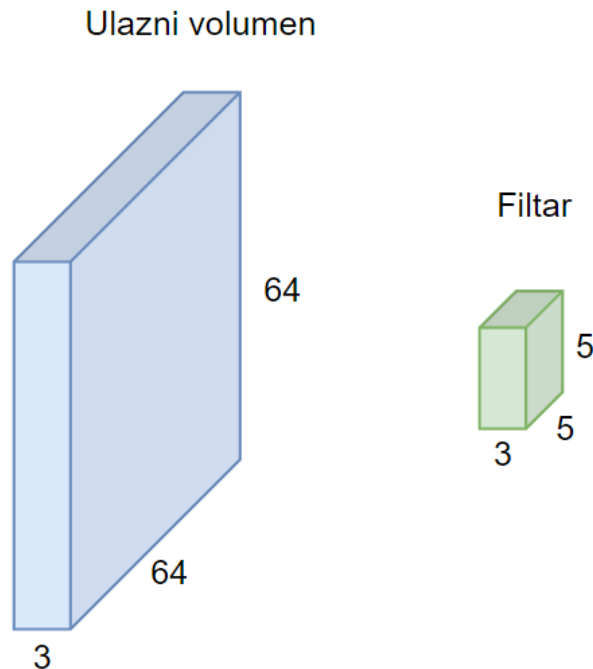
- CNN je poseban tip neuronskih mreža koji se koristi za obradu slika i drugih vrsta podataka koji imaju prostornu strukturu.
- Tipična struktura CNN-a sastoji se od nekoliko glavnih slojeva:
 1. **Ulazni sloj** - prihvaća ulazni podatak, obično sliku
 2. **Konvolucijski slojevi** - koriste se za izdvajanje značajki iz slike u obliku aktivacijskih mapa, svaki neuron je povezan s malim područjem ulaznog volumena
 3. **Aktivacijski slojevi** – primjena aktivacijske funkcije (npr. ReLU) na svaki pojedinačni element aktivacijske mape
 4. **Slojevi sažimanja** - koriste se za smanjenje prostorne dimenzionalnosti mape značajki
 5. **Potpuno povezani slojevi** – svaki neuron sloja je povezan sa svakim neuronom prethodnog sloja (npr. s ReLU aktivacijskim funkcijama)
 6. **Izlazni sloj** – obično se koristi potpuno povezani sloj sa softmax aktivacijskom funkcijom

Konvolucijski sloj

- **Konvolucijski sloj** je glavni gradbeni element u CNN
- Djeluje na ulazni volumen (prisjetite se, hoćemo zadržati prostornu strukturu podataka tijekom izvlačenja značajki)
- Svaki konvolucijski sloj sastoji se od većeg broja **filtera** čiji su parametri podesivi (mogu se naučiti)
- Svaki filter je prostorno malen (npr. tipično 3x3 ili 5x5), ali „djeluje” po dubini ulaznog volumena
- Npr. ako je ulazni volumen RGB slika s tri kanala, tada je filter dimenzija 3x3x3 ili 5x5x3
- Pogledajmo primjer

Konvolucijski sloj

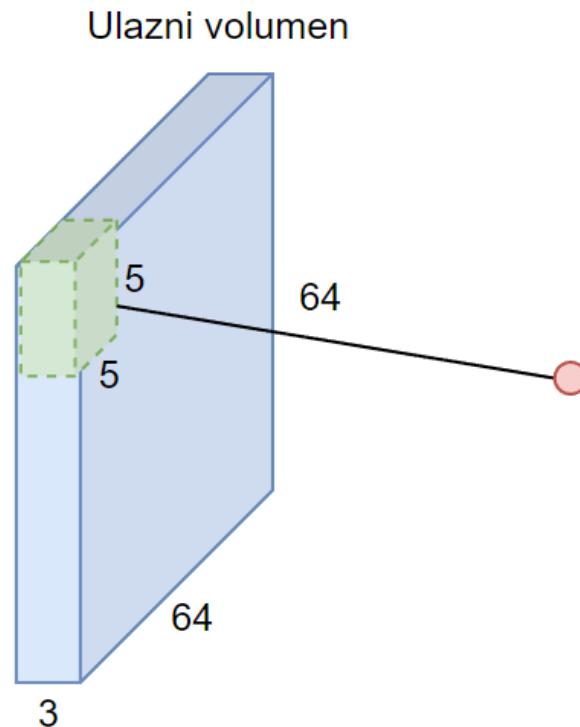
- Tijekom propagacije signala unaprijed kroz mrežu svaki neuron povezan je samo s malim područjem (receptivno polje) u ulaznom volumenu
- Primjer: ulaz je RGB slika 64x64 piksela, filter je dimenzija 5x5x3



Taj (isti) filter će se pomicati po ulaznom volumenu

Konvolucijski sloj

- Primjena filtra na određenom dijelu ulaznog volumena rezultira u skalarnoj vrijednosti (skalarni produkt dijela volumena i filtra)
- Filtar pokriva malo prostorno područje, ali se prostire preko sva tri kanala ulaznog volumena
- Težine (\mathbf{w}) filtra i pomak (b) predstavljaju parametre filtra i mogu se podešavati (to su parametri mreže; mijenjaju se tijekom treniranja mreže)



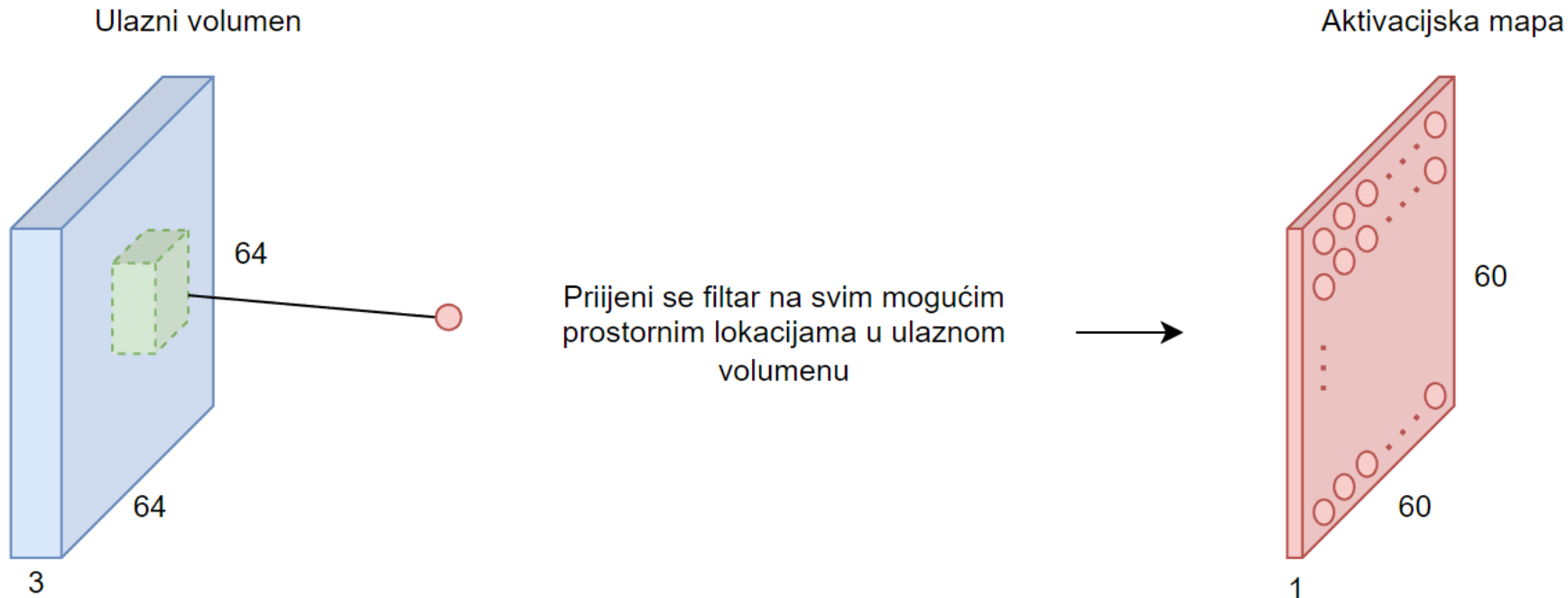
Za jednu poziciju filtra u ulaznoj slici dobiva se jedan broj - skalarni produkt tog dijela slike i filtra

$$\mathbf{w}^T \mathbf{x} + b$$

Broj parametara filtra: $5 \times 5 \times 3 + 1 = 76$

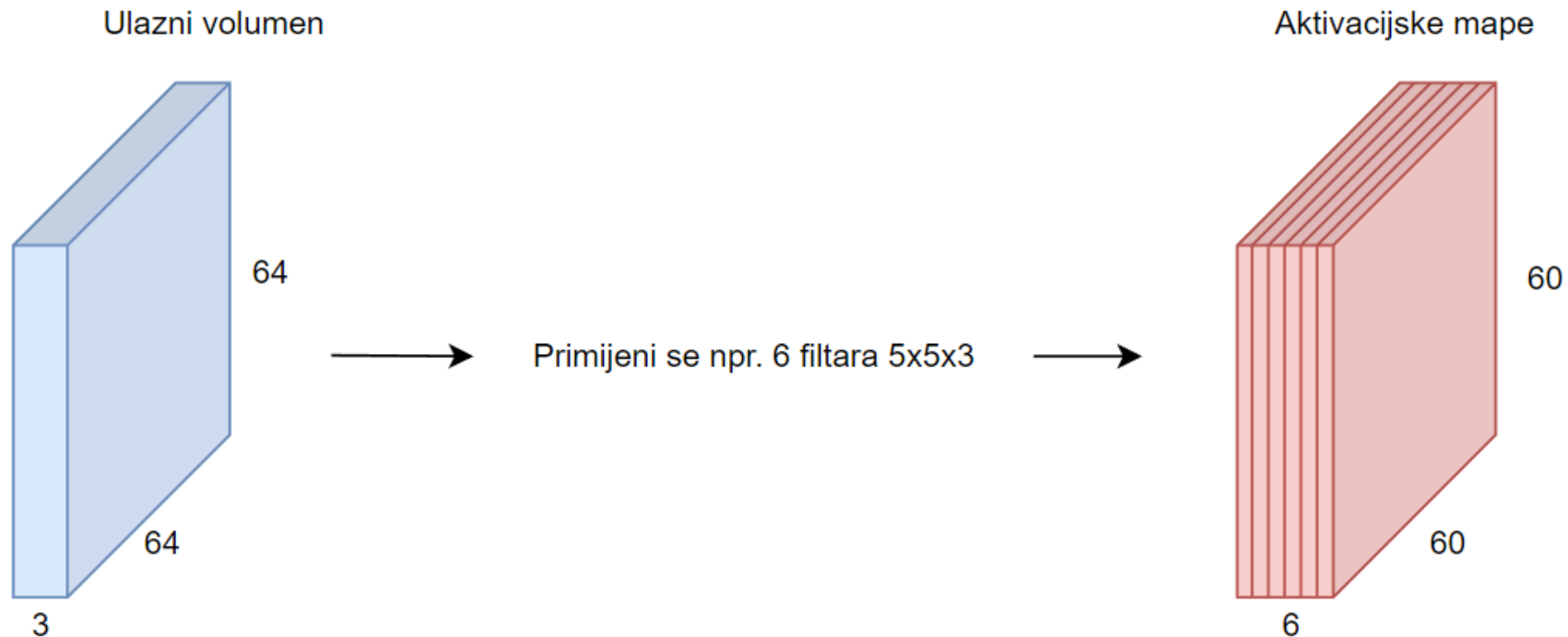
Konvolucijski sloj

- Primjenom istog filtra na različitim pozicijama ulazne slike uz primjenu aktivacijske funkcije rezultira u **aktivacijskoj mapi** → dvodimenzionalno polje koje sadrži odziv filtra na pojedinom dijelu ulazne slike
- Svi neuroni iste aktivacijske mape imaju zajedničke parametre (znatno manje parametara nego kod potpuno povezanog sloja)



Konvolucijski sloj

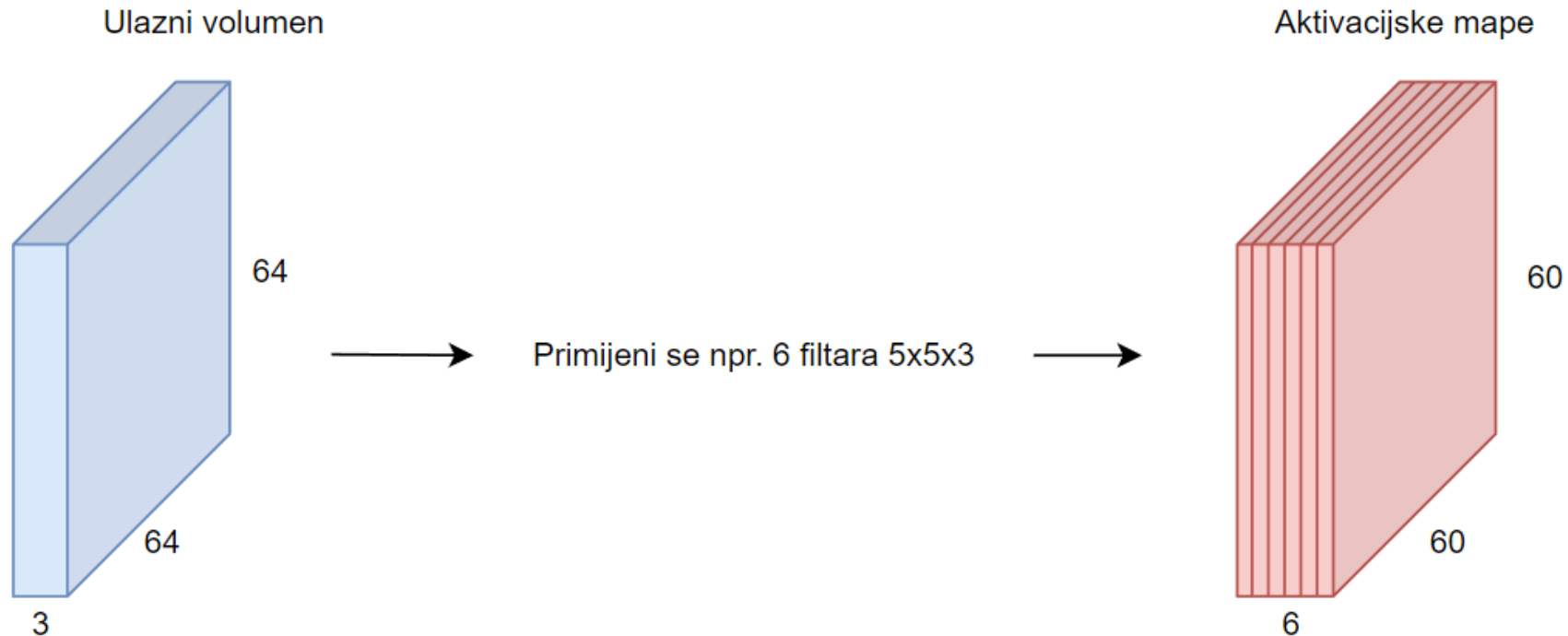
- Moguće je primijeniti više filtara (jednakih dimenzija) na ulazni volumen → dobivaju se zasebne aktivacijske mape koje se „slažu” jedna pored druge (dobivamo drugačiju reprezentaciju ulazne slike, npr. 60x60x6)



Ovo se podrazumijeva pod konvolucijskim slojem.

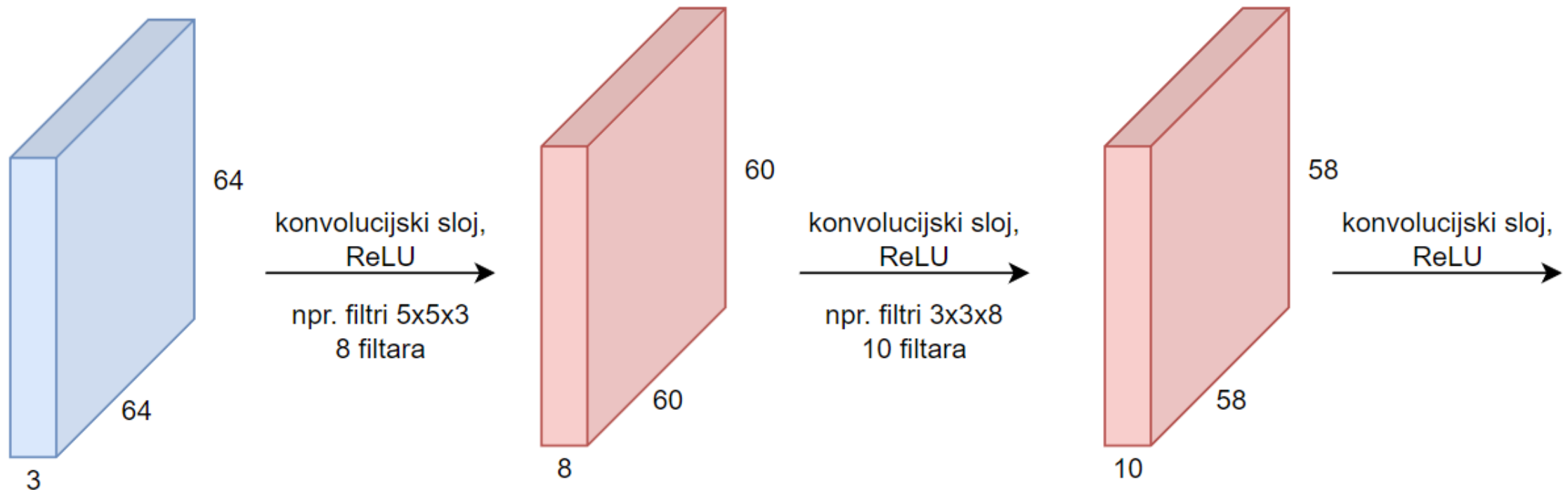
Konvolucijski sloj

- Koliko parametara ima ovakav konvolucijski sloj? Ukupno ima parametara $(5*5*3*6 + 6) = 456$
- Ekvivalentni potpuno povezani sloj: $64*64*3*60*60*6 = 265.4M$



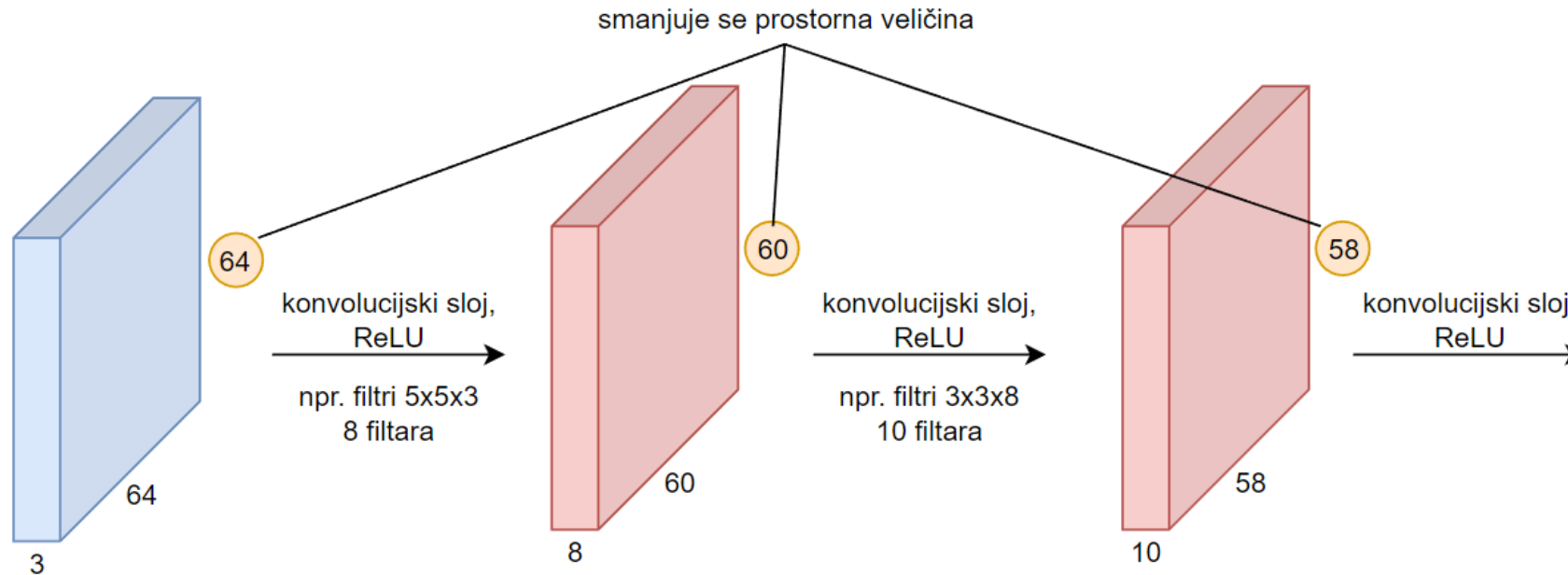
Konvolucijski sloj

- CNN se temelji na sekvenci takvih konvolucijskih slojeva između kojih se nalaze aktivacijski slojevi (najčešće ReLU)



Konvolucijski sloj – *zero padding*

- Vidjeli smo da se sekvencijalnom primjenom konvolucijskih filtara smanjuje prostorna veličina volumena
- U većini programskih okruženja (npr. *Keras*) moguće je izvršiti dodavanje ruba u pojedinom konvolucijskom sloju tako da prostorna dimenzija izlaznog volumena bude jednaka prostornoj dimenziji ulaznog volumena



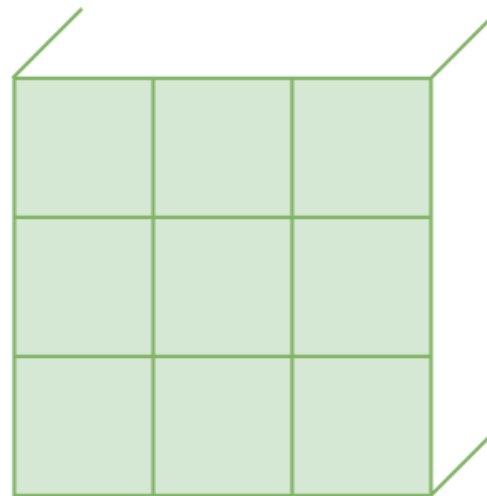
Konvolucijski sloj – *zero padding*

- Primjer

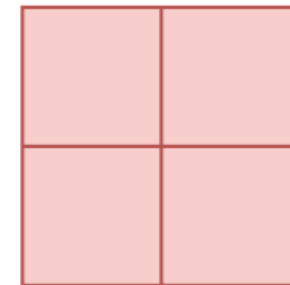


0.32	0.27	0.65	1.0
0.71	1.0	0.90	0.90
0.78	0.98	0.87	0.88
0.46	0.50	0.56	0.79

Ulazni volumen 4x4



Filtar 3x3



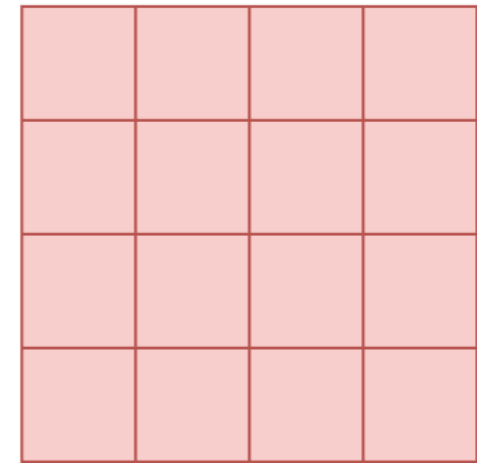
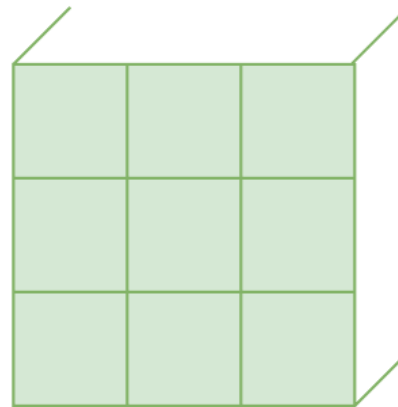
Aktivacijska mapa 2x2

Konvolucijski sloj – *zero padding*

- Primjer

0	0	0	0	0	0
0	0.32	0.27	0.65	1.0	0
0	0.71	1.0	0.90	0.90	0
0	0.78	0.98	0.87	0.88	0
0	0.46	0.50	0.56	0.79	0
0	0	0	0	0	0

Ulazni volumen 4x4 s zero paddingom

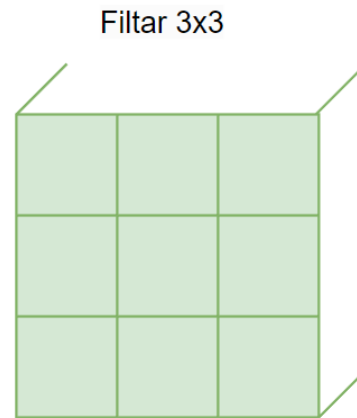
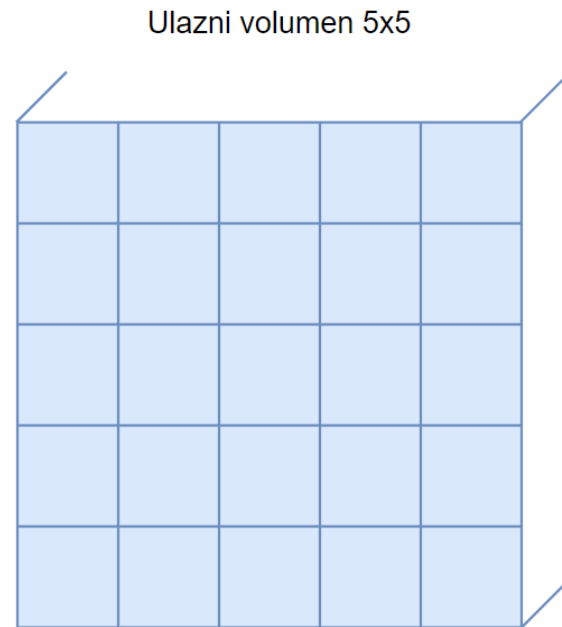


Filtar 3x3

Aktivacijska mapa 4x4

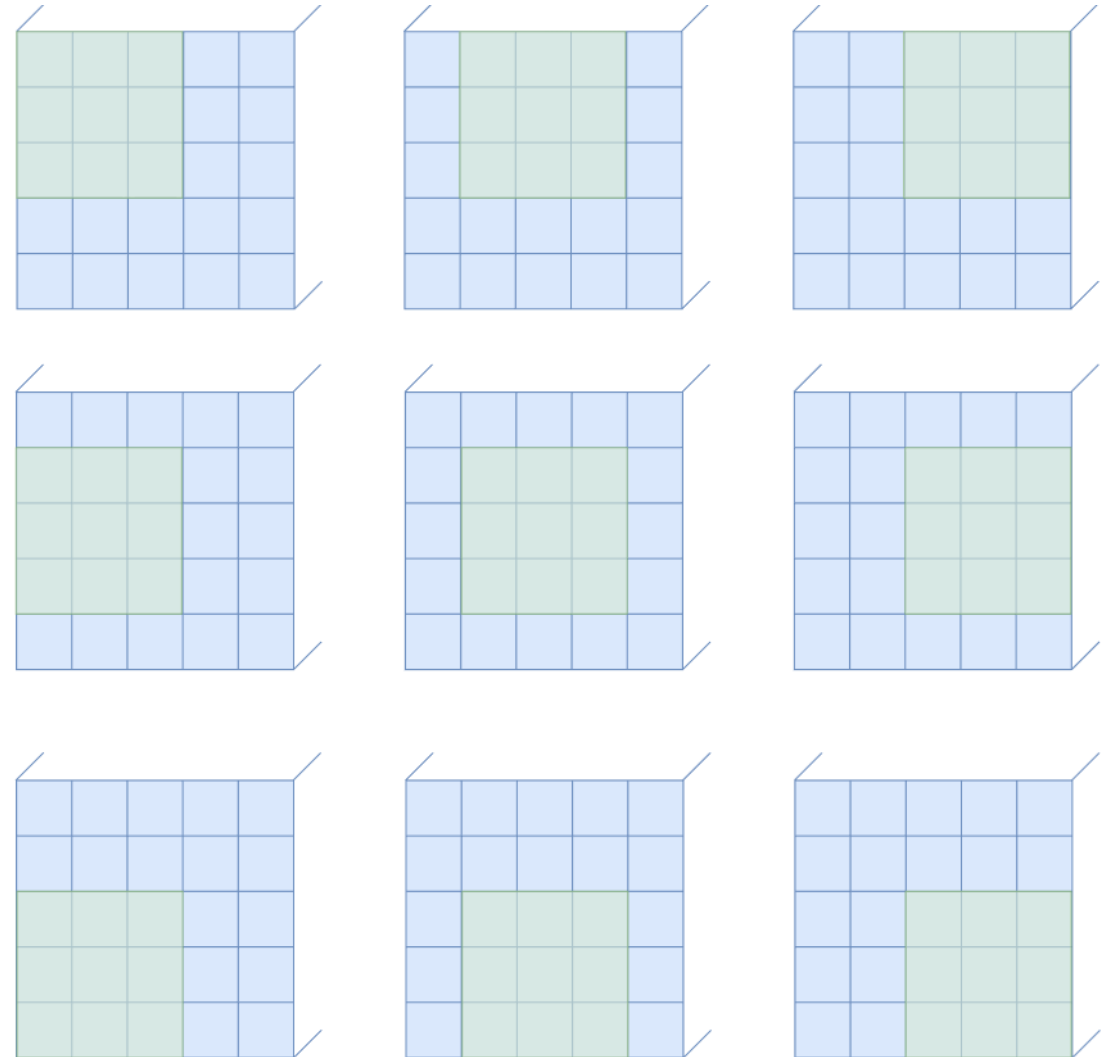
Konvolucijski sloj - *stride*

- Kod definiranja konvolucijskog sloja moguće je odabrati *stride* – cjelobrojni broj koji definira za koliko se elemenata pomiče filter po prostornoj dimenziji ulaznog volumena
- Ako je *stride* veći od 1 dolazi do smanjenja prostorne dimenzije
- Primjer



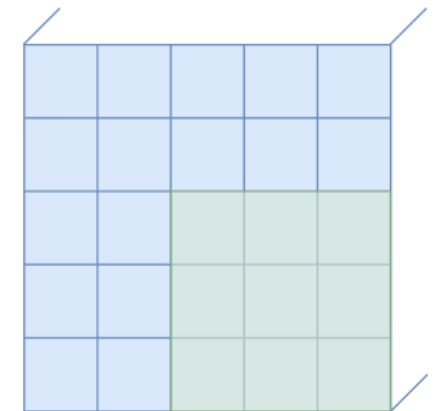
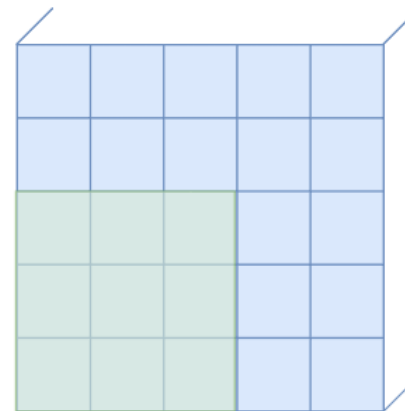
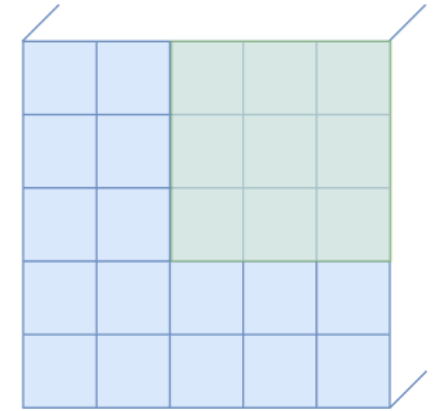
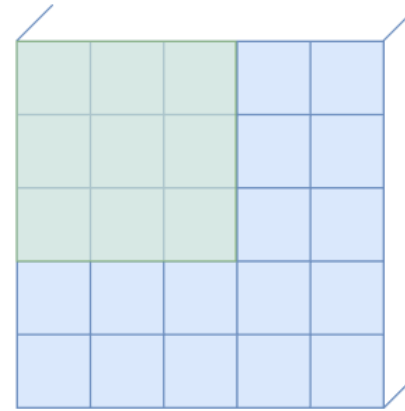
Konvolucijski sloj - *stride*

- Stride 1
- Aktivacijska mapa je dimenzija 3x3



Konvolucijski sloj - *stride*

- Stride 2
- Aktivacijska mapa je dimenzija 2x2

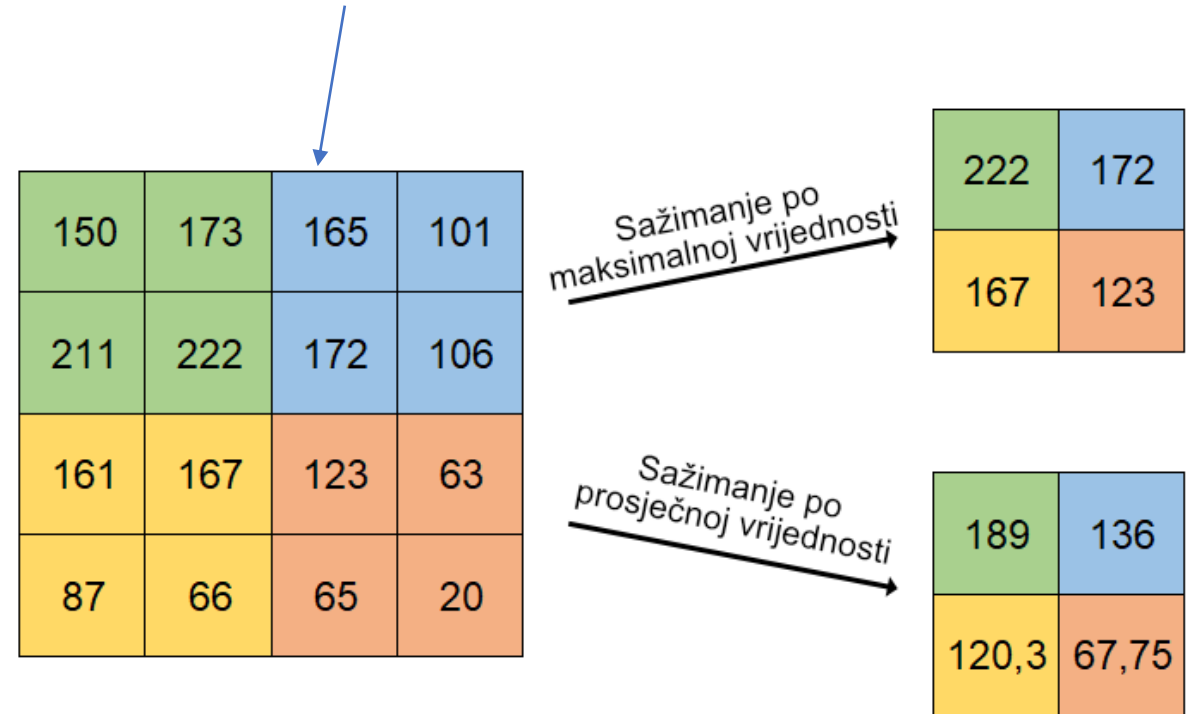
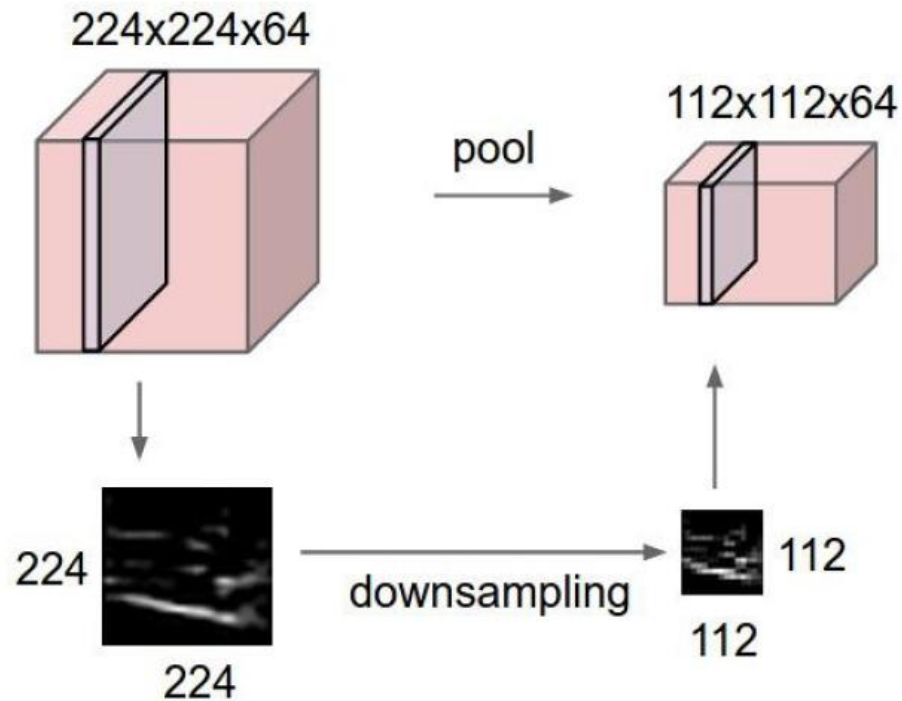


Sloj sažimanja

- Uobičajeno je u niz konvolucijskih slojeva ubaciti **sloj sažimanja** na pojedina mjesta kako bi se postepeno smanjivala prostorna dimenzija volumena koji se obrađuje
- Ovaj sloj radi smanjivanje svake pojedine mape značajki u ulaznom volumenu
- Na taj način se smanjuje potrebna količina računskih operacija i memorijski zahtjevi (sjetite se kako radi konvolucijski sloj), a na taj način smanjuje se i broj parametara u mreži (pa time možemo spriječiti i pretjerano usklađivanje (*overfitting*))
- Korištenjem sloja sažimanja se izdvajaju i robusnije značajke tj. sloj pomaže u izdvajanju značajki koje su relativno neosjetljive na male translacije u ulaznim podacima
- Također može pomoći u kako bi se postigla određena invarijantnost s obzirom na rotaciju

Sloj sažimanja

Potrebno je definirati veličinu prozora (npr. 2x2) i *stride* (npr. 2)



Najčešći način sažimanja je po maksimalnoj vrijednosti – **max pooling**

Primijetite kako ovaj sloj nema parametre koji se treniraju!

Konvolucijska neuronska mreža - prednosti

- **Lokalna prostorna informacija:** kod slika je lokalna prostorna informacija važna što znači da svaki piksel u slici ima neku vrstu veze s pikselima oko njega. Ovaj oblik informacije je bolje obraditi pomoću CNN nego FCN mreže.
- **Parametri:** CNN-ovi imaju manje parametara u odnosu na FCN, što omogućuje efikasnije učenje i manje izloženosti pretjeranom usklađivanju (engl. *overfitting*). FCN-ovi obično imaju veliki broj parametara zbog potpune povezanosti, dok se u CNN-ovima koriste dijeljene težine u konvolucijskim slojevima.
- **Rotacija i translacija:** CNN-ovi su bolji u prepoznavanju rotiranih i pomaknutih uzoraka. Kada se slika rotira ili pomiče, CNN-ovi će i dalje izdvojiti značajke i prepoznati objekte, dok FCN može biti osjetljiva na ovakve promjene.
- **Višeslojnost:** CNN-ovi imaju niz konvolucijskih slojeva koji se zajedno ponašaju kao ekstraktor značajki; svaki sloj ima specifičnu ulogu u obradi slike. Ovaj pristup omogućuje CNN-ovima da nauče složenije značajke slike iz prethodnih slojeva, što često dovodi do bolje kvalitete klasifikacije.

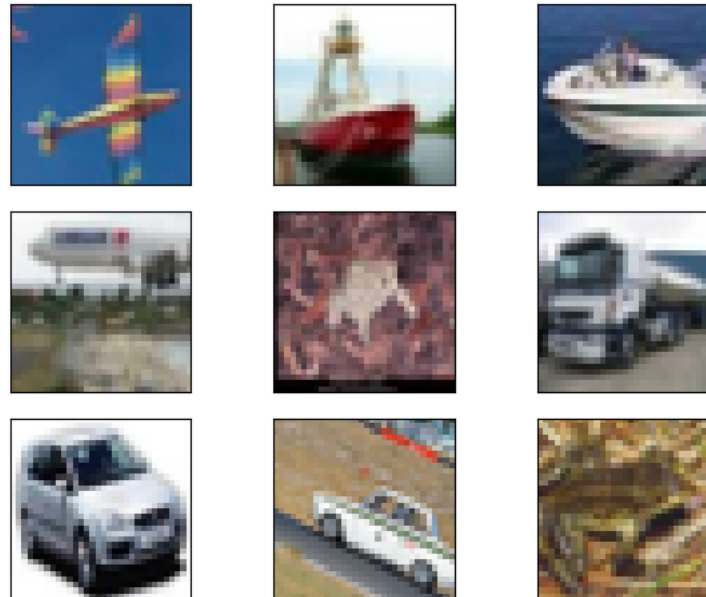
Konvolucijska neuronska mreža - detaljno

- Vizualizacija CNN za klasifikaciju slike – aktivacijske mape za određenu ulaznu sliku



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- **CIFAR-10** podatkovni skup – izazovniji nego MNIST skup
- RGB slike dimenzija 32x32 piksela, 10 klasa
- 50,000 slika za učenje, 10,000 slika za testiranje (6,000 po klasi)
- Primjeri:



Klase

- 0: airplane
- 1: automobile
- 2: bird
- 3: cat
- 4: deer
- 5: dog
- 6: frog
- 7: horse
- 8: ship
- 9: truck

Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Predložena mreža sastoji se od:
 - Ulazni sloj (slika dimenzija 32x32x3)
 - Konvolucijski sloj (32 filtra, 3x3, stride 1, padding 'same', aktivacija ReLU)
 - Max pooling sloj (prozor 2x2, stride 2)
 - Konvolucijski sloj (64 filtra, 3x3, stride 1, padding 'same', aktivacija ReLU)
 - Max pooling sloj (prozor 2x2, stride 2)
 - Konvolucijski sloj (128 filtara, 3x3, stride 1, padding 'same', aktivacija ReLU)
 - Max pooling sloj (prozor 2x2, stride 2)
 - Potpuno povezani sloj (250 neurona, ReLU aktivacijska funkcija)
 - Potpuno povezani sloj (10 neurona, softmax aktivacijska funkcija)
- Skicirajte ovu mrežu, naznačite dimenziju izlaznog volumena svakog sloja te broj parametara u svakom sloju

Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Primjer ispisa dimenzija i broja parametara pojedinog sloja u okruženju *Keras*

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
maxpooling_1 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_1 (Conv2D)	(None, 16, 16, 64)	18496
maxpooling_2 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	73856
maxpooling_3 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 250)	512250
dense_1 (Dense)	(None, 10)	2510

```
=====  
Total params: 608,008  
Trainable params: 608,008  
Non-trainable params: 0  
=====
```

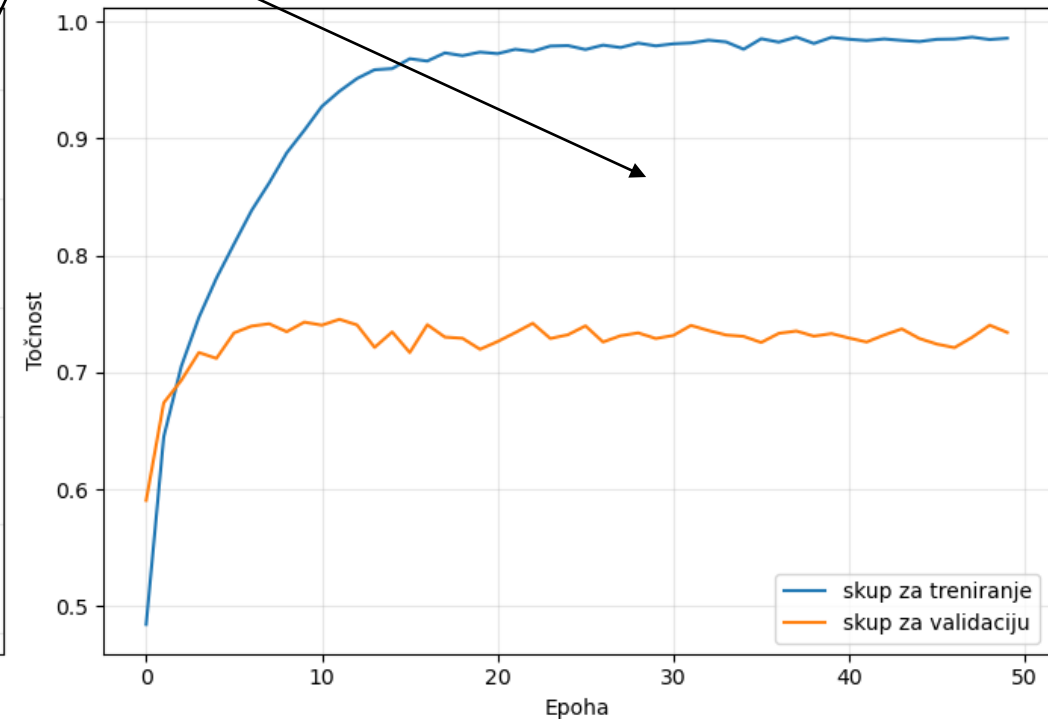
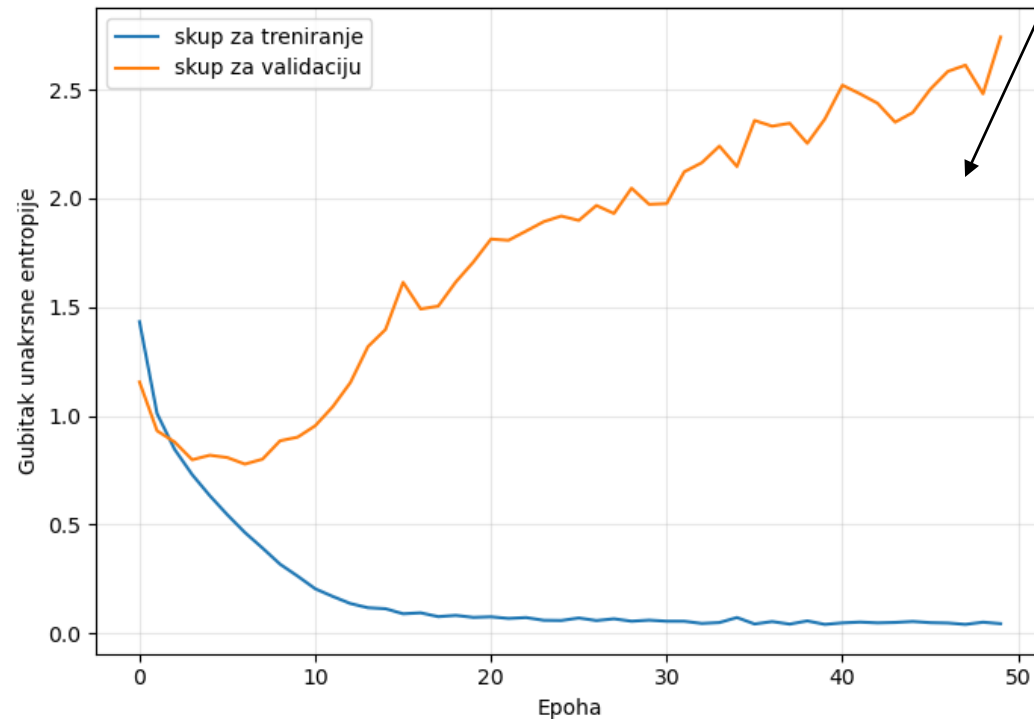
Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Postavke prilikom treninga:
 - Broj epoha 50
 - Veličina *batcha* 64
 - Stopa učenja (engl. *learning rate*) jednaka je 0.001
 - Adam optimizacijski algoritam
 - Oznake su kodirane 1-od- K kodiranjem (pri čemu je $K=10$)
 - Validacijski skup je 10% skupa za učenje
 - 45,000 slika za treniranje, 5,000 slika za validaciju
 - Koliko će onda jedna epoha imati iteracija?
- Pratit ćemo gubitak unakrsne entropije i točnost klasifikacije na skupu za treniranje i skupu za validaciju

Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

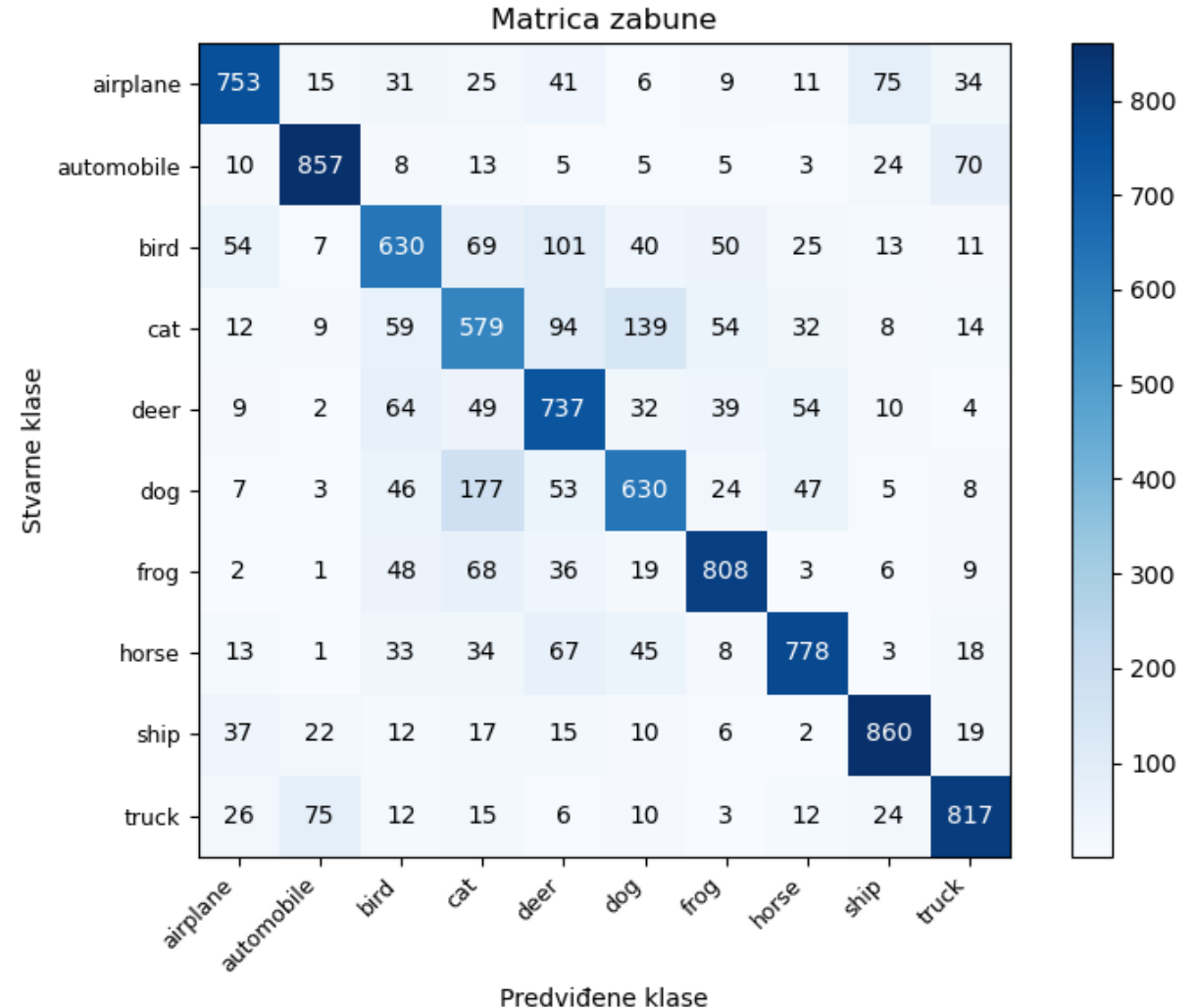
- Tijek treniranja mreže

Očito se događa pretjerano usklađivanje na podatke za treniranje (engl. *overfitting*)



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Kao rezultanti model uzimamo model s parametrima iz 12. epohe koji ima najveću točnost na validacijskom skupu
- Točnost na testnom skupu ovog modela je: **74.49%**



Popularne arhitekture
konvolucijskih neuronskih mreža

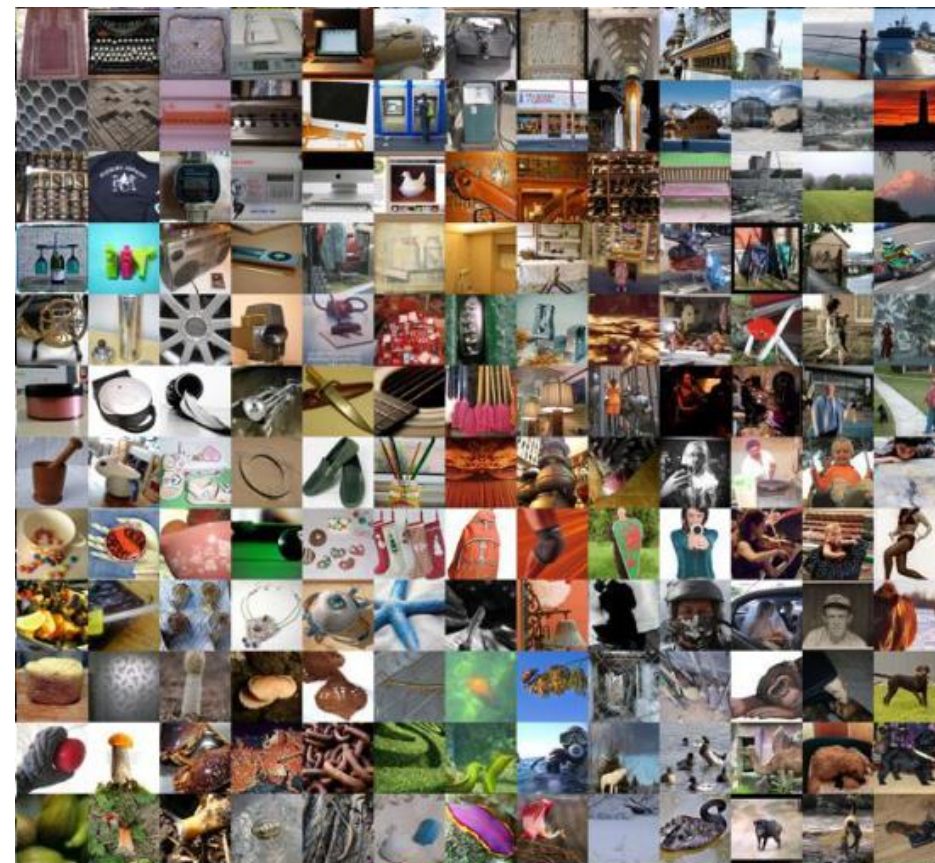
Popularne arhitekture CNN

- Tijekom godina razvijene su različite arhitekture konvolucijskih neuronskih mreža, neki poznati primjeri:
 1. **LeNet-5 (1998.)** - jedna od prvih arhitektura konvolucijskih neuronskih mreža, koja je korištena za prepoznavanje rukom pisanih znamenki.
 2. **AlexNet (2012.)** – revolucionarna arhitektura koja je osvojila ImageNet natjecanje 2012. godine i postavila temelje za razvoj dubokih konvolucijskih neuronskih mreža.
 3. **VGG-16 i VGG-19 (2014.)** - arhitekture koje su se također natjecale na ImageNetu i koje su poznate po svojoj dubini i velikom broju slojeva
 4. **GoogLeNet / Inception (2014.)** - arhitektura koja je uvela "Inception" blokove
 5. **ResNet (2015.)** - arhitektura koja koristi rezidualne veze u mreži kako bi se izbjegao problem nestajućih gradijenata i omogućila treniranje vrlo dubokih mreža
 6. **MobileNet (2017.)** - arhitektura koja je optimizirana za mobilne uređaje i ugradbene računalne sustave, ima manje parametara i manje je računalno zahtjevana
 7. **EfficientNet (2019.)** - balansira širinu, dubinu i rezoluciju modela za optimalnu točnost i učinkovitost; lako se skalira ovisno o resursima i zadatku

ImageNet



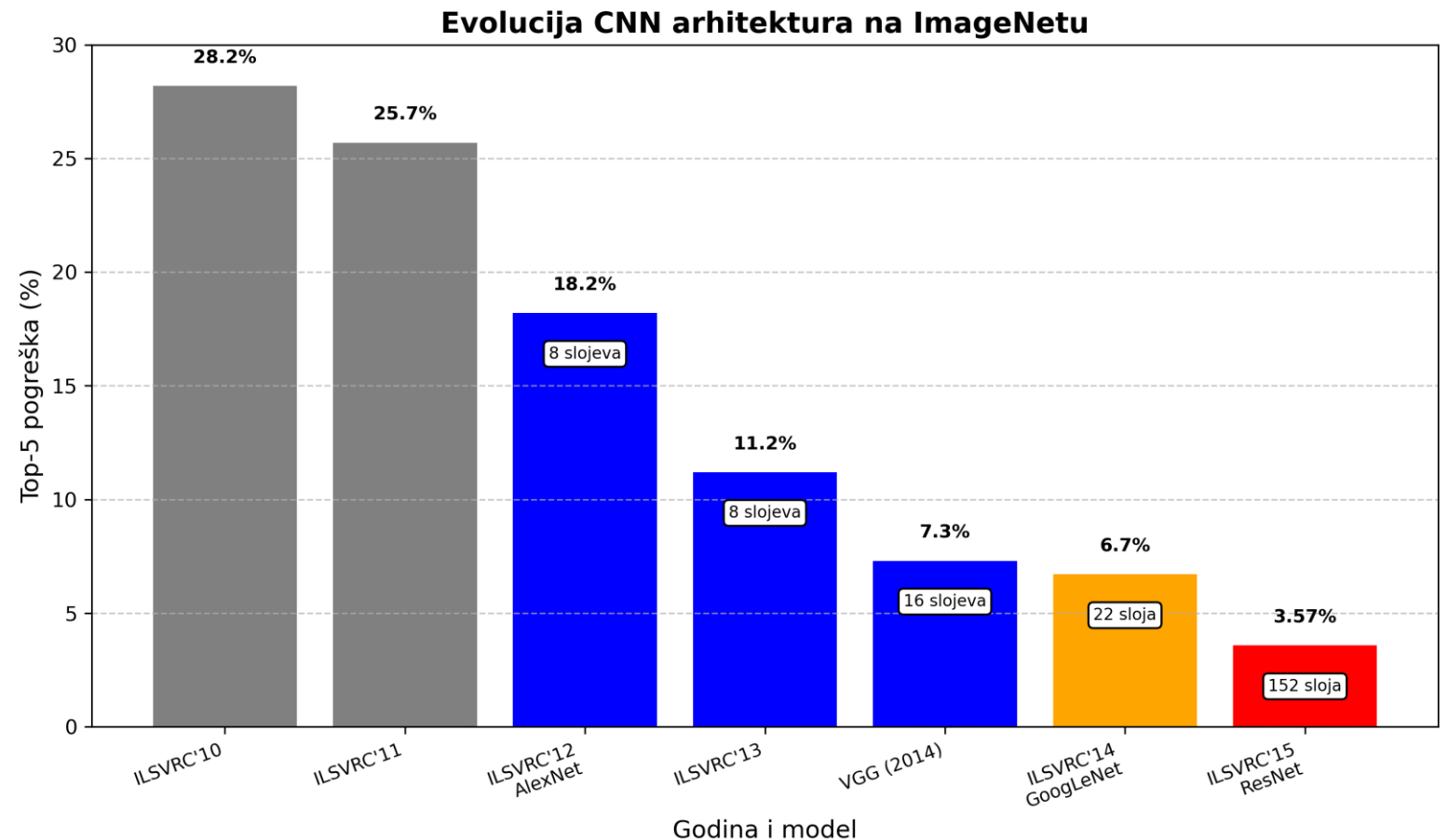
- Veliki skup podataka predstavljen 2009. godine, još uvijek je standard prilikom *benchmarkinga* pojedinih algoritama u području klasifikacije slika i prepoznavanju objekata (identifikacija i lokalizacija objekta na slici)
- Nastala je uslijed potrebe za velikim i raznolikim skupom označenih podataka u području računalnog vida i razvoja dubokog učenja
- Sadrži više od 14 milijuna označenih slika u preko 22,000 kategorija
 - Npr. postoji 120 različitih pasmina pasa
- Godine 2010. godine pokrenuto je ImageNet Large Scale Visual Recognition Challenge (ILSVRC) natjecanje koje je potaknulo razvoj konvolucijskih neuronskih mreža



ImageNet pobjednici

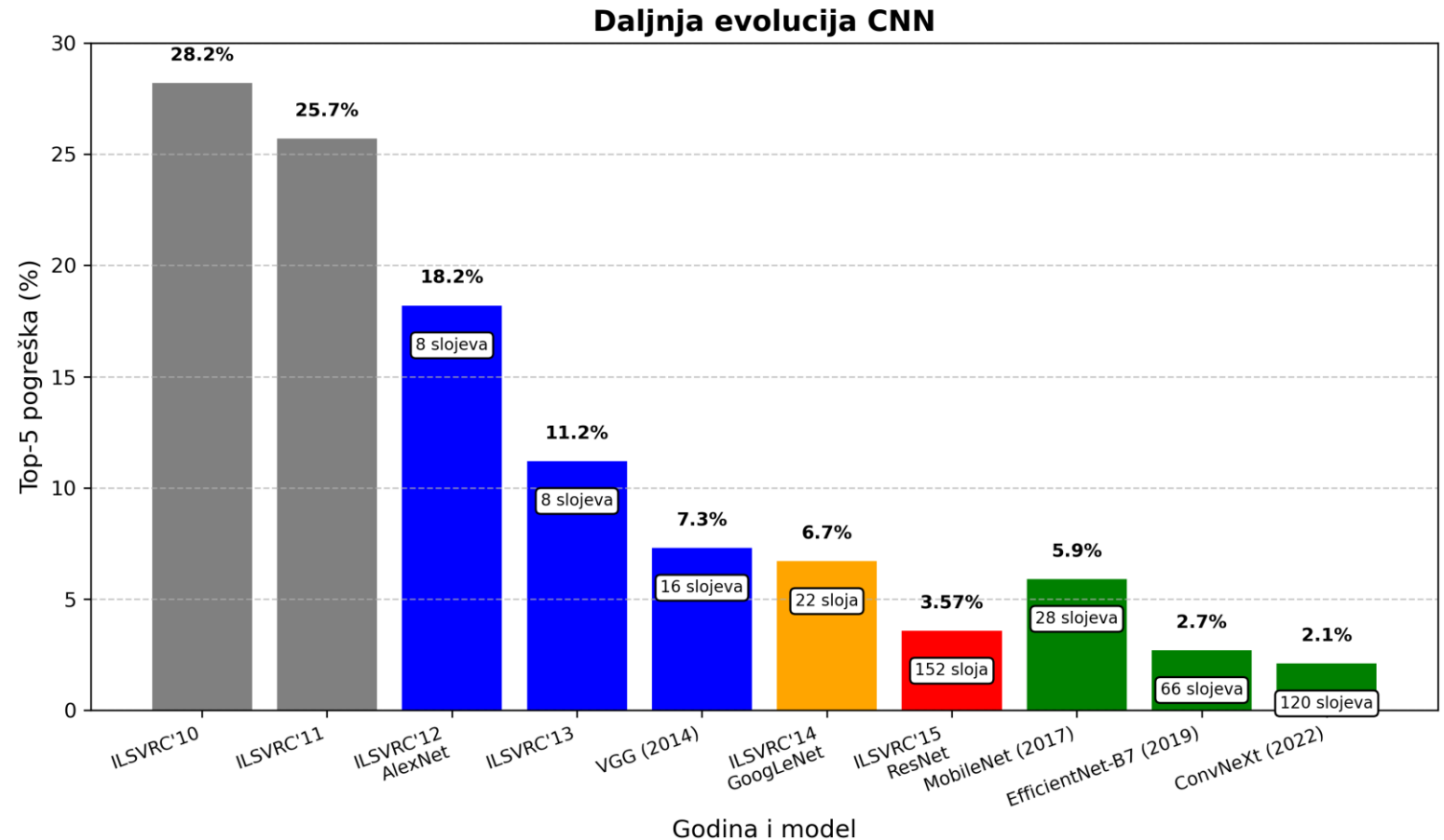
Tendencija produbljivanja mreže tijekom godina

- 1.2M skup za treniranje
- 50k skup za validaciju
- 100k skup za testiranje
- 1000 kategorija
- Procjena ljudske pogreške na ovom skupu je oko **5%** (Andrej Karpathy)
- Pogledajmo detaljnije neke od ovih mreža



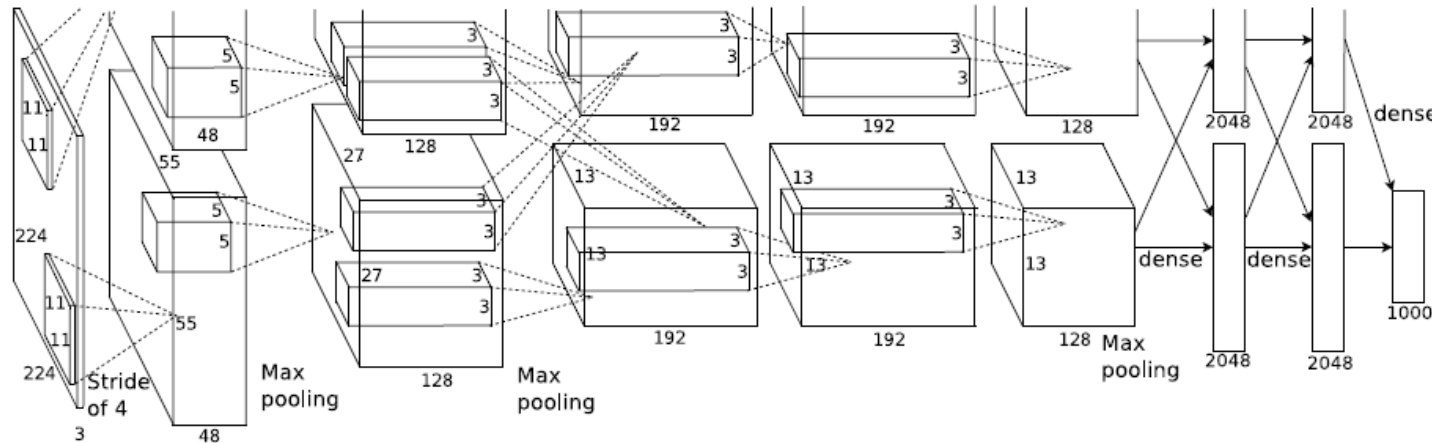
Daljnji razvoj

- Nakon 2015., razvoj CNN arhitektura usmjerio se s povećanja dubine na veću efikasnost i prilagodljivost – s ciljem smanjenja memorijskih zahtjeva i ubrzanja izvođenja (npr. MobileNet, EfficientNet).
- Kasnije su razvijeni i modernizirani CNN modeli poput ConvNeXt, koji kombiniraju jednostavnost konvolucija s principima iz transformera.



AlexNet (2012.)

- [Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks*, 2012.](#)



Full (simplified) AlexNet architecture:

[227x227x3] INPUT

[55x55x96] **CONV1**: 96 11x11 filters at stride 4, pad 0

[27x27x96] **MAX POOL1**: 3x3 filters at stride 2

[27x27x96] **NORM1**: Normalization layer

[27x27x256] **CONV2**: 256 5x5 filters at stride 1, pad 2

[13x13x256] **MAX POOL2**: 3x3 filters at stride 2

[13x13x256] **NORM2**: Normalization layer

[13x13x384] **CONV3**: 384 3x3 filters at stride 1, pad 1

[13x13x384] **CONV4**: 384 3x3 filters at stride 1, pad 1

[13x13x256] **CONV5**: 256 3x3 filters at stride 1, pad 1

[6x6x256] **MAX POOL3**: 3x3 filters at stride 2

[4096] **FC6**: 4096 neurons

[4096] **FC7**: 4096 neurons

[1000] **FC8**: 1000 neurons (class scores)

Details/Retrospectives:

- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

VGG (2014.)

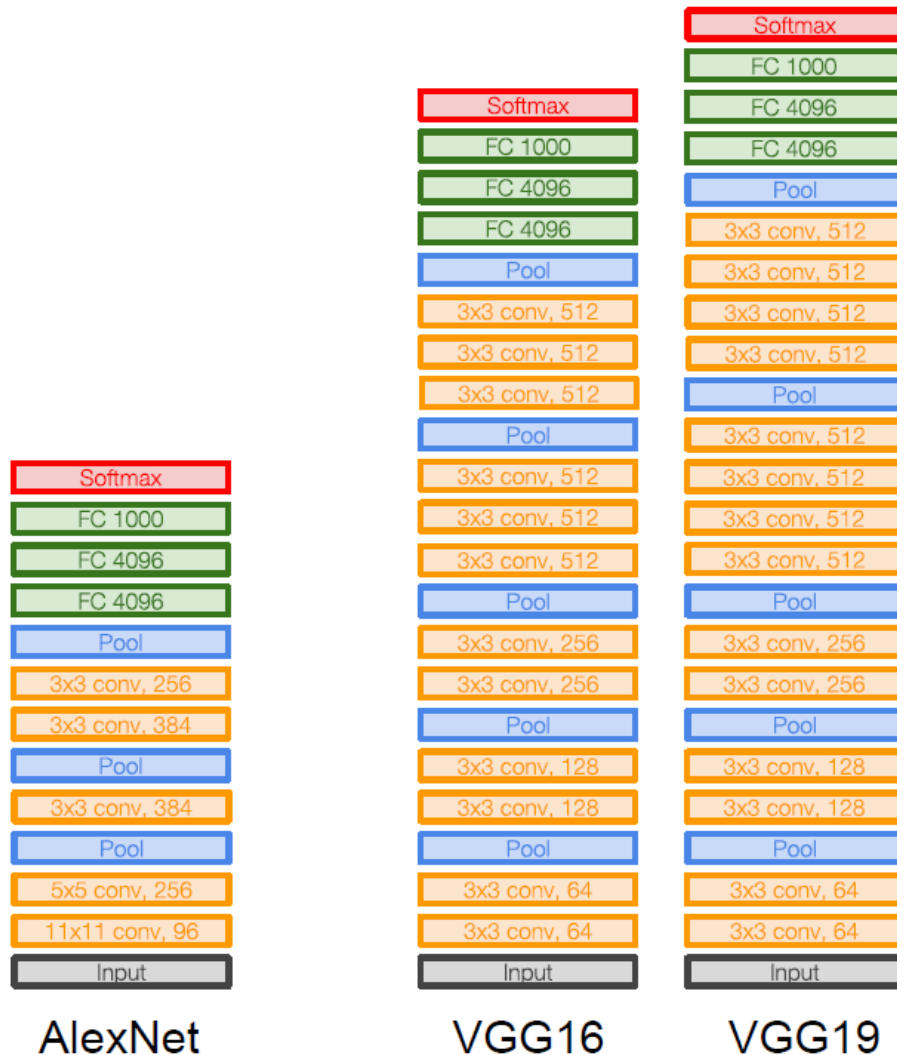
- [Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014.](#)
- Manji filtri (3x3)
- Više slojeva (dublja mreža)
- ReLU
- Sličan proces učenja kao kod AlexNet
- 7.3% top 5 error na ILSVRC 2015 (ansambl 7 modela)
- Najbolji model VGG19

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

VGG i AlexNet usporedba

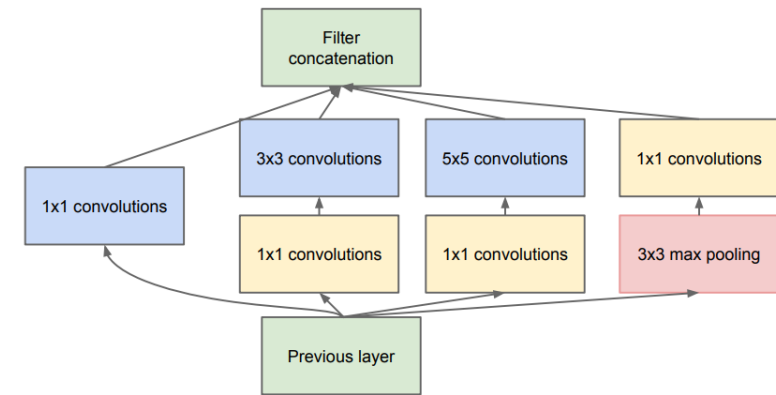


Motivacija za korištenje 3x3 filtara u VGG mreži

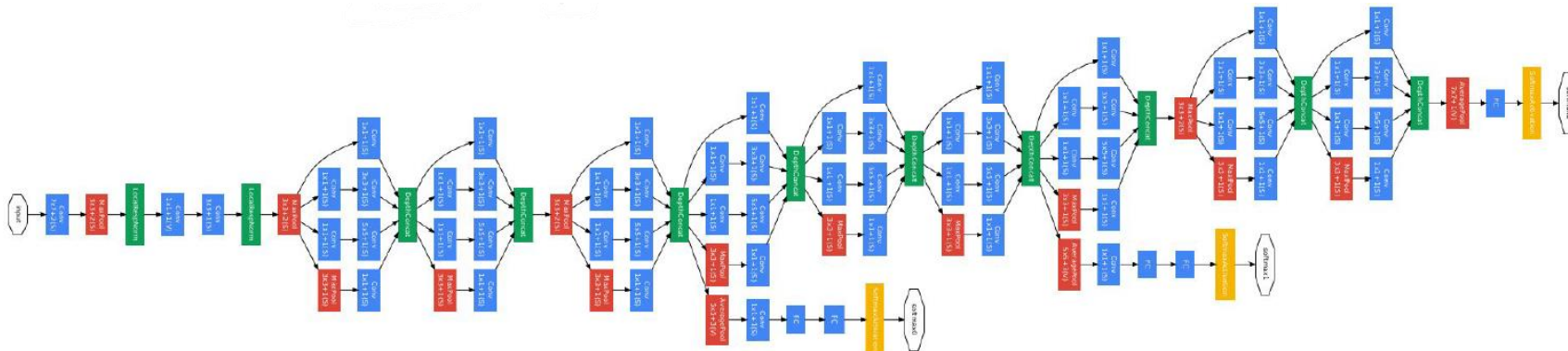
„It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an effective receptive field of 5×5; three such layers have a 7 × 7 effective receptive field. So what have we gained by using, for instance, a stack of three 3×3 conv. layers instead of a single 7×7 layer? First, we incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Second, we decrease the number of parameters: assuming that both the input and the output of a three-layer 3 × 3 convolution stack has C channels...”

GoogleLeNet (Inception v1) (2014.)

- [Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going Deeper with Convolutions, 2014.](#)
- Nastavlja se trend produbljivanja mreže s naglaskom na računalnu efikasnost
- GoogLeNet:
 - Efikasni Inception moduli
 - Nema potpuno povezanih slojeva
 - Samo 5 milijuna parametara (12x manje nego AlexNet)
 - 6.7% top 5 error ILSVRC 2014



(b) Inception module with dimension reductions



ResNet (2015.)

- Problem kada se uče izrazito duboke mreže na klasičan način
- Očekujemo da se poveća točnost s povećanjem dubine mreže na skupu za treniranje

[Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015.](#)

- Klasične jako duboke mreže imaju problem degradacije (gradijenti postaju jako mali pa se raniji slojevi ne optimiziraju učinkovito)
- Optimizacija mreže je otežana

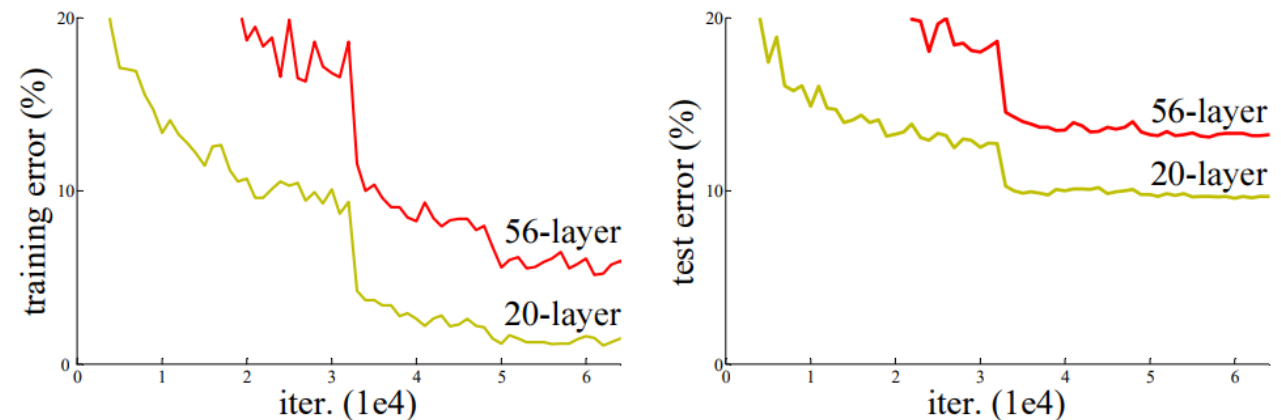
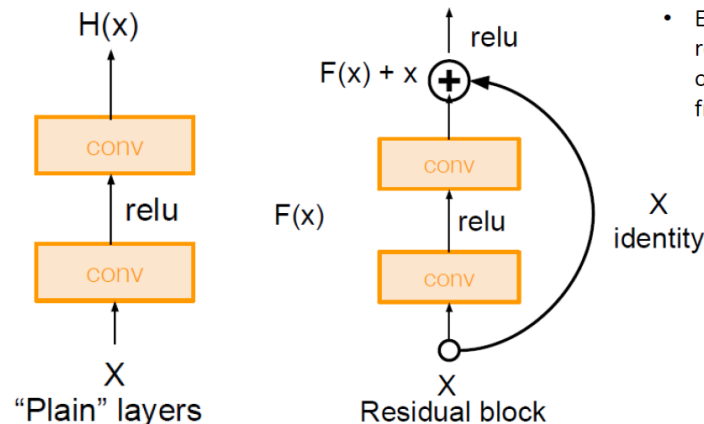


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

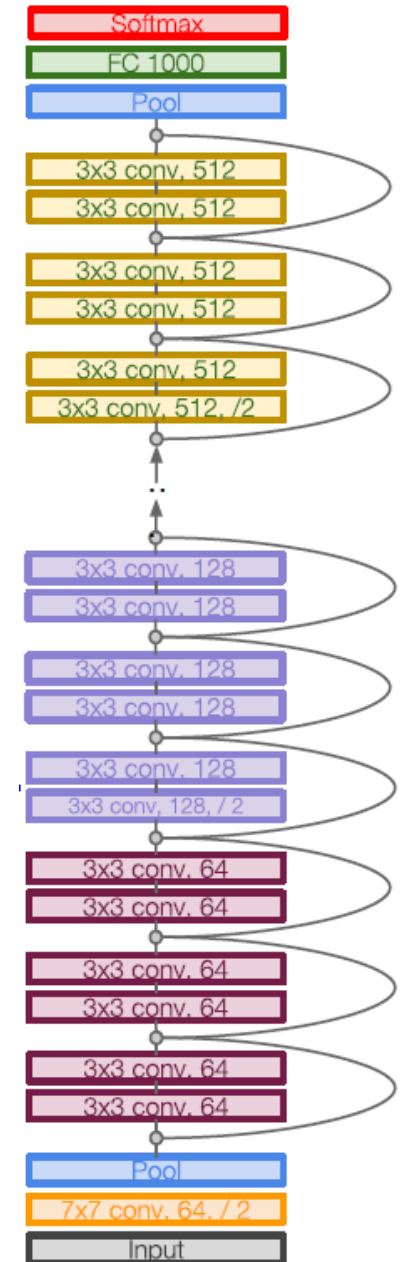
ResNet (2015.)

- Revolucija u dubini mreže (uspješno istrenirali mreže koje imaju i 150 slojeva)
- Prvo mjesto u 5 različitih natjecanja (ImageNet i COCO)
- Koristi rezidualne blokove koji omogućuju učinkovit prijenos gradijenata; svaki ima dva 3x3 konvolucijska sloja



- Empirical evidence showing that residual networks are easier to optimize, and can gain accuracy from increased depth.

Vrlo popularna mreža je ResNet 50



Resnet (2015.)

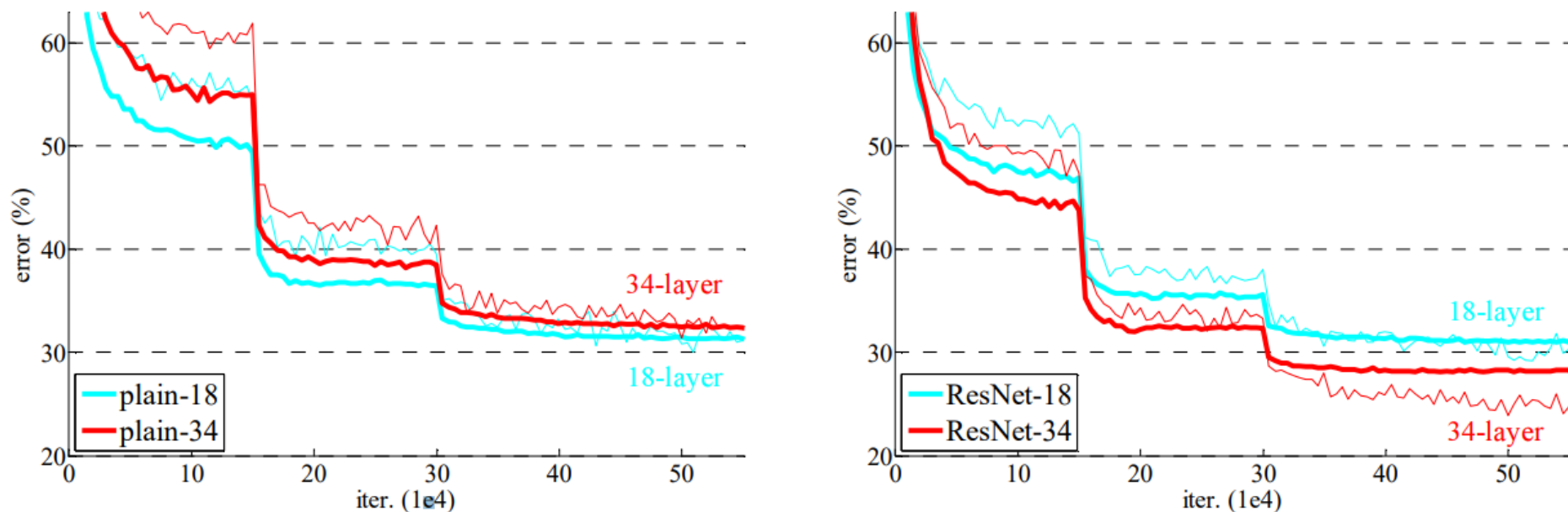


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	25.03

Table 2. Top-1 error (% , 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

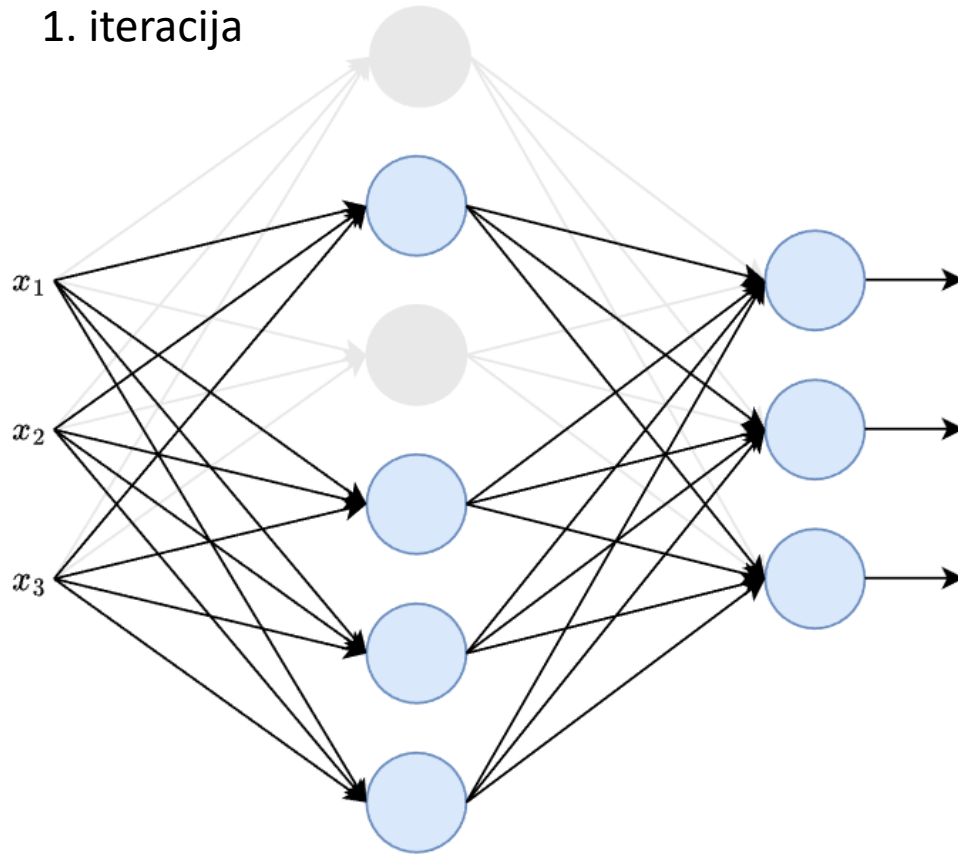
Poboljšanje procesa učenja
(konvolucijskih) neuronskih mreža

Nasumično izbacivanje neurona tijekom treniranja

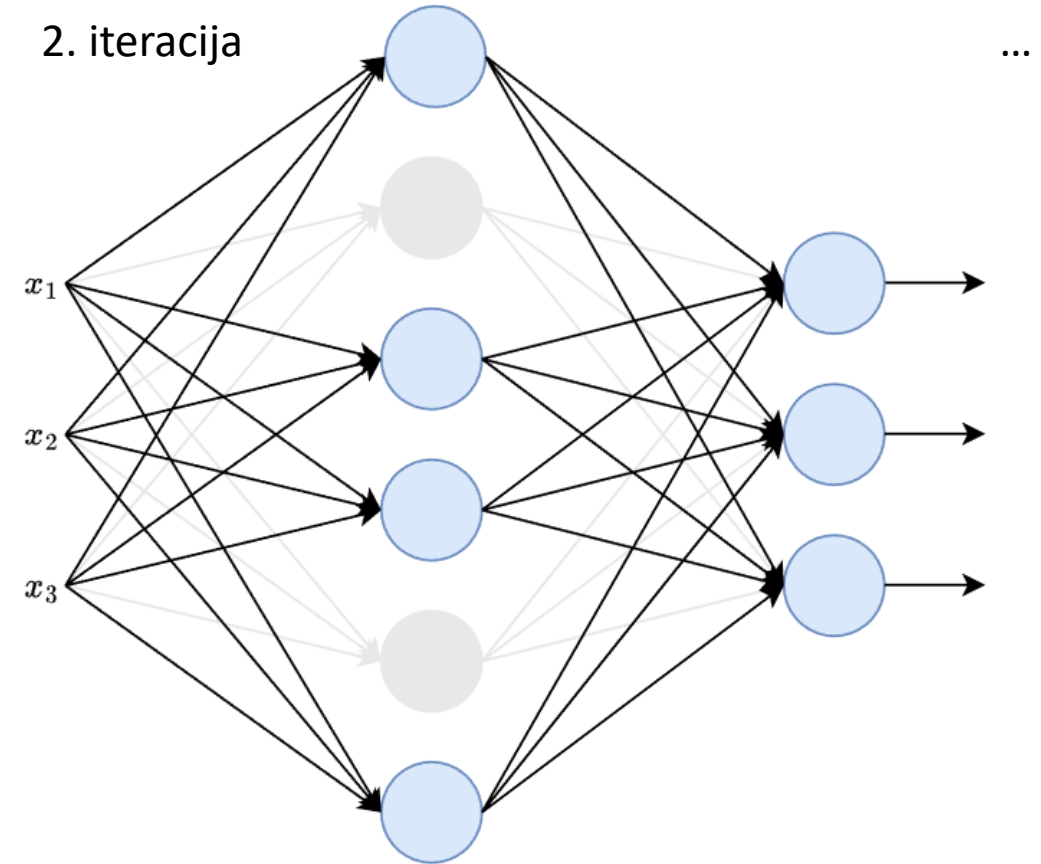
- **Nasumično izbacivanje neurona** (engl. *dropout*) je česta tehnika koja se primjenjuje u neuronskim mrežama kako bi se spriječilo pretjerano usklađivanje na podatke (Hinton, 2012.)
- Ideja ove tehnike je nasumično izbacivanje (isključivanje) neurona u određenom sloju neuronske mreže s nekom vjerojatnošću p
- Ova vjerojatnost se definira unaprijed, a tipične vrijednosti su 0.2, 0.3 i 0.5
- Kada se neki neuron izbací, to znači da se on neće koristiti za propagiranje signala kroz mrežu tijekom te iteracije niti se parametri povezani s njim osvježavaju BP algoritmom
- To znači da se mreža u svakoj iteraciji mora prilagoditi novim uvjetima odnosno koristiti preostale neurona za rješavanje problema
- Ovo može pomoću u sprječavanju *overfittinga* jer se neuronska mreža ne može oslanjati na specifične skupove neurona prilikom rješavanja problema
- Prilikom inferencije se koristi naučena mreža koja se sastoji od svih neurona

Nasumično izbacivanje neurona tijekom učenja

1. iteracija



2. iteracija



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

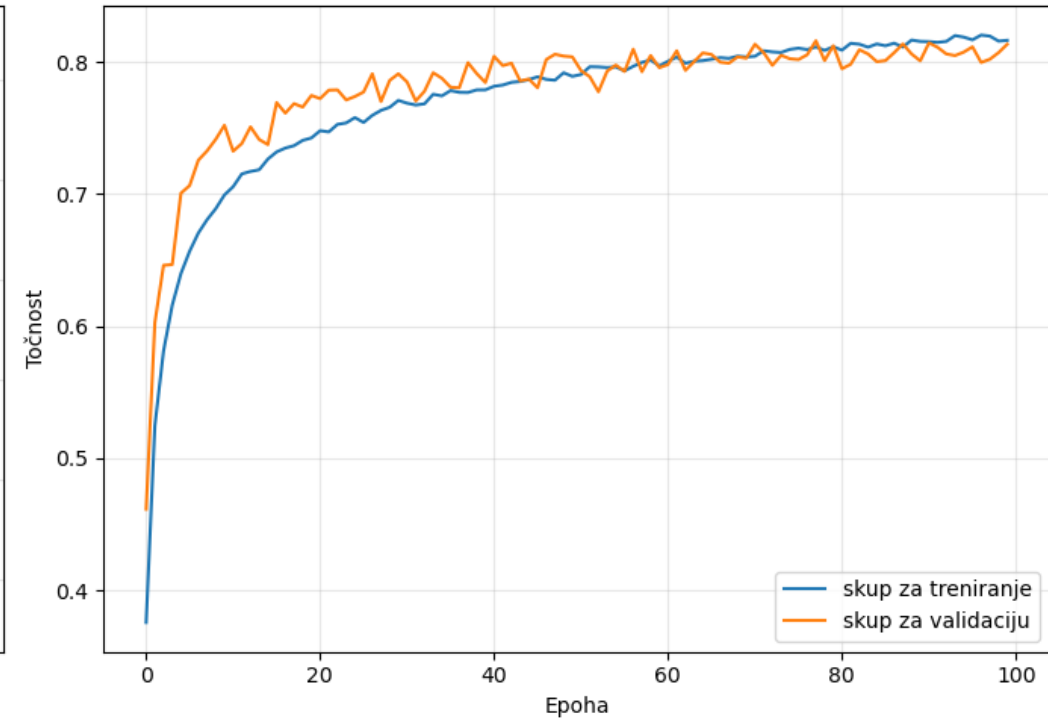
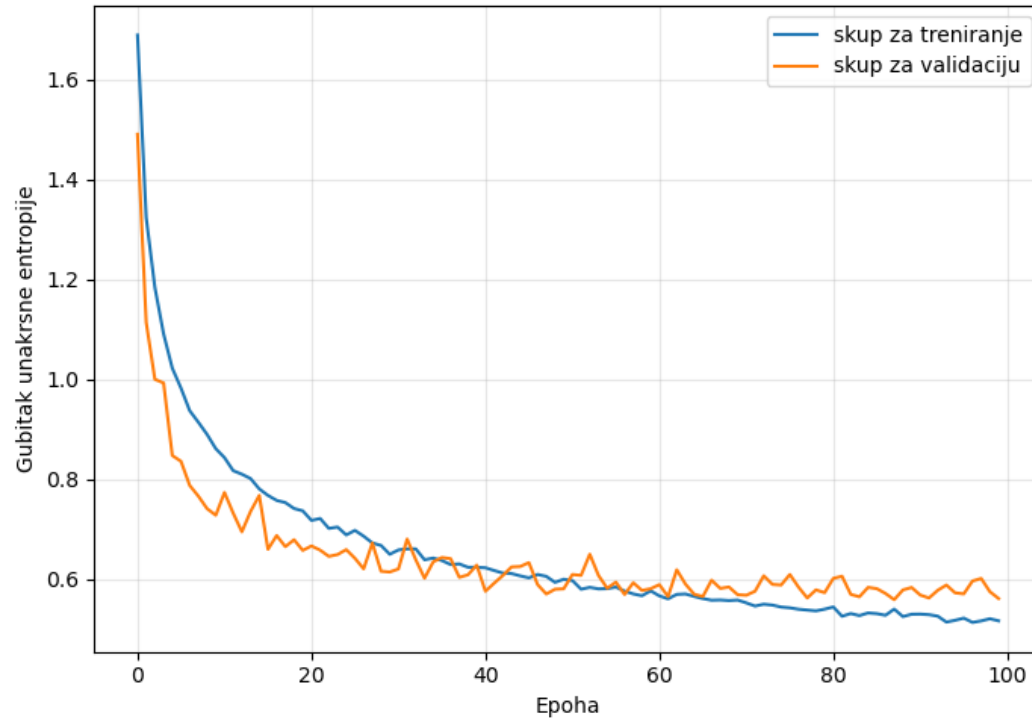
- Modificirati ćemo mrežu na način da ćemo ubaciti *dropout* sloj na određena mjesta kako bismo spriječili pretjerano usklađivanje:

- Ulazni sloj (slika dimenzija 32x32x3)
- Konvolucijski sloj (32 filtra, 3x3, stride 1, padding 'same', aktivacija ReLU)
- Max pooling sloj (prozor 2x2, stride 2)
- Dropout (vjerojatnost izbacivanja 0.3)
- Konvolucijski sloj (64 filtra, 3x3, stride 1, padding 'same', aktivacija ReLU)
- Max pooling sloj (prozor 2x2, stride 2)
- Dropout (vjerojatnost izbacivanja 0.3)
- Konvolucijski sloj (128 filtara, 3x3, stride 1, padding 'same', aktivacija ReLU)
- Max pooling sloj (prozor 2x2, stride 2)
- Dropout (vjerojatnost izbacivanja 0.3)
- Potpuno povezani sloj (250 neurona, ReLU aktivacijska funkcija)
- Dropout (vjerojatnost izbacivanja 0.5)
- Potpuno povezani sloj (10 neurona, softmax aktivacijska funkcija)

Dropout sloj nema parametre koji bi se podešavali tijekom treninga (tj. mreža ima jednak broj parametara sa i bez dropout sloja)

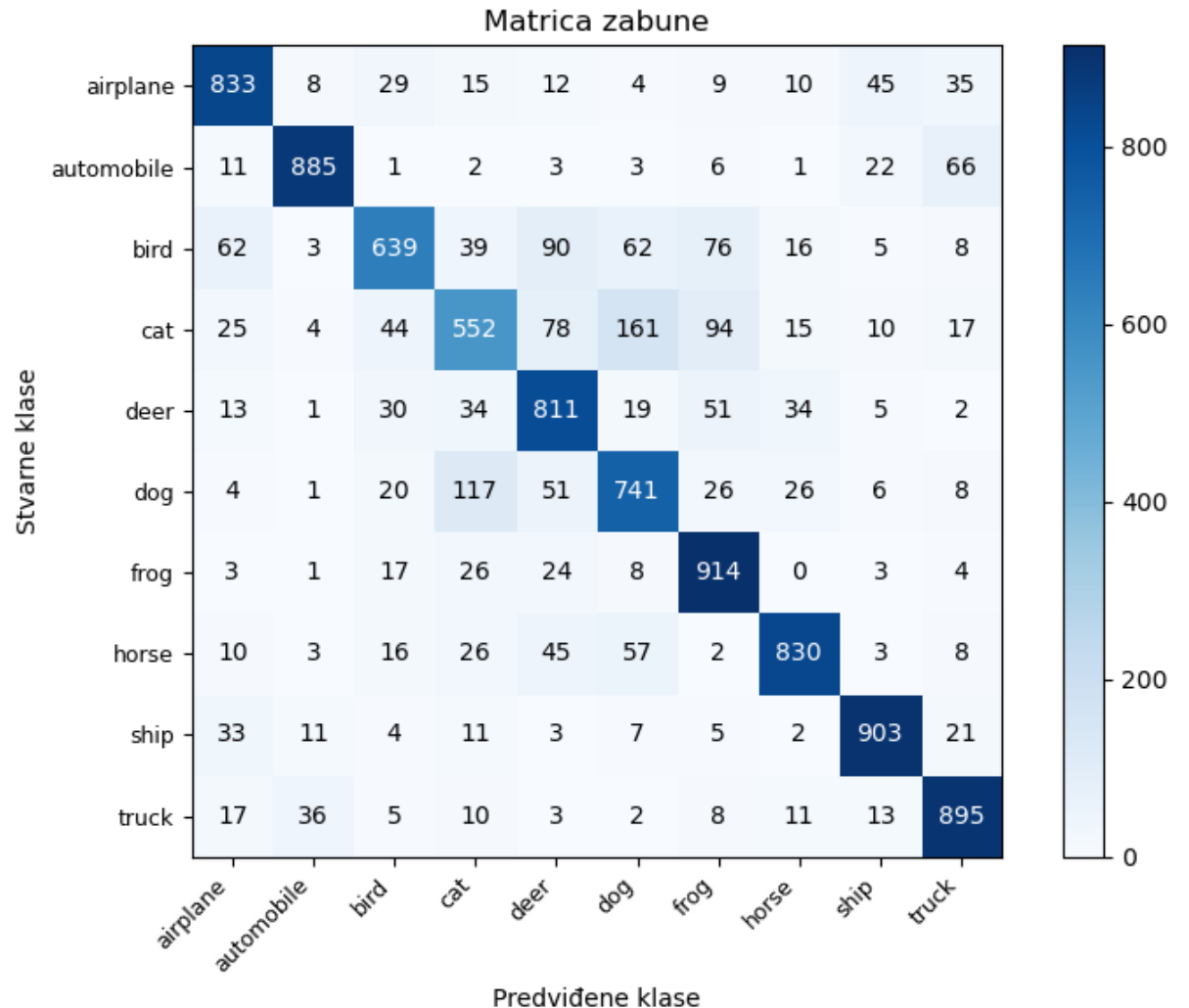
Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Tijek treniranja mreže (sada više nije prisutan *overfitting*)



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Kao rezultat uzimamo model s parametrima iz 78. epohe
- Točnost na testnom skupu ovog modela je: **80.03%**
- Relativno jednostavnom modifikacijom mreže (u odnosu na osnovni pristup) postigli smo oko **5-6%** veću točnost na testnom skupu
- Ovo pokazuje važnost korištenja dodatnih tehnika kako bi se pospješio proces treniranja mreže



Augmentacija skupa podataka za učenje

- **Augmentacija** skupa podataka za učenje je postupak umjetnog povećavanja skupa podataka za učenje na način da se novi primjeri kreiraju od postojećih primjera
- U području računalnog vida augmentacija se svodi na modifikaciju postojećih slika u podatkovnom skupu pri čemu se koriste različite transformacije poput rotacije, skaliranja, izrezivanja dijelova slike, zrcaljenje, promjena osvjetljenja i sl.
 - Pri tome je za svaku „novu” sliku poznata oznaka
 - Relativno računalno jeftin postupak
- Glavni cilj augmentacije podataka jest sprječavanje *overfittinga* budući da je za efikasno učenje CNN-a obično potrebna velika količina označenih primjera
- Na primjer, u području računalnog vida rotiranje i skaliranje slika omogućuje CNN-u da nauči različite perspektive i veličine objekata, što može biti korisno za prepoznavanje objekata u stvarnom svijetu
- Augmentaciju je moguće primijeniti i kod obrade teksta i zvuka
 - Zamjena riječi sa sinonimom u tekstu, dodavanje varijacija u sintaksi
 - Ubacivanje šuma u audio zapis, promjena brzine, modulacija tona

Augmentacija skupa podataka za učenje

- **Primjer**
- Koristiti realistične transformacije!
- Originalna slika:



Rotacija



Zoom



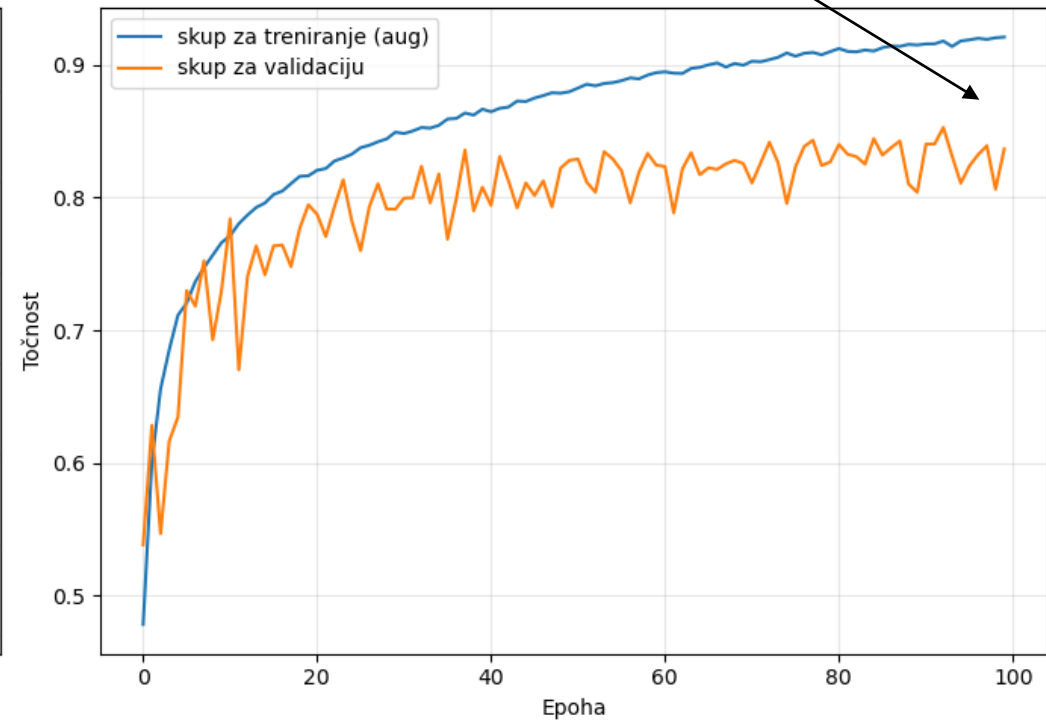
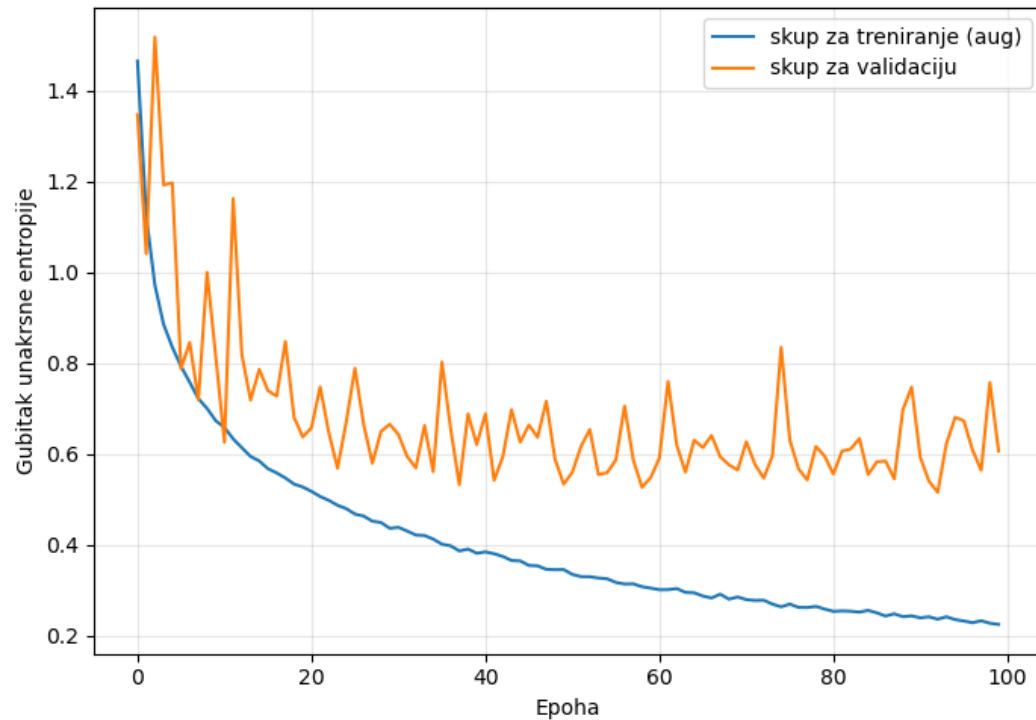
Osvjetljenje



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

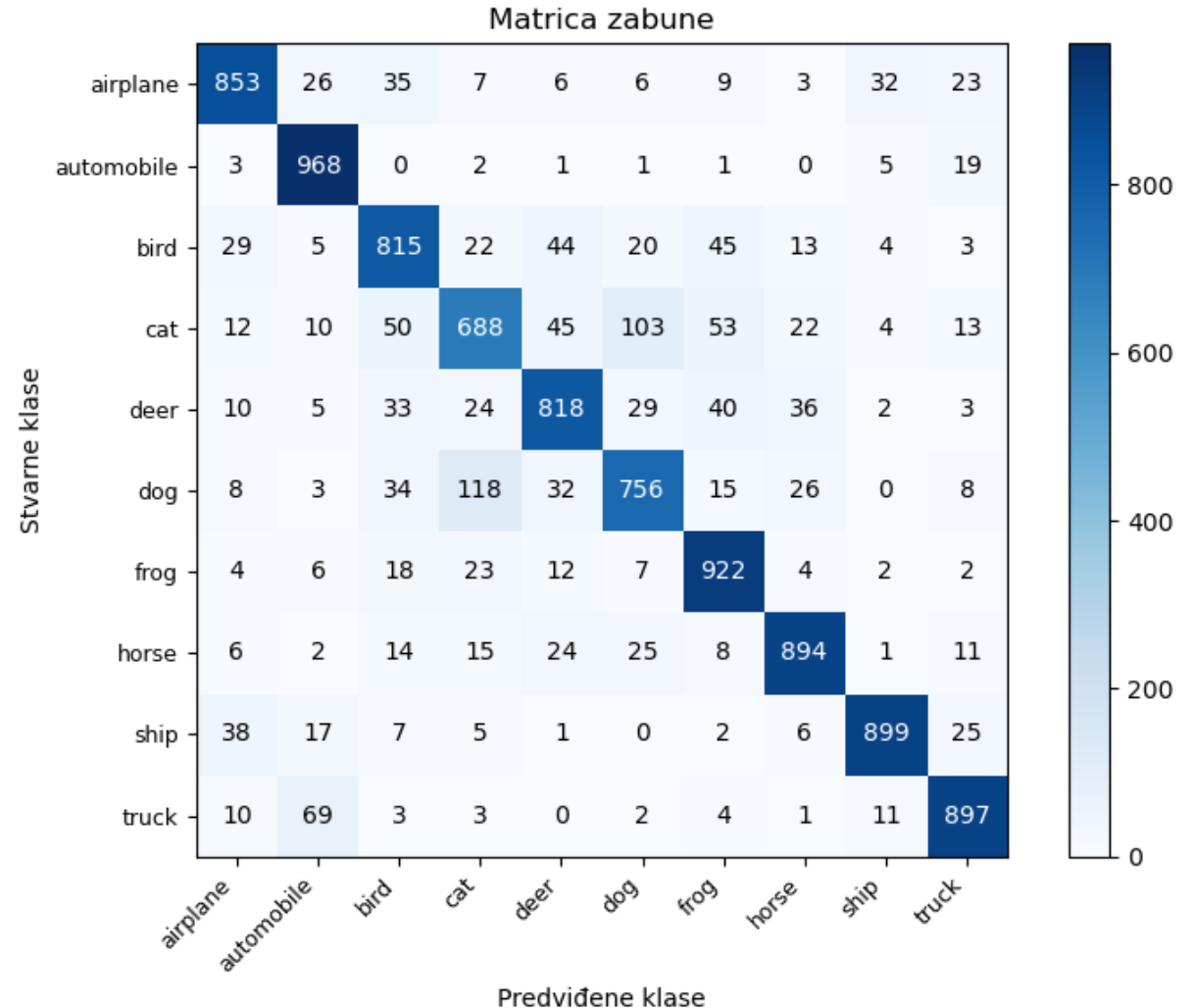
- Tijek treniranja mreže uz korištenje augmentacije

I dalje imamo blagi *overfitting*



Primjer - klasifikacija CIFAR-10 podatkovnog skupa pomoću CNN

- Kao rezultat uzimamo model s parametrima iz 93. epohe
- Točnost na testnom skupu ovog modela je: **85.10%**
- Značajno bolji rezultat u usporedbi s osnovnim pristupom (oko **10%** bolji rezultat na testnom skupu)



Učenje prijenosom (engl. *transfer learning*)

- Složene CNN mogu imati velik broj parametara
- Često naš skup podataka nije dovoljno velik
- Postavlja se pitanje može li se i u tom slučaju efikasno istrenirati CNN (bez problema pretjeranog usklađivanja na podatke?)
- **Učenje prijenosom** (engl. *transfer learning*) je tehnika u okviru strojnog učenja pri čemu se za izradu modela koristi već istrenirani model kao polazna točka
 - Omogućuje treniranje modela i kad nemamo velik skup podataka
 - Značajno smanjuje vrijeme treniranja
 - U mnogim primjenama se pokazao kao vrlo dobar pristup!

Učenje prijenosom (engl. *transfer learning*)

- Ideja je preuzeti naučene značajke na jednom problemu i primijeniti ih na novi, sličan problem

Imagenet: 1million



My dataset: 1,000



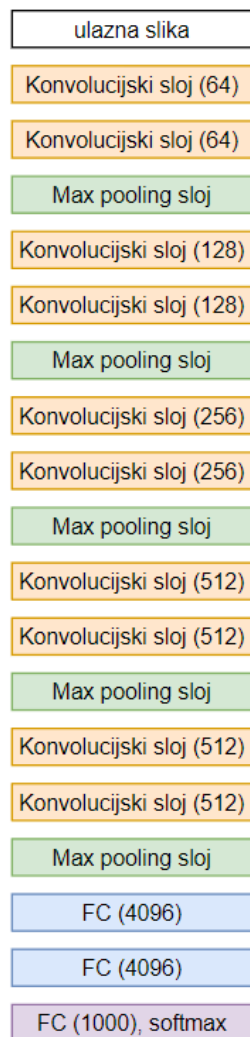
Model istreniran na Imagenet

Primijetite da oznake u skupovima podataka nisu jednake

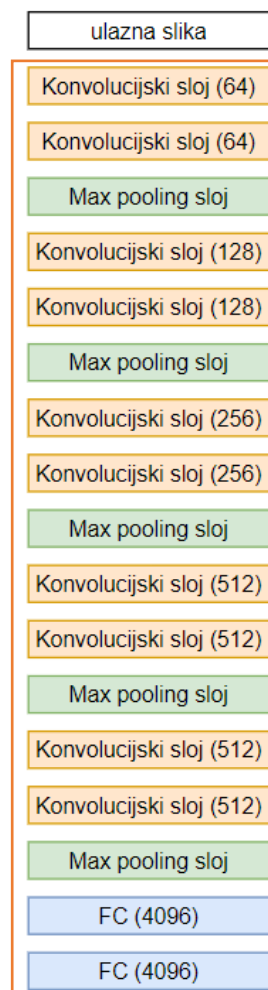
Konačni model

Učenje prijenosom (engl. *transfer learning*)

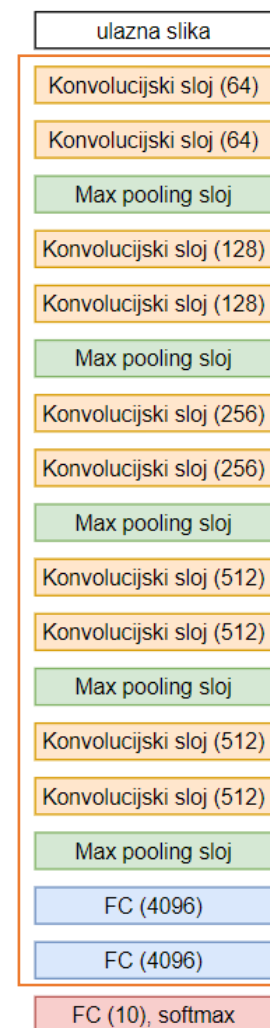
1. CNN mreža istrenirana na ImageNet (npr. VGG)



2. Izbacuje se zadnji sloj i mreža se „zamrzava”



3. Dodaje se novi sloj koji se trenira na novom skupu podataka



4. Dobro je na kraju cijelu mrežu istrenirati na novom skupu podataka s vrlo niskom stopom učenja