Vještačka inteligencija

Predavanje 3: Klasifikacija

Jedinu pravu sigurnost u današnjem svijetu čovjeku mogu pružiti znanje, iskustvo i sposobnost.
~Henry Ford

Odgovorna nastavnica: Vanr. prof. dr Amila Akagić

Univerzitet u Sarajevu

Uvodne informacije

Najave

- Danas u 14:00 je prvi kviz u okviru realizacije vježbi.
  - 5 pitanja, 5 minuta vremena.
  - Automatsko ocjenjivanje nakon isteka vremena.

Aktuelnosti / Vještačka inteligencija

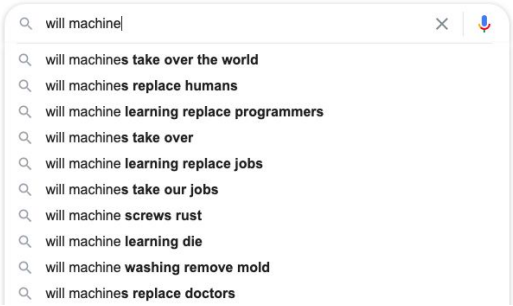
- Da li će VI (AI) ugrožavati / ugrožava radna mjesta? Oblast pružanja usluga?

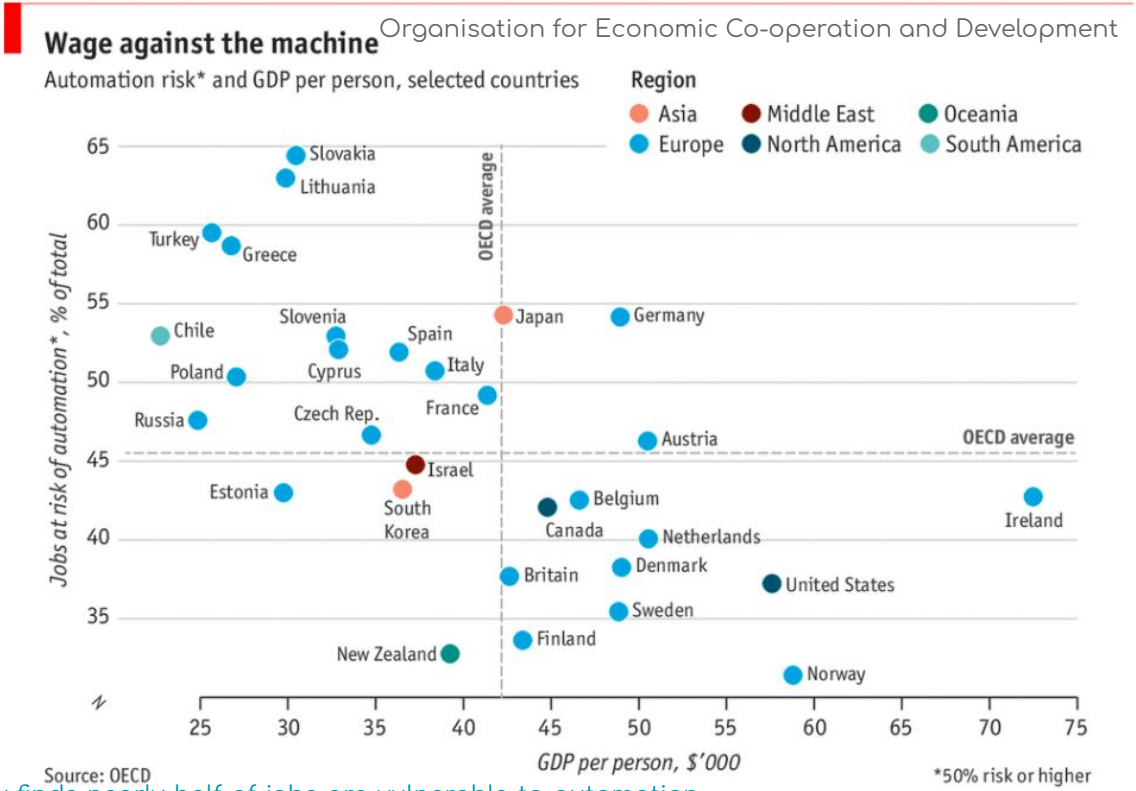
Daily chart

A study finds nearly half of jobs are vulnerable to automation

That could free people to pursue more interesting careers

Google





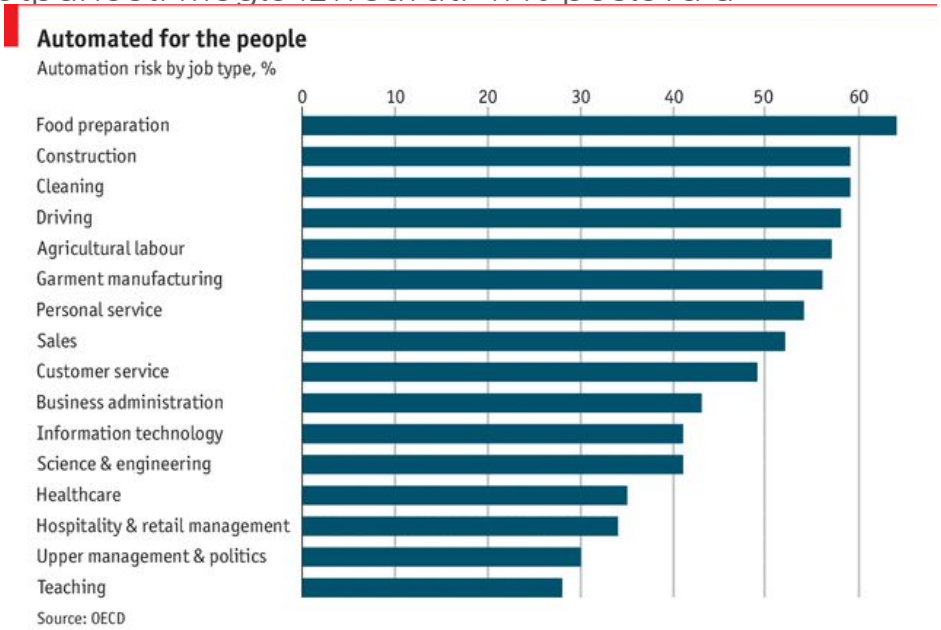
https://www.economist.com/graphic-detail/2018/04/24/a-study-finds-nearly-half-of-jobs-are-vulnerable-to-automation

Aktuelnosti / Vještačka inteligenc

In April, the OECD criticised **an influential 2013 forecast by Oxford University** that found about 47% of jobs in the US in 2010 and 35% in the UK were at "high risk" of being automated over the following 20 years.

The OECD instead put the US figure **at about 10% and the UK's at 12%** - although it did suggest many more workers would see their tasks changing significantly.

- U 2013. godini, Carl Benedikt Frey i Michael Osborne sa Univerziteta Oxford koristili su algoritam mašinskog učenja kako bi procijenili koliko lagano se može automatizirati ~700 različitih vrsta poslova u Americi.
  - Zaključili su da bi računari (mašine) u potpunosti mogle izvršavati 47% poslova u narednih deset ili dvadeset godina.
  - Koji su to poslovi?
  - Poslovi u Slovačkoj su dva puta više ranjiviji nego poslovi u Norveškoj.
  - Generalno, poslovi u bogatijim državama su znatno manje ranjiviji.
  - Na primjer, u Južnoj Koreji 30% radnih mjesta je u prerađivačkoj industriji, u usporedbi s 22% u Kanadi. Bez obzira na to, u prosjeku je teže automatizirati korejske poslove nego kanadske (zbog vrste posla).



Organisation for Economic Co-operation and Development

https://www.economist.com/graphic-detail/2018/04/24/a-study-finds-nearly-half-of-jobs-are-vulnerable-to-automation

Aktuelnosti / Vještačka inteligencija

People also ask

Will GPT 3 kill coding?

GPT-3 would never **kill** jobs skilled developers. Instead its a wake-up call for cargo coders and developers. It'**ll** urge them to buckle up and upskill to ensure they're up for solving complex computer **programming** problems.

Quora

Search for questions, people, and topics

Software Development | Software Developers | Computer Programmers | Computer Science | Software Engineering | Software and Applications | Computer Programming

Will GPT-3 replace software developers?

3 Answers

Alastair Stell, Javascript, Python, C++, C#, Smalltalk, C, Web Programming

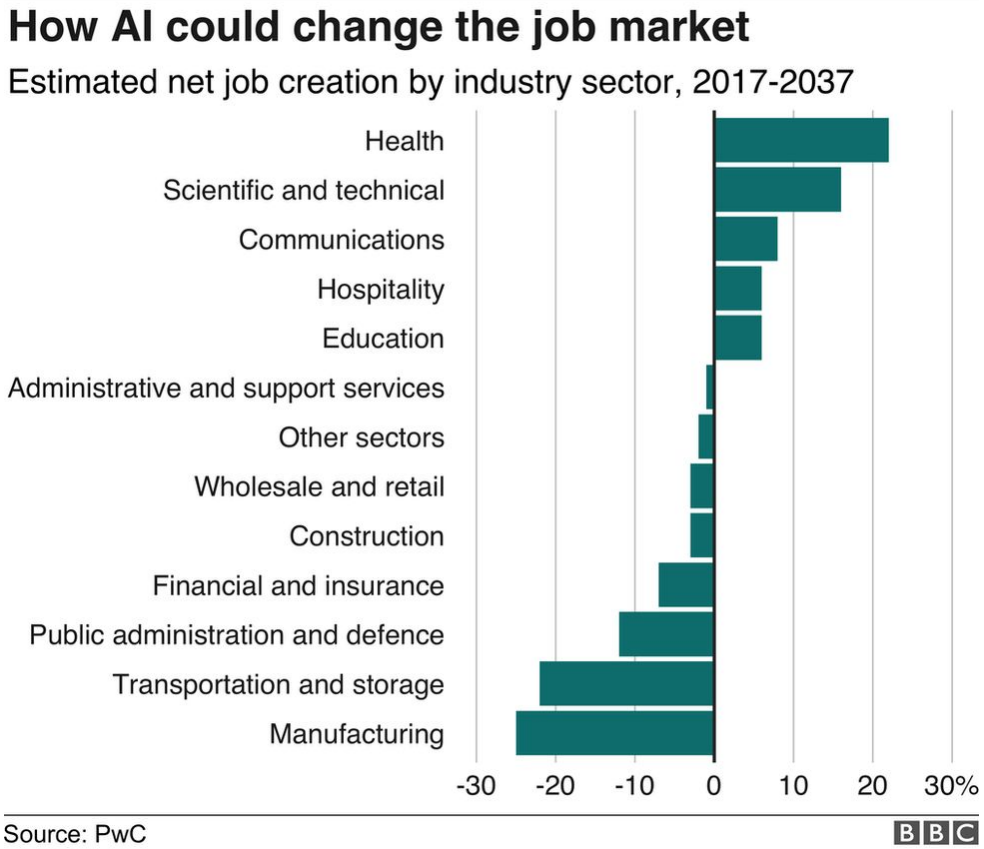
Answered July 25, 2020 · Author has **440** answers and **649.1K** answer views

Coders? Yes. Software engineers? Not so much. GPT-3 rather like Watson, is essentially focused on machine learning. It doesn't have creative thought and problem solving is restricted to "what is known" and what can be projected from data. That may sound like AI but it isn't because GPT-3 doesn't understand what it is doing - there is no conscious process involved - which severely limits creativity.

Aktuelnosti / Vještačka inteligencija

- AI će kreirati onoliko poslova koliko i uništi?

"*Vjerovatno će četvrta industrijska revolucija favorizirati one s **jakim digitalnim vještinama**, kao i sposobnostima poput **kreativnosti** i **timskog rada** koje je mašinama teže replicirati.*"

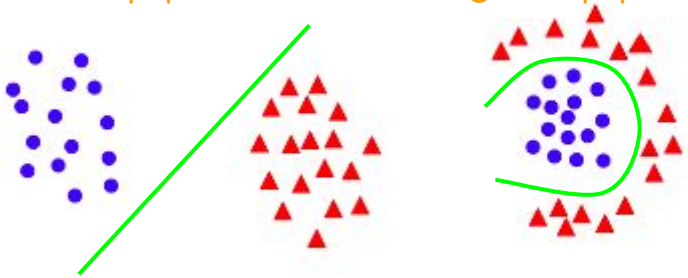


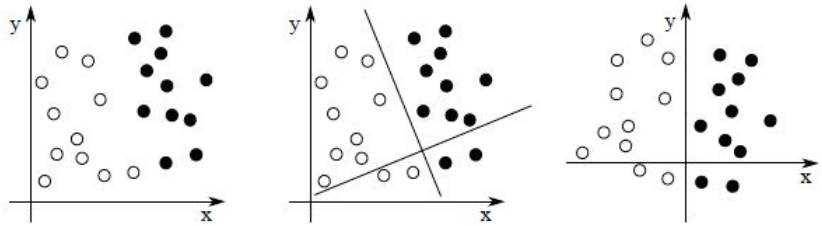
https://www.bbc.com/news/business-44849492

Klasifikator

- **Klasifikator** je bilo koji algoritam koji sortira podatke u označene klase ili kategorije informacija.
  - Primjeri:
    - Filteri za neželjenu poštu koji skeniraju dolazne email-ove i klasificiraju ih kao "neželjenu poštu" (*spam*) ili "poželjenu poštu".
    - Predikcija da li osoba ima neko oboljenje na osnovu predočenih nalaza.
    - Predikcija da li je neka finansijska transakcija lažna (*fraud detection*).
- Klasifikatori su konkretna implementacija prepoznavanja obrazaca (*pattern recognition*) u mnogim oblicima mašinskog učenja.

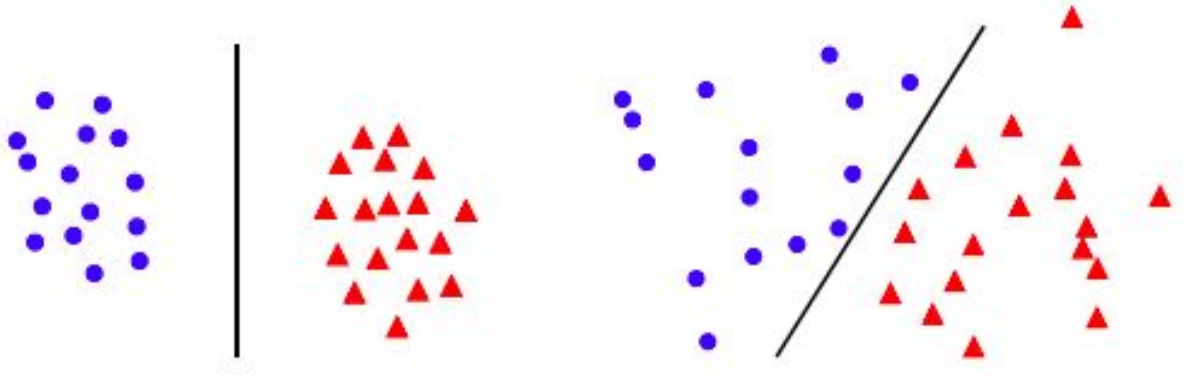
Prvi skup podataka Drugi skup podataka



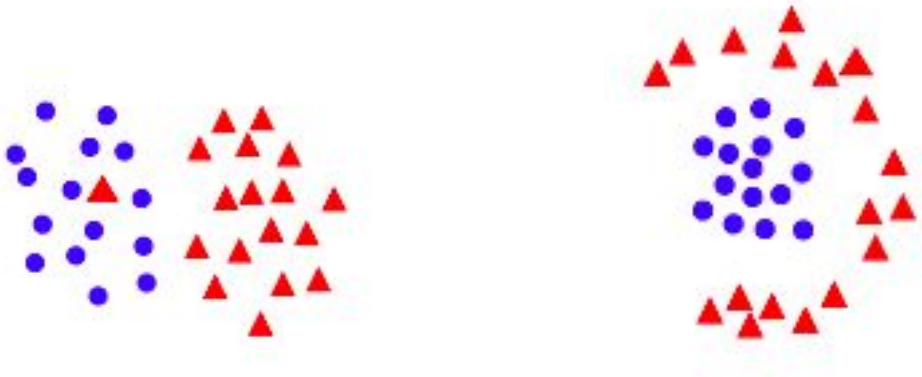


Primjeri skupova

Mogu se klasificirati linearnim klasifikatorom



Ne mogu se klasificirati linearnim klasifikatorom

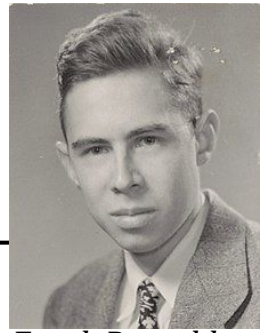


Klasifikator

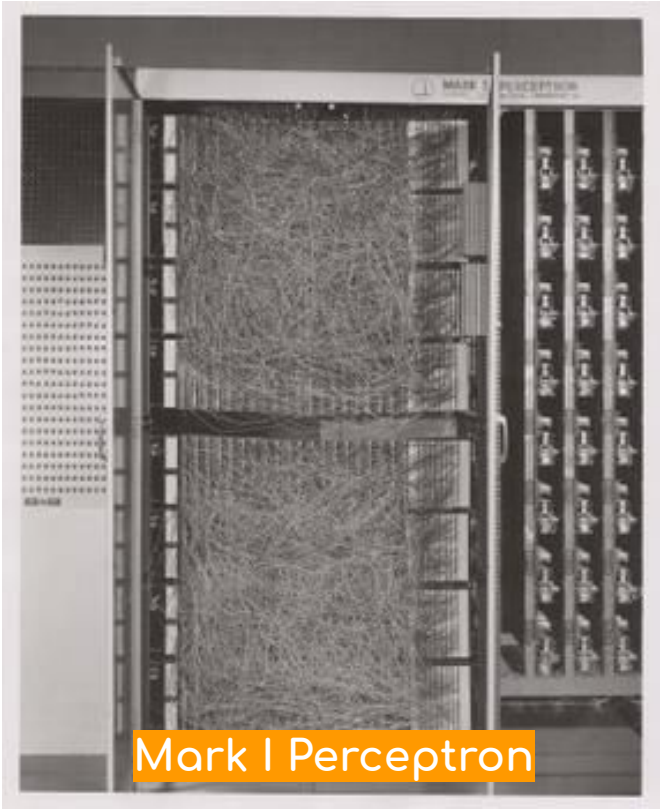
- Primjeri klasifikatora iz prakse:
  - Perceptron
  - Naivni Bayes
  - Stablo odlučivanja
  - Logistička regresija
  - K-Nearest Neighbor
  - Vještačke neuronske mreže / duboko učenje
  - Support Vector Machine (SVM)
  - ....

1958: Perceptron

- Jedan od prvih algoritama koji je imao mogućnost učenja iz podataka.
- Bio je implementiran u hardveru!
  - Vrijednosti koeficijenta su čuvani u potenciometarima, a električni motori su korišteni za njihovo ažuriranje!!!
- Rad perceptrona je demonstriran na prepoznavanju alfabeta sa 20x20 slike. Kamera je koristila fotoćelije kadmijum sulfida.
- Po današnjoj klasifikaciji metoda, ova metoda bi spadala u metode linearnih klasifikatora.



Frank Rosenblatt



Mark I Perceptron

1958
Perceptron

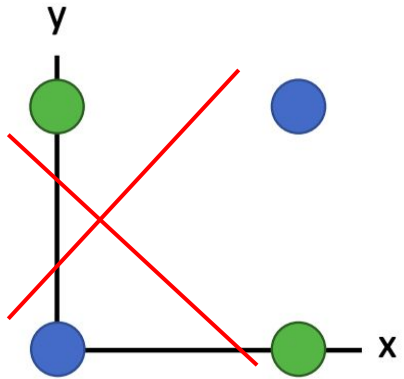
1958: Perceptron

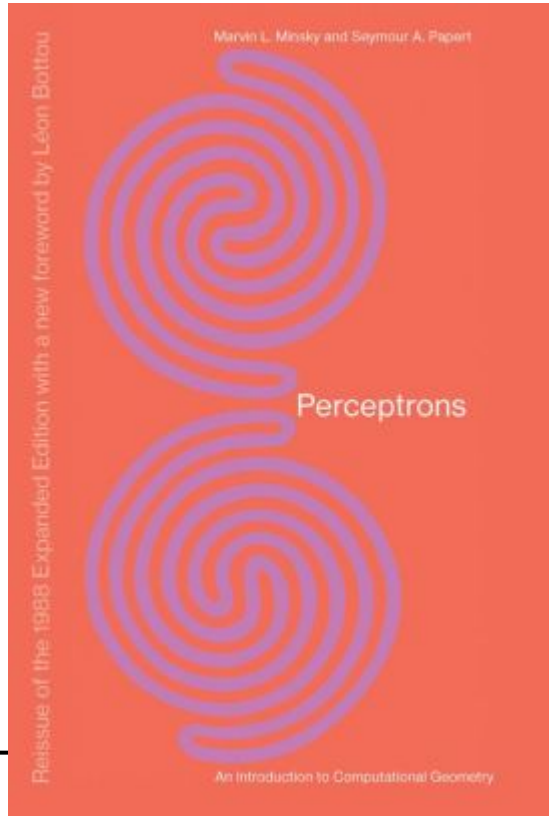
"The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." New York Times Archives, July 8, 1958, Page 25

1969: Knjiga "Perceptrons..." (Minsky & Papert)

- 1969: "*Perceptrons: an introduction to computational geometry*", knjiga autora Marvin **Minsky** i Seymour **Papert.**
  - Matematički dokaz o ograničenjima dvoslojnih *feed forward* perceptrona.

X	Y	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0



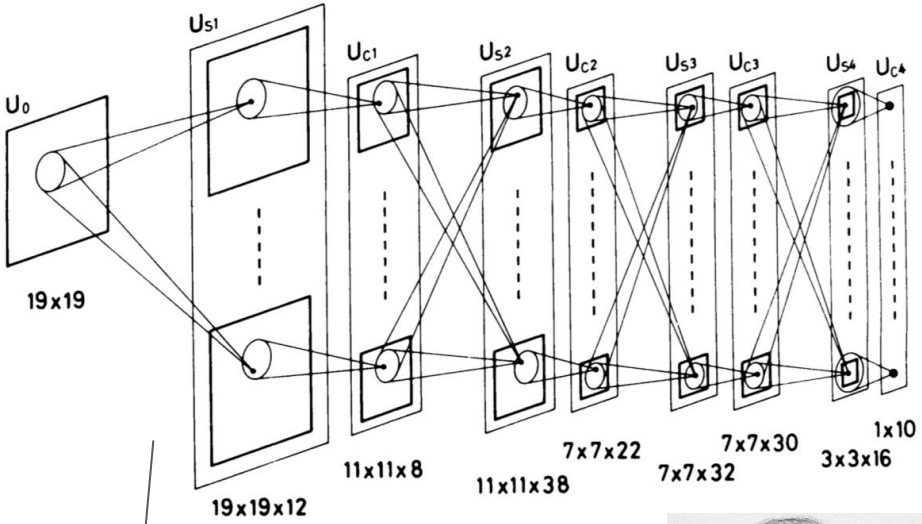


1958 Perceptron

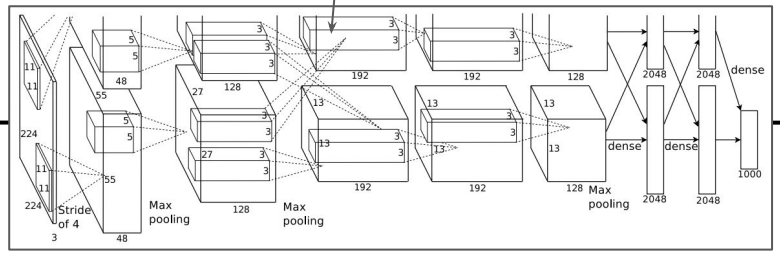
1969 Minsky & Papert

1980: Neocognitron

- 1980: Novi algoritam za prepoznavanje šablona (*pattern recognition*) baziran na modelu vizuelnog sistema.
- Višeslojna neuronska mreža.
- **Neocognitron** je <u>samoorganizirajuća mreža nastala učenjem bez nadzora.</u>
- Osnovne karakteristike: hijerarhija zasnovana na jednostavnim (**konvolucija**) i složenim (**pooling**) ćelijama.



32 godine!



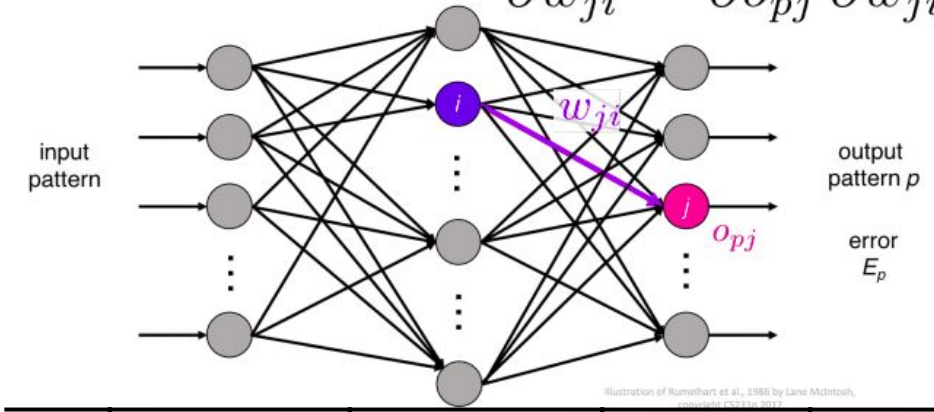
Kunihiko Fukushima

1958 Perceptron | 1969 Minsky & Papert | 1980 Fukushima

1986: Backprop / Rumelhart, Hinton i Williams

- Uvođenjem backpropagacije omogućeno je uspješno treniranje višeslojnih perceptrona.
- Koristi se kod izračunavanja gradijenta u neuronskim mrežama.

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial w_{ji}}$$



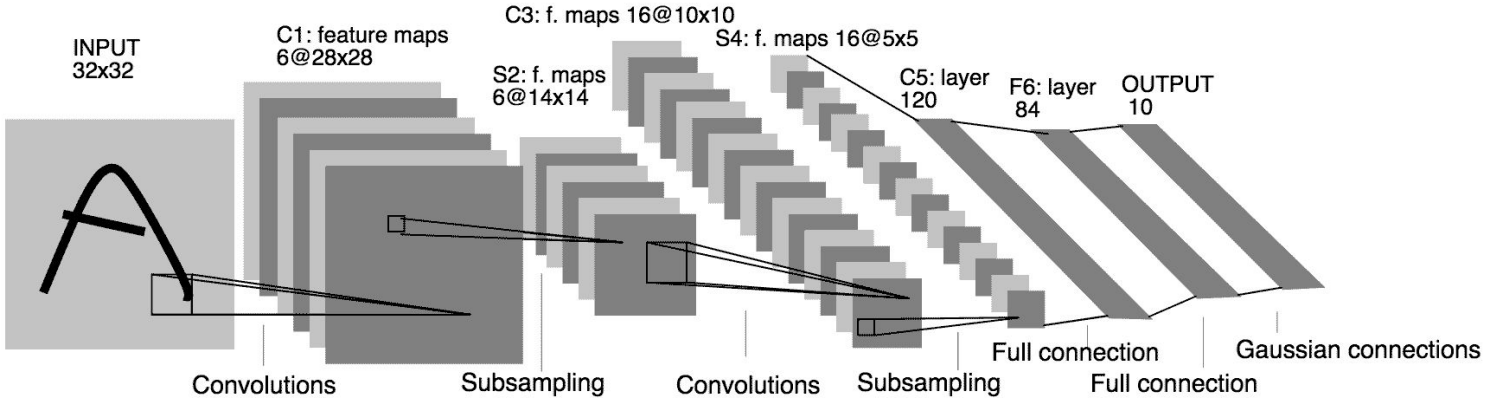
Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

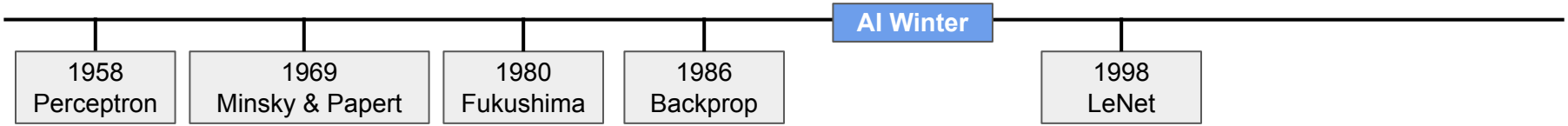
1958 Perceptron | 1969 Minsky & Papert | 1980 Fukushima | 1986 Backprop

1998: Konvolucijske mreže (LeCun)



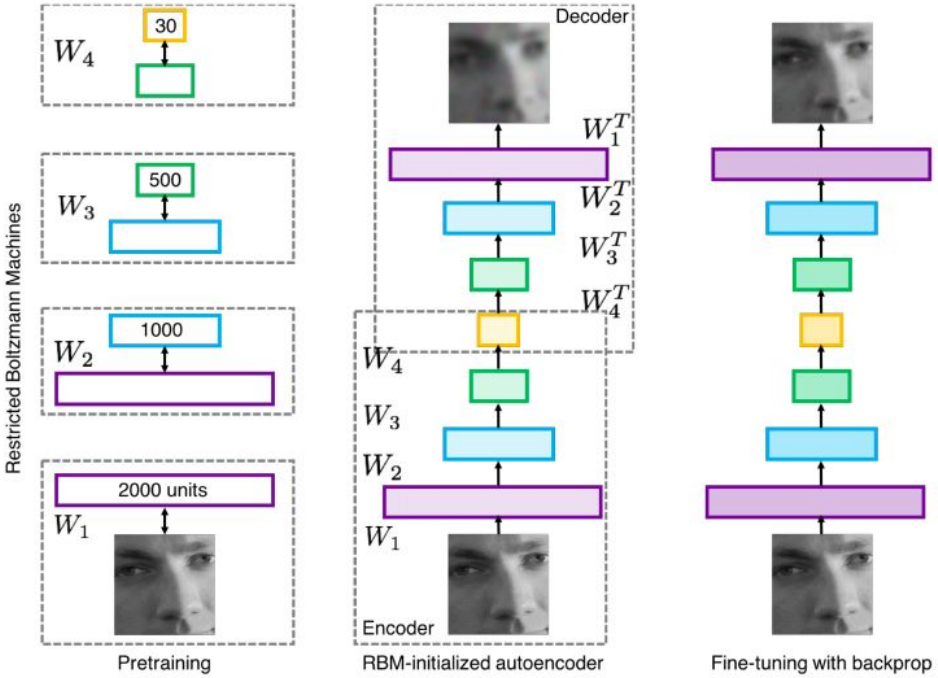
- Prvi put primijenjena backpropagacija na mrežu koja podsjeća na *Neocognitron*.
- Primjena: prepoznavanje rukom pisanih slova alfabeta.
- Sistem je korišten od strane kompanije NEC za prepoznavanje rukom pisanih tekstova.

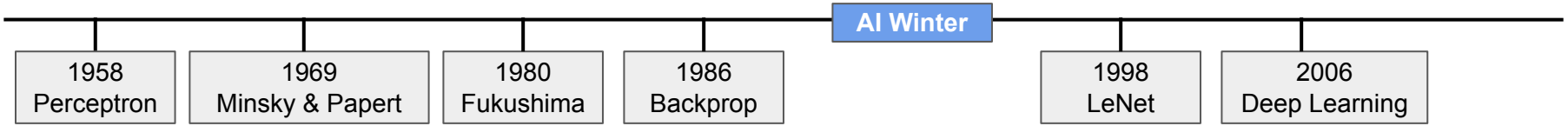
Source: LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998) 2278-2324.



2000-te: Duboko učenje ili Deep Learning (DL)

- Osnovna ideja: nakon uspjeha LeNet-a istraživači su pokušali trenirati sve dublje i dublje mreže (sa više slojeva).
- Trebalo je značajno vremena da ova ideja postane "*mainstream*".
- Značajni radovi iz oblasti:
  - Hinton and Salakhutdinov, 2006
  - Bengio et al, 2007
  - Lee et al, 2009
  - Glorot and Bengio, 2010





2009: ImageNet takmičenje

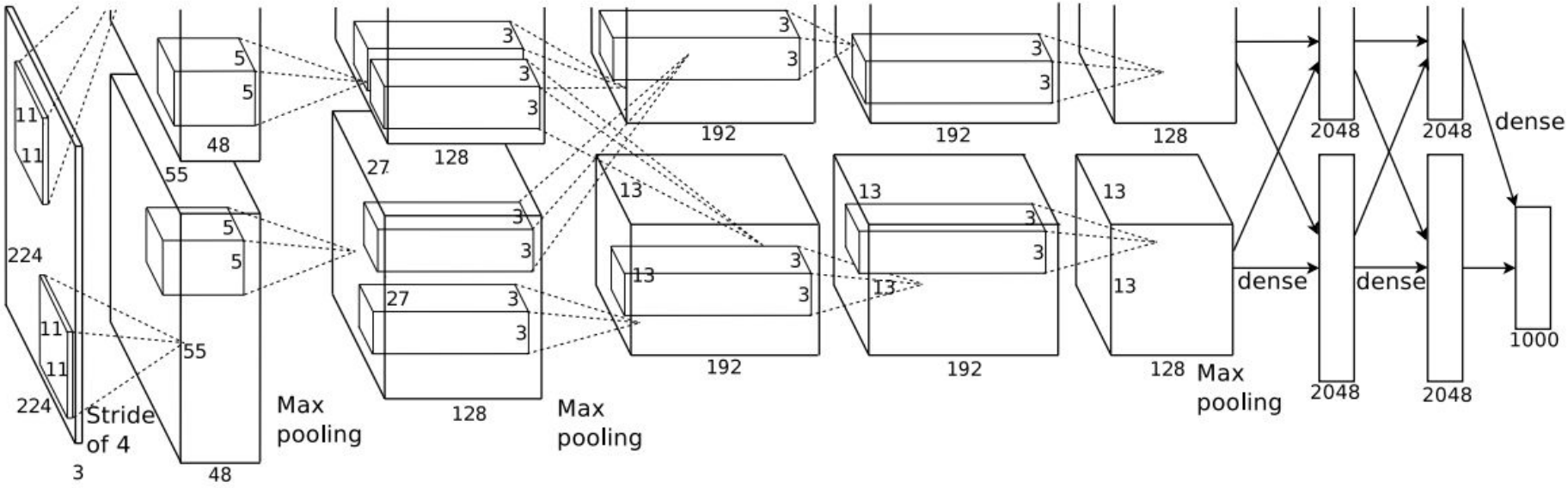
1,000 object classes
1,431,167 images

Primjer izlaza:
Cat
Dog
Car
Mouse
Hat

2012: AlexNet

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, University of Toronto
Ilya Sutskever, University of Toronto
Geoffrey E. Hinton, University of Toronto



ImageNet Classification with Deep Convolutional Neural Networks, Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

2012: Primjeri korištenja ConvNets

Image Classification

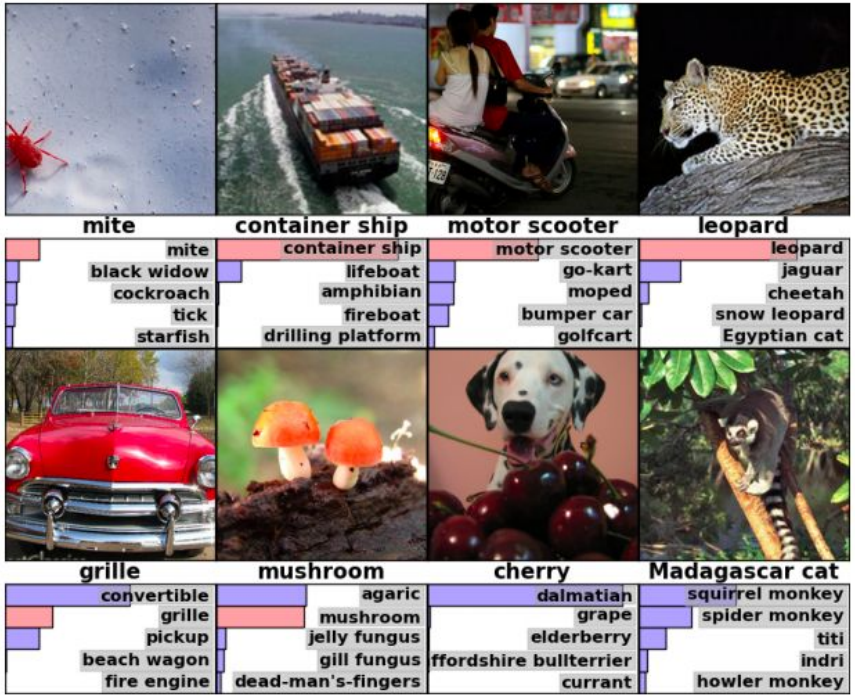
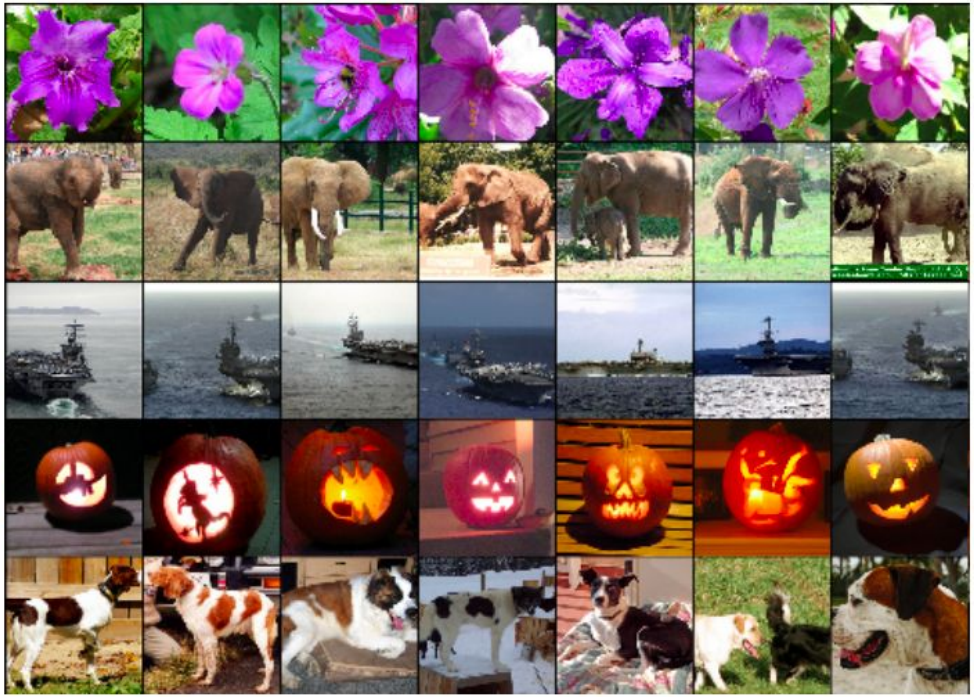


Image Retrieval



ImageNet Classification with Deep Convolutional Neural Networks, Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Primjeri korištenja ConvNets

Object Detection

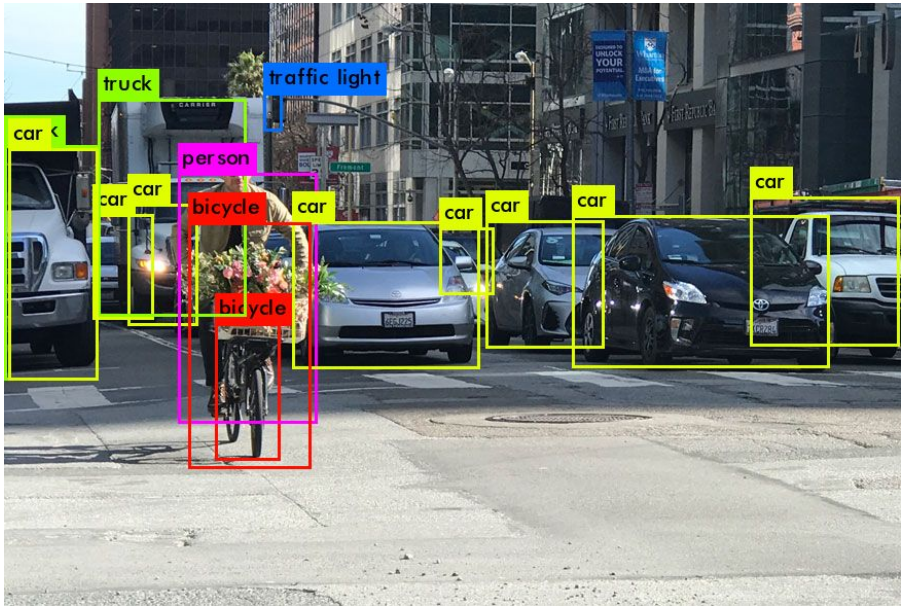


Image Segmentation

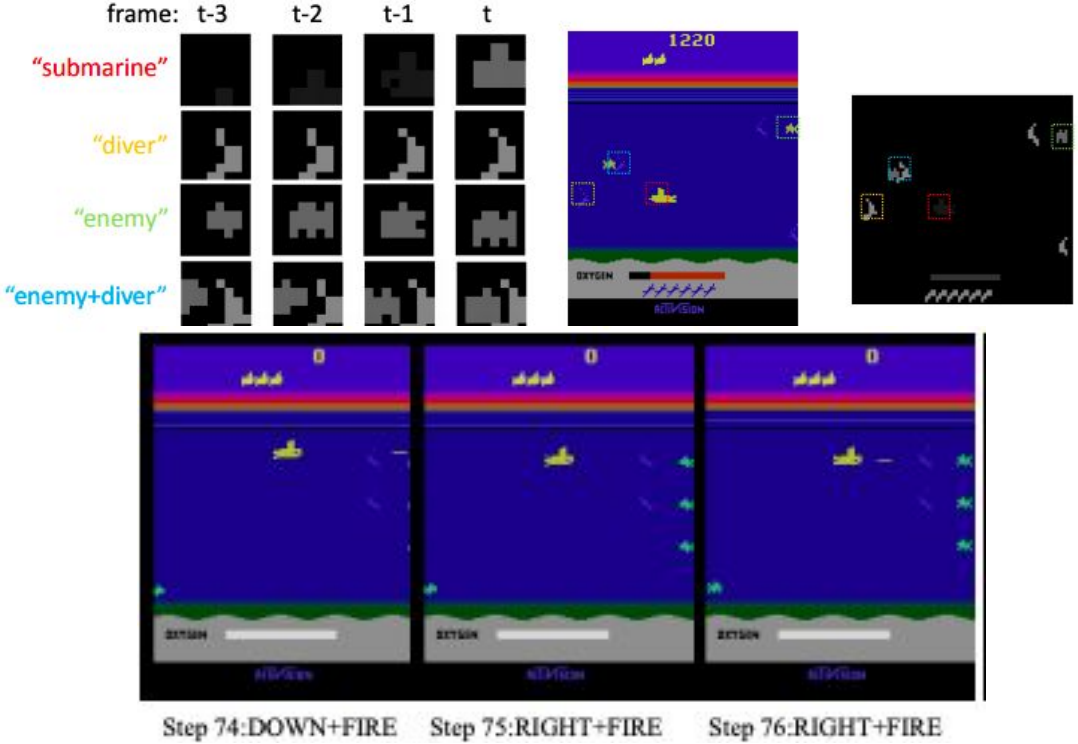
Primjeri korištenja ConvNets

Pose Recognition

DeepPose: Human Pose Estimation via Deep Neural Networks, Alexander Toshev, Christian Szegedy (Google, 2014)
https://arxiv.org/pdf/1312.4659.pdf

Sistem za igranje igrica?

Playing Atari games



Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning (2014)
https://papers.nips.cc/paper/5421-deep-learning-for-real-time-atari-game-play-using-offline-monte-carlo-tree-search-planning.pdf

Kako koristiti podatke za proces učenja?

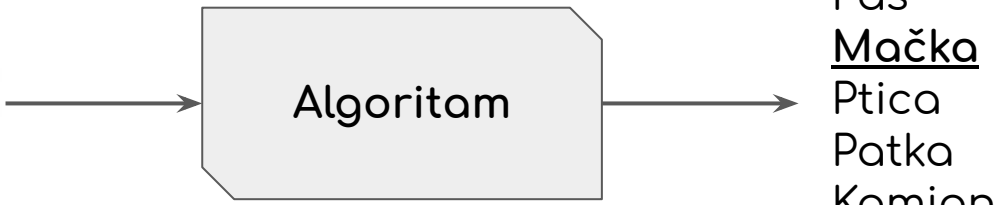
- Prošli put smo vidjeli različite vrste podataka koje koristimo za treniranje različitih modela za učenje.
- Koji je najbolji način da ih koristimo? Koje su to ustaljene prakse podjele jednog skupa podataka?
- U slučaju KNN, kako izabrati **K** i **distancu**?
  - **K** i **distanca** su primjeri **hiperparametara** koje biramo prije treniranja!
  - Ove parametre ne učimo iz podataka, nego ih *a priori* određujemo.
  - Izbor ovih parametara će uveliko zavisiti od problema koji pokušavamo riješiti.
  - Najbolje rješenje: Isprobati sve varijacije i odlučiti se za najbolju.

Klasifikacija slika

Klasifikacija slika: osnovni koncept

Ulaz

Izlaz: dodijeli neku predefinisanu kategoriju slici na ulazu



Pas
Mačka
Ptica
Patka
Kamion

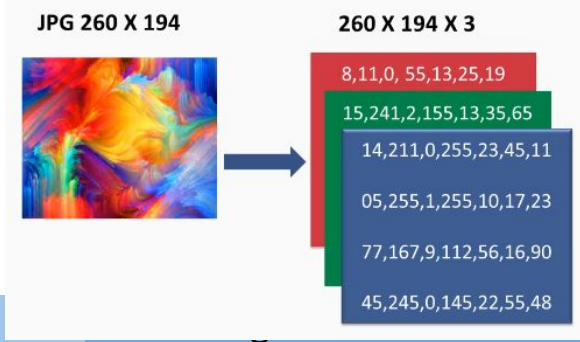
Za čovjeka je klasifikacija slika trivijalan zadatak, međutim za računar je to složen zadatak prvenstveno zbog toga što računar slike vidi u drugačijem obliku (nema jedinstveni vizuelni sistem).

Predstavljanje slika u računaru

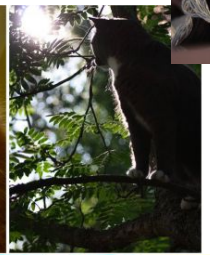
```
[(72, 89, 83), (66, 81, 76), (58, 73, 70), (53, 65,
63), (42, 54, 52), (45, 55, 54), (44, 55, 51), (40,
49, 46), (36, 45, 40), (40, 47, 40), (48, 55, 48),
(55, 62, 54), (50, 53, 44), (66, 69, 60), (109, 112,
103), (122, 125, 116), (124, 127, 116), (119, 122,
111), (99, 101, 88), (109, 111, 98), (124, 126, 112),
(137, 139, 125), (142, 145, 128), (124, 125, 109),
(120, 121, 103), (133, 134, 116), (141, 142, 124),
(159, 160, 142), (165, 159, 143), (154, 148, 132),
(168, 162, 146), (160, 154, 138), (141, 136, 117),
(151, 144, 126), (171, 164, 146), (190, 183, 165),
(187, 180, 161), (193, 185, 166), (195, 187, 168),
(195, 187, 168), (199, 191, 172), (206, 198, 179),
(210, 202, 181), (210, 202, 181), (208, 200, 179),
(197, 189, 168), (181, 170, 150), (158, 147, 127)]
```

→ mačka?

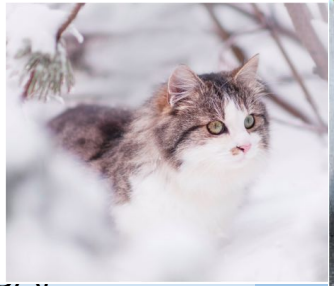
Šta računar vidi?
Šta ako se promijeni ugao kamere?

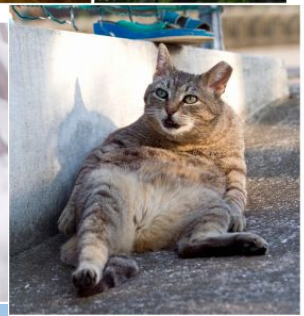


Razni izazovi



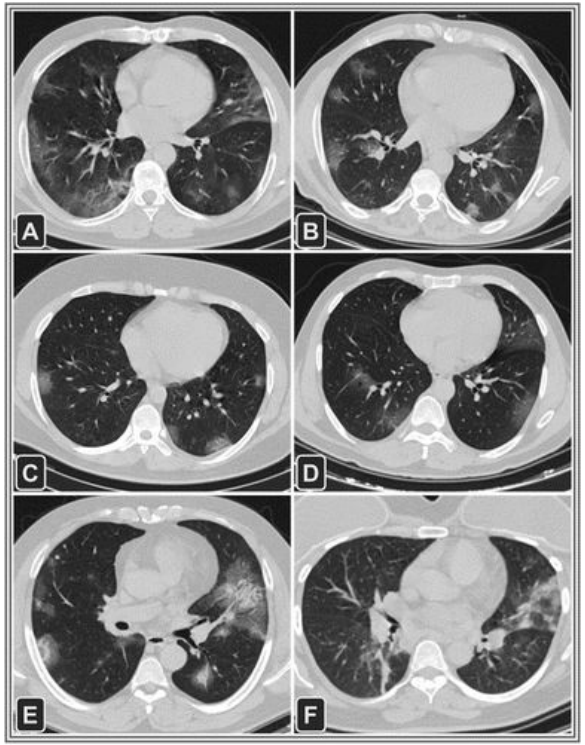
Klasifikator mora biti robustan na različite promjene koje se mogu desiti među slikama.



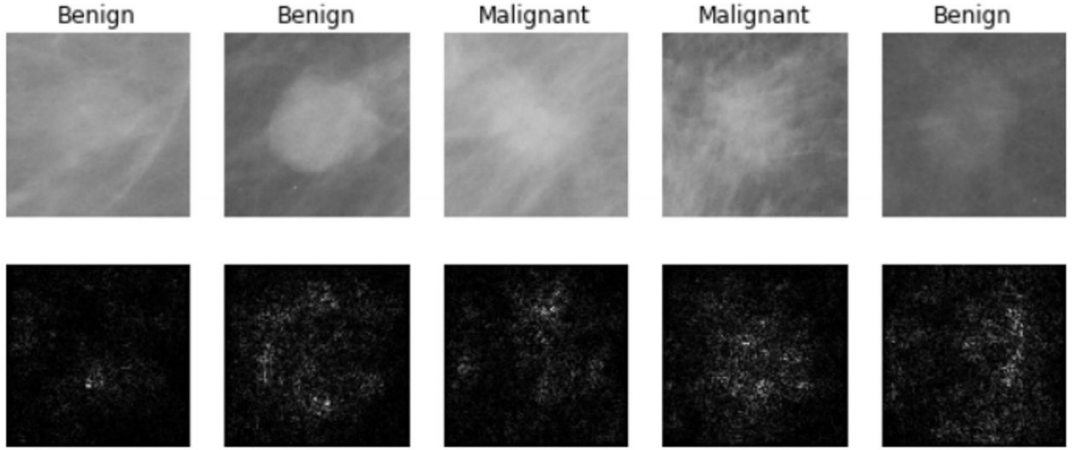


Upotreba klasfikacije slika

Predviđanja ishoda upale pluća kod COVID-19 pacijenata

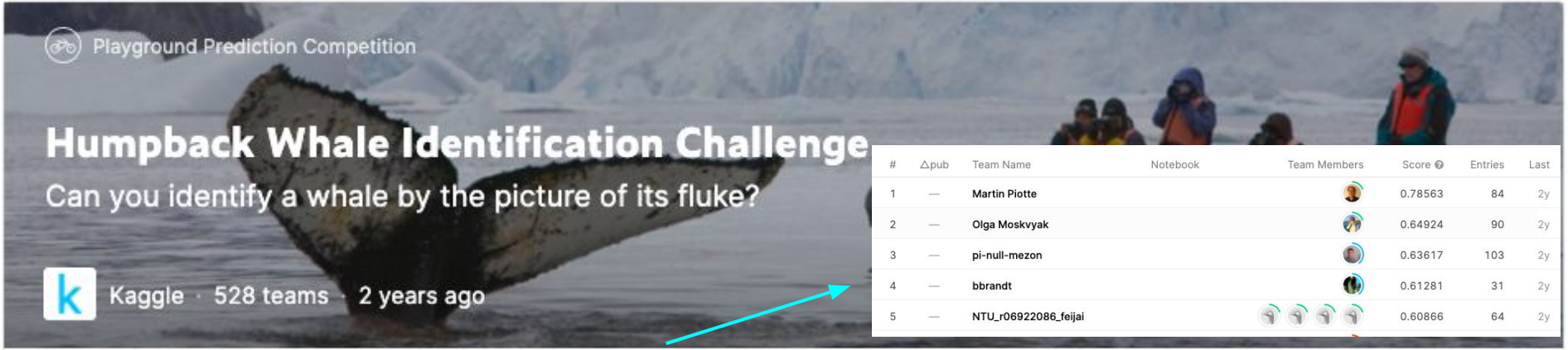


Analiza medicinskih slika



Klasfikacija galaksija

Još jedan primjer: Identifikacija kitova!



#	△pub	Team Name	Notebook	Team Members	Score	Entries	Last
1	—	Martin Piotte			0.78563	84	2y
2	—	Olga Moskvyak			0.64924	90	2y
3	—	pi-null-mezon			0.63617	103	2y
4	—	bbrandt			0.61281	31	2y
5	—	NTU_r06922086_feijai			0.60866	64	2y

Overview Data Notebooks Discussion Leaderboard Rules

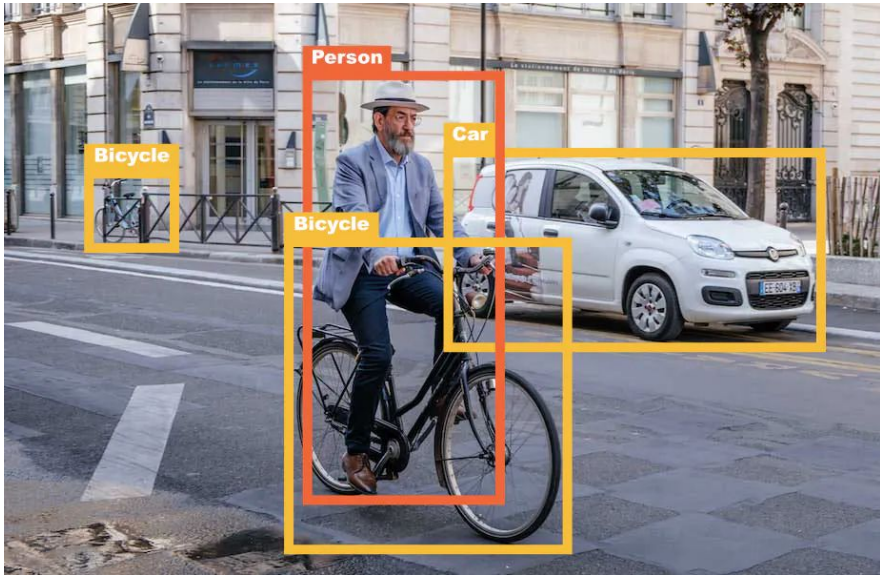
Join Competition

File descriptions

- train.zip - a folder containing the training images
- train.csv - maps the training `Image` to the appropriate whale `Id`. Whales that are not predicted to have a label identified in the training data should be labeled as `new_whale`.
- test.zip - a folder containing the test images to predict the whale `Id`
- sample_submission.csv - a sample submission file in the correct format

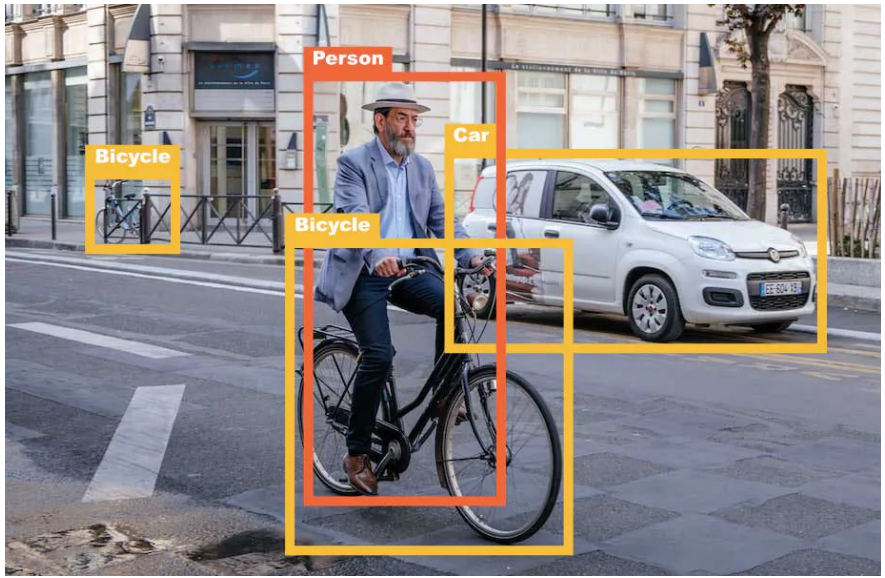
Inside Kaggle you'll find all the code & data you need to do your data science work. Use over 50,000 public datasets and 400,000 public notebooks to conquer any analysis in no time.

Detekcija objekata



automobil
biciklo
vozi
osoba
zgrada
ulica
...

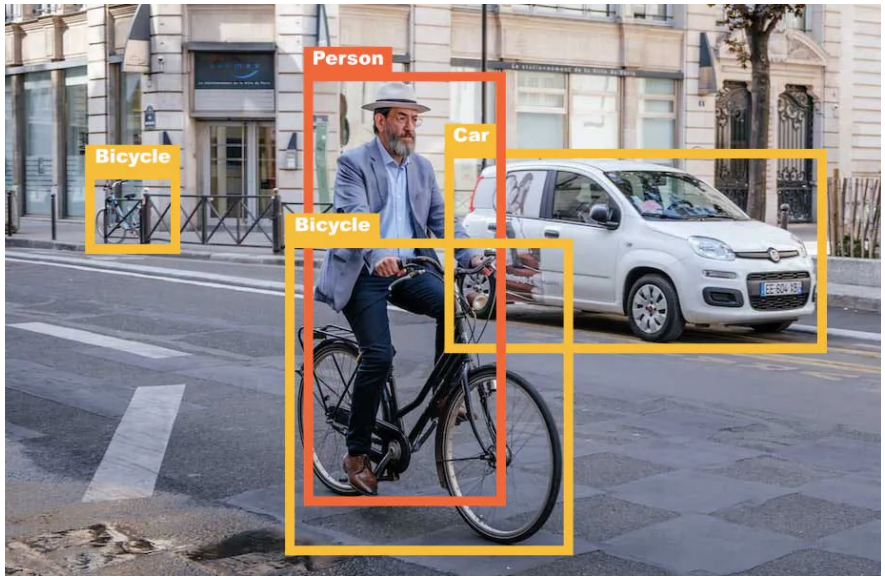
Titlovanje slika / Dodjeljivanje naslova



automobil
biciklo
vozi
osoba
zgrada
ulica
…
<STOP>

Osoba …

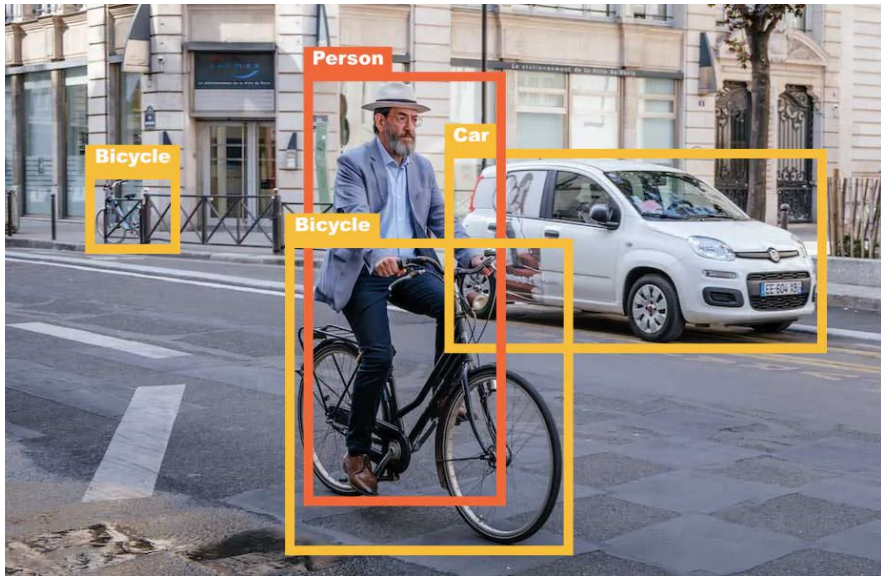
Titlovanje slika / Dodjeljivanje naslova



automobil
biciklo
vozi
osoba
zgrada
ulica
…
<STOP>

Osoba vozi...

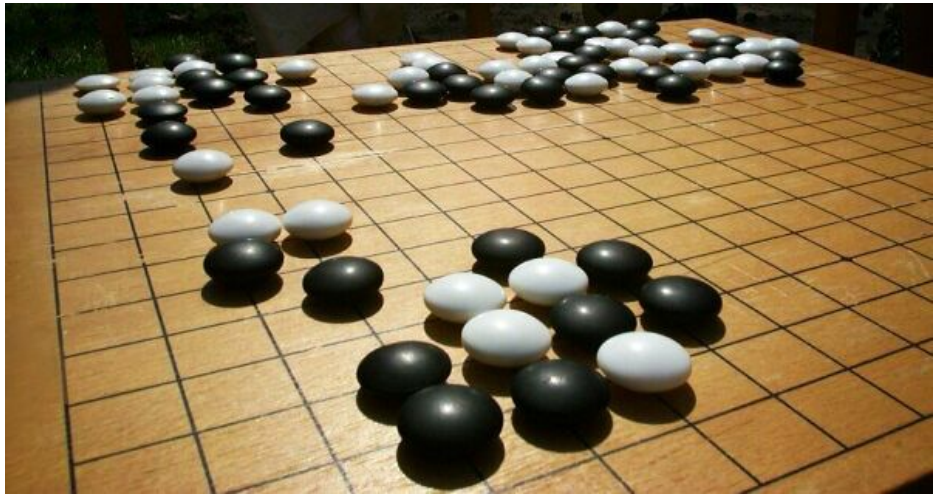
Titlovanje slika / Dodjeljivanje naslova



automobil
biciklo
vozi
osoba
zgrada
ulica
…
<STOP>

Osoba vozi biciklo...

Igrice: koji sljedeći potez da napravim?



$(2, 19)$
$(4, 19)$
$\ldots$
$(1, 19)$
...
$(14, 8)$
...

Sljedeći potez?

Klasifikator slika

Ulaz

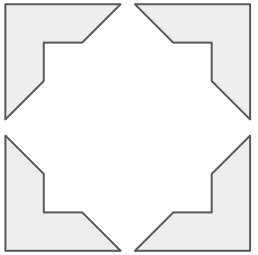
Izlaz: dodijeli neku predefinisanu kategoriju slici na ulazu

Pas
Mačka
Ptica
Patka
Kamion

```
def classify_image(image):
    # some magic here
    return class_label
```

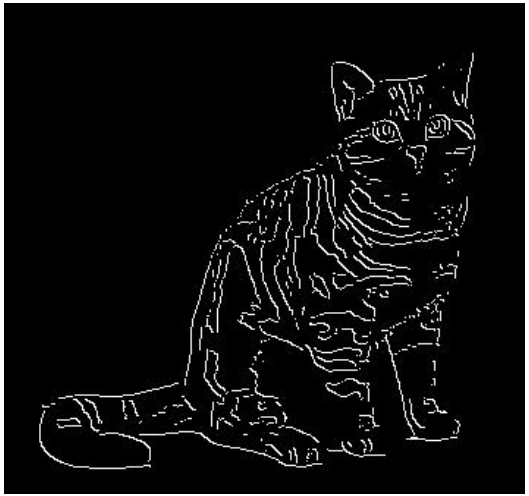
Ne postoji algoritam koji može prepoznati "mačku" na osnovu brojeva. Problem nije jednostavan.

Klasifikator slika

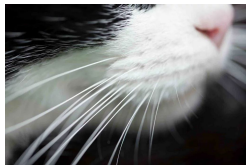


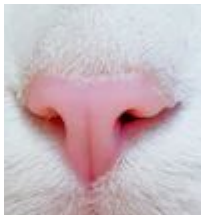
Ulaz

Pronađi ivice



Pronađi neke značajke



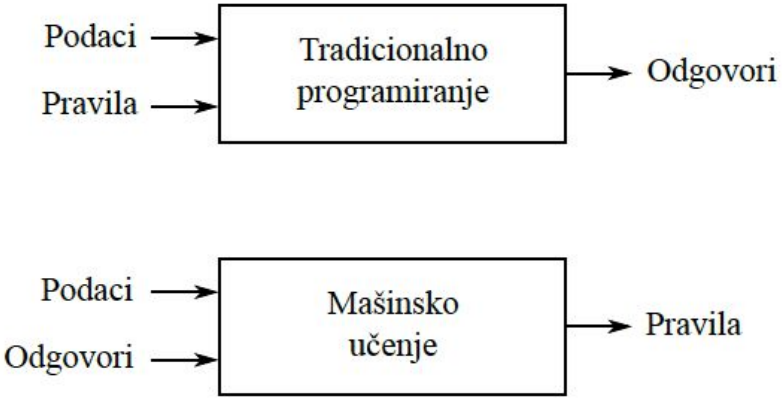


Suština ovakvog pristupa je kodiranje ljudskog znanja o samom izgledu objekta.

Međutim, problem je što će uvijek biti izuzetaka!

Rješenje: mašinsko učenje

- Pristup koji je vođen podacima (*data-driven approach*)
- Ideja: umjesto da ručno kodiramo značajke objekata - **hajde da (na)učimo iz podataka**.
- Uobičajeni koraci:
  - Nađi ili skupi puno podataka i svakom podatku u skupu pridruži labelu.
  - Koristi jedan od metoda mašinskog učenja da istreniraš klasifikator.
  - Napravi evaluaciju klasifikatora nad nekim skupom testnih podataka.



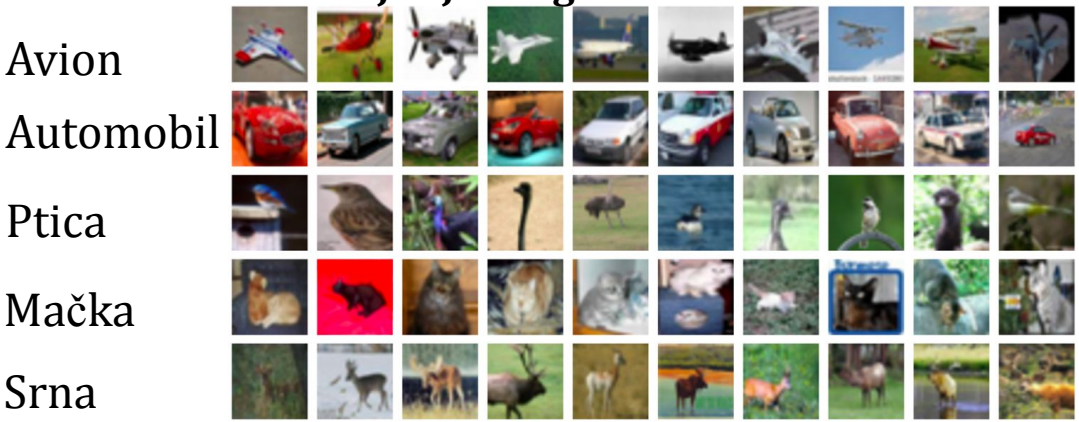
Rješenje: mašinsko učenje

- Nova paradigma programiranja: računar se programira preko podataka koje dovodimo na ulaz programa, a ne preko eksplicitnih pravila.
- Program možemo mijenjati dovođenjem novih podataka!
- Ova paradigma je posebno važna za probleme za koje ne znam kako riješiti ili pristupiti njihovom rješavanju.

```
def train(image):
    # machine learning
    return model
```

```
def predict(image):
# Use model to predict new data
    return test_label
```

Primjer jednog dataset-a



Model mašinskog učenja će pokušati da nauči statističku povezanost između podataka (ulaznih slika i labela).

Poznati izvori podataka: MNIST

10 klasa: Cifre od 0 do 9
28x28 slike sive nijanse
50k slika za treniranje
10k slika za testiranje

http://yann.lecun.com/exdb/mnist/

Poznati izvori podataka: MNIST

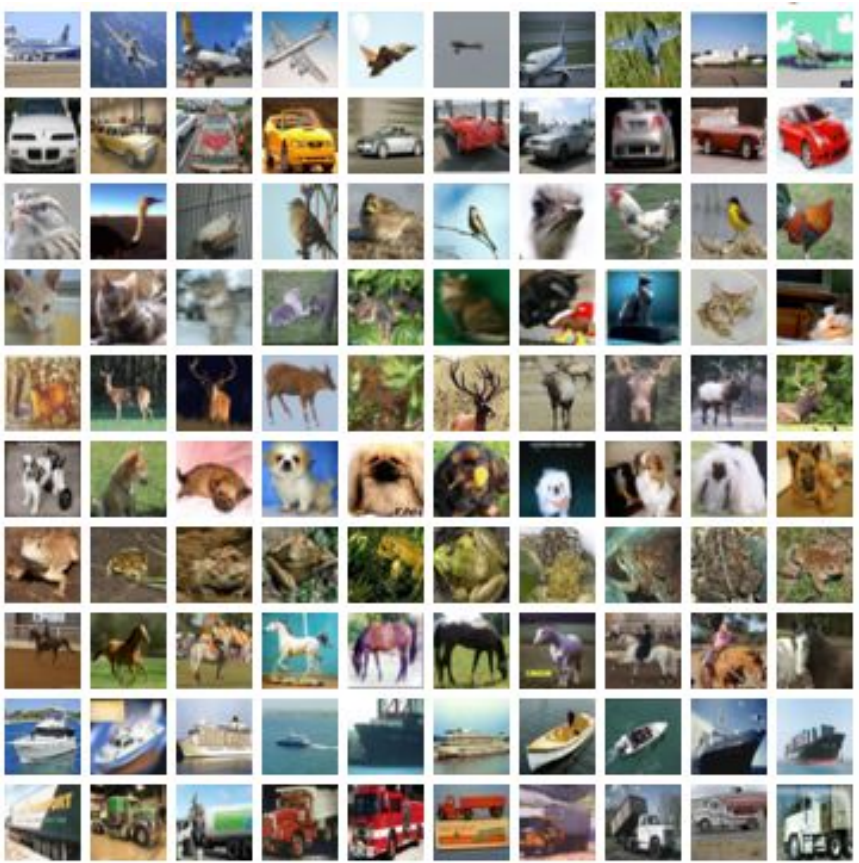
CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
linear classifier (1-layer NN)	deskewing	8.4	LeCun et al. 1998
pairwise linear classifier	deskewing	7.6	LeCun et al. 1998
K-Nearest Neighbors			
K-nearest-neighbors, Euclidean (L2)	none	5.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	deskewing, noise removal, blurring	1.80	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring	1.73	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 pixel shift	1.33	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 pixel shift	1.22	Kenneth Wilder, U. Chicago

Non-Linear Classifiers			
40 PCA + quadratic classifier	none	3.3	LeCun et al. 1998
1000 RBF + linear classifier	none	3.6	LeCun et al. 1998
SVMs			
SVM, Gaussian Kernel	none	1.4	
SVM deg 4 polynomial	deskewing	1.1	LeCun et al. 1998
Reduced Set SVM deg 5 polynomial	deskewing	1.0	LeCun et al. 1998
Virtual SVM deg-9 poly [distortions]	none	0.8	LeCun et al. 1998
Virtual SVM, deg-9 poly, 1-pixel jittered	none	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 1-pixel jittered	deskewing	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 2-pixel jittered	deskewing	0.56	DeCoste and Scholkopf, MLJ 2002
Convolutional nets			
Convolutional net LeNet-1	subsampling to 16x16 pixels	1.7	LeCun et al. 1998
Convolutional net LeNet-4	none	1.1	LeCun et al. 1998
Convolutional net LeNet-4 with K-NN instead of last layer	none	1.1	LeCun et al. 1998
Convolutional net LeNet-4 with local learning instead of last layer	none	1.1	LeCun et al. 1998
large/deep conv. net, 1-20-40-60-80-100-120-120-10 [elastic distortions]	none	0.35	Ciresan et al. IJCAI 2011
committee of 7 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.27 +-0.02	Ciresan et al. ICDAR 2011
committee of 35 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.23	Ciresan et al. CVPR 2012

http://yann.lecun.com/exdb/mnist/

Poznati izvori podataka: CIFAR10

Avion
Automobil
Ptica
Mačka
Srna
Pas
Žaba
Konj
Brod
Kamion



10 klasa
50k slika za treniranje (5k po klasi)
10k slika za testiranje (1k po klasi)
32x32 RGB slika

https://www.kaggle.com/c/cifar-10/overview

#	△pub	Team Nam	Team Members	Score	Entries	Last
1	—	DeepCNet		0.95530	18	6y
2	—	jiki		0.94740	42	6y
3	—	Anil Thoma		0.94300	3	6y
4	—	Frank Shar		0.94190	13	6y
5	—	nagadomi		0.94150	16	6y

Poznati izvori podataka: CIFAR100

- Ovaj skup podataka sličan je CIFAR-10, osim što ima **100 klasa** koje sadrže po 600 slika. Postoji **500 slika za trening** i **100 slika za testiranje po klasi.** 100 klasa u CIFAR-100 grupirano je u **20 super klasa**. Svaka slika dolazi s oznakom "fine" (klasa kojoj pripada) i oznakom "coarse" (superklasa kojoj pripada).

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Poznati izvori podataka: CIFAR100

100 klasa
50k slika za treniranje (500 po klasi)
10k slika za testiranje (100 po klasi)
32x32 RGB slika

20 super klasa u kojoj je po 5 klasa

https://benchmarks.ai/cifar-100

CIFAR-100
Classify 32x32 colour images into 100 categories.

Method (expand all | collapse all)	Accuracy (%)
Big Transfer (BiT): General Visual Representation Learning (Dec 2019) Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, Neil Houlsby Transfer of pre-trained representations improves sample efficiency and simplifies hyperparameter tuning when training deep neural networks for vision. We revisit the paradigm of pre-training on large supervised datasets and fine-tuning the model on a target task. We scale up pre-training, and propose a simple recipe that we call Big Transfer (BiT). By combining a few carefully selected components, and transferring using a simple heuristic, we achieve strong performance on over 20 datasets. BiT performs well across a surprisingly wide range of data regimes -- from 1 example per class to 1M total examples. BiT achieves 87.5% top-1 accuracy on ILSVRC-2012, 99.4% on CIFAR-10, and 76.3% on the 19 task Visual Task Adaptation Benchmark (VTAB). On small datasets, BiT attains 76.8% on ILSVRC-2012 with 10 examples per class, and 97.0% on CIFAR-10 with 10 examples per class. We conduct detailed analysis of the main components that lead to high transfer performance.	93.51%
EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (May 2019, arXiv 2019) Mingxing Tan, Quoc V. Le Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. We demonstrate the effectiveness of this method on scaling up MobileNets and ResNet. To go even further, we use neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets. In particular, our EfficientNet-B7 achieves state-of-the-art 84.4% top-1 / 97.1% top-5 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet. Our EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters. Source code is at https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet.	91.70%

Poznati izvori podataka: ImageNet

1000 klasa

Ukupno slika u skupu podataka: 14,197,122 (okt 2020)

~1.3M slika za treniranje (~1.3K po klasi)
50K slika za validaciju (50 po klasi)
100K slika za testiranje (100 po klasi)

Slike su različite veličine, ali se najčešće smanjuju na 256x256 u procesu treniranja.

Performanse se mjere sa: **Top 5 accuracy** (Algoritam vrši predikciju o 5 klasa za svaku sliku; jedna od tih pet slika mora biti tačna.)

http://image-net.org/index

MIT Places

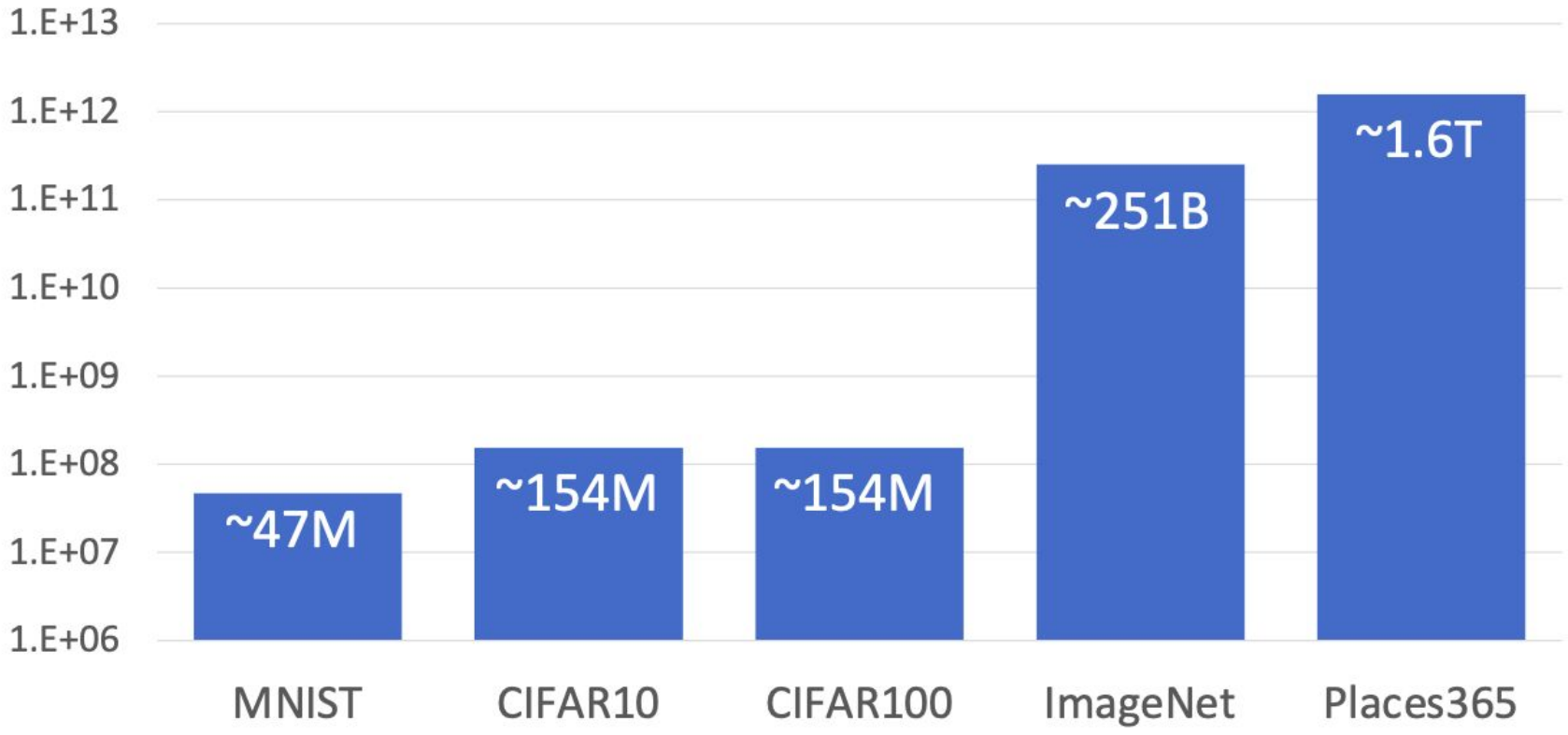
365 klasa različitih vrsta scena

~8M slika za treniranje
18.25K slika za validaciju (50 po klasi)
328.5K slika za testiranje (900 po klasi)

Slike su različite veličine, ali se najčešće smanjuju na 256x256 u procesu treniranja.

http://places.csail.mit.edu/

Poređenje skupova podataka po broju piksela



Prvi klasifikator: Nearest Neighbor

```
def train(image):
     # machine learning
    return model
```

Memoriziraj sve podatke i labele!

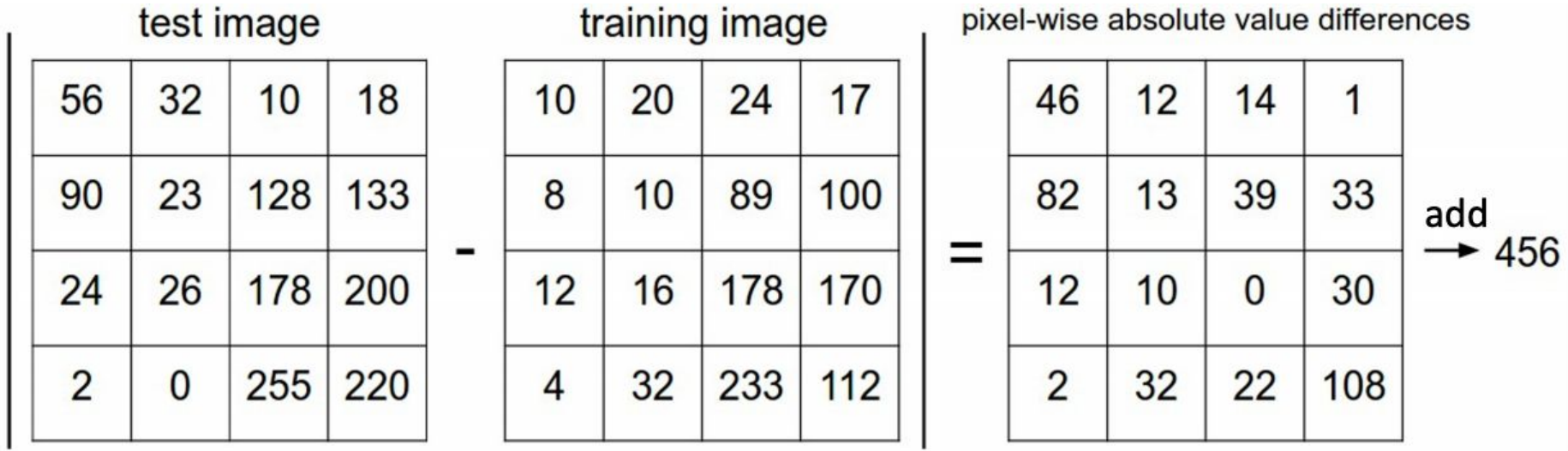
```
def predict(image):
# Use model to predict new data
    return test_label
```

Napravi predikciju na osnovu najsličnije slike korištene u procesu treniranja!

Kako izračunati sličnost između dvije slike?

Metrika za distancu

L1 ili Manhattan distanca: $$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



Prvi klasifikator: Nearest Neighbor

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Memoriziraj podatke za treniranje

Prvi klasifikator: Nearest Neighbor

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Za svaki testni podatak:
- Pronađi najsličniju sliku iz skupa za treniranje
- Vrati labelu te slike

Prvi klasifikator: Nearest Neighbor

```python
import numpy as np

class NearestNeighbor:
  def __init__(self):
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):
      # find the nearest training image to the i'th test image
      # using the L1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Ako imamo N slika u skupu za treniranje, koliko je brzo treniranje?

O(1)

Koliko je brzo testiranje?

O(N)

U praksi se sporo treniranje toleriše, ali ne i sporo testiranje.

Postoji mnogo metoda koje efikasnije mogu da izvrše klasifikaciju:

https://github.com/facebookresearch/faiss

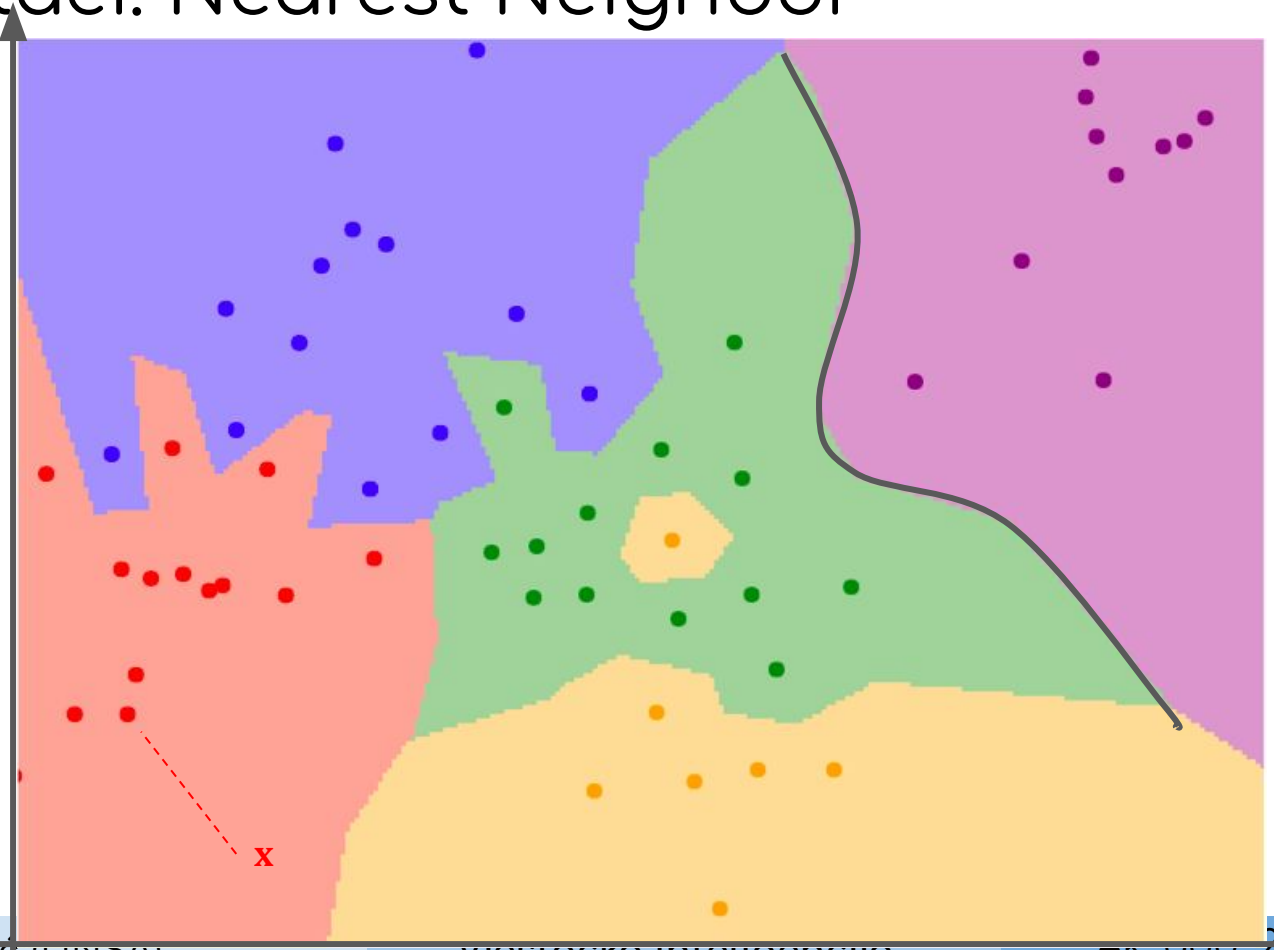
Prvi klasifikator: Nearest Neighbor

Nedostaci: Nearest Neighbor

Tačke predstavljaju podatke iz skupa za treniranje.

Boja podloge predstavlja klasu.

Outliers



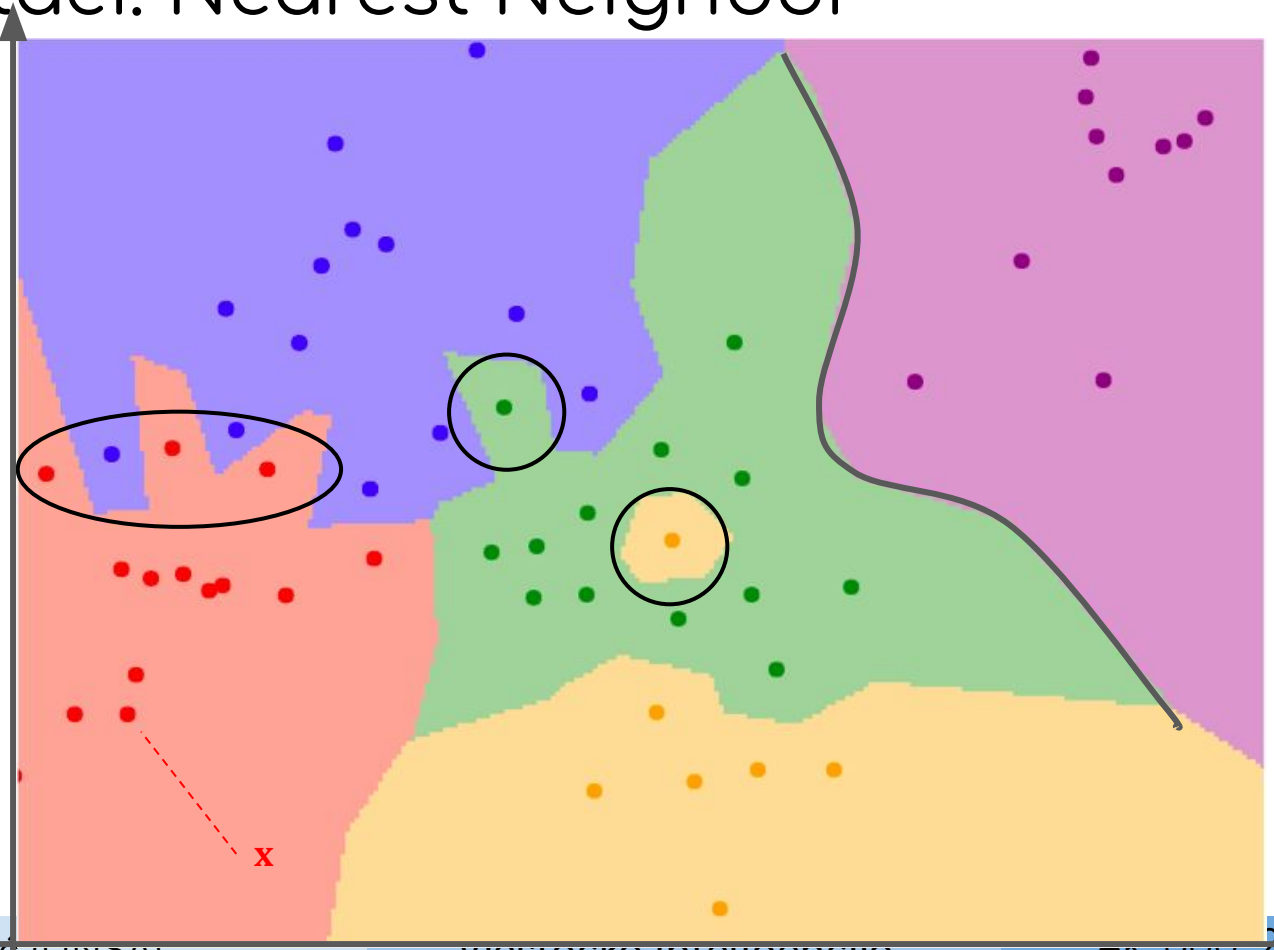
Postoje granice između dva skupa, kao što je prikazano crnom linijom.

Da li je moguće definisati novi algoritam koji će dati preciznije granice?

Nedostaci: Nearest Neighbor

Tačke predstavljaju podatke iz skupa za treniranje.

Boja podloge predstavlja klasu.



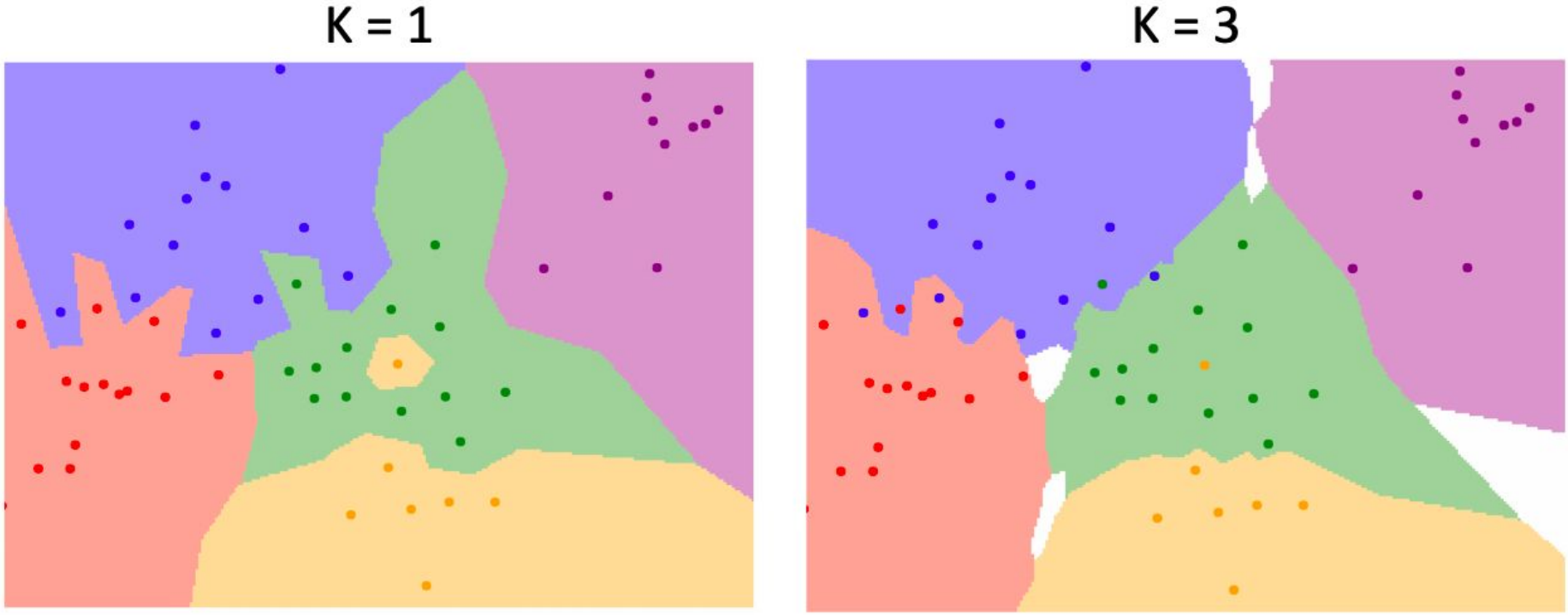
Postoje granice između dva skupa, kao što je prikazano crnom linijom.

Granice nisu dovoljno precizno određene! (Outliers)

Da li je moguće definisati novi algoritam koji će dati preciznije granice?

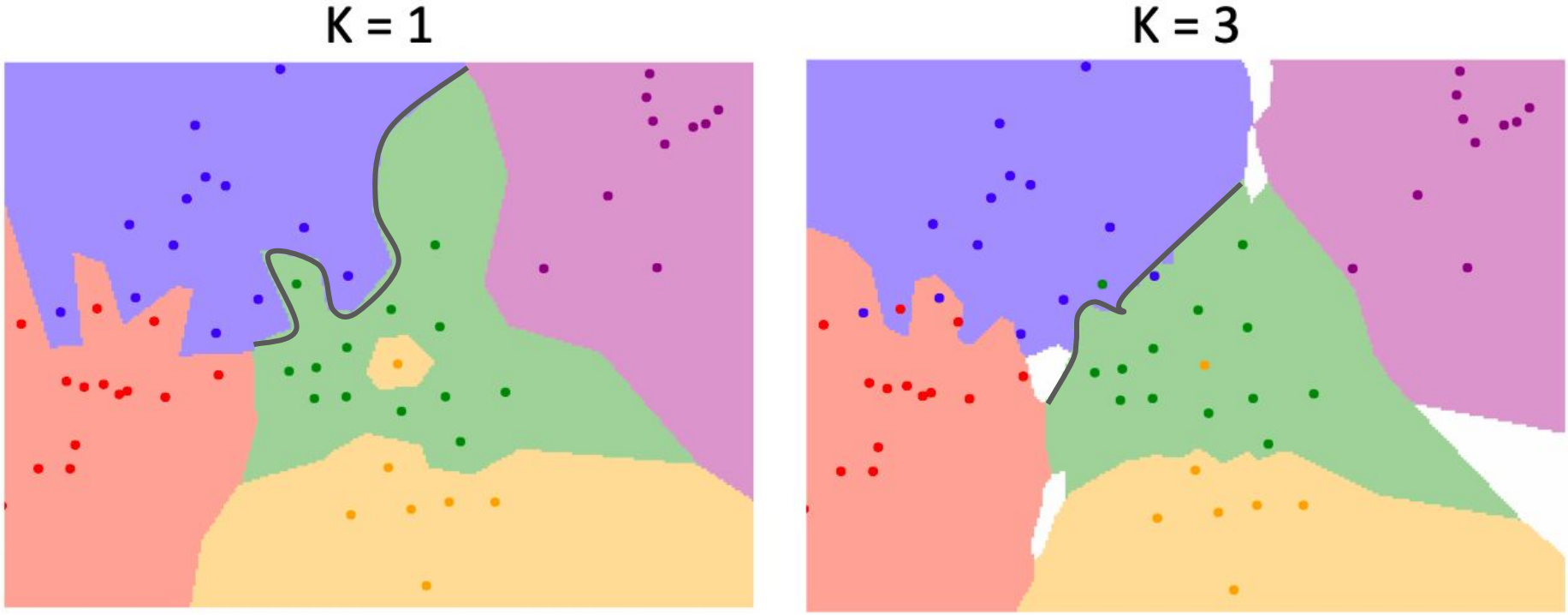
k-Nearest Neighbor

- Osnovna ideja: Uzeti k najsličnijih slika i "glasati" koju od njih uzeti kao klasu slike koju testiramo.



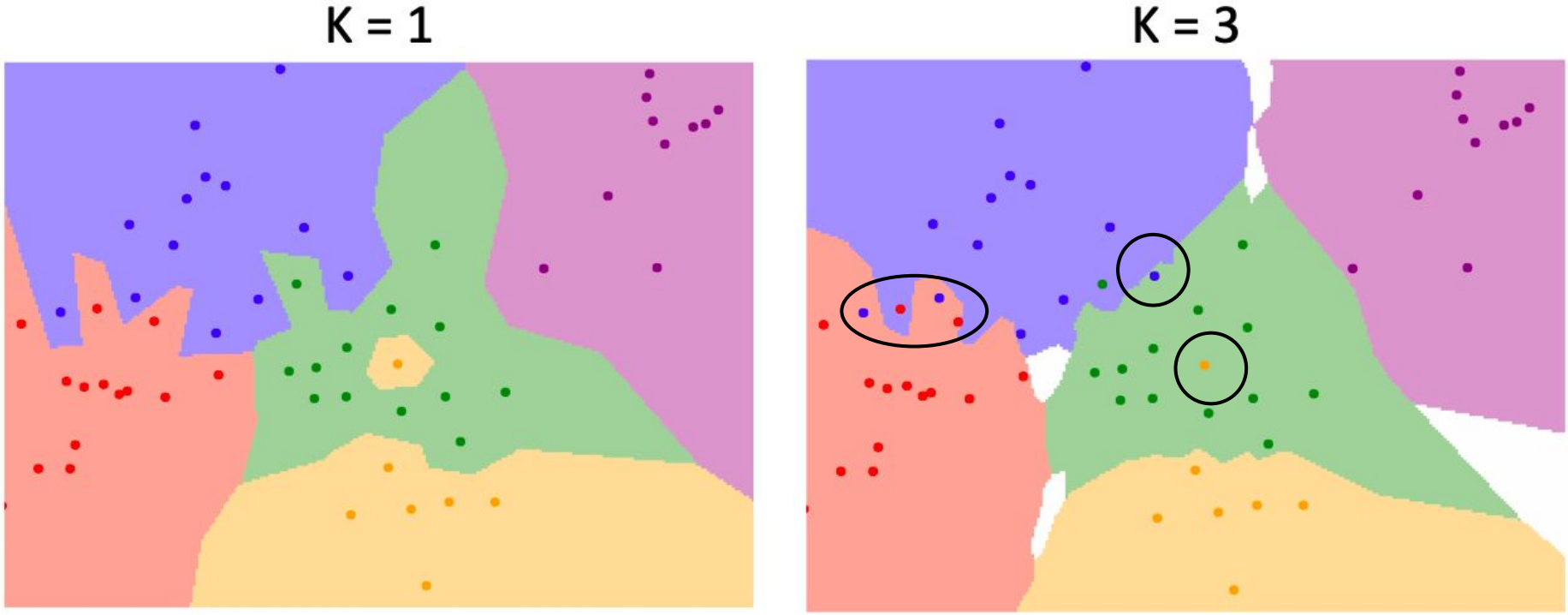
k-Nearest Neighbor

- Osnovna ideja: Uzeti k najsličnijih slika i "glasati" koju od njih uzeti kao klasu slike koju testiramo.



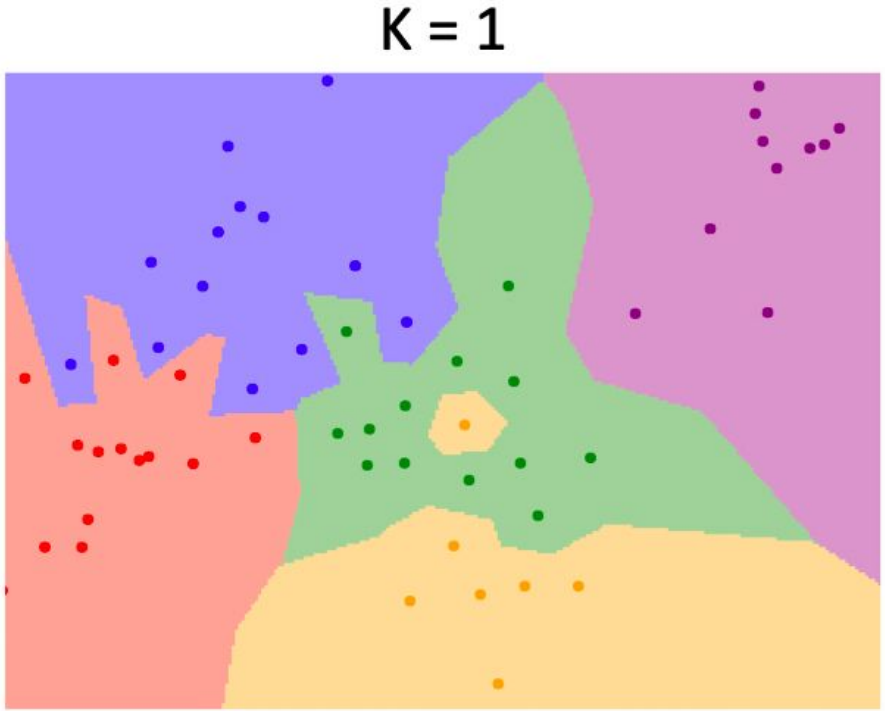
k-Nearest Neighbor

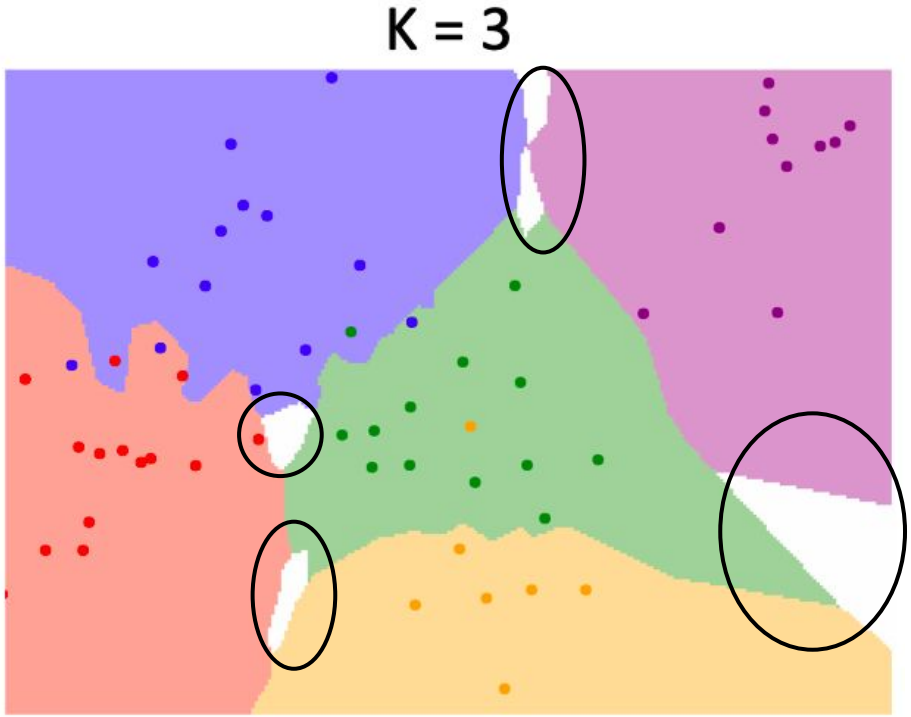
- Osnovna ideja: Uzeti k najsličnijih slika i "glasati" koju od njih uzeti kao klasu slike koju testiramo.



k-Nearest Neighbor

- Osnovna ideja: Uzeti k najsličnijih slika i "glasati" koju od njih uzeti kao klasu slike koju testiramo.





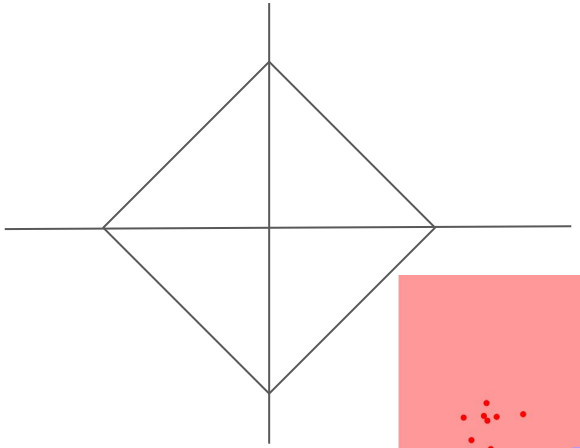
Metrika za distancu

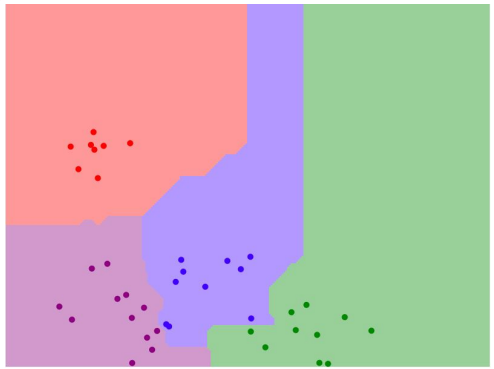
L1 ili *Manhattan* distanca:

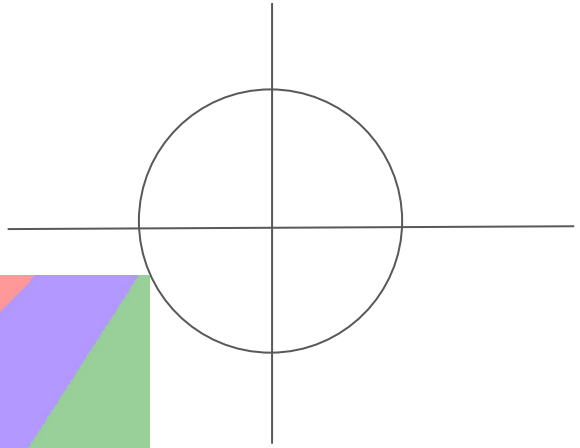
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

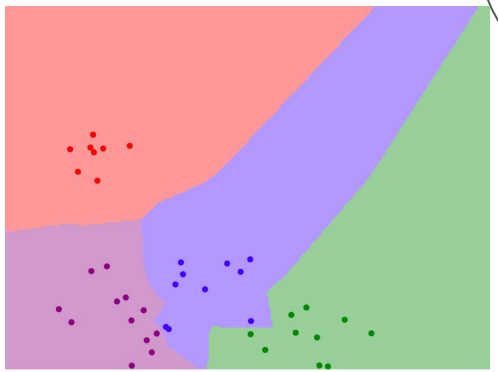
L2 ili *Euclidean* distanca:

$$d_2(I_1, I_2) = \left(\sum_p (I_1^p - I_2^p)^2\right)^{1/2}$$









Broj klasa: 4
Broj tačaka: 40
K = 1
Različite metrike

http://vision.stanford.edu/teaching/cs231n-demos/knn/

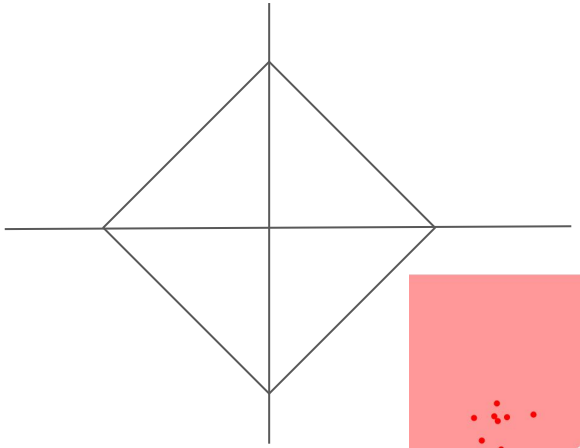
Metrika za distancu

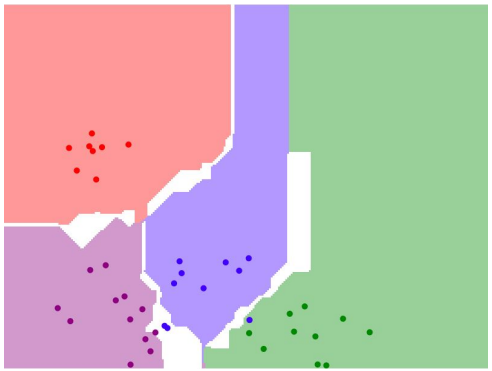
L1 ili *Manhattan* distanca:

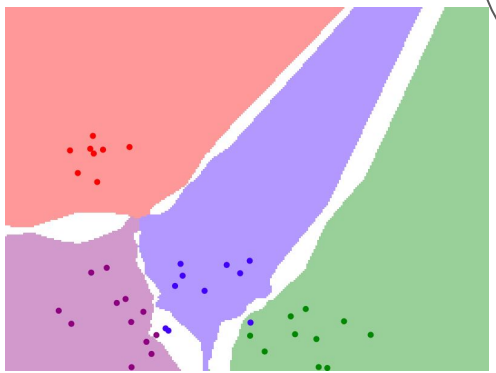
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

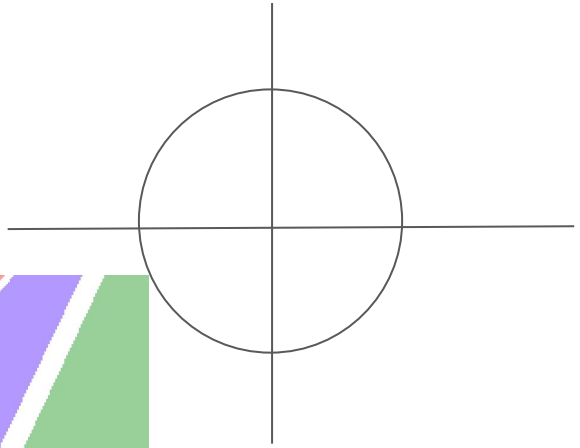
L2 ili *Euclidean* distanca:

$$d_2(I_1, I_2) = \left(\sum_p (I_1^p - I_2^p)^2\right)^{1/2}$$









Broj klasa: 4
Broj tačaka: 40
K = 4
Različite metrike