



## Sun'iy intellekt asoslari

### Mundarija

1. SI

2. Tarix va Rivojlanish

3. ML -> Mashinani o'qitish asoslari

4. DL -> Chuqur o'qitish asoslari

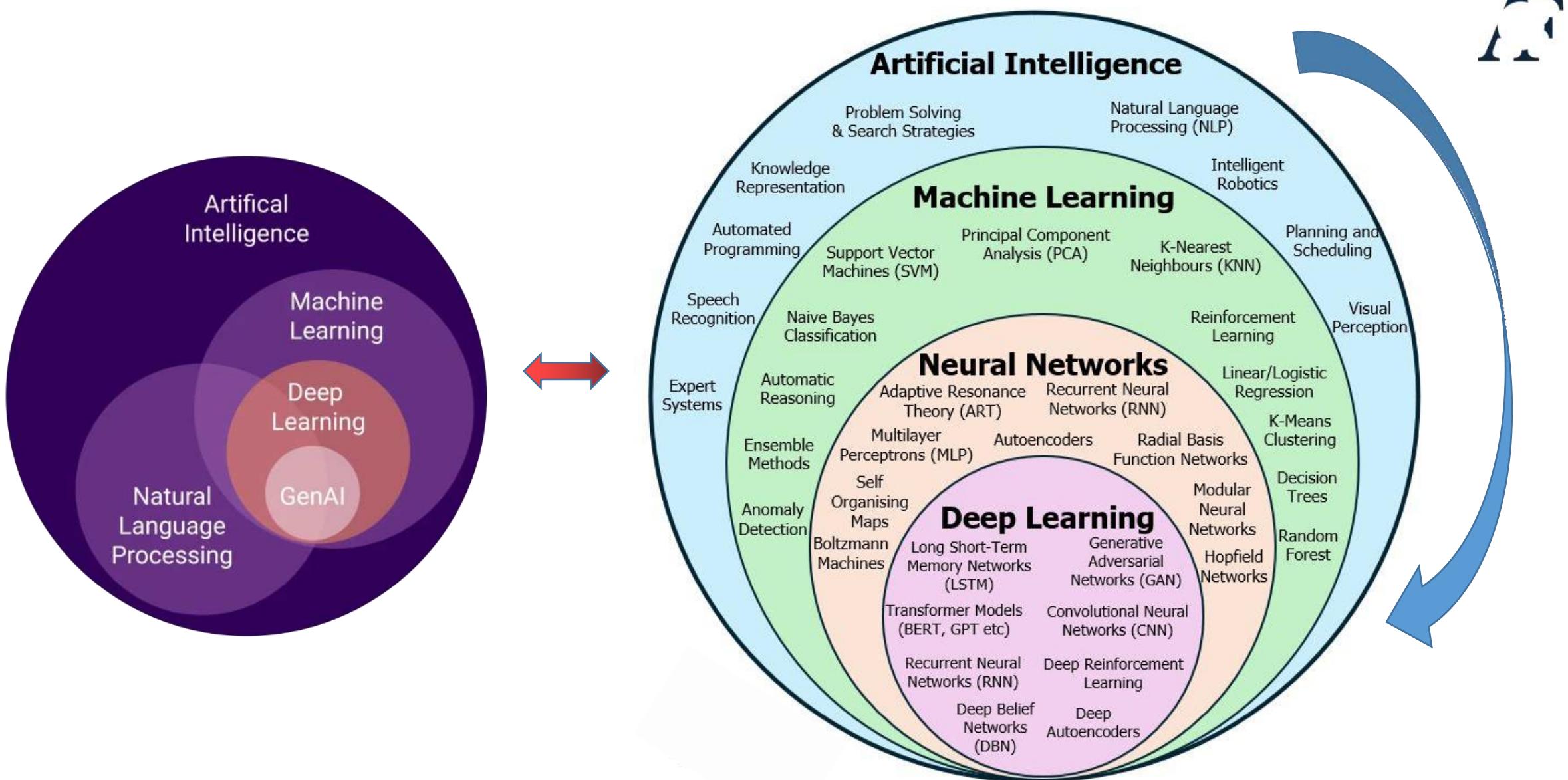


**Fanning asosiy mazmuni** - SI (AI) haqida bilimga ega bo'lish va amalda qo'llash. AI ni rivojlanishi, bugungi kundagi tadbiqi va kelajakdagi ehtimoliy ko'rinishlari ko'rib chiqish. AI nimalardan tashkil topadi, shularni o'rganish.

Bu fan (AI)ni biz asosan kompyuter ko'rishi (CV) yo'nalishida o'rganishga harakat qilamiz, tasvirli malumotlar bilan ishlashni o'rganamiz (Data processing), model quramiz (model training) va uni hayotga tadbiq qilish usullarini o'rganamiz (Ya'ni, sanoatda, ta'limda, meditsinada...).

Hozirgi kunda Til modellari (LLM) bizning hayotimizga chuqur kirib kelayotganligi sababli NLP va CV ni bir-biriga bog'lanish asoslarini ham ko'rib chiqamiz.

*Malumotlar sohaga oid inglizcha so'zlardan iborat*



**SI** - bu bir sistemaki insonlarga qaror qabul qilishda yordam beradi va insonlardek ilk ko'rgan malumotlarni mulohaza qilib, qaror qabul qila oladi.

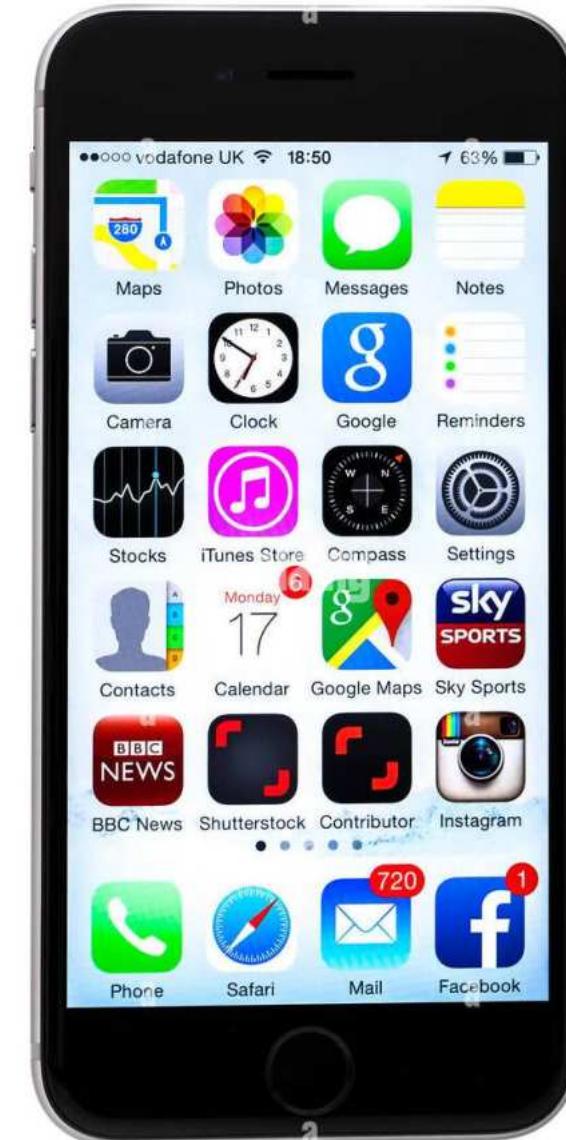
?



?



?



A



## Nima AI

- AI biz gaplashsak (og'zaki / yozma) tushinadi, va javob beradi (Alexa, Siri, GPT)
- Rasmlarni tushinadi, tahlil qila oladi. Rasmga qarab obyektlarni sinflarga ajrata oladi.
- Bashorat qila oladi: Malumotlarni tahlil qilish orqali kelajak bashoratlarini aytadi (Oh-Havo tahlili, YouTube dan bizga maqul mavzularning tavsiya qilishi)...
- ...

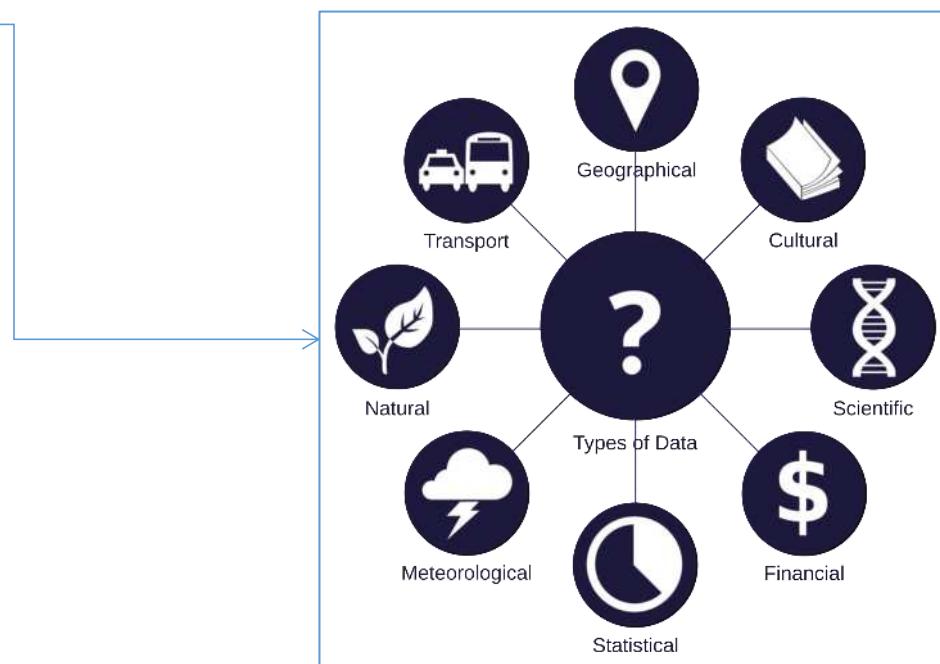
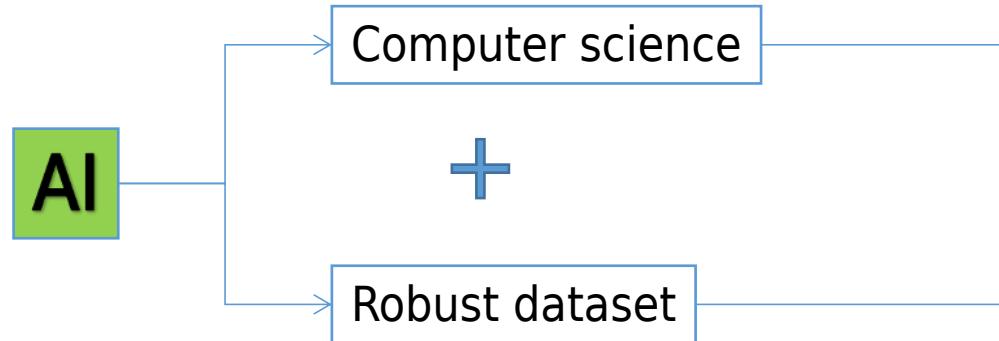
## va Nima AI emas

- An'anaviy qoidalarga asoslangan tizimlar,
- Oddiy avtomatlashtirish vositalari,
- Mexanik qurilmalar,
- Fixed funksiyali apparat,
- Interaktiv bo'limgan tizimlar,
- Oddiy sensorlar.

**Sun'iy intellekt** mashinalari ma'lumotlardan o'rganadi va ular bo'yicha qaror qabul qilishi mumkin. Misol uchun, aqli kir yuvish mashinasi nima yuvayotganiga qarab o'z sozlamalarini sozlashi mumkin. AI esa qoidalarga rioya qilishdan tashqariga chiqadi, u ma'lumotlar va kontekstga asoslangan holda o'rganadi, moslashadi va qaror qabul qiladi.



AI (SI) - bu kompyuterlarga turli xil ilg'or funksiyalarni bajarishga imkon beruvchi texnologiyalar, algoritmlar va funksiyalar majmuasi.



## History [edit]

See also: [History of artificial intelligence](#) § [Perceptrons](#)

The artificial neuron network was invented in 1943 by [Warren McCulloch](#) and [Walter Pitts](#) in *A logical calculus of the ideas immanent in nervous activity*.<sup>[5]</sup>

Miyani qanday ishlashi haqida matematik model taqdim etilgan

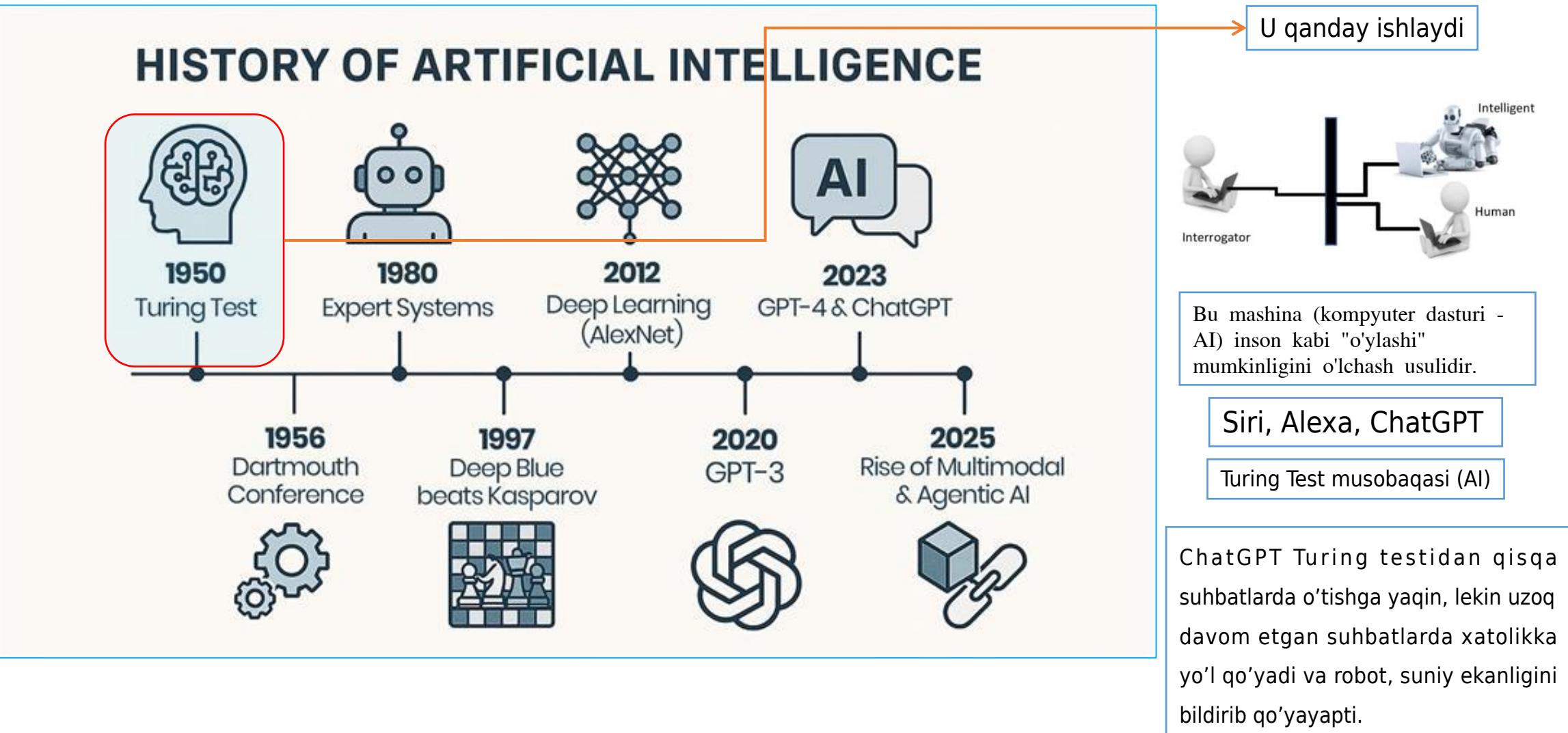
Miya ishlash funksiyasini tushunish uchun olimlar suniy biologik neyron modelini ishlab chiqishgan.

**biology + mathematics + logic**

**Muhimliliği:** Bog'langan neyronlar hamma narsani hisoblashi mumkin degan qarorga kelishgan

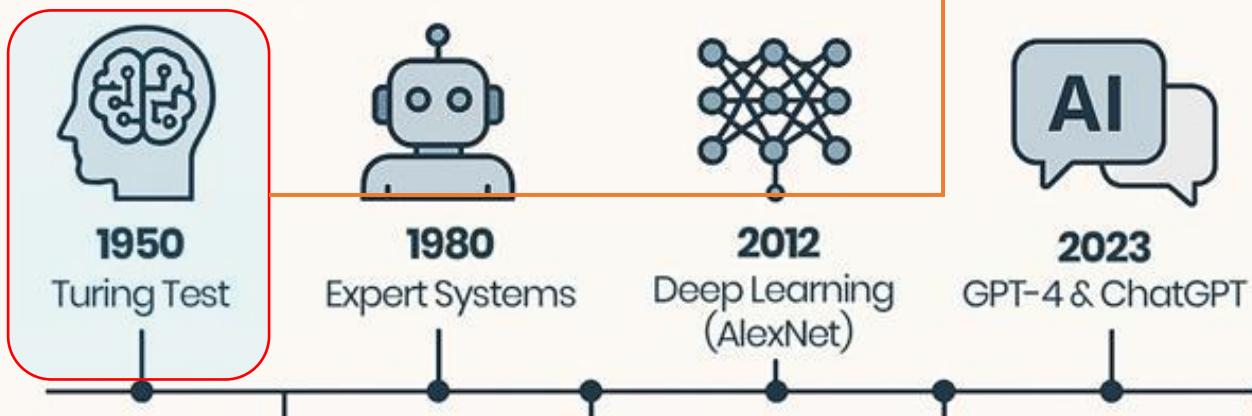


Demak AI - turli xil funksiyalarni bajarishga imkon beruvchi texnologiyalar, algoritmlar va funksiyalar majmuasi ekan. Bu algorithm, funksiya va texnologiyalar birdan rivojlanib qolmagan. Bu bosqich AI revojlanish tarixiga kiradi.





## HISTORY OF ARTIFICIAL INTELLIGENCE



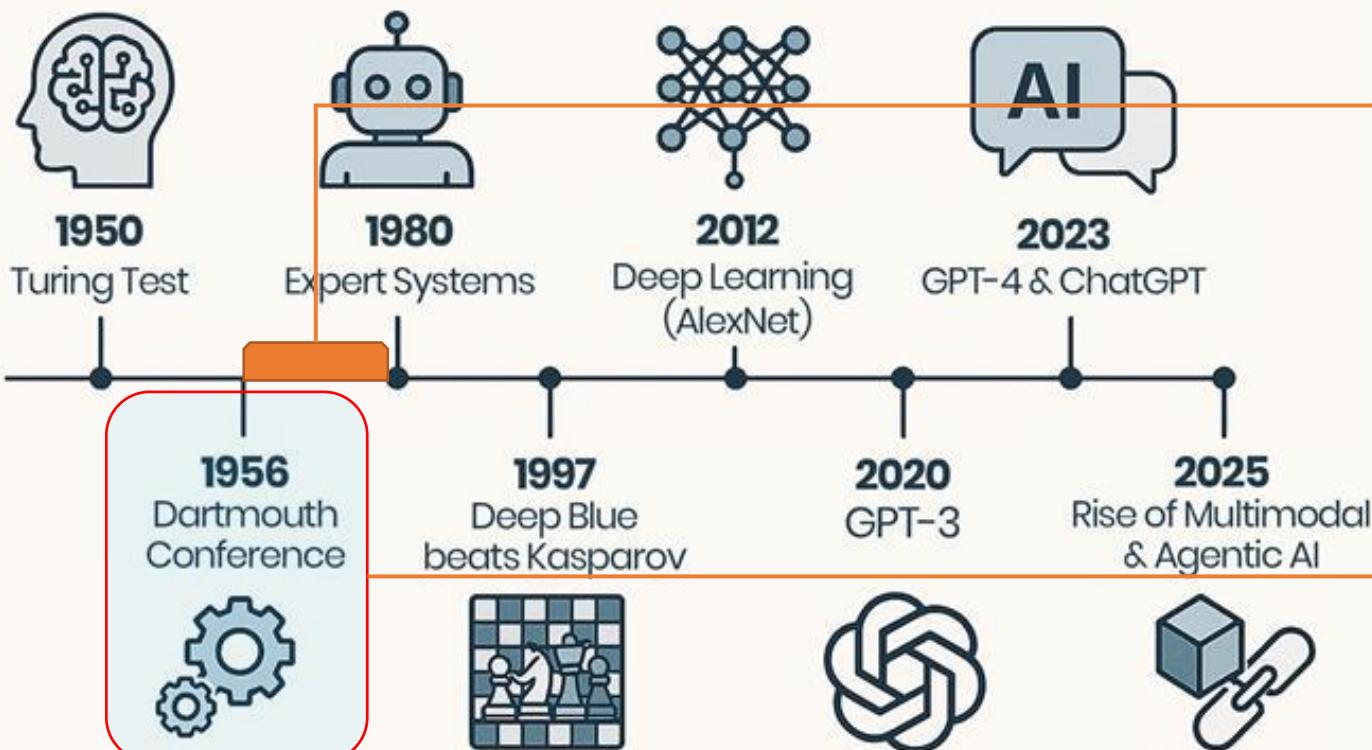
Turing Testining kamchiliklari



- Faqat Til suhbatiga etibor beradi,
- Turing testidan o'tish degani inson darajasidek his-tuyg'uli, ongli degani emas.



## HISTORY OF ARTIFICIAL INTELLIGENCE



### 1960 -1970 yillarda AI:

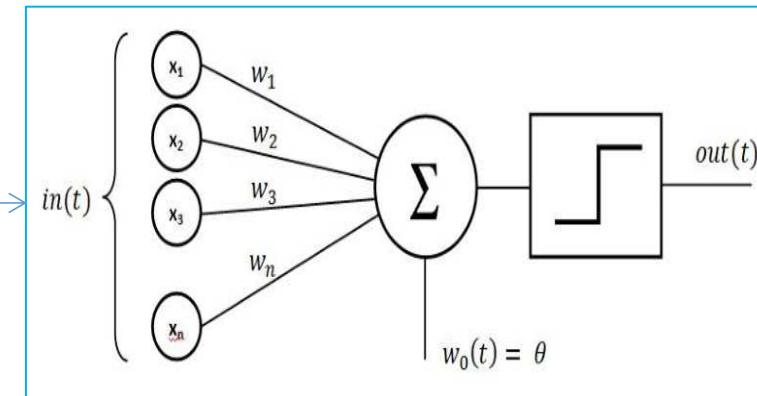
Bu yillar Ekspert tizimlarining rivojlanishiga olib kelgan. AI tadqiqotlarda sezilarli yutuqlar bo'lgan, dastlabki neyron tarmoqlari, ramziy fikrlashni o'rganish va muammolarni hal qilish texnikalar revojlangan.

### Dartmut konferentsiyasi

Makkanzi tomonidan "AI" - Sun'iy Intellekt iborasi ilk bor ishlatildi. Shu kanfirensiyada AI ga Makkanzi, Turing, Misky va Simon lar birga asos solgan.

## 1960 -1970 yillarda AI:

Perceptron (1958, Frank Rosenblatt)



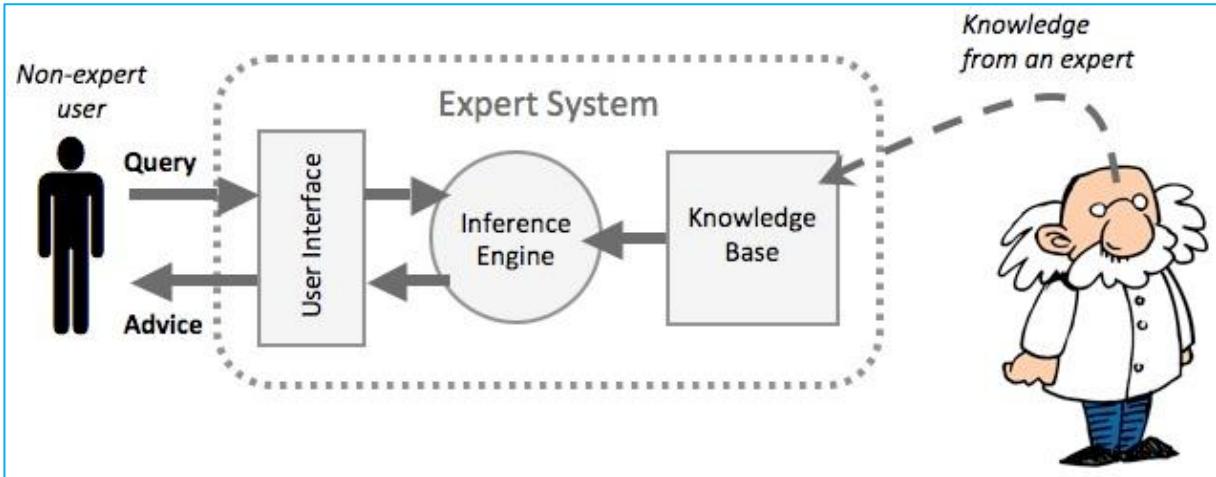
Ko'chaga aylanishga chiqish uchun qaror qabul qilish

Input (savol)	Qarorni baholash	Misol	Qiymat	Ma'nosi
x1	Havo quyoshlimi?	Ha (1)	+2	Quyoshli havoni yaxshi ko'rasiz
x2	Havo sovuqmi?	Ha (1)	-3	Sovuqni yomon ko'rasiz
x3	Vaqtim bemalolmi?	Ha (1)	+4	Bo'sh vaqtida aylanishni yaxshi ko'rasiz

Natija:  $(1*2)+(1*-3)+(1*4)= 2-3+4=3$

Qaror: Bias -1 deb hisoblasak:  $3-1=\mathbf{2}$   
**- Ko'chada aylanaman (2)**

**1960-yillarda** tadqiqotchilar miya funksiyalaridan ilhomlangan holda sun'iy neyronlarni o'rganishdi.



If [X] And [Y], Then ____		
Condition 1	Condition 2	True or False?
True	True	True
True	False	False
False	True	False
False	False	False

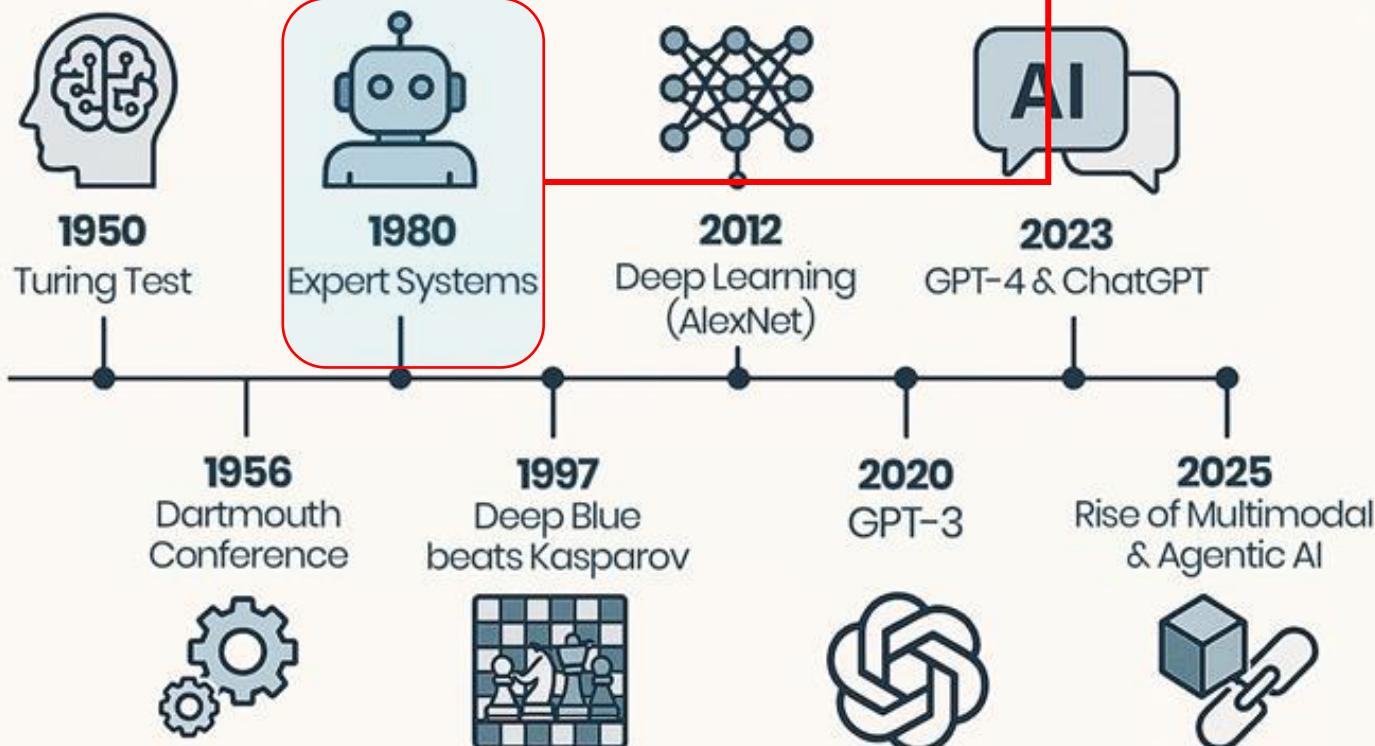
**DENDRAL (1965):** Kimyogarlarga molekulyar tuzilmalarni aniqlashga yordam berish uchun Stenfordda ishlab chiqilgan **ekspert tizimi**.

**MYCIN (1970s):** Bakteriyalarga tashxis qo'yish va antibiotiklar tavsiya etish





## HISTORY OF ARTIFICIAL INTELLIGENCE



If [X] And [Y], Then \_\_\_\_\_

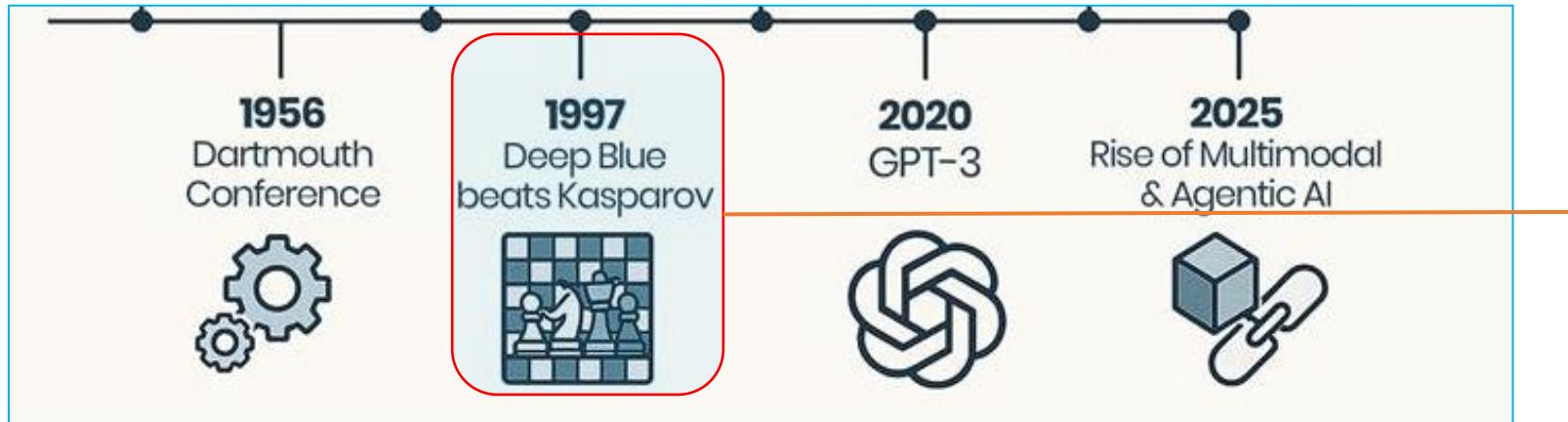
Condition 1	Condition 2	True or False?
True	True	True
True	False	False
False	True	False
False	False	False

**If - then** qoidasida ishlash boshlanishi 1970- yillarda bo'ldi, 1980-yillarda bu qoida asosida bo'lgan ishlar ancha rivojlandi. Bu qoida kelajak AI -si deb hisoblana boshlandi toki **ANN** (Suniy neural tarmoqlari) rivojlanganiga qadar.



## 1980 -1990 yillarda AI:

Mashinani o'rganishdagi yutuqlar bilan **AI** haqida aralash optimizm, skeptitsizm va neyron tarmoqlari **AI** sohasida katta o'zgarishlarga va yutuqlar olib kelmadi.



Deep Blue **IBM** tomonidan ishlab chiqilgan shaxmat o'ynaydigan superkompyuter **1997** yilda G.Kasparovni shaxmatda yutdi.

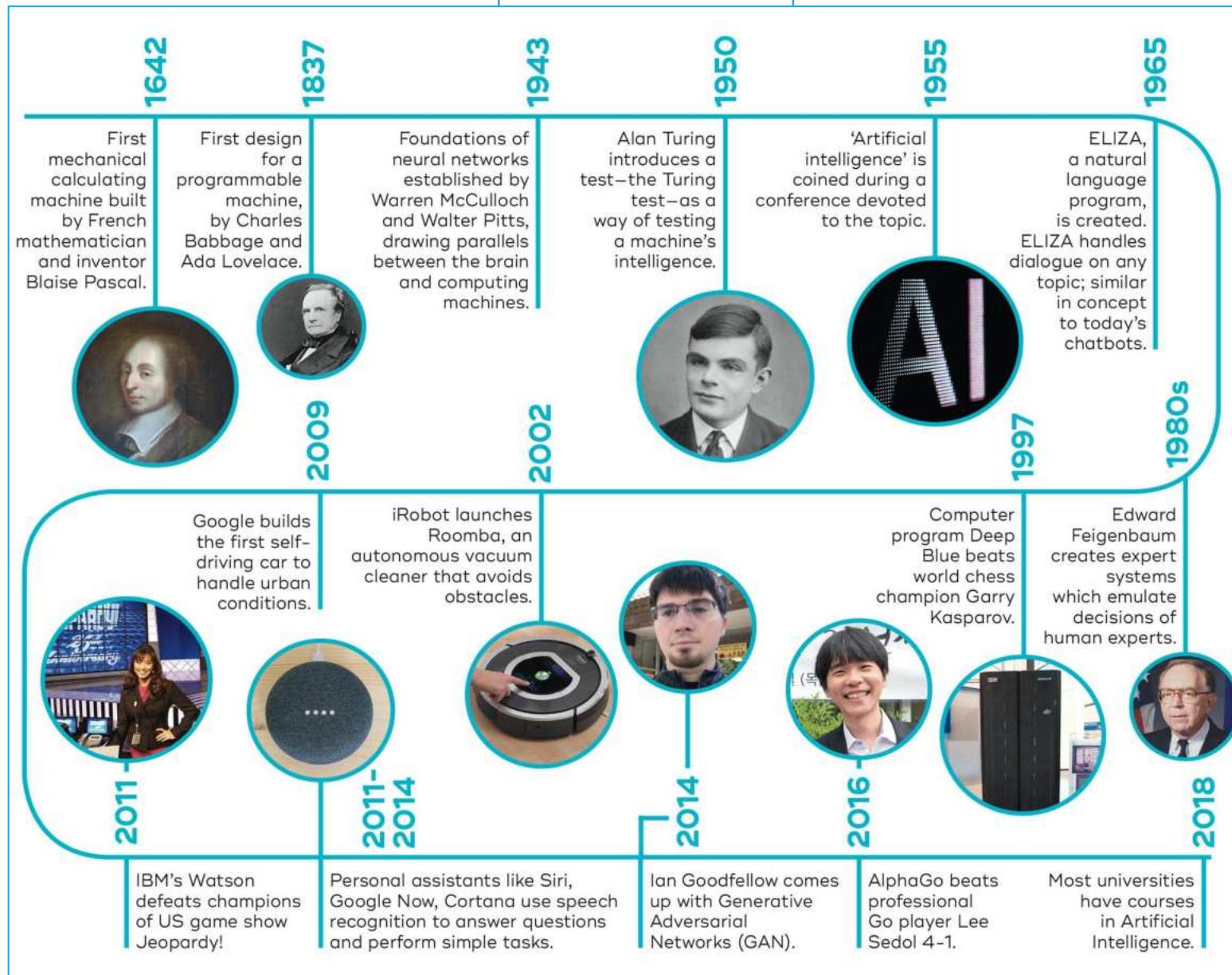
Birinchi marta kompyuter jahon championini mag'lub etdi.

**Game Over: Kasparov and the Machine**

**GAME OVER:  
KASPAROV AND THE MACHINE**



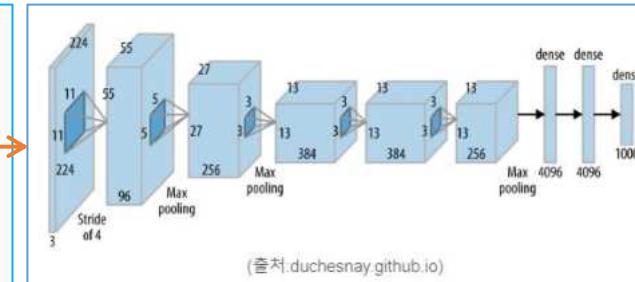
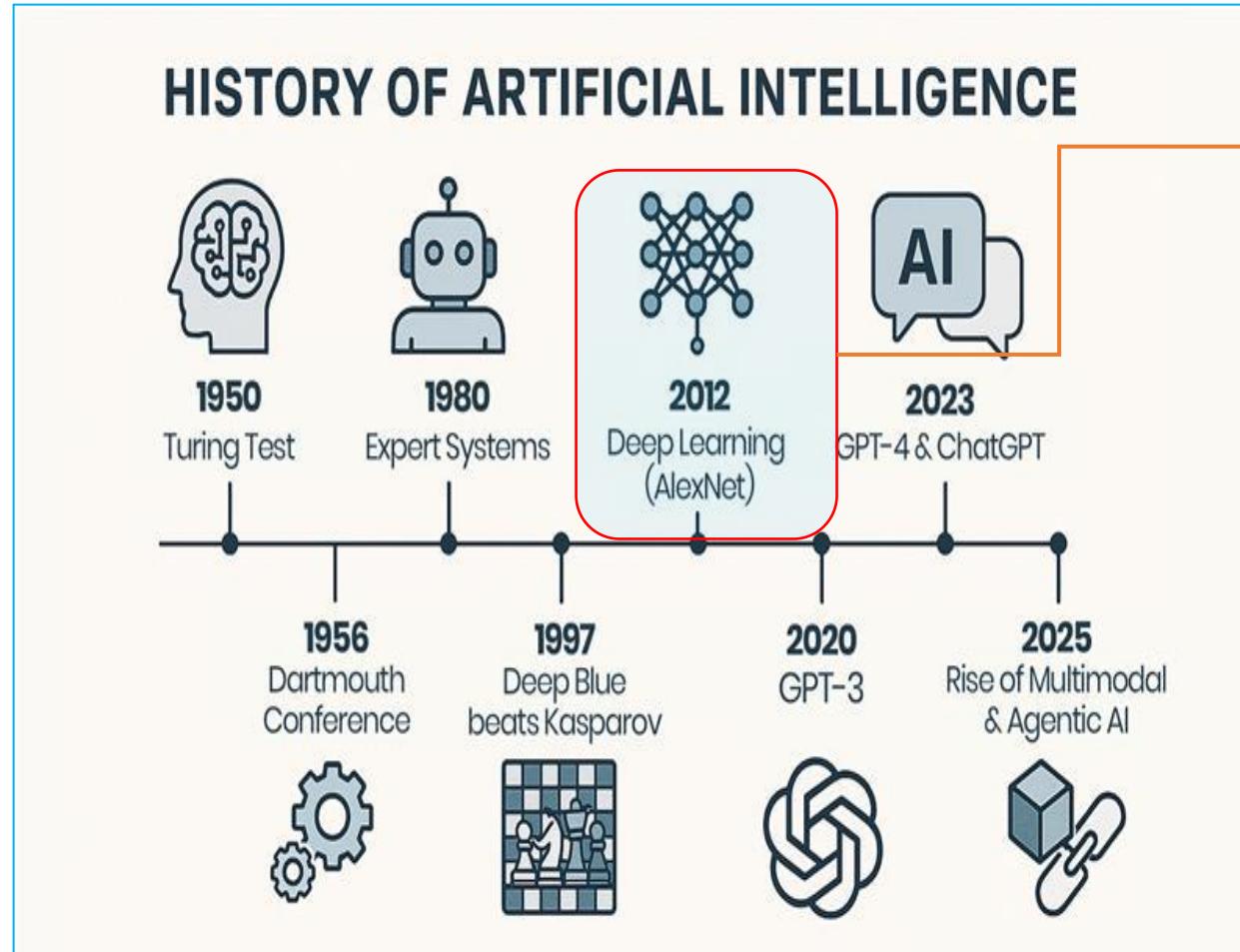
## 21- Arsda AI



A



- Neocognitron (1980) - turtki.
- LeNet-5 (1998) **CNN**
- 2000** - CNNs (AI rad etilgan vaqtlar)
- AlexNet (2012) - revolyutsiya. **GPUs + Big Data.**  
Zamonaviy chuqur o'qitish (DL)  
CV yo'nalishida rivojlantirdi.



**AlexNet** - bu CNN arxitekturasidir. U rasmni sinflarga ajratuvchi arxitekturadir.

**AlexNet** dan oldin CNN ga bo'lgan etibor past bo'lgan. Chunki GPU lar ommalashmagan, CPU lar juda sekin hisob kitob qilgan. Lekin, **AlexNet** katta rasmli malumot bilan (1.2 mln yuqori pixelli rasmlar) model qurban. Haftalab, oylab vaqt oladigan jarayonni bir necha kunlarga o'zining maxsus arxitekturasi bilan tushirgan.

**CV - vision transformers**

2012 - AlexNet,  
2014 - VGGNet,  
2014 - GoogLeNet / Inception (Google),  
2014 - R-CNN,  
2015 - Fast R-CNN,  
2015 - ResNet (Microsoft),  
2016 - DenseNet,  
2016 - YOLO,  
2017 - RetinaNet (Facebook AI)  
2017 - YOLOv2  
2017 - Mask R-CNN (Facebook AI)  
2017 - SENet,  
2018 - Detectron (Facebook AI research),  
2018 - YOLOv3  
2019 - Detectron2,  
2019 - EfficientDet (Google brain)  
2020 - Vision Transformer (ViT, Google),  
2020 - YOLOv4,  
2021 - YOLOv5 (Ultralytics),  
2021 - Swin Transformer,  
...

**Speech & Audio**

2014 - DeepSpeech (Baidu),  
2016 - WaveNet (DeepMind),  
2018 - Tacotron 2 (Google),  
2022 - Wisper (OpenAI)  
...

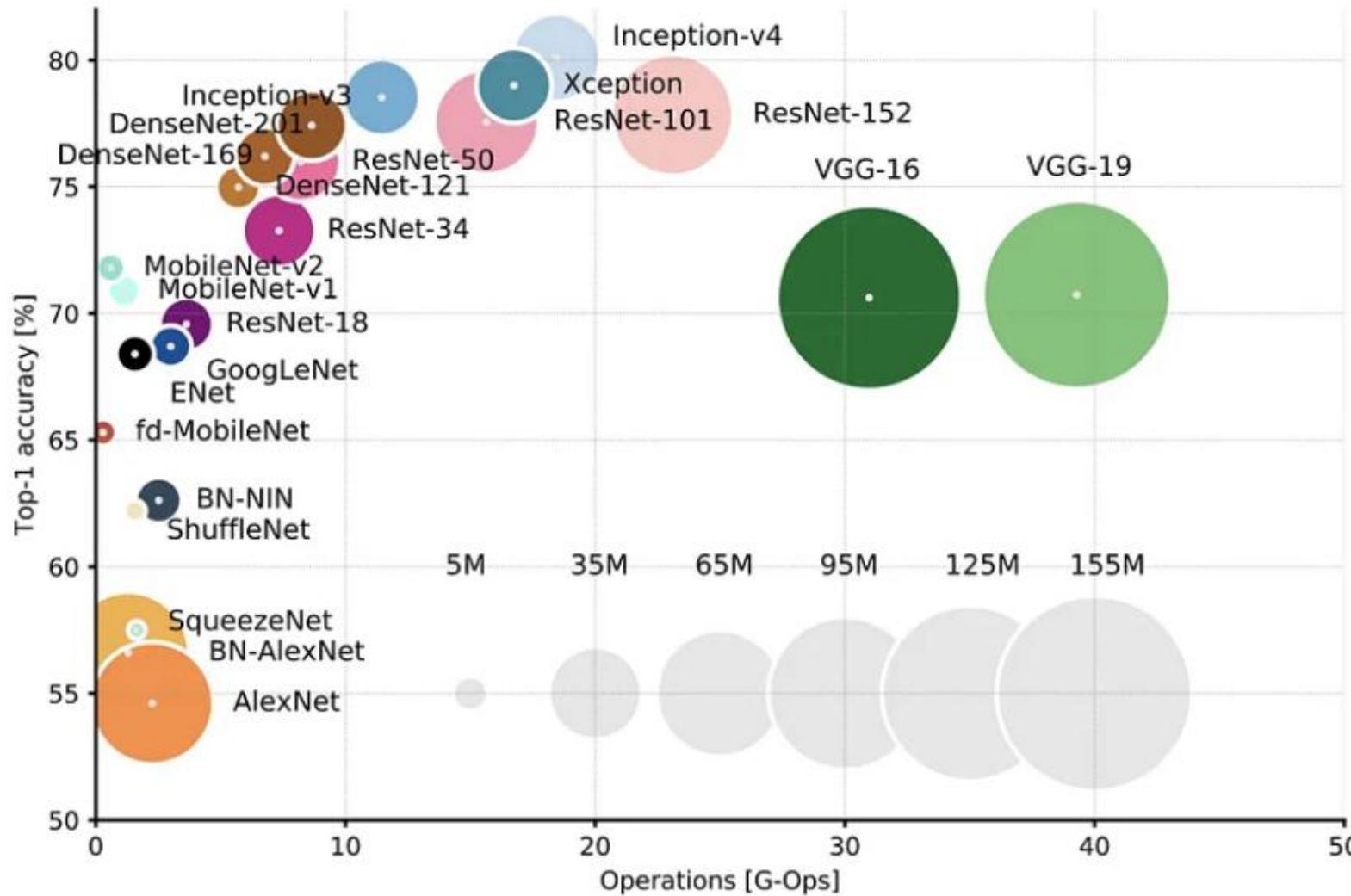
**NLP**

2013 - Word2Vec (Google),  
2014 - GloVe (Stanford),  
2015 - Seq2Seq (Google),  
2015 - Attention Mechanism,  
2017 - Transformer (Google),  
2018 - BERT (Google),  
2019 - GPT-2(OpenAI),  
2019 - XLNet (Google),  
2020 - T5 (Google),  
2020 - GPT-3 (OpenAI),  
2022 - BLOOM (BigScience),  
2023 - GPT-4 (OpenAI),  
2023 - LLaMA (Meta),  
2023 - Claude (Anthropic),  
2024 - Mistral, Mixtral,  
2024 - GPT-4o (OpenAI)  
...

**Multimodal / Generativ AI**

2014 - GANs (Goodfellow),  
2015 - DCGAN,  
2017 - CycleGAN,  
2018 - BigGAN (DeepMind),  
2021 - DALL-E (OpenAI),  
2021 - CLIP (OpenAI),  
2022 - Stable Diffusion,  
2022 - Imagen (Google),  
2022 - MidJourney,  
2023 - Sora (OpenAI)  
...

<https://alexlenail.me/NN-SVG/>

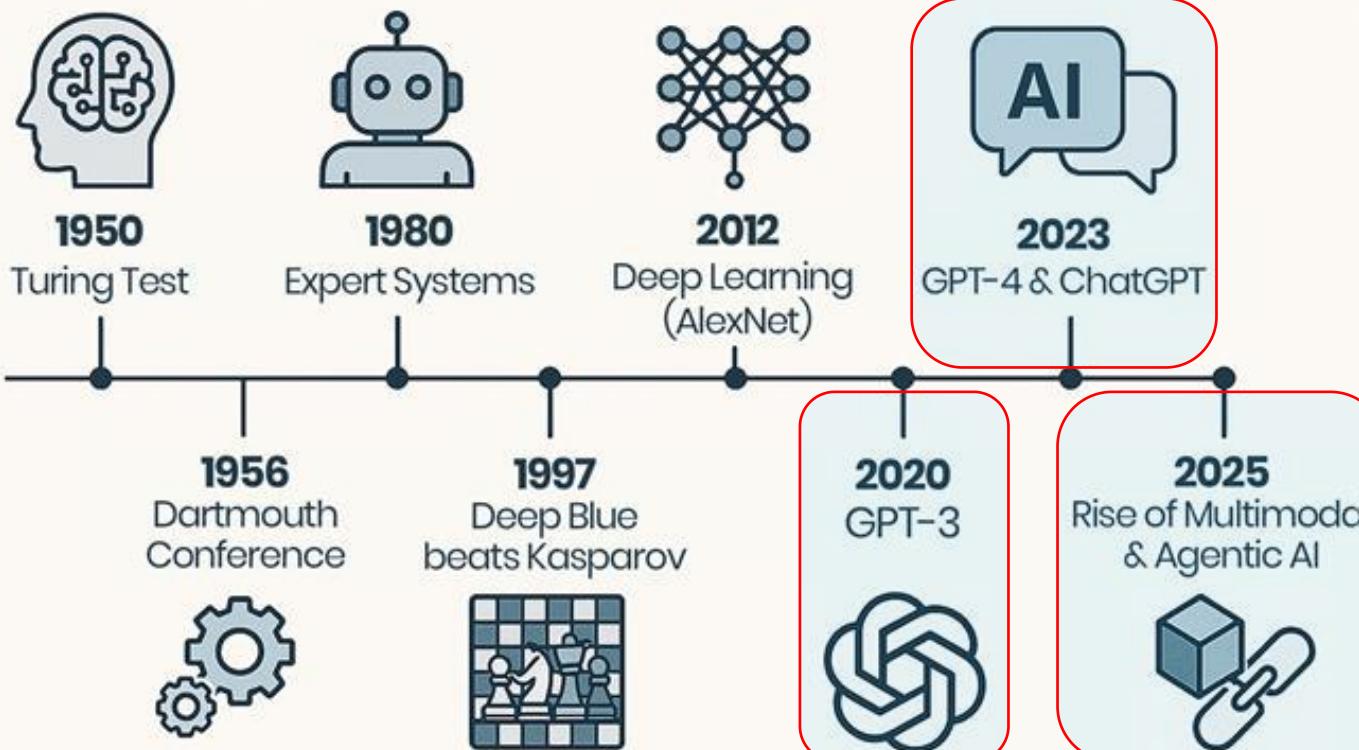


<https://towardsdatascience.com/>

<https://playground.tensorflow.org/>



## HISTORY OF ARTIFICIAL INTELLIGENCE



**2020-2025:**

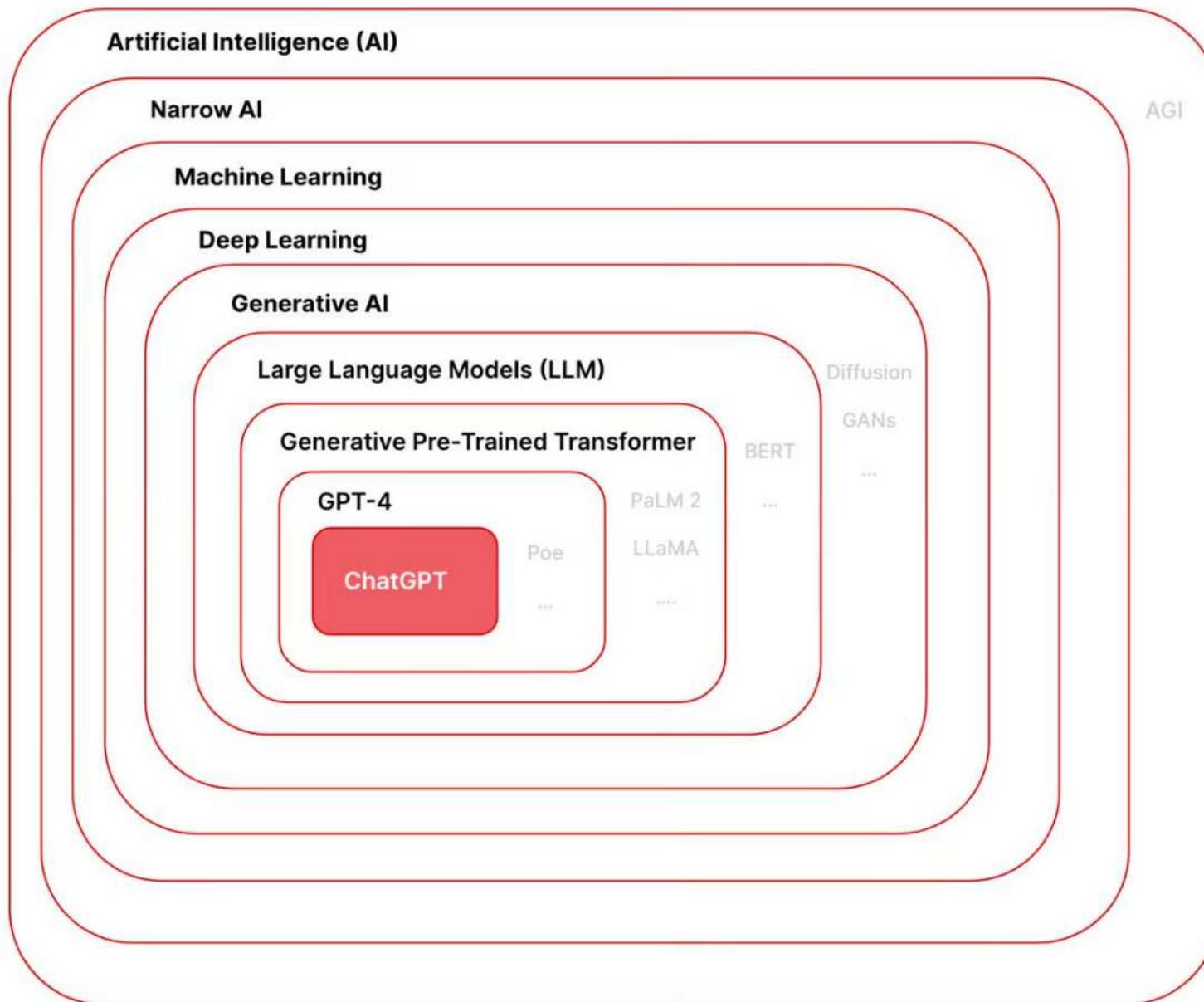
Kichik parametrlardan katta parametrlarga o'sib bordi. 2020 da faqat so'zni bilan ishlagan bo'lsa, 2023 da so'z va rasmni tahlil qilish ham qo'shildi. **GPT-3, GPT-4, Claude, Gimini, LLaMA, Grok** kabi modellarda optimallashdi.

Bu modellar millionlagan, trillionlagan so'zlarni analiz qilish(model training)i orqali optimallashyapti.

**Agentic AI** lar :

- API lar bilan bog'lanib, ishlata oladi,
- Ketma-ket barariladigan ishlarni planlashtira oladi,
- Ayni damdagiga vazifalarni qila oladi, bilet olish, sotish, bozor narxlari analizi...





Hamma **LLM** lar **Generative AI** ga kiradi, lekin hamma **Generative AI** modellari ham **LLM** ga kirmaydi

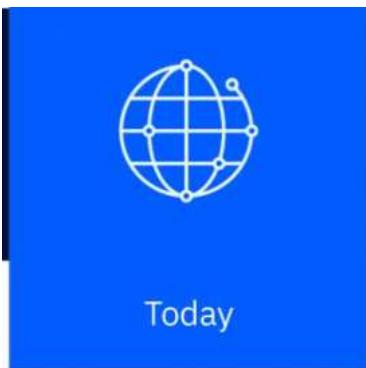


Narrow AI



2010-2015

- Face detection
- Object recognition
- speech recognition
- Recommendation systems



Today

General AI



2050 and beyond

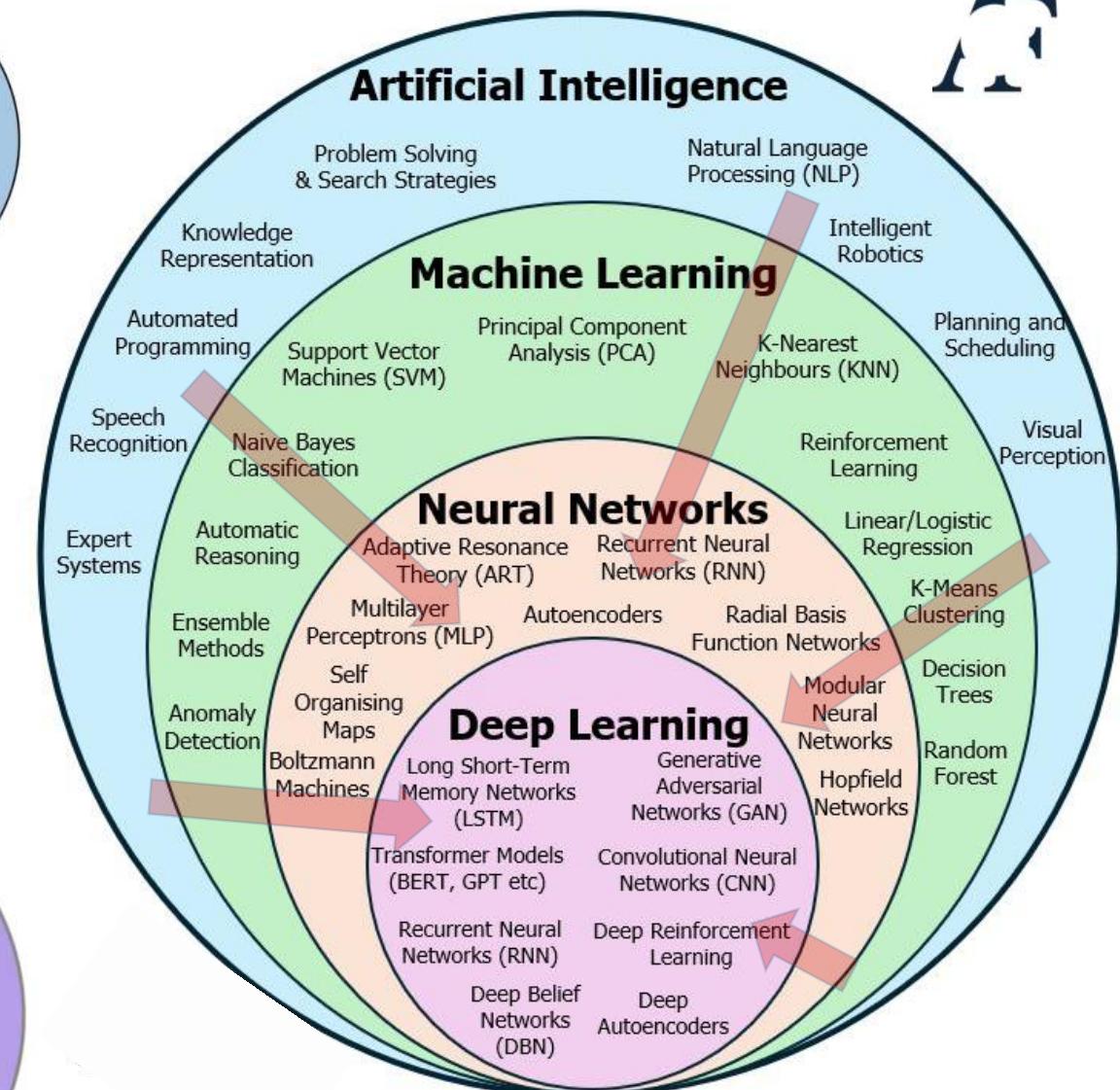
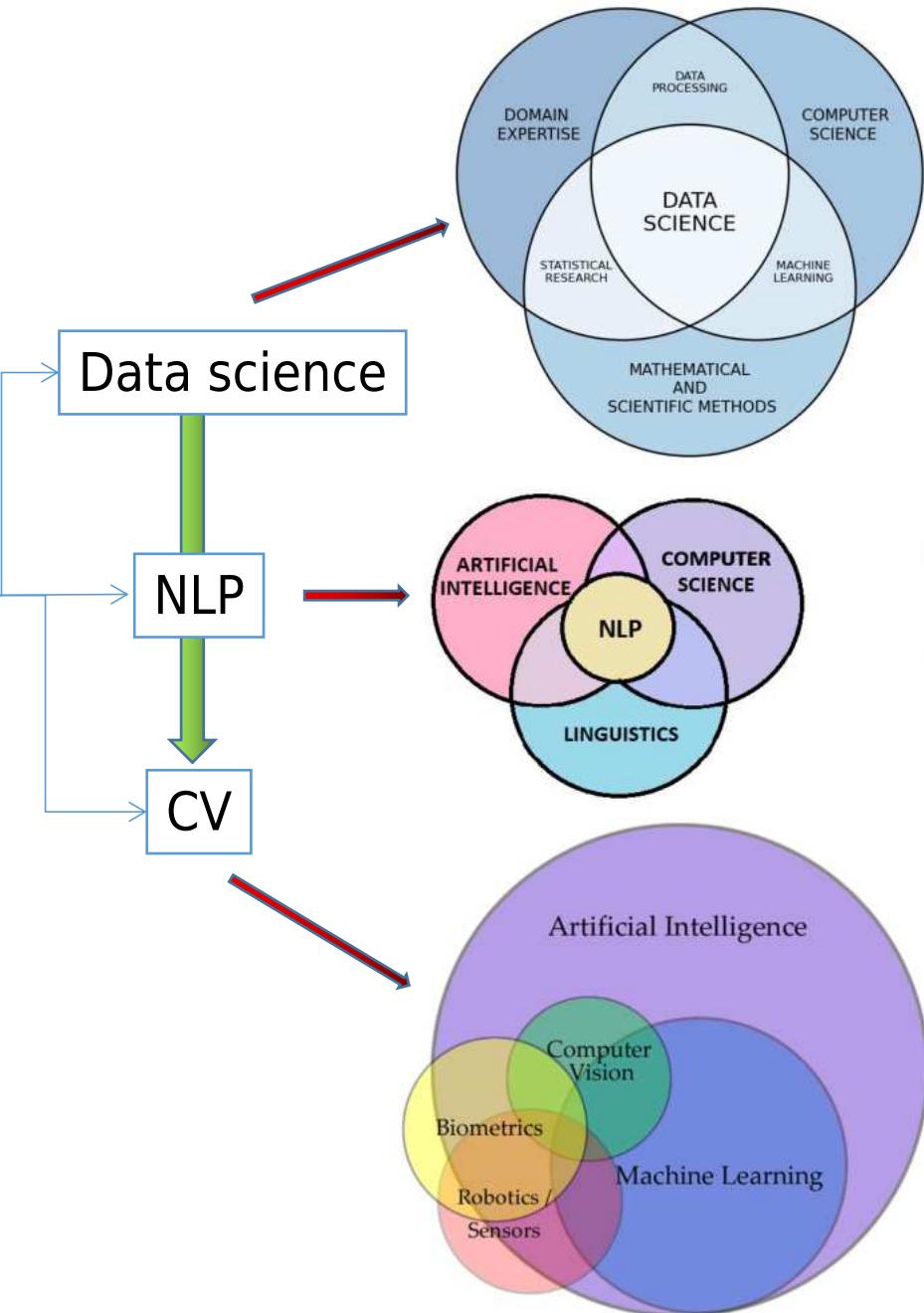


Kuchsiz AI (Narrow AI) ga  
nisbatan kuchli, ko'plab  
vazifalarni bajara oladi

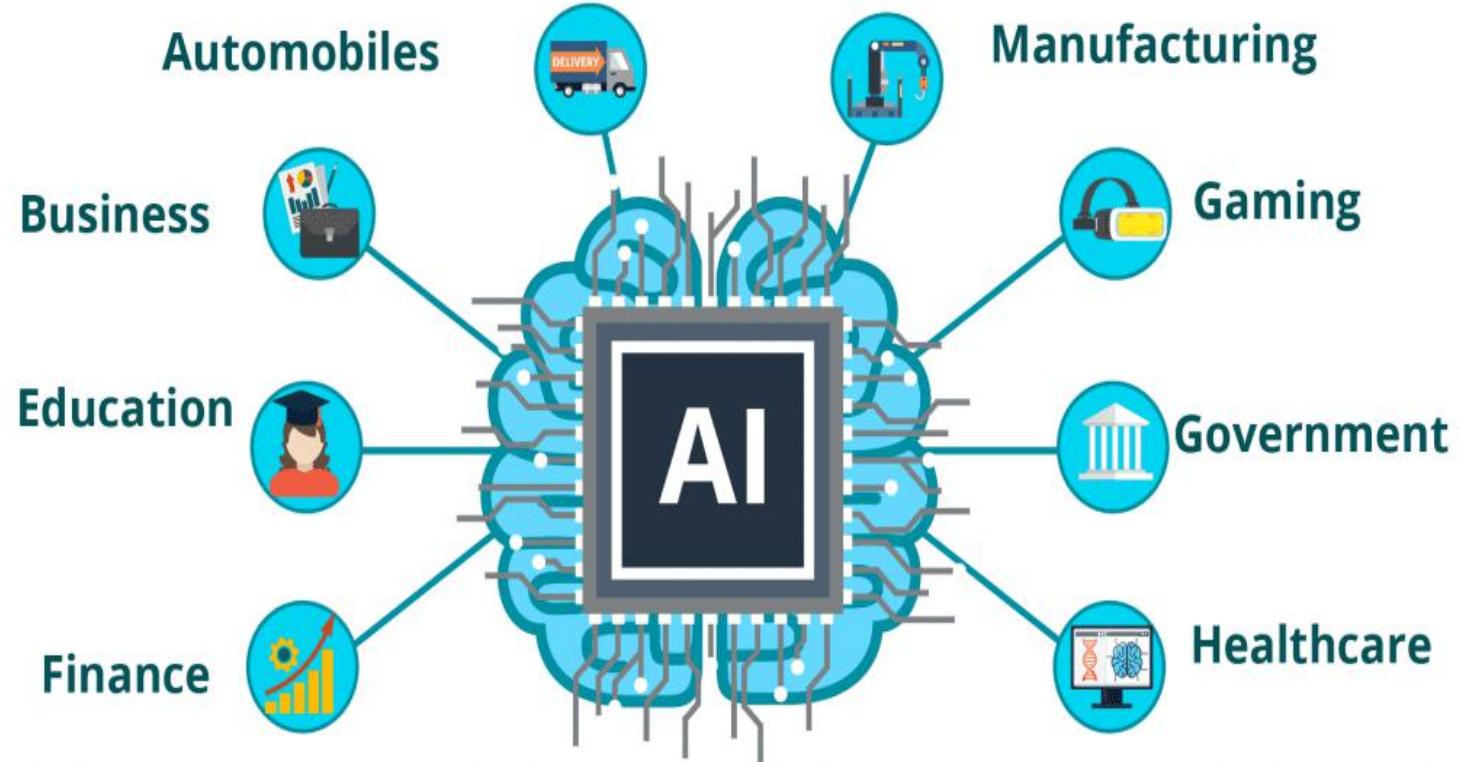
Kuchli AI davri. Insondek  
fikrlash...

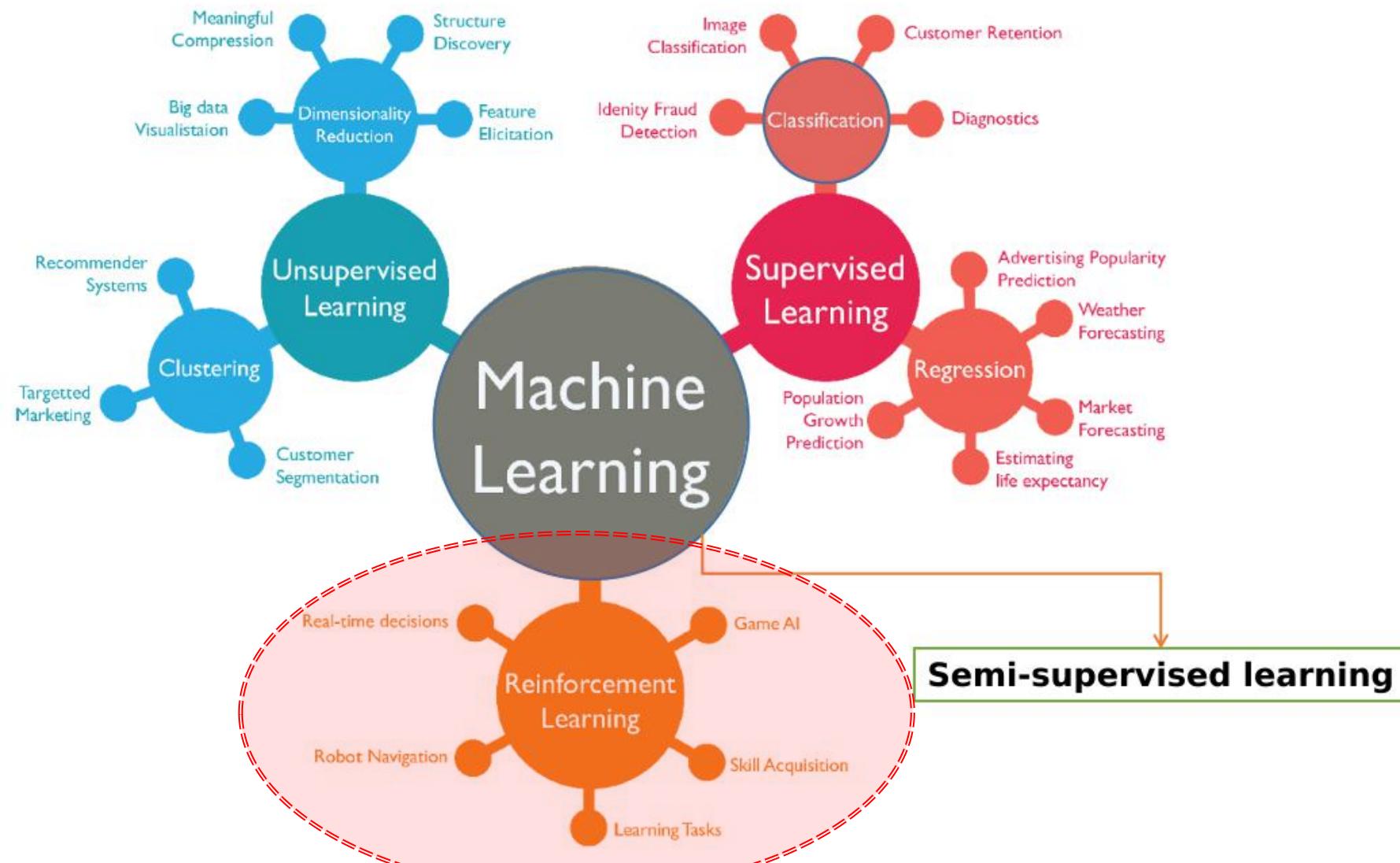


## AI - ni asosiy yo'nalishlari



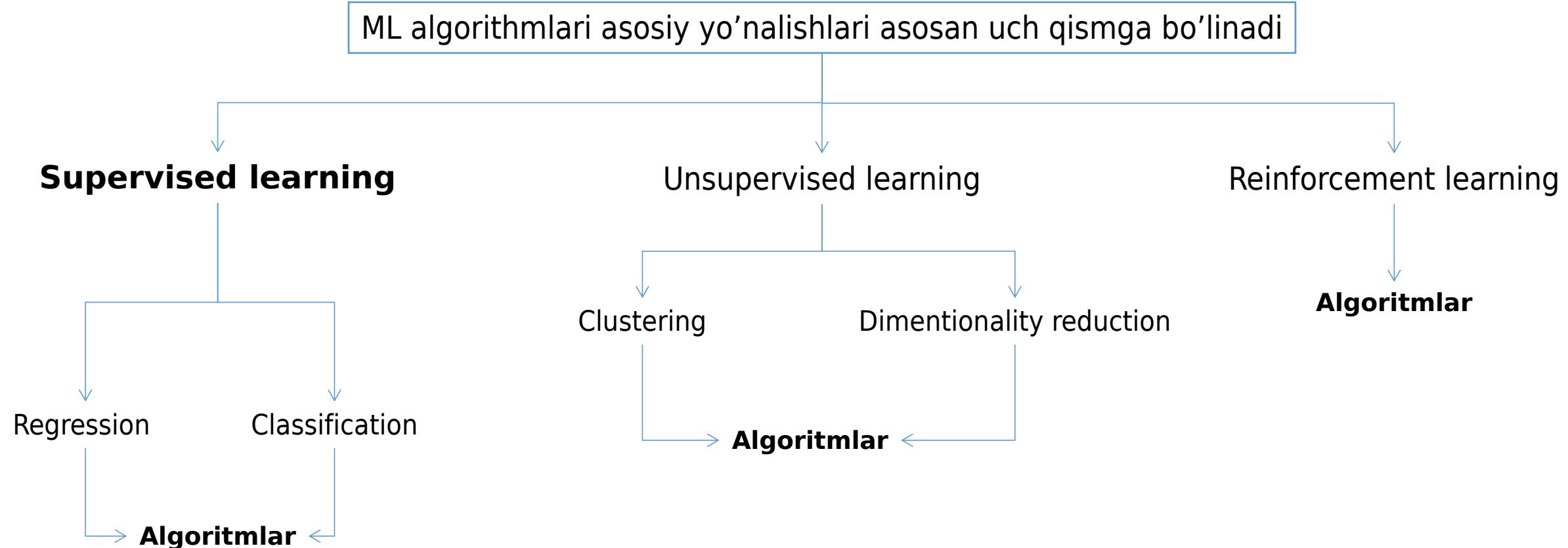








**ML o'zi nima:** Bunda compyuterlarga malumotlardan o'ziga ko'nikma hosil qilishni o'rgatish sohasi.





## Supervised Learning (Yo'naltirilgan o'qitish)

Bu yo'nalishda model qurish uchun model aniq malumotni ko'rsatish orqali o'rgatiladi. Aniq malumotlar taqdim etilgandan keyin, model shu malumotni tarorlayverish orqali o'zida ko'nikma hosil qiladi.

Misol uchun insonlarda ham shunday o'rganish turi mavjud.

Masalan, biz yangi shaharda ko'chada aylanib ketayapmiz.

Birinchi marta bu ko'chadan yurishimiz va hamma narsa bizga yangi malumot bo'lib qabul bo'lyapti. Va bizda bu ko'chada nimalar borligi haqida so'rashsa, taxminan 30% narsalar joylashuvini to'g'ri aytib berishimiz mumkin. Agar biz bu ko'chadan 2 yil mobaynida har kuni yursak va huddi 1-kundagidek kuzatib o'tsak, bizda ko'chaga oid malumotlar xotiramizda aniqroq joylashadi, ko'chaga oid yaxshi xotiraga ega bo'lamiz. Yani, malumotlarni avval o'rganamiz va keyin ko'cha haqida so'ralsa adashmay javob beramiz.





## Supervised Learning (Yo'naltirilgan o'qitish)

K-Nearest Neighbors  
(K-NN)

Random  
Forest

Decision  
Trees

Support Vector  
Machines (SVM)

Linear  
Regression

Naive bayes

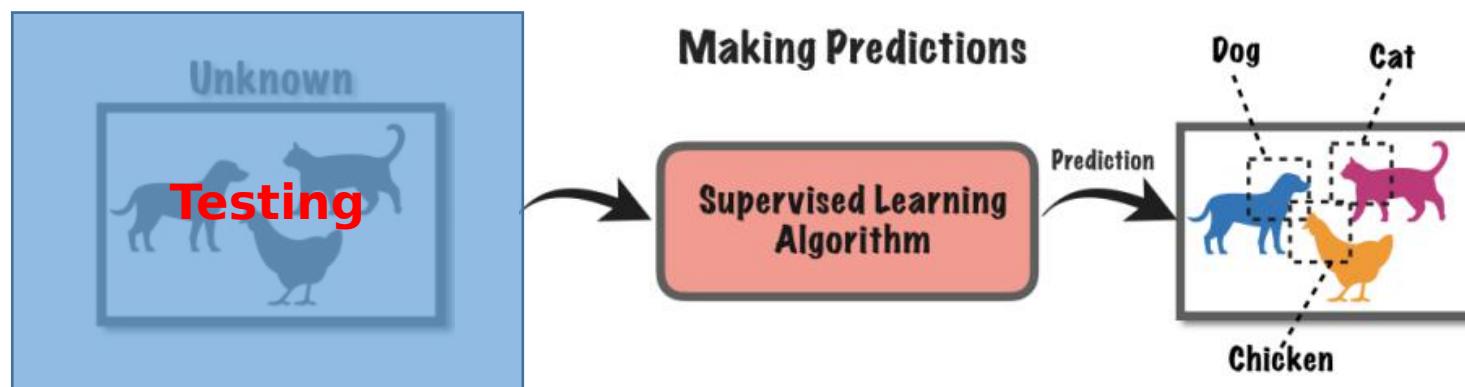
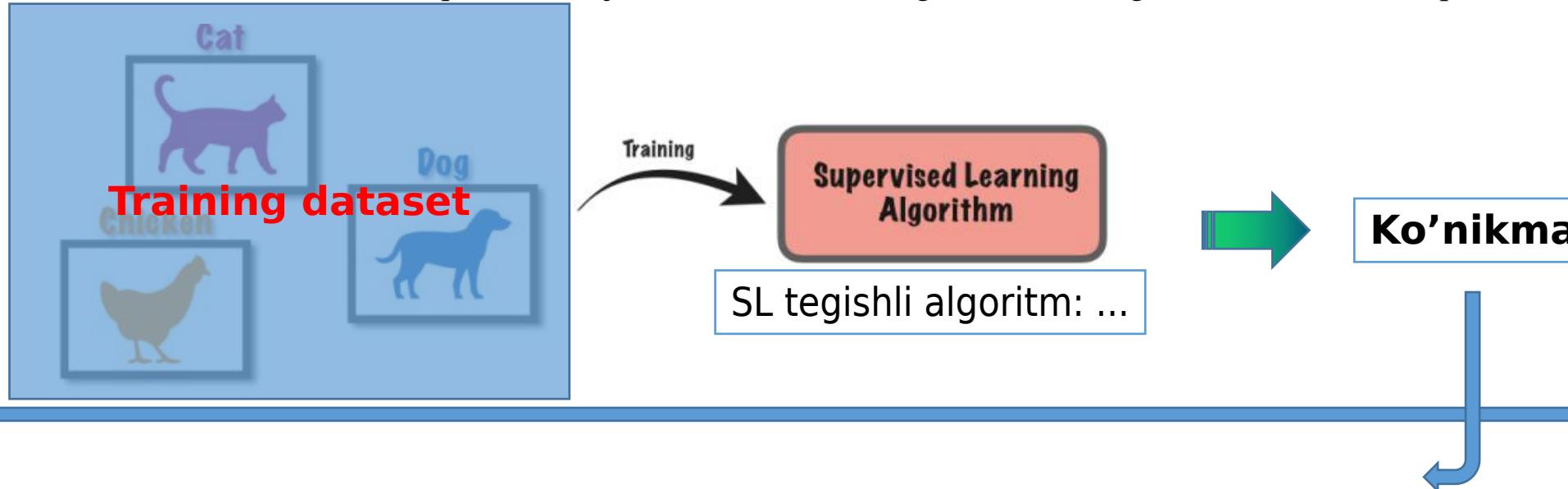
Neural Networks

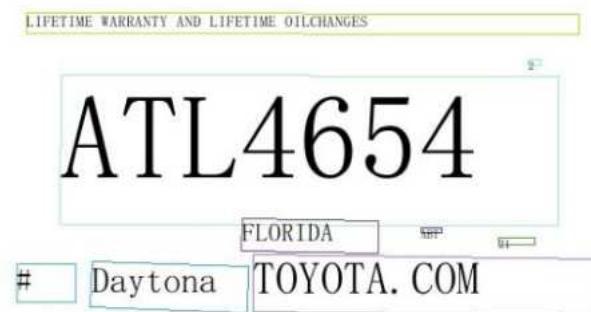
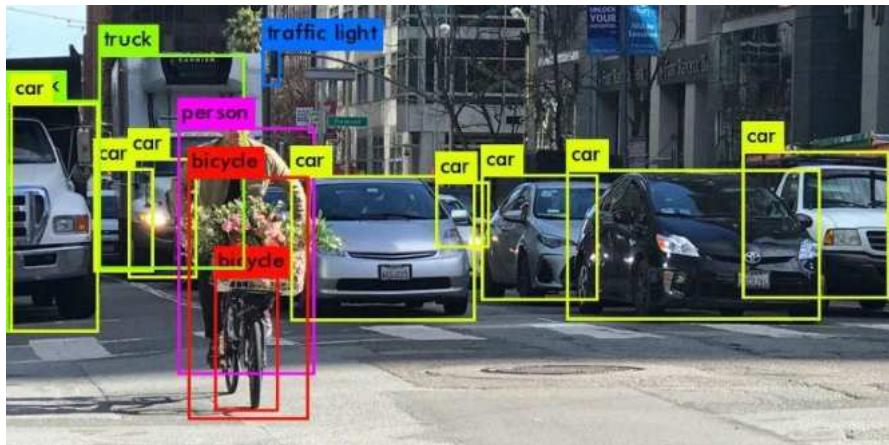
Logistic  
Regression



## Supervised Learning (Yo'naltirilgan o'qitish)

SL algoritmlari ham huddi shunday ishlaydi. Yani avval model aniq malumot bilan tanishririlib, takror-takror o'qitiladi. Natijada model o'zida berilgan malumotlarga oid ko'nikma hosil qiladi.





## Unsupervised Learning (Yo'naltirilmagan o'qitish)



Bu turgadi algoritmlar malumotlardan o'zlarini mustaqil o'rganadi. Ular bilan model qurishda malumotlar (dataset) bo'yicha yo'nalish (label) berish shart emas. Yani, biz modelga javobni aytmaymiz, va model o'zi o'rganib javobni topadi.

Masalan: Siz maktabga 1-marta borib sinfdoshlar bilan tanishdingiz. Oradan vaqt o'tgandan keyin avtomatik tarzda siz sinfdoshlaringizni ularning xolatidan kelib chiqib guruhlarga ajratasiz.

**Guruh 1:** O'qishni yaxshi ko'radian bollar,

**Guruh 2:** Dars qilishni yomon ko'radian bollar,

**Guruh 3:** Alochilar,

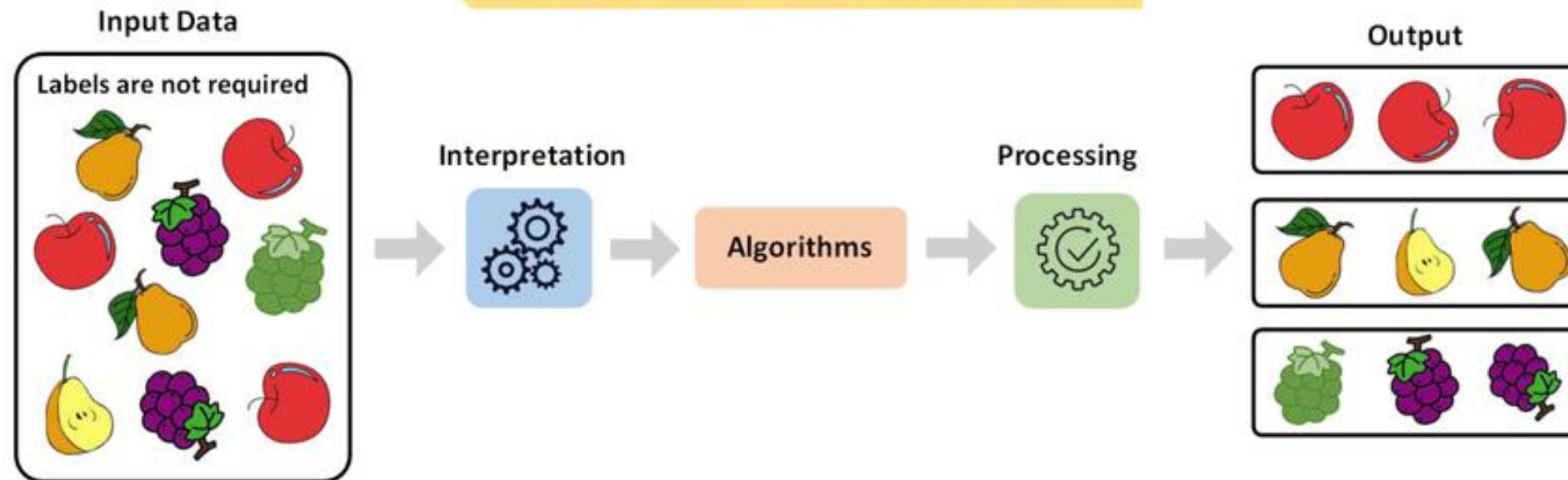
**Guruh 4:** Uyquchilar,

...





## UNSUPERVISED LEARNING



## Unsupervised Learning



Asosiy maqsadi malumotlardan guruhlarni topish, o'xshashliklarni topish va yashirin strukturani aniqlash.

Algoritm turi	Maqsadi	Misol
<b>Clustering (Guruhash)</b>	Malumotlardan bir sinfga tegishlilarni guruhlash	Masalan, insonlarni xulqiga qarab guruhlash
<b>Dimentionality reduction</b> (Hajmni zichlash, qisqartirish)	Malumotni asosiy qismlarini saqlagan holda malumotni soddalashtirish	Misol uchun yuqori pikselli rasmni pekselini pasaytirish
<b>Association Rule Learning</b> Bog'liqlilik qoidalarini o'rganish	Malumotlarni bir-biriga tegishlilik joylarini o'rganish	Tish cho'tkasi olgan odam, pastasini ham oladi.
<b>Anomaly detection</b> (No-odatiylikni aniqlash)	Guruhdan ajralganlarni topish	Sinfda tik turgan o'qituvchi

## Unsupervised Learning - Un-L



## Un-L ning hayotda qo'llanilishi

**CV** : rasmlarni ulardagi malumotlarda guruhlarga ajratish,

**Tibbiyotda**: yangi kasallik paydo bo'lishini aniqlash,

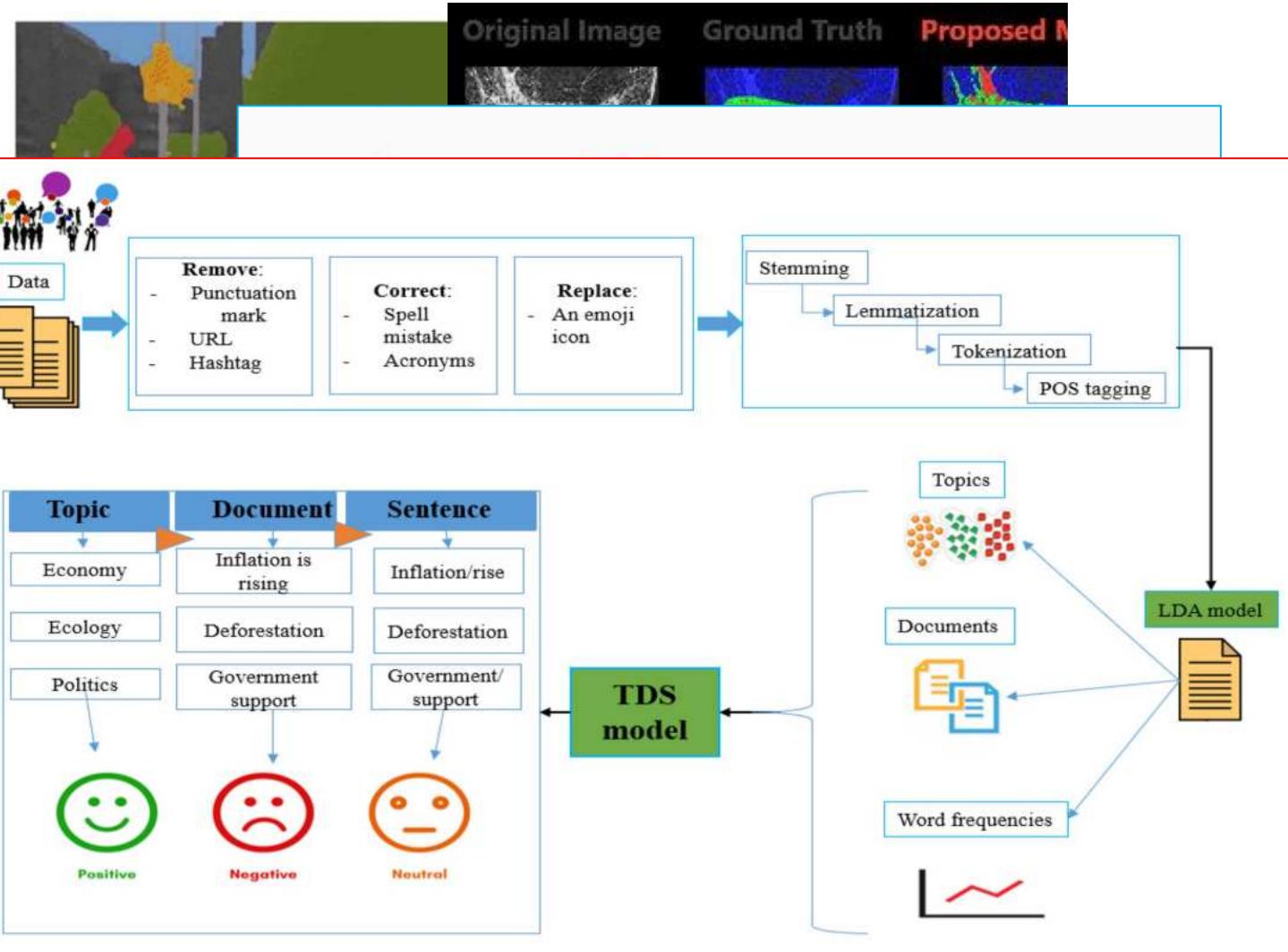
**Moliyada**: No-odatiy pul ko'chirmalarini aniqlash

**Marketingda**: Xaridchlarni segmentlash

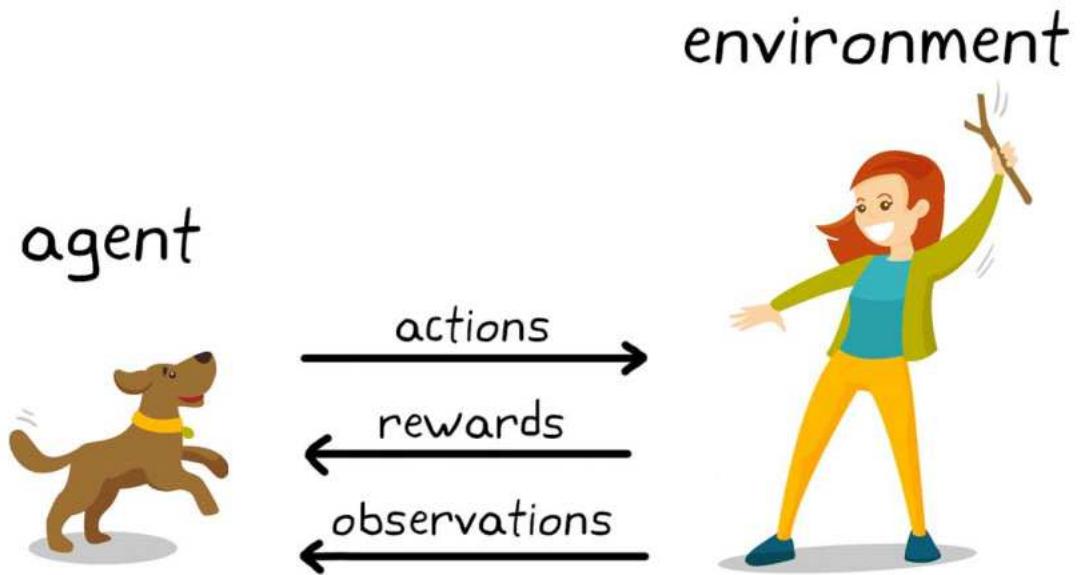
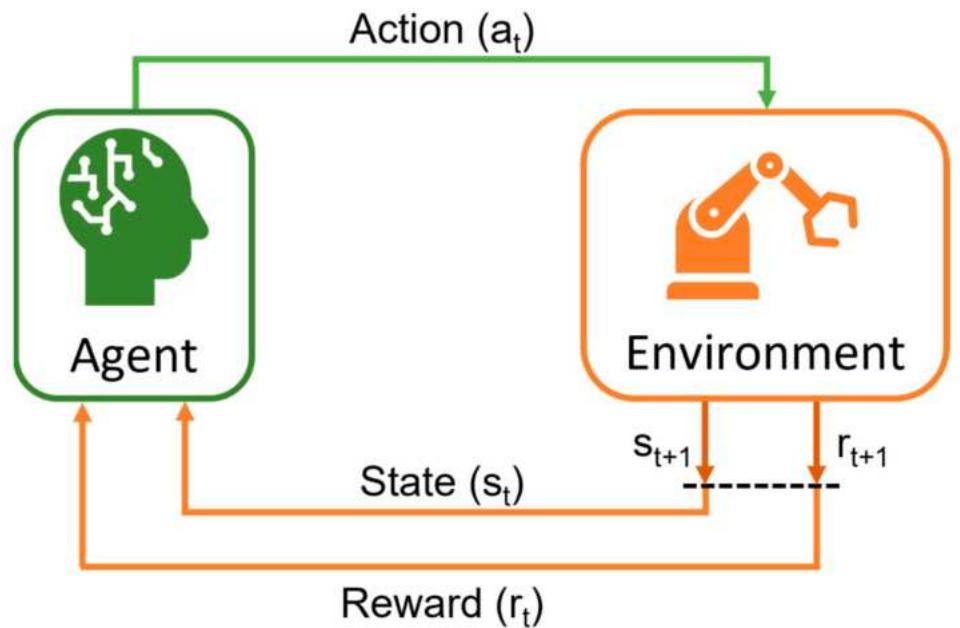
**Qishloq xo'jaligida**: O'simliklarni holatiga ko'ra sinflash

**NLP**: so'zlardan umumiyl mavzuni aniqlash

2021



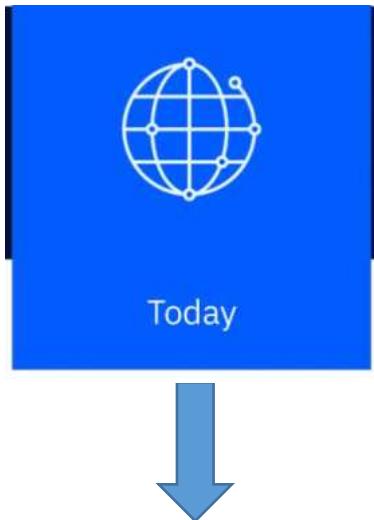
## Reinforcement learning



<https://youtu.be/Aut32pR5PQA?si=Y4jMt1kIT9JI3FE0>

[https://youtu.be/L\\_4BPjLBF4E?si=wFV\\_5lxa4wioKpye](https://youtu.be/L_4BPjLBF4E?si=wFV_5lxa4wioKpye)

### Bugungi kundagi ML modellarini rivojlanishi



Kuchsiz AI (Narrow AI) ga  
nisbatan kuchli, ko'plab  
vazifalarni bajara oladi



## Exploring Large Language Models (LLMs)



LLM: ➔

- Text - to - image
- Text - to - video
- Image - to - image
- Image - to - video
- Image - to - text

# Large Language Models

BCV



## Application Layer

### Copywriting

- Jasper
- copy.ai
- Headline co:here
- HyperWrite
- Writesonic
- Contenda
- unbounce
- copysmith

### Coding

- tabnine
- MUTABLE AI
- Codiga
- co:here
- GitHub Copilot
- CODEGEN

### Dev Tools

- algolia
- warp
- Mintlify
- cogram
- Debuild
- repl.it

### Chat / Comms

- MessageBird
- Replier.ai
- Sapling
- FABLE

### BizOps

- viable
- Enterpret
- tabulate
- Anecdote
- OTHERSIDE AI
- casetext
- Dover



## Infrastructure Layer

### Model Creation

- AI21labs
- OpenAI
- NVIDIA
- Adept
- EleutherAI
- ANTHROPIC
- Google AI

### Hardware

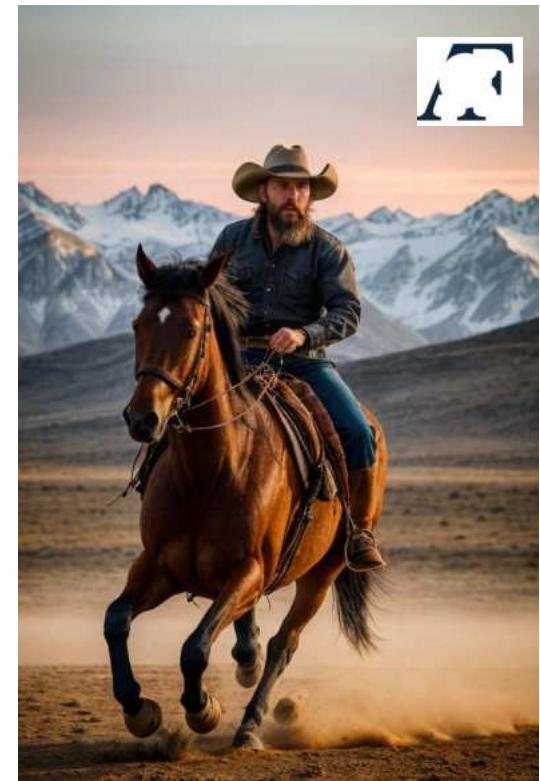
- habana
- SambaNova SYSTEMS
- cerebras
- GRAPHCORE
- NVIDIA

### Fine Tuning

- Google AI
- aws
- OpenAI
- Hugging Face

### Inference

- OpenAI
- Hugging Face



 Hugging Face

 Text Generation

 Any-to-Any

 Image-Text-to-Text

 Image-to-Text

 Image-to-Image

 Text-to-Image

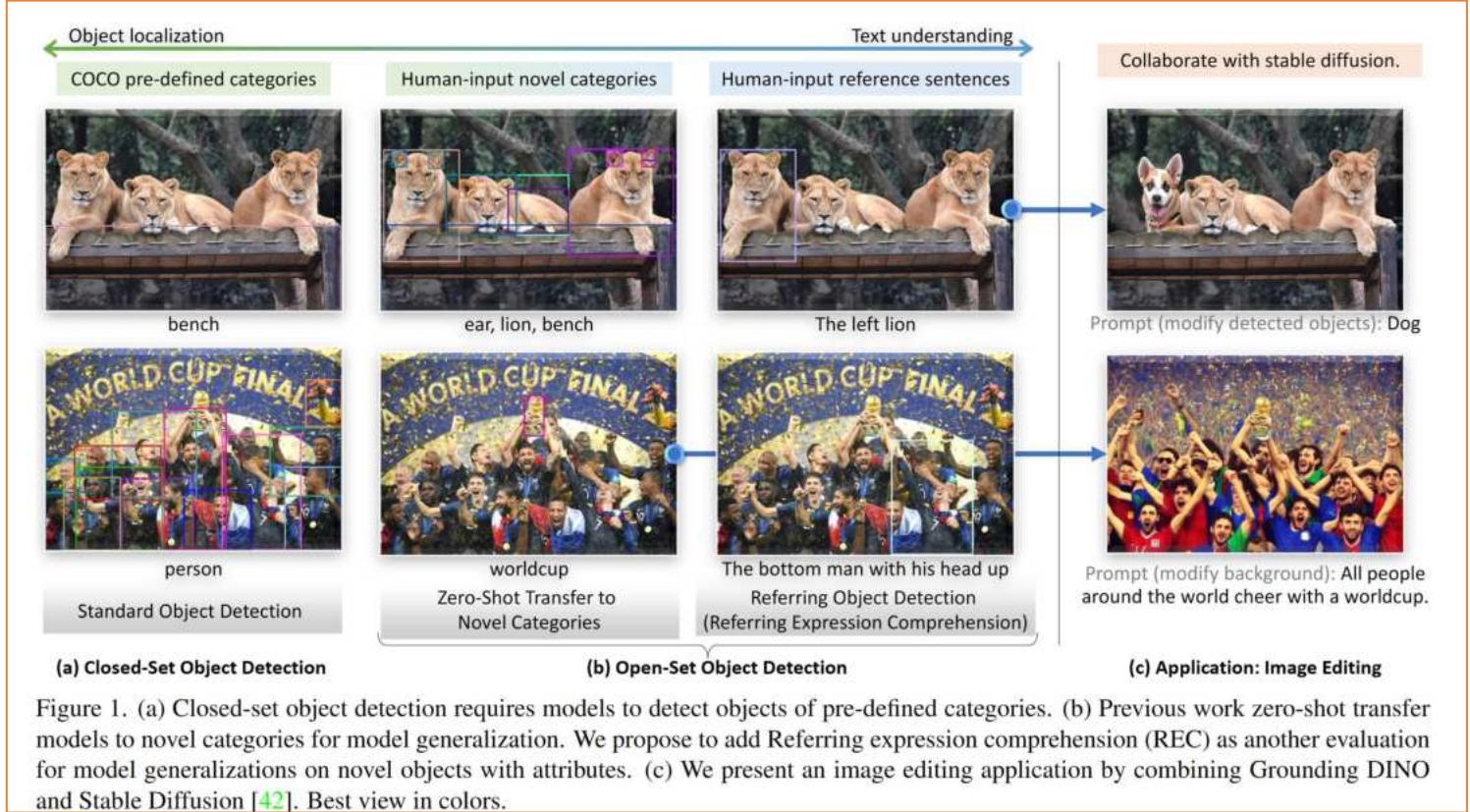
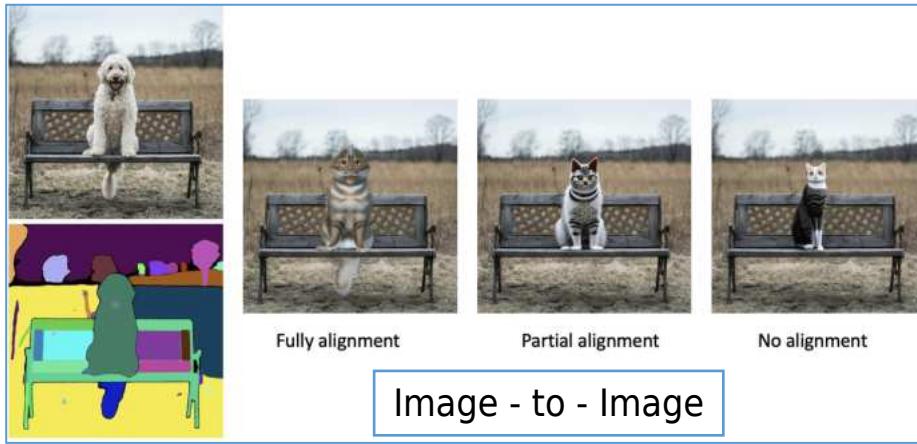
 Text-to-Video

 Text-to-Speech

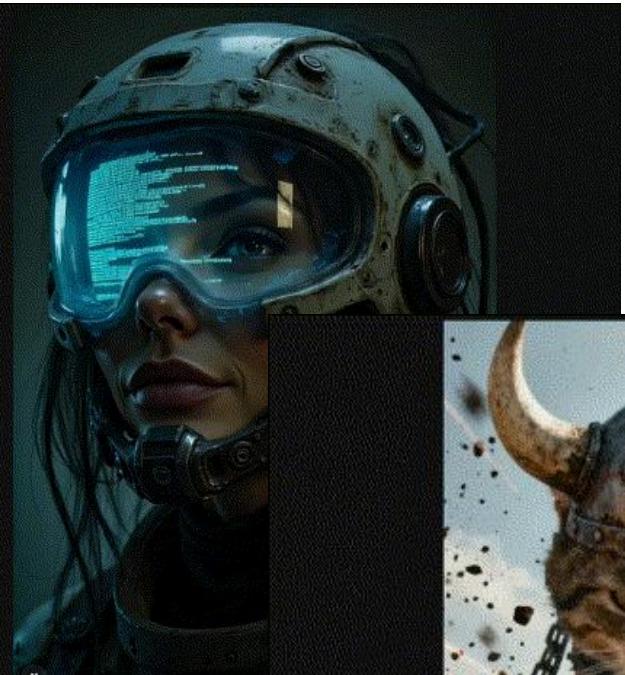
+ 42

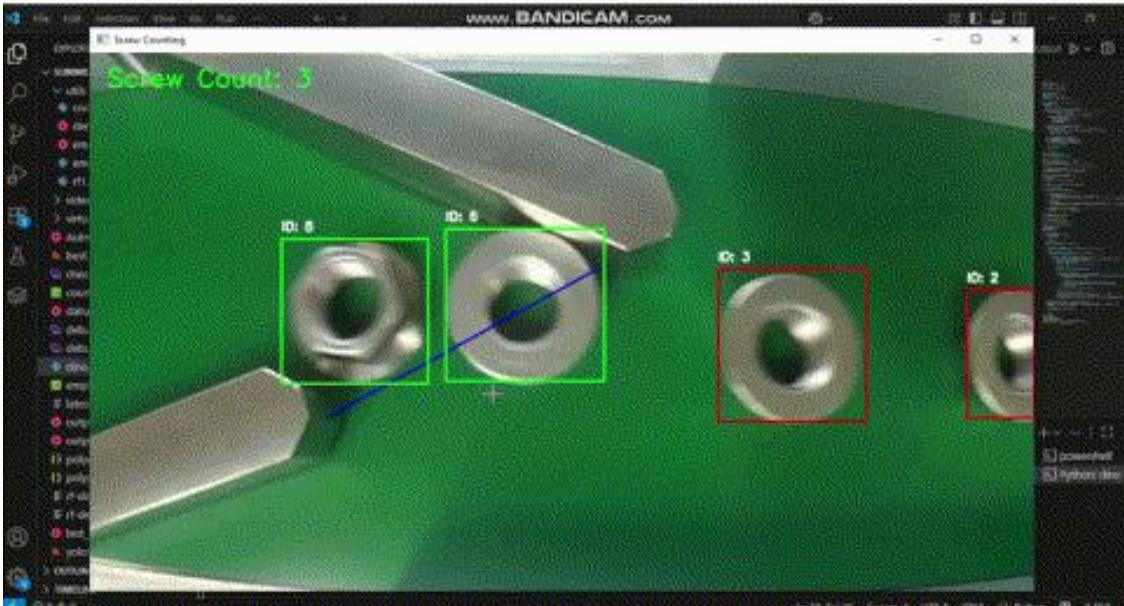
<https://huggingface.co/>

## Text-to-Image



Text-to-Video





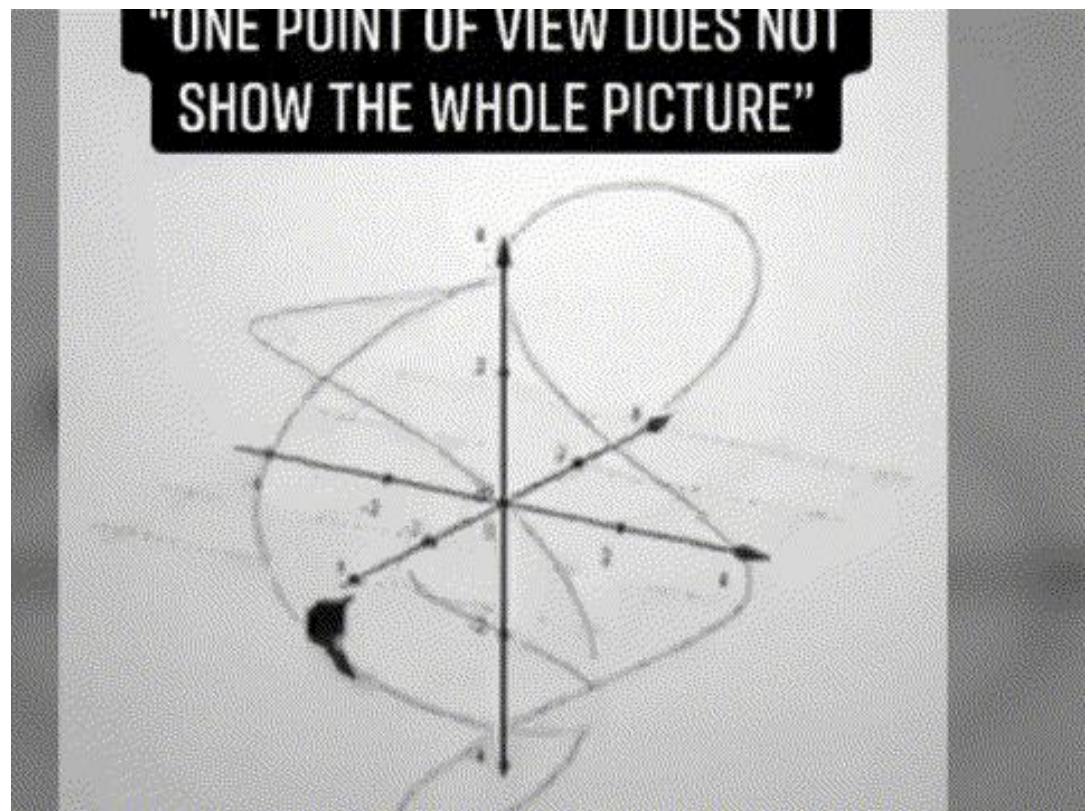
Hayoda qo'llanilishi

↗





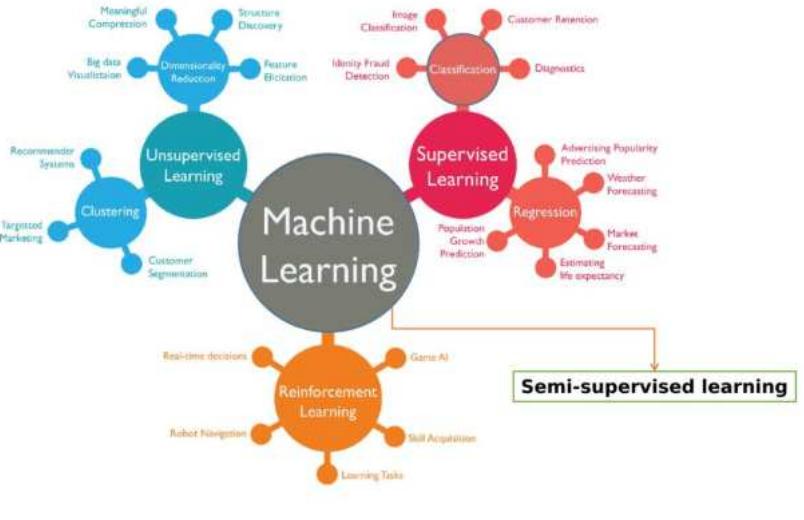
Text - to - video



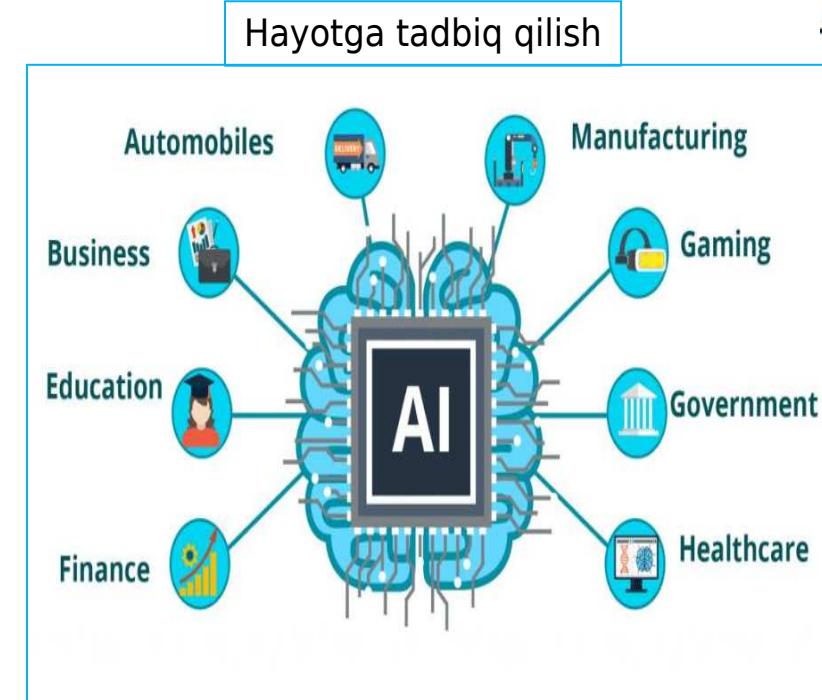




## Mashinani o'qitish asoslarini o'rganish



## ML o'qitish asoslari



## Integratsiya qilish





### Dasturlash tillari

- **Python**: AI / ML (Sun'iy aql va mashinani o'qitish uchun )
- **R**: statistik tahlillar
- **C++ / C** : kritikal Si, robotics / SI o'yinlar
- SQL

### AI/ML kutibxonalari / arxitekturalari

- **DL** (Chuqur o'qitish): pyTorch / TensorFlow / Keras / JAX
- **Ananaviy ML**: scikit-learn
- **CV** (Kompyuter Ko'rishi): OpenCV / Albuminations / YOLO
- **NLP** (tabiiy til bilan ishlash): Hugging Face / Transformers / NLTK / SpaCy

### Malumotlar bilan ishlash / Visualizatsiya

- Pandas / Numpy
- Matplotlib / Seaborn / plotly
- OpenCV / PIL
- Dask / PySpark

### Qurish vositalari

- IDE / Editors** - **VS Code**, PyCharm, Jupyter Notebook  
**Version Control** - Git, GitHub / GitLab  
**Package management** - pip, conda, poetry  
**Virtual Environments** - conda, venv, Docker,



## Joylashtirish va bulut vositalari

**Docker** - modellarni konteynerlashtirish.

**Kubernetes** - katta til modelllari uchun

**AWS / GCP / Azure** - bulutli infratuzilma

**ONNX / TorchScript** - modelni eksport qilish va joylashtirish uchun optimallashtirish.

Tajribalarni kuzatish va monitoring  
qilish

**Weights & Biases (wandb)**

**TensorBoard**

**MLflow**

## Valuable Tools

FastAPI / Flask / Django - building AI APIs.

Streamlit / Gradio - quick AI app demos.

OpenAI API / LangChain - integrating LLMs.

SQL + NoSQL databases - PostgreSQL,  
MongoDB.

Versioned datasets - DVC (Data Version Control).



Bilish zarur !

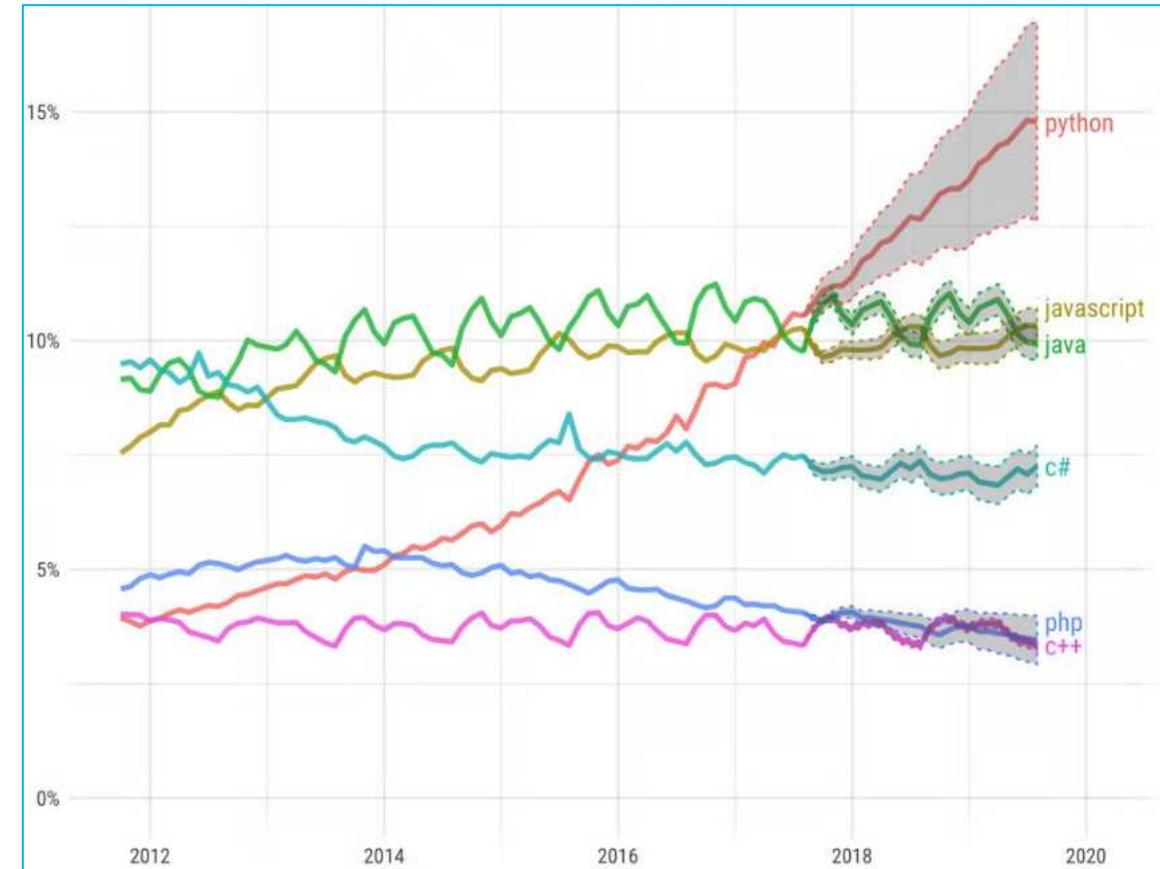
**Visual Studio** (VS) code ni ,  
yoki  
**PyCharm** ni o'rnatish

**Git Hub** account ochish va o'rganish

**Hugging Face** account ochish va o'rganish

**Kaggle.com** account ochish va o'rganish

**Google colab** o'rganish



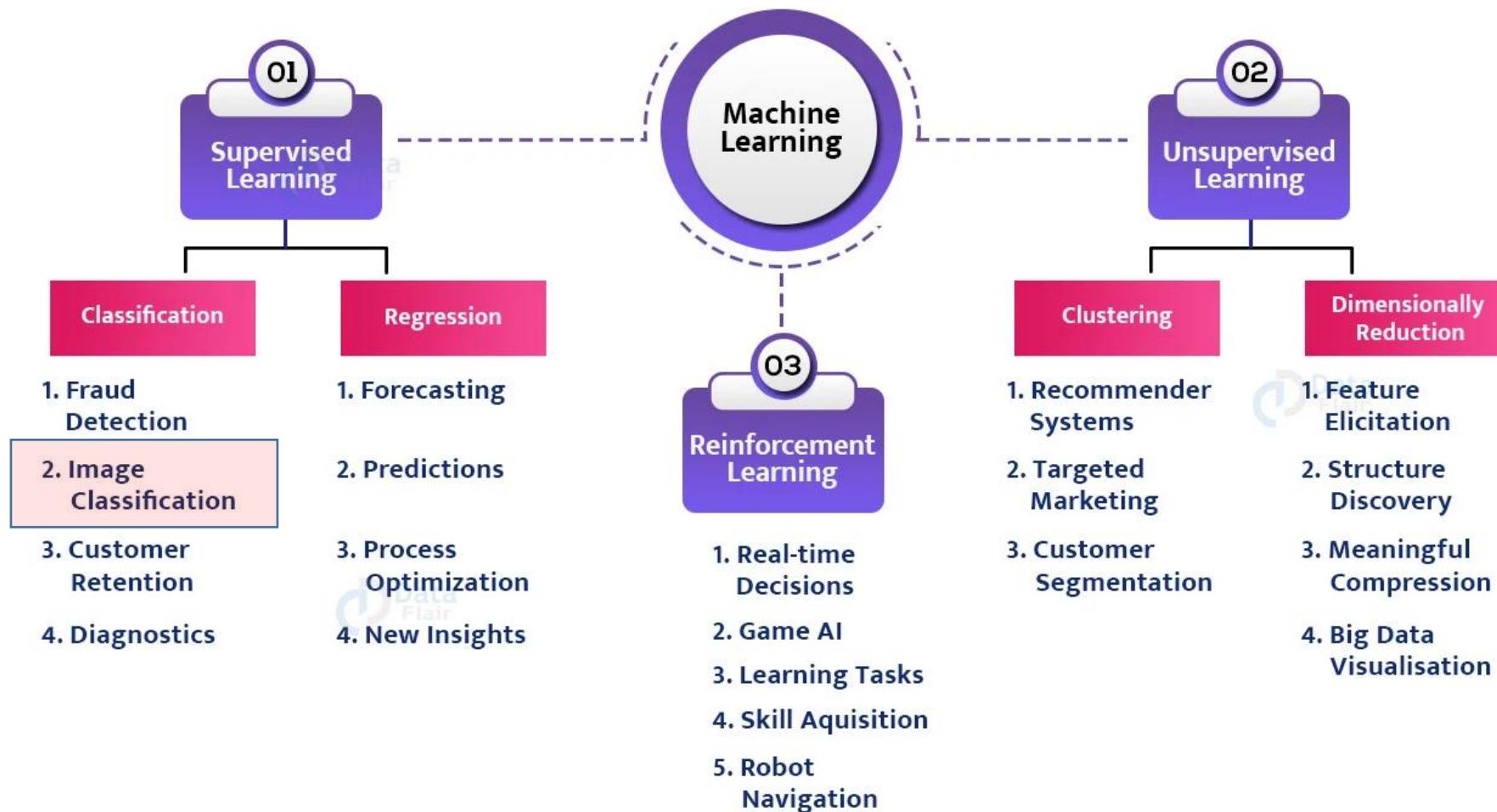
Biz **Python** DT (dasturlash tili) orqali dasrni o'rganamiz!



[https://github.com/nivu/ai\\_all\\_resources](https://github.com/nivu/ai_all_resources)

<https://github.com/armankhondker/awesome-ai-ml-resources>







## Supervised Learning (Nazorat ostidagi o'qitish)

Bu turdagi o'qitishda, algoritmlarga avval yo'nalish, analiz qilish imkoni beriladi va so'ng shu yo'nalish asosida algoritmlarda ko'nikma hosil qilinadi.

**SL** - algoritmlari 2 xil sinfga bo'linadi: **classification** and **regression**

### CV (Kompyuter ko'rishi) da SL Algoritm lari qo'llanilishi

#### K-Nearest Neighbors (K-NN)

Support Vector  
Machines (SVM)

Random  
Forest

Logistic  
Regression

Decision  
Trees

Naive bayes

**Kompyuter ko'rishi haqida qisqacha tushuncha**

5

R:165 G:157 B:073	R:159 G:140 B:064	R:131 G:101 B:037	R:098 G:058 B:009	R:081 G:035 B:000
R:117 G:100 B:028	R:109 G:085 B:021	R:093 G:062 B:008	R:081 G:042 B:000	R:085 G:039 B:006
R:064 G:047 B:000	R:070 G:042 B:000	R:074 G:040 B:000	R:090 G:048 B:010	R:136 G:067 B:034
R:052 G:027 B:000	R:064 G:034 B:000	R:086 G:050 B:016	R:118 G:078 B:042	R:154 G:106 B:070
R:059 G:031 B:007	R:086 G:051 B:024	R:119 G:070 B:053	R:154 G:111 B:079	R:183 G:133 B:098

**Inson ko'rishi****CV**

```

0 2 15 0 0 11 10 0 0 0 0 9 9 0 0 0 0
0 0 0 4 60 157 236 255 255 177 95 61 32 0 0 29
0 10 16 119 238 255 244 245 243 250 249 255 222 103 10 0
0 14 170 255 255 244 254 255 253 245 255 249 253 251 124 1
2 98 255 228 255 251 254 211 141 116 122 215 251 238 255 49
13 217 243 255 155 33 226 52 2 0 10 13 232 255 255 36
16 229 252 254 49 12 0 0 7 7 0 70 237 252 235 62
6 141 245 255 212 25 11 9 3 0 115 236 243 255 137 0
0 87 252 250 248 215 60 0 1 121 252 255 248 144 6 0
0 13 113 255 255 245 255 182 181 248 252 242 208 36 0 19
1 0 5 117 251 255 241 255 247 255 241 162 17 0 7 0
0 0 0 4 58 251 255 246 254 253 255 120 11 0 1 0
0 0 4 97 255 255 255 248 252 255 244 255 182 10 0 4
0 22 206 252 246 251 241 100 24 113 255 245 255 194 9 0
0 111 255 242 255 158 24 0 0 6 39 255 232 230 56 0
0 218 251 250 137 7 11 0 0 0 2 62 255 250 125 3
0 173 255 255 101 9 20 0 13 3 13 182 251 245 61 0
0 107 251 241 255 230 98 55 19 118 217 248 253 255 52 4
0 18 146 250 255 247 255 255 249 255 240 255 129 0 5
0 0 23 113 215 255 250 248 255 255 248 248 118 14 12 0
0 0 6 1 0 52 153 233 255 252 147 37 0 0 4 1
0 0 5 5 0 0 0 0 0 0 14 1 0 6 6 0 0

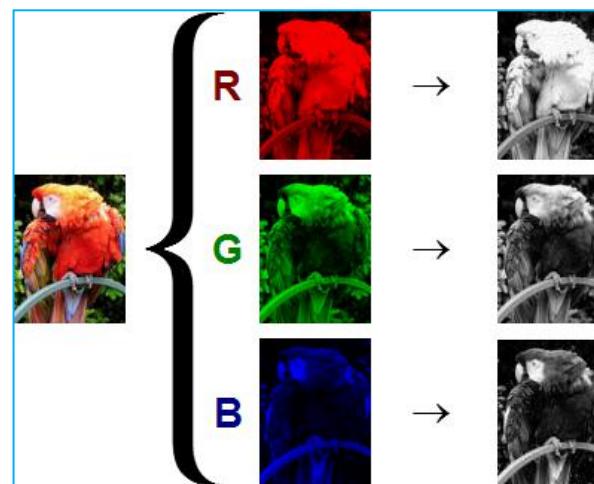
```

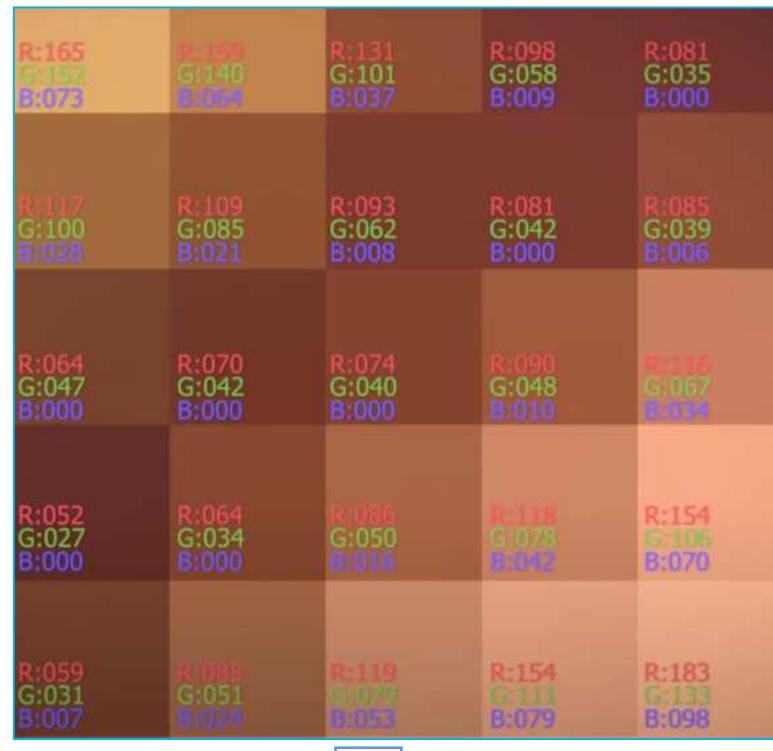
[https://en.wikipedia.org/wiki/RGB\\_color\\_model](https://en.wikipedia.org/wiki/RGB_color_model)

<https://rgbcolorpicker.com/>

**JPEG**

<https://youtu.be/Kv1Hiv3ox8I?si=KYbhjhp2y1mP8va>



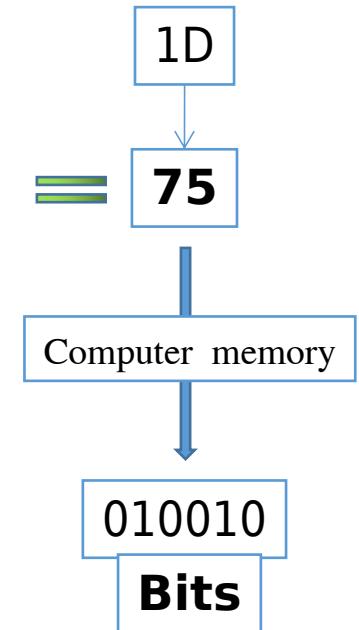


5 + 5 = 10

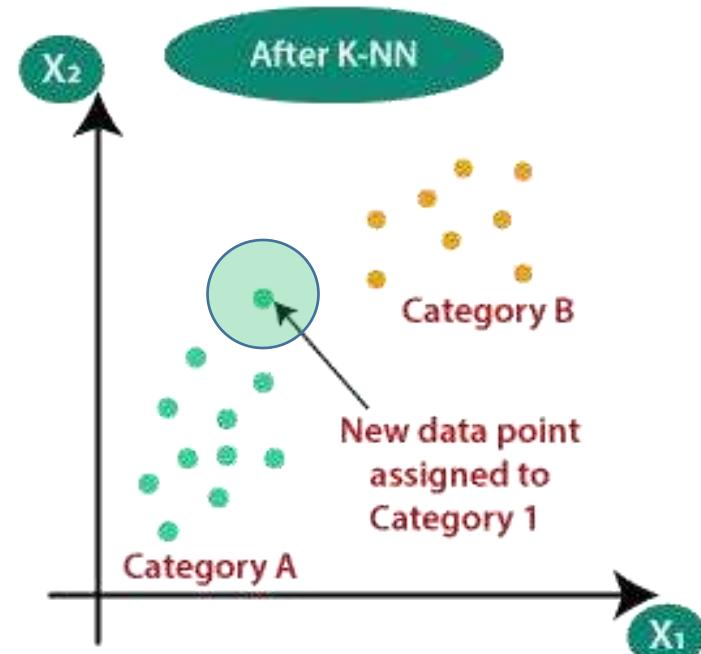
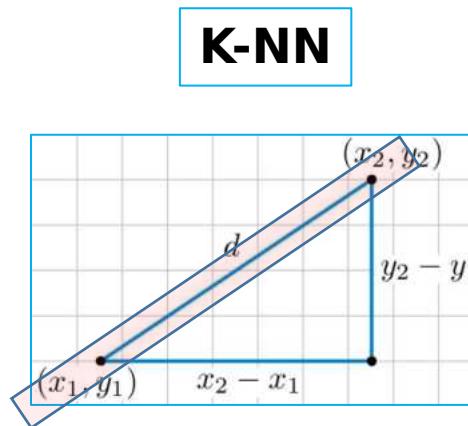
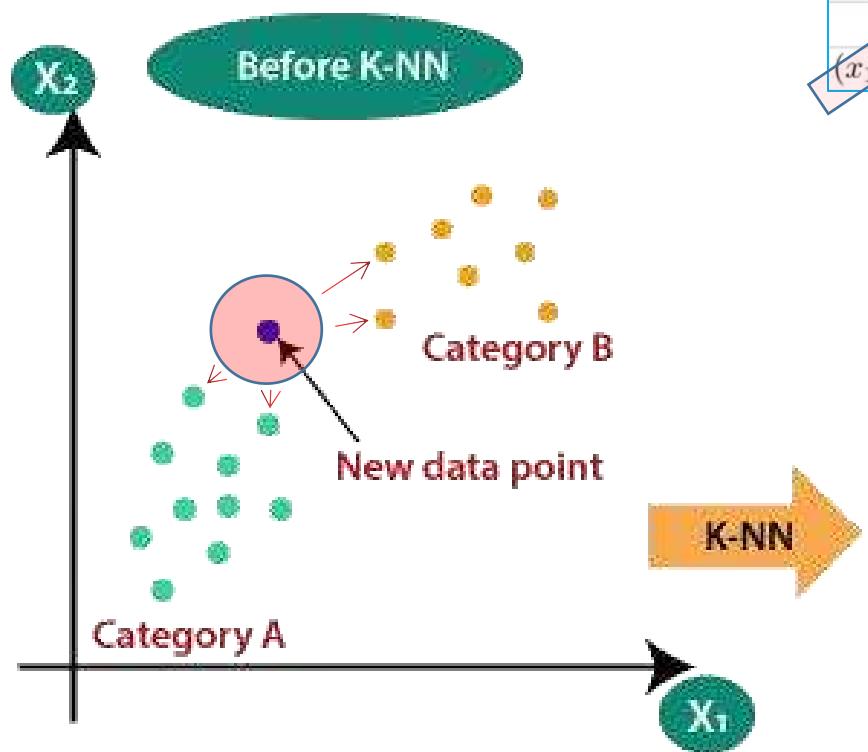


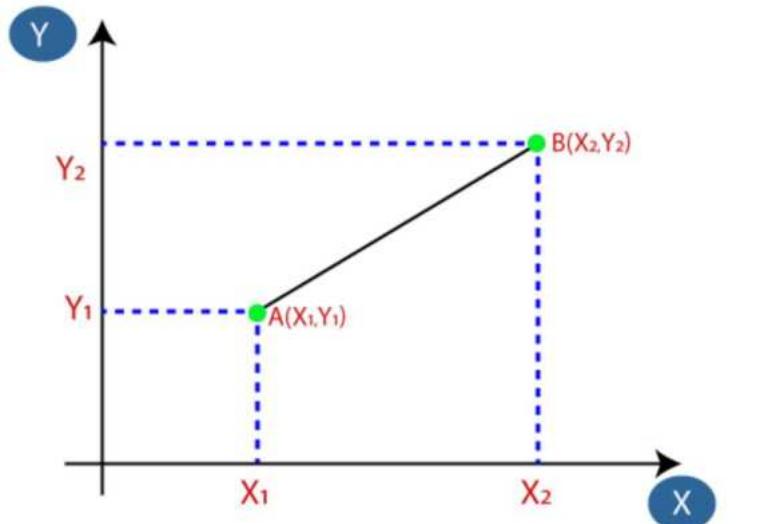
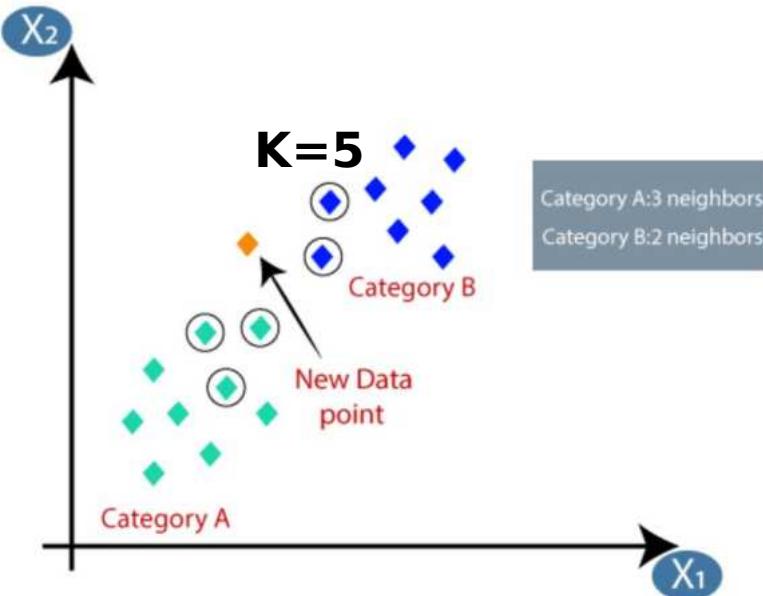
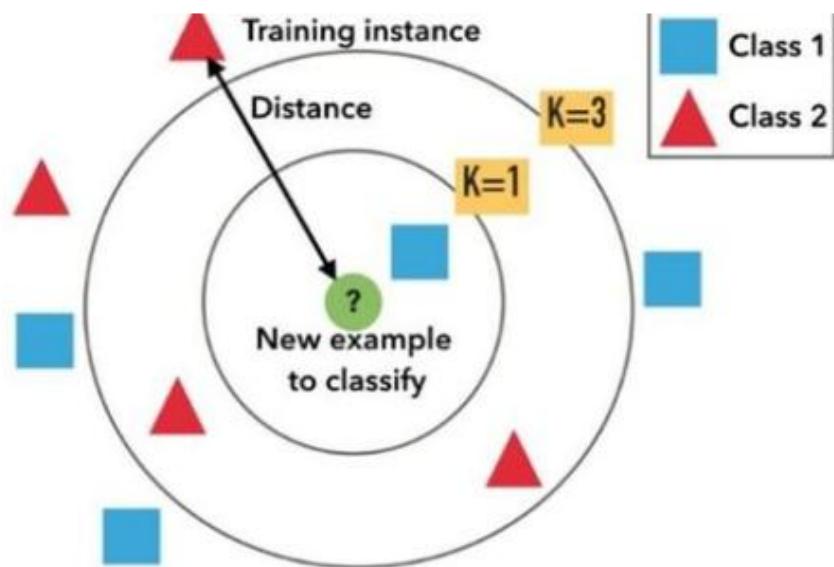
$$3 = 3 \times 3 \times 3 = 27$$

$$5 * 5 * 3 = 75$$



<https://youtu.be/7J7X7aZvMXQ?si=i0NiTUOYqCUFLw3r>



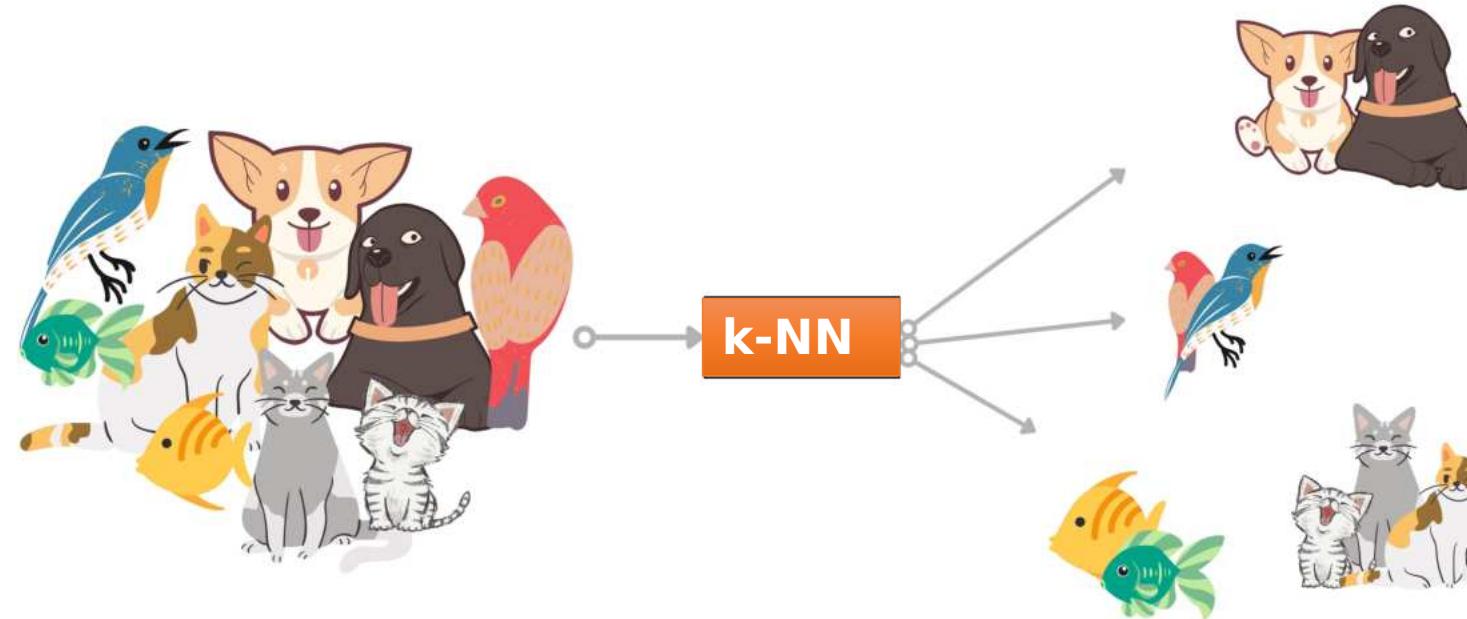


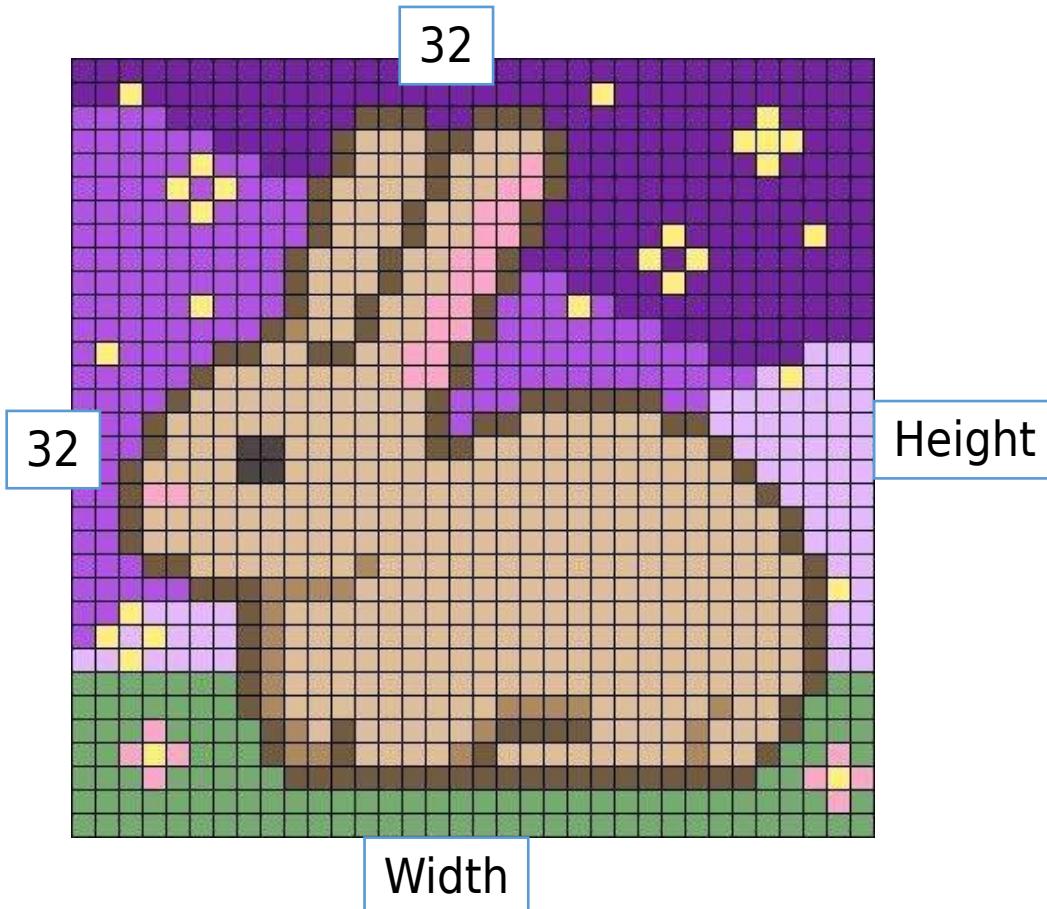
$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$



**K-NN** algoritmi rasmni qanday qilib guruhlaydi, gurularga ajratadi?

yoki rasmi guruhlashda qanday ishlaydi?



**K-NN****A**

→  $32 \times 32 \text{ RGB rasm} \rightarrow 32 \times 32 \times 3 = \mathbf{3,072}$

→

Muhim, etiborli qismlarni ajratadi

→

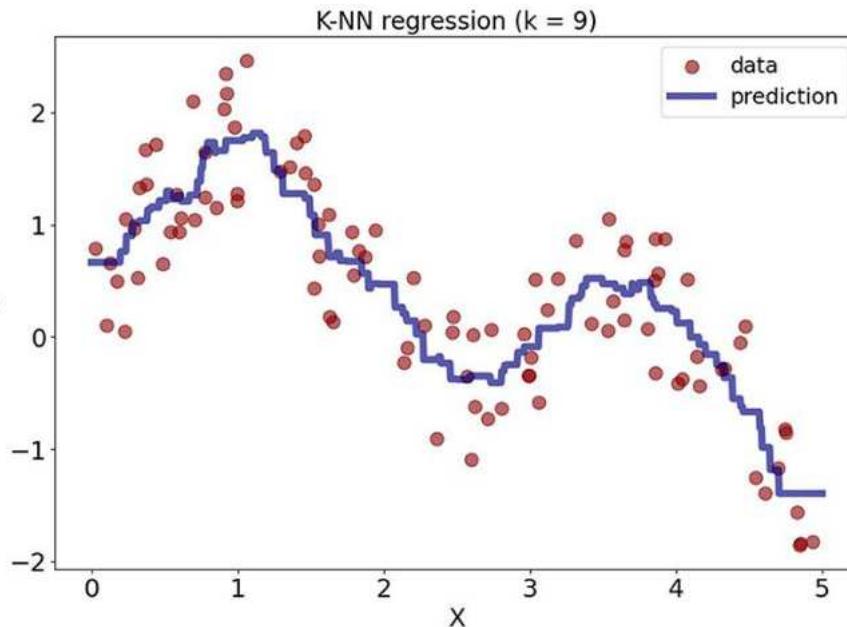
Olingan muhim qismlarni taqqoslaydi

Test qilinuvchi rasmlarni **A** bilan taqqoslaydi

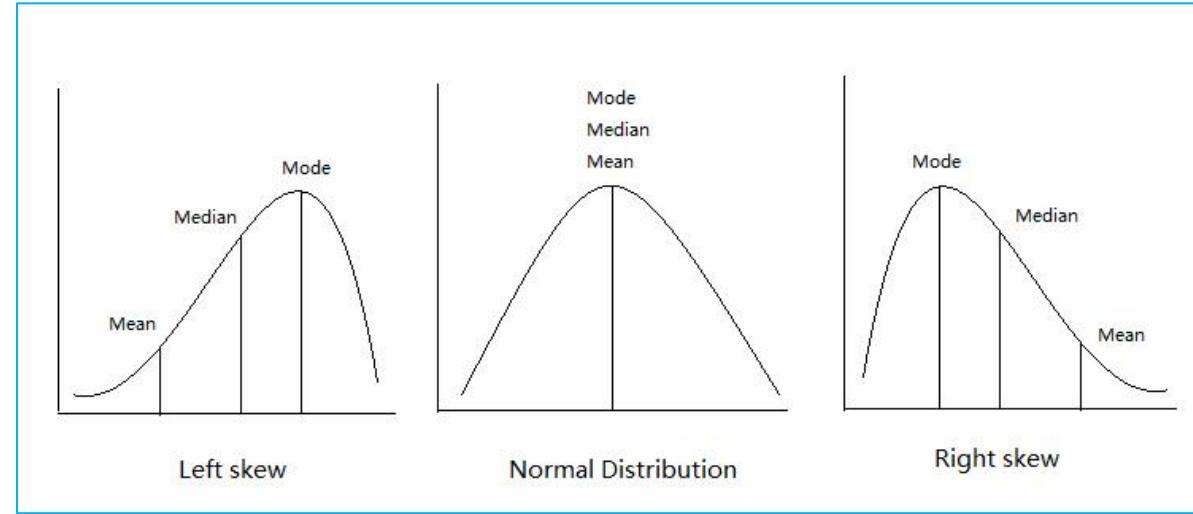
K-NN - dangasa, erinchoq algoritm deb qiyosiy nomlanadi. K-NN algoritmi o'rganishdan **tajribaga, ko'mikmaga** ega bo'lmaydi. U shunchaki butun o'rganishi kerak bo'lgan malumotlarni o'rganib chiqadi va yangi malumotni taqqoslab, guruhlarga ajratadi.



### K-NN regression masalalarini yechishda ishlatalishi



Regression masalalarini K-NN **mean** yoki **median** masofalarni o'lchab hal qiladi



**mean** - o'rtacha, **mode** - eng ko'p takrorlanuvchi, **median** - o'rta qiymat, **range** = max - min

**Example 01** Find the Mean, Median, Mode, and Range of the data set:

Goals Scored Over the Last 7 Games



1 3 4 6 6 7 8

**mean** 5  
average

**mode** 6  
most common

**median** 6  
middle

**range** 7  
largest - smallest

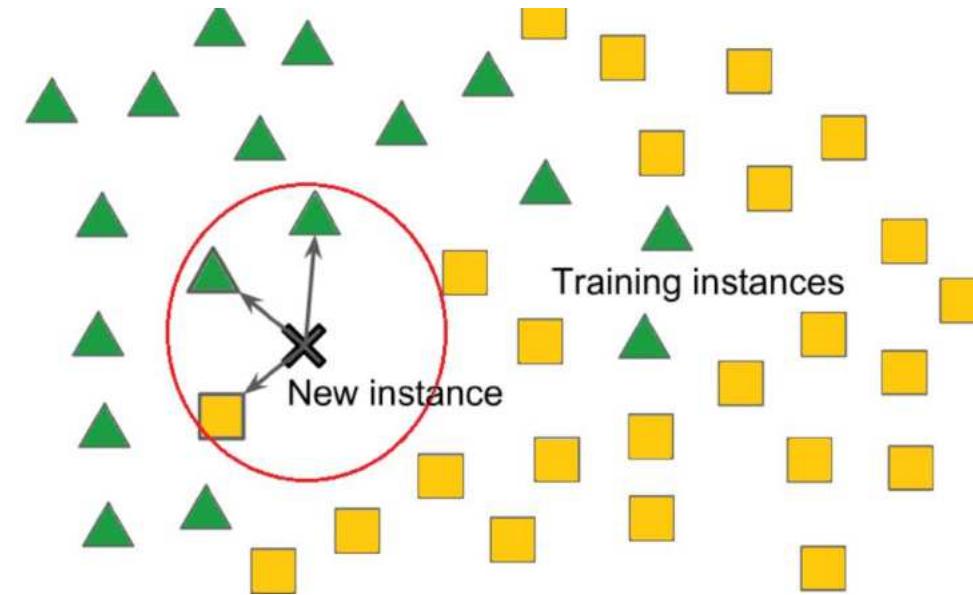


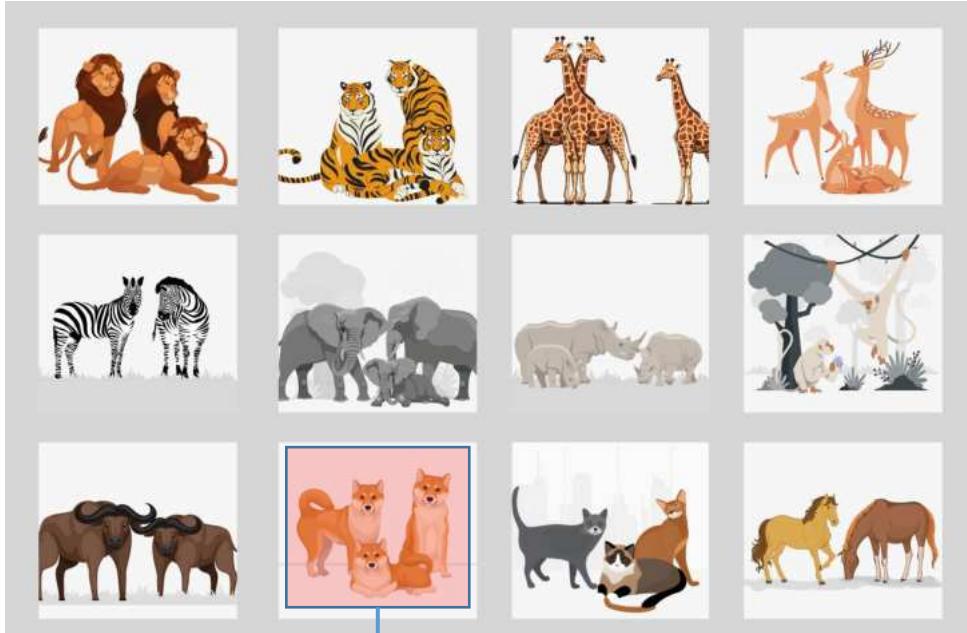
## K-NN

**K-NN classification** masalalarini yechishda ishlatalishi



Bunda, avval aytganimizdek, **K-NN** butun o'rganishi kerak bo'lgan malumotlarni o'rganib chiqadi va yangi malumotni taqqoslab, guruhlarga ajratadi





**K-NN**



A

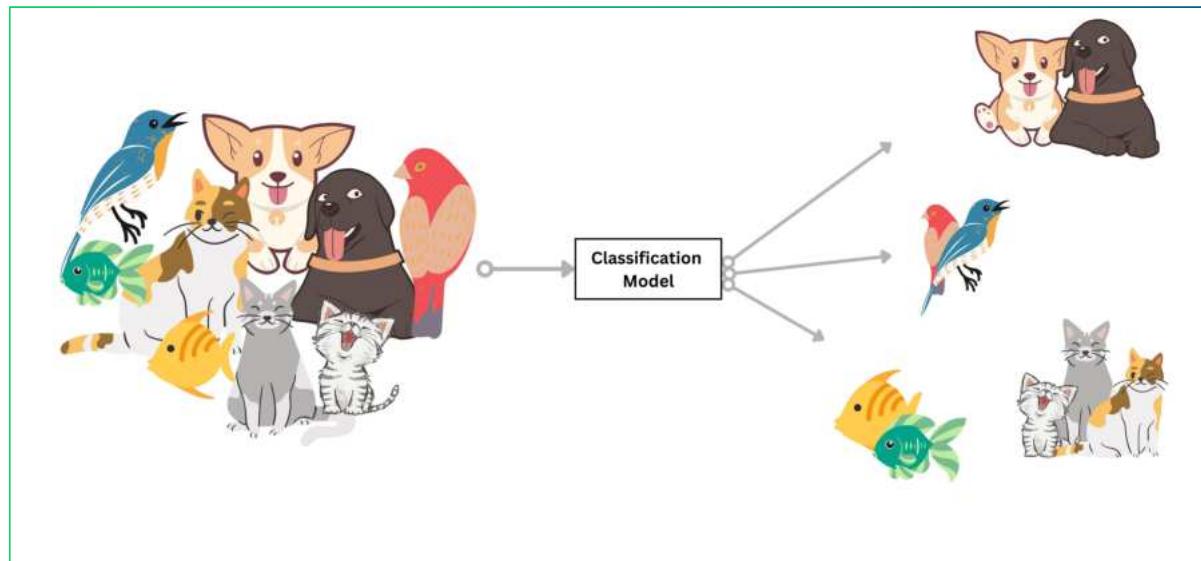
98403

98433

98464

98....

A





## K-NN



Xulosa !

Bu algoritm o'rganadigan malumotlar ko'p bo'lqanda xato qilish ehtimoli oshadi.

Bigger dataset's size

Higher computational cost





### K-NN coding

[https://github.com/fartech00/Prompt-refining/blob/main/TAFU\\_KNN.ipynb](https://github.com/fartech00/Prompt-refining/blob/main/TAFU_KNN.ipynb)





K-Nearest Neighbors (K-NN)

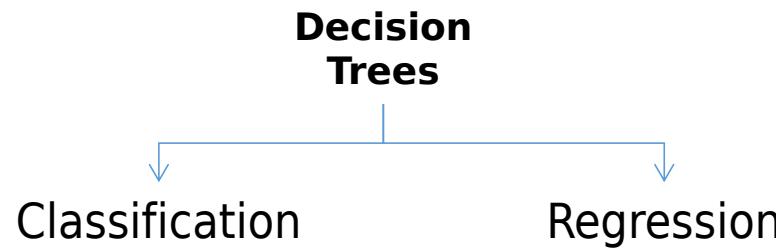
Support Vector Machines (SVM)

## Decision Trees

Logistic Regression

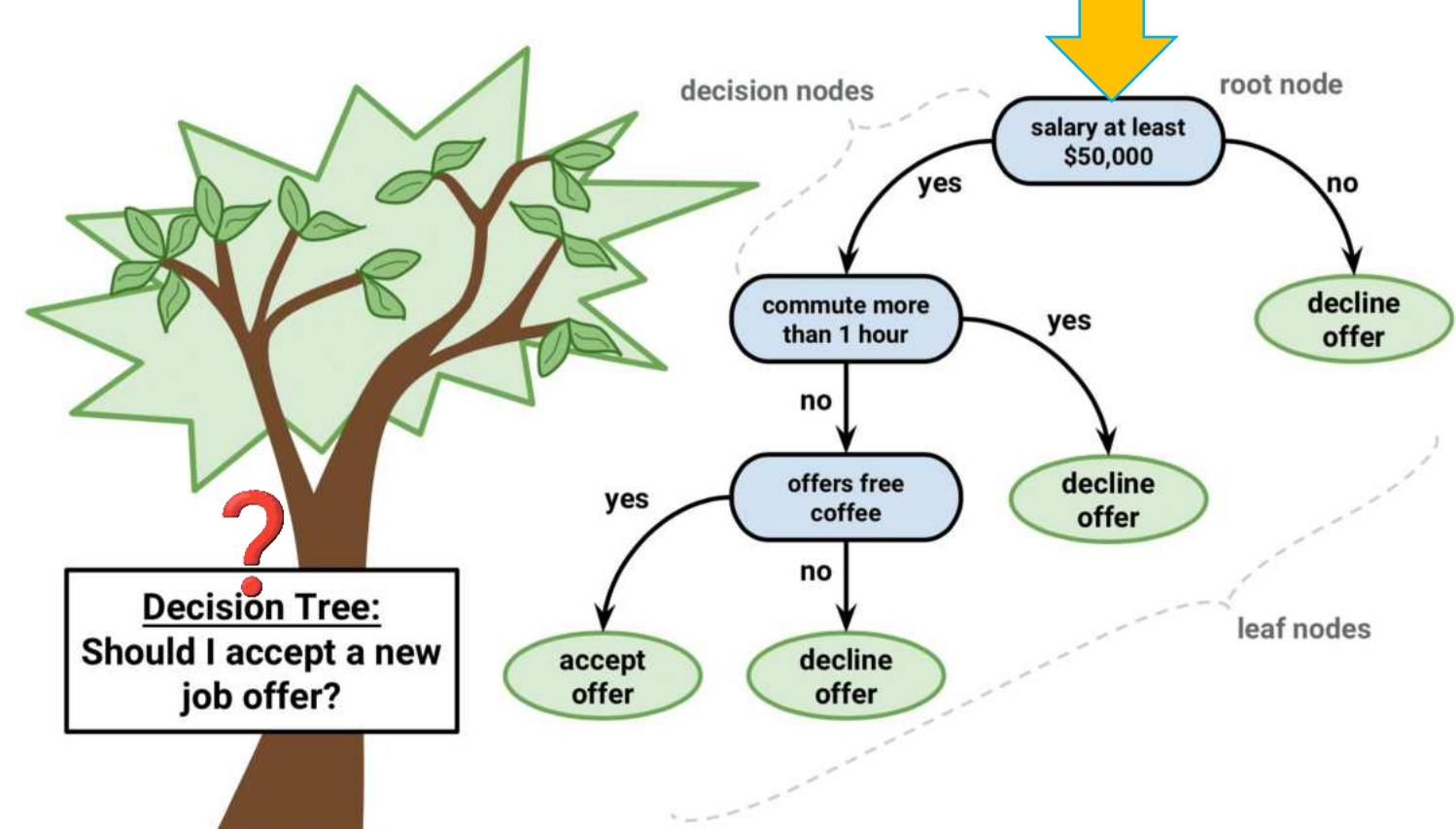
Random Forest

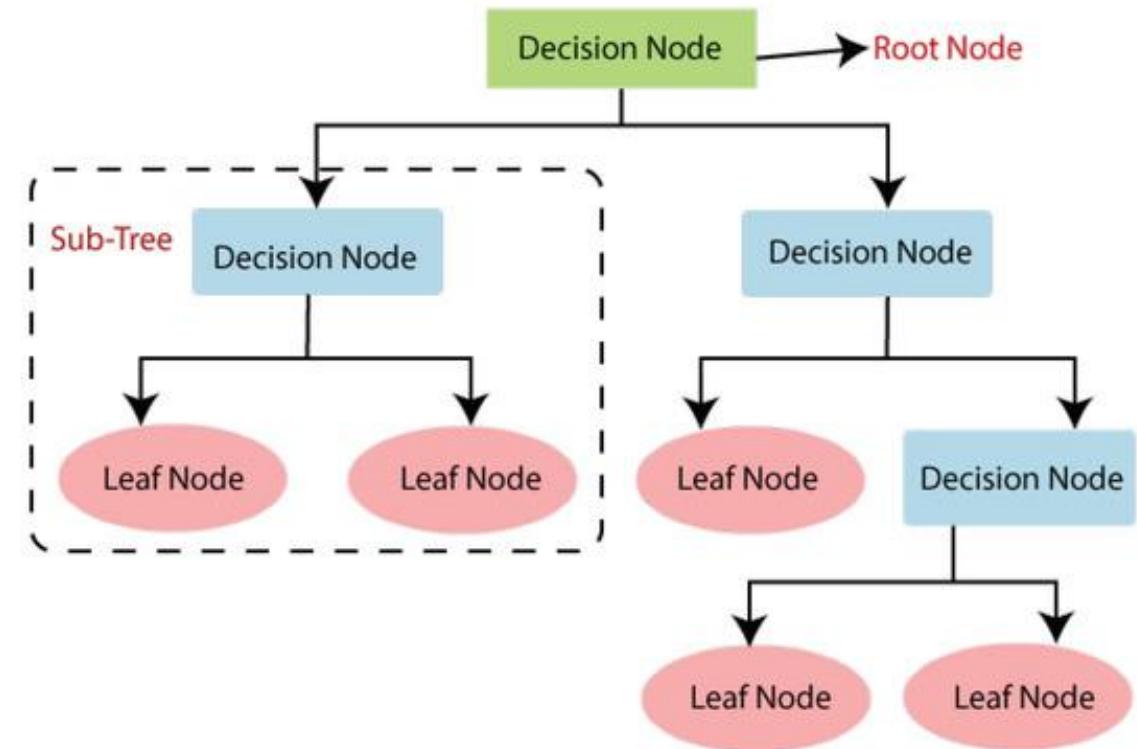
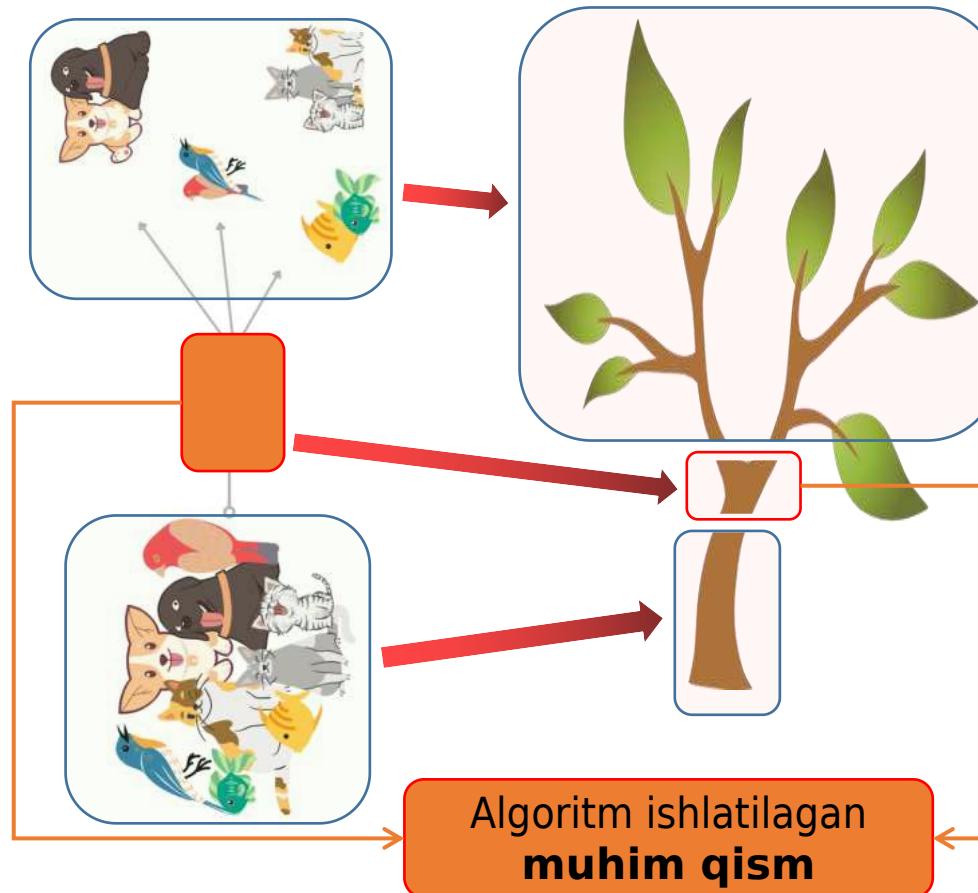
Naive bayes

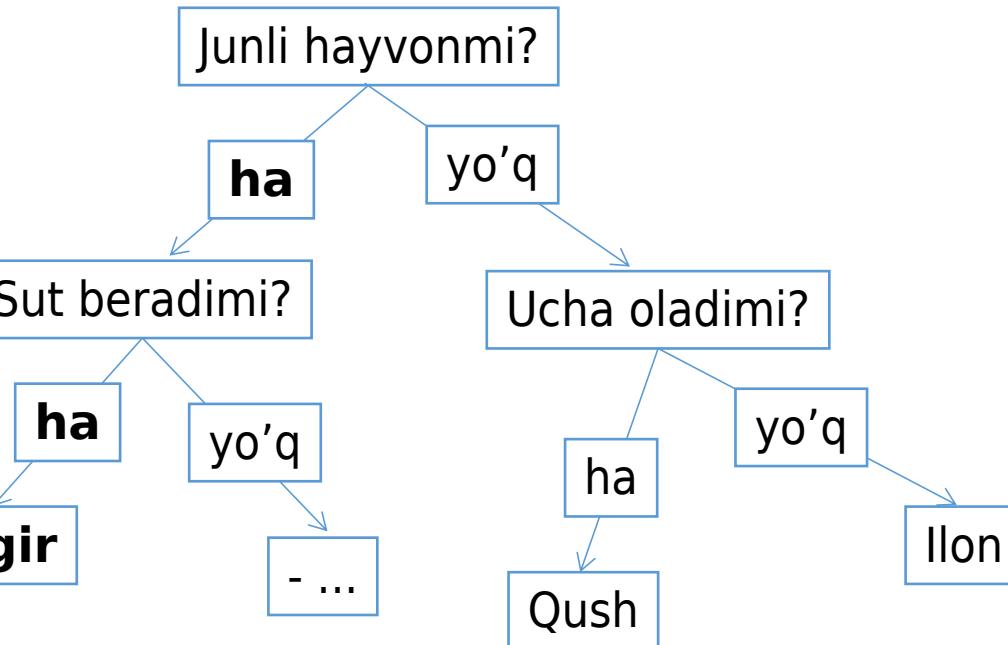


Bu algoritm bir muammoni yechishda qarorni guruhlarga bo'lib boradi, va javobga yetguncha davom etadi. Qaror qabul qilib borishi daraxt kabi ko'rinishni ifodalaganini uchun "**decision tree**" deb nomlanga, ya'ni "**qaror qabul qilish dataxti**"

Savolni o'zak, asos deb olsak, javobga yetish barg bo'ladi va bargdan uyog'iga boshqa savol qolmaydi.







Oyoqlari soni?  
Shoxlimi?  
...



Rasmli malumot bilan DT qanday ishlaydi?

Bu holatda CNN kabi har bir sinfni kerakli, **muhim qismlari** guruhlanadi va o'sha sinf bo'yicha klassifikatsiyaga ajratiladi.

Masalan:

- Qushlar sinfini o'rtacha rangi
- Edge lari
- Bo'yi / Eni
- Tekstura naqshlari

Edge	F1	F2	F3	F <sub>n-1</sub>	F <sub>n</sub>	Klassifikatsiya sinfi
	0.21	0.34	0.56	...	0.12	Mammal
	0.14	0.78	0.65	...	0.33	Bird
	0.66	0.12	0.77	...	0.44	Reptile

## CV (Kompyuter ko'rishi) da SL Algoritm lari qo'llanilishi



~~K-Nearest Neighbors (K-NN)~~

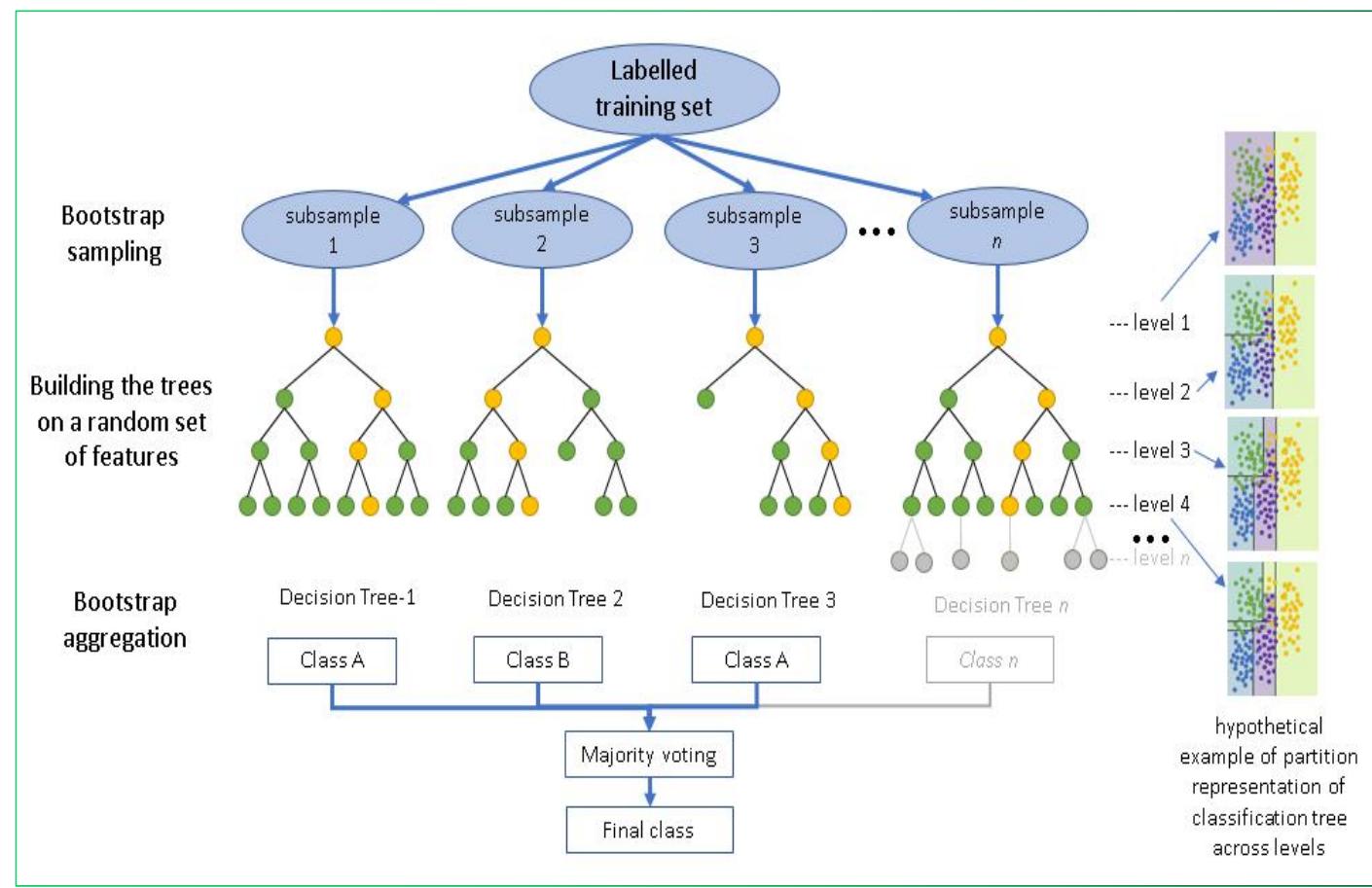
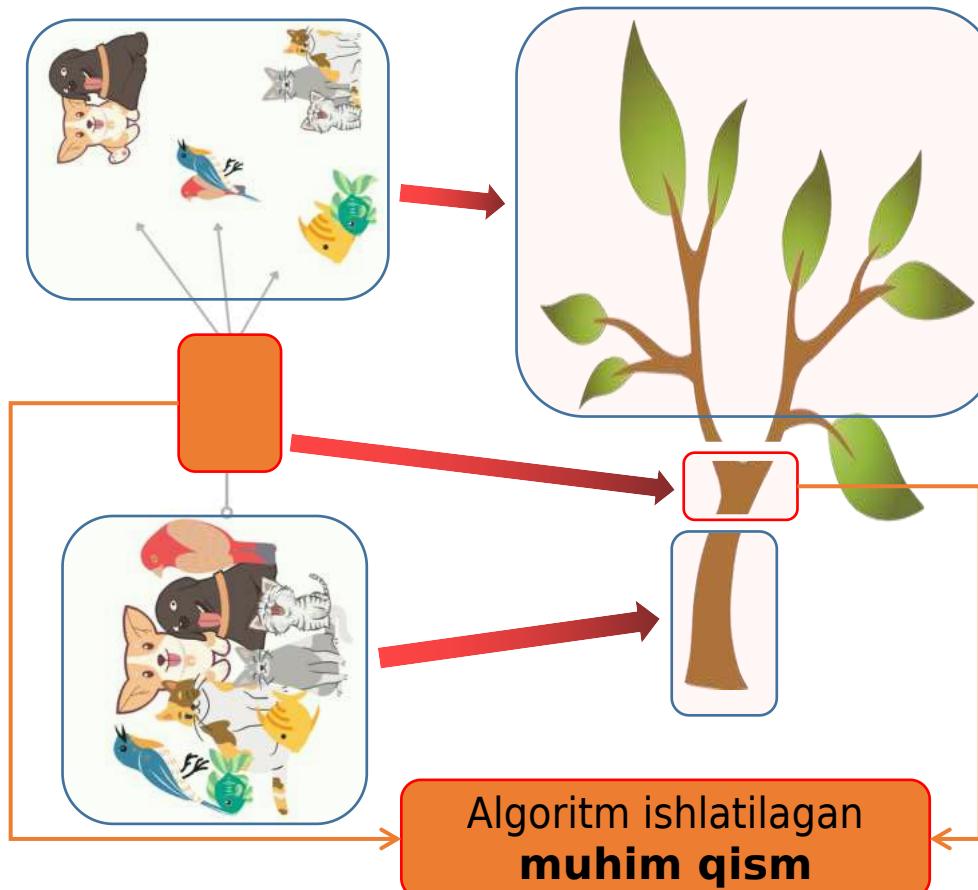
Support Vector Machines  
(SVM)

Decision  
Trees

Logistic  
Regression

**Random  
Forest**

Naive bayes





Random Forest (**RF**) - strukturasi DT ga o'xshaydi, lekin ko'plab DT lardan tashkil topadi.

DT ga nisbatan aniqliligi ancha yuqori. Katta malumotlar bilan ishlashda xatolik ehtimoli DT ga nisbata kam.

Har bir daraxt DT qarorini ifodalasa, shu daraxtlarning kop'chiligi maqullagan qarorni **RF** algoritmi yakuniy qaror qilib qabul qiladi. Shuning uchun ham bu algoritm **Taxminiy o'rmon** - **Random forest** deb nomlangan.

[https://github.com/fartech00/Prompt-refining/blob/main/TAFU\\_KNN.ipynb](https://github.com/fartech00/Prompt-refining/blob/main/TAFU_KNN.ipynb)

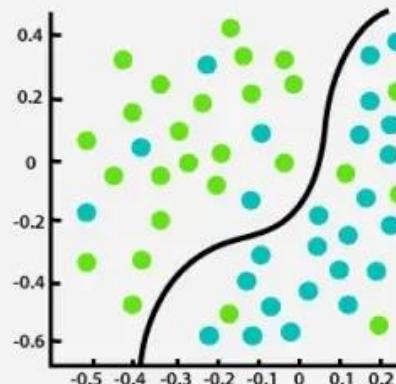


~~K-Nearest Neighbors (K-NN)~~~~Decision  
Trees~~~~Random  
Forest~~Support Vector Machines  
(SVM)Logistic  
Regression

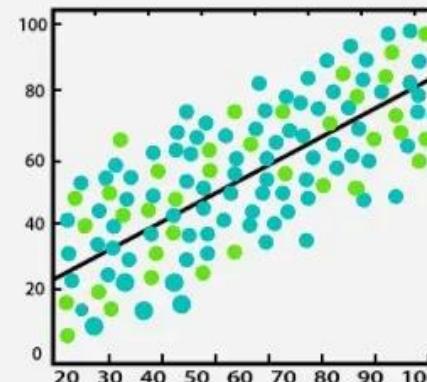
Naive bayes

SVM algoritmi SL turiga kiradigan algorithm hisoblanadi. U asosan **klassifikatsiya** va **regressiya** uchun ishlatiladi.

SVM ning asosiy maqsadi malumotlarni guruhga ajratuvchi **eng maqul chiziqni chizish**. Chiziq egri va to'g'ri bo'lishi mumkin.



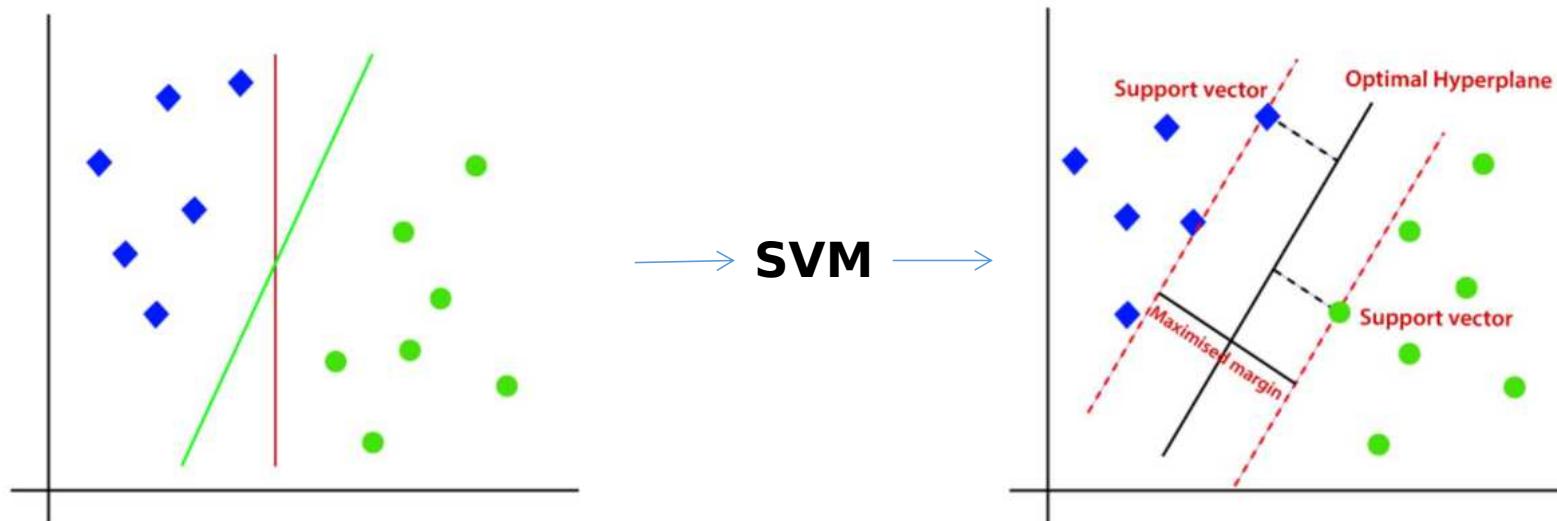
Classification



Regression



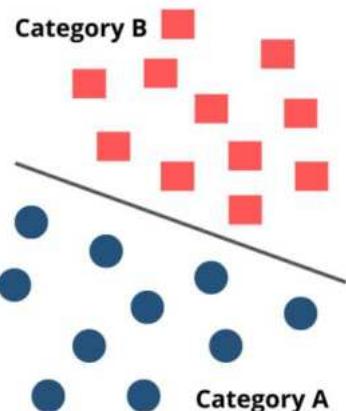
## SVM klassifikatsiya uchun qo'llanilishiga misol



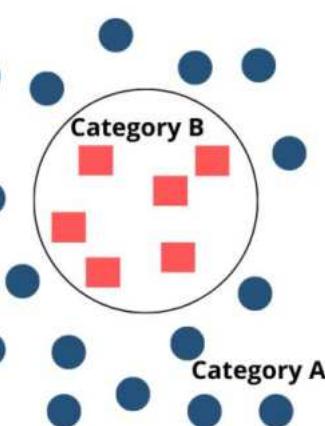


SVM ni ikki turi bor:

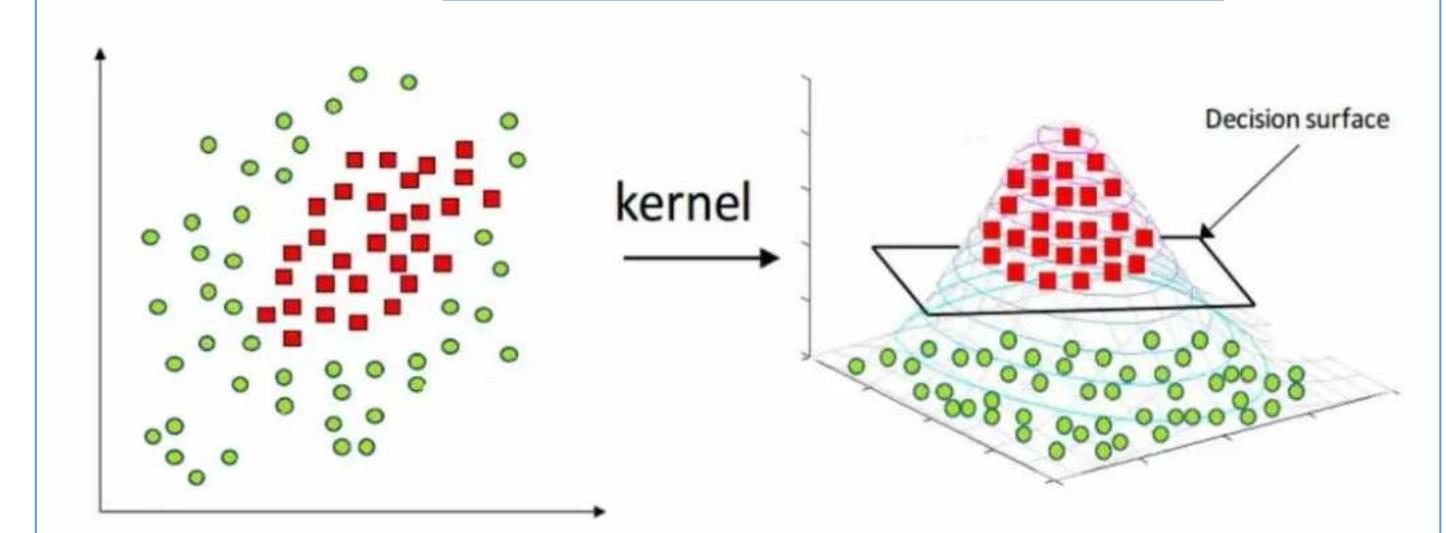
Linear SVM



Non Linear SVM

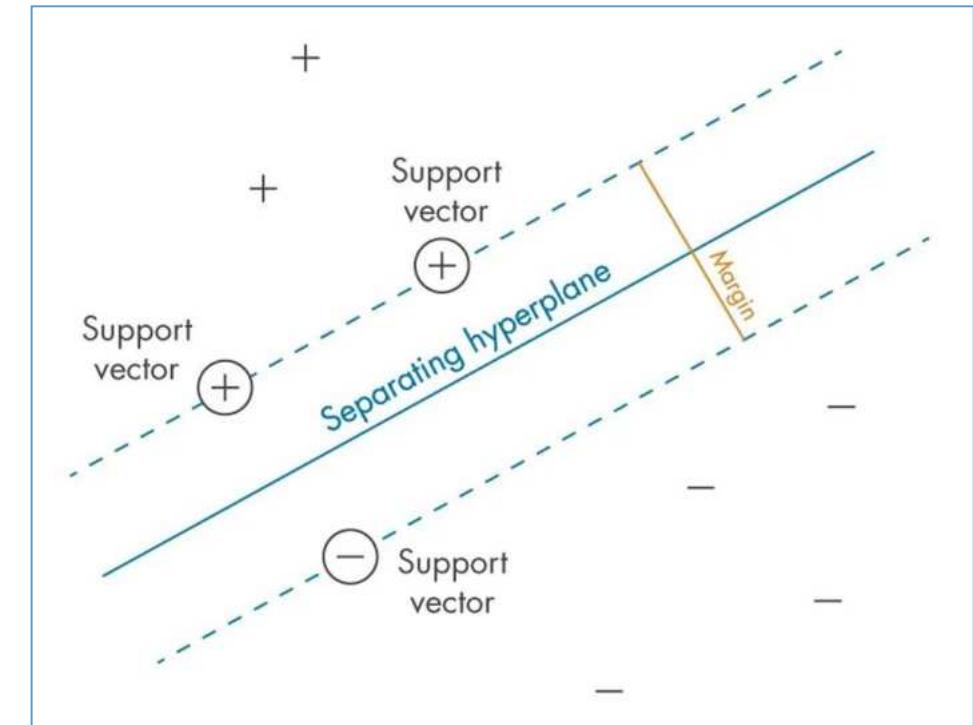
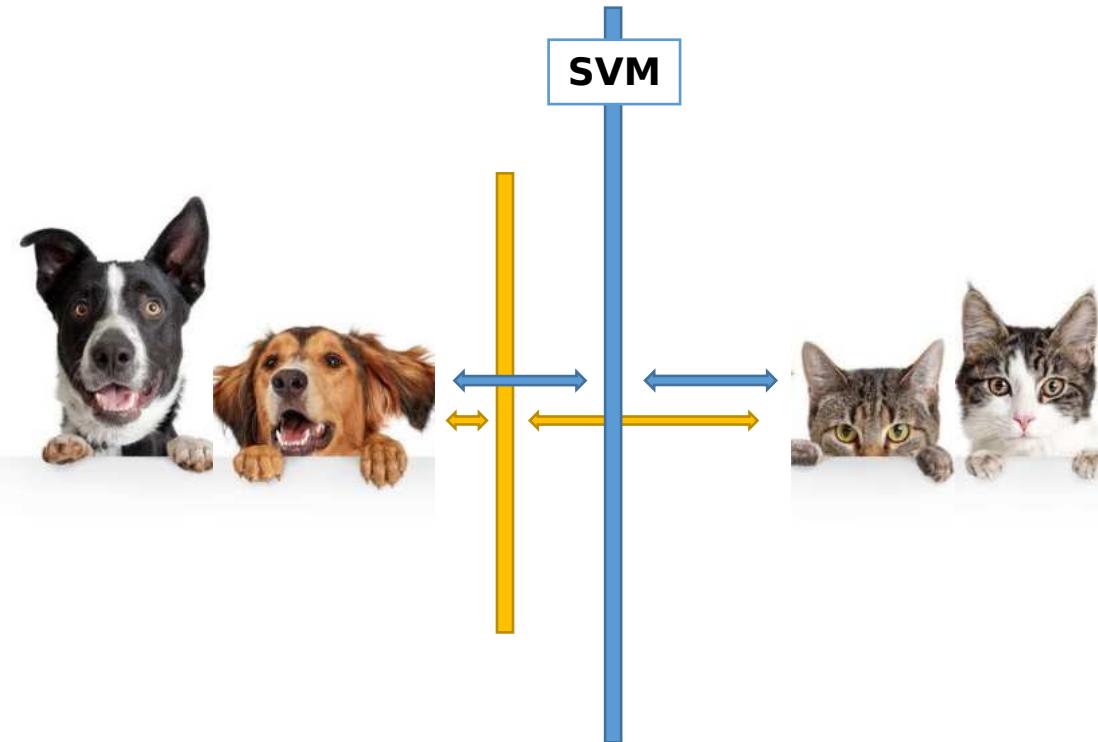


Non linear malumotlar bilan ishlash



F1 = qulqoq hajmi  
F2 = jun teksturasi





**SVM:** kichik va o'rta dataset lar bilan ishlash uchun qulay. Lekin katta dataset bilan ishlashda juda ko'p memory talab qiladi.

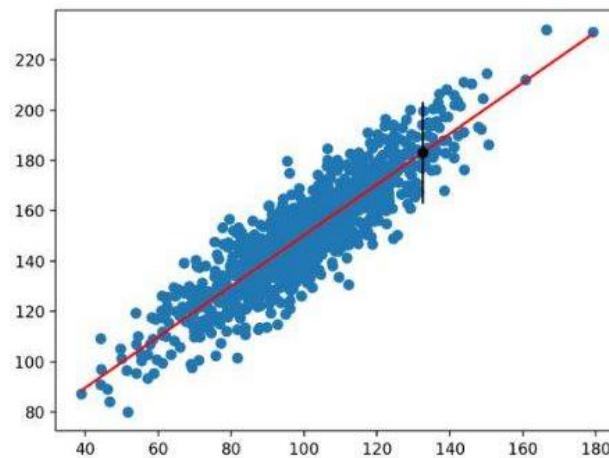
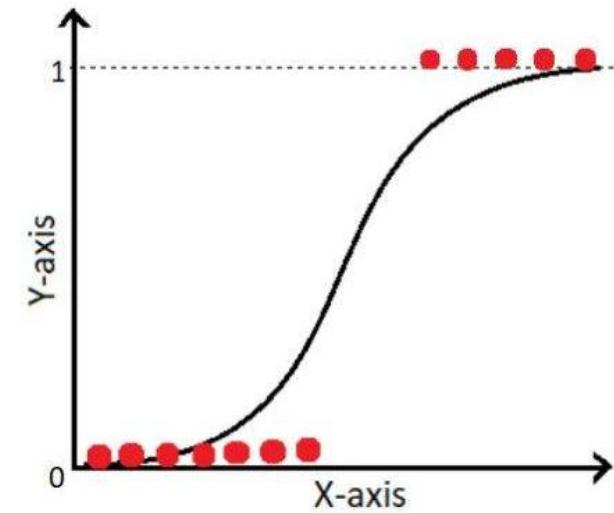




K-Nearest Neighbors (K-NN)

Support Vector Machines  
(SVM)Decision  
TreesRandom  
ForestLinear  
RegressionLogistic  
Regression

Naive bayes

**Linear Regression****VS****Logistic Regression**

**Linear Regression** va **Logistic Regression** algoritmlari SL algoritmlari guruhiga tegishli.



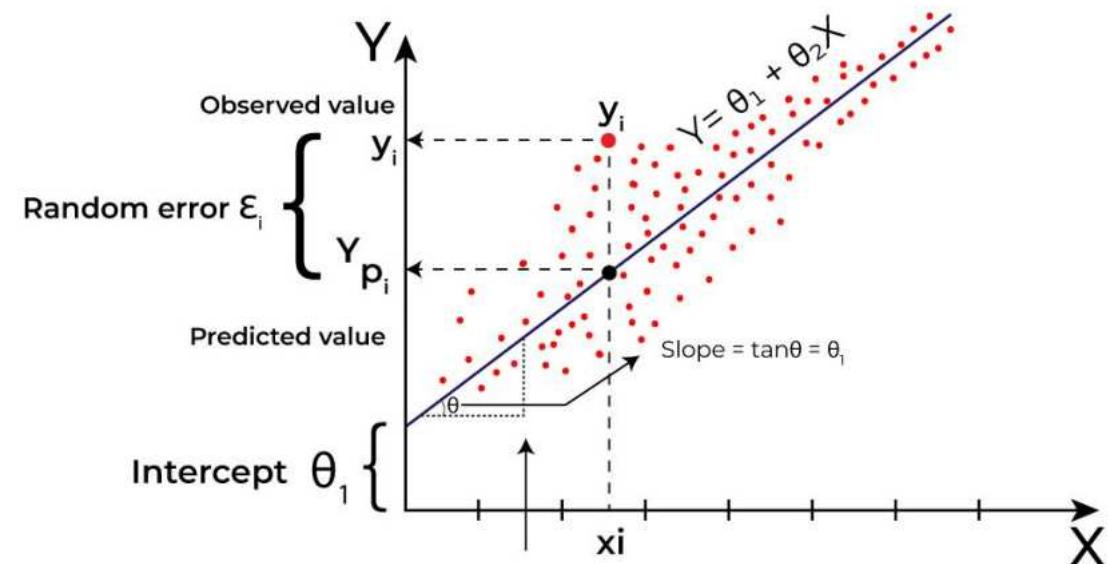
## Linear regression

Regressiya analizlari mashxur statistik metodlardan hisoblanadi. Ular bitta qaram o'zgaruvchi bilan bir yoki bir nechta qaram bo'limgan o'zgaruvchi o'rtaсидagi bog'liqlikni aniqlash uchun ishlatiladi.

$$y = b_0 + b_1 * x \longrightarrow \text{linear regression formularsi}$$

CV da, ramlar non-linear bo'lganligi uchun bu algoritm juda kam ishlatiladi. Ishlatilish o'rni ham Deep Learning (**DL**) modelini qurayotganda oxirgi layer da ishlatilishi mumkin. Bunda, arxitektura ichidagi rasm 2D dan 1D shaklga o'tgan bo'ladi, va oxirgi layerga davomli malumot ko'rinishiga keladi. Agar biz rasmdan biror qaramli to'plamni ajratib olmoqchi bo'sak, u yaxshi ishlashligi mumkin.

**Masalan:** Qizil rangli olmalarni pishganligini o'rtacha rangni 1 vektorgagi qiymatini hisoblash bilan pishgan yoki pishmagan olma klassifikatsiyasiga ajratish mumkin.



## Linear regression



Linear regression raqamli natijalarni bashorat qilishda ko'p ishlataladi.

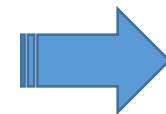
- Uy narxini hisoblash: Kv-m, xonalar soni, joyi, qavati,...

$$\text{uy narxi} = 0 \cdot \text{kv-m} + 1 \cdot \text{xonalar soni} + 2 \cdot \text{joyi} + 3 \cdot \text{qavati}$$



## Linear regression ishlatalish o'rnlari:

- Narxni bashorat qilishda
- Ob-havo bashoratida
- Moliyada (trading)
- ...



Umumiy olganda, bizdagi malumotlardan mano chiqara olsak, biz bu algoritmni qo'llashimiz mumkin. Lekin malumotlar shu algoritm klassifikatsiyasiga tushsa (Linear)



## Logistic regression

**Logistic regression** - Klassifikatsiya uchun keng qo'llaniladi. Bu algoritm ehtimollik darajasini aniqlab, kerakli sinfga taqsimlash uchun ishlataladi.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid funksiyasi formulası

Bu formula to'g'ri chiqiz hozil qilmaydi va chiqqan qiymatni  $[0,1]$  intervali oralig'ida hisoblab qaror qabul qiladi.



$$z = w_1x + b$$

$$P(\text{Pass}) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

x - o'qish vaqtি

y - qaror [0,1]

w<sub>1</sub>, b - model parametrlari

w<sub>1</sub> = 2,

b = -7

Hours	Results
1	0
2	0
3	0
4	1
5	1
6	1

$$z=2x-7$$

$$2 \times 1 - 7 = -5$$

$$2 \times 2 - 7 = -3$$

$$2 \times 3 - 7 = -1$$

$$2 \times 4 - 7 = 1$$

$$2 \times 5 - 7 = 3$$

$$2 \times 6 - 7 = 5$$

Sigmoid

0.0067

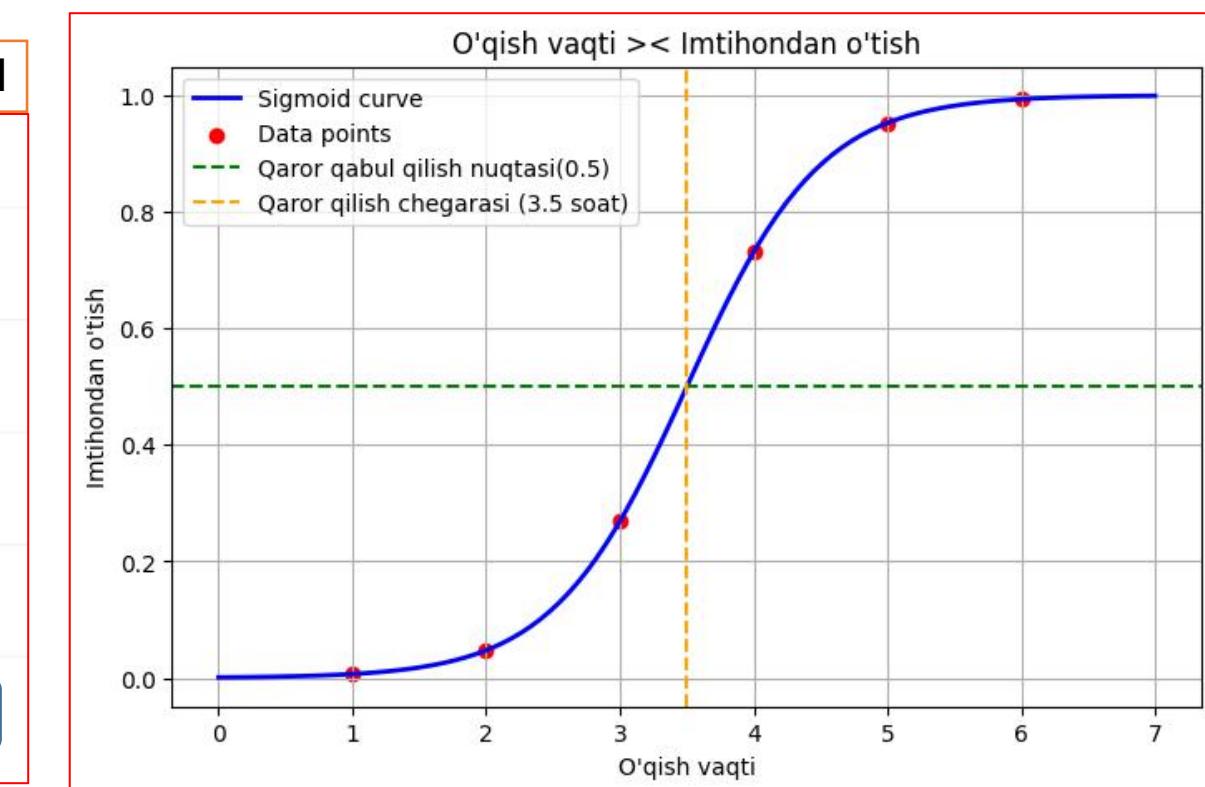
0.047

0.269

0.731

0.953

0.993



<https://www.desmos.com/calculator/c5jolbzsxp>

Ko'p vaqt o'qisa imtihonda o'tadi



~~K Nearest Neighbors (K-NN)~~

~~Support Vector Machines  
(SVM)~~

~~Decision  
Trees~~

~~Linear  
Regression~~

**Random  
Forest**

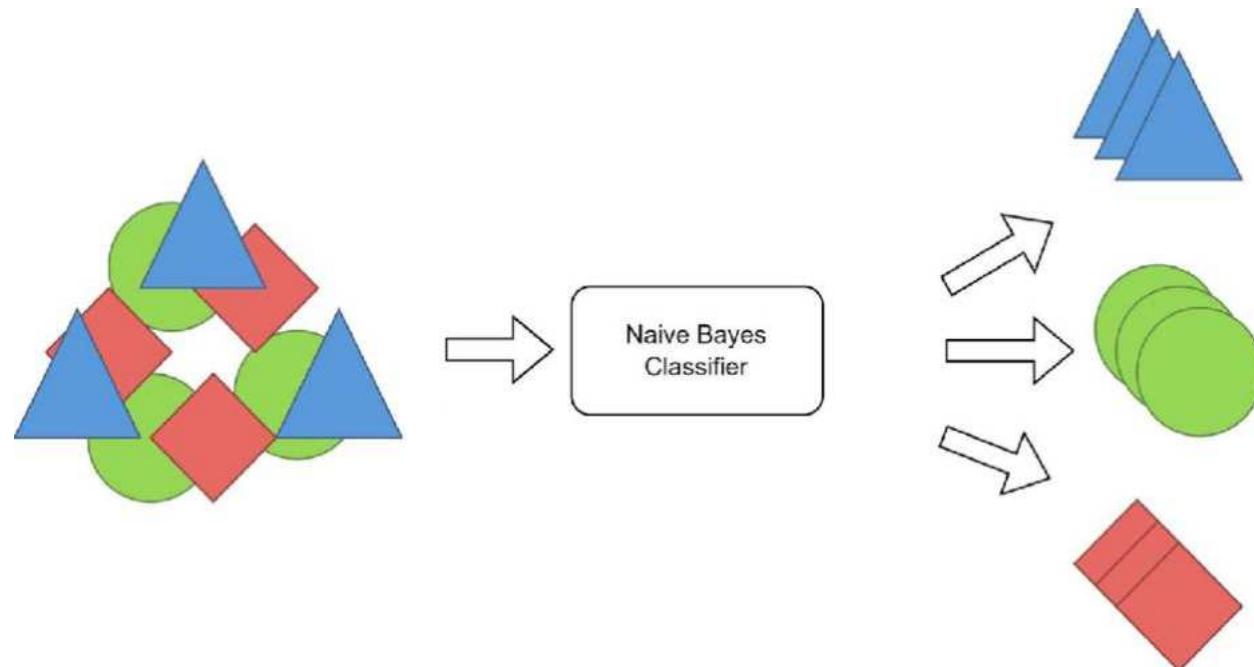
Naive bayes

Neural Networks

## Naive Bayes

**Naive Bayes** - bu algoritm Bayes Teoremasi monand ishlaydi. Bu ham **SL** algoritmlari guruhiga kiradi.

Bu algoritmning asosiy vazifasi klassifikatsiya qilishdir.



### Naive

malumotdagi bir-biriga qaram emas deb hisoblaydi

### Bayes

**Bayes Teoriyasini** ishlatgan holda malumotlaring tegishlilik ehtimolliklarini hisoblaydi

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$



$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

**C** = sınıf (guruh) (Misol: Email xabarni spam yoki non-spam deb filter qilinishi)

**X** = ahamiyatli qismlari (features) (Misol: email dagi so'zlar)

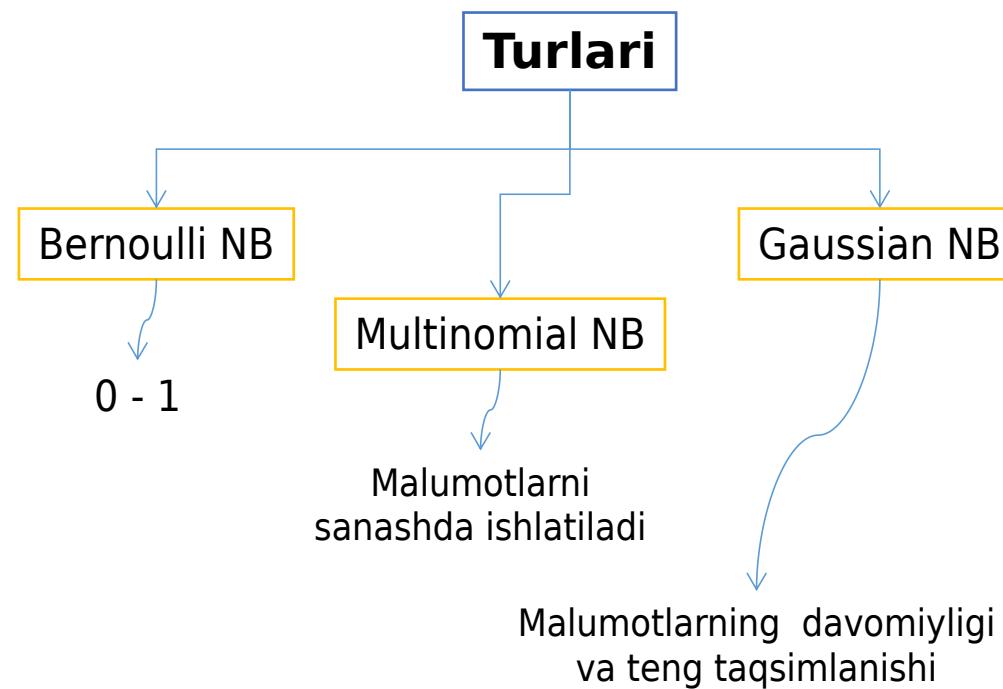
**P(C|X)** = **X** ni **C**-sinfiga (gruppasiga) tegishlilik ehtimolligi (probability)

**P(X|C)** = **X** dagi so'zlarning **C** sinfga tegishlilik ehtimolliklari

**P(C)** = C sinfining ehtimolligi

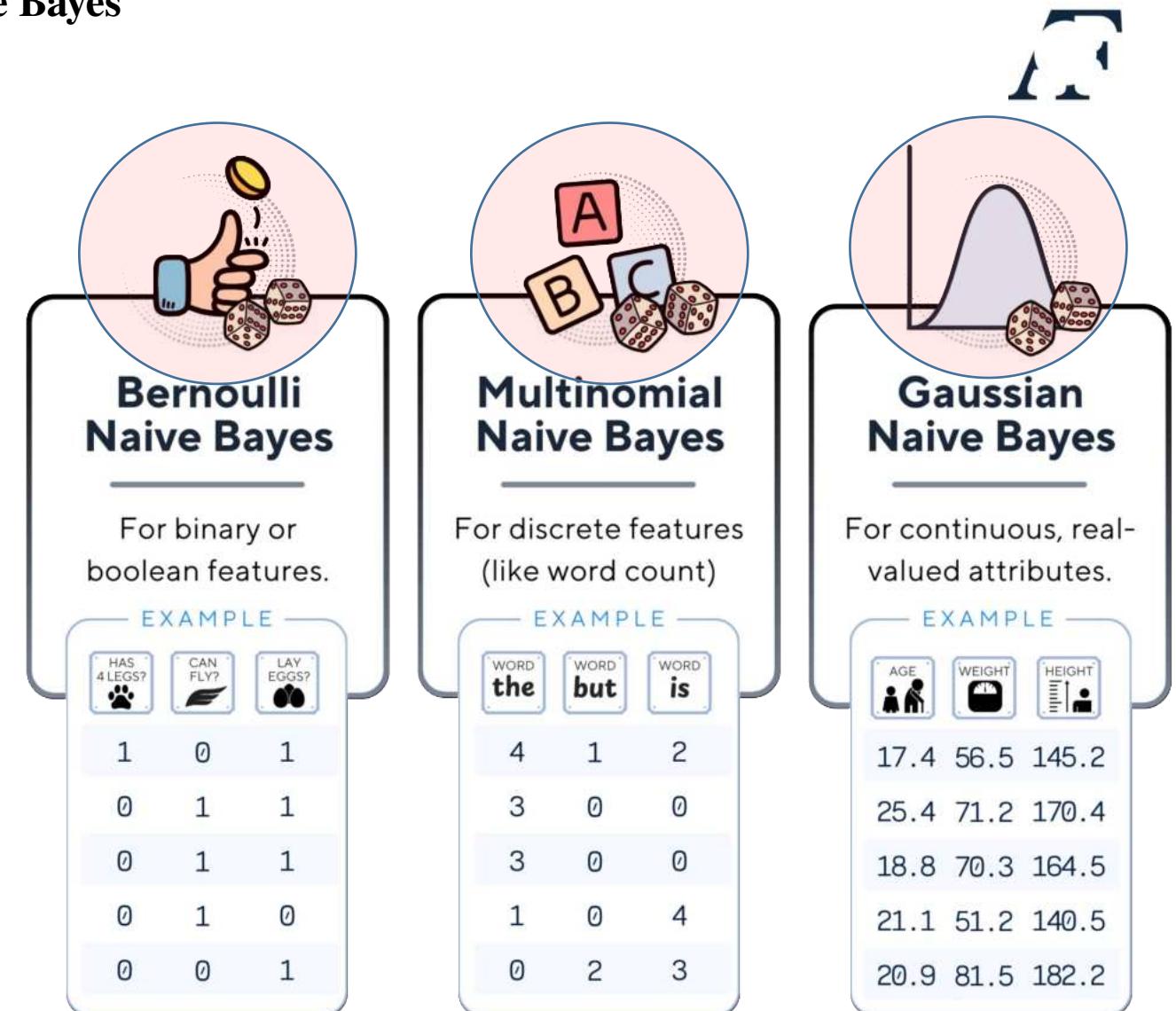
**P(X)** = so'zlarning ehtimolligi

## Naive Bayes

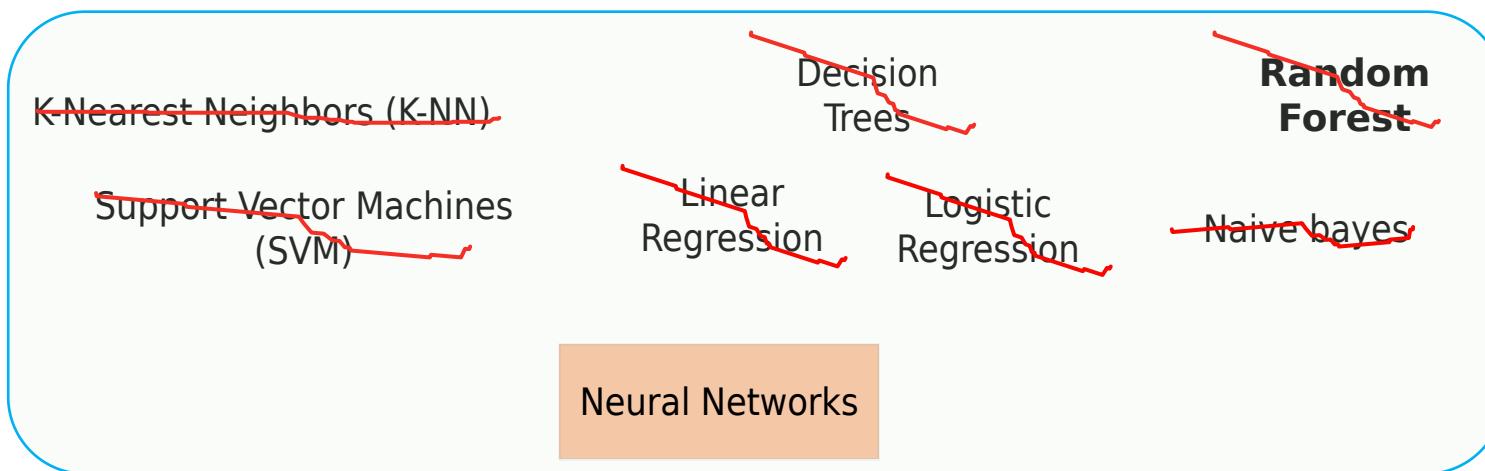


Rasm klassikatasiyida pixellarni bir-biriga bog'liqli ekanligi muhim. Bu klassifikatsiya uchun yordam beradi. NB har bir pixelni mustaqil deb hisoblab olganligi uchun, bog'liqlilikni bog'lolmaydi va aniqliligi nisbatan past bo'ladi.

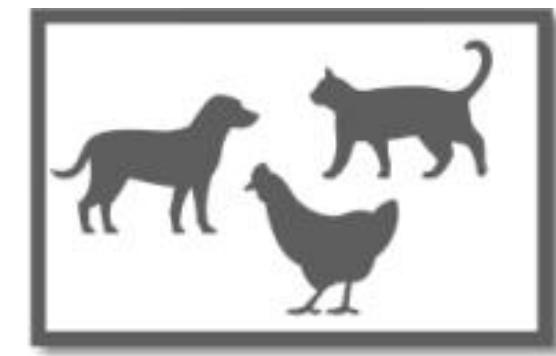
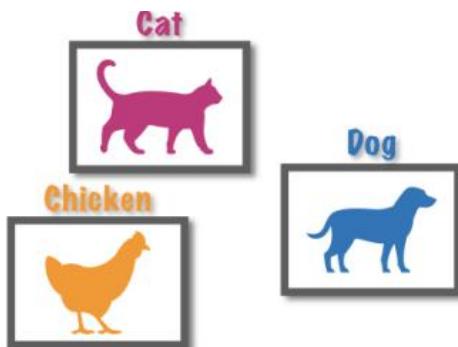
[https://github.com/fartech00/Prompt-refining/blob/main/TAFU\\_KNN.ipynb](https://github.com/fartech00/Prompt-refining/blob/main/TAFU_KNN.ipynb)



<https://towardsdatascience.com/bernoulli-naive-bayes-explained-a-visual-guide-with-code-examples-for-beginners-aec39771ddd6/>

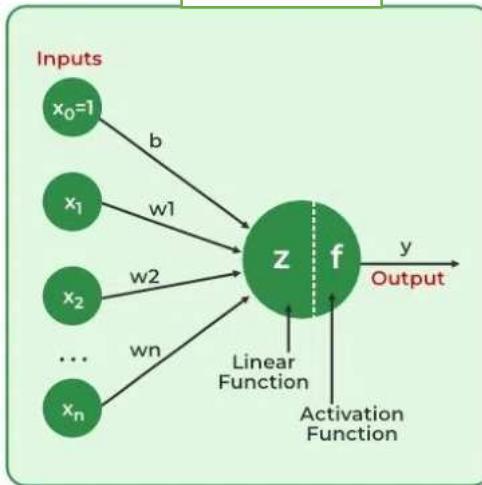
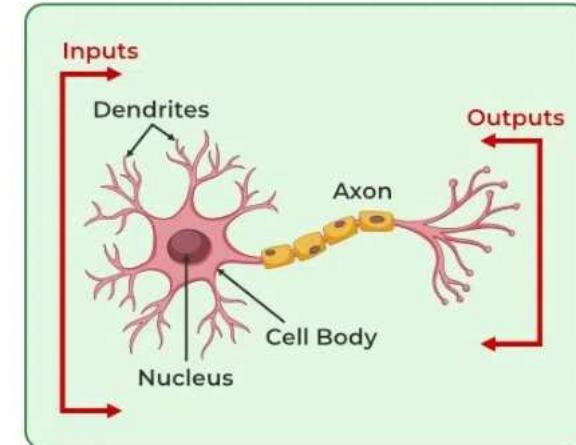
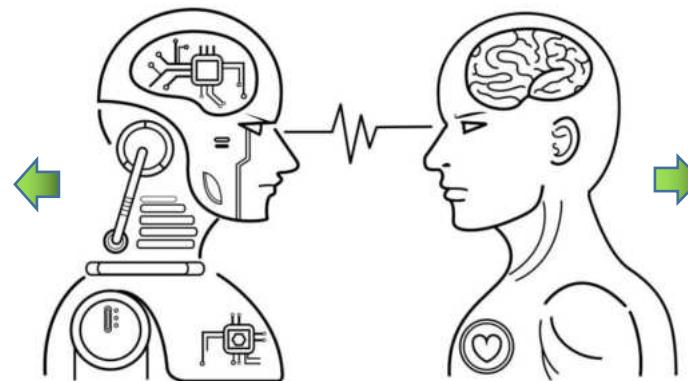


Supervised learning ← → Unsupervised learning

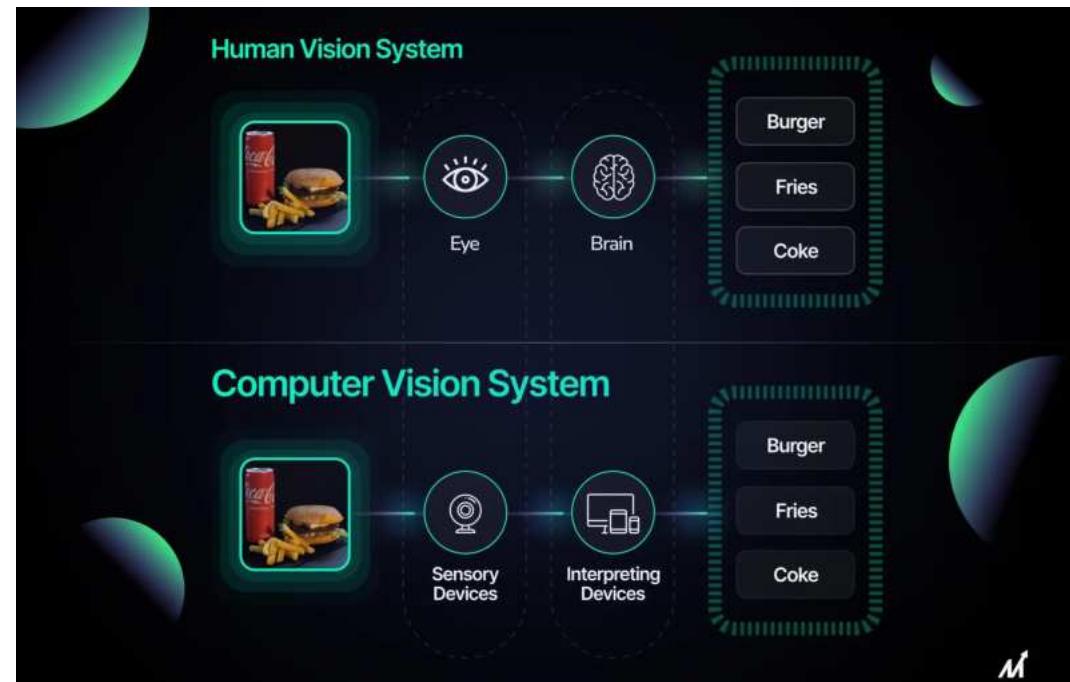


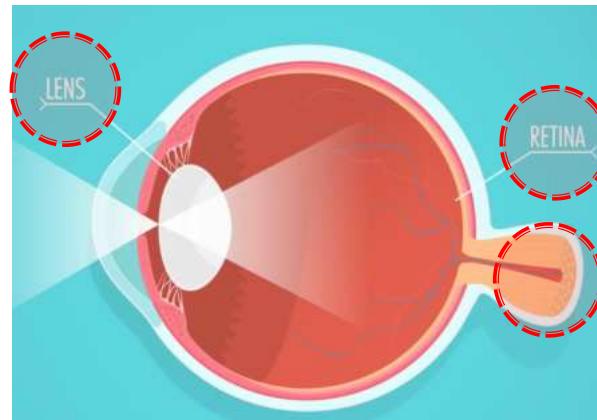


## Perceptron

NN haqida umumiyl  
tushinchalar

Lekin, NN strukturasi aynan  
inson miyasini nusxasi emas.



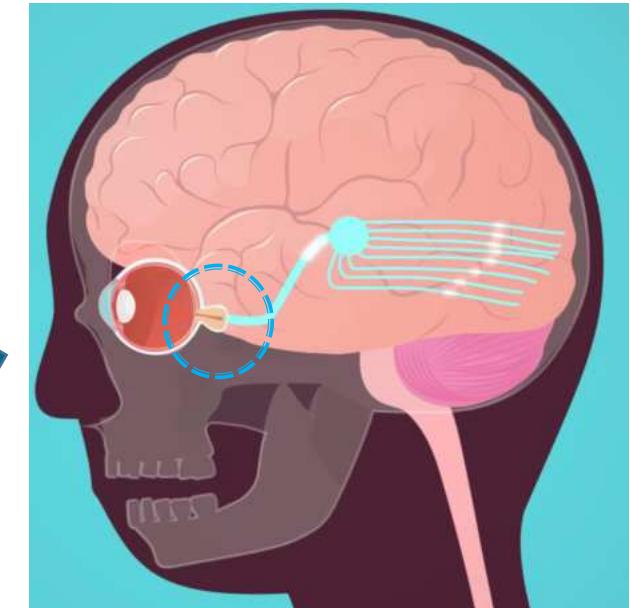
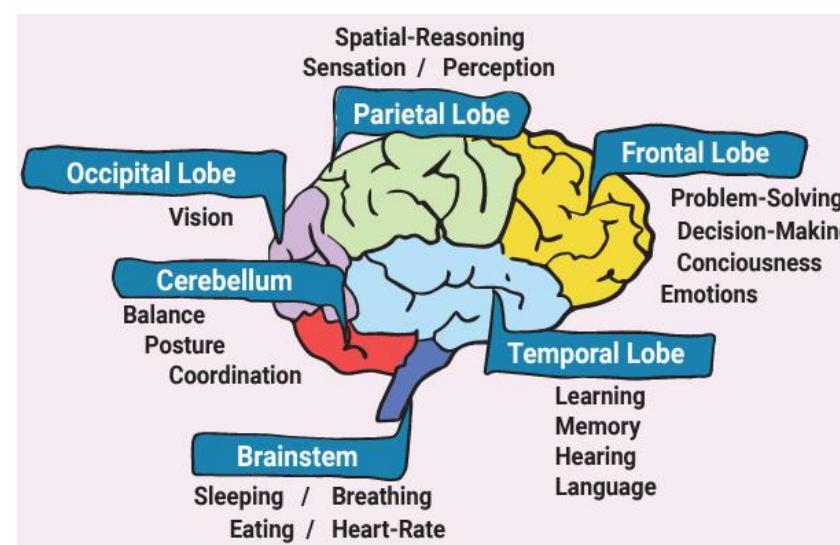
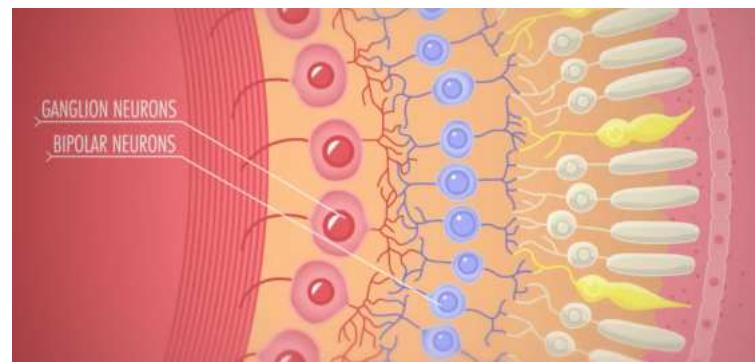


Insonni malumotlarni **ko'z** orqali qabul qilishi.

Malumotlar kelish kanallari:

Ko'z / Qulqoq / Burun / Tam bilish /  
Va Tanadagi har qanday nervli azo.

**Ko'z misolida qarasak**

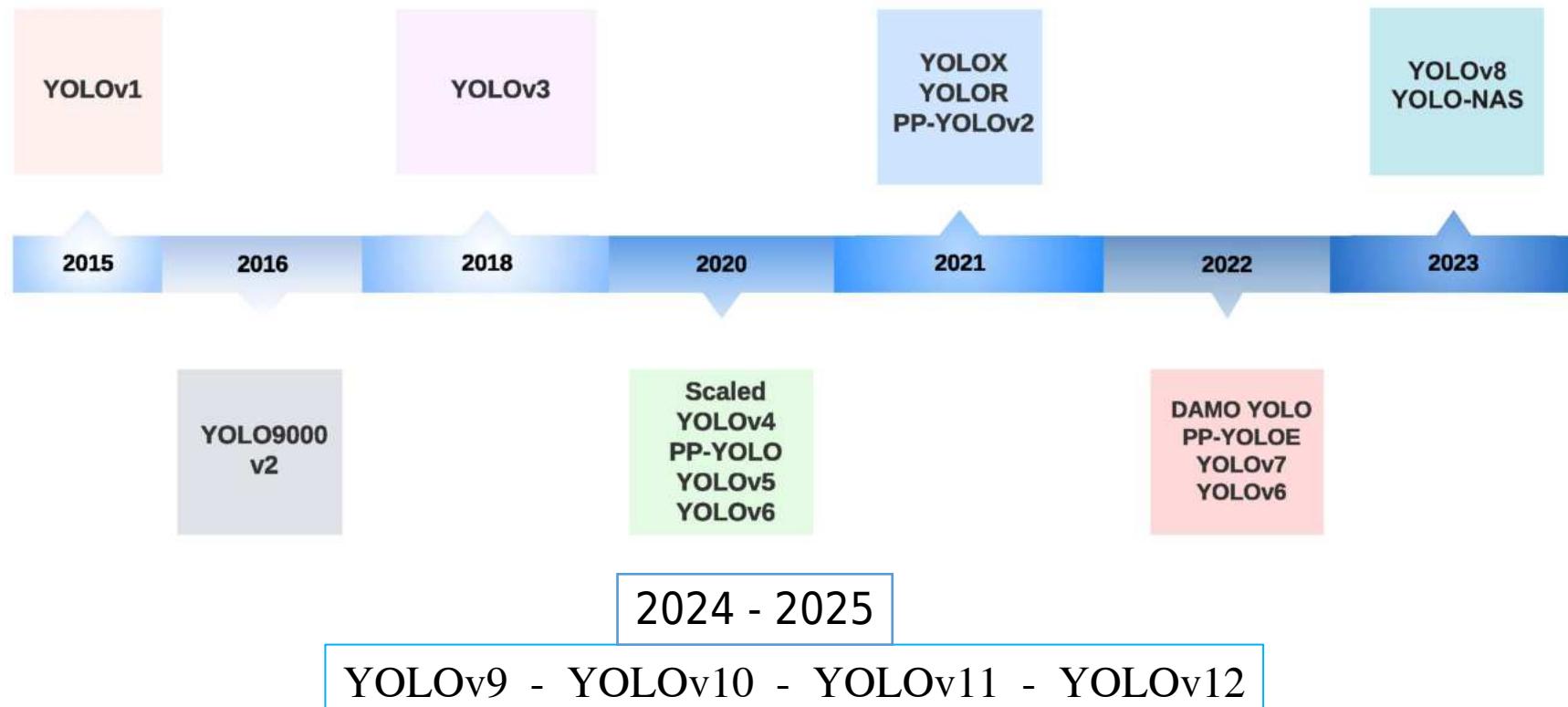


## CV arxitexturalari NN dan iborat



CV arxitexturalari sodda NN arxitekturasidan murakkab arxitekturalar ko'rinishga keldi

2012 - AlexNet,  
 2014 - VGGNet,  
 2014 - GoogLeNet / Inception (Google),  
 2014 - R-CNN,  
 2015 - Fast R-CNN,  
 2015 - ResNet (Microsoft),  
 2016 - DenseNet,  
 2016 - YOLO,  
 2017 - RetinaNet (Facebook AI)  
 2017 - YOLOv2  
 2017 - Mask R-CNN (Facebook AI)  
 2017 - SENet,  
 2018 - Detectron (Facebook AI research),  
 2018 - YOLOv3  
 2019 - Detectron2,  
 2019 - EfficientDet (Google brain)  
 2020 - Vision Transformer (ViT, Google),  
 2020 - YOLOv4,  
 2021 - YOLOv5 (Ultralytics),  
 2021 - Swin Transformer,  
 ...

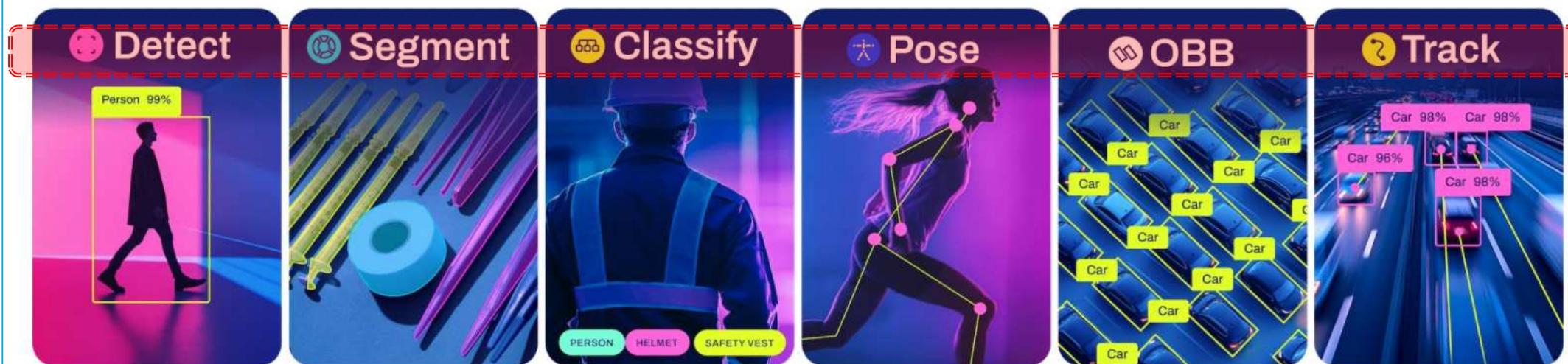




## YOLO - You Only Look Once

### Models

Ultralytics supports a wide range of YOLO models, from early versions like [YOLOv3](#) to the latest [YOLO11](#). The tables below showcase YOLO11 models pretrained on the [COCO](#) dataset for [Detection](#), [Segmentation](#), and [Pose Estimation](#). Additionally, [Classification](#) models pretrained on the [ImageNet](#) dataset are available. [Tracking](#) mode is compatible with all Detection, Segmentation, and Pose models. All [Models](#) are automatically downloaded from the latest Ultralytics [release](#) upon first use.

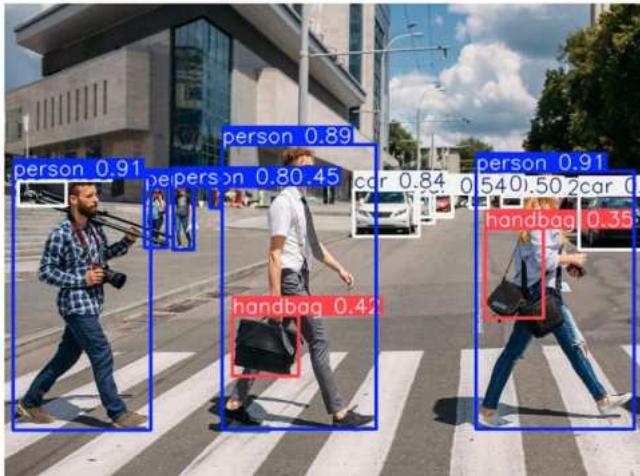


<https://youtu.be/L23oIHZE14w?si=mDSvmpOys-czcQ3e>

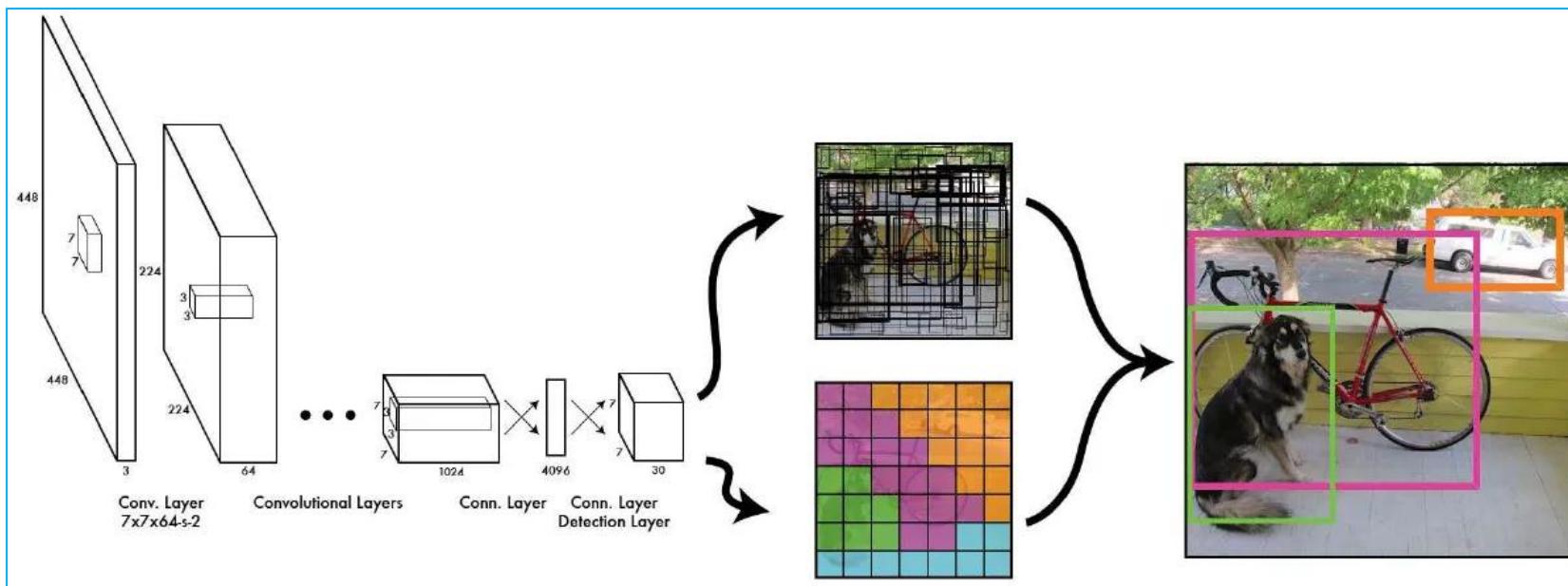
<https://youtu.be/neBZ6huolkg?si=ajt0rNjKCyQXOPQX>



## Object detection - OD



**Object detection** (Obyektni aniqlash) - bu rasmida qanday obyektlar borligini va lokatsiyasini aniqlash.



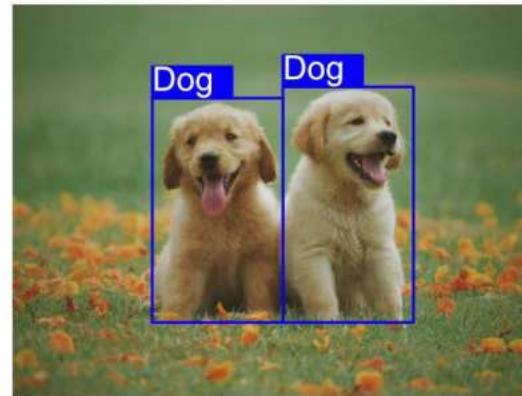


Klassifikatsiyada algoritm ramdagi malumotni o'rganib, undagi obyektlarni sinflariga bo'lsa,  
Obyektni aniqlashda algoritm o'sha obyektlar qaysi nuqtalarda joylashganligini ham topadi.

## Classification



## Detection



Demak, **OD** da 2 ta jarayon sodir bo'ladi.

- 1: Bu qanaqa obyekt ?: (daraxt, quyosh, bulut, qushlar, vertalyot..)
- 2: Obyektlar qayerda joylashgan? : ( $x_1, y_1 ; x_2, y_2$ ) kordinata o'qlari bo'ylab **bounding box** ga oladi



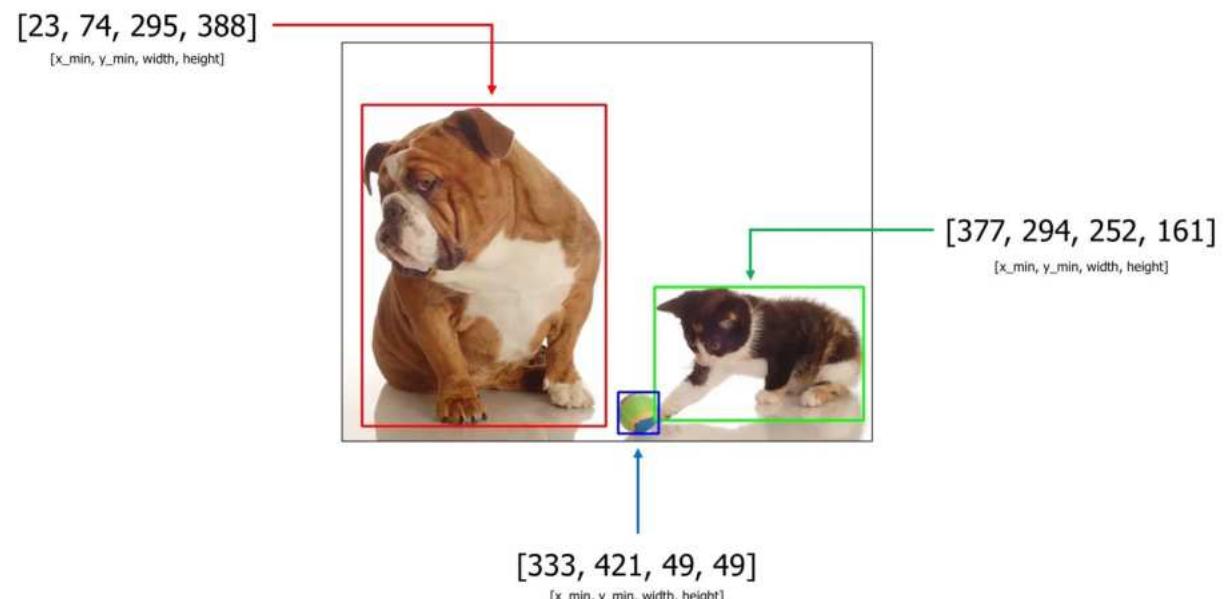
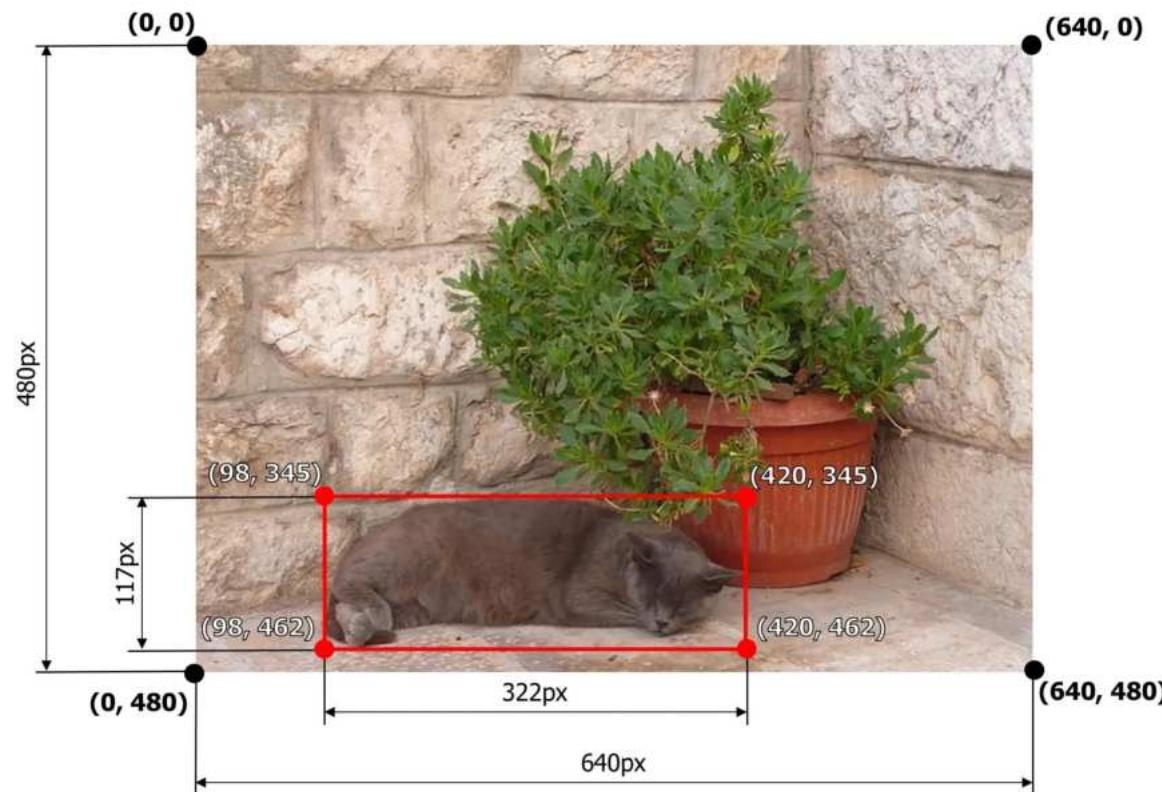
## OD ishlash jarayonda:

Rasm

CNN arxitekturalari rasmdan muhim, ahamiyatli qismlarni ajratib oldi

Obyeklatni bilish, **tanish ilmiga asoslanib** sinflarga (guruhlarga) ajratdi

Bounding box koordinata o'qiga moslab chiqizdi

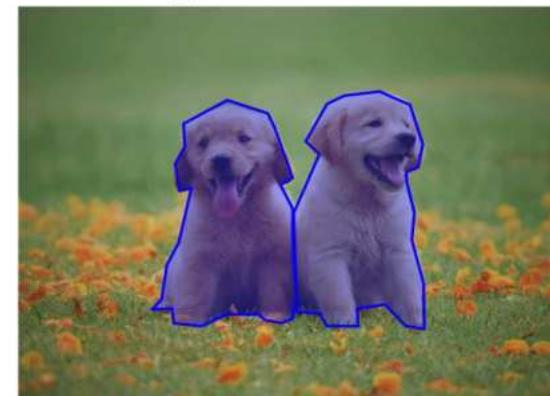




2 ta kuchukcha o'tloqda o'tiribdi

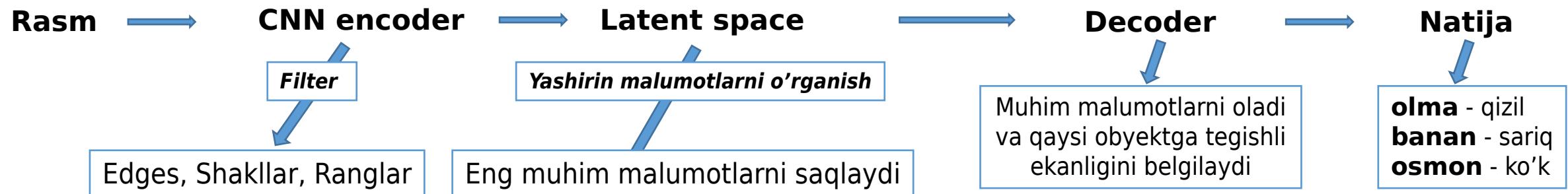
'2' - 'kuchukcha' - 'o'tloq' - 'o'tiribdi'

## Segmentation



**Segmentation** - bunda rasmdagi pixellar qaysi aniqlangan obyektlarga tegishli ekanligi topishdir.

Demak - pixellarni klassifikatsiya qilish bilan ishlaydi.



rasmni razmeri qisqarib boradi ammo - muhim malumotlar ajralib, yig'ilib ko'payib boradi



## 11M images, 1B+ masks

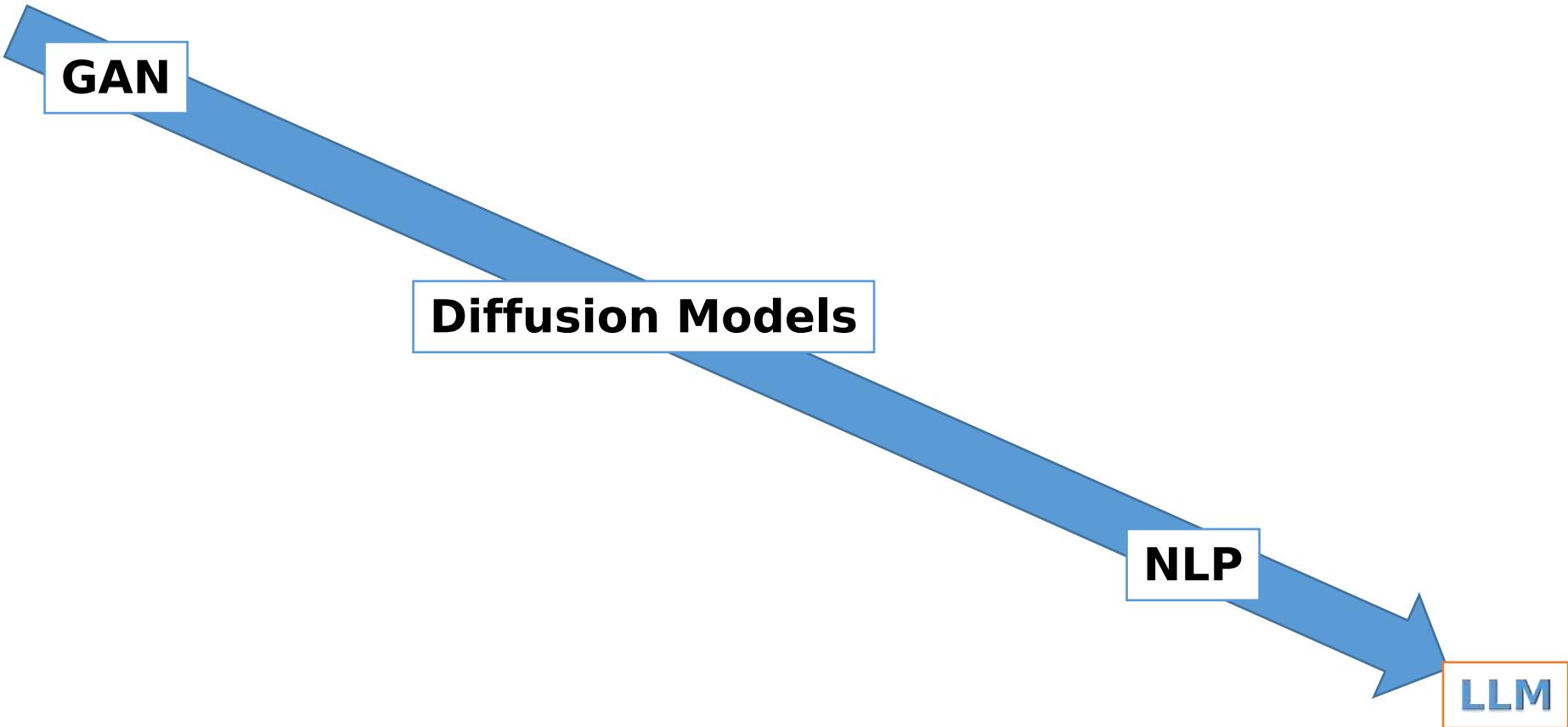
After annotating enough masks with SAM's help, we were able to leverage SAM's sophisticated ambiguity-aware design to annotate new images fully automatically. To do this, we present SAM with a grid of points on an image and ask SAM to segment everything at each point. Our final dataset includes more than 1.1 billion segmentation masks collected on ~11 million licensed and privacy preserving images.

 [Explore the dataset](#)

 [Download full dataset](#)

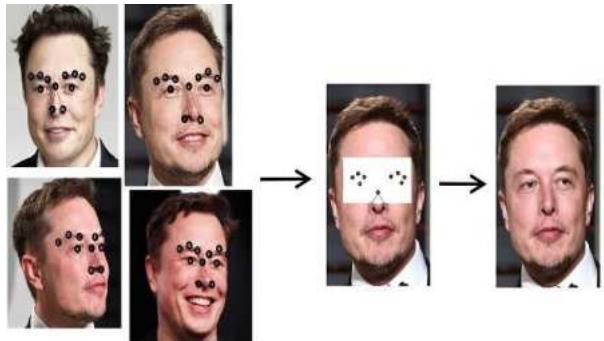


<https://segment-anything.com/>



## GANs

GAN - bu turdagi NN arxitekturalari haqiqiyga o'xshagan rasm, ovoz, so'z yasay oladi.



Zebras ↘ Horses



zebra → horse



horse → zebra

<https://github.com/eriklindernoren/PyTorch-GAN>



Input  
*From real life*

HiFaceGAN  
*ACMMM 20*

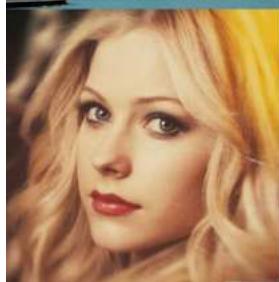
DFDNet  
*ECCV 20*

Wan *et al.*  
*CVPR 20*

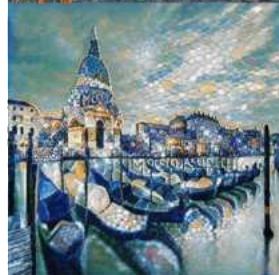
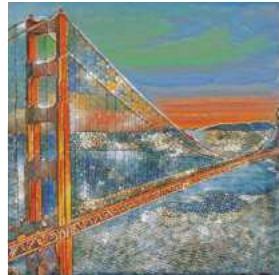
PULSE  
*CVPR 20*

GFP-GAN  
*Ours*

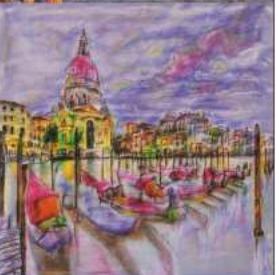
Content Image



“Mosaic”



“A sketch with crayon”



“White wool”

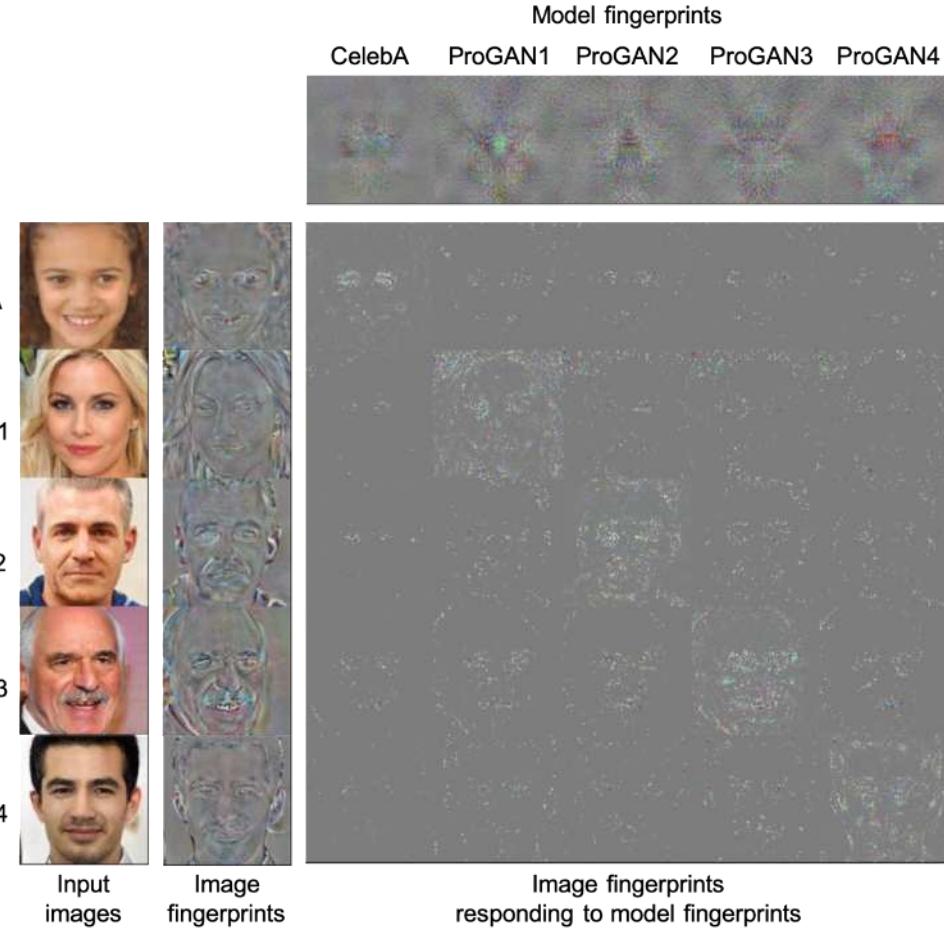
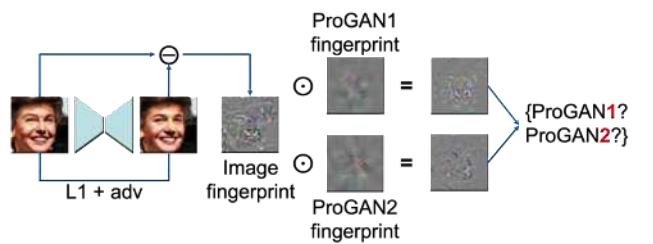
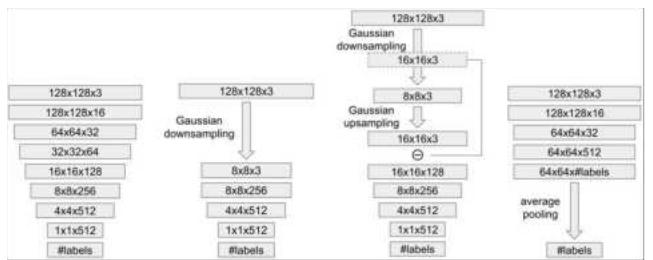
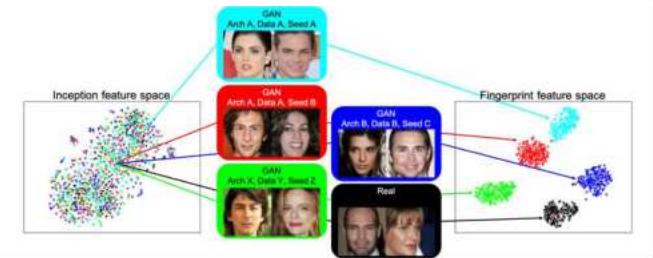


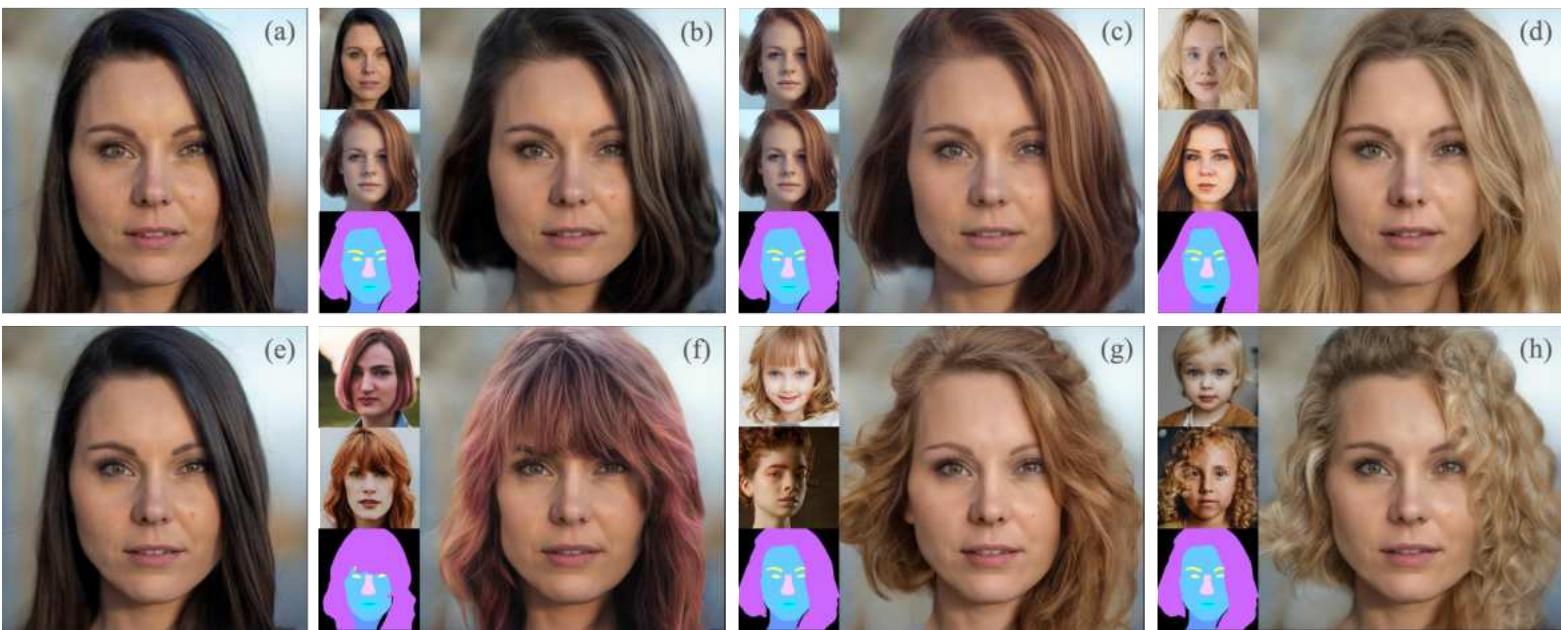
“Green crystal”



“A graffiti style painting”





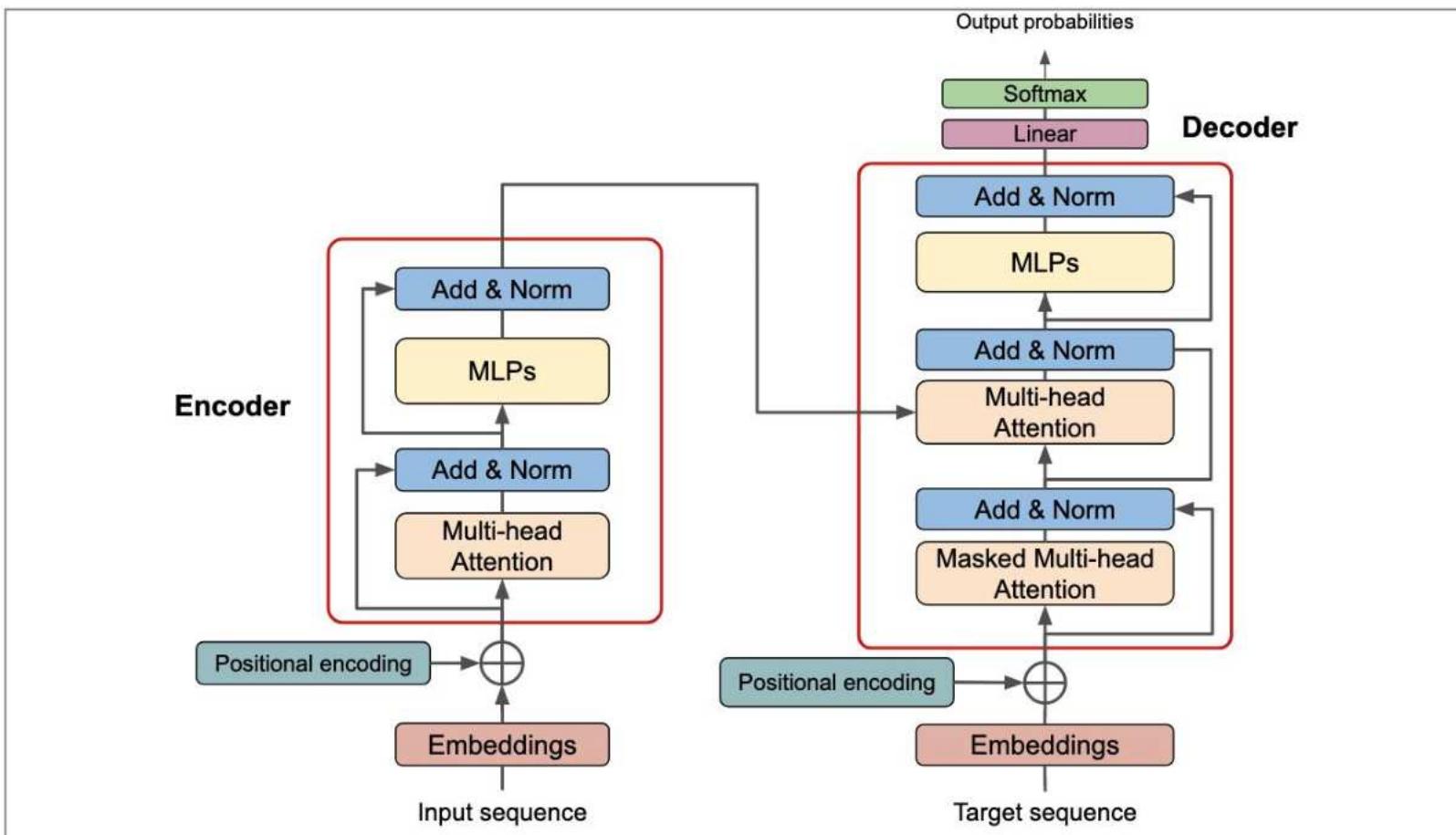


<https://youtu.be/-SQr6uucDTk>

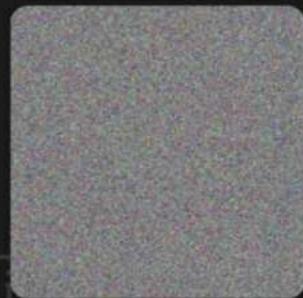
<https://github.com/sail-sg/EditAnything>

<https://youtu.be/Q463j1tOWP8?si=Y63wO54-RDqEJlfo>

LLM

[https://youtu.be/iv-5mZ\\_9CPY?si=DUOuonJ9Ng2PXqqQ](https://youtu.be/iv-5mZ_9CPY?si=DUOuonJ9Ng2PXqqQ)



**Algorithm 1** Training

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
     
$$\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$$

6: until converged
  
```

 $x_0$ 

Clean Image

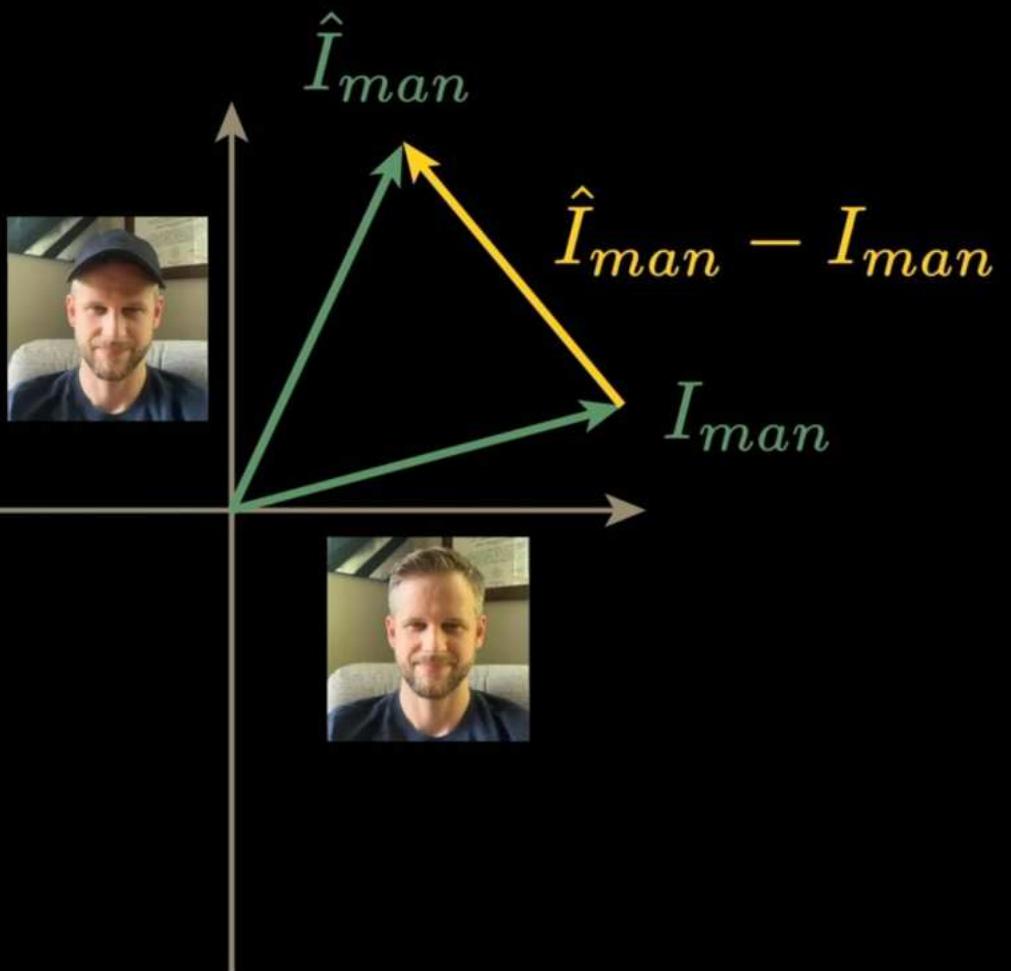
**Total Added Noise**

(MODEL LEARNS TO PREDICT THIS)

**Model Output****Algorithm 2** Sampling

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
  
```



TOP MATCHES	
hat	0.165
cap	0.113
helmet	0.106
angry	0.061
exercising	0.06

**Algorithm 1** Training

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
      
$$\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}, t)\|^2$$

6: until converged
  
```

**Algorithm 2** Sampling

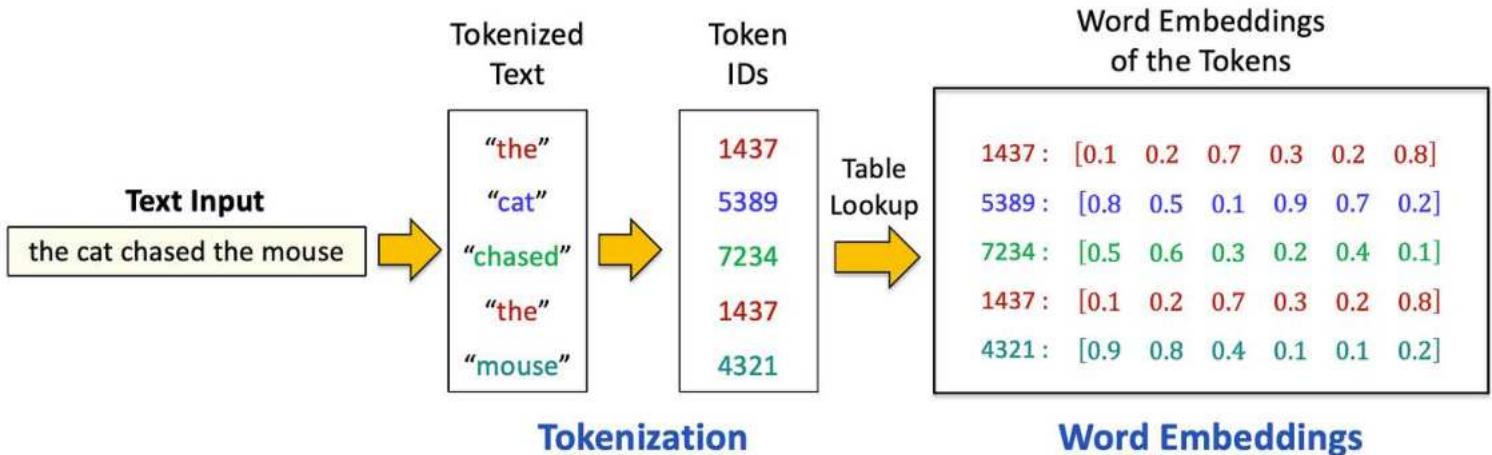
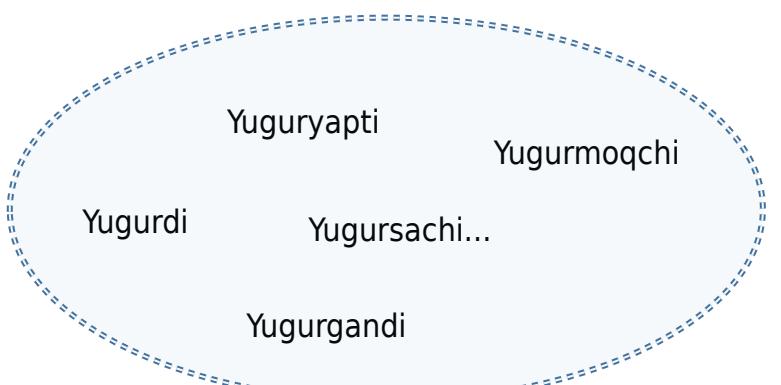
```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: Output  $\mathbf{x}_0$  Scaled Model Output Random Noise Image
  
```

**Word2Vec****Tokenization****Embedding**So'zlarni songa aylanishi (**1D** - 1 vektorli bo'lishi)<https://youtu.be/viZrOnJcIY0?si=YhSgpjpQ9NLUpBAA>

## Lemmanization

### Yugur



### Stemming

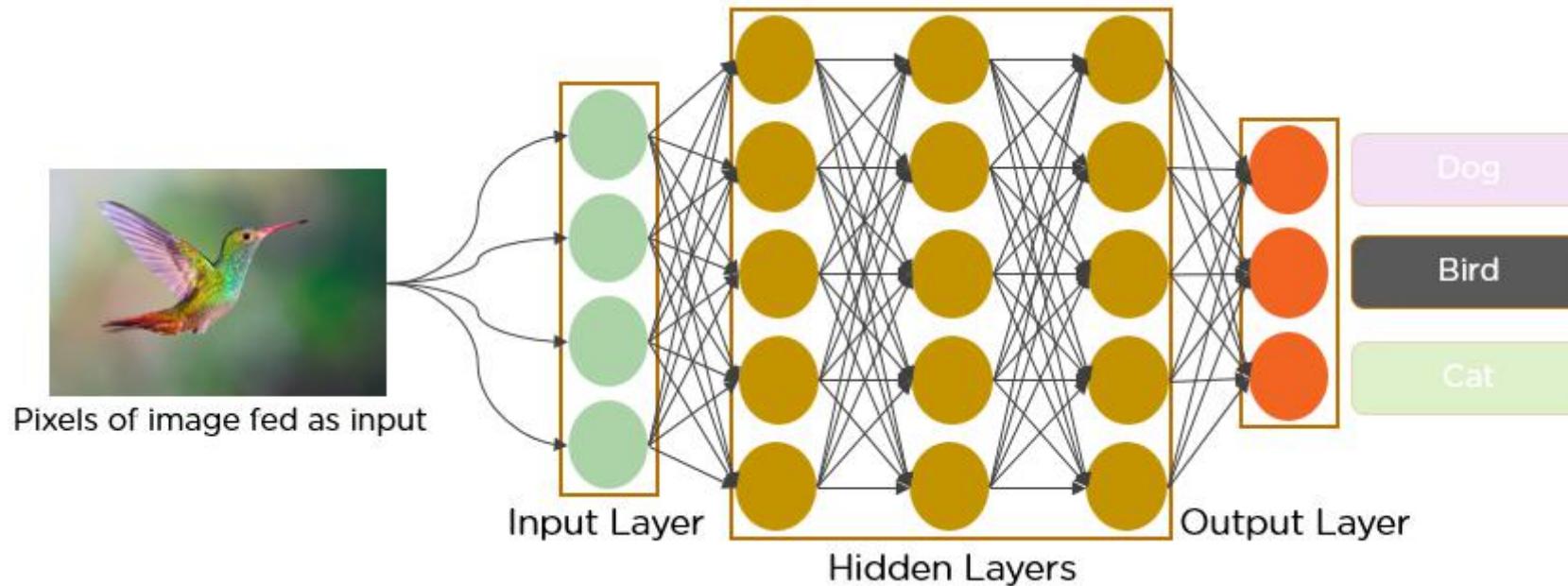
adjustable → adjust  
formality → formaliti  
formaliti → formal  
airliner → airlin

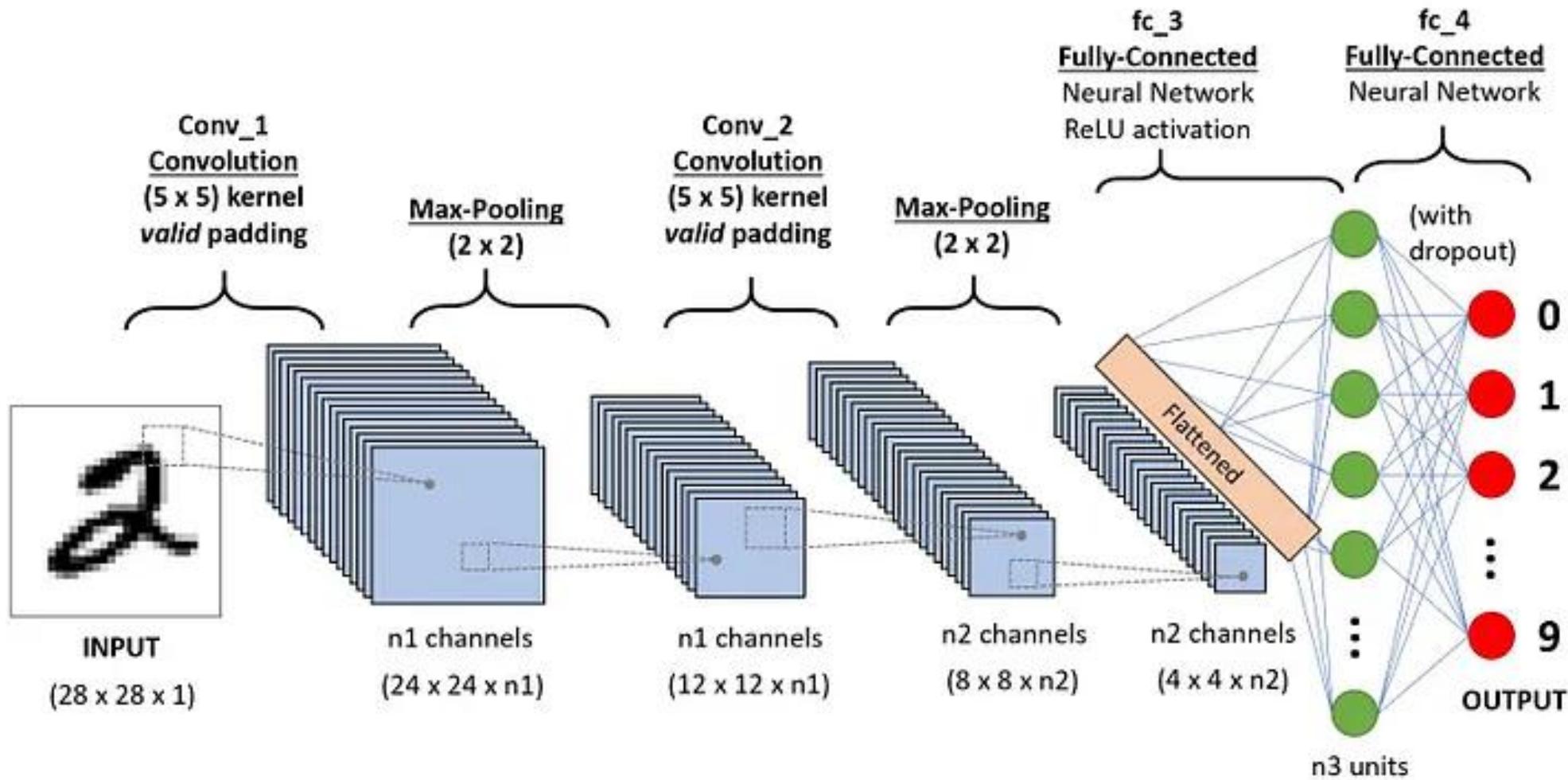
### Lemmatization

was → (to) be  
better → good  
meeting → meeting

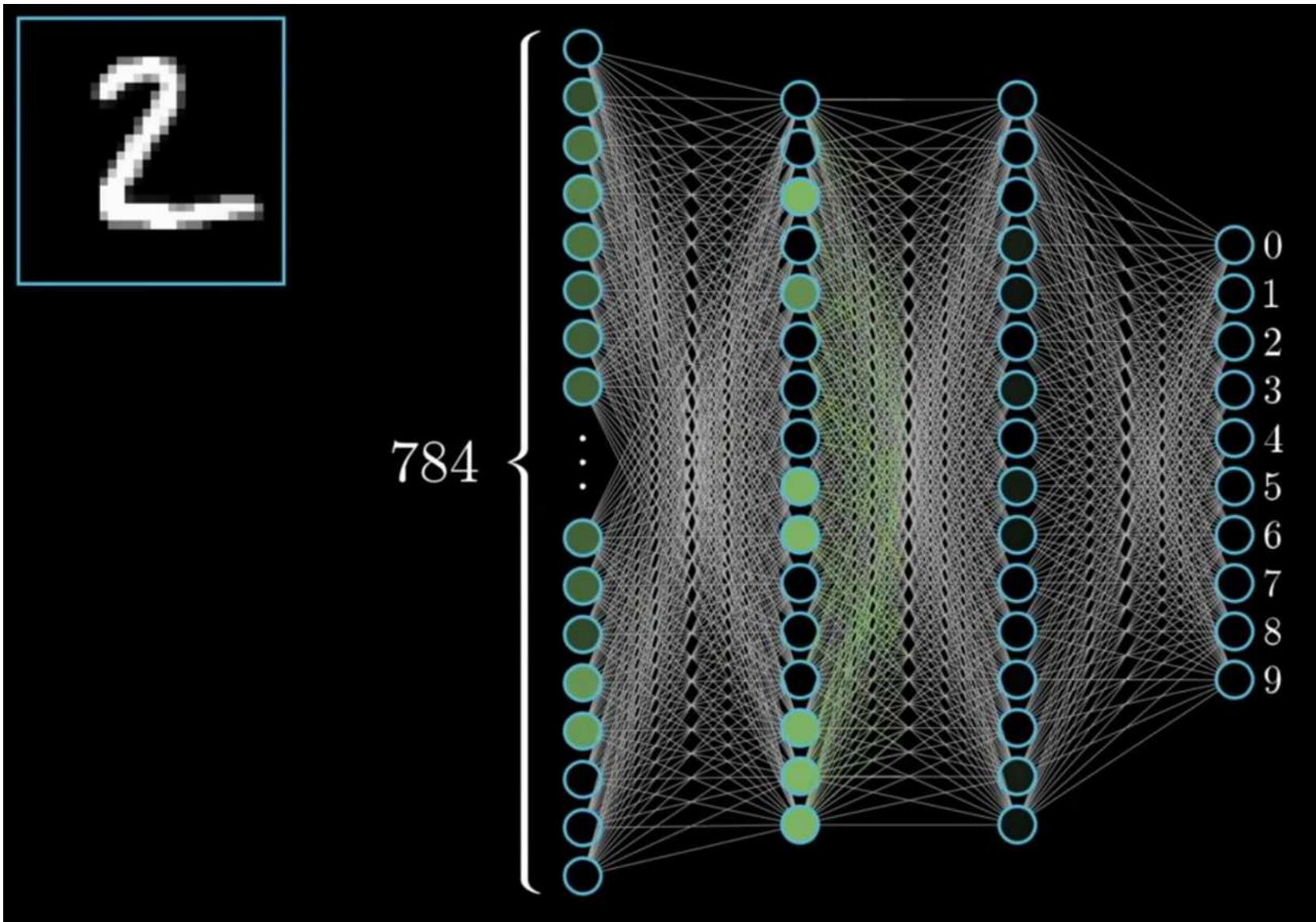
Yuqorida ko'rilgan malumotlar almashinuvi asosan  
**NN** arxitekturasida ishlaydi,

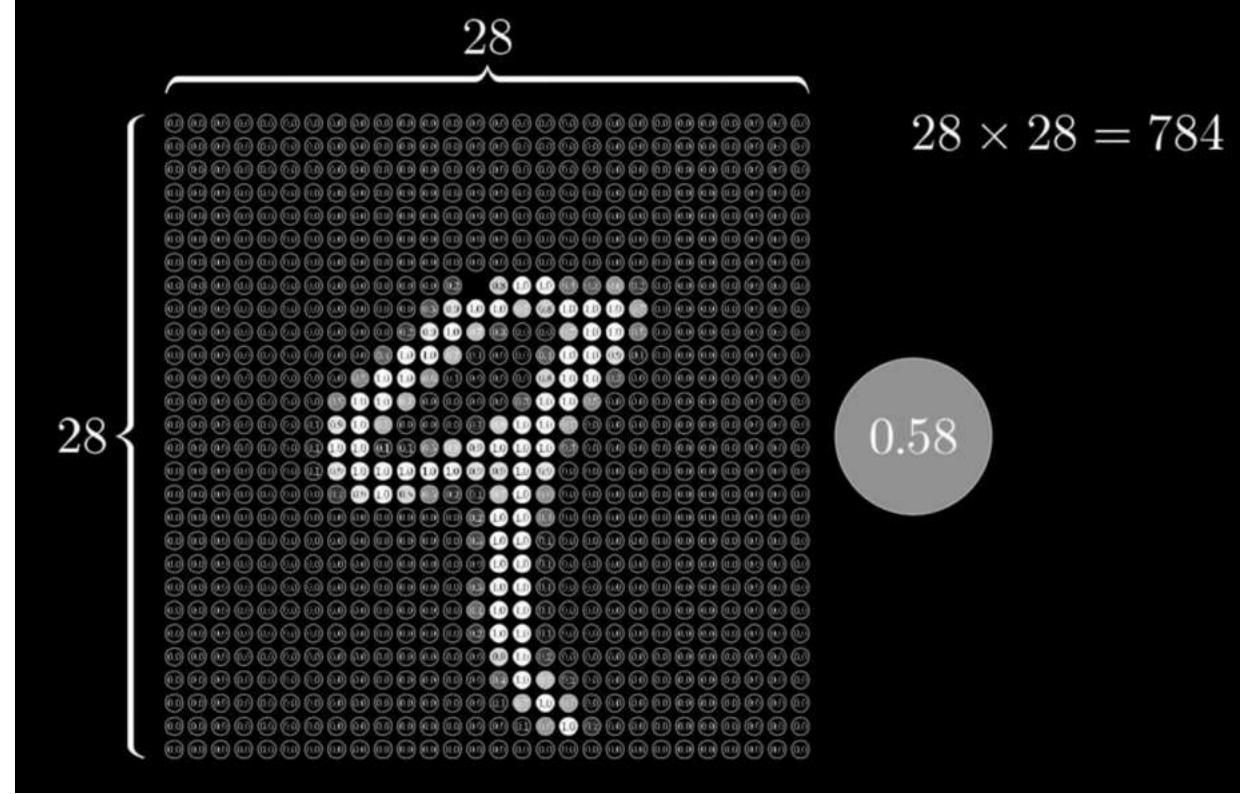
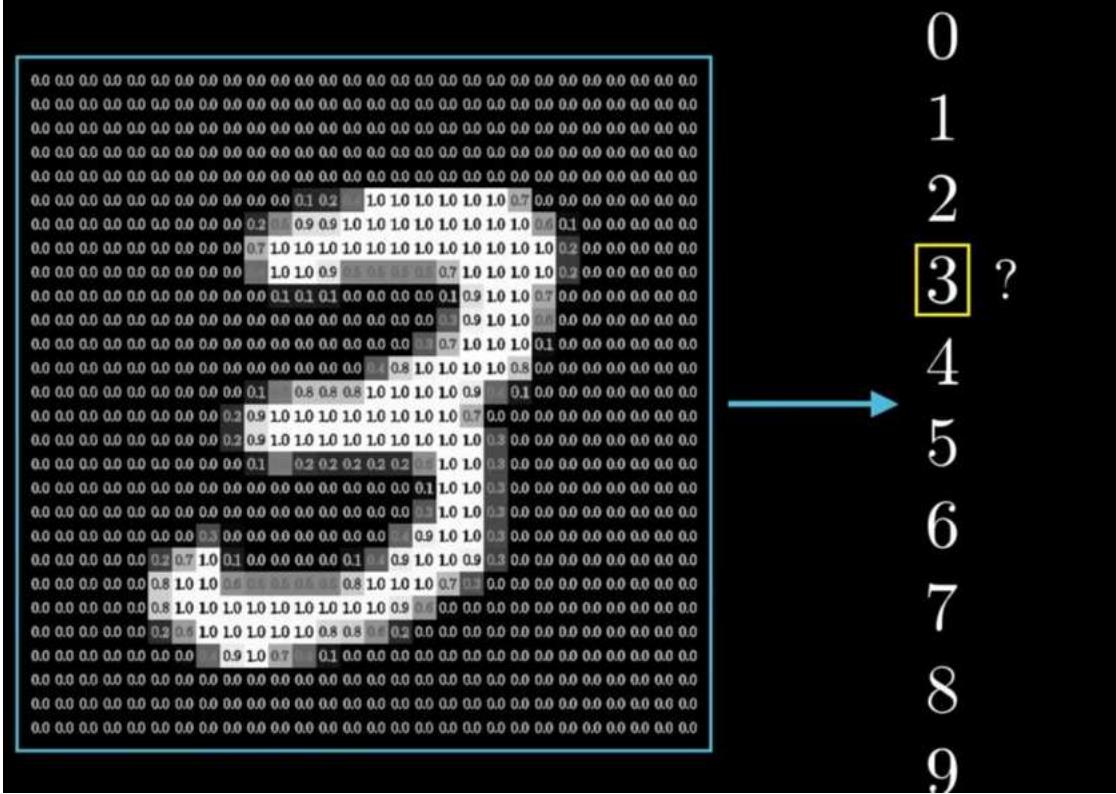
**NN** o'zi qanday ishlaydi?





$$28 \times 28 = 784$$



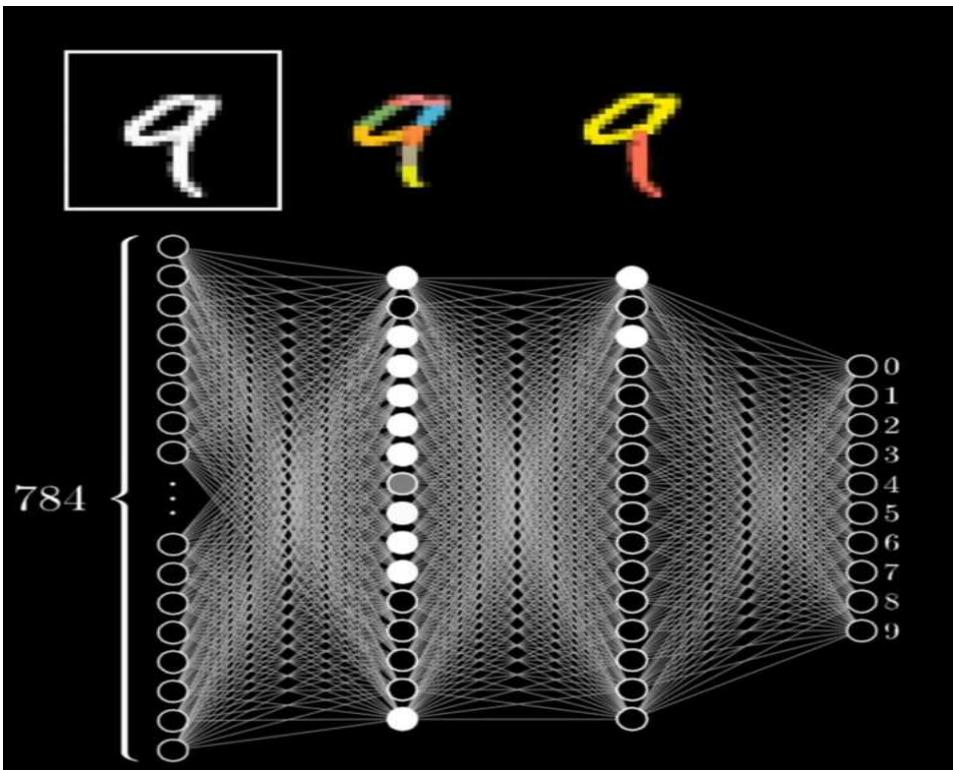
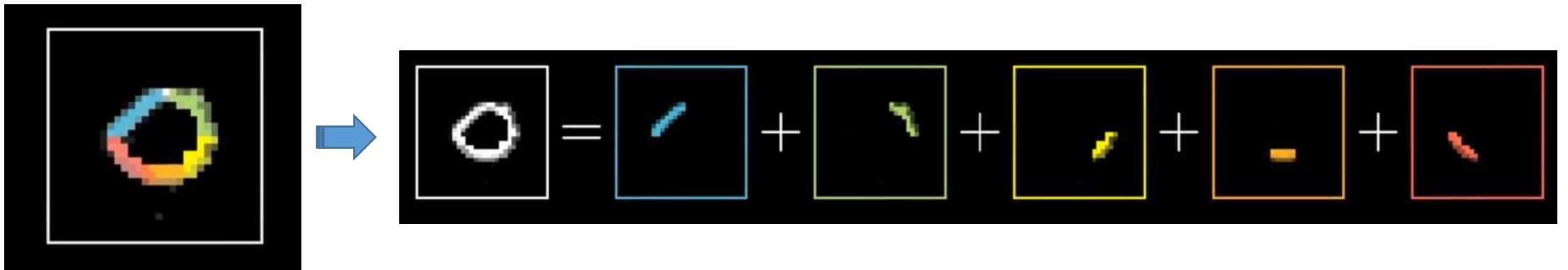


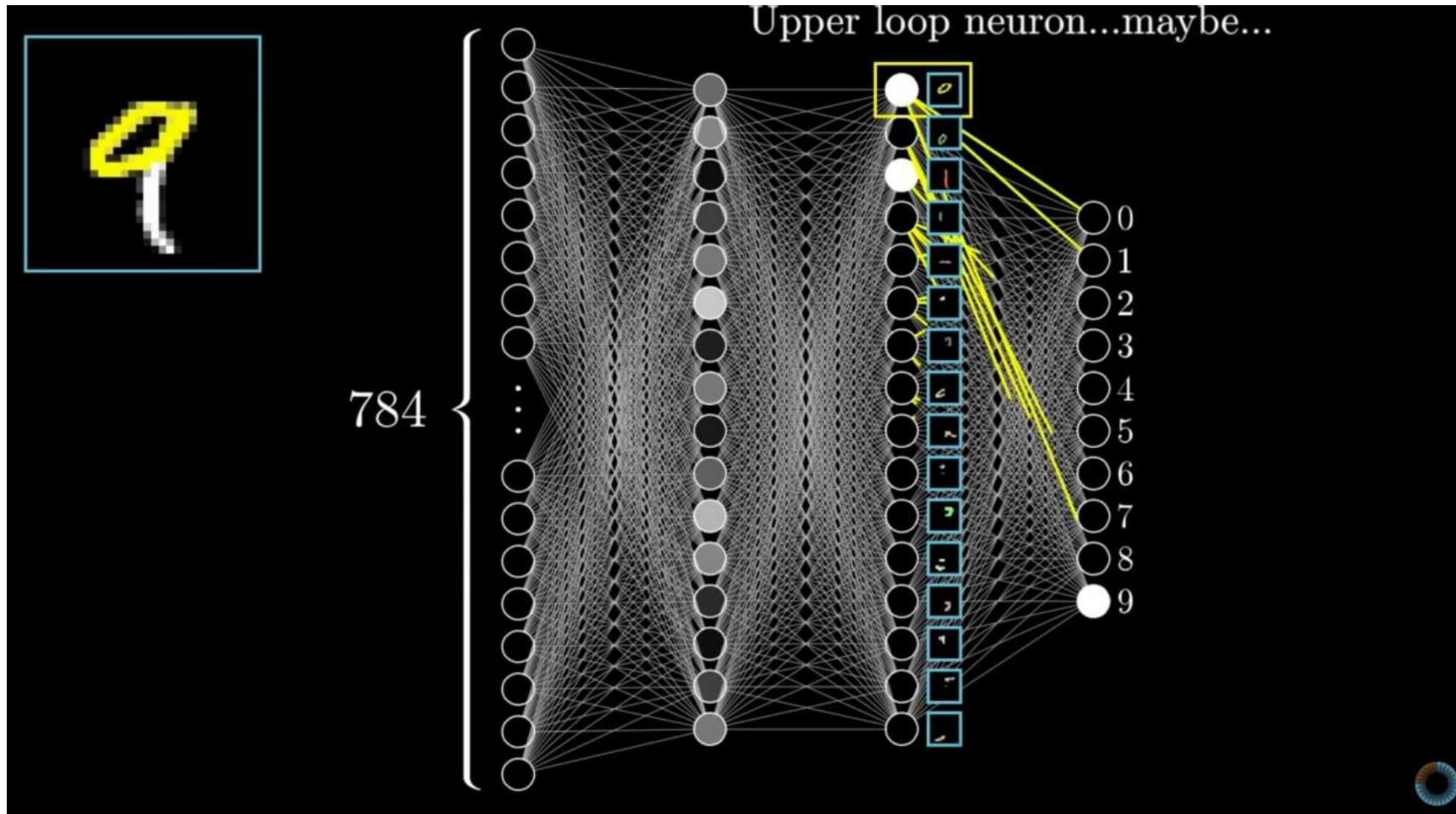
$$28 \times 28 = 784$$

$$q = \text{a} + \text{l}$$

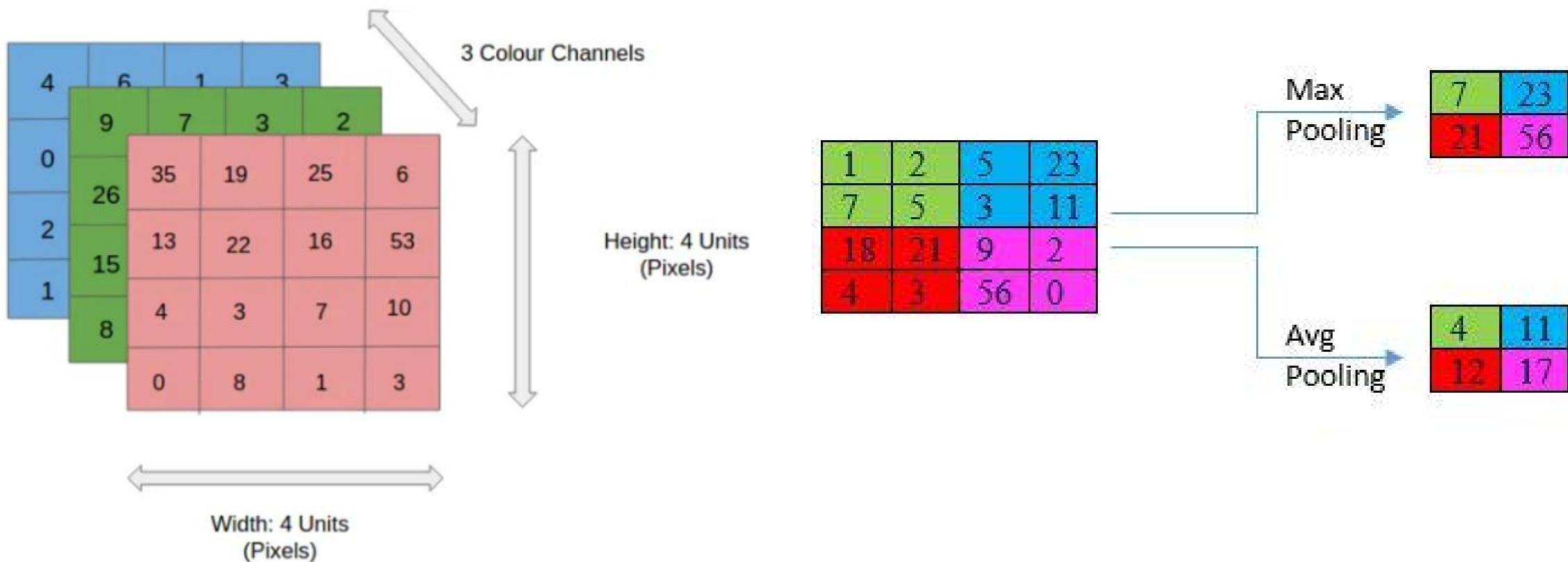
$$g' = \text{a} + \text{o}$$

$$q = \text{l} + \text{f} + \text{r}$$





## Filter



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

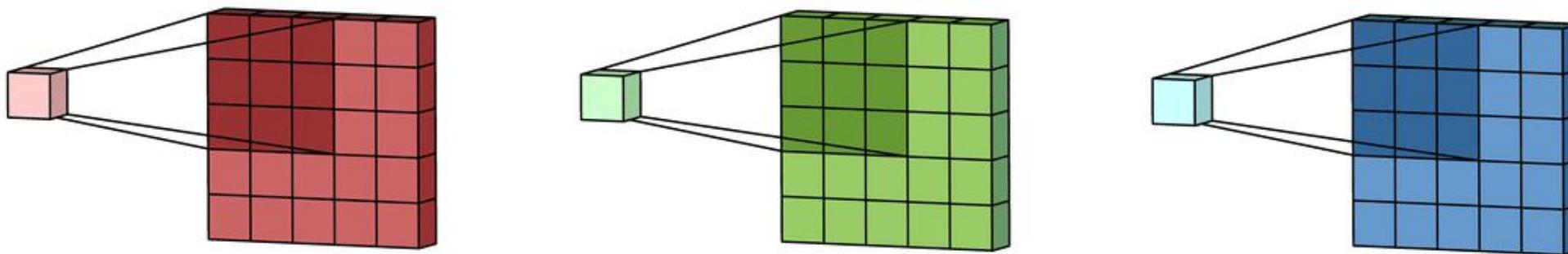
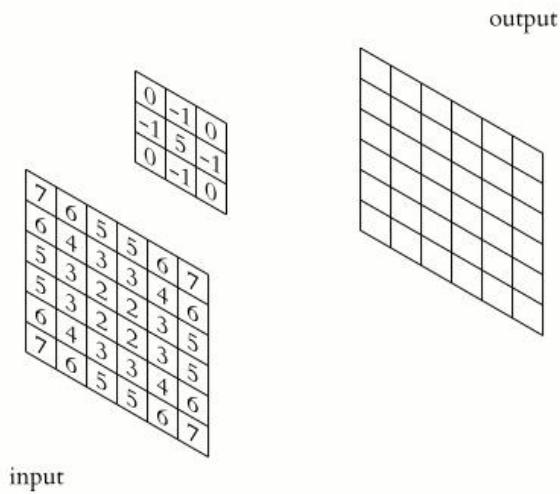
4	3	

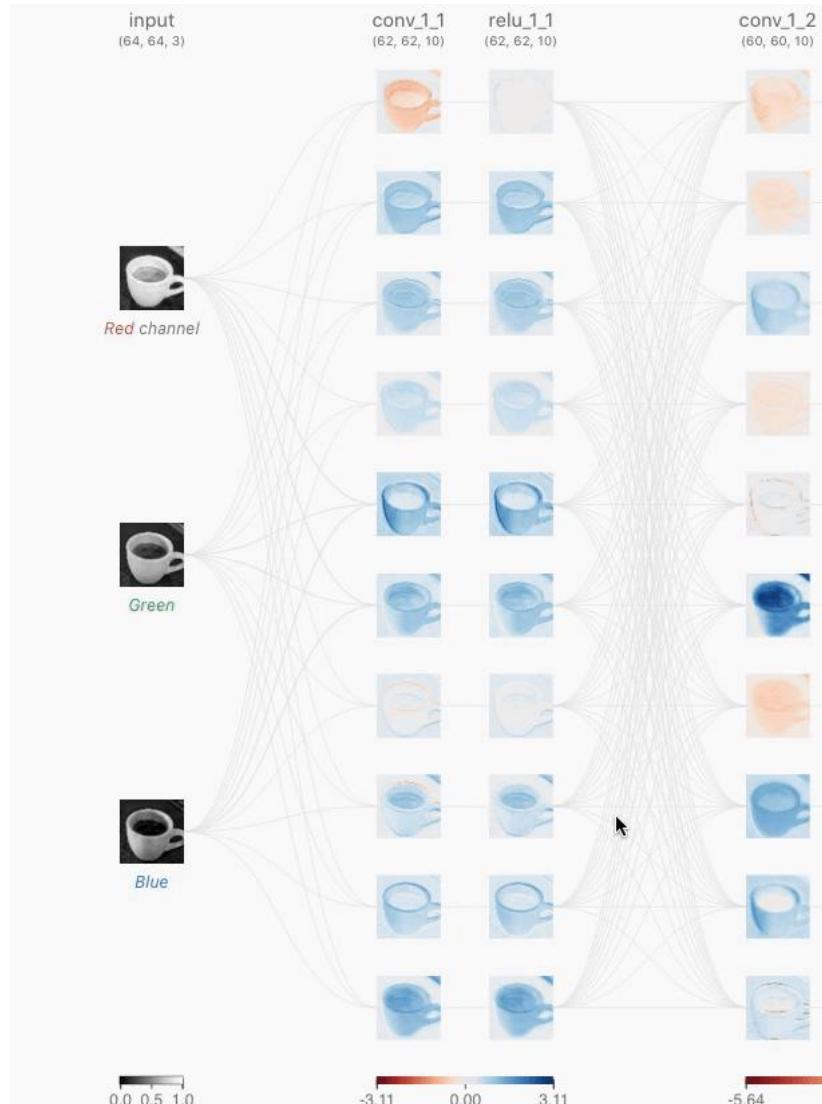
0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

0	-1	0
-1	5	-1
0	-1	0

Kernel

114				





Input Matrix			
45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

$$\begin{array}{c} \text{Kernel} \\ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \text{Result} \\ \begin{bmatrix} -45 & 12 \\ 22 & 10 \end{bmatrix} \end{array}$$

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

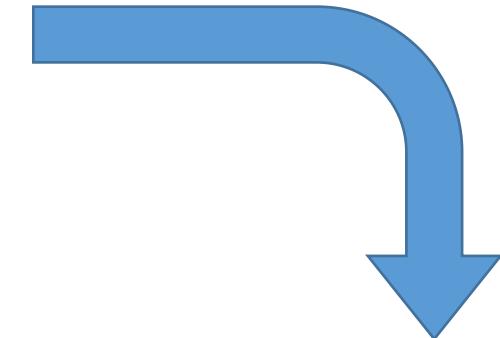
$$\begin{array}{c} \text{Kernel} \\ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \text{Result} \\ \begin{bmatrix} -45 & 109 \\ 22 & 10 \end{bmatrix} \end{array}$$

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

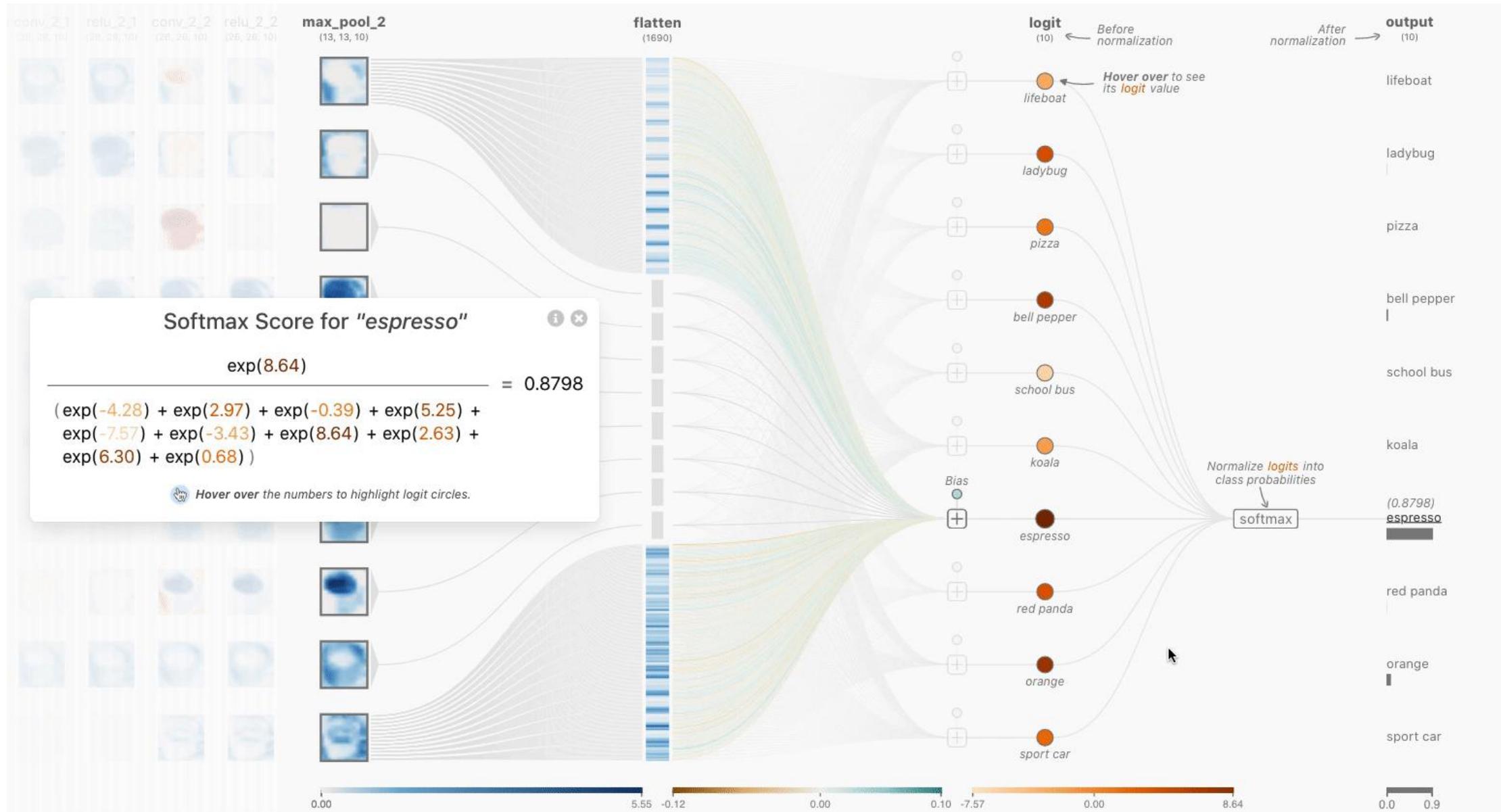
$$\begin{array}{c} \text{Kernel} \\ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \text{Result} \\ \begin{bmatrix} -45 & 103 \\ -96 & 10 \end{bmatrix} \end{array}$$

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

$$\begin{array}{c} \text{Kernel} \\ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \text{Result} \\ \begin{bmatrix} -45 & 103 \\ -176 & 133 \end{bmatrix} \end{array}$$

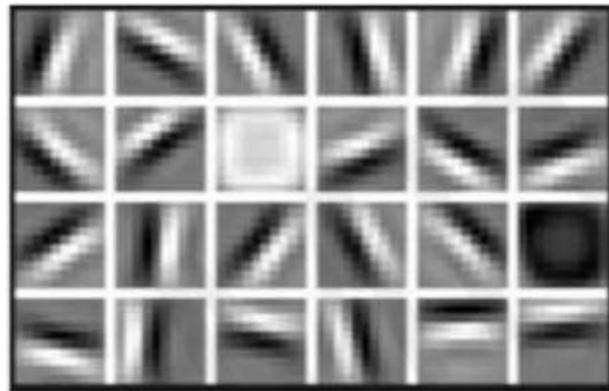


Original	Gaussian Blur	Sharpen	Edge Detection
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$



Insonni yuzida kerakli malumotlarni olishda ishlatalishi mumkin bo'lgan filterlar

Sodda filterlar



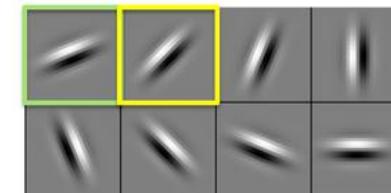
O'rta filterlar



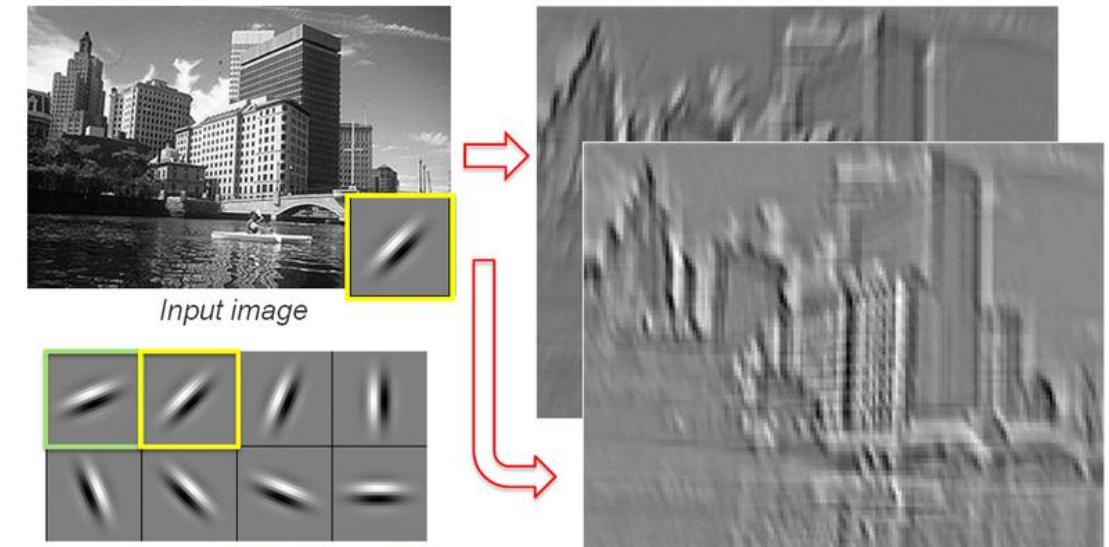
Yuqori filterlar



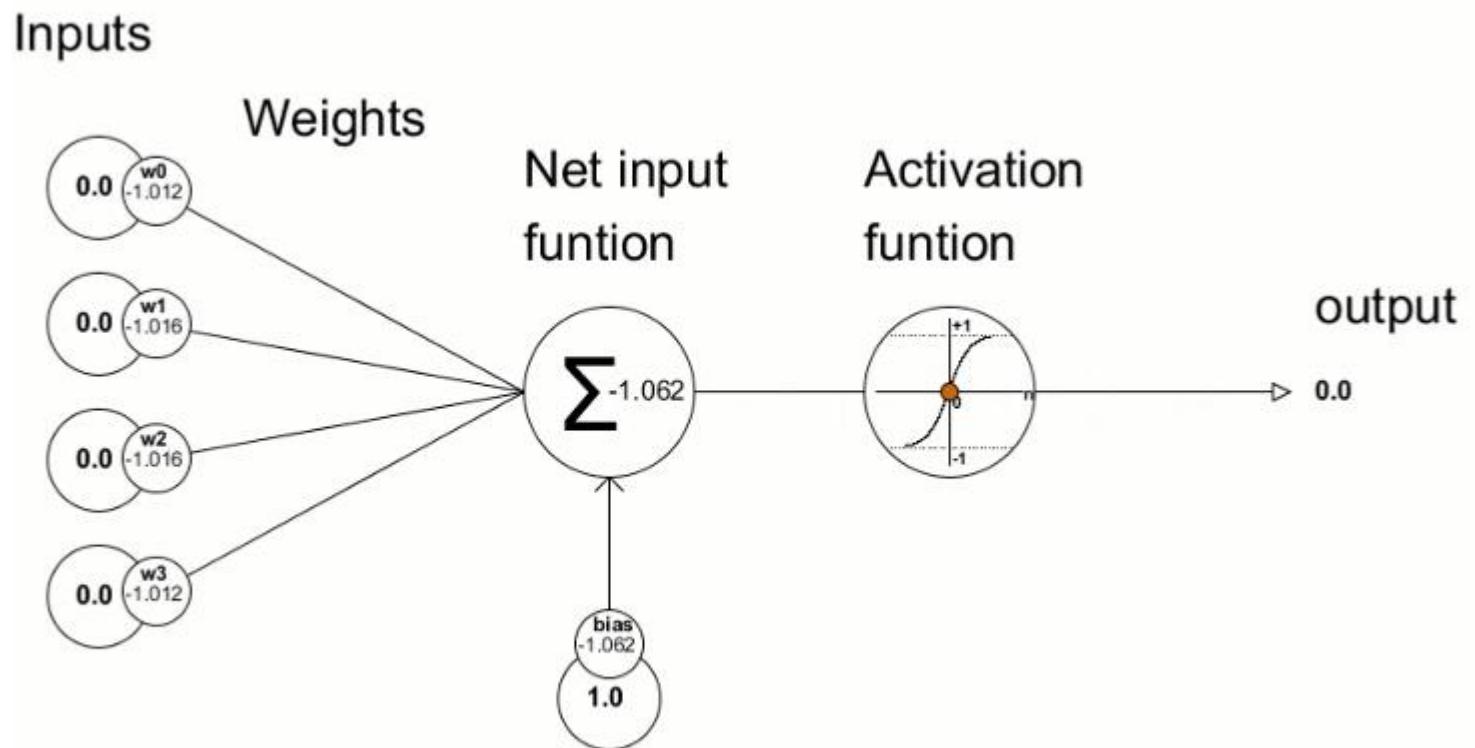
*Input image*

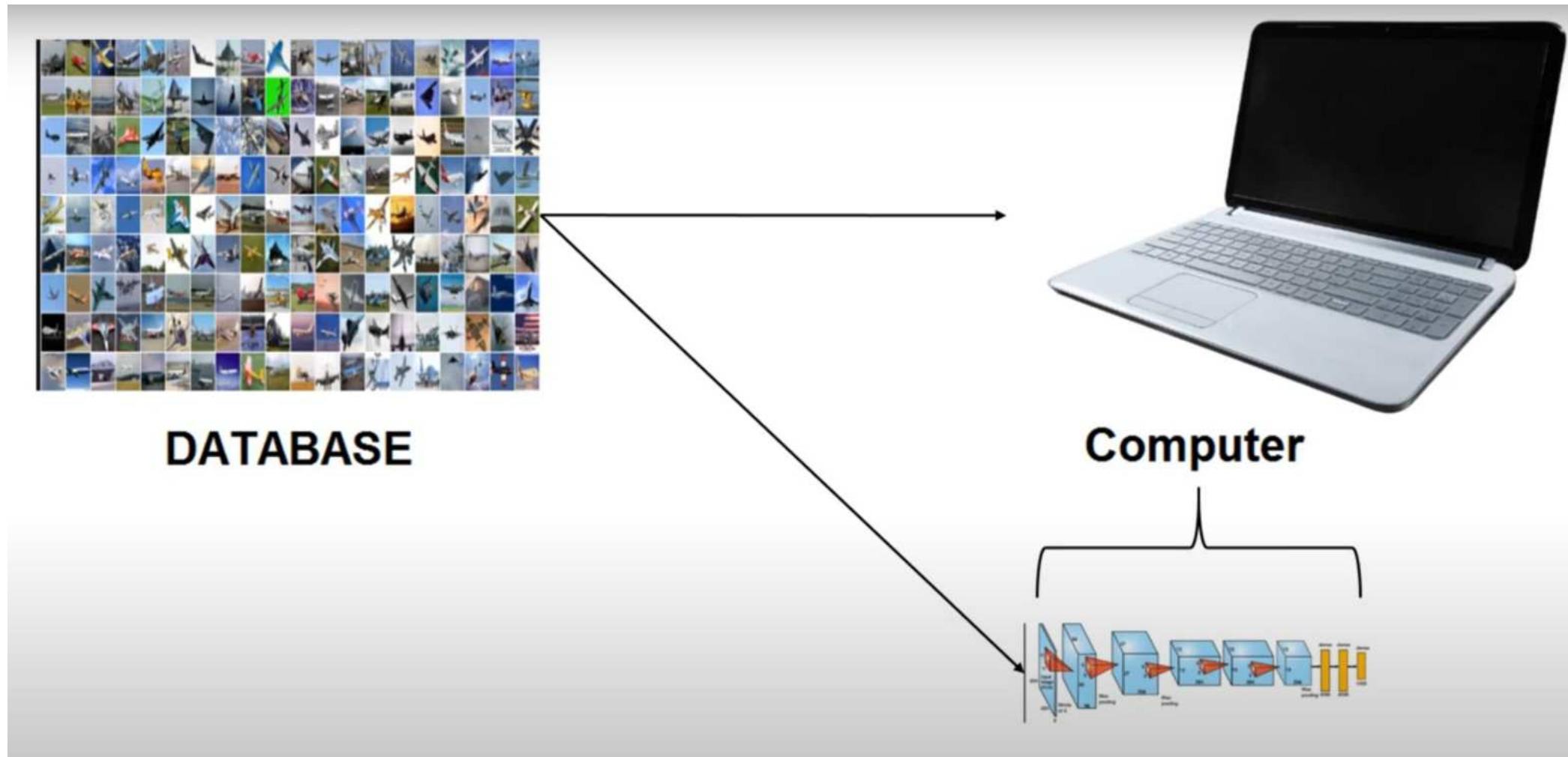


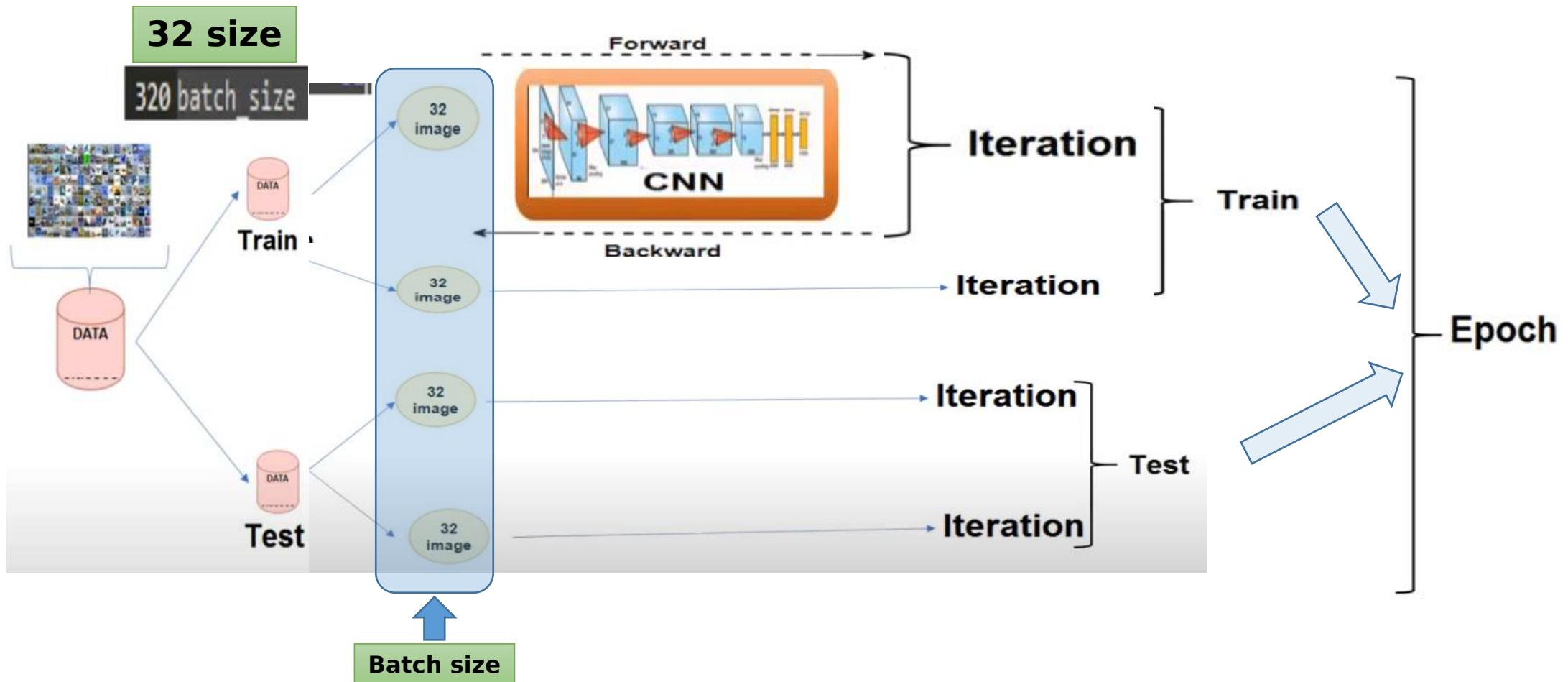
*Filter bank (to be learned)*

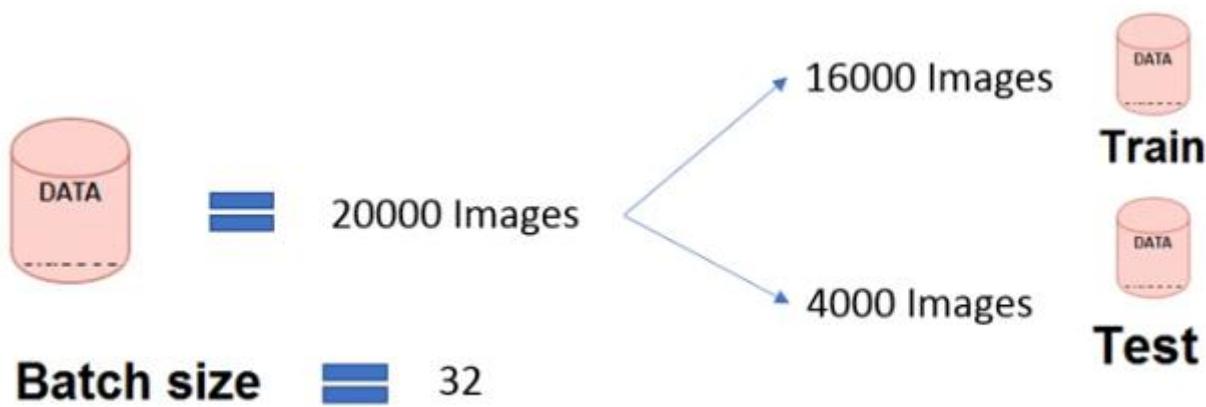


*Feature maps*





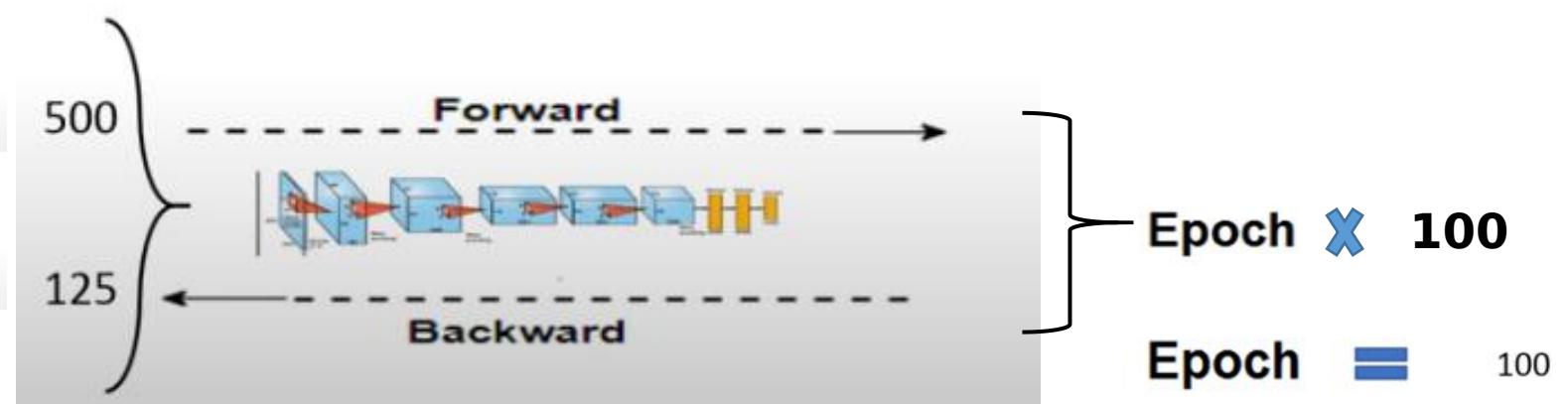


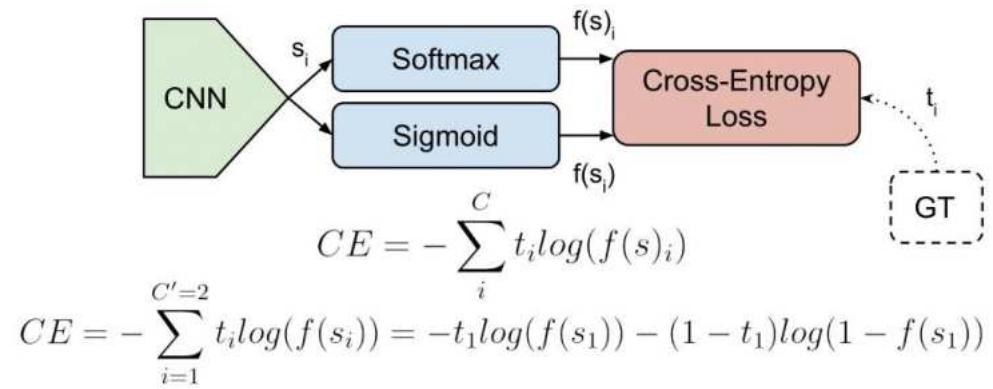
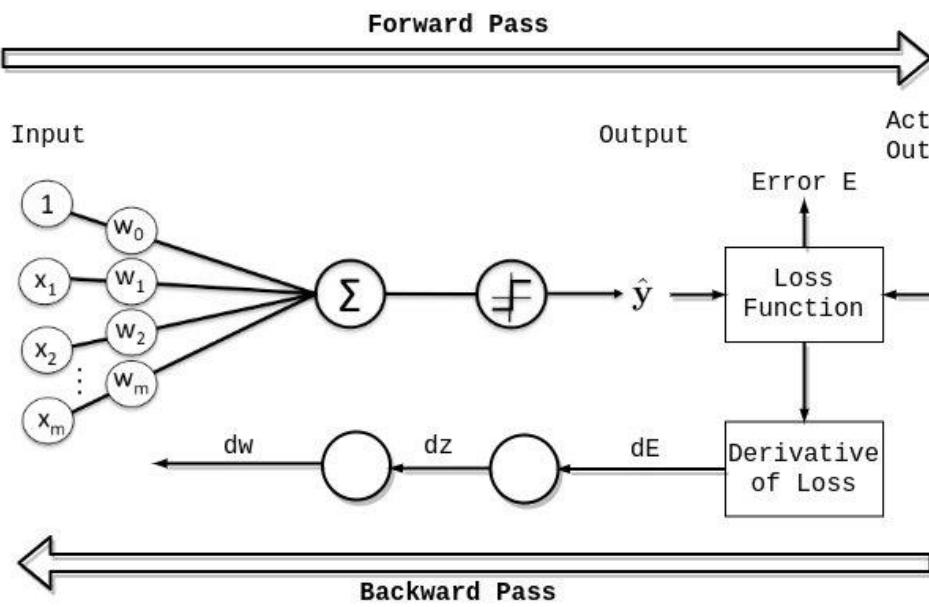


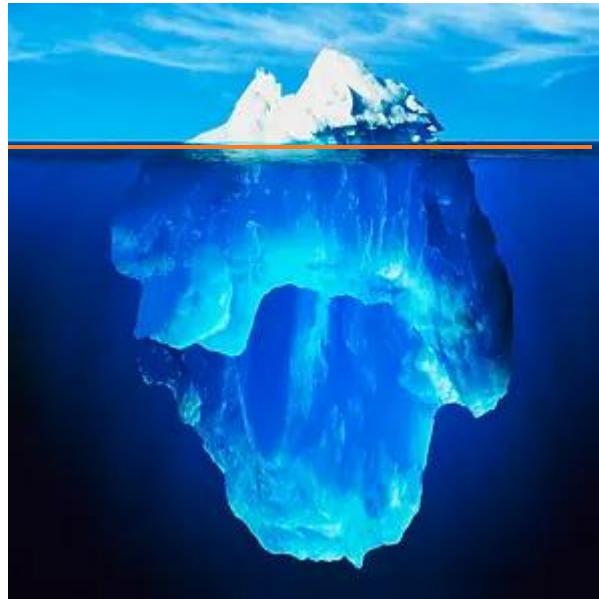
Batch size = 32

Iteration Train =  $\frac{16000}{32}$  = 500

Iteration Test =  $\frac{4000}{32}$  = 125

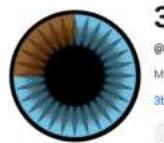






Hozirgacha ko'rganlarimiz

Ko'rishimiz, o'rganishimiz kerak



## 3Blue1Brown •

@3blue1brown • 7.78M subscribers • 223 videos

My name is Grant Sanderson. Videos here cover a variety of topics in math, or adjacent ...more

[3blue1brown.com](#) and 7 more links

Subscribed



## Branch Education •

@BranchEducation • 2.55M subscribers • 50 videos

How do Microchips work? How does your smartphone camera work? How does Bluetooth ...more

[branch.education](#) and 1 more link

Subscribed

Join



## freeCodeCamp.org •

@freecodecamp • 11.2M subscribers • 1.9K videos

Learn math, programming, and computer science for free. A 501(c)(3) tax-exempt charity ...more

[donate.freecodecamp.org](#) and 1 more link

Subscribed

Join



## StatQuest with Josh Starmer •

@statquest • 1.51M subscribers • 292 videos

Statistics, Machine Learning, Data Science, and AI seem like very scary topics, but I'm here to ...more

[patreon.com/statquest](#) and 4 more links

Subscribe

Join



## Indenty •

@Indenty • 316K subscribers • 975 videos

Simple is better than complex...more

[indenty.io](#) and 2 more links

Subscribed

Join



## Science Time •

@ScienceTime24 • 1.12M subscribers • 561 videos

Science Time for science enthusiasts. This channel is dedicated to people who are inter...more

[patreon.com/scientime](#) and 2 more links

Subscribed



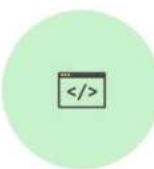
## Normalized Nerd •

@NormalizedNerd • 129K subscribers • 90 videos

Hello, people from the future! Welcome to Normalized Nerd...more

Subscribed

Home Videos Shorts Playlists Posts Q:



## Code In a Jiffy

@codeinajiffy • 45.3K subscribers • 50 videos

More about this channel ...more

[linkedin.com/in/abdullah-tarek](#) and 2 more links

Subscribed

Visit Community



## mCoding •

@mCoding • 246K subscribers • 121 videos

More about this channel ...more

[patreon.com/mCoding](#) and 6 more links

Subscribed



## Jared Owen •

@JaredOwen • 4.22M subscribers • 116 videos

I use 3D Animation to show and explain how things work...more

[patreon.com/jaredowenanimations](#) and 5 more links

Subscribed



**Rahmat**

<https://github.com/DeepSE/deeplearning-models>

2025 / 10 / 18



1. Activation functions - Aktivlashtiruvchi funksiyalar
2. Colab coding

## Activation functions



Sigmoid



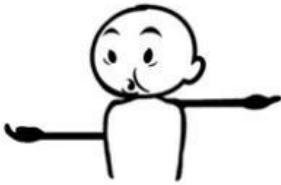
$$y = \frac{1}{1+e^{-x}}$$

Tanh



$$y = \tanh(x)$$

Step Function



$$y = \begin{cases} 0, & x < n \\ 1, & x \geq n \end{cases}$$

Softplus



$$y = \ln(1+e^x)$$

ReLU



$$y = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Softsign



$$y = \frac{x}{(1+|x|)}$$

ELU



$$y = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases}$$

Log of Sigmoid



$$y = \ln\left(\frac{1}{1+e^{-x}}\right)$$

## Activation functions



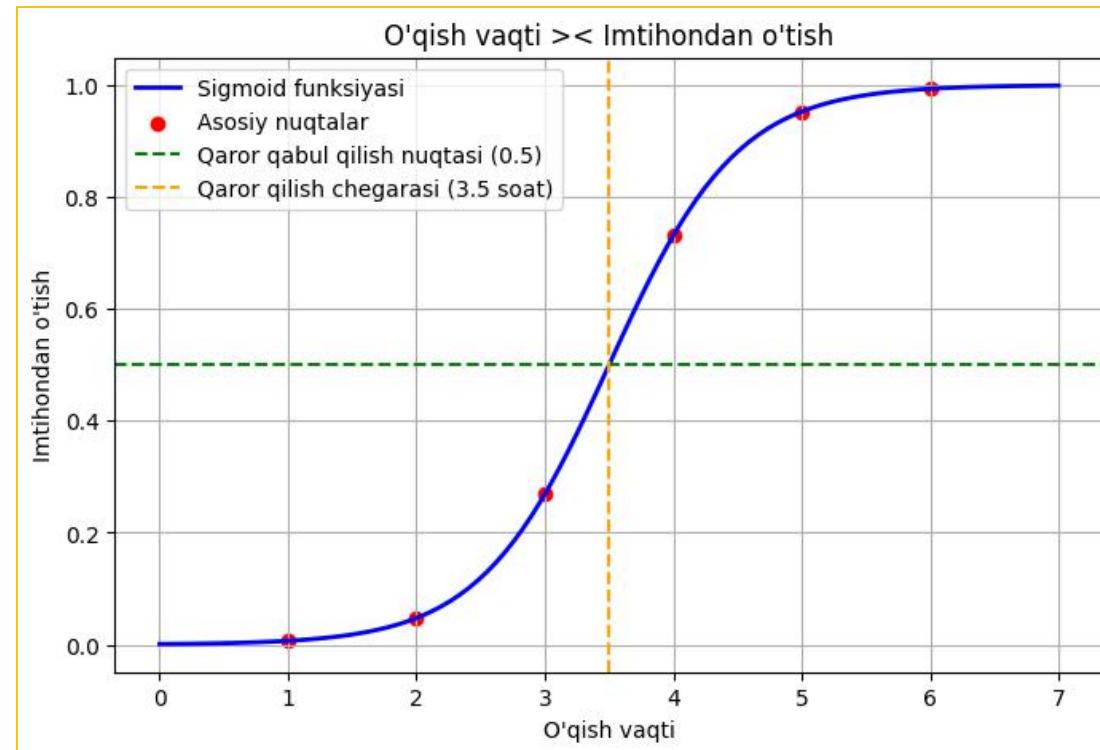
## Ko'chaga aylanishga chiqish uchun qaror qabul qilish

Input (savol)	Qarorni baholash	Misol	Qiymat	Ma'nosi
x1	Havo quyoshlimi?	Ha (1)	+2	Quyoshli havoni yaxshi ko'rasiz
x2	Havo sovuqmi?	Ha (1)	-3	Sovuqni yomon ko'rasiz
x3	Vaqtim bemalolmi?	Ha (1)	+4	Bo'sh vaqtida aylanishni yaxshi ko'rasiz

Natija:  $(1*2)+(1*-3)+(1*4)= 2-3+4=3$ Qaror: Bias -1 deb hisoblasak:  $3-1=2$  // **Ko'chada aylanaman (2)**

Soat	Natija
1	0
2	0
3	0
4	1
5	1
6	1

$z=2x-7$	Sigmoid
$2 \times 1 - 7 = -5$	0.0067
$2 \times 2 - 7 = -3$	0.047
$2 \times 3 - 7 = -1$	0.269
$2 \times 4 - 7 = 1$	0.731
$2 \times 5 - 7 = 3$	0.953
$2 \times 6 - 7 = 5$	0.993



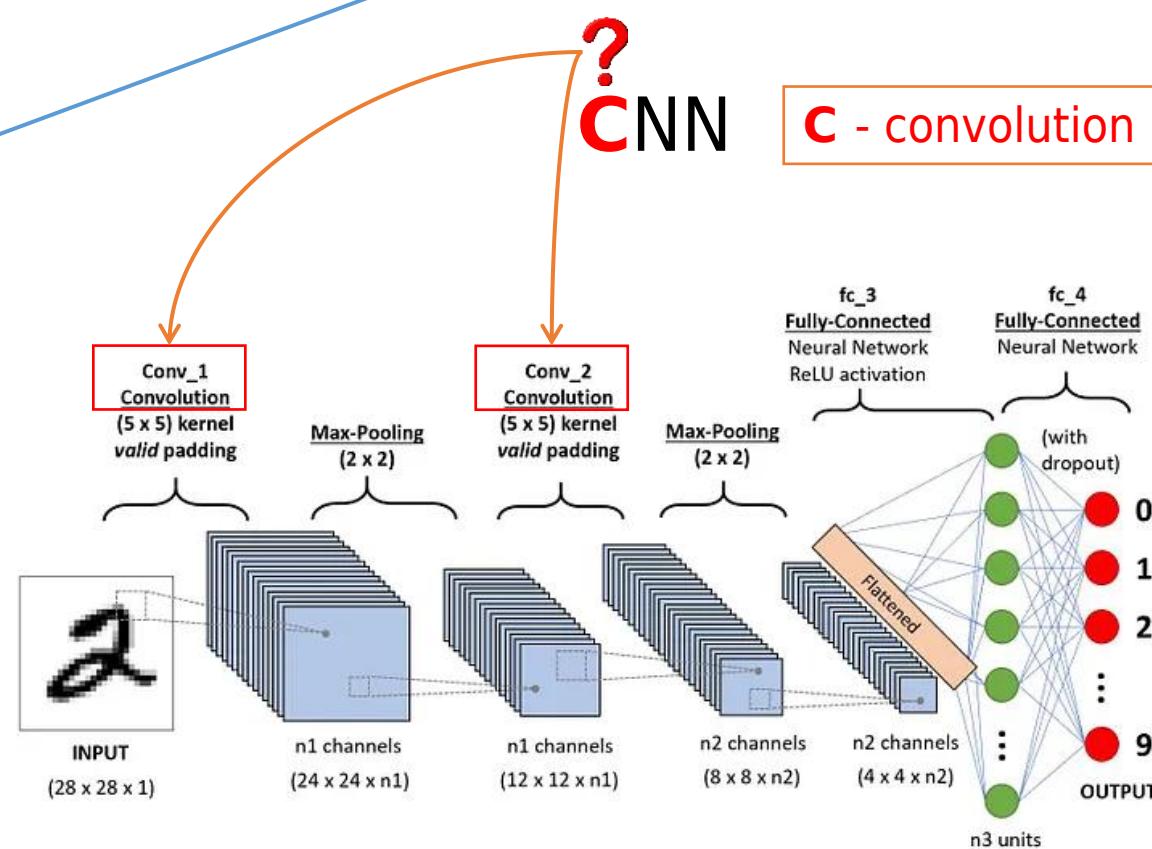
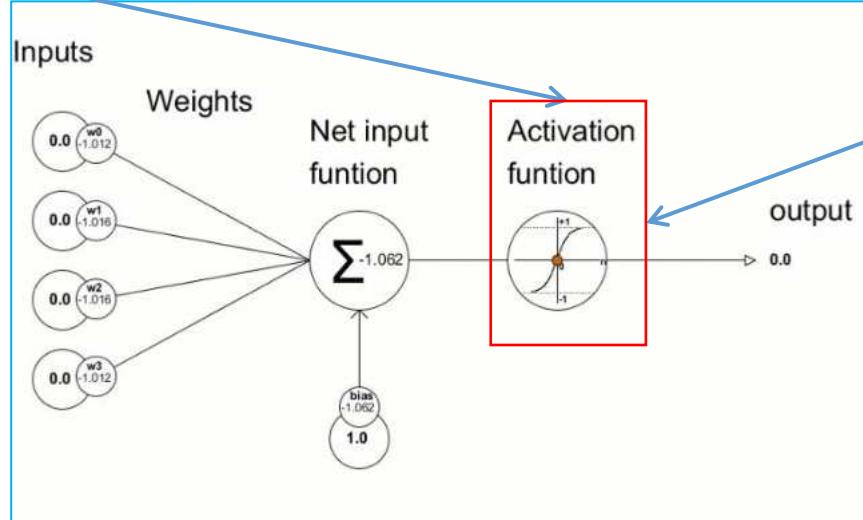


## Activation functions

## Activation functions in NN

Bu **funksiyalar** **NN** larda oxirida qoror qabul qilishda ishlataladigan **funksiyalar**

NN ✓



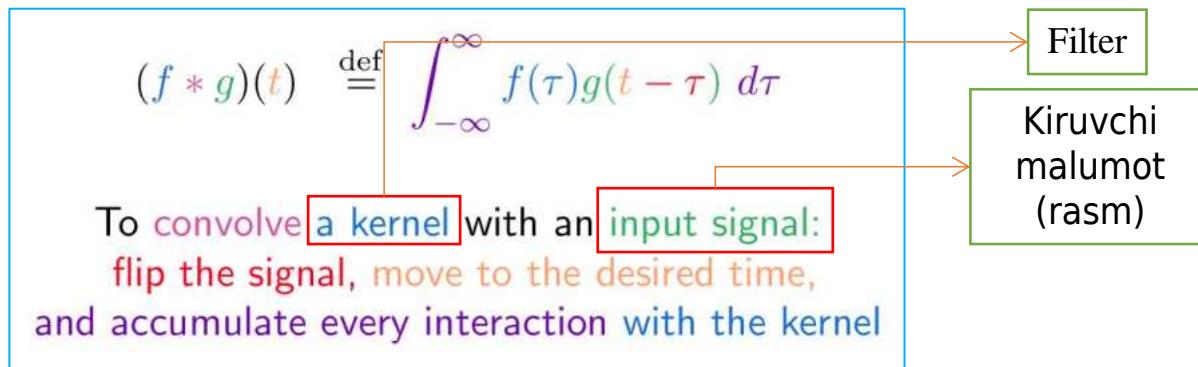
**C** - convolution



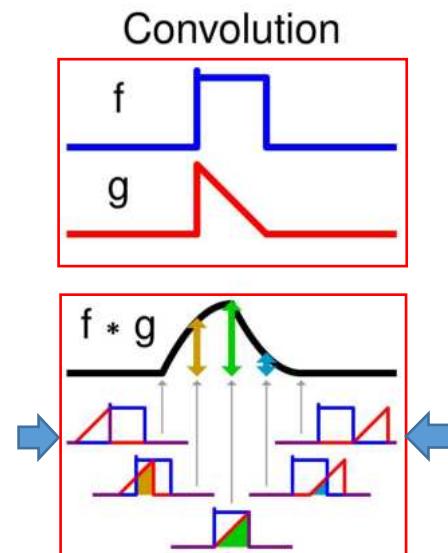
## Activation functions

## CNN

Convolutional Neural Network (CNN) da C :



bu matematik operatsiya hisoblanib,  
 funksiyalarini bir - biriga bog'laydi va  
 yangi bog'langan funksiyani xosil qiladi.



## Activation functions

Bu funksiyalarning asosiy vazifasi bog'lanib, suyanib qolmaslik (**non - linearity**) taminlash.

Yani, biz o'rgatayotgan algoritm faqat bir turdag'i malumotlar asosida qaror qabul qilmasligini taminlash.



Klassifikatsiya qilish uchun muhim,  
ahamiyatli malumotlari

**Ot**      **Zebra**      **Sigir**      **Qo'chqor**      **Qo'y**

Oyoqlar soni

4      4      4      4      4

Rangi

oq- qizil      oq - qora      oq -qizil      oq      oq

Shoxi

yo'q      yo'q      bor      bor      yo'q



Activation  
Functions

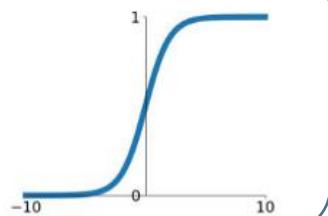
## Activation functions



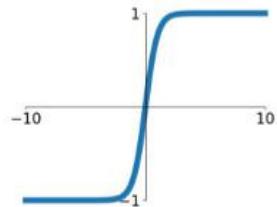
## Activation functions

**Sigmoid**

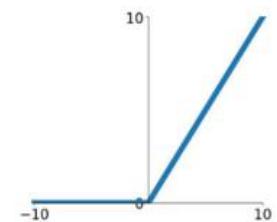
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

**tanh**

$$\tanh(x)$$

**ReLU**

$$\max(0, x)$$



$$a = f(z) = f(Wx + b)$$

W - qiymay

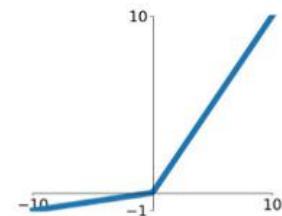
x - kiruvchi malumot

b - bias

f - AF (activation functions)

**Leaky ReLU**

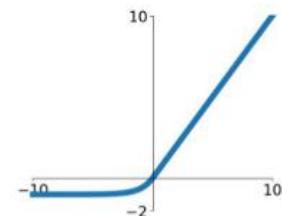
$$\max(0.1x, x)$$

**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

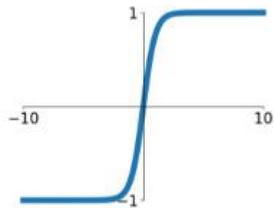
**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



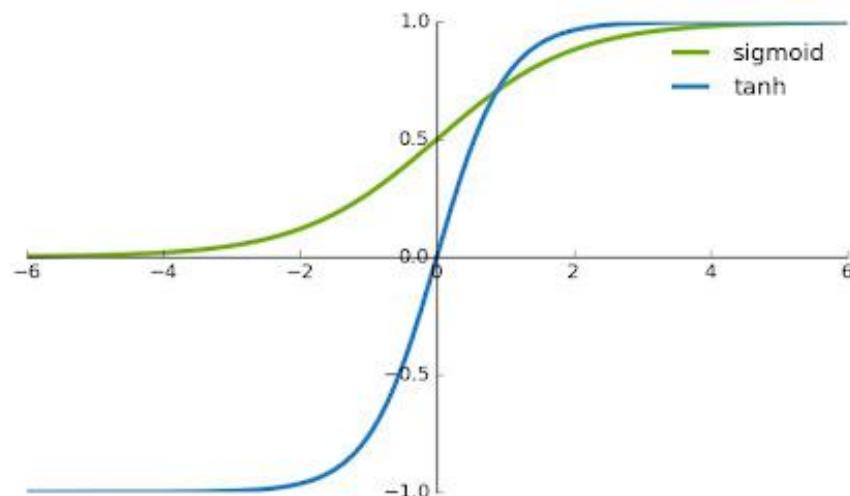
**bias** - aktivasiya qiluvchi funksiyalarga tenglamani y o'qida harakatini taminlaydi. AF larga qaror qabul qilish chizig'ini belgilab olishda yordam beradi (threshold line).

## Activation functions

**tanh** $\tanh(x)$ 

$$= \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

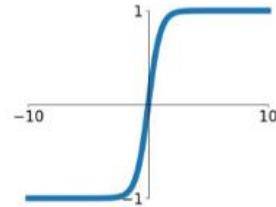
hyperbolic tangent (**tanh**) - S ko'rinishida ifodalangan algoritm, CNN da yakuniq qaror qabul qilishda ishlatiladi. Olingan malumotlarni muhim qismlaridan kelib chiqib qaror qabul qiladi. U qarorni  $-1$  va  $1$  intervali oralig'ida hisoblab chiqadi. Bu algoritm  $0$  da markazlashgan hisoblanadi, va ko'rinishida Sigmoid funksiyasiga o'xshaydi.



## Activation functions



**tanh**  
 $\tanh(x)$



$$= \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

**0** - da markazlashganligi nima ahamiyat kasb etadi: Bunda, ijobiy va salbiy kiruvchi xabarlar malumotiga ko'ra javobga uzatiladi. (ijobiy / salbiy).

Ishlatilishi:

- + Sentiment analysis
- + Ovozni aniqlash
- CNN (optimal emas)

Toshkent



London

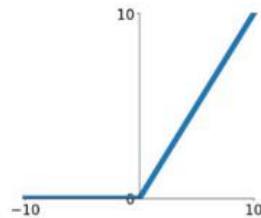




## Activation functions

Rectified Linear Unit - **ReLU**

**ReLU**  
 $\max(0, x)$



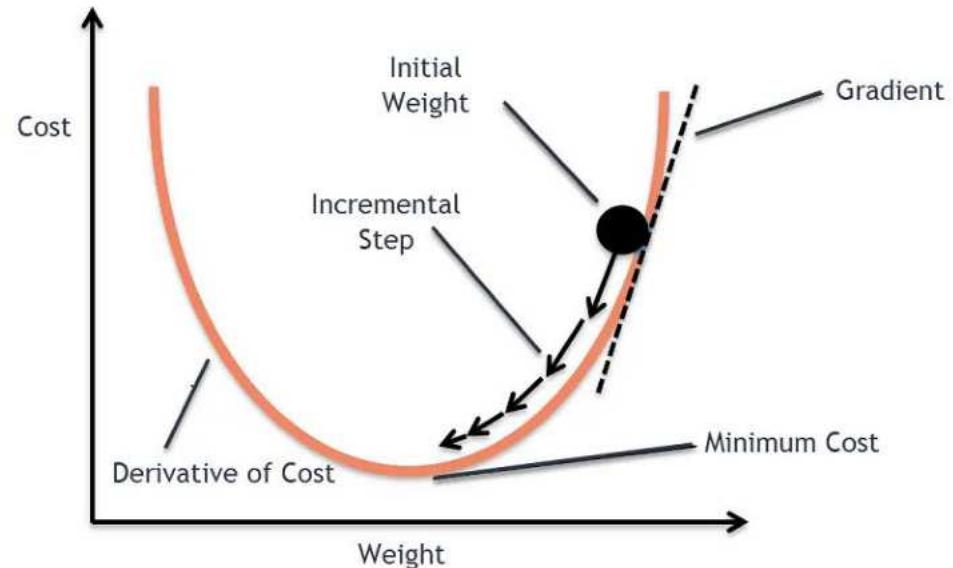
$$f(x) = \max(0, x)$$

- Oddiy, lekin NN lar uchun juda foydali AF

Agar kiruvchi malumot (input)  $x > 0$  bo'lsa, natija =  $x$

Agar kiruvchi malumot (input)  $x \leq 0$  bo'lsa, natija = 0

Sigmoid va Tanh algoritmlari ReLU dan oldin mashxur bo'lgan. Lekin u algoritmlar malumotlar katta (ijobiy, salbiy), gradient nolga yaqin keladi va modelni o'qishi juda sekinlashib qoladi.



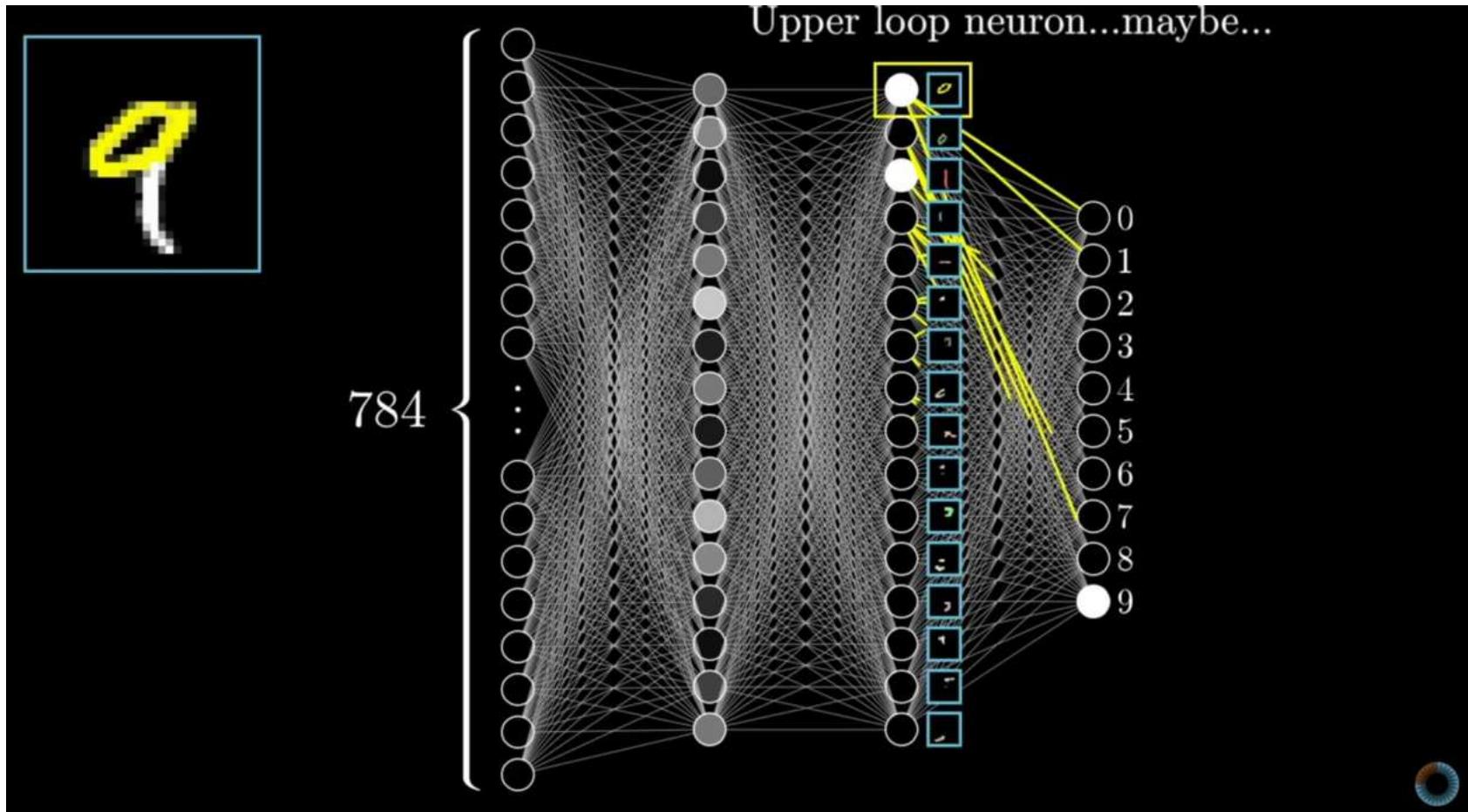


## Activation functions

**ReLU** har bir Convolution Layer dan do'ng yoki to'liq bog'langan layerlarda so'ng qo'yilib, malumotlarni liniyalı emasligini taminlaydi. Bunday qilmasa, malumotlar liniyalı deb hisoblanib ketilishi mumkin.

Demak, ReLU ishlatilgandan so'ng hosil bo'ladigan ko'nikmalar non-linear malumotlarga qaratiladi:

ReLU dan ijobiy, ahamiyati yuqori qiymatlar o'tadi, kichkina, salbiy malumotlar tushib qoladi.



Agar, neyronda faqat salbiy qiymat bu algoritm o'qimasligi sababli gradient 0 bo'lin, to'xtab qoladi, yani model malumotdan ko'nikma olmaydi.

**Leaky ReLU / PReLU**

## Activation functions



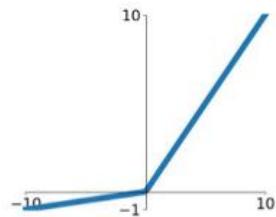
**ReLU** ni ishlatalish o'rirlari:

- CNNs (image recognition)
- GAN
- Ovozni aniqlash
- NLP / Transformers

## Activation functions



**Leaky ReLU**  
 $\max(0.1x, x)$



$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

Leaky ReLU neyronda salbiy qiymat kelib qolsa ham ularni hisoblab, gradientni 0 bo'lib qolishini oldini oladi.

Ishlatilish o'rnlari ReLU bilan bir xil, va **segmentatsiya** uchun ham keng qo'llaniladi

## Activation functions



## Umumiy xulosa

Nomlanishi	Manosi	CNN da
Weight (w)	Qaysi qismga etibor berishni o'rganadi	Filter qiymatlar
Bias	Qachon aktivlashtirishni o'rganadi	Convolutionda keyin, activationdan oldin
Activation	Malumotni qanchalik etibor bilan uzatishni o'rganadi	ReLU, Sigmoid, Leaky ReLU, Tanh...,



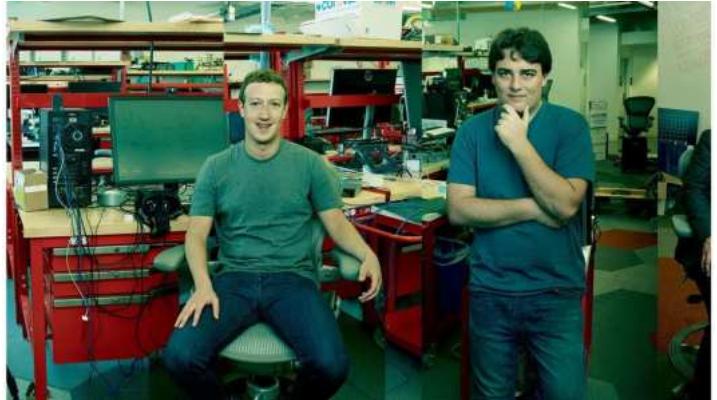
Alexandr Wang (Chinese: 汪滔; pinyin: Wāng Tāo;<sup>[2]</sup> born January 1997) is an American billionaire entrepreneur who was named Meta's chief AI officer in 2025.<sup>[3]</sup> He is the co-founder and former CEO of [Scale AI](#), an artificial intelligence company that provides [data labeling](#) and [large language model](#) evaluation services to develop AI applications.<sup>[4][5]</sup> In 2021, he was the world's youngest billionaire at age 24.<sup>[6][7][8]</sup> [Forbes](#) estimated his net worth at \$3.6 billion as of April 2025.<sup>[9]</sup>

This Ivy League reject taught himself to code at 7, built a \$30m app by 18 and is treating college like a \$100k vacation: 'not necessarily things that are directly related to my company'



⚡ 18 yoshli talaba oziq-ovqat fotosuratlari orqali kaloriyalarni hisoblaydigan ilovasidan allaqachon 30 million dollar [ishlab topdi](#).

Dastur oziq-ovqat fotosuratlarini tahlil qilish uchun oddiy ChatGPT'dan foydalanadi, yigit shunchaki o'z dasturiga ChatGPT'ni ulab qo'ygan.



Oculus asoschisi [Palmer Luckey](#) va uning "do'sti" Zuckerberg yangi harbiy shlem - **EagleEyen**i taqdim etdi.

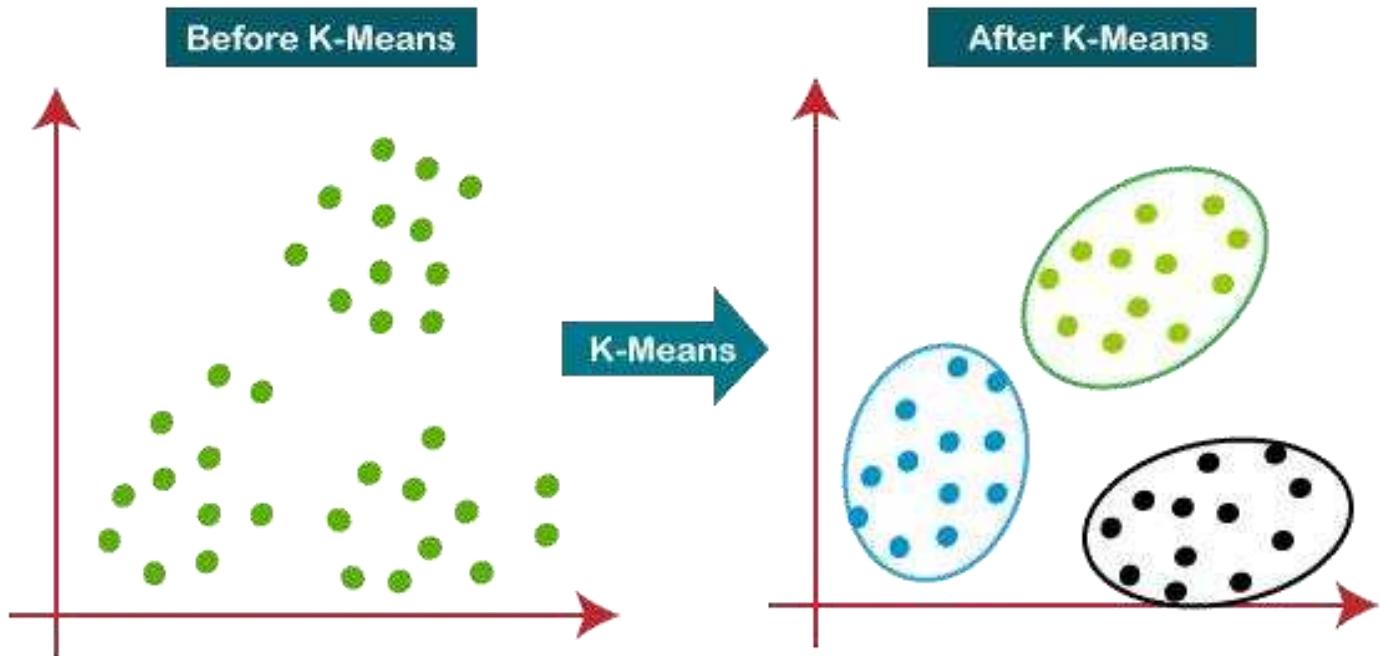
U AI va AR texnologiyasi orqali askarlarga real vaqtida ma'lumot ko'rsatadi - joylashuv, xavfli hududlar va dron tasvirlari.

### Obyektni aniqlaydigan aqqli shlemlar

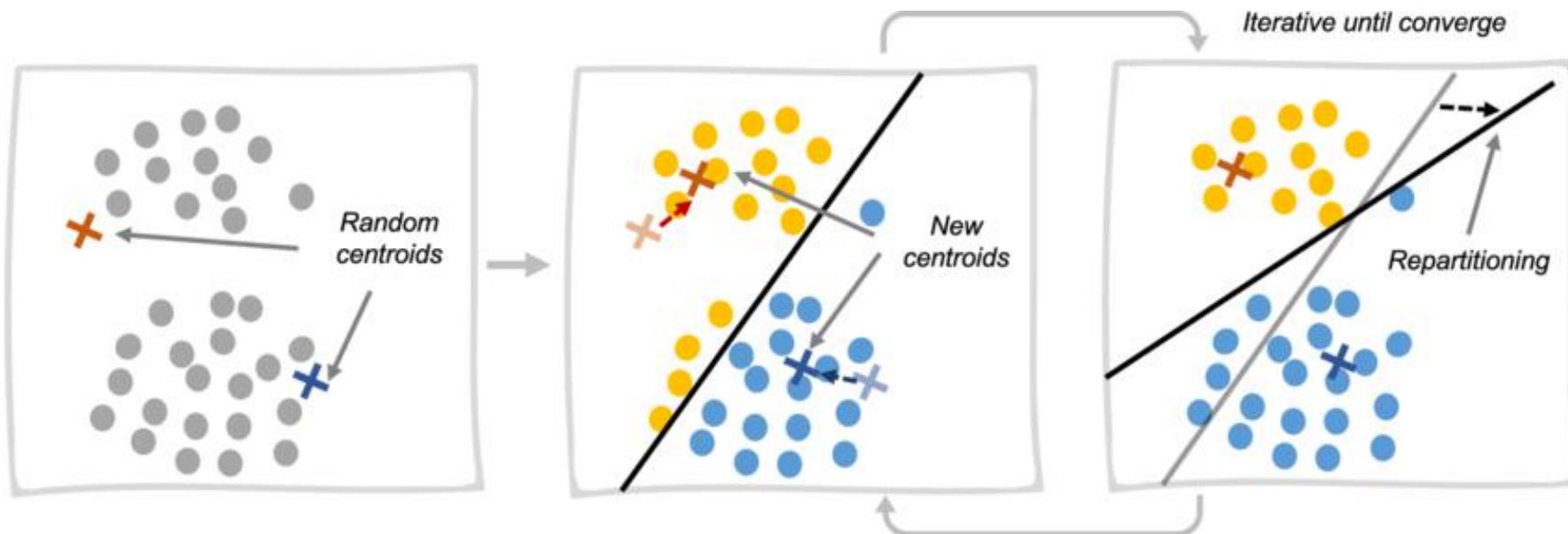


**Colab coding**

Unsupervised Learning algoritmi: K - means klustering



## Colab coding



<https://github.com/DeepSE/deeplearning-models>