




Computer Vision


A brief, interactive introduction to the challenge and the methods


Review - what (extra) DA tasks are involved in working with these sources of data?





 (files with tables of data)

 (databases)

 (spatial,lat/long)

 (scraping)

 (json, xml)

 (text, descriptive)





What is the challenge of computer vision?

This is a task that humans find trivial

1. Describe a photo you have seen once
2. Summarise a video
3. Recognise a face

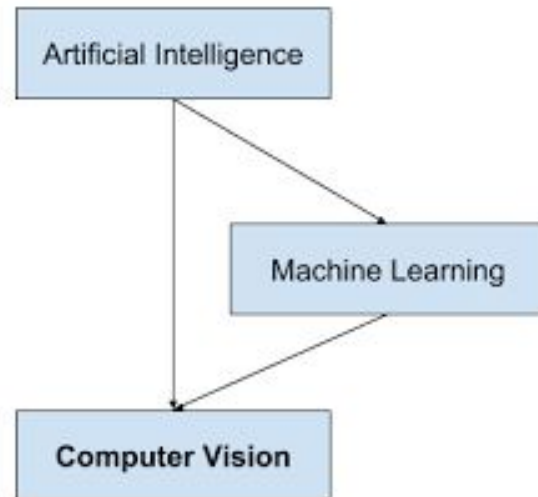
Even the youngest children can do this - they have been trained to since they first opened their eyes.

90%

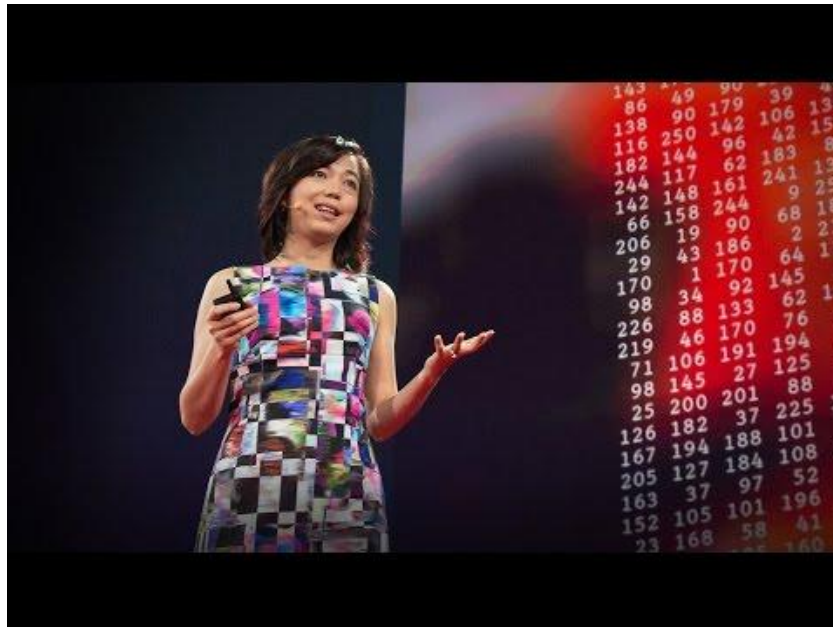
13ms

60,000 times

3 months old

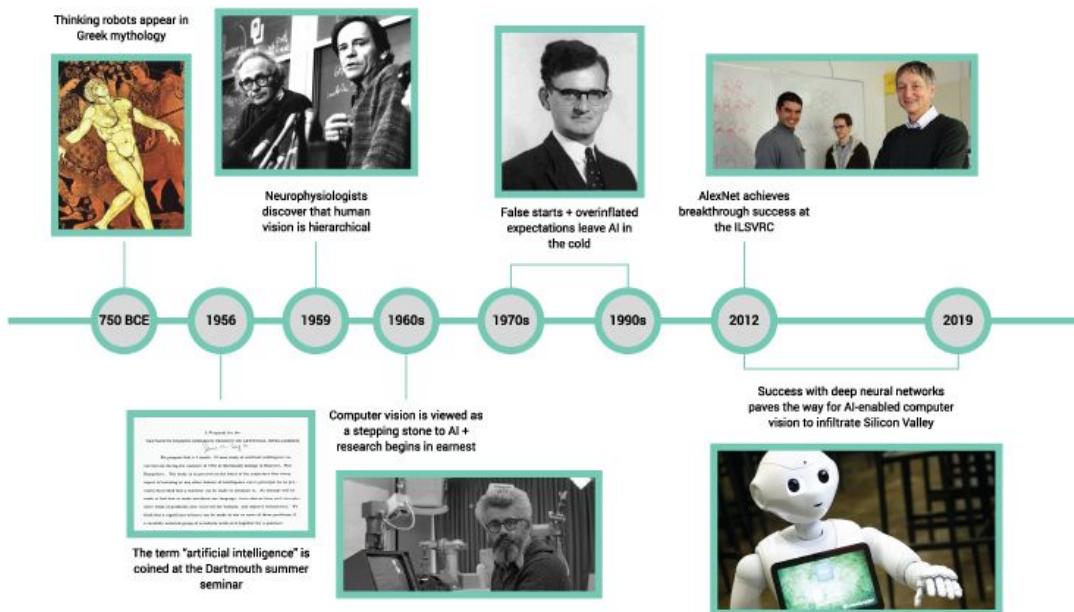


Even the youngest children can do this...



Successes and failures

History of stunted progress in computer vision

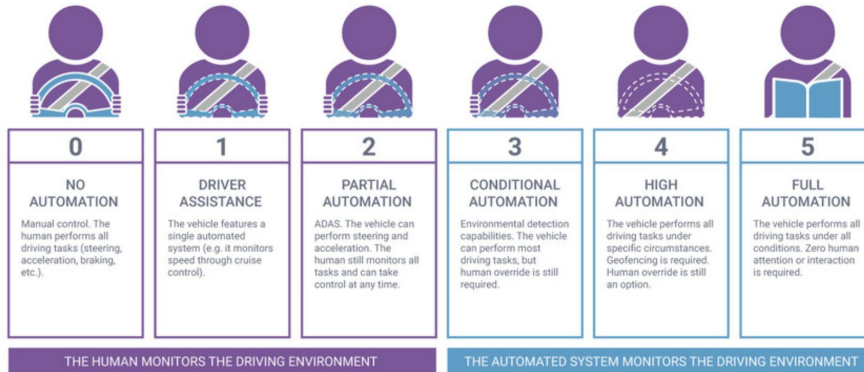




Case study / success stories of computer vision

- Automated checkouts - retail stores
- Medical imaging
- 3d model building
- Automotive safety
- Surveillance
- Fingerprint recognition / biometrics
- Machine inspection

The self driving car challenge



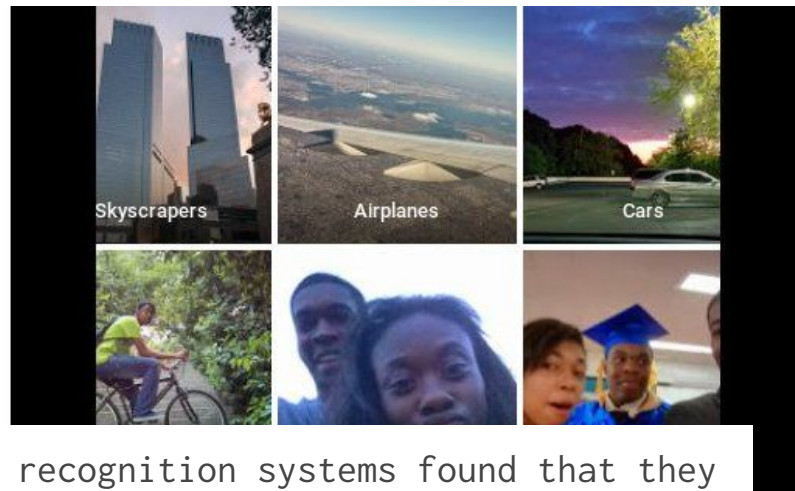
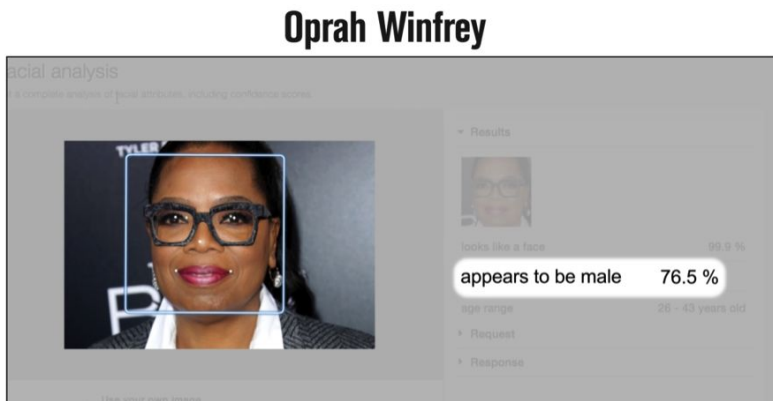
George
Hotz

#31

Lex
Fridman



The race challenge in facial recognition



“In 2019, a federal study of over 100 facial recognition systems found that they falsely identified African-American and Asian faces 10 times to 100 times more than Caucasian faces.”



Why is it so difficult to interpret images?

We don't understand how vision works in humans

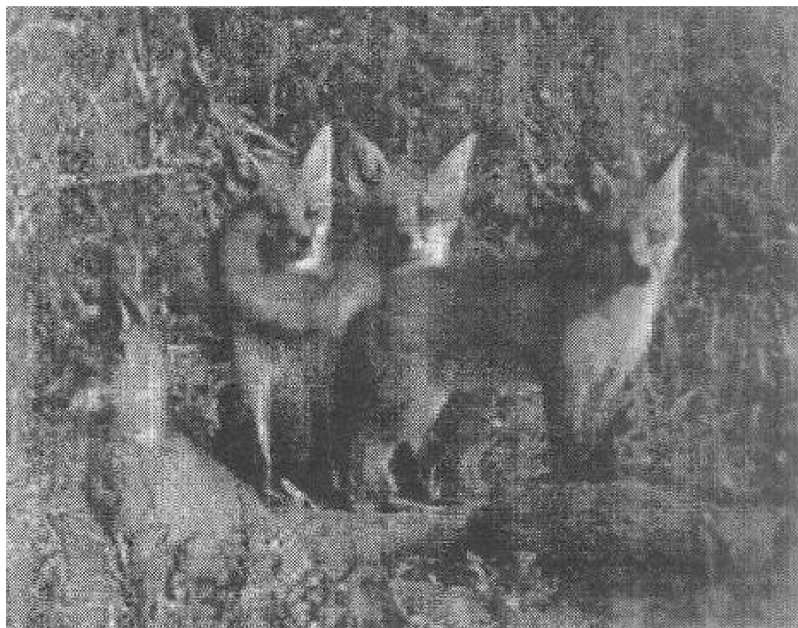
The visual world is enormously complex

Too many variables - orientation, lighting, occlusion

An infinite number of scenes must be comprehended

Context is extremely important

Visual tasks are not purely data driven



Challenges this image presents to “bottom up” image analysis

- perform figure-ground segmentation of the scene (into its objects and background)
- infer the 3D arrangements of objects from their mutual occlusions
- infer surface properties (texture, colour) from the 2D image statistics
- infer volumetric object properties from their 2D image projections
- do all of this in “real time?” (if survival depends on it.)

**When does the human brain
struggle with images?**


**Describe
what the
image
shows**



**Describe
what
happens in
the video**



Can you
tell me
who is in
this image?



Applying the techniques

Tasks of computer vision

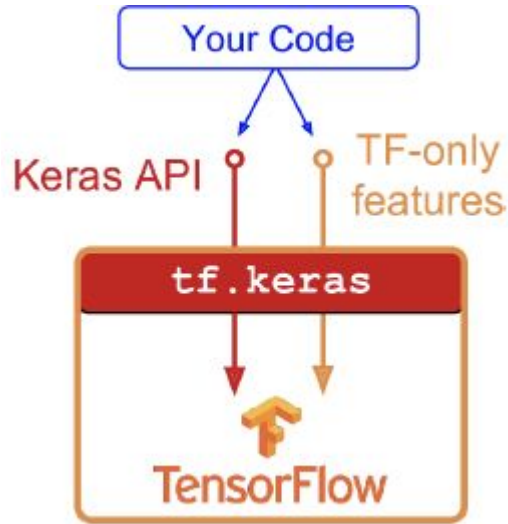


- Image classification with localisation - Labeling an x-ray as cancer or not and drawing a box around the cancerous region.
- Object detection - Drawing a bounding box and labeling each object in an indoor photograph.
- Object Segmentation - Segmenting all pixels in an image into different categories of object.
- Style Transfer - Applying the style of specific famous artworks (e.g. by Pablo Picasso or Vincent van Gogh) to new photographs.
- Image Colourisation - Datasets often involve using existing photo datasets and creating grayscale versions of photos that models must learn to colorize
- Object landmark detection - face recognition, pose recognition, emotion recognition
- Image Reconstruction - filling in missing or corrupt parts of an image.
- Image synthesis - Generating targeted modifications of existing images or entirely new images.
- Image captioning - Generating a textual description of an image
- Text to Image - Synthesizing an image based on a textual description - artistic style synthesis

“a giraffe
standing on
dirt ground
near a tree.”



Lets try it- image classifier with Sequential (stack)



Start google colab

Import tensor flow & keras

Data set = fashion_mnist

Create validation set

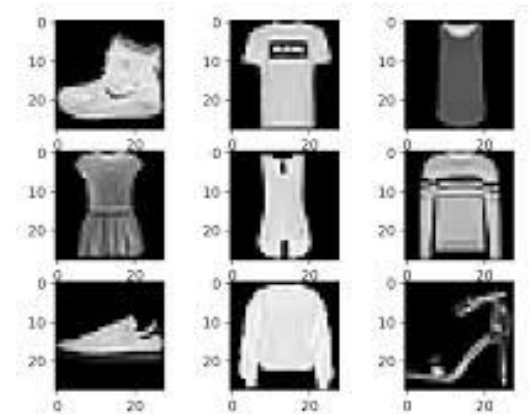
Scale data appropriate to algorithm

Define classes - use plot to validate y

Build neural network

Compile model , Train & Evaluate

Predict using model (new images)

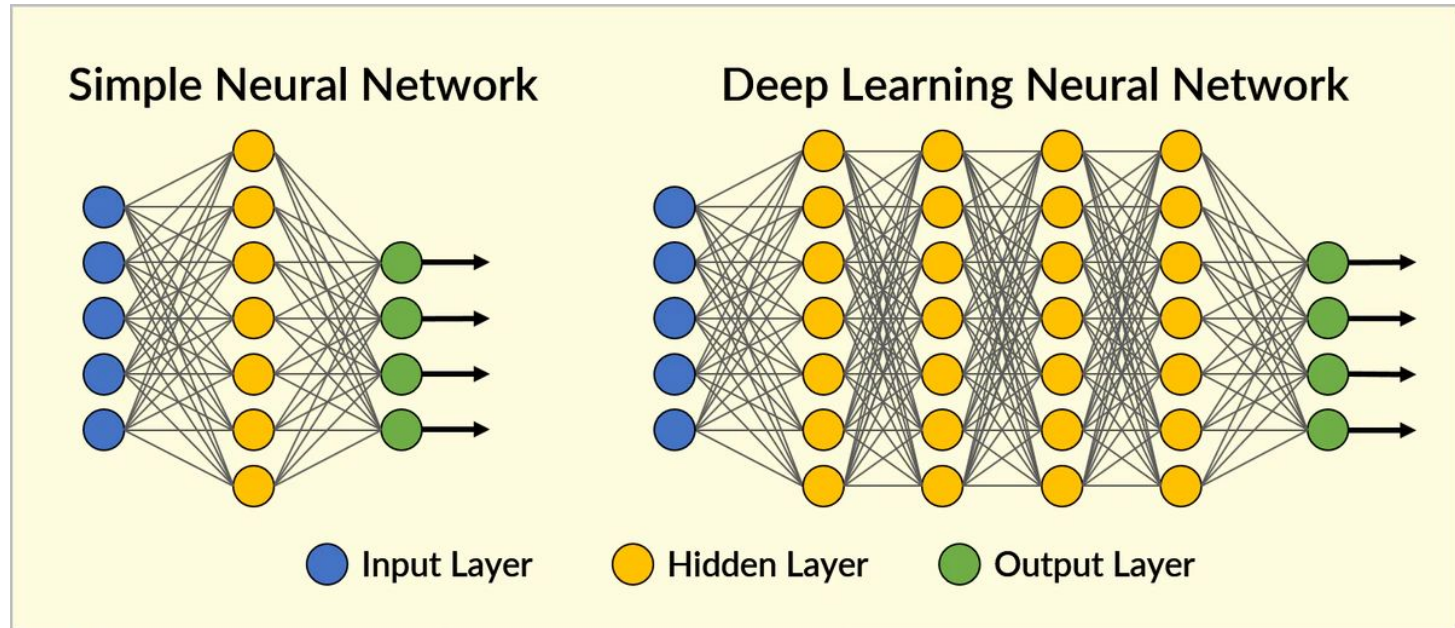




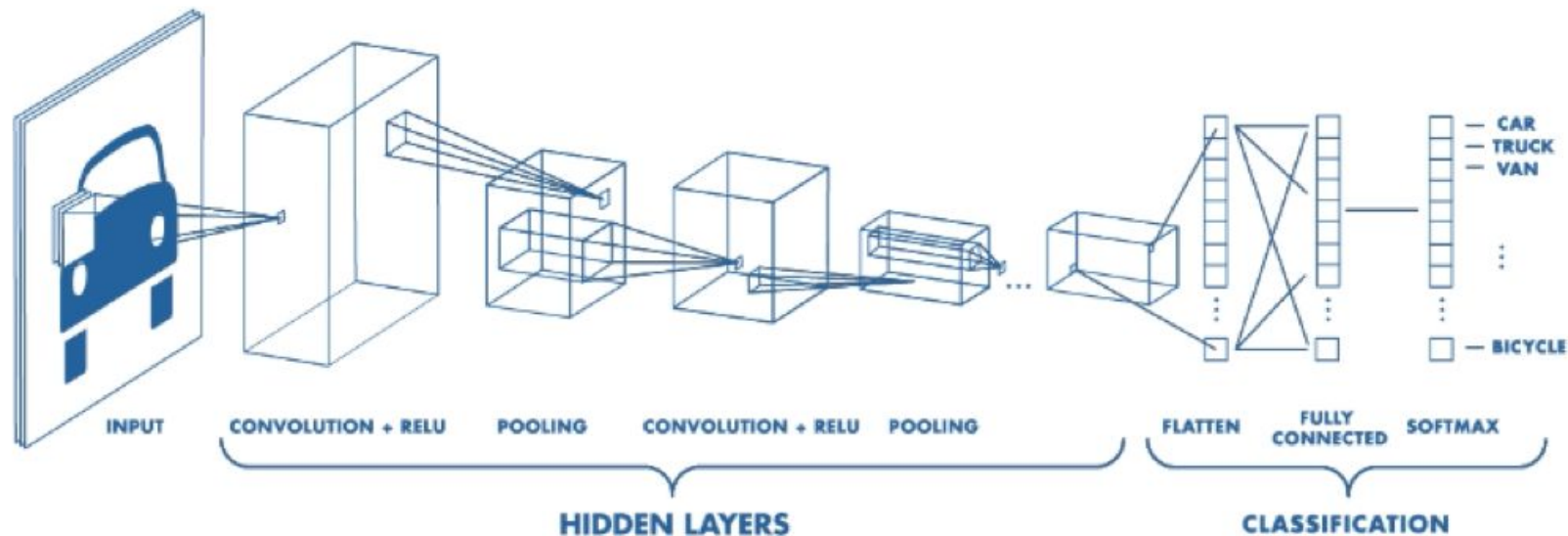
Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot



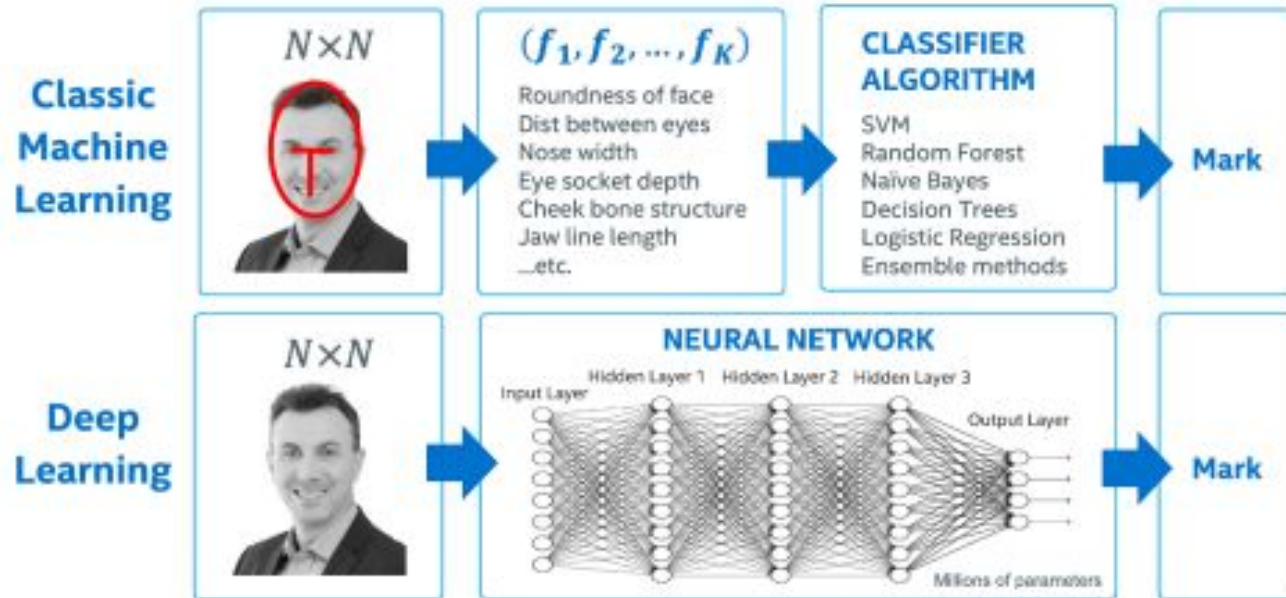
Neural network - artificially recreating the biological



Example CNN (convolutional neural network)



Deep learning NN v classic Machine learning



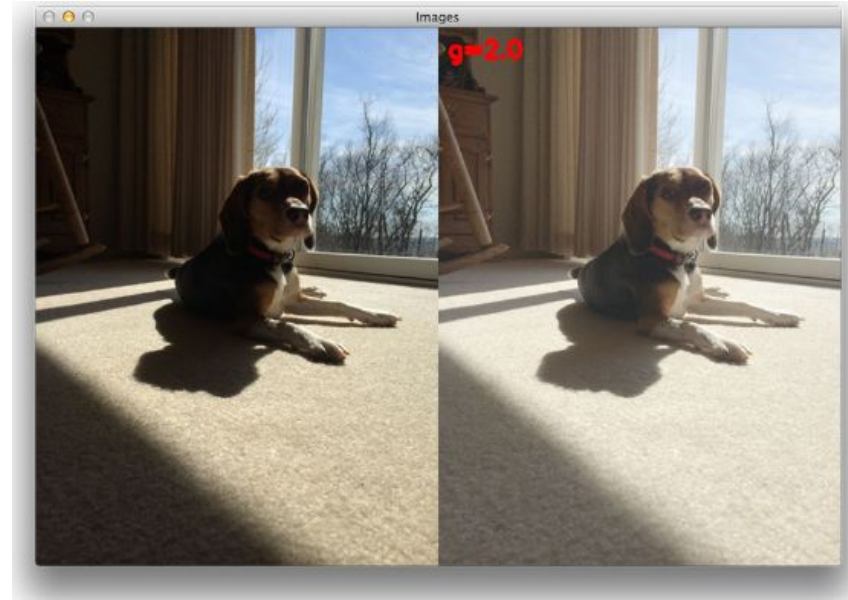


Required steps of any CV project

- Google colab (using google CPU, not your laptop)
- Keras + tensorflow ; sciPy (eg. [scipy.ndimage](#) for 3d)
- Images [pre processing](#)
- [Data_augmentation](#) processor
- Rescaling
- Dropout
- Construct a NN model appropriate for the scenario
- Train, Validation
- Inference + reusability
- Test and Evaluate


Image pre-processing

- Normalise brightness/ colour
- Crop image
- Geometric transformation
- Remove digital noise
- Filter and segment




(and by the way, you will want to deploy an enormous library of labelled training images)

Kaggle image resources

 Playground Prediction Competition

Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

 Kaggle · 213 teams · 7 years ago

[Overview](#) [Data](#) [Code](#) [Discussion](#) [Leaderboard](#) [Rules](#)

Overview

Description

Prizes

Evaluation

Winners

In this competition, you'll write an algorithm to classify whether images contain either a dog or a cat. This is easy for humans, dogs, and cats. Your computer will find it a bit more difficult.



 Dataset

LEGO Minifigures

"Do or do not. There is no try" - Jedi Master Yoda

 Yaroslav Isalenkov and 2 collaborators • updated a month ago (Version 21)


[Data](#) [Tasks \(6\)](#) [Code \(21\)](#) [Discussion \(8\)](#) [Activity](#) [Metadata](#)

[Download \(23 MB\)](#)

[New Notebook](#)

 Usability 9.4

 License Attribution 4.0 International (CC BY 4.0)

 Tags arts and entertainment, games, image data, computer vision, beginner

Description

Content

This dataset contains a lot of pictures of various LEGO Minifigures.

There are several images in different poses or with different environments for each Minifigure in the dataset. You can use this dataset for the image classification tasks or try to create more interesting things. You can try getting more accurate predictions for the test set.

Google image resources

README.md

The Quick, Draw! Dataset



IMAGENET

14,197,122 Images, 21841 synsets indexed

SEARCH

Download

- Download Image URLs
- Download Original Images
- Download Features
- Download Object Bounding Boxes
- Download Object Attributes
- API documentation
- FAQ

You can:

- ☒ Download Image URLs
- ☐ Download Original Images (for non-commercial r
- ☐ Download Features
- ☐ Download Object Bounding Boxes
- ☐ Download Object Attributes **New!**

Next

To learn more about ImageNet API, Click Here.

Other image databases

Open Images Dataset V6 + Extensions

15,851,536 boxes on 600 categories
2,785,498 instance segmentations on 350 categories
3,284,280 relationship annotations on 1,466 relationships
675,155 localized narratives
59,919,574 image-level labels on 19,957 categories
Extension - 478,000 crowdsourced images with 6,000+ categories

**COCO**
Common Objects in Context

info@cocodataset.org

Home People **Dataset** Tasks Evaluate

COCO Explorer

COCO 2017 train/val browser (123,287 Images, 886,284 Instances). Crowd labels not shown.



The Street View House Numbers (SVHN) Dataset

SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to [MNIST](#) (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images). SVHN is obtained from house numbers in Google Street View images.

Overview

10 classes, 1 for each digit. Digit '1' has label 1, '9' has label 9 and '0' has label 10.
73257 digits for training, 26032 digits for testing, and 531131 additional, somewhat less difficult samples, to use as extra training data
Comes in two formats:

1. Original images with character level bounding boxes.
2. MNIST-like 32-by-32 images centered around a single character (many of the images do contain some distractors at the sides).

Downloads

Format 1: Full Numbers: [train.tar.gz](#), [test.tar.gz](#), [extra.tar.gz](#) (Note: for non-commercial use only)



questions?