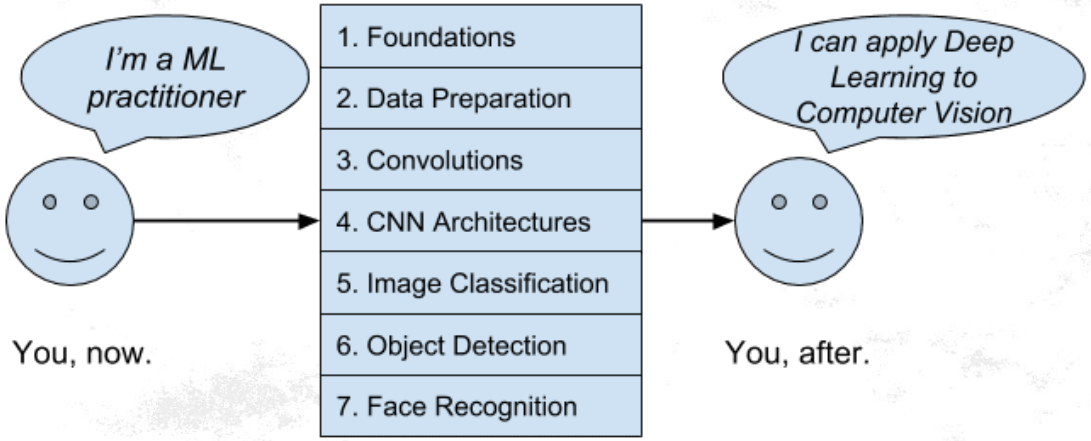
Artificial Intelligence

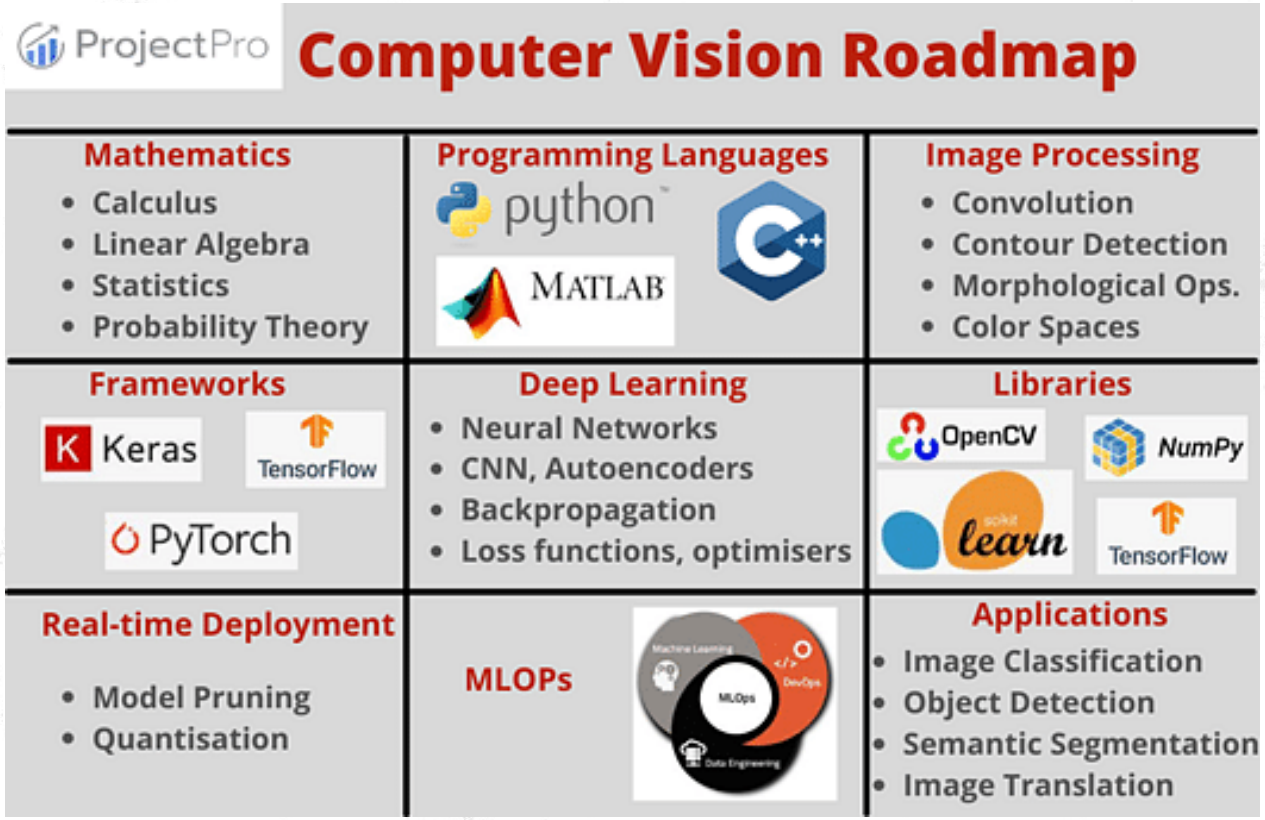
Computer Vision – Face Detection

Dr. Sarwan Singh

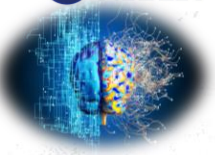
Scientist – D, NIELIT Chandigarh

Computer vision Learning Roadmap



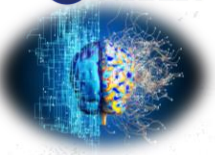


Source /



AI-Roadmap to Computer Vision

Step 1 :	Step 2 :	Step 3 :	Step 4 :	Step 5 :	Step 6 :	Step 7 :	Step 8 :	AI-Roadmap to Computer Vision
• Python & Statistics : Basics of computer vision	• Solving Image Classification : Basic Machine Learning Image pre-processing Techniques	• Introduction to Keras & Neural Networks: Fundamentals of Neural Networks (keras)	• Understanding CNNs, Transfer Learning : Convolutional Neural Networks (CNNs) Transfer learning	• Object Detection Algorithms	• Image Segmentation & Attention Models	• Deep Learning Tools : PyTorch & TensorFlow	• Generative Adversial Networks (GANs)	



Introduction to computer vision

Getting started with Image Data
- Reading, stacking
- Understanding Pixels/ colorspace
- Image processing
- Edge/blur/ HOG…

CNN
- Need, filter…
- Neural network
- Pooling, maths..
- CNN Visualization

Transfer Learning

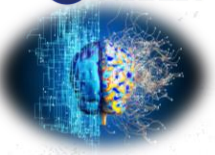
Pretrained Models

Inception

Resnets
- Densnet
- CSRnet

Object Detection

Image Generation
- GAN



AI-Roadmap to Computer Vision

1. Python & Statistics : Basics of computer vision
2. Solving Image Classification : Basic Machine Learning Image pre-processing Techniques
3. Introduction to Keras & Neural Networks: Fundamentals of Neural Networks (keras)
4. Understanding CNNs, Transfer Learning : Convolutional Neural Networks(CNNs) Transfer learning
5. Object Detection Algorithms
6. Image Segmentation & Attention Models
7. Deep Learning Tools : PyTorch & TensorFlow
8. Generative Adversial Networks (GANs)

References

- https://arxiv.org/pdf/1311.2524.pdf
- https://arxiv.org/pdf/1504.08083.pdf
- https://arxiv.org/pdf/1506.01497.pdf
- https://arxiv.org/pdf/1506.02640v5.pdf
- Deep Learning for Computer Vision- Standford
- https://pypi.org/project/face-recognition/

  - websites – machinelearningmastery.com, levelup.gitconnected.com, simplilearn.com, analyticsvidhya.com, towardsdatascience.com

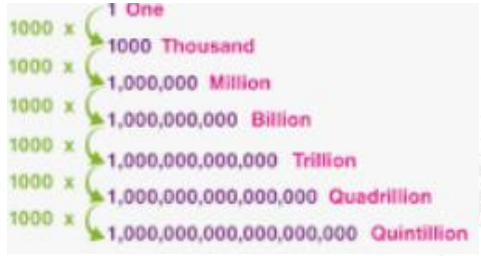
It is anticipated that the market for computer vision will approach **$41.11** billion by the year 2030, with a compound annual growth rate (CAGR) of **16.0%** between the years 2020 and 2030.

Allied Market Research

Computer Vision

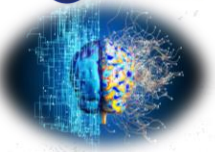
- Computer vision is one of the fields of artificial intelligence that trains and enables computers to understand the visual world.
- Computers can use digital images and deep learning models to accurately identify and classify objects and react to them.
- Computer vision in AI is dedicated to the development of automated systems that can interpret visual data (such as photographs or motion pictures) in the same manner as people do.
- The amount of data that we generate today is tremendous

  2.5 quintillion bytes ($2.5 * 10^{18}$) of data every single day.
- This growth in data has proven to be one of the driving factors behind the growth of computer vision



Billion of billion

History



1959

neurophysiologists started showing a cat a variety of sights in an effort to correlate a reaction in the animal's brain

1960's

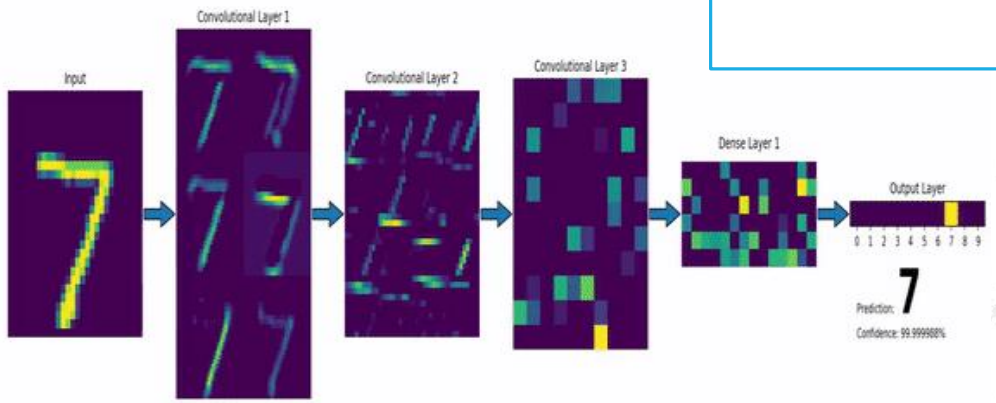
artificial intelligence (AI) emerged as an area of research, and the effort to address AI's inability to mimic human vision began.

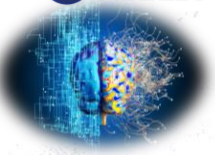
1982

Neuroscientists demonstrated, vision operates hierarchically and presented techniques enabling computers to recognize edges, vertices, arcs, and other fundamental structures

2000

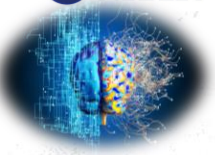
researchers were concentrating their efforts on object identification, and the industry saw the first-ever **real-time face recognition** solutions.





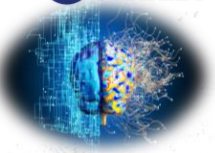
OpenCV

- Open cv is the most popular library in computer vision.
- It is originally written in C and C++, now it is available in python also.
- It is originally developed by intel.
- The library is a cross-platform open-source library. It is free to use.
- Open CV library is a highly optimized library with its main focus on real-time applications.
- Open CV is also used for Creating Face Recognition Systems.



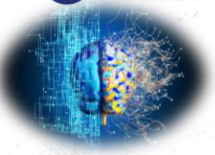
OpenCV

- The library has more than 2500 optimized algorithms.
  - which can be used to detect and recognize faces, identify objects, classify human actions using videos, tracking camera movements, tracking moving objects, extracting 3D models of objects, stitch images together to produce a high-resolution image of an entire scene,
  - find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, etc
- It has around 47 thousand people of users community and an estimated number of downloads exceeding 18 million.
- Many big companies like google, amazon, Tesla, Microsoft, Honda, etc. uses Open cv to make their products better and more AI-driven.



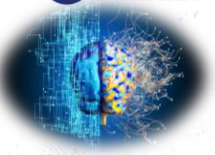
OpenCV

- OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers.
- For something like a face, you might have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course). But therein lies the problem: for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, "Is this a face? … Is this a face? … Is this a face?"
- Since there are 6,000 or more tests per block, you might have millions of calculations to do, which will grind your computer to a halt.



OpenCV

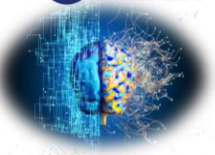
- To get around this, OpenCV uses cascades. What's a cascade? The best answer can be found in the dictionary: "a waterfall or series of waterfalls."
- Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.



OpenCV alternatives

1) Microsoft Computer Vision API
2) AWS Rekognition
3) Google Cloud Vision API
4) Scikit-Image
5) SimpleCV
6) Azure Face API
7) DeepDream
8) IBM Watson Visual Recognition
9) Clarifi
10) DeepPy

Face Recognition



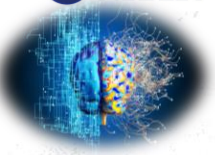
Face recognition

- Face recognition is a technique is recognition the name of the person available in the picture.
- It comprises of two steps.
  - first step is to detect the faces and
  - second step is to recognize the faces.
- Face Recognition is a library in python developed by Adam Geitgey.
- This library provides us one of the easiest and simplest methods to detect and recognize faces.
- face recognition library generates a total number of 128 digital prints for each face it detects.
- later these prints are encoded in some vector encodings that can be used later to decode the prints and compare them to fetch the label(name) of the person.

Real-World Applications of Face Recognition

Face recognition is currently being used to make the world safer, smarter, and more convenient.

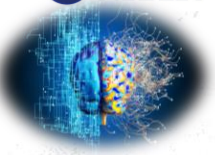
- There are a few use cases :
- Finding Missing Person
- Retail Crime
- Security Identification
- Identifying accounts on social media
- School Attendance System
- Recognizing Drivers in Cars



Traditional Face Recognition Algorithms

During the 1990s holistic approaches were used for face recognition. Handcrafted local descriptors became popular In the early 1920s, and then the local feature learning approaches were followed in the late 2000s. Nowadays algorithms that are widely used and are implemented in OpenCV are as follows:

- Eigenfaces (1991)
- Local Binary Patterns Histograms (LBPH) (1996)
- Fisherfaces (1997)
- Scale Invariant Feature Transform (SIFT) (1999)
- Speed Up Robust Features (SURF) (2006)



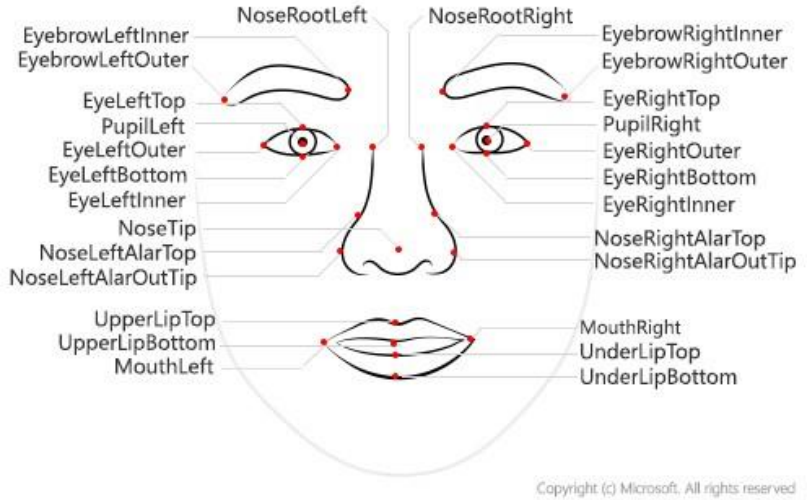
Traditional Face Recognition Algorithms

- Each method follows a different approach to extracting the image information and matching it with the input image.
- Fischer-faces and Eigenfaces have almost similar approaches as well as SURF and SIFT.
- LBPH is a simple yet very efficient method but it's slow compared to modern days face -recognizers.
- These algorithms are not faster compared to modern days face-recognition algorithms. Traditional algorithms can't be trained only by taking a single picture of a person

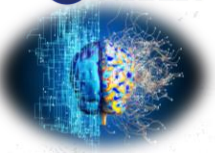
Deep Learning for Face Recognition

- Some of the widely used Deep Learning-based

  Face Recognition systems are as follows:
  - DeepFace
  - **DeepID** series of systems
  - VGGFace
  - FaceNet



- Face recognizers generally take face images and find the important points such as the corner of the mouth, an eyebrow, eyes, nose, lips, etc. Coordinates of these points are called facial-features points, there are such 66 points. In this way, a different technique for finding feature points give different results.

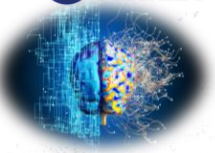


Deep Learning for Face Recognition

Steps involved in a face recognition model:

- **Face Detection:** Locate faces and draw bounding boxes around faces and keep the coordinates of bounding boxes.
- **Face Alignments:** Normalize the faces to be consistent with the training database.
- **Feature Extraction:** Extract features of faces that will be used for training and recognition tasks.
- **Face Recognition:** Matching of the face against one or more known faces in a prepared database.

In the traditional method of face recognition, we had separate modules to perform these four steps, which was painful.



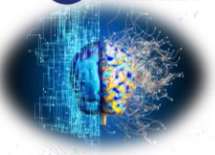
Challenges in Recognition Systems

These are significant challenges faced by recognition systems and need to be resolved.

- **Pose:** Recognition systems are susceptible to the human pose. Facial recognition systems will not be able to predict if the person's face is not visible.
- **Illumination:** Illumination changes the face contours drastically. Pictures for face recognition should be taken in proper lighting conditions.
- **Facial Expressions**: Different facial expressions can result in different predictions of the same person's Image.
- **Low Resolution**: Low-resolution pictures contain less information, hence not good for face recognition training.

- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

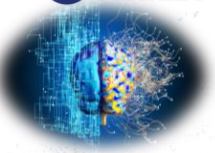
- paper.dvi (iitd.ac.in) Voila Jones 2001



Face detection

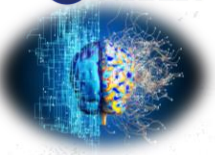
- Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras.
- However, much more reliable solutions exist now. We're going to use a method invented in 2005 called Histogram of Oriented Gradients — or just ***HOG*** for short.
- To find faces in an image, we'll start by making our image black and white because we don't need color data to find faces

Source : Face Detection with Haar Cascade. | Towards Data Science



Face detection

- *According to Wikipedia…* ***Woody Bledshoe, Helen Chan Wolf, and Charles Bisson were the first ones to do the first ever Face Detection on a Computer back in the 1960s****. A person had to manually pinpoint the coordinates of facial features such as the pupil centers, the inside and outside corner of eyes, and the widows peak in the hairline. The coordinates were used to calculate 20 distances, including the width of the mouth and of the eyes. A human could process about 40 pictures an hour in this manner and so build a database of the computed distances. A computer would then automatically compare the distances for each photograph, calculate the difference between the distances and return the closed records as a possible match.*

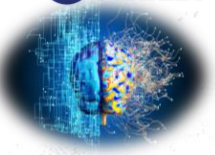


What is Haar Cascade

- ***It is an Object Detection Algorithm used to identify faces in an image or a real time video.***
- The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001.
- The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.
- The github OpenCV repository has the models stored in XML files, and can be read with the OpenCV methods. These include models for face detection, eye detection, upper body and lower body detection, license plate detection etc

Working of Haar Cascade

- how a haar feature traverses on an image from its left towards its right.



- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- https://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf
- https://github.com/ageitgey/face_recognition?tab=readme-ov-file#face-recognition
- https://www.youtube.com/watch?v=cPmjQ9V6Hbk
- https://www.wolfib.com/Image-Recognition-Intro-Part-1/
- https://www.freecodecamp.org/news/how-to-build-a-simple-image-recognition-system-with-tensorflow-part-2-c83348b33bce/
- https://colab.research.google.com/github/Hvass-Labs/TensorFlow-Tutorials/blob/master/06_CIFAR-10.ipynb