



From the days of ENIAC to today. Progress  
has been tremendous!

Where is AI...how is AI progressing

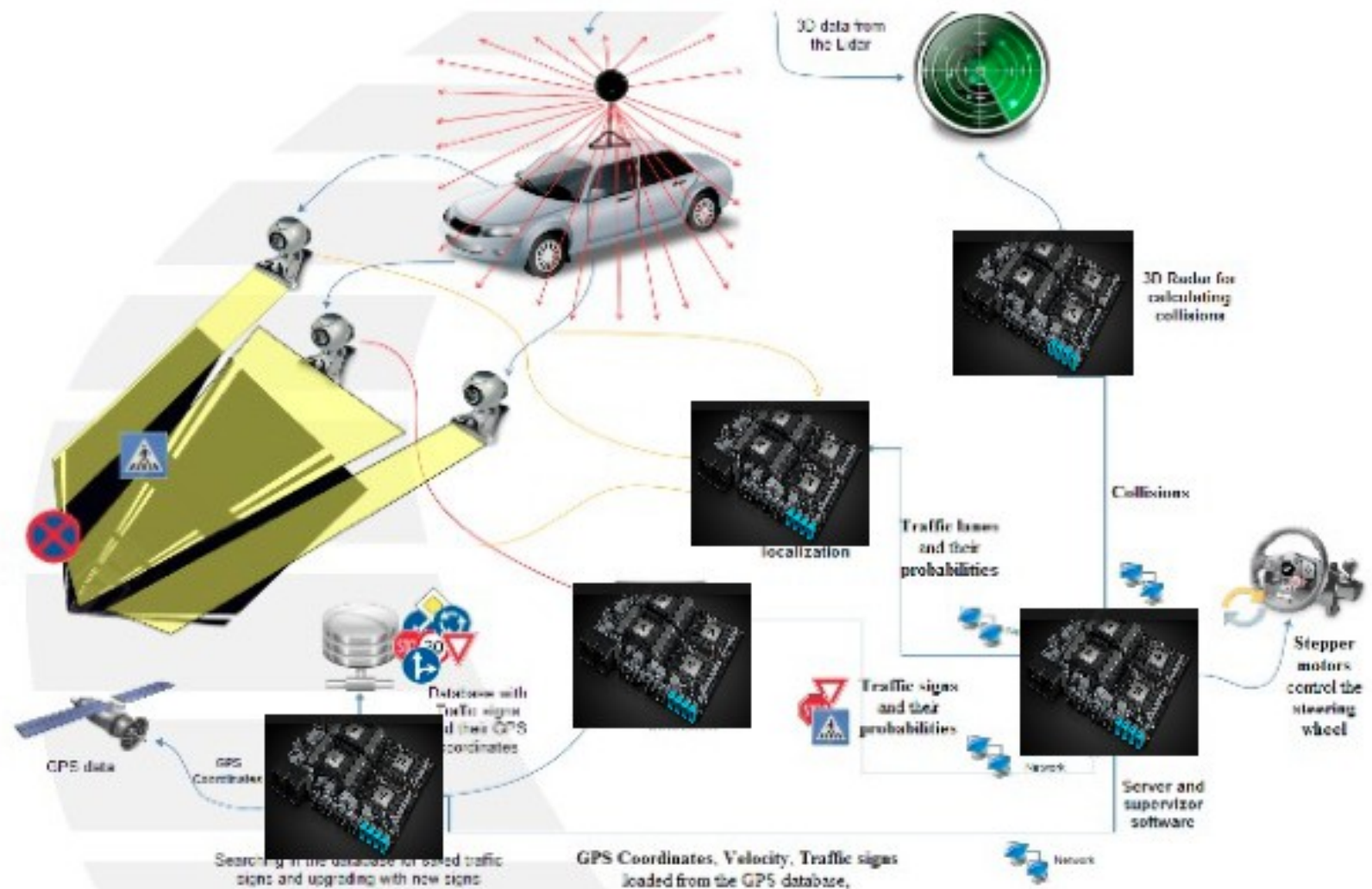
Lets See AI in pictures!



# Nsynth Super: NN creates new music!

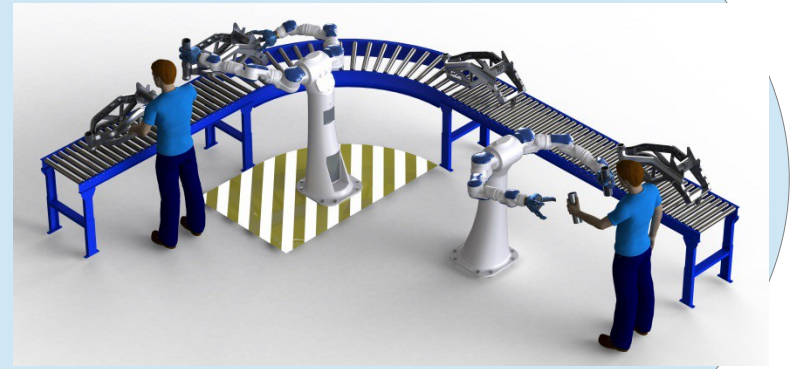


# Simplified view of Autonomous car system



# Continental AG's SMART Factory

- Active RFID tags and Geo-location are used to move the tire components throughout the factory
- Collaborative robots
  - Robots are “shown” how to do a task once and then they can repeat that action
  - Reduces risks of injuries and reduces the need for additional assisting employees



# SNIPER



GENERAL ATOMICS, MQ-9 REAPER



# AMAZON ECHO: YOUR BUTLER

IS JEFF BEZOS A NICE GUY?

IS ARSENAL WINNING THE LEAGUE?

SHOULD I WATCH LA-LA LAND?

PLAY MY FAVOURITE MILES TUNE

ORDER AN UBER TO LEAVE IN HALF AN HOUR

GET ME A MARGHARITA FROM DOMINO'S

WHAT'S THE WEATHER ON MONDAY IN HULL?

SET A REMINDER IN 90 MINUTES...



# The Dartmouth Conference and the Name Artificial Intelligence

J. McCarthy, M. L. Minsky, N. Rochester, and C.E. Shannon. August 31, 1955. "We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

And , AI was born!



# The Origins of AI Hype

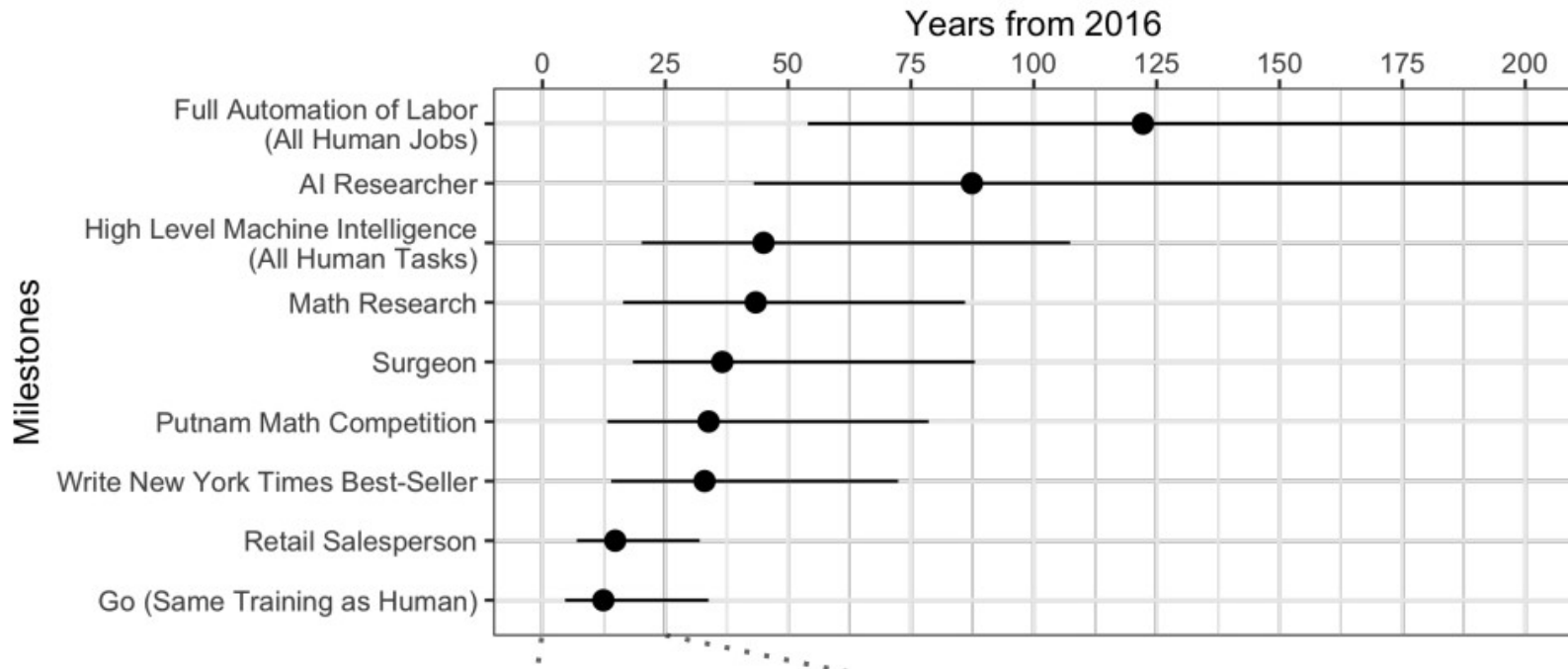
1950     Turing predicted that in about fifty years "an average interrogator will not have more than a 70 percent chance of making the right identification after five minutes of questioning".

1957     Newell and Simon predicted that "Within ten years a computer will be the world's chess champion, unless the rules bar it from competition."

# AI History

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1952–69 Look, Ma, no hands!
- 1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1956 Dartmouth meeting: "Artificial Intelligence" adopted
- 1965 Robinson's complete algorithm for logical reasoning
- 1966–74 AI discovers computational complexity  
Neural network research almost disappears
- 1969–79 Early development of knowledge-based systems
- 1980–88 Expert systems industry booms
- 1988–93 Expert systems industry busts: "AI Winter"
- 1985–95 Neural networks return to popularity
- 1988– Resurgence of probabilistic and decision-theoretic methods  
Rapid increase in technical depth of mainstream AI  
"Nouvelle AI": ALife, GAs, soft computing

# Will Intelligent machines surpass human beings





# Will Intelligent machines surpass human beings

Ellon Musk says ““Robots will do everything better than us“

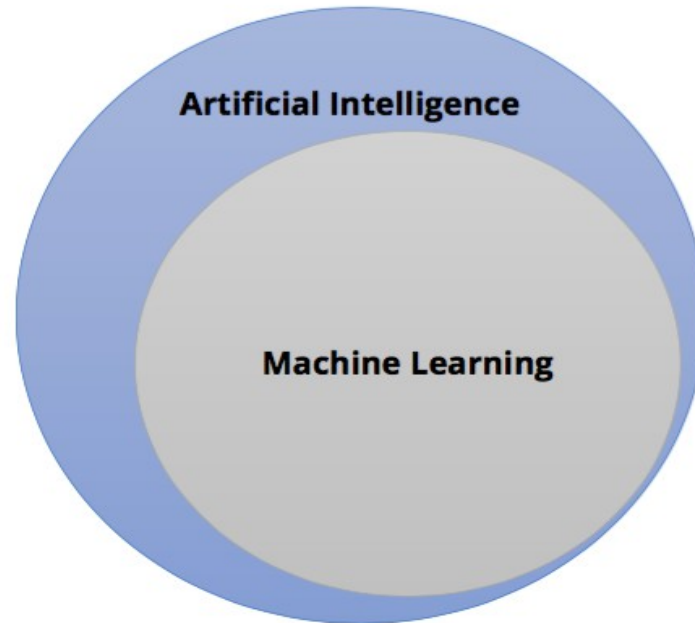
"There certainly will be job disruption. Because what's going to happen is robots will be able to do everything better than us. ... I mean all of us," saidÂ  
Musk, speaking to the National Governors AssociationÂ in July. "Yeah, I am not sure exactly what to do about this. This is really the scariest problem to me, I will tell you."

# What is AI? (Cont'd)

<p>“The exciting new effort to make computers think ... <i>machines with minds</i>, in the full and literal sense” (Haugeland, 1985)</p> <p>“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...” (Bellman, 1978)</p>	<p>“The study of mental faculties through the use of computational models” (Charniak and McDermott, 1985)</p> <p>“The study of the computations that make it possible to perceive, reason, and act” (Winston, 1992)</p>				
<p>“The art of creating machines that perform functions that require intelligence when performed by people” (Kurzweil, 1990)</p> <p>“The study of how to make computers do things at which, at the moment, people are better” (Rich and Knight, 1991)</p>	<p>“A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes” (Schalkoff, 1990)</p> <p>“The branch of computer science that is concerned with the automation of intelligent behavior” (Luger and Stubblefield, 1993)</p>				
<p><b>Figure 1.1</b> Some definitions of AI. They are organized into four categories:</p> <table> <tr> <td>Systems that think like humans.</td><td>Systems that think rationally.</td></tr> <tr> <td>Systems that act like humans.</td><td>Systems that act rationally.</td></tr> </table>		Systems that think like humans.	Systems that think rationally.	Systems that act like humans.	Systems that act rationally.
Systems that think like humans.	Systems that think rationally.				
Systems that act like humans.	Systems that act rationally.				

# What is Machine Learning?

Machine Learning is an approach or subset of Artificial Intelligence that is based on the idea that machines can be given access to data along with the ability to learn from it.



“

The capability of Artificial Intelligence systems to learn by extracting patterns from data is known as Machine Learning.

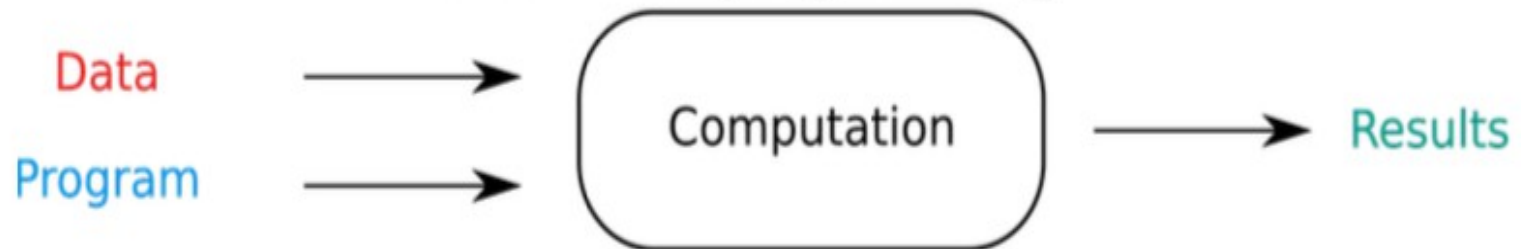
”



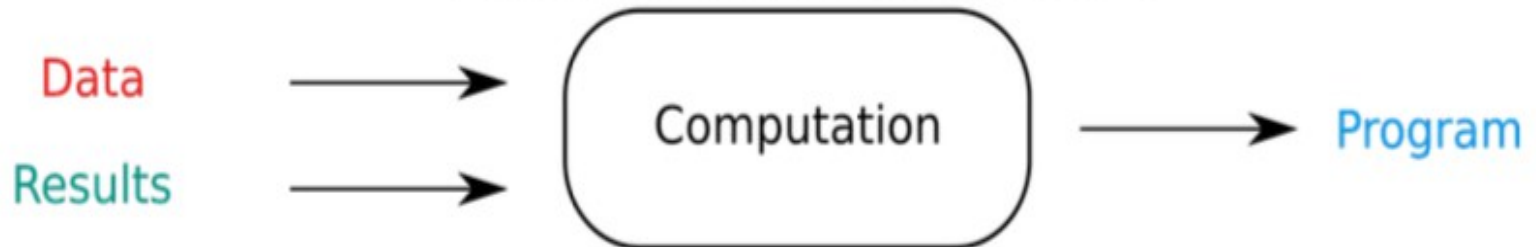
# Traditional Programming vs. Machine Learning Approach

---

## Traditional programming

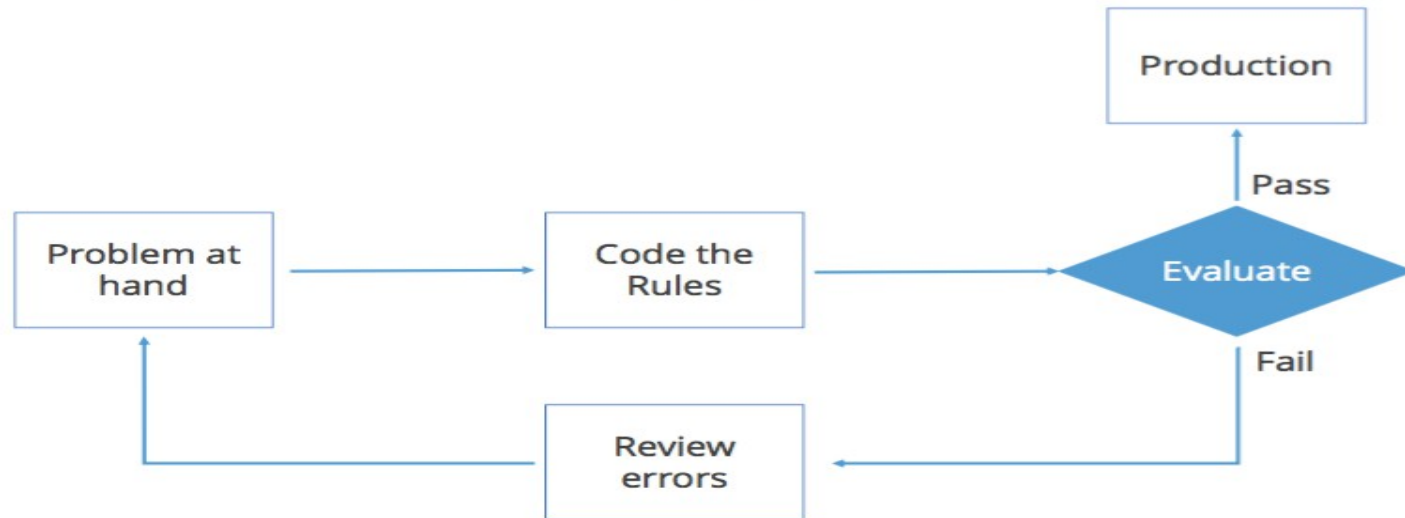


## Machine Learning Approach



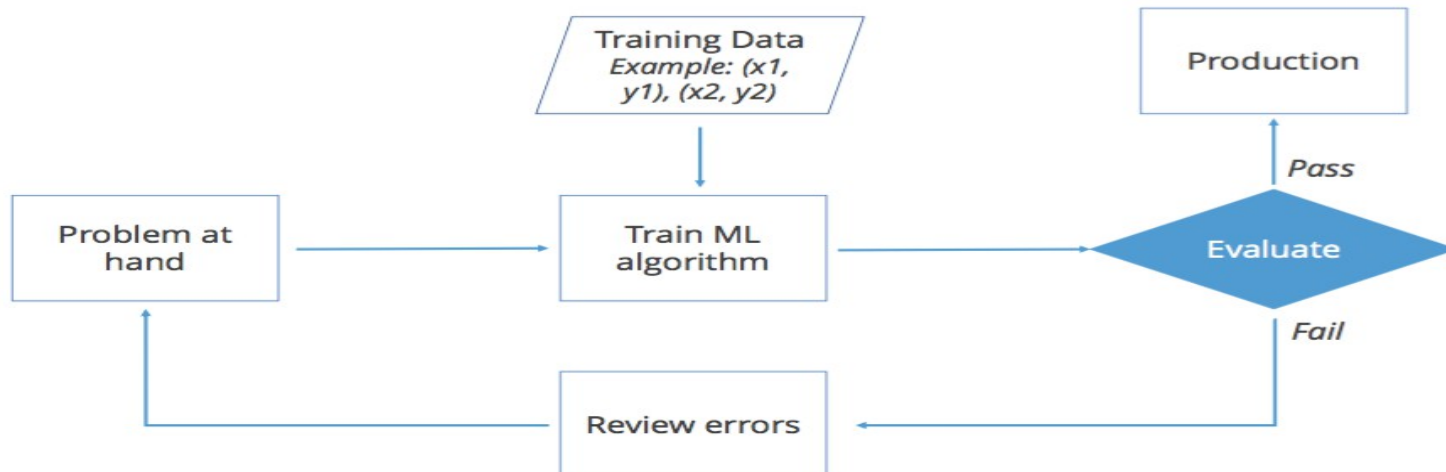
## Traditional Approach

Traditional programming relies on hard-coded rules.



## Machine Learning Approach

Machine Learning relies on learning patterns based on sample data.



# How can machines learn?

$$y = f(X)$$

## Learning Approaches



**Supervised Learning:** Learning with a **labeled training set**  
*Example: email spam detector with training set of already labeled emails*



**Unsupervised Learning:** **Discovering patterns** in unlabeled data  
*Example: cluster similar documents based on the text content*



**Reinforcement Learning:** learning based on **feedback** or reward  
*Example: learn to play chess by winning or losing*



# Inductive (Supervised) Learning

Basic Problem: Induce a representation of a function (a systematic relationship between inputs and outputs) from examples.

**target function**  $f: X \rightarrow Y$

**example**  $(x, f(x))$

**hypothesis**  $g: X \rightarrow Y$  such that  $g(x) = f(x)$

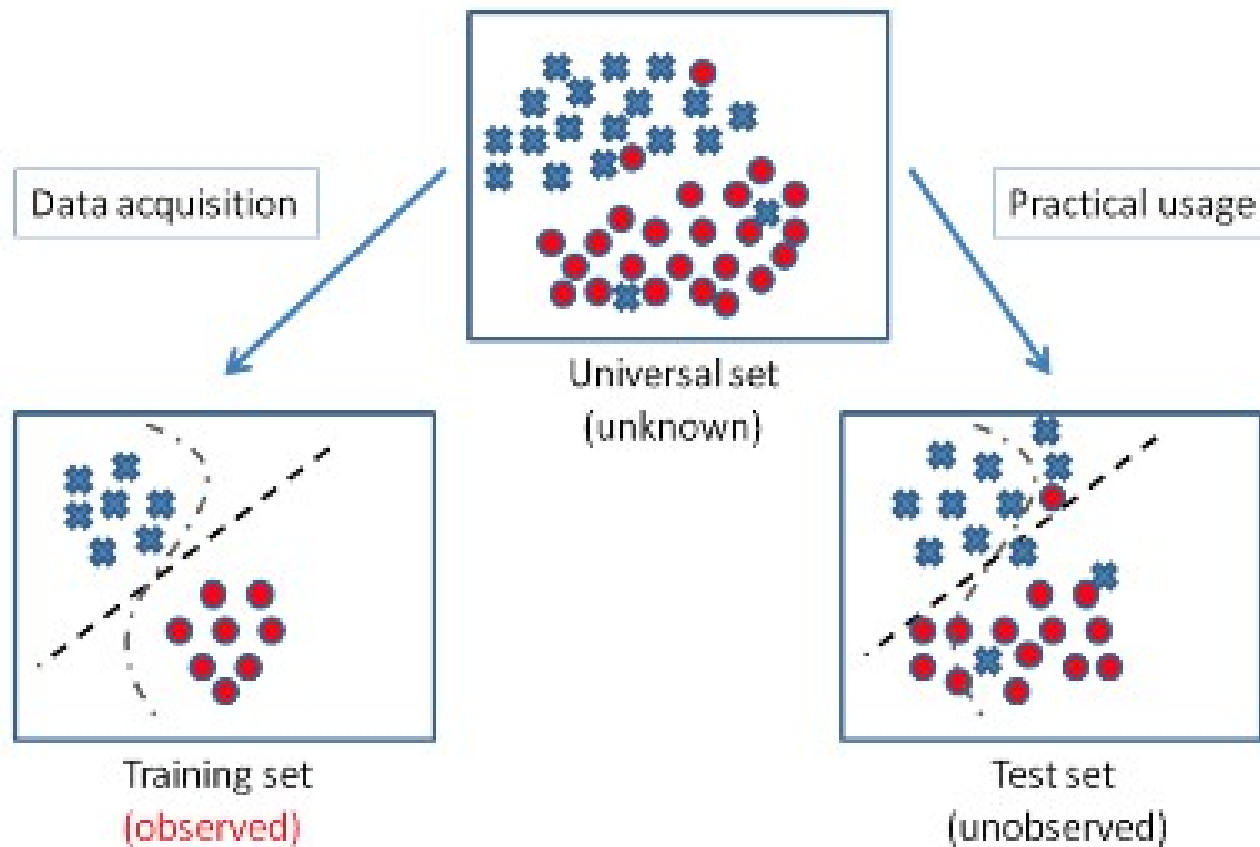
$x$  = set of attribute values (*attribute-value representation*)

$x$  = set of logical sentences (*first-order representation*)

$Y$  = set of discrete labels (*classification*)

$Y = \mathbb{R}$  (*regression*)

# Learning : Training and Test Set



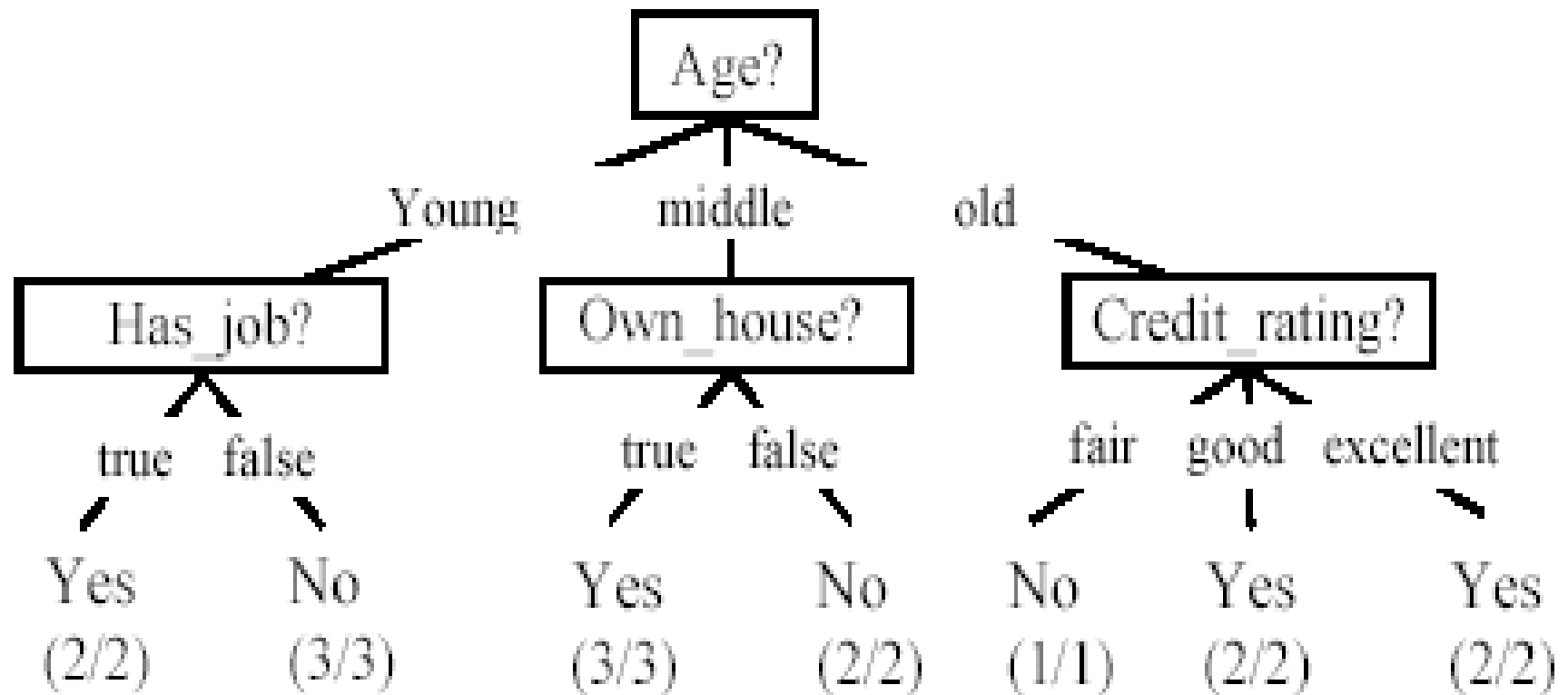
Lets look at a usecase  
Loan application approval



This is a loan application data.....  
Lets analyze

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

# A decision tree from the loan data



Hey....wait ....Is that correct?

How do we build correct and optimal tree?

How do we find out amount of information in  
a dataset?

Which part of data contains more  
information?

# Entropy measure: let us get a feeling

1. The data set  $D$  has 50% positive examples ( $\Pr(\text{positive}) = 0.5$ ) and 50% negative examples ( $\Pr(\text{negative}) = 0.5$ ).

$$\text{entropy}(D) = -0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 = 1$$

2. The data set  $D$  has 20% positive examples ( $\Pr(\text{positive}) = 0.2$ ) and 80% negative examples ( $\Pr(\text{negative}) = 0.8$ ).

$$\text{entropy}(D) = -0.2 \times \log_2 0.2 - 0.8 \times \log_2 0.8 = 0.722$$

3. The data set  $D$  has 100% positive examples ( $\Pr(\text{positive}) = 1$ ) and no negative examples, ( $\Pr(\text{negative}) = 0$ ).

$$\text{entropy}(D) = -1 \times \log_2 1 - 0 \times \log_2 0 = 0$$

- For classification problems, data with entropy 0 or close to 0 is rather useless. ML can't learn to distinguish between positive and negative samples using that data!

# An example

entropy(D)

$$-\left(\frac{6}{15} \times \log_2\left(\frac{6}{15}\right) - \left(\frac{9}{15} \times \log_2\left(\frac{9}{15}\right)\right)\right)$$

ID

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

Age	Has_Job	Own_House	Credit_Rating	Class
young	false	false	fair	No
young	false	false	excellent	No
young	true	false	good	Yes
young	true	true	good	Yes
young	false	false	fair	No
middle	false	false	fair	No
middle	false	false	good	No
middle	true	true	good	Yes
middle	false	true	excellent	Yes
middle	false	true	excellent	Yes
old	false	true	excellent	Yes
old	false	true	good	Yes
old	true	false	good	Yes
old	true	false	excellent	Yes
old	false	false	fair	No

$$\frac{6}{15} \times \left( -\left(\frac{6}{6} \times \log_2\left(\frac{6}{6}\right) - (0 \times \log_2(0))\right) + \frac{9}{15} \times \left( -\left(\frac{3}{9} \times \log_2\left(\frac{3}{9}\right) - \left(\frac{6}{9} \times \log_2\left(\frac{6}{9}\right)\right)\right) \right)$$

Age	Yes	No	entropy(Di)
young	2	3	0.971
middle	3	2	0.971
old	4	1	0.722

Calculate entropy for  
Age, has job and credit  
rating

$$gain(D, \text{Age}) = 0.971 - 0.888 = 0.083$$

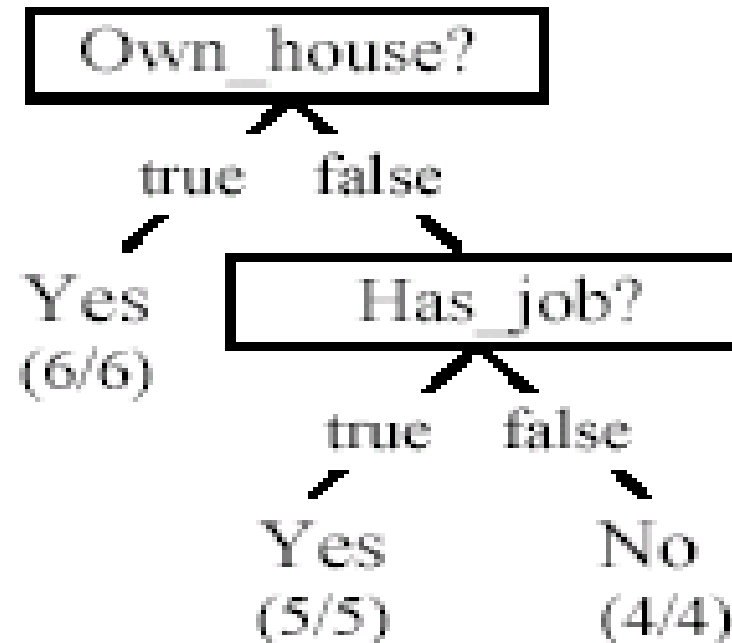
$$gain(D, \text{Own\_house}) = 0.971 - 0.551 = 0.420$$

$$gain(D, \text{Has\_Job}) = 0.971 - 0.647 = 0.324$$

$$gain(D, \text{Credit\_Rating}) = 0.971 - 0.608 = 0.363$$



We build the final tree



Phew that was too much Maths.Lets Automate it  
via AI!

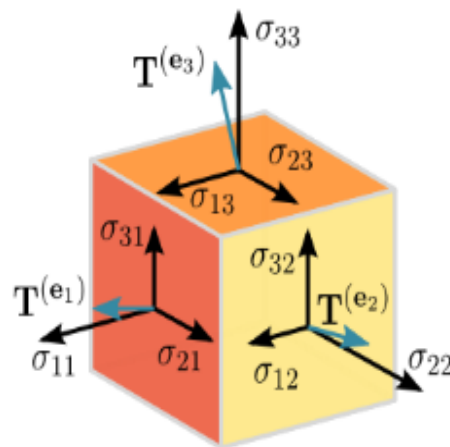
And , it just takes 20 lines of code





# AI behind google rankings

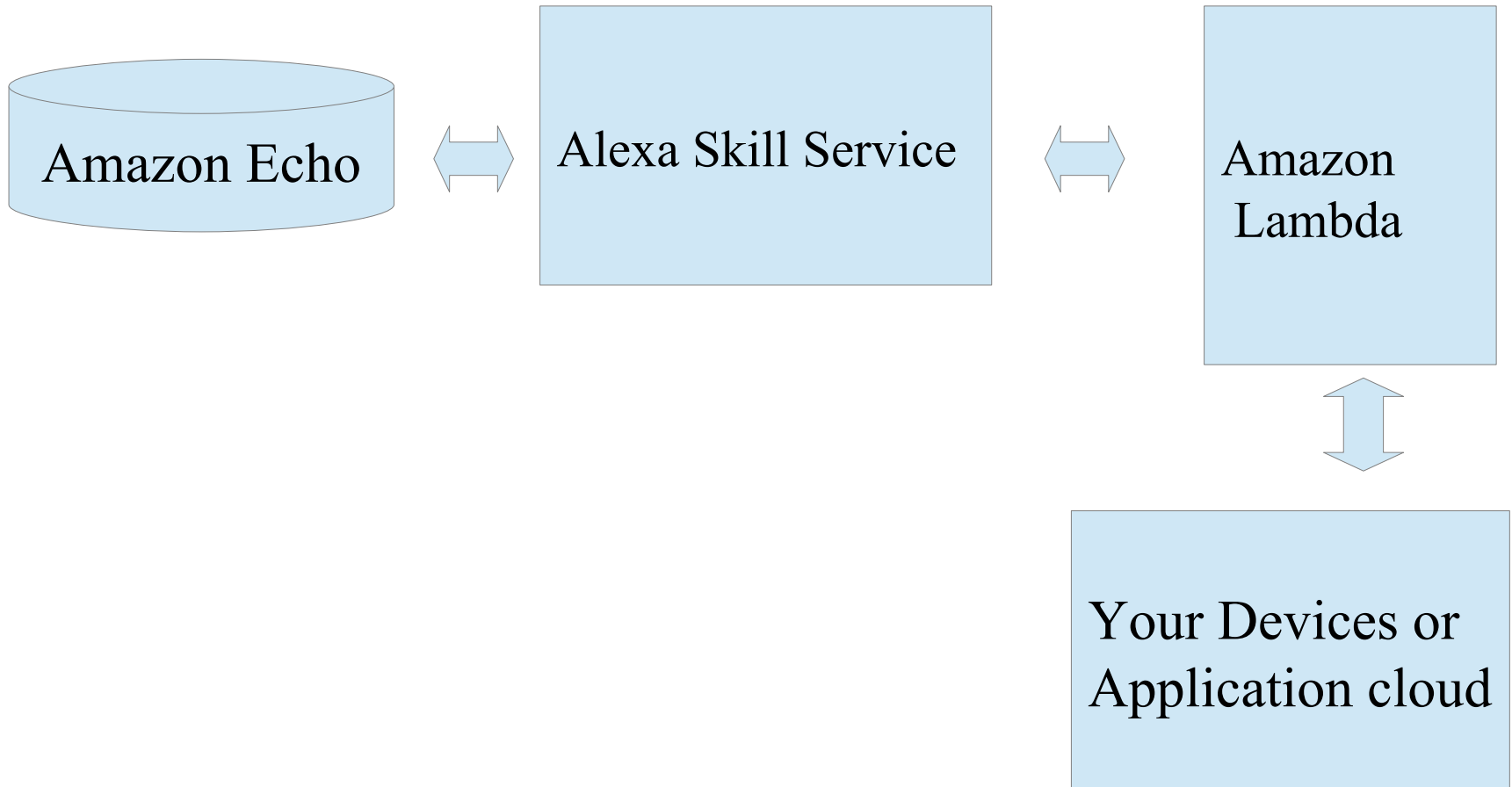
It was November 9, 2015, the day Google publicly released [TensorFlow](#). TensorFlow is an (now) open-source software library for machine intelligence. It is, in fact, the [library that powers](#) most of Google's technology like Gmail, Photos, Voice and RankBrain.



TensorFlow originally was released as an evolution of Google's internal neural network training framework "DistBelief" by the Google Brain team. On the simplest level, TensorFlow enables the large-scale and parallel manipulation of "Tensors," multi-dimensional arrays that carry vectorized data.



# Human like interactions ( eg Alexa )

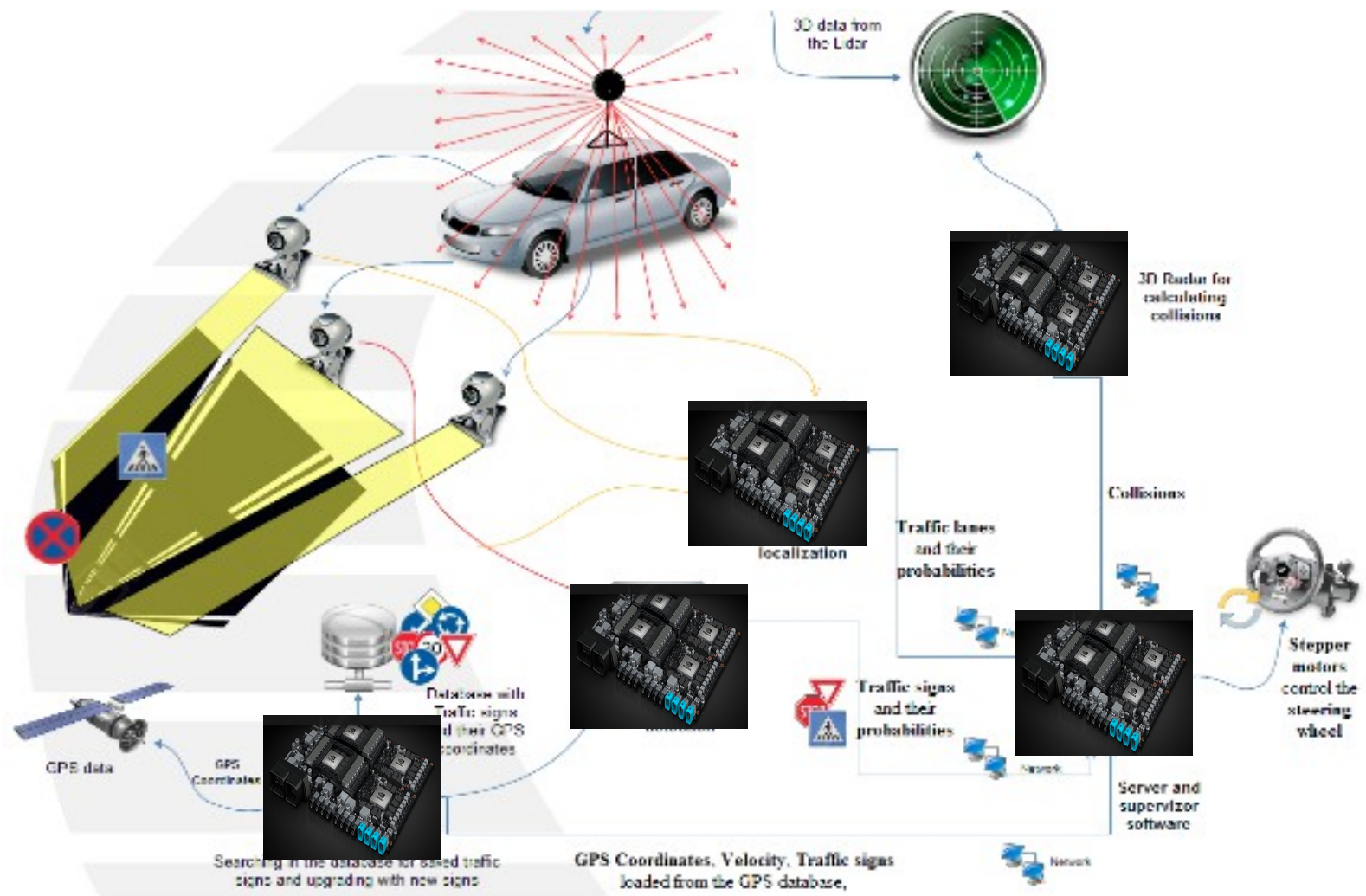


# AI for Autonomous Cars



**Uber builds and tests self-driving cars at engineering centre in Pittsburgh**

# Simplified view of Autonomous car system



# Example platform: NVIDIA

## DRIVE Linux Supported Features

- Kernel version 4.4.38 kernel with RT\_PREEMPT patches
- Support for 64-bit user space and runtime libraries
- NvMedia APIs for HW accelerated multimedia and camera input processing
- NVIDIA CUDA 8.0 parallel computing platform
- Graphics APIs:
  - OpenGL 4.5
  - OpenGL ES 3.2
  - EGL 1.5 with EGLStream extensions
- Ubuntu 16.04 LTS target Root File System (RFS)

[Read More >](#)

### TensorRT

NVIDIA TensorRT™ is a high performance neural network inference engine for production deployment of deep learning applications. TensorRT can be used to rapidly optimize, validate and deploy a trained neural network for inference to hyperscale data centers, embedded, or automotive product platforms.

[Read More >](#)

### cuDNN

NVIDIA CUDA® Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers.

[Read More >](#)

## Software Tools

Find the right tools for your needs. Our repository has the latest versions of compatible tools available for immediate download, and new updates are added on a regular basis.

Follow the links below to get started.

### Tegra® System Profiler

Tegra System Profiler is a multi-core CPU sampling profiler that provides an interactive view of captured profiling data, helping improve overall application performance.

[Read More >](#)

### Tegra® Graphics Debugger

Tegra Graphics Debugger is a console-grade tool that allows you to debug and optimize your OpenGL and OpenGL ES applications, enabling you to get the latest, most advanced GPU features.

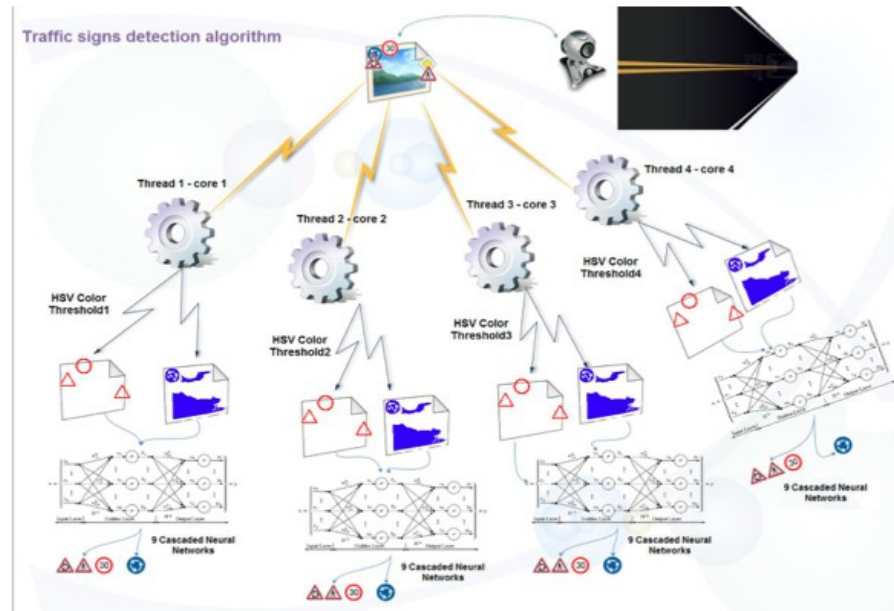
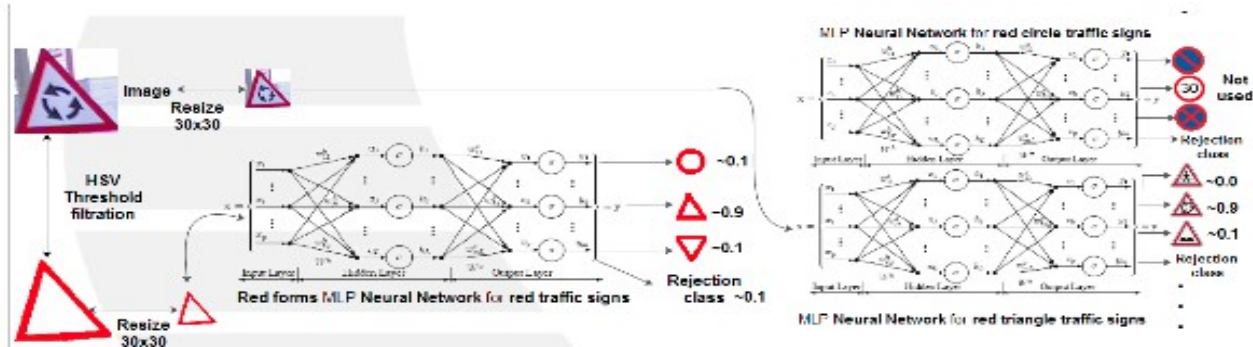
[Read More >](#)

### PerfKit

NVIDIA PerfKit is a comprehensive suite of performance tools to help debug and profile OpenGL and Direct3D applications.

[Read More >](#)

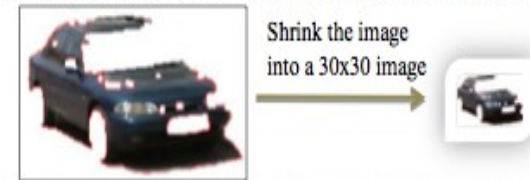
# Traffic signs detection



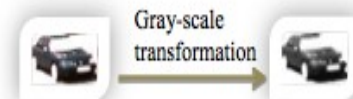
# Real time vehicle detection



In order to check whether there is a car in the retrieved image from the segmentation, the algorithm uses a Multi-Layer-Perceptron Neural Network. As the resolution of the cropped image is very high, (for computers high resolution images are mostly redundant), the algorithm conceived by me will shrink the image into a small image  $30 \times 30$ .



The Feed-forward Neural Network will be able only to classify and recognize whether in that image it is a car, so the colors are redundant.





Thank You!