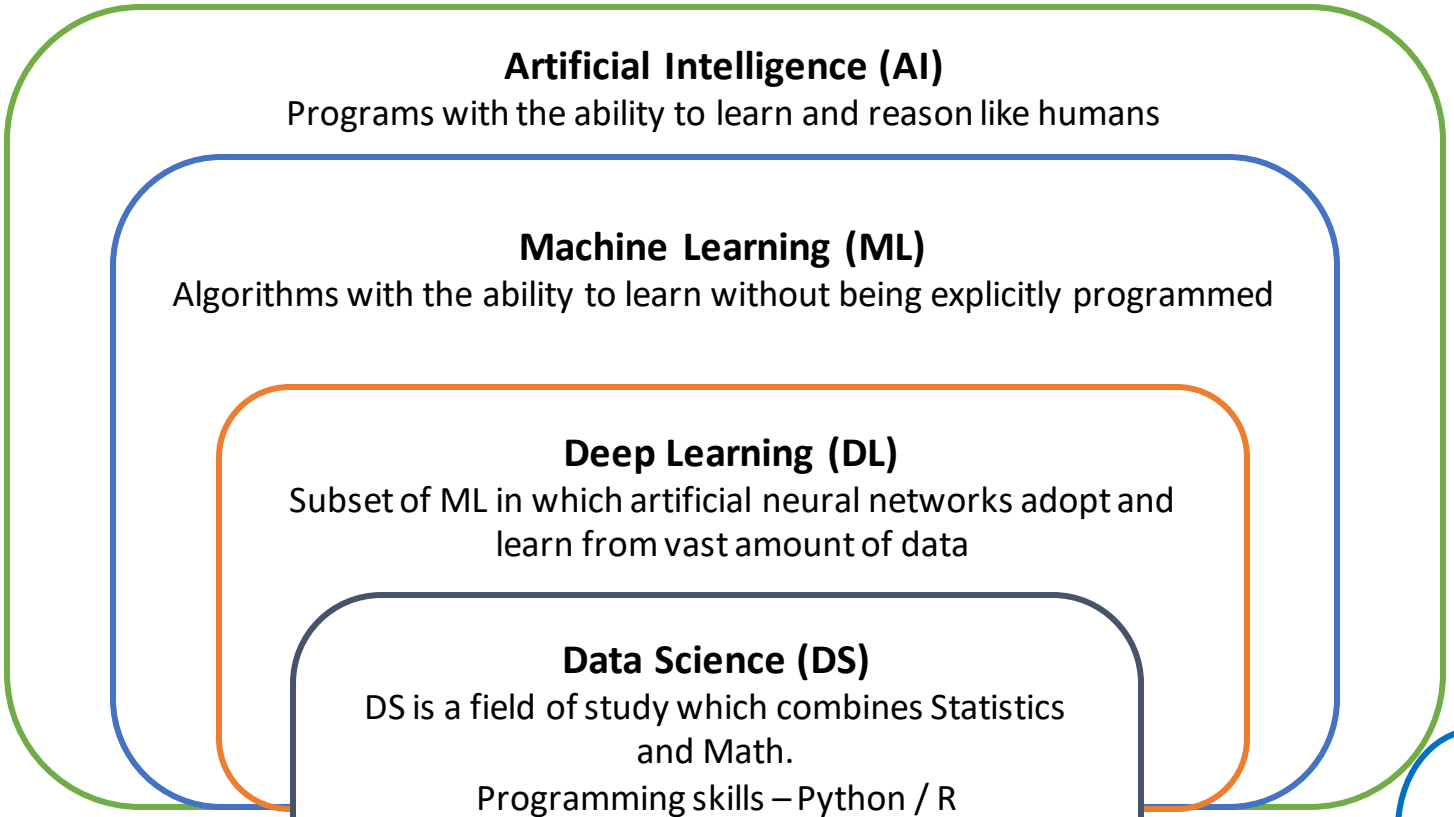
# Agenda

1. What is AI, ML, DL & DS – Terminology Know How
2. Machine Learning Overview
3. Life Cycle of a Typical ML Model
4. AI Adaptability in Semiconductors
5. Top Trends in Semiconductors
6. Why AI makes sense in Semiconductors
7. Semiconductor Ideal Expectations from AI
8. AI Technology Verticals to meet The Expectations
9. Digital Image Processing – CNN Architecture

# What is AI, ML, DL and DS




**Data Science (DS)**

DS is a field of study which combines Statistics and Math.

Programming skills – Python / R

And most importantly Domain expertise to extract meaningful insights from data

---

Statistics | Domain Knowledge | Programming Skills

**~80%**

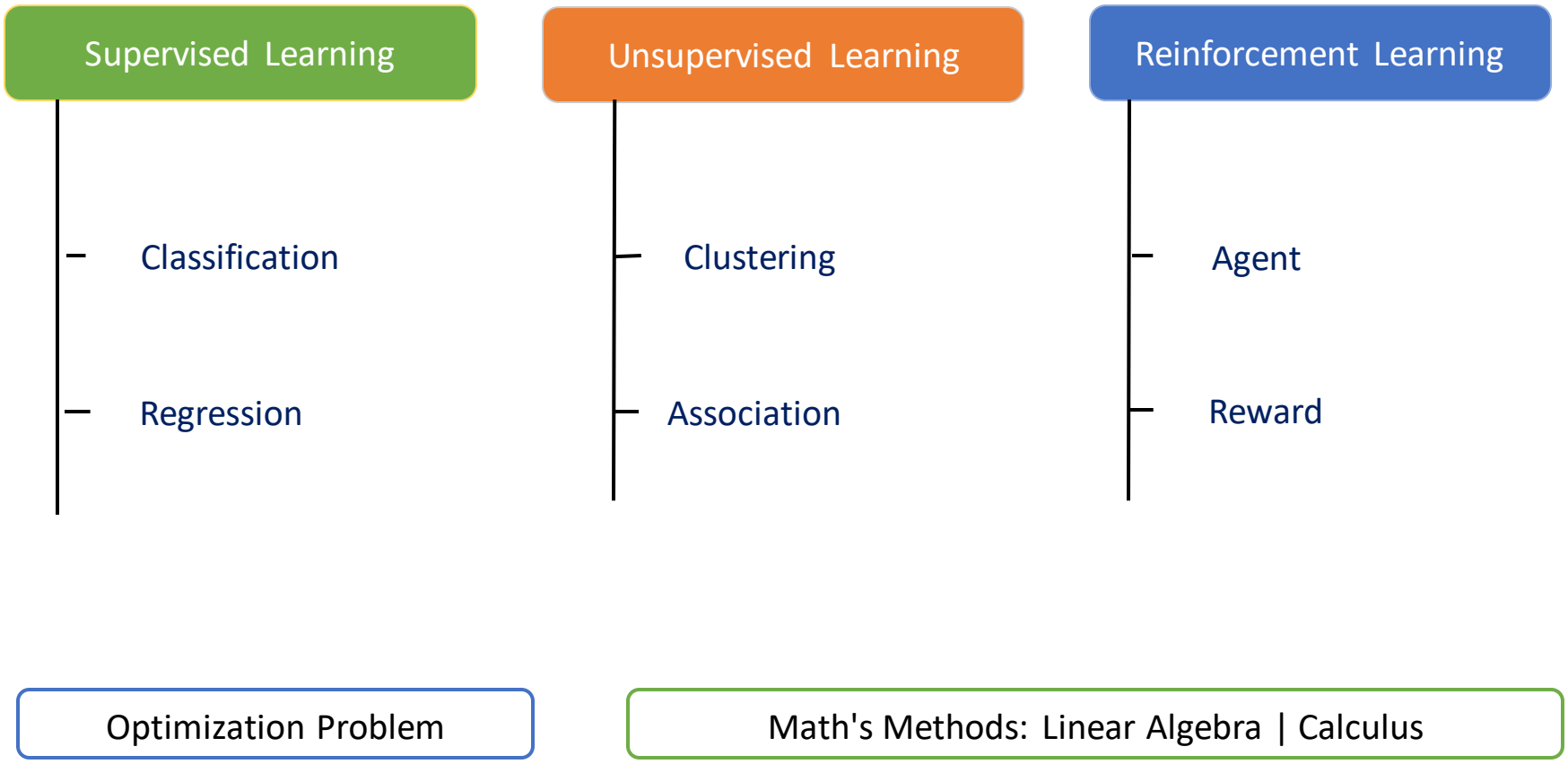
~80% of World's Data generated in last 10 Years

---

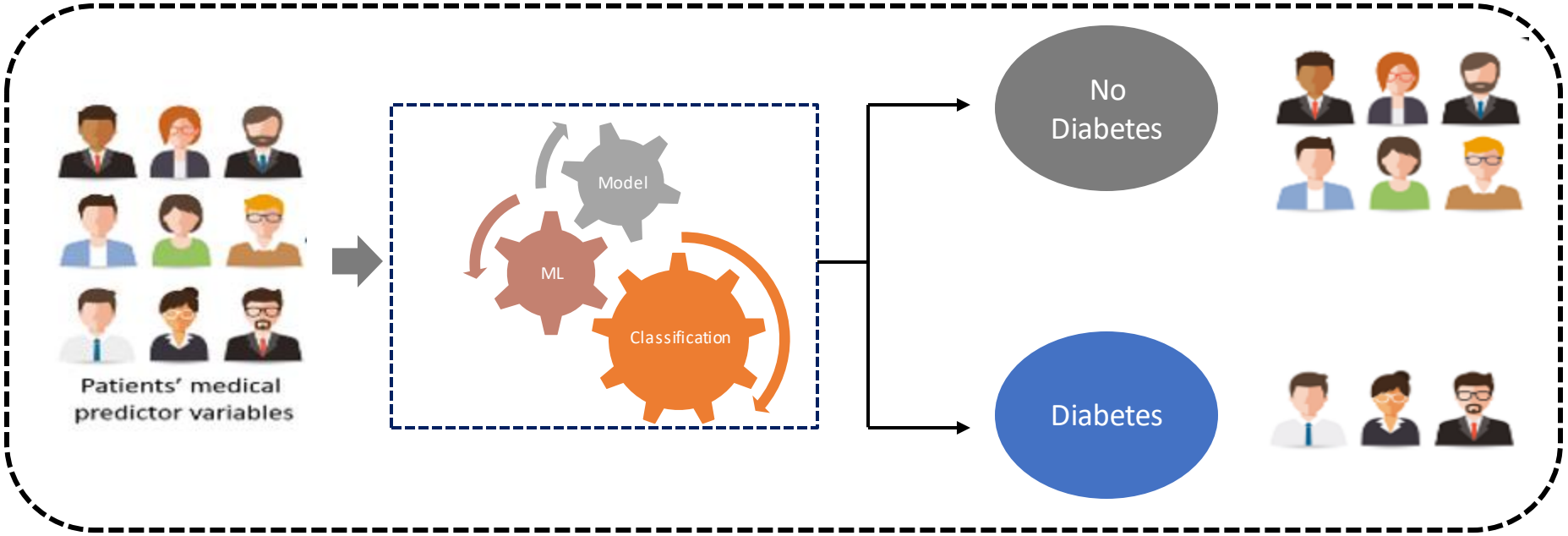
~80% of Resources improvement in last 10 Years

Data Science integrates all the above AI, ML & DL to extracts insight from data (EDA) and make predictions from large datasets (Predictive Analysis)

# What is AI, ML, DL and DS



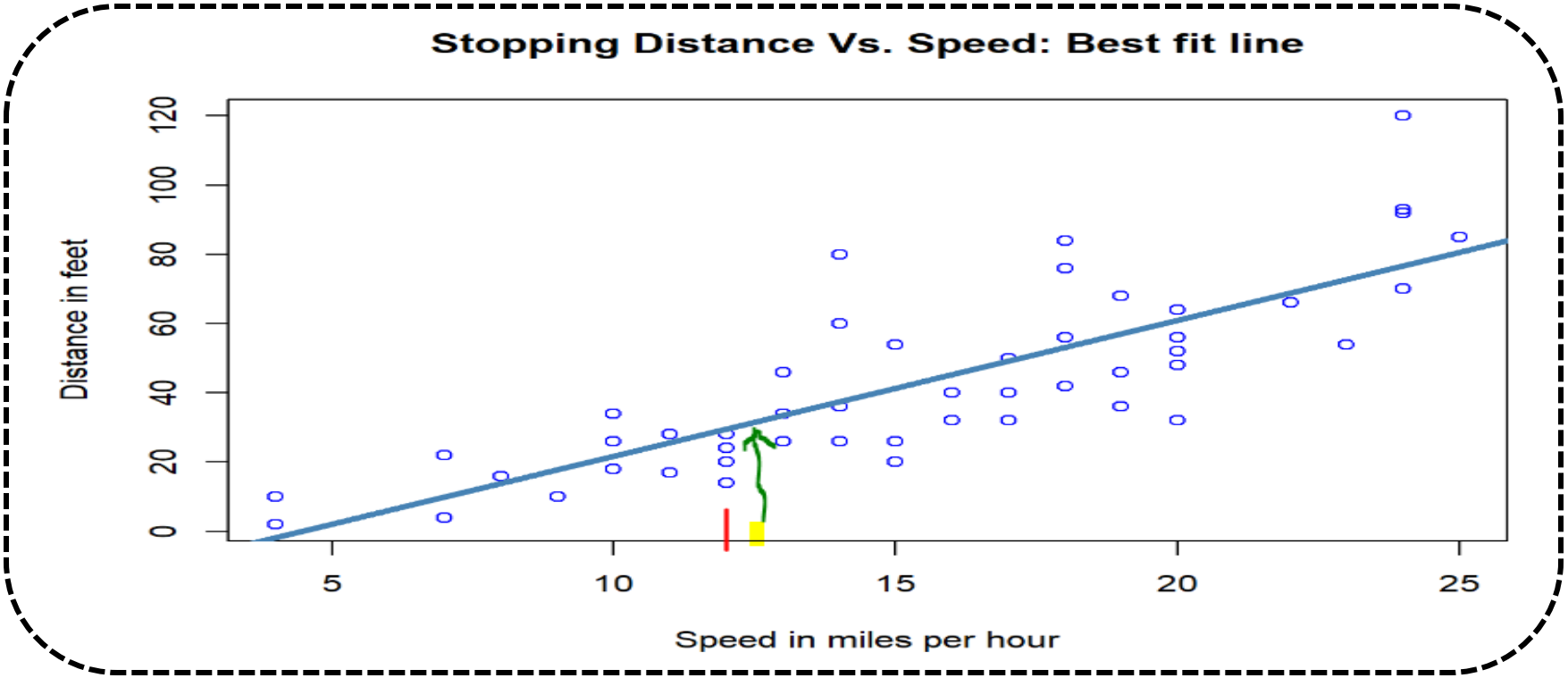
# Supervised Learning – Classification




**Important Algorithms for Classification**

1. Logistic Regression
2. Naïve Bayes' Classifier
3. K-Nearest Neighbors – KNN
4. Decision Tree
5. Ensemble Methods: Bagging & Boosting
6. Random Forest
7. Support Vector Machine
8. ……

# Supervised Learning – Regression



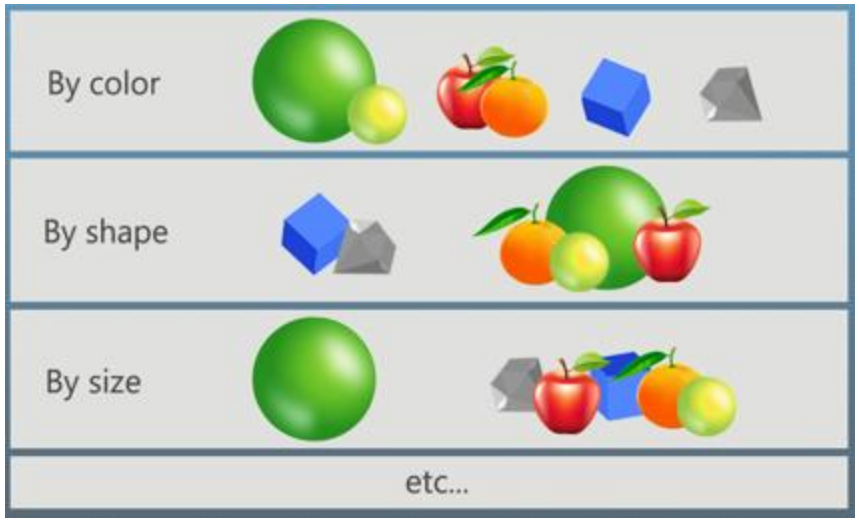

**Important Algorithms for Regression**

1. Linear Regression
2. Multiple Linear Regression
3. Decision Trees
4. Random Forests
5. ……

# Unsupervised Learning – Clustering

Pre Clustering

Post Clustering




**Clustering**

- Find elements (rows, tuples) which are similar.
- Finding “areas” in space where data is concentrated
- WYSIWYG : What You Select Is What You Get

**Important Algorithms:**

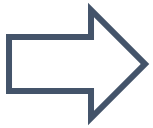
- K-Means
- Fuzzy C-means
- Expectation Maximization
- Hierarchal Clustering
- ……

**Applications**

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Image segmentation
- Anomaly detection

# Unsupervised Learning – Association

| TID | Items |
|---|---|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |



| | Beer | Bread | Milk | Diaper | Eggs | Coke |
|---|---|---|---|---|---|---|
| $T_1$ | 0 | 1 | 1 | 0 | 0 | 0 |
| $T_2$ | 1 | 1 | 0 | 1 | 1 | 0 |
| $T_3$ | 1 | 0 | 1 | 1 | 0 | 1 |
| $T_4$ | 1 | 1 | 1 | 1 | 0 | 0 |
| $T_5$ | 0 | 1 | 1 | 1 | 0 | 1 |

**What does the value of one feature tell us about the value of another feature?**

People who buy diapers are likely to buy baby powder
If (people buy diaper), then (they buy baby powder)

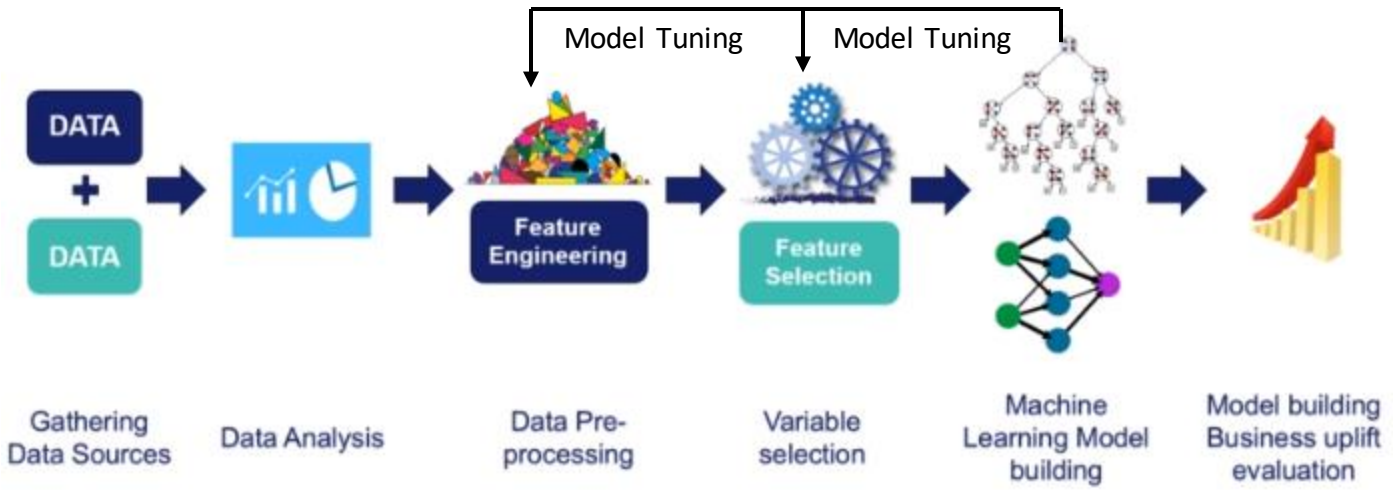
**Association rules**

- Are statements about relations among features (attributes) : across elements (tuples)
- Use a transaction-item set data model

**Association Rules beyond Market Basket Analysis**

- People who visit webpage X are likely visit webpage Y.
- Nodes which run a web server are likely to run Linux.
- People who have age-group [30,40] & income [>$100k] are likely to own home

# Machine Learning Model Flow




**Gathering Data**

- Collecting the data is first and an essential step towards any data science and Machine Learning project.
- It can be from business units or from public datasets for building ML models.

**Data Analysis**

Answer questions such as:

- what variables are available?
- how are they related?
- what is the characteristics of those variables? (numerical or categorical?)
- missing values? outliers?

**Data Pre-processing (Feature Engineering)**

**Our goal** here is to make the data ready for building ML models! To this end, many things can be done but not limited to such as

- Filling missing values in the data
- Dealing with (e.g., removing) outliers
- Transforming categorical values

**Variable Selection (Feature Selection)**

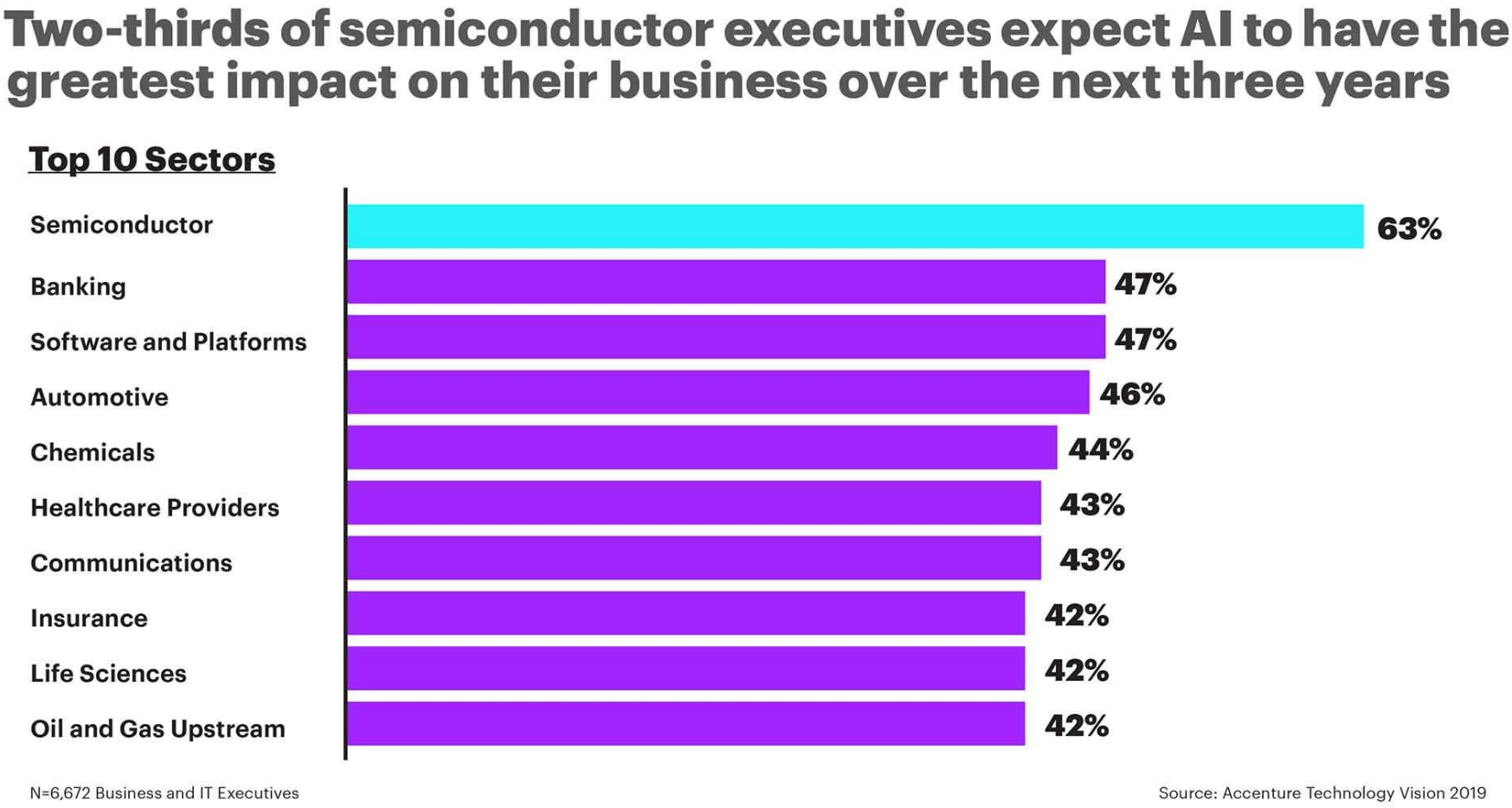
- Select a subset of features out of all features which is critical to ML model performance.
- Many feature could just be noise so removing them is important for overfitting and complexity reduction etc.

**ML Model Building**

Select the ML Model family for training which suited to best of our need:

- Linear Regression
- Logistic Regression
- KNN
- DT
- RF
- Ensembles
- ....

# AI Adaptability in Semiconductors | Market Research




**Semiconductor Industry Leads in Artificial Intelligence Adaption, Accenture Report Finds**

https://newsroom.accenture.com/industries/electronics-high-tech/semiconductor-industry-leads-in-artificial-intelligence-adoption-accenture-report-finds.htm

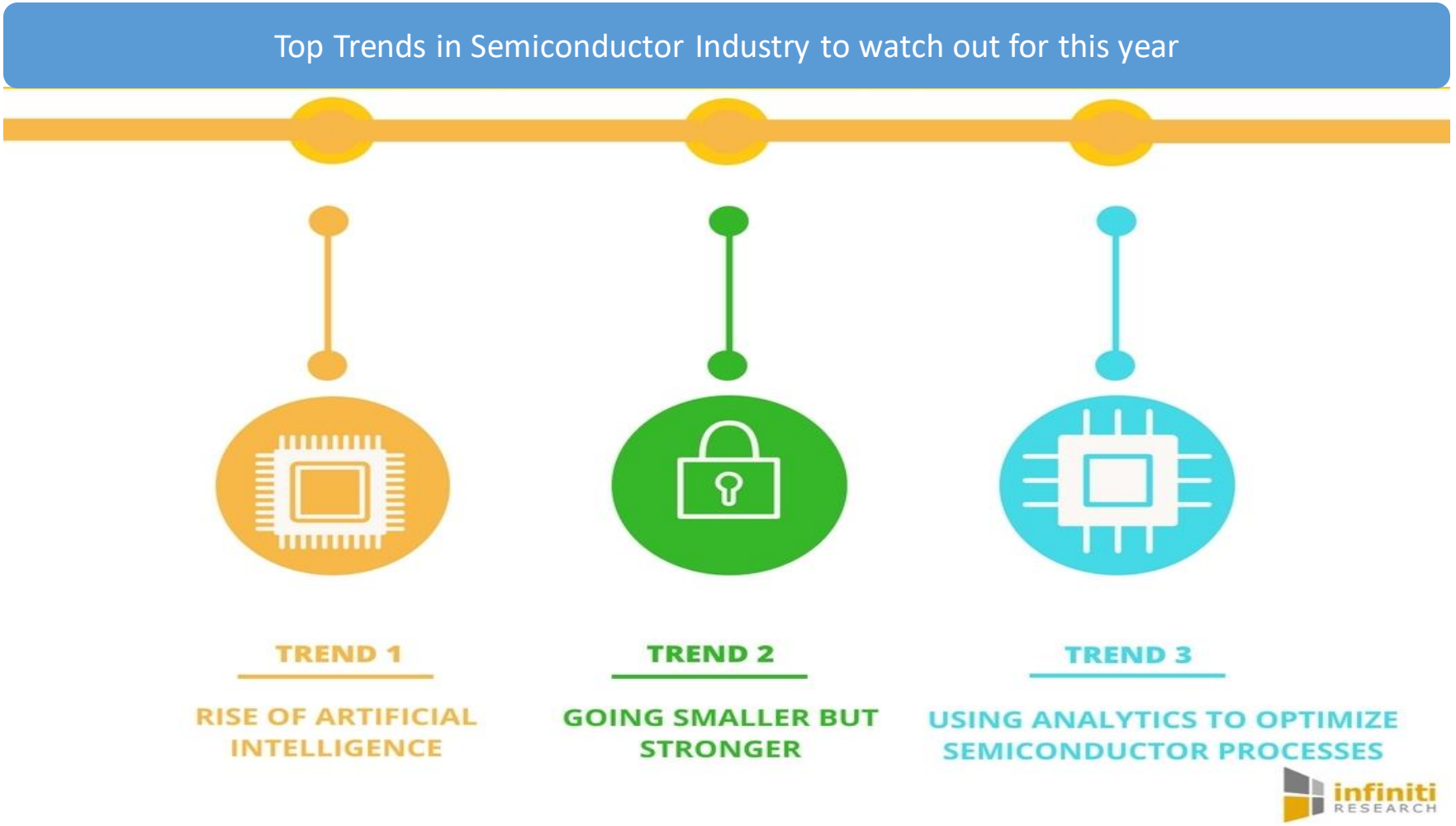
# Market Opinion

**“AI will be a major growth driver for the semiconductor industry in light of high manufacturing costs and the growing complexity of chip development,”**

**“To capture this opportunity, chipmakers should leverage AI technologies and partnerships to increase efficiency across their operations.”**

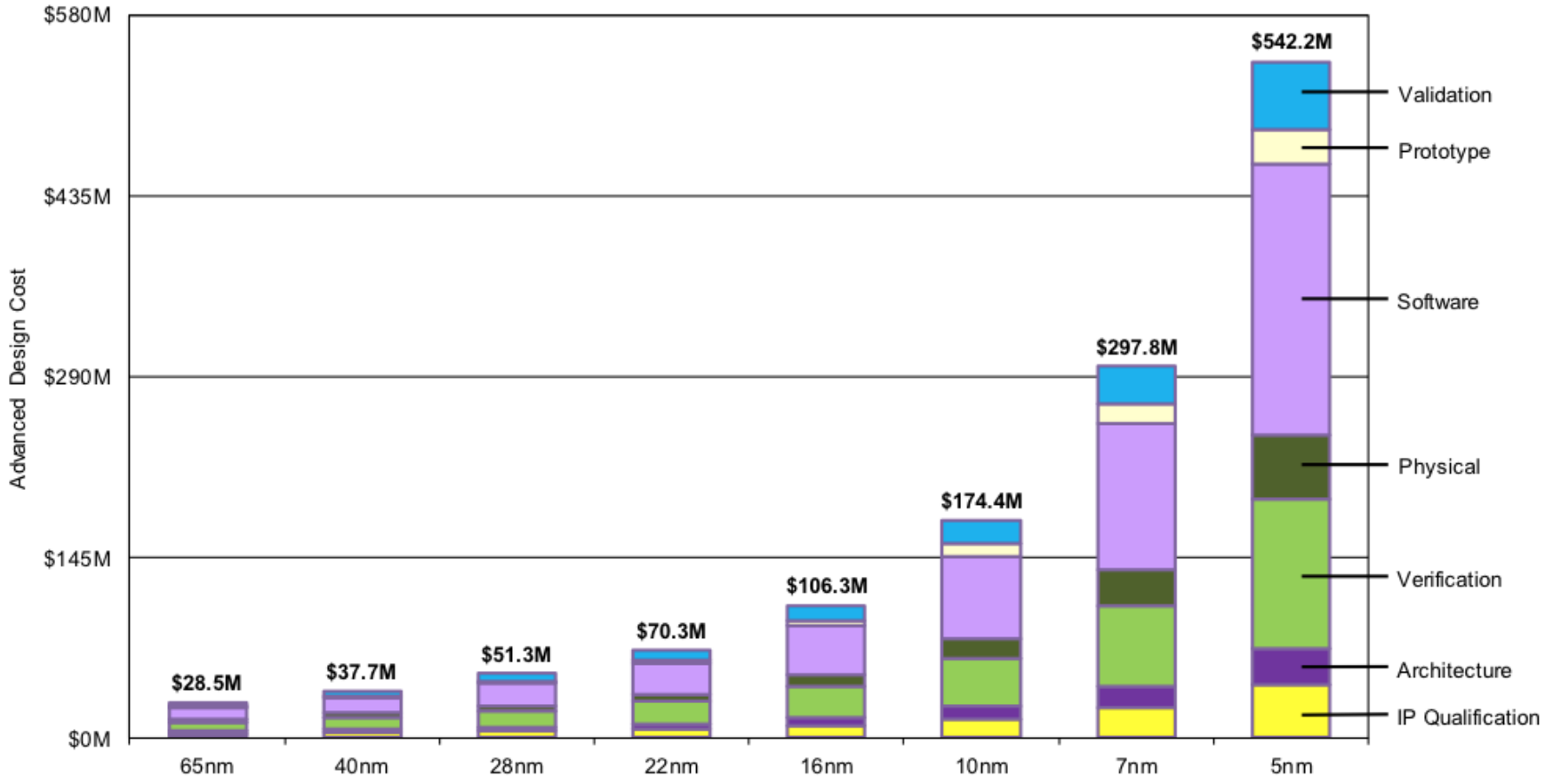
---Syed Alam, Managing Director at Accenture, leads its Semiconductor practice globally.

# Top Trends in Semiconductor Industry




https://www.businesswire.com/news/home/20180524005772/en/Key-Semiconductor-Industry-Trends-Infiniti-Research

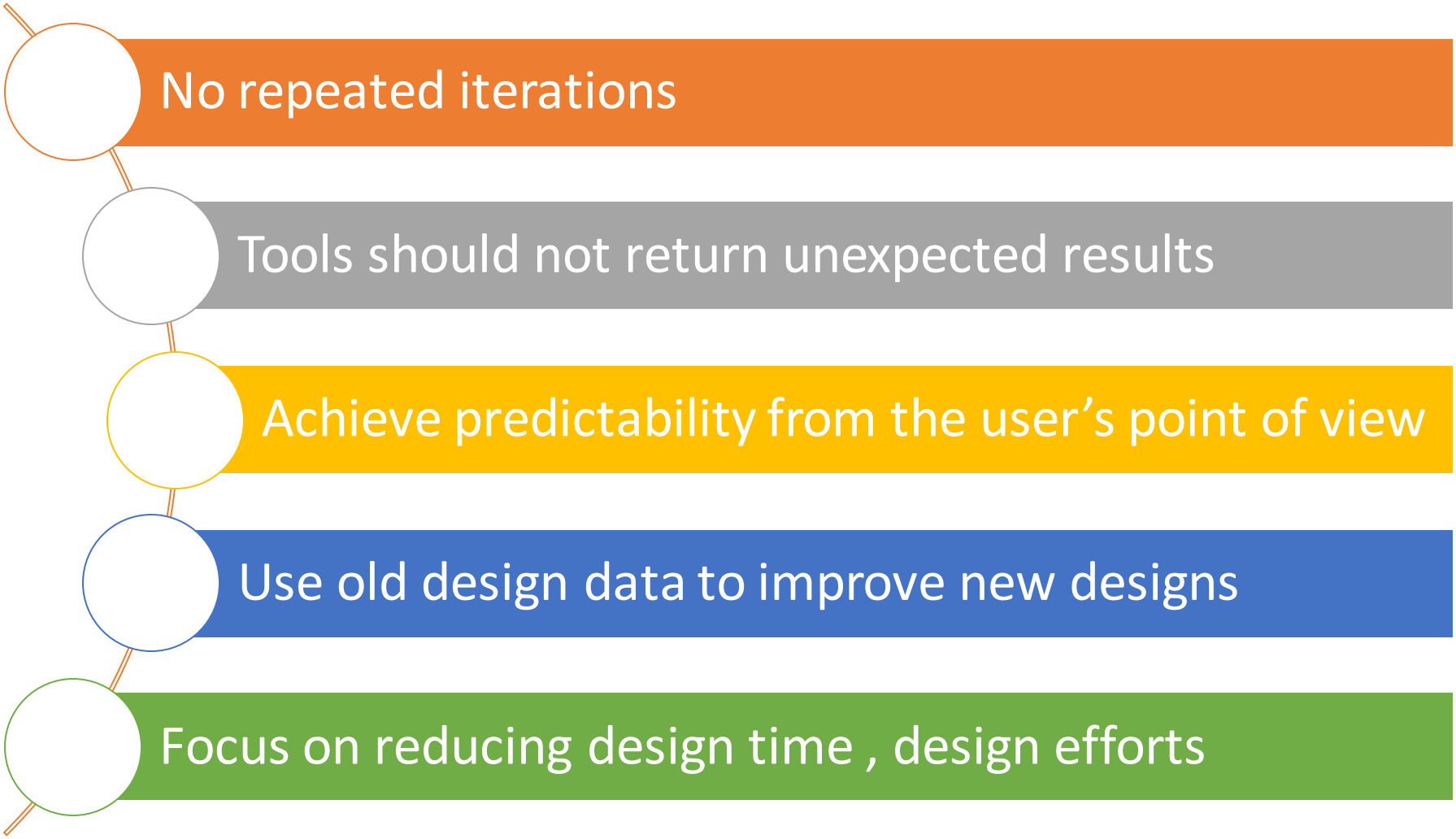
# Why AI make sense in Semiconductors




As you can see there is one clear conclusion - **the design and manufacturing of advanced node (10 nm and below) are HUGE and increasing exponentially.**

https://www.quora.com/How-much-does-it-cost-to-design-and-fabricate-an-integrated-circuit

# Semiconductor Ideal Expectations from AI



# What is the solution?

Machine Learning will be a key piece of this . . .

# AI Technology Verticals used in Semiconductors

## Natural Language Processing

Text Processing Technique to process raw and unstructured text to generate information.

Ex.

1. Q & A Bot
2. Sentiment Analysis
3. Feedback Rating System
4. Document Retrieval

Important Libraries: NLTK, spaCy.

## Computer Vision

Computer Vision or CV is a digital Image Processing Technique where we apply different Neural Network Architecture to process image and classify them.

Ex.

1. Wafer Defect Classification
2. Congestion Prediction
3. Image to Image Generation

Etc...

Neural Net: CNN

## Machine Learning

ML has wide range of application which fall in predictive modelling as well as time series forecasting.

Ex.

1. Anomalies Detection
2. Predicting upcoming market trends
3. Stock price forecasting

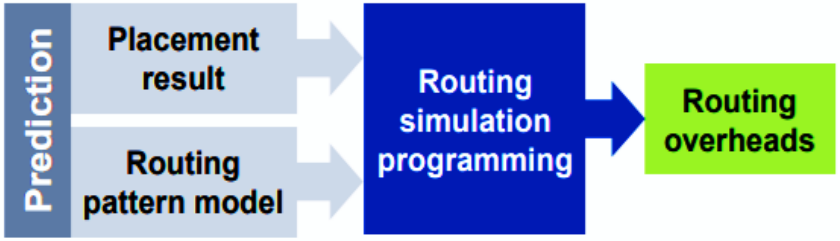
Etc...

Types: Classification & Regression

# Use-case 1 (Congestion Prediction)

***Eliminate human and fixed model subjective bias with statistical important features***

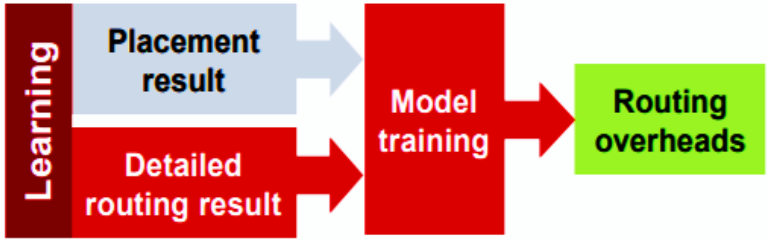
## Traditional EDA Approach




Easy to be biased and no guaranteed quality

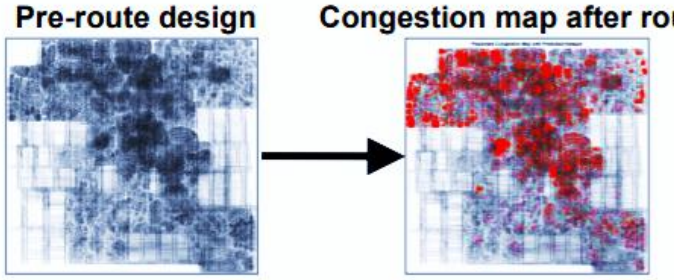
Only heuristics are possible

## Machine Learning Approach



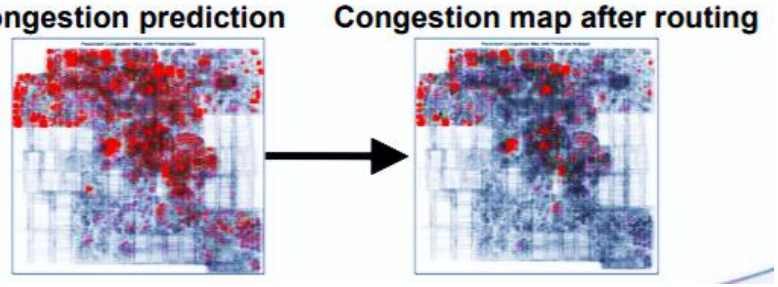

Leverage widely used deep learning packages

## Traditional (EDA) flow




Only know routing results after running routing

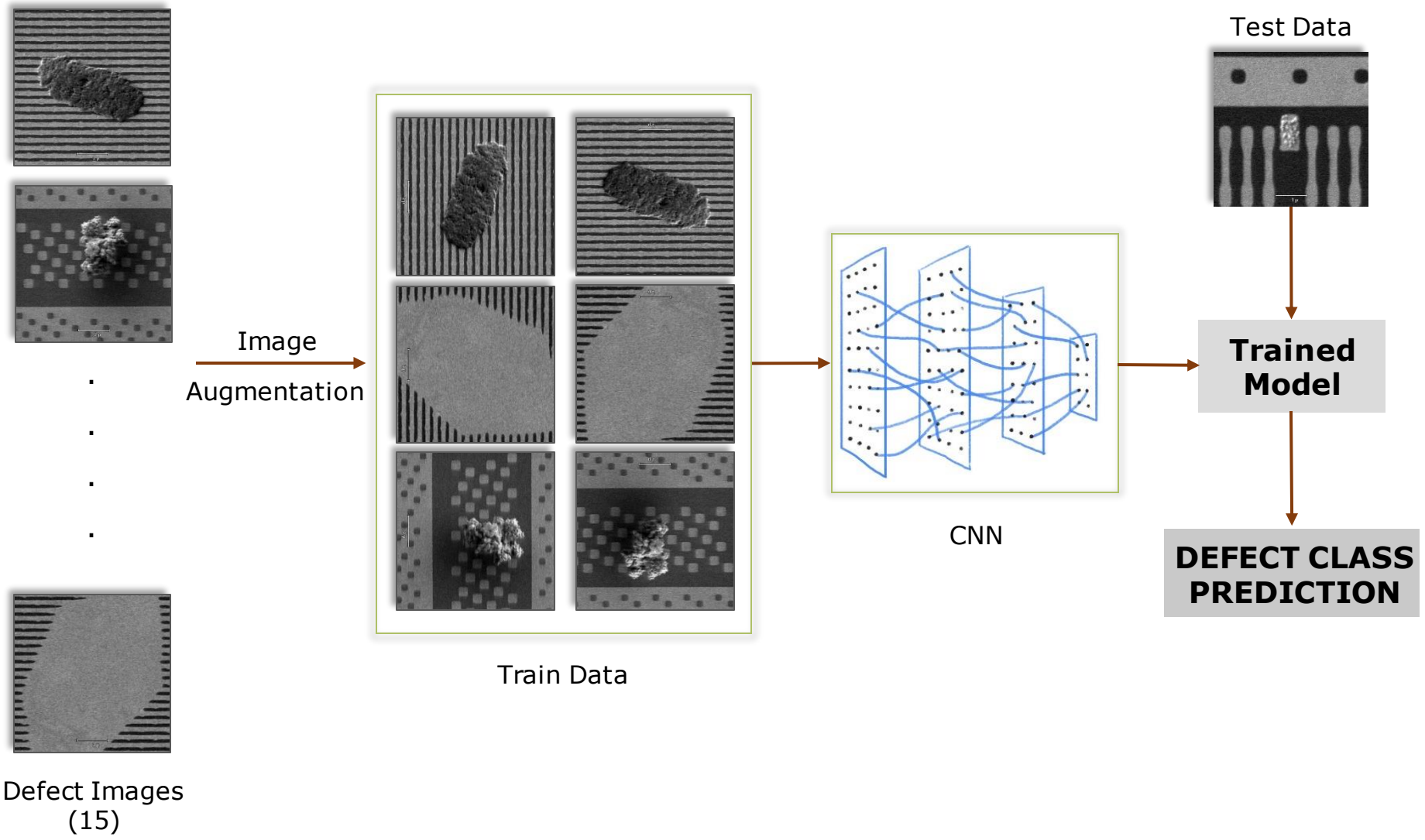
## Flow enables by Machine Learning



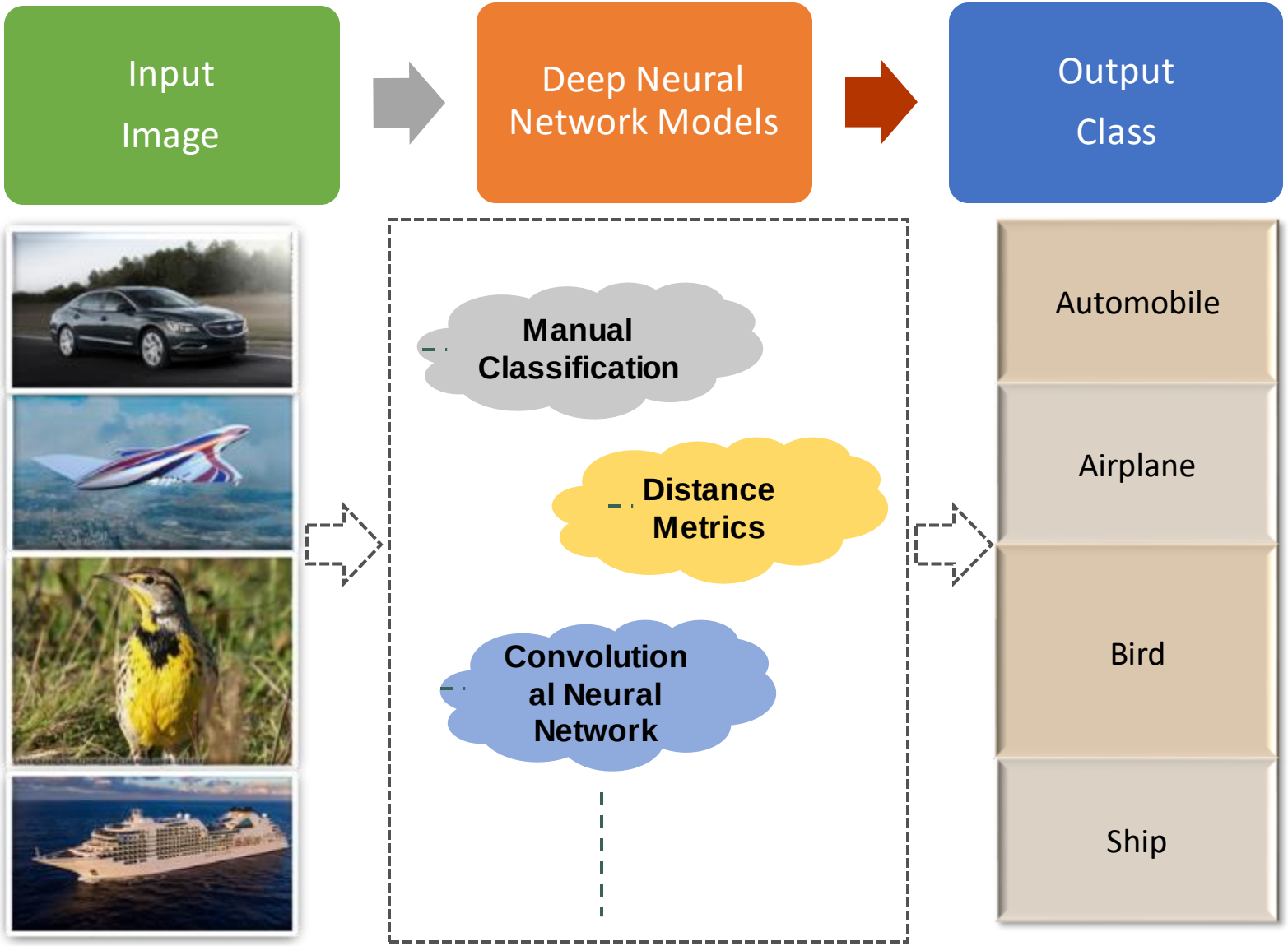

Able to predict routing behavior and improve congestion with new recipes to achieve 40Mhz

http://www.ispd.cc/slides/2018/s4_1.pdf

# Use-case 2 (Wafer Defect Classification)

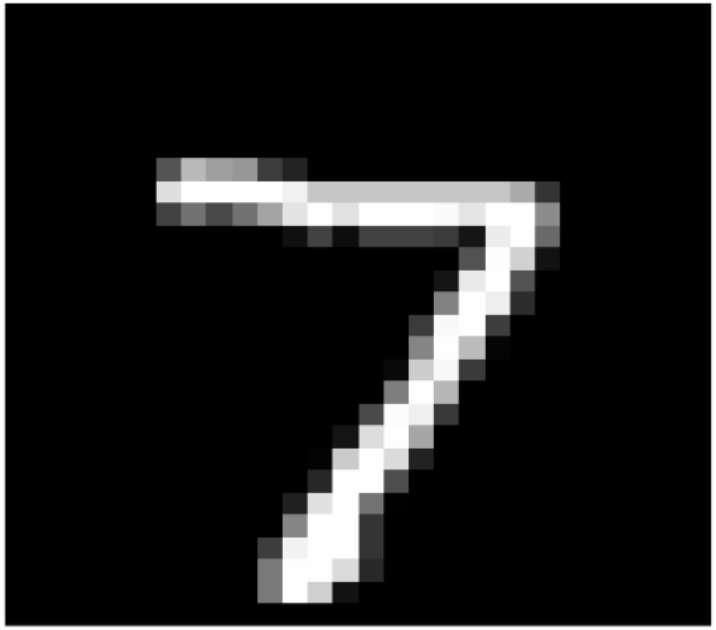


# Introduction to Image Classification



# How Images are Represented

28 x 28 image



28 columns

28 rows

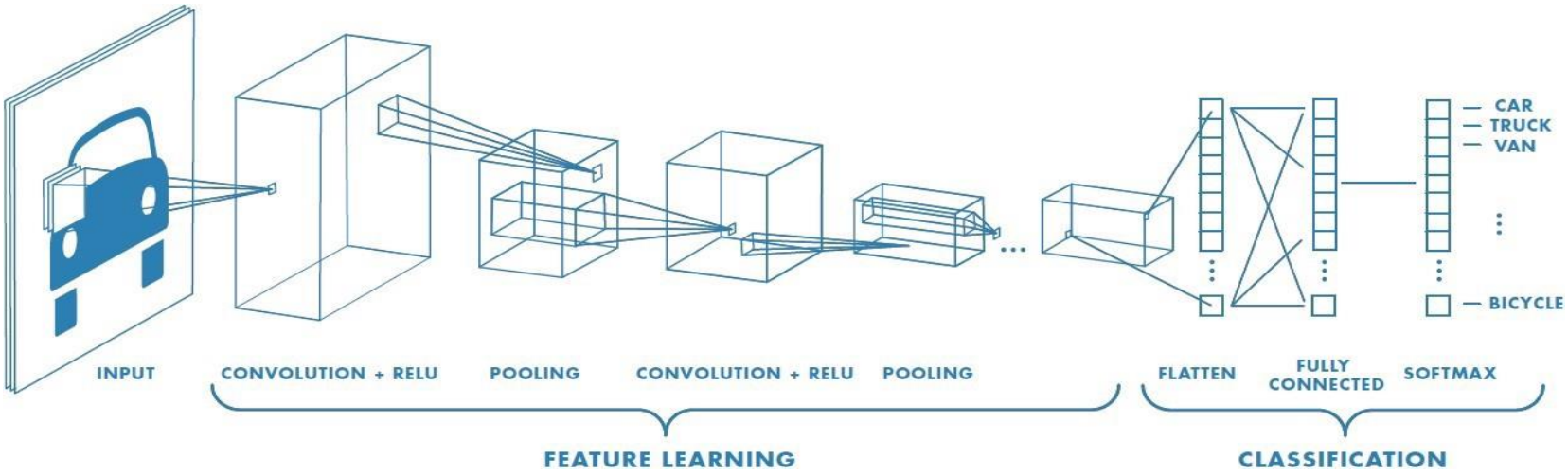
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 | V21 | V22 | V23 | V24 | V25 | V26 | V27 | V28 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 84 | 185 | 159 | 151 | 60 | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 222 | 254 | 254 | 254 | 254 | 241 | 198 | 198 | 198 | 198 | 198 | 198 | 198 | 198 | 170 | 52 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 67 | 114 | 72 | 114 | 163 | 227 | 254 | 225 | 254 | 254 | 254 | 250 | 229 | 254 | 254 | 140 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 66 | 14 | 67 | 67 | 67 | 59 | 21 | 236 | 254 | 106 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 83 | 253 | 209 | 18 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 233 | 255 | 83 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 129 | 254 | 238 | 44 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 59 | 249 | 254 | 62 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 133 | 254 | 187 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 205 | 248 | 58 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 126 | 254 | 182 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 75 | 251 | 240 | 57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 221 | 254 | 166 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 203 | 254 | 219 | 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 38 | 254 | 254 | 77 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31 | 224 | 254 | 115 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 133 | 254 | 254 | 52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 61 | 242 | 254 | 254 | 52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 121 | 254 | 254 | 219 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 121 | 254 | 207 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Pixel Value (brightness) range : 0 - 255

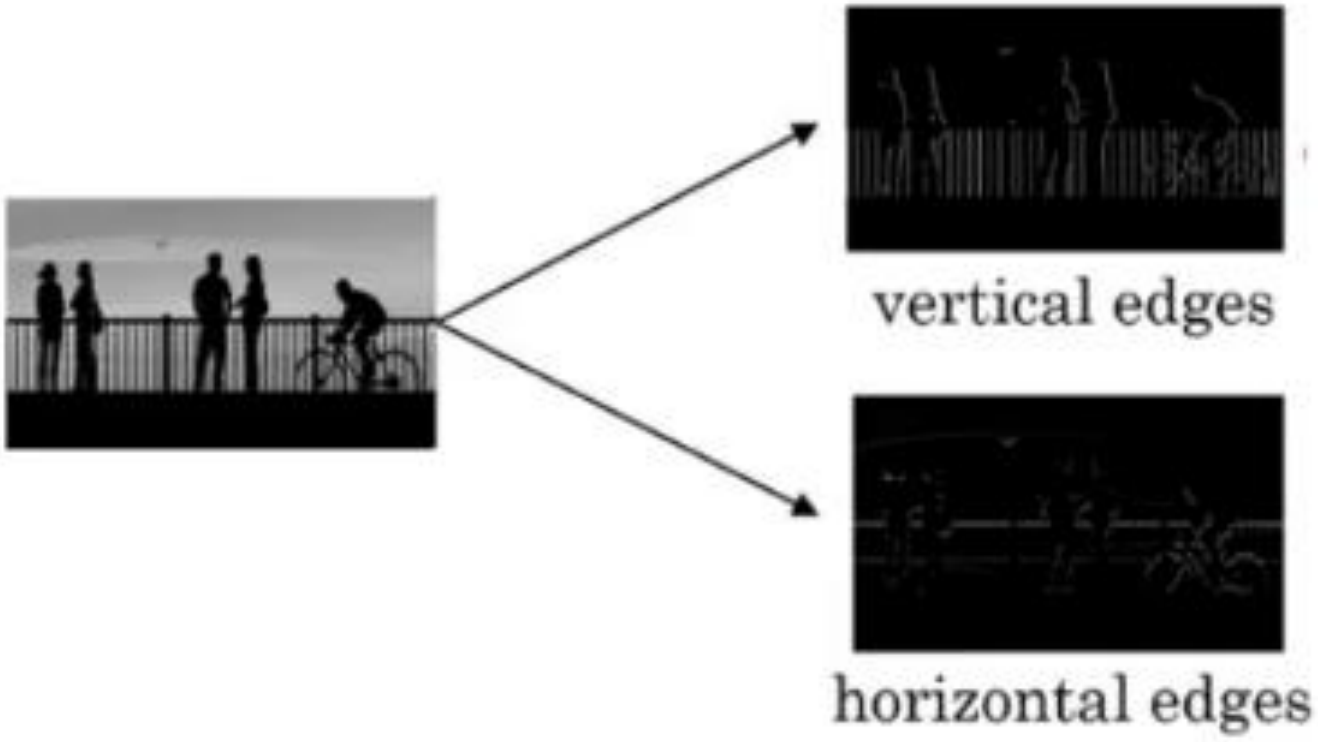
# Major Challenges in Image Processing

- Size of the input data.
- Suppose an image is of the size 68 X 68 X 3. The input feature dimension then becomes 12,288.
- This will be even bigger if we have larger images (say, of size 720 X 720 X 3).
- Now, if we pass such a big input to a neural network, the number of parameters will swell up to a HUGE number (depending on the number of hidden layers and hidden units).
- This will result in more computational and memory requirements – not something to be handled easily.
- Convolutional Neural Network solves all these challenges in a step by step Approach.

# Architecture of Convolutional Neural Network – CNN



# 1. Convolution Layer - Edge Detection




https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/

# 1. Convolution Layer - Edge Detection

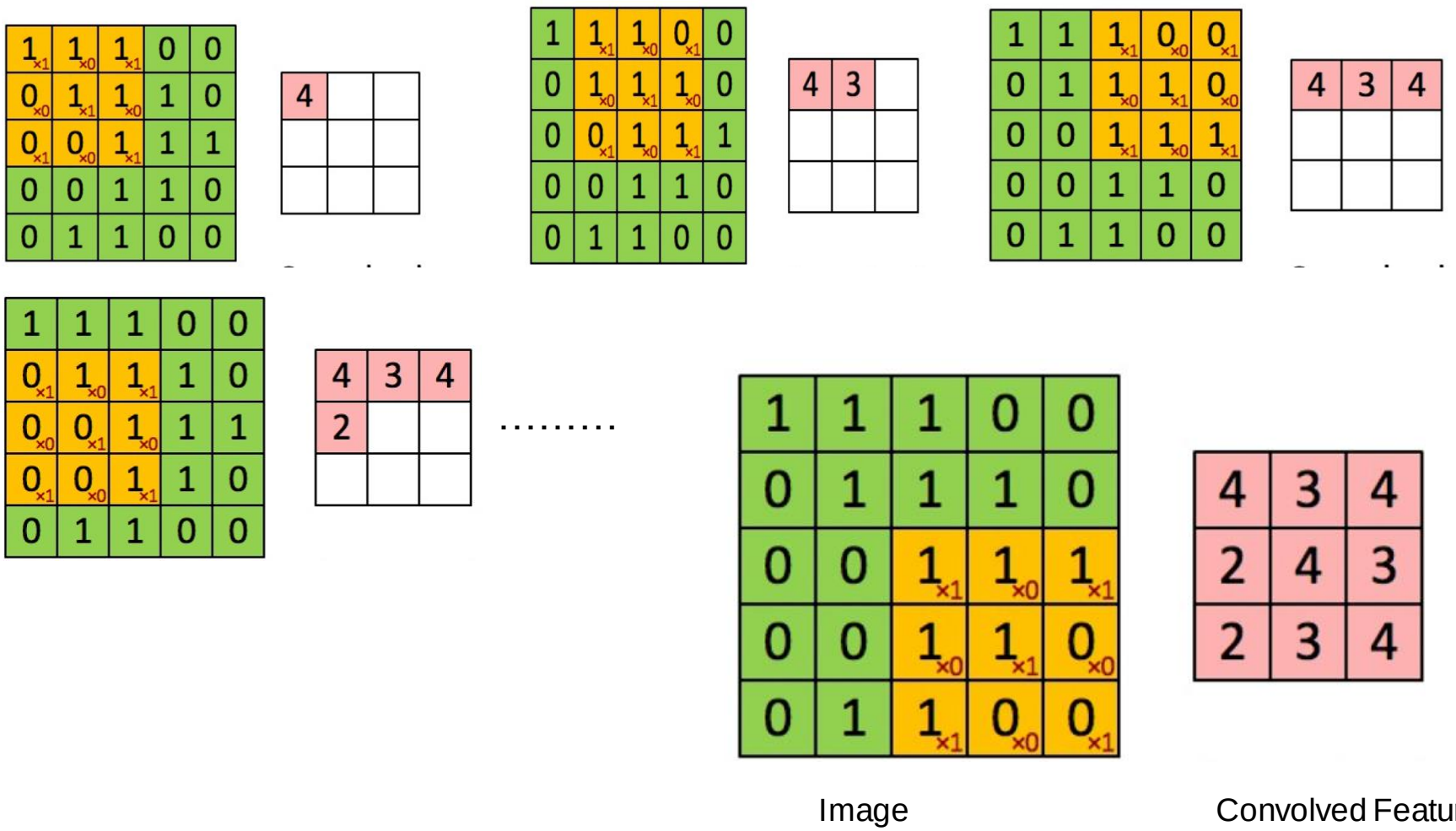
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Image

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Filter

# 1. Convolution Layer - Edge Detection



# What Weight Matrix or Filter does?

- The weight matrix behaves like a filter in an image extracting particular information from the original image matrix.

- One weight combination might be extracting edges, while another one might a particular color, while another one might just blur the unwanted noise.

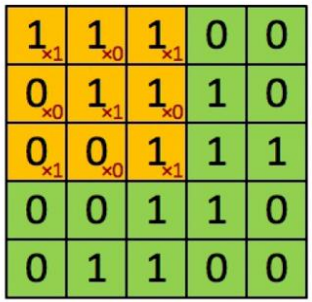
- When we have multiple convolutional layers, the initial layer extract more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more complex and more suited to the problem at hand.

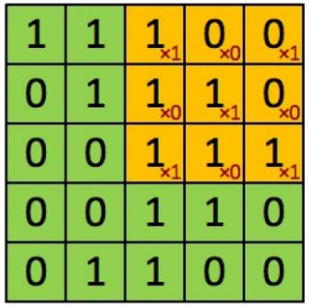
# Importance of Padding in Convolution Layer

- Every time we apply a convolutional operation, the size of the image shrinks
- Pixels present in the corner of the image are used only a few number of times during convolution as compared to the central pixels.
- Hence, we do not focus too much on the corners since that can lead to information loss
- To overcome these issues, we can pad the image with an additional border, i.e., we add one pixel all around the edges.
- This is where padding comes to the fore

# Strided Convolutions

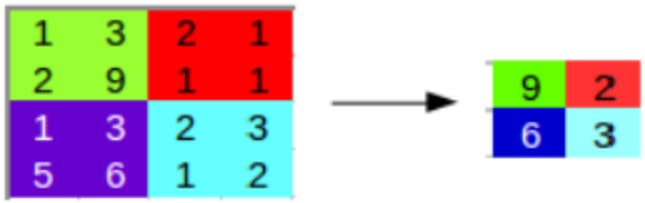
- Stride is nothing but the steps that filter takes while convoluting over the image matrix.
- For example if stride is 2 then filter will jump two steps while convoluting.
- Stride helps to reduce the size of the image, a particularly useful feature.



# 2. Pooling Layers

- Pooling is another method to reduce the size of the image and hence speed up the computation.
- Consider a 4 X 4 matrix as shown below:
- Applying max pooling on this matrix will result in a 2 X 2 output:



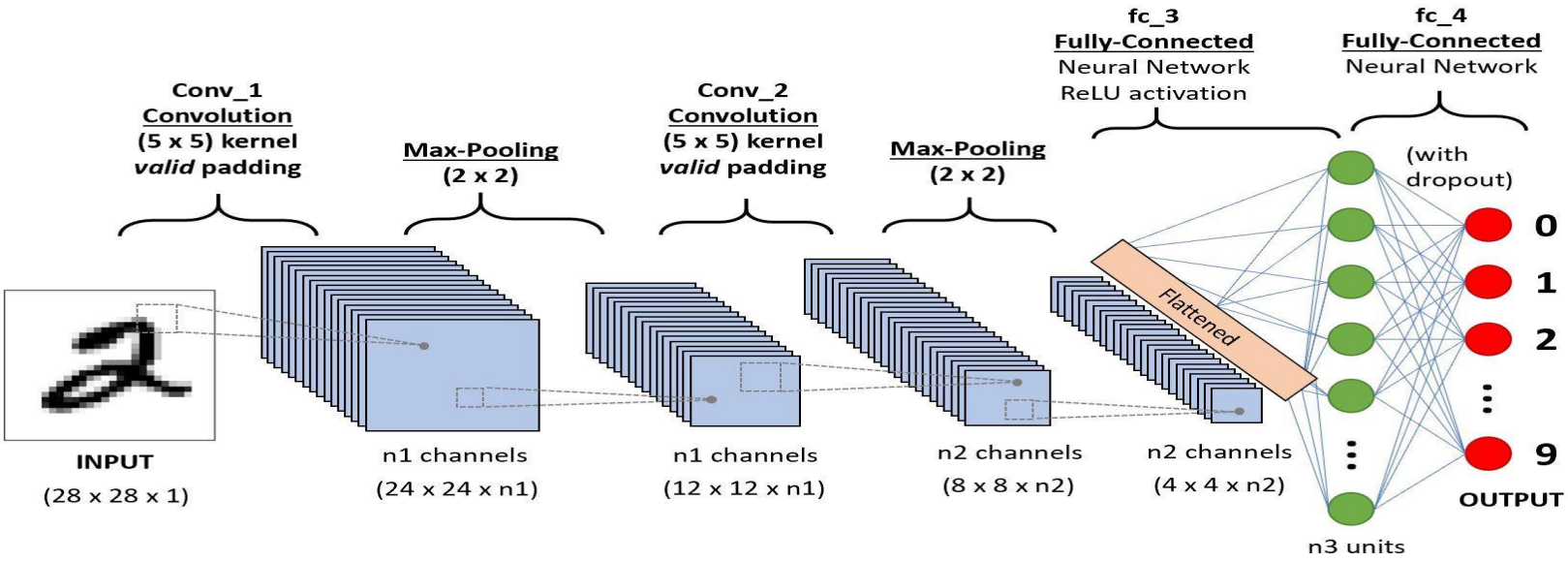

- For every consecutive 2 X 2 block, we take the max number.
- Here, we have applied a filter of size 2 and a stride of 2.
- Another method could be the average pooling instead of Max pooling.
- These all are the hyper parameters which we set to fine tune the model.

# Formula to calculate the feature size

**Input:** n X n X $n_c$
**Filter:** f X f X $n_c$
**Padding:** p
**Stride:** s
**Output:** [(n+2p-f)/s+1] X [(n+2p-f)/s+1] X $n_c$'

Here, $n_c$ is the number of channels in the input and filter, while $n_c$' is the number of filters.

# 3. The Output layer



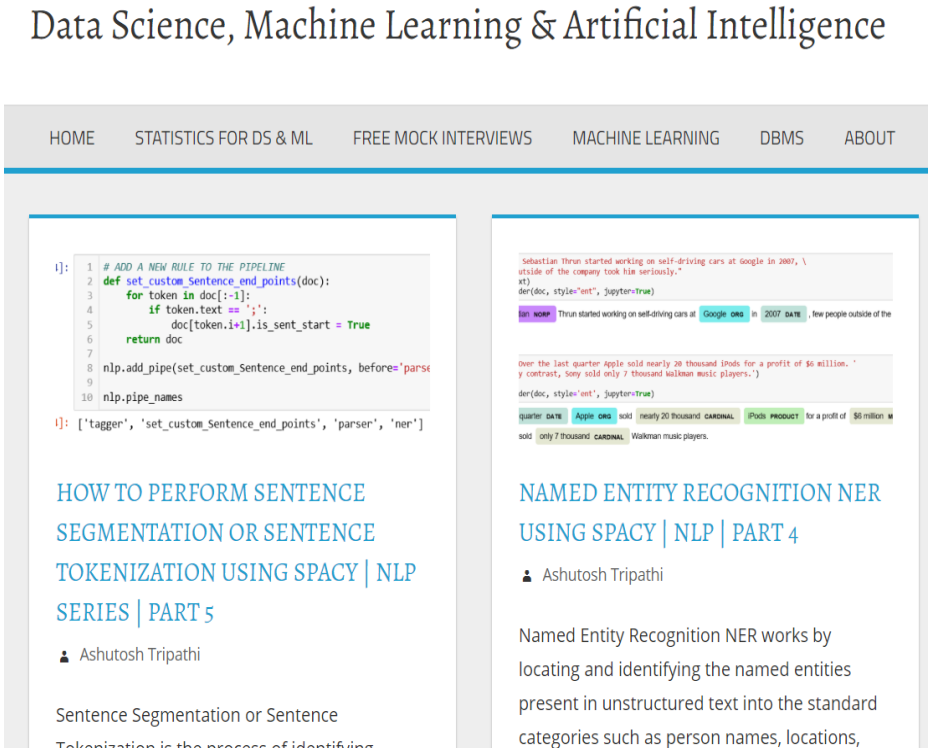
# Future of Semiconductors with AI

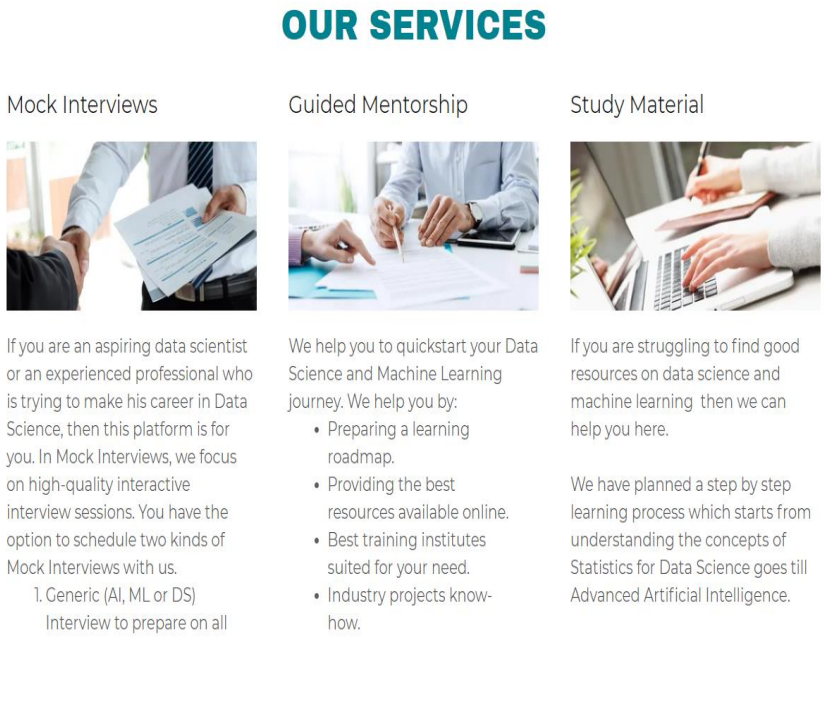
https://venturebeat.com/2020/04/23/google-claims-its-ai-can-design-computer-chips-in-under-6-hours/

# Contact Details

Website: https://ashutoshtripathi.com

Website: https://enetwork.ai




LinkedIn: https://www.linkedin.com/in/ashutoshtripathi1/

Instagram: https://www.instagram.com/ashutosh_ai/

# Thank You