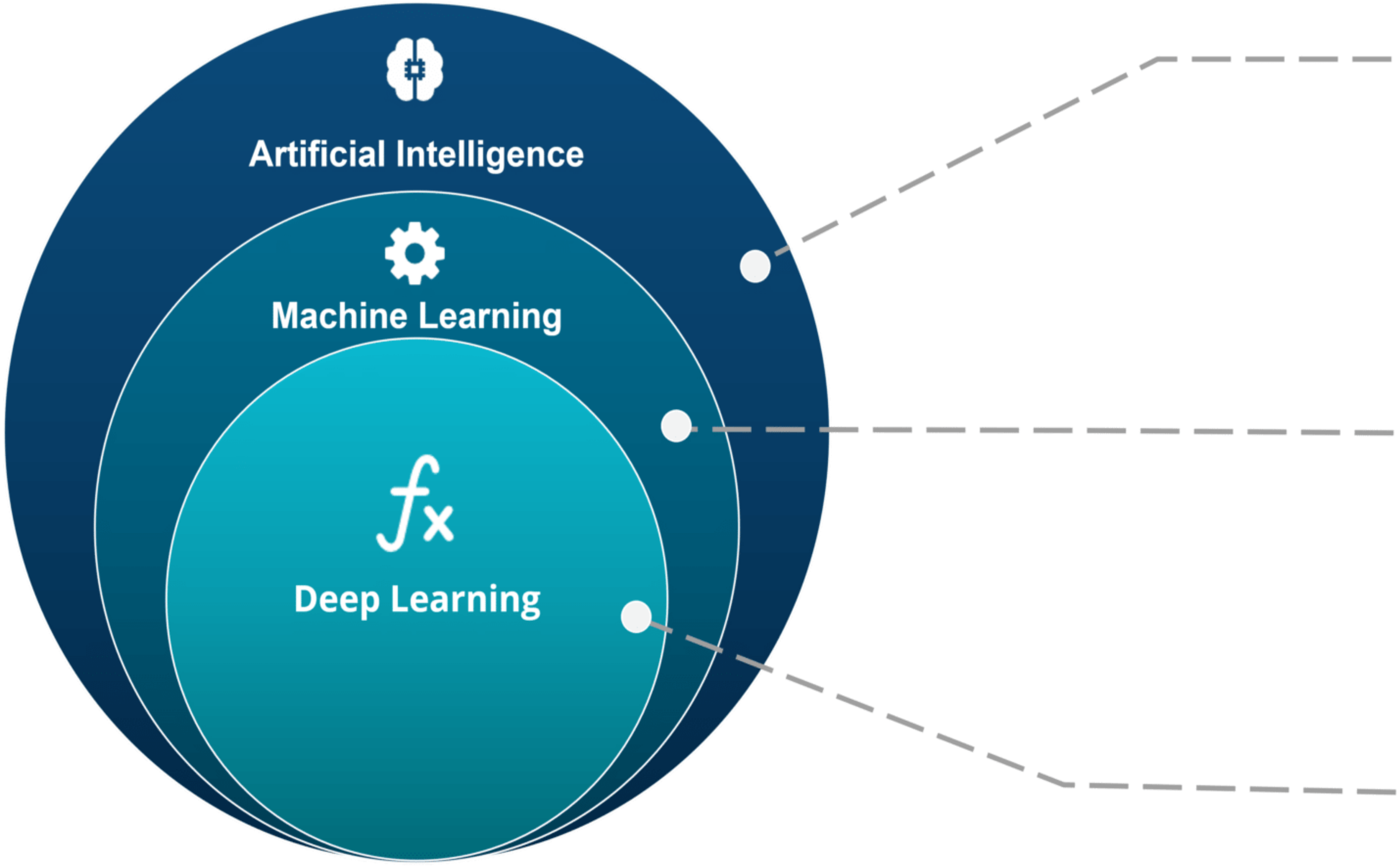
# Model architecture & Selecting the best model

DeepVisionTech.AI

Realize your AI vision with us

# Lets discuss about Deep Learnin

# What is deep learning ?



**ARTIFICIAL INTELLIGENCE**

A technique which enables machines to mimic human behaviour

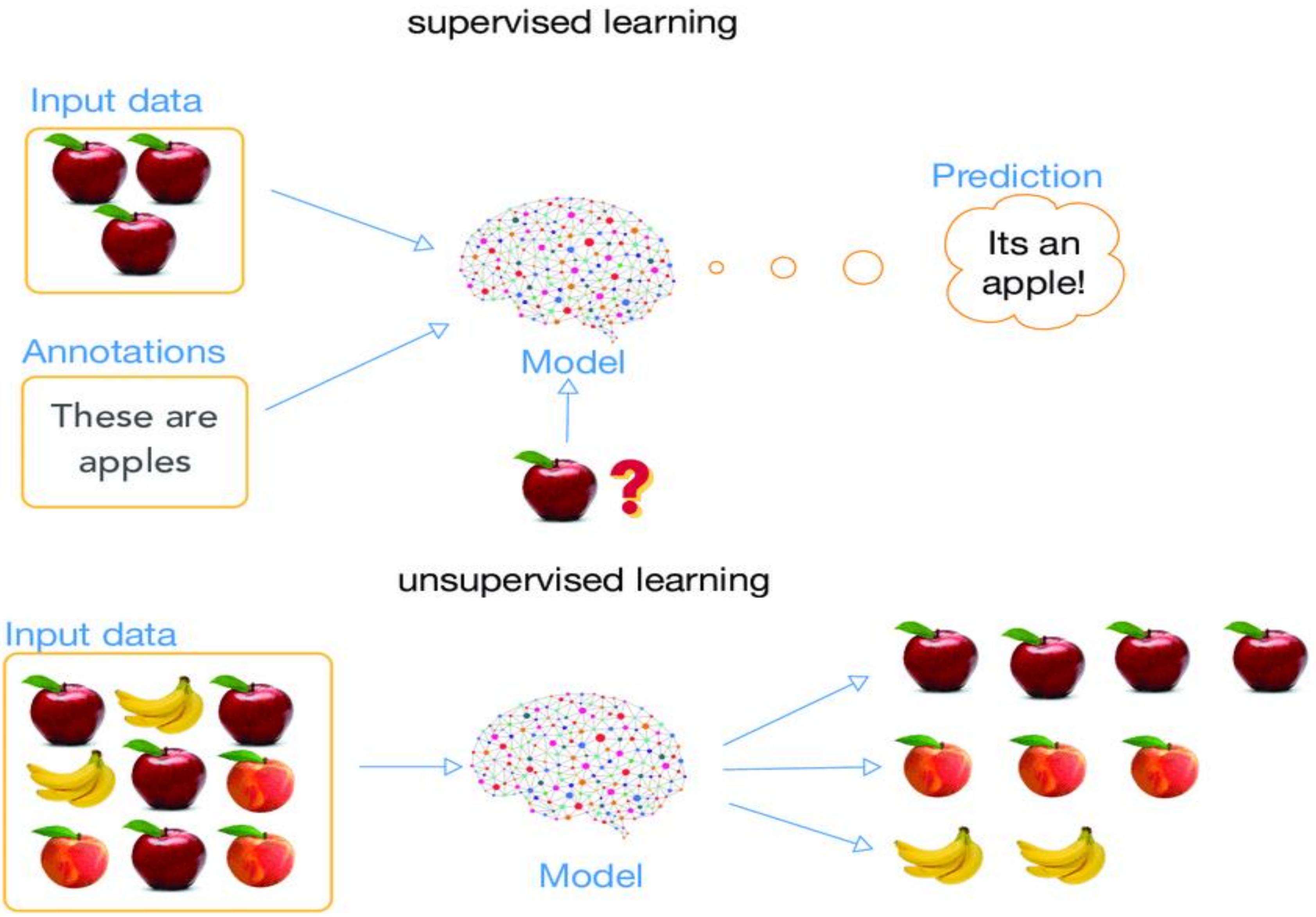
**MACHINE LEARNING**

Subset of AI technique which use statistical methods to enable machines to improve with experience

**DEEP LEARNING**

Subset of ML which make the computation of multi-layer neural network feasible

# Supervised & Unsupervised Learning



supervised learning

Input data

Annotations

These are apples

Model

Prediction

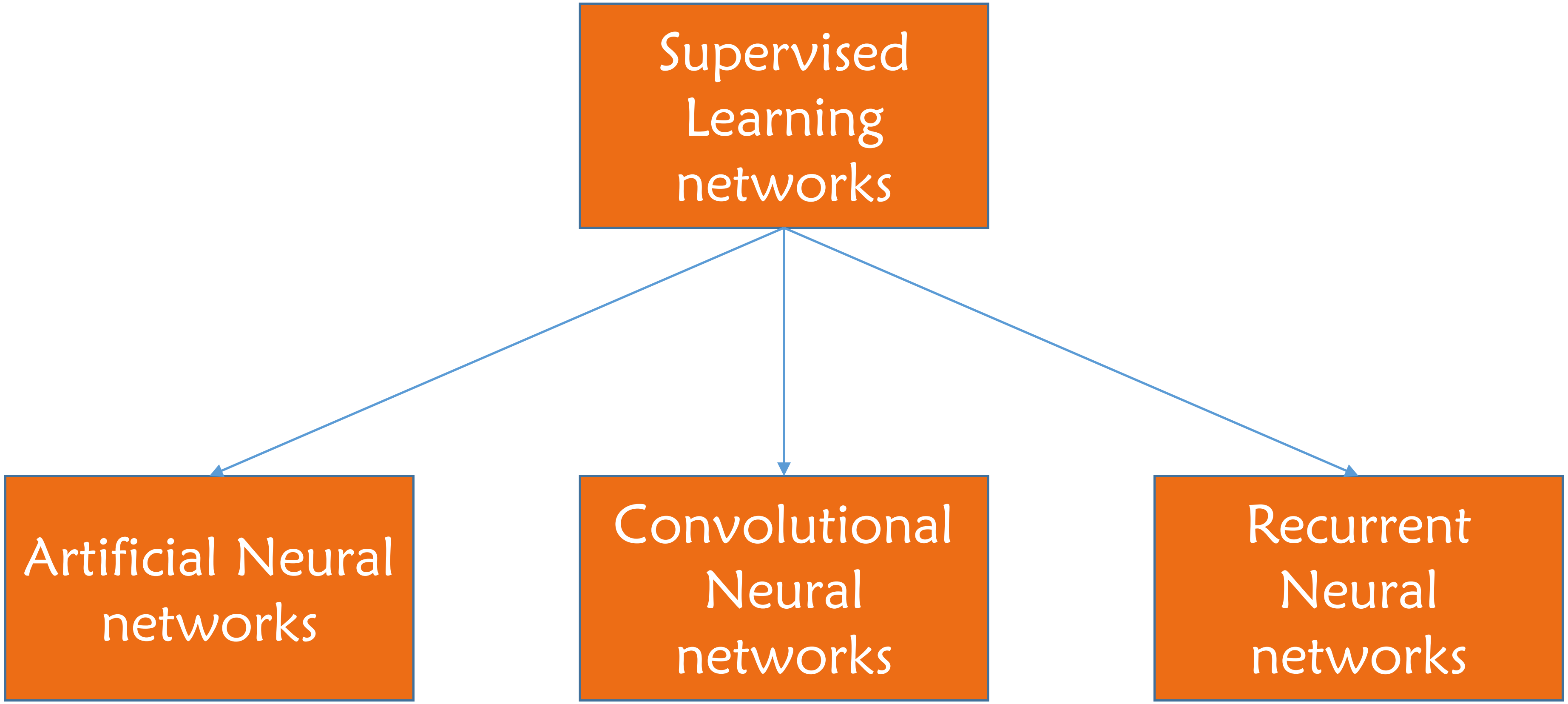
Its an apple!

unsupervised learning

Input data

Model

# Different types of networks



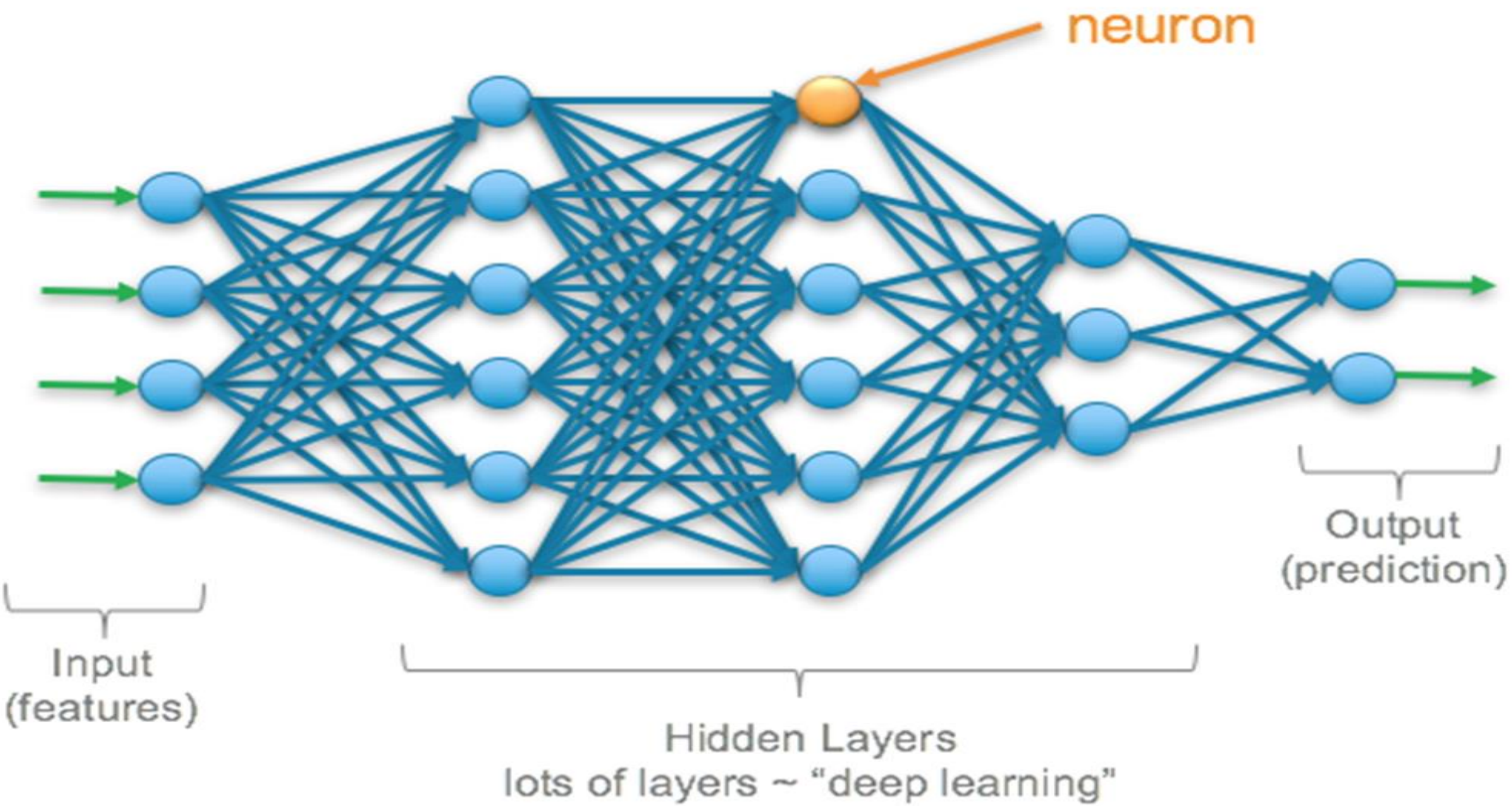
Supervised Learning networks

Artificial Neural networks

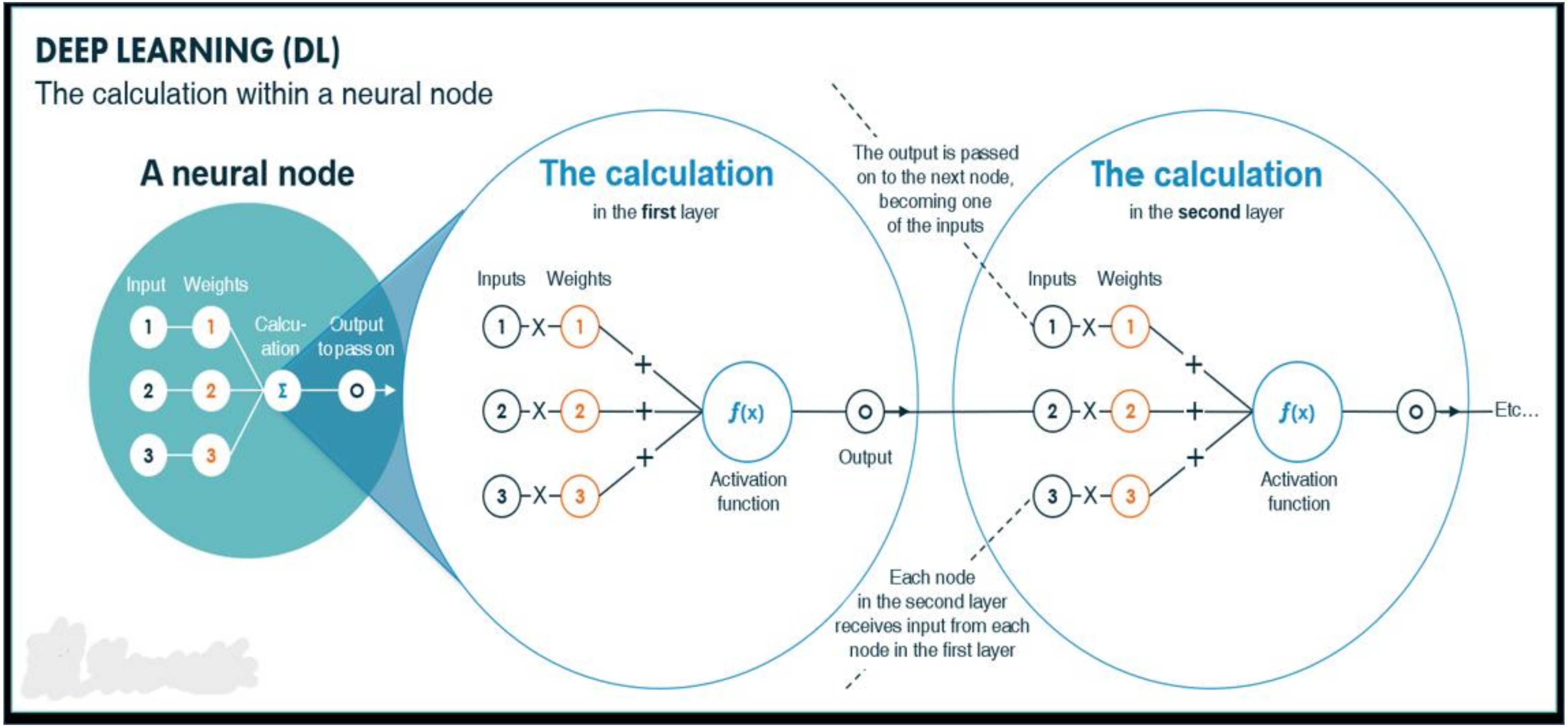
Convolutional Neural networks

Recurrent Neural networks

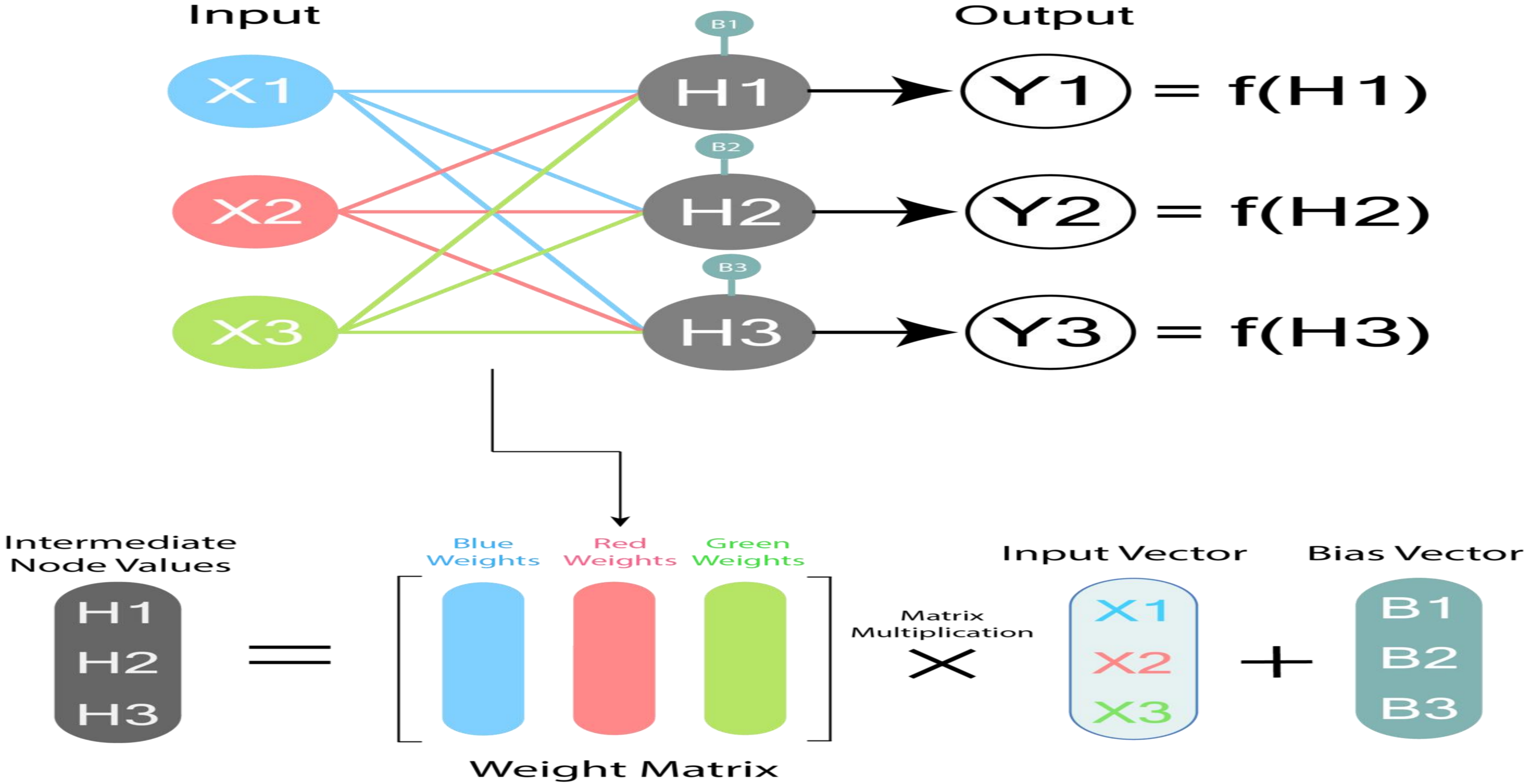
# Model & Architecture



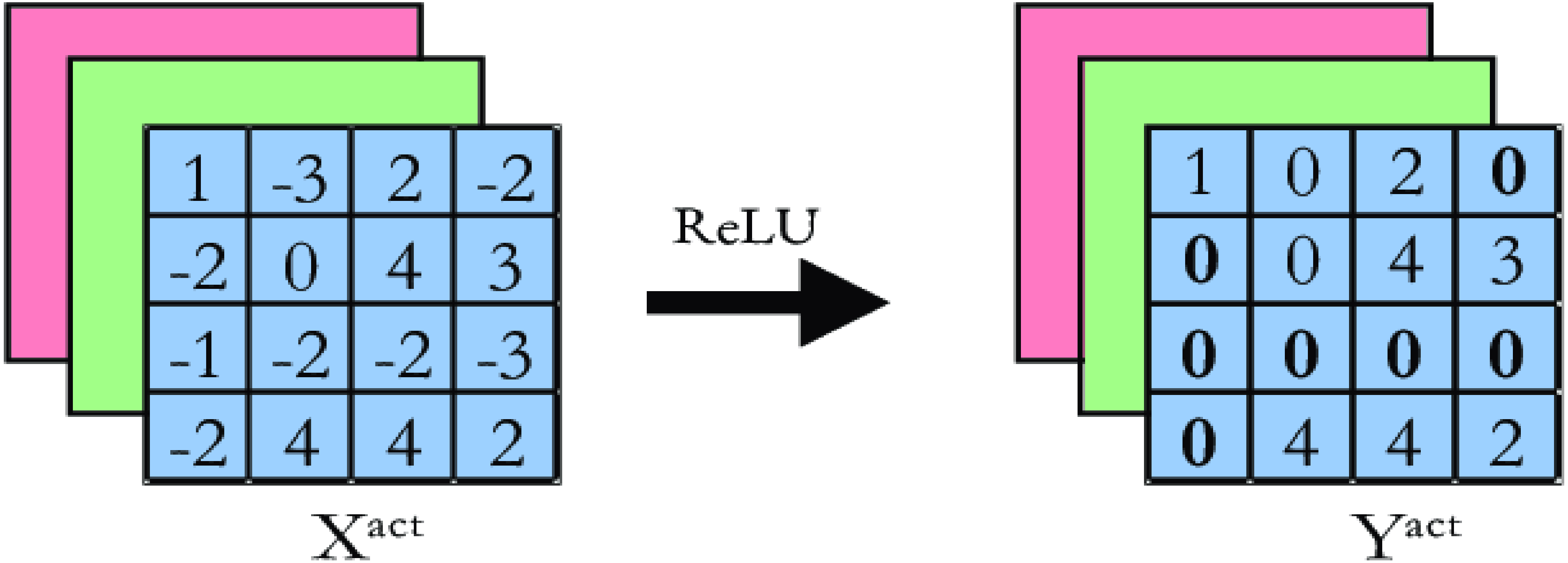
# Dense layer


DEEP LEARNING (DL)
The calculation within a neural node
A neural node
Input
Weights
1
1
2
2
3
3
Calcu-ation
Σ
Output to pass on
The calculation
in the first layer
Inputs
Weights
1
X
1
2
X
2
3
X
3
+
+
+
ƒ(x)
Activation function
Output
The output is passed on to the next node, becoming one of the inputs
Each node in the second layer receives input from each node in the first layer
The calculation
in the second layer
Inputs
Weights
1
X
1
2
X
2
3
X
3
+
+
+
ƒ(x)
Activation function
Etc...

# Dense layer


Input
Output
X1
X2
X3
B1
B2
B3
H1
H2
H3
Y1 = f(H1)
Y2 = f(H2)
Y3 = f(H3)
Intermediate Node Values
H1
H2
H3
=
Blue Weights
Red Weights
Green Weights
Weight Matrix
Matrix Multiplication
×
Input Vector
X1
X2
X3
+
Bias Vector
B1
B2
B3

# Activation function (ReLU)



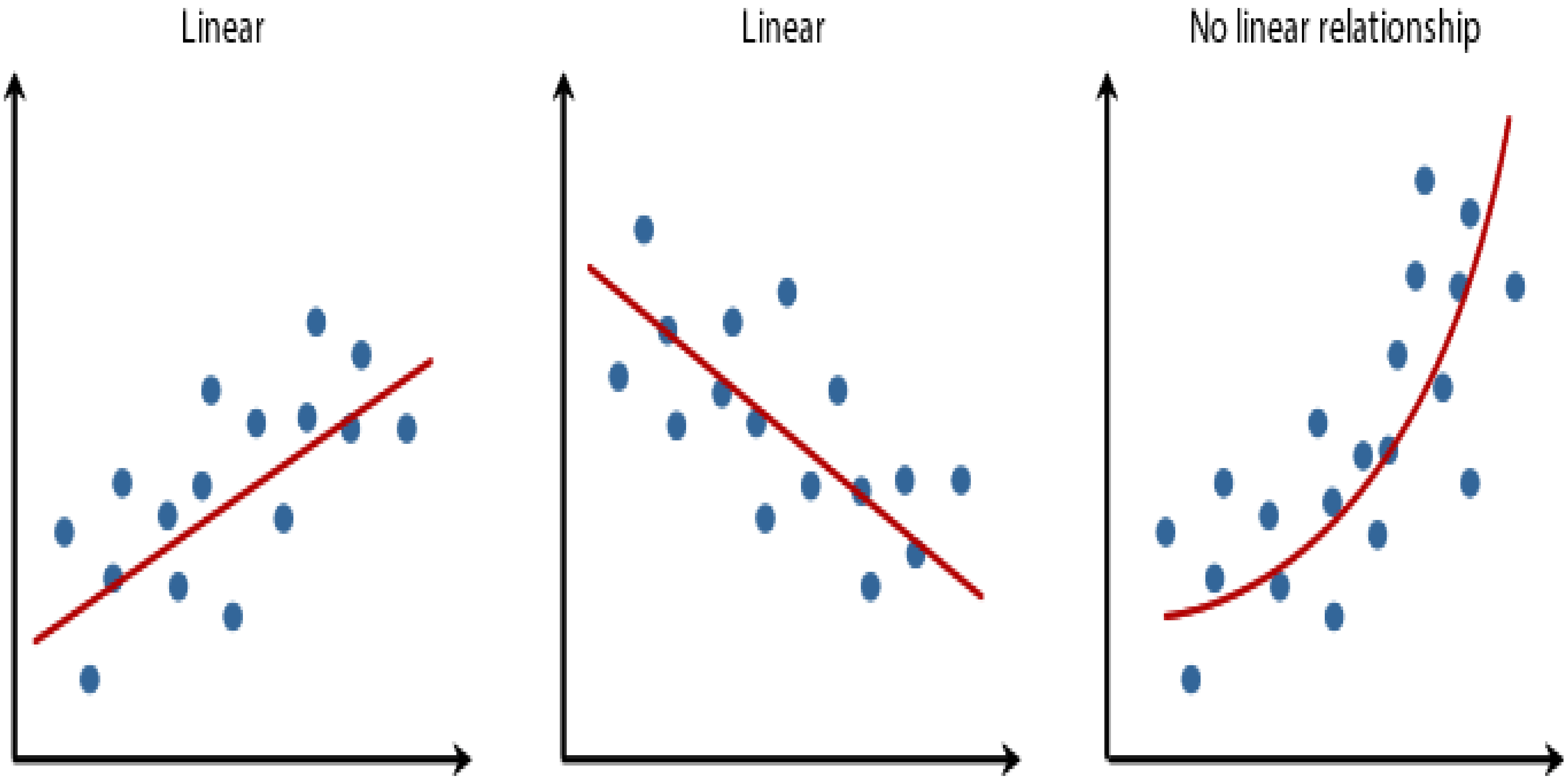
|  |  |  |  |
|---|---|---|---|
| 1 | -3 | 2 | -2 |
| -2 | 0 | 4 | 3 |
| -1 | -2 | -2 | -3 |
| -2 | 4 | 4 | 2 |

$X^{act}$

|  |  |  |  |
|---|---|---|---|
| 1 | 0 | 2 | 0 |
| 0 | 0 | 4 | 3 |
| 0 | 0 | 0 | 0 |
| 0 | 4 | 4 | 2 |

$Y^{act}$

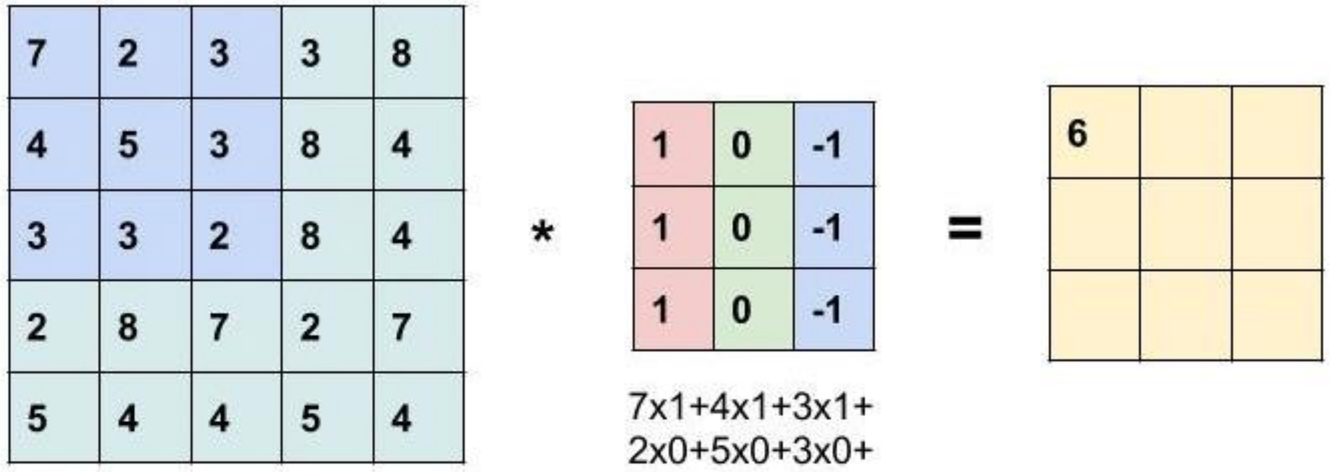
# How a activation function helps the neural network ?



# Some famous activation functions

| Activation functions | Uses |
|---|---|
| Relu | Most commonly used in hidden layers |
| Sigmoid | Binary classification |
| Softmax | Categorical classification |
| Tanh | They seem to provide some good results while using in some situations with LSTM based models |

# Convolution layer



| | | | | |
|---|---|---|---|---|
| 7 | 2 | 3 | 3 | 8 |
| 4 | 5 | 3 | 8 | 4 |
| 3 | 3 | 2 | 8 | 4 |
| 2 | 8 | 7 | 2 | 7 |
| 5 | 4 | 4 | 5 | 4 |

*

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

=

| | | |
|---|---|---|
| 6 | | |
| | | |
| | | |

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

Image

Kernel

Output

# Max Pooling

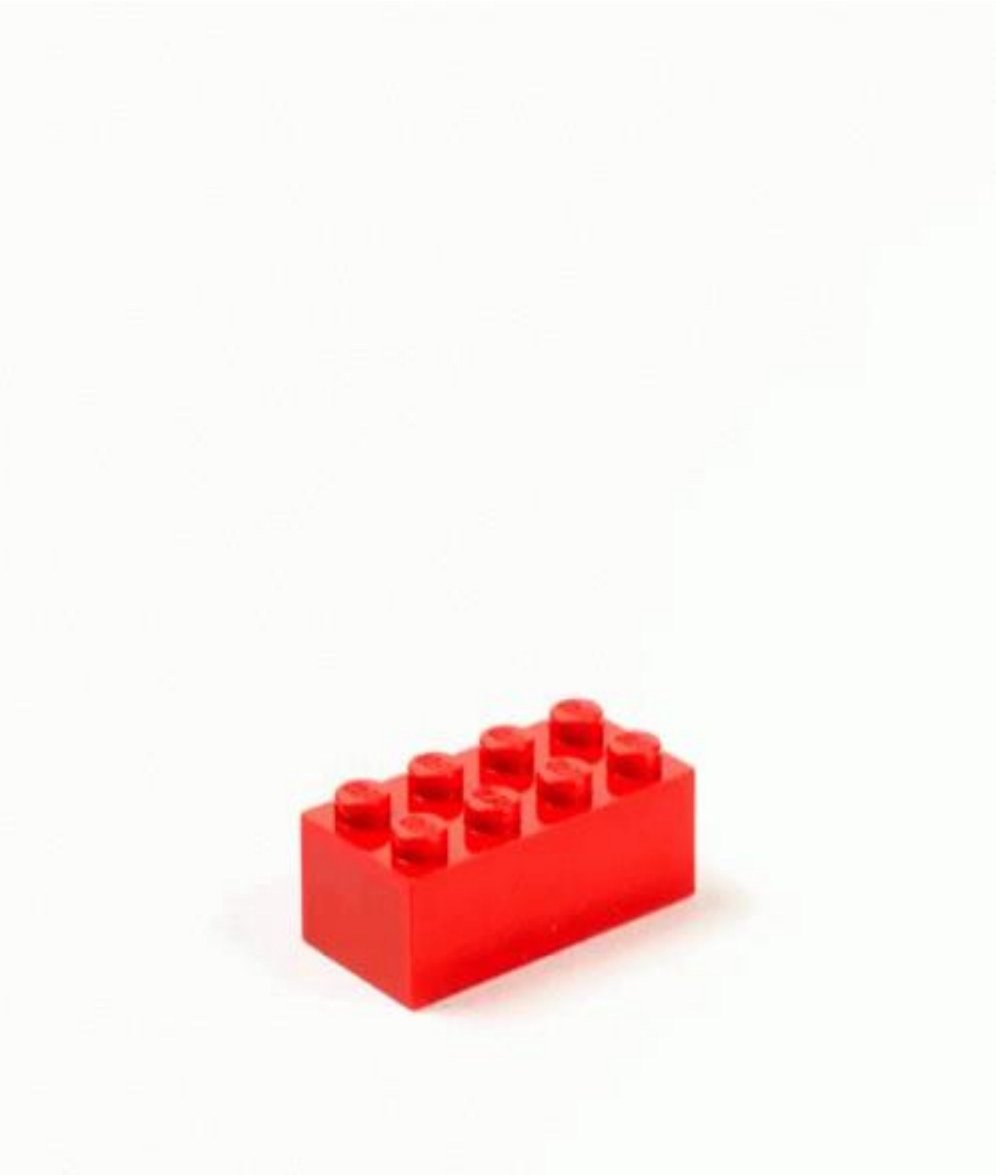
|   |   |   |   |
|---|---|---|---|
| 4 | 6 | 1 | 1 |
| 1 | 3 | 1 | 3 |
| 4 | 0 | 0 | 8 |
| 8 | 5 | 4 | 0 |

Input

|   |   |
|---|---|
|   |   |
|   |   |

Output

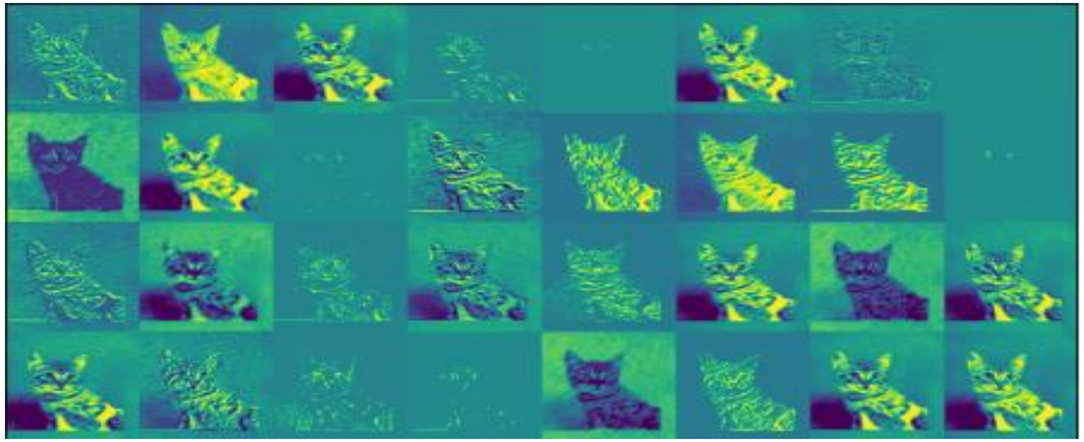
# Just like a Lego



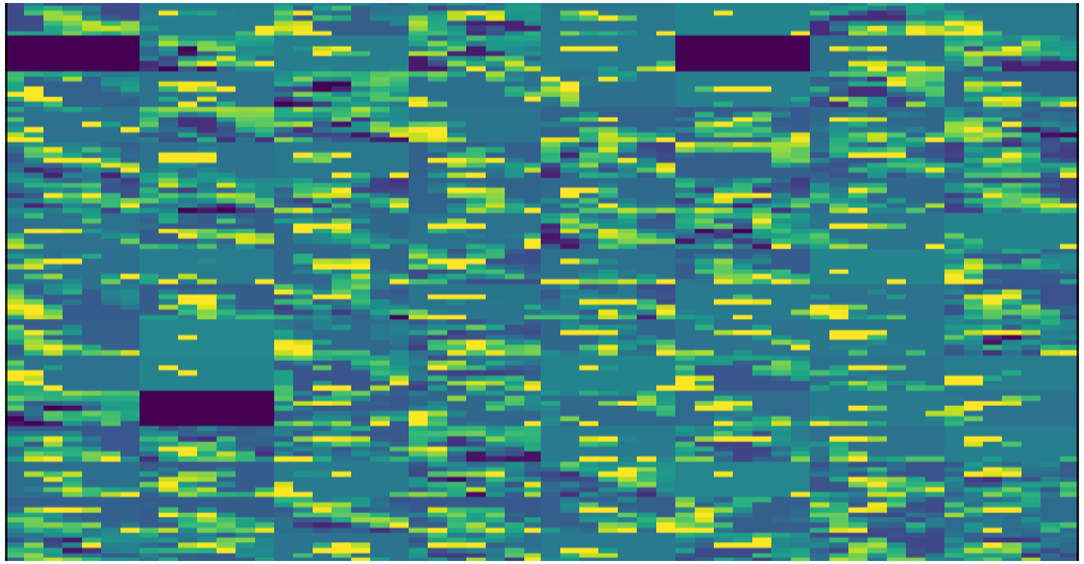
You can think of the layers as a Lego block that is used to build a bigger structure(DL model)

# Feature visualization

Input



Top layers



Deeper layers

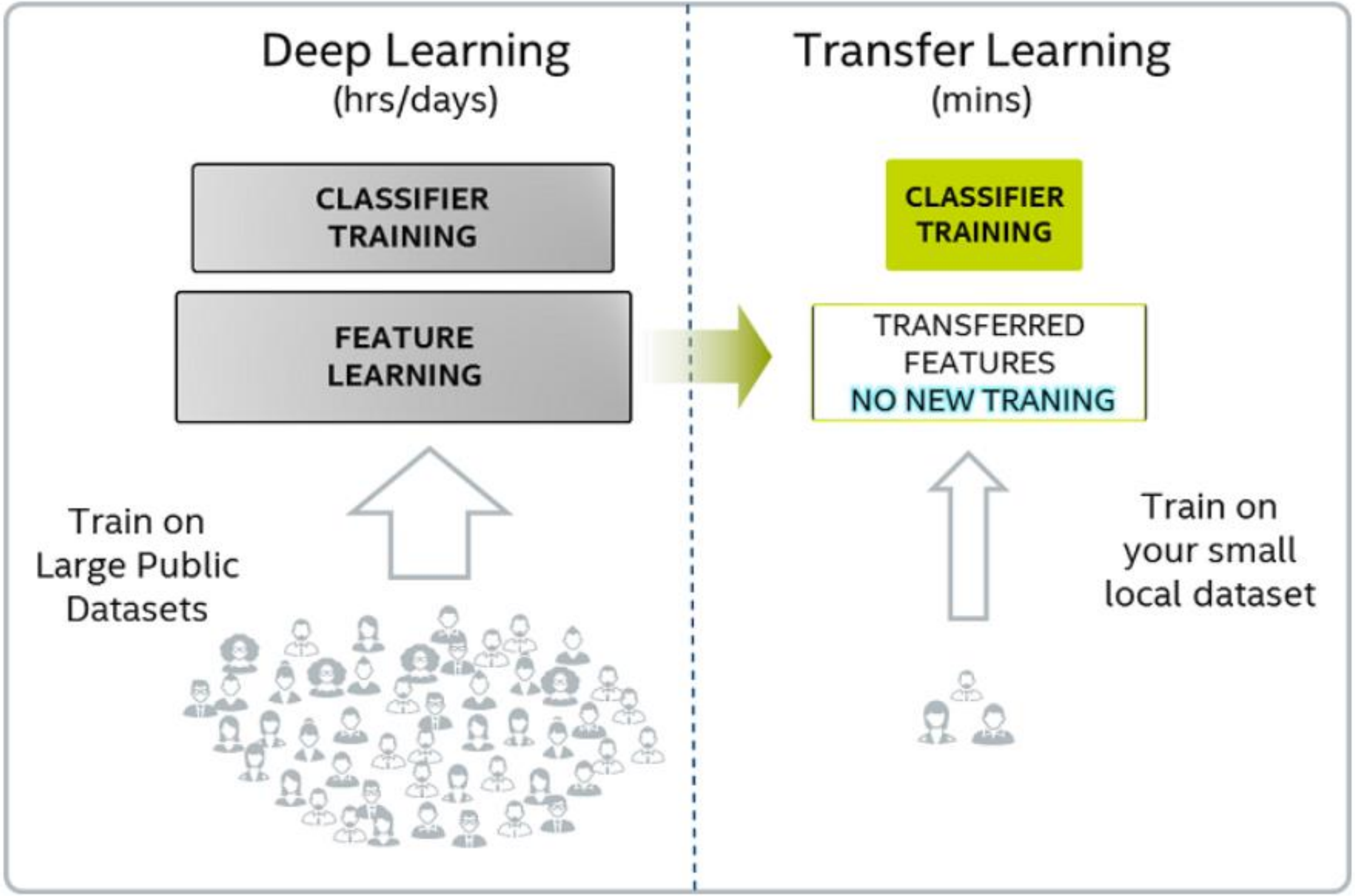
# Types of CV tasks & Suitable archs

| Task | Architecture |
|---|---|
| Classification | Alexnet |
| Classification | VGG16 |
| Object detection | YOLO |
| Object detection | RCNN |
| Segmentation | U-NET |
| Segmentation | Masked RCNN |

# Types of NLP tasks & Suitable archs

| Task | Architecture |
|---|---|
| Tokenization and Text Classification | CNN/RNN based models |
| Generating Captions for Images | Google NIC |
| Machine Translation | RNN based networks |
| Question Answering | RNN based networks |
| Document Summarization | Encoder Decoder arch using RNN |
| Speech Recognition | CNN/RNN based models |

# Transfer learning



Deep Learning
(hrs/days)

CLASSIFIER TRAINING

FEATURE LEARNING

Train on Large Public Datasets

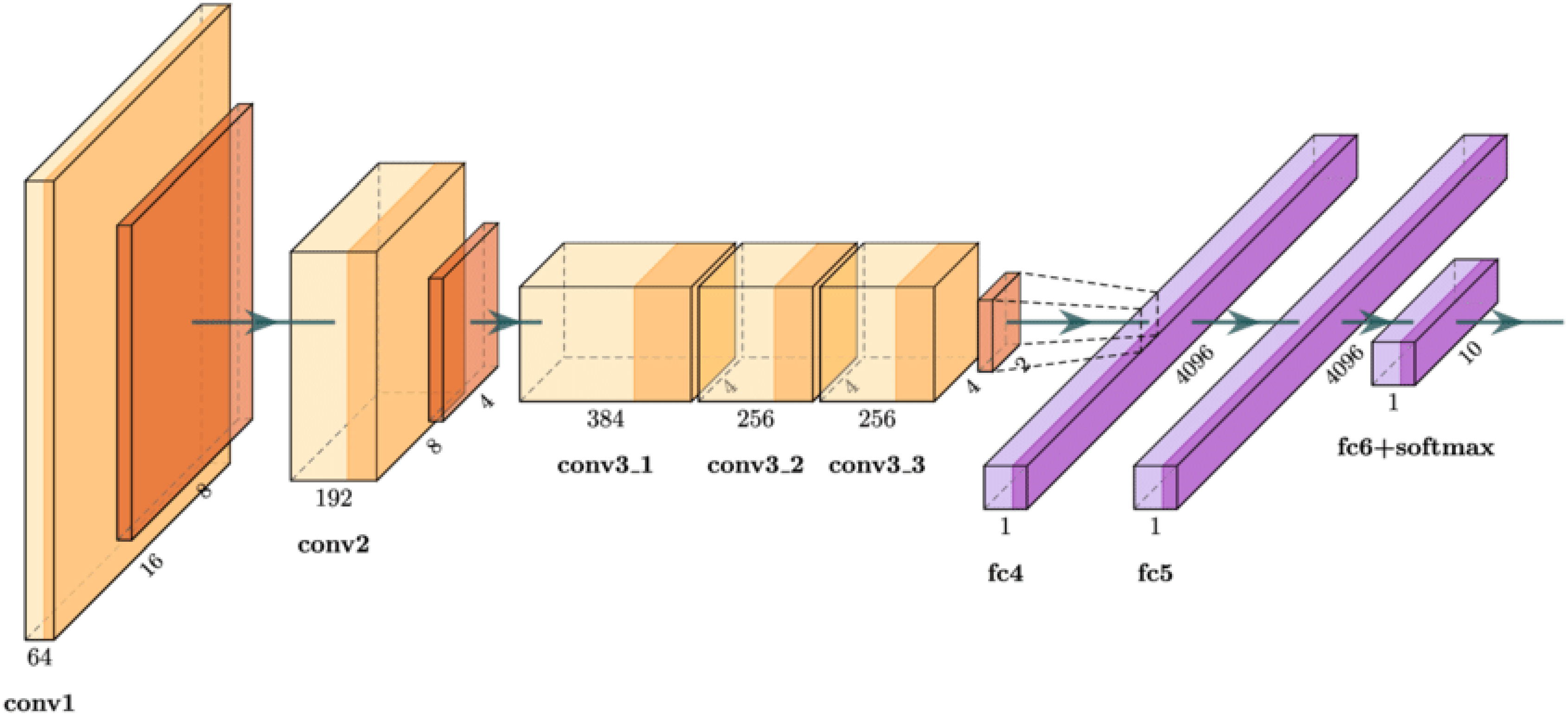
Transfer Learning
(mins)

CLASSIFIER TRAINING

TRANSFERRED FEATURES
NO NEW TRANING

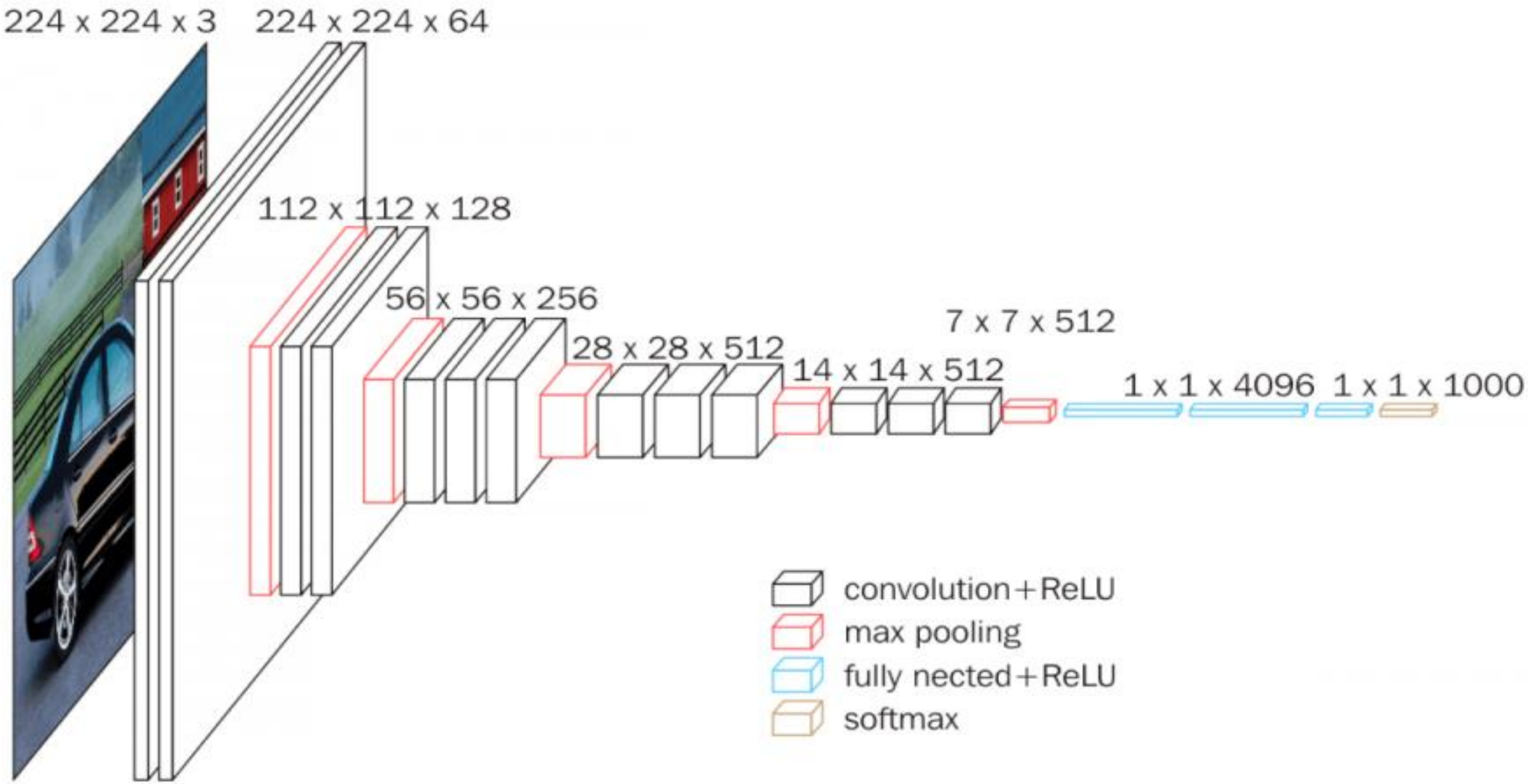
Train on your small local dataset

# Architecture suitable for our project



Alex Net

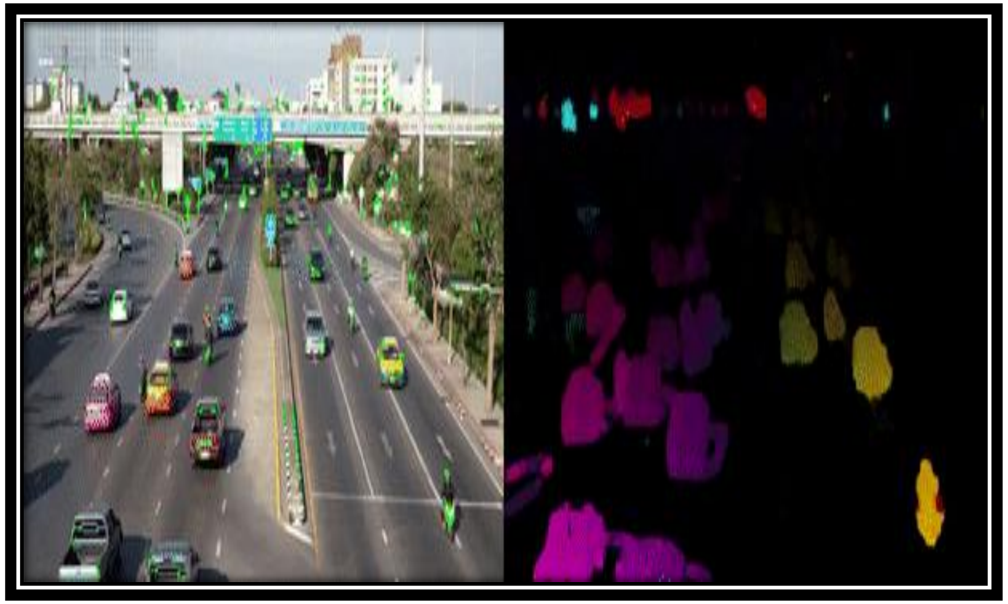
# Architecture suitable for our project



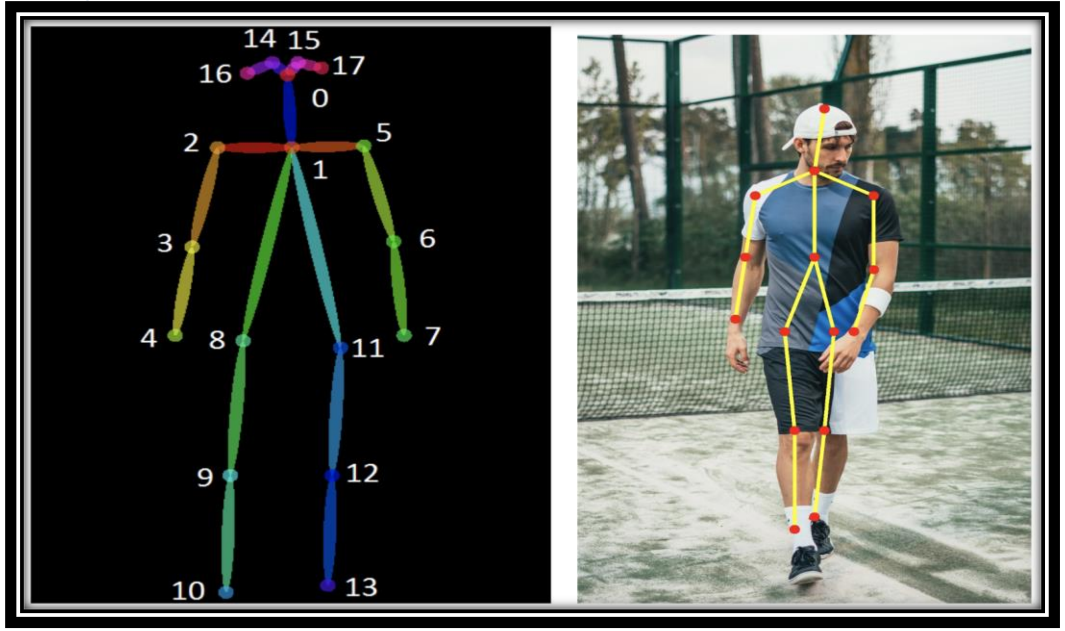
VGG 16

# Selecting a architecture

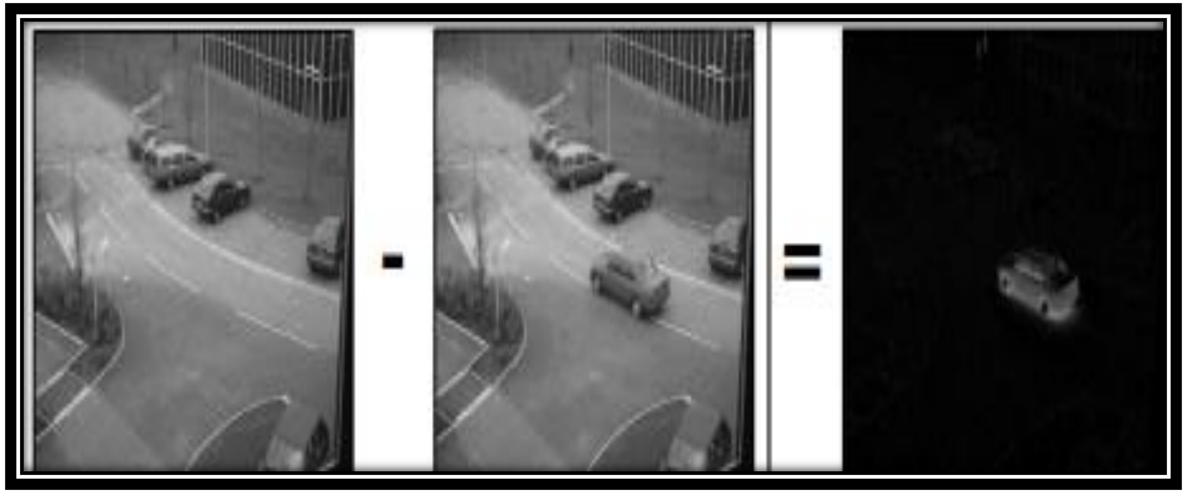
# Ways to approach the problem



Dense optical flow

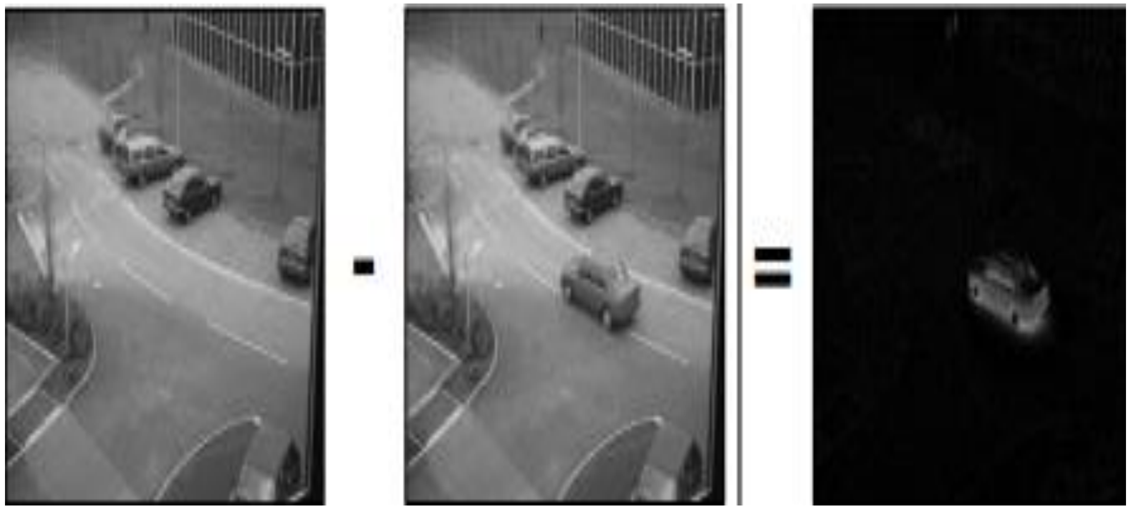


Pose prediction



Absdiff

# The approach we are going to use



Absdiff based

# Things we need to check

- ➢Size of the dataset we are going to use.
- ➢Total number of parameters in the model.
- ➢Where are we going to deploy the model.
- ➢Overall file size.
- ➢Compatibility for the lib we use.
- ➢Pre-processing required for the dataset based on the arch.
- ➢Inference speed(Compute time).
- ➢Training time.
- ➢Hardware requirement.

# Our choice

The architecture that we are going to choose is

VGG-16

# FAQ

## Do we need to implement all this math to build a dl model ?

- ➢Generally there is no need to use math to build a DL model.
- ➢There are many libraries present now that we can build a model without help of any math. But a understanding of the basics will always help.

## Can we implement our own architecture instead using already existing one ?

- ➢Definitely you can, There are many ways you can build a model its totally up to you to decide what layers to use and how to use them.
- ➢You can even publish the network you created as paper and name it as you like.

# TROUBLE SHOOTING & TIPS

- Colab provides you with GPU runtime which you can use to train your models faster.
- You can also attach your google drive to colab and use the dataset from there.
- There are multiple ways to build a model in keras the sequential api allows you to build a model sequentially in colab. You can use the  functional api to build a  model as collection of functions which gives more flexibility. Finally if you need even more customizability you can even create your own layers using objects.

# References

## Basic concepts of DL & ML

- https://www.youtube.com/playlist?list=PLQY2H8rRoyvwWuPiWnuTDBHe7I0fMSsfOhttps://docs.opencv.org/4.x/d9/df8/tutorial_root.html
- https://www.youtube.com/playlist?list=PLQY2H8rRoyvwWuPiWnuTDBHe7I0fMSsfO

## Tensorflow & Keras

- https://www.tensorflow.org/api_docs/python/tf/keras

## VGG 16

- https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918

## Alex-net

- https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet

## Advanced architectures

- https://www.analyticsvidhya.com/blog/2017/08/10-advanced-deep-learning-architectures-data-scientists/