Sponsored by:

DATA SCIENCE INSTITUTE

The World Data Science Institute is a Financial Data Science Research & Development Company

FINANCIAL DATA SCIENCE TEAM

Data Science Manager - Anade Davis

Blockchain Engineer/Project Lead - Alberto Navarrete

Financial Data Scientist -Muhammad Jawwad Javeed Iqbal

Financial Data Scientist - Moxú

Financial Data Scientist - Mahmoud Hosny

Financial Data Scientist - Amirhossein Abaskohi

- Financial Data Scientist
  - https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/
  - https://www.sciencedirect.com/science/article/abs/pii/S1077314219301183
  - https://www.hitechnectar.com/blogs/deep-learning-face-recognition/
  - https://analyticsindiamag.com/top-8-algorithms-for-object-detection/

NOTES:

Training works better, but there are other methods like using HOG3D descriptor and color histogram. These can give better spacio-temporal features. For more details you can go through CVPR papers.

The principle behind deep learning can be applied to most of the problem domains. In the context of the problem stated by you, I believe you would still need to use a face detection mechanism, let it be HOG, SIFT, Haarscale detector etc.. The 'deep' learning mechanism would be needed to either to give feature vector representation (unsupervised learning) or labels (supervised learning) to the image frames.

These feature vectors can either be learnt using a deep learning mechanism or can be hand crafted like the HOG or LoG or anything. So, principally, if you do not want to hand craft these features, use deep learning (unsupervised) after extracting a face from the video frame or otherwise split the video into frames, extract face images, apply HOG/LoG etc. and extract feature vectors and then feed it to a multi-layered network giving it a label for recognition. Let me know if this helps, I can perhaps give you a better response as well based on your specific questions.

Subjects were grouped into four categories: darker-skinned females, darker-skinned males, lighter-skinned females, and lighter-skinned males. All three algorithms performed the worst on darker-skinned females, with error rates up to 34% higher than for lighter-skinned males (Figure 1). Independent assessment by the National Institute of Standards and Technology (NIST) has confirmed these studies, finding that face recognition technologies across 189 algorithms are least accurate on women of color.

Each team was required to use the same dataset, which consisted of 152,917 photos of 6,139 males and females ranging in age from under 34 to over 65. AnyVision annotators labeled images according to age, skin color, and other attributes, with multiple annotators verifying the labels for accuracy before the dataset was divided into training, validation, and testing subsets.

How might we go about writing an algorithm that can classify images into distinct categories? Computer Vision researchers have come up with a data-driven approach to solve this. Instead of trying to specify what every one of the image categories of interest look like directly in code, they provide the computer with many examples of each image class and then develop learning algorithms that look at these examples and learn about the visual appearance of each class.

In other words, they first accumulate a training dataset of labeled images, then feed it to the computer to process the data. Given that fact, the complete image classification pipeline can be formalized as follows:

Our input is a training dataset that consists of N images, each labeled with one of K different classes.

Then, we use this training set to train a classifier to learn what every one of the classes looks like.

In the end, we evaluate the quality of the classifier by asking it to predict labels for a new set of images that it's never seen before. We'll then compare the true labels of these images to the ones predicted by the classifier.

Best Neural Networks & Algorithms for Image Recognition

Data Science Report

September 2021

Table of Contents

- Introduction into Deep Learning and Neural Networks.
- Single Layer and MultiLayer Feed Forward Networks
- Convolution Networks
- Architecture of a FeedForward Network
- Activation Functions
- The problem with feedforward Neural Networks
- Convolutions and Pooling Layers
- Structure of Convolutional Neural Network
- Activation Functions
- Inception V3/Loading Models

Table of Contents 1.2

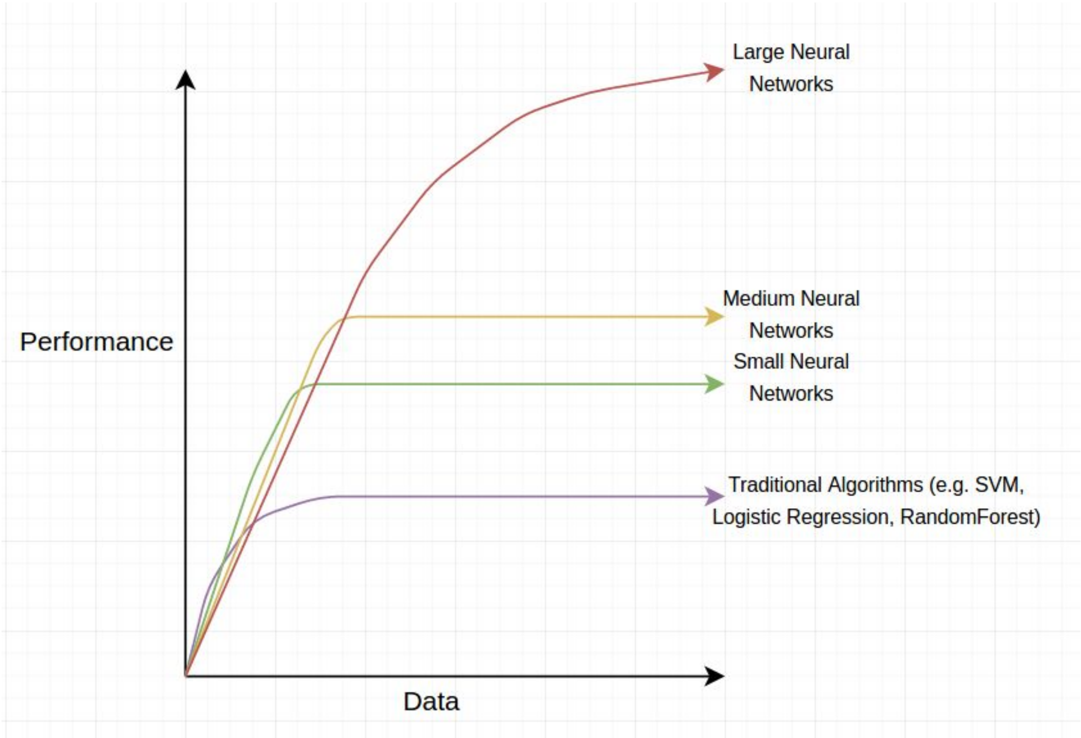
- Fast Image Restoration with Multi-Bin Trainable Linear Units
- Face recognition with Bayesian convolutional networks for robust surveillance systems.
- An ELU Network with Total Variation for image denoising
- ELU vs ReLU for Image Recognition
- Face recognition Based on Deep Learning
- Type & process of Image Recognitions
- Algorithms of DL of Face Recognitions
- Challenges in Image Detection
- Databases Sources

Table of Contents 1.2

- Image Processing using convolutional neural networks
- Problem with multiple channels
- Why ReLU for Image Processing?
- Role of Pooling Layer
- Image Flattening
- Object Detection
- Object Tracking
- Animal Face Classification using Dual Deep Convolutional Neural Network
- Several tests results

Anade Davis - Data Science Manager - Points of Discussion

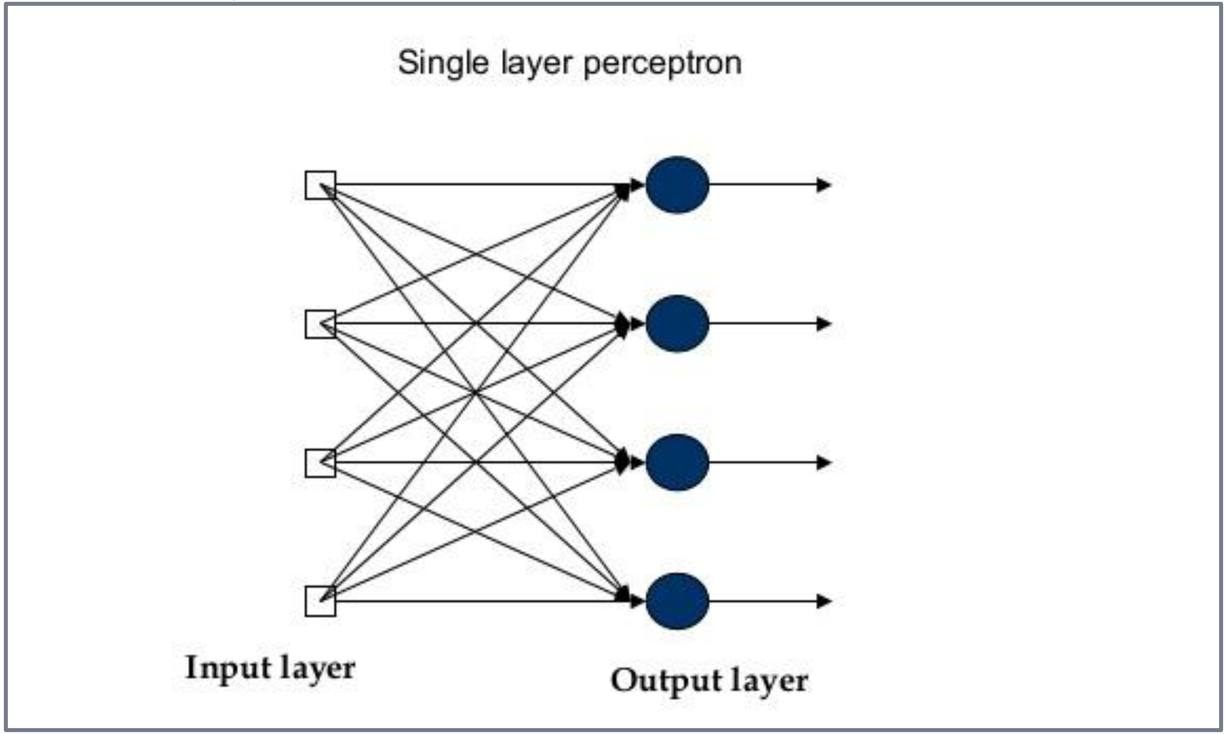
- Introduction into Deep Learning and Neural Networks.
- Single Layer and MultiLayer Feed Forward Networks
- Convolution Networks
- Architecture of a FeedForward Network
- Activation Functions
- Conclusion



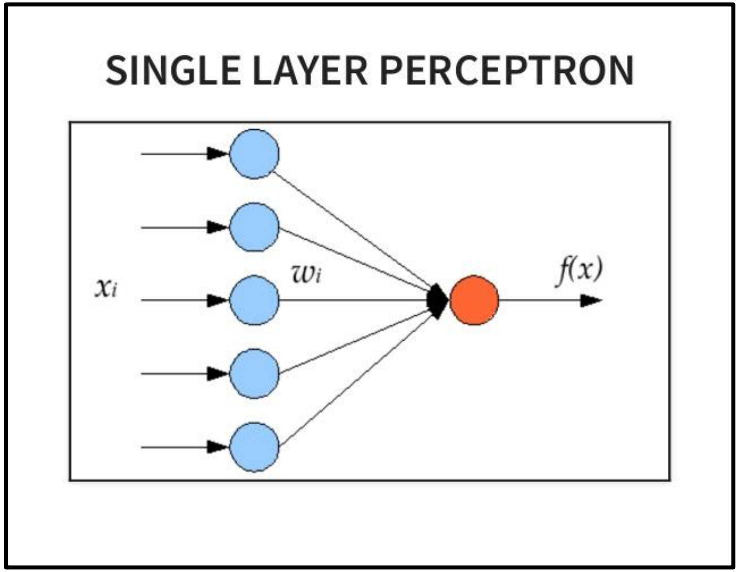
Deep Learning is a more advanced form of Machine Learning

- **Deep Learning is possible because of Neural Networks**
    - Deep Learning closely mimics the Human Brain
    - Deep Learning has basically improved upon Machine Learning
    - Deep Learning has been incredibly helpful with managing bigger amounts of Data
    - Deep Learning works by a series of layers
    - These layers are responsible for finding correlations and patterns
    - These is where Deep Learning earns its reputation
    - Some Deep Learning algorithms have more complex layer patterns than others
    - Deep Learning had become very useful in the medical industry and etc.
    - The issue is you need a powerful computer/internet connection
    - These layers help discover correlations previously impossible to correlate
        - Especially within video. Image, and audio

- **Perceptron** is an High Level algorithm with Supervised Learning
    - Perceptron comes in 2 different Models for processing
        - **Single Layer Perceptron**
        - Multi Layer Perceptron



- **Perceptron** is an High Level algorithm with Supervised Learning
  - Perceptron comes in 2 different Models for processing
    - **Single Layer Perceptron**
      - Called Feed Forward Networking
      - Form of Artificial Neural Network
      - Only has 2 layers
      - Input Layer (Blue Balls)
      - Output Layer (Orange Ball)
      - Used for Simple Analysis

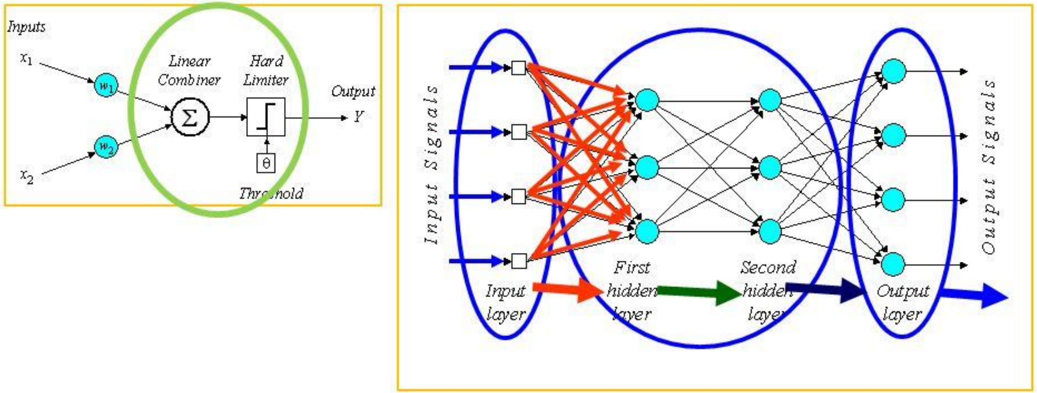


DATA SCIENCE INSTITUTE

- **Perceptron** is an High Level algorithm with Supervised Learning
  - Perceptron comes in 2 different Models for processing
    - Single Layer Perceptron
    - **Multi Layer Perceptron**

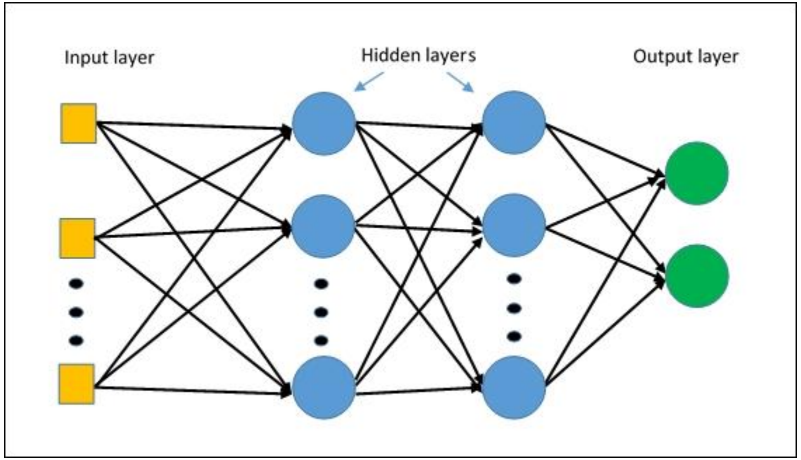
Multilayer Neural Networks

- A multilayer perceptron is a feedforward neural network with ≥ 1 hidden layers.

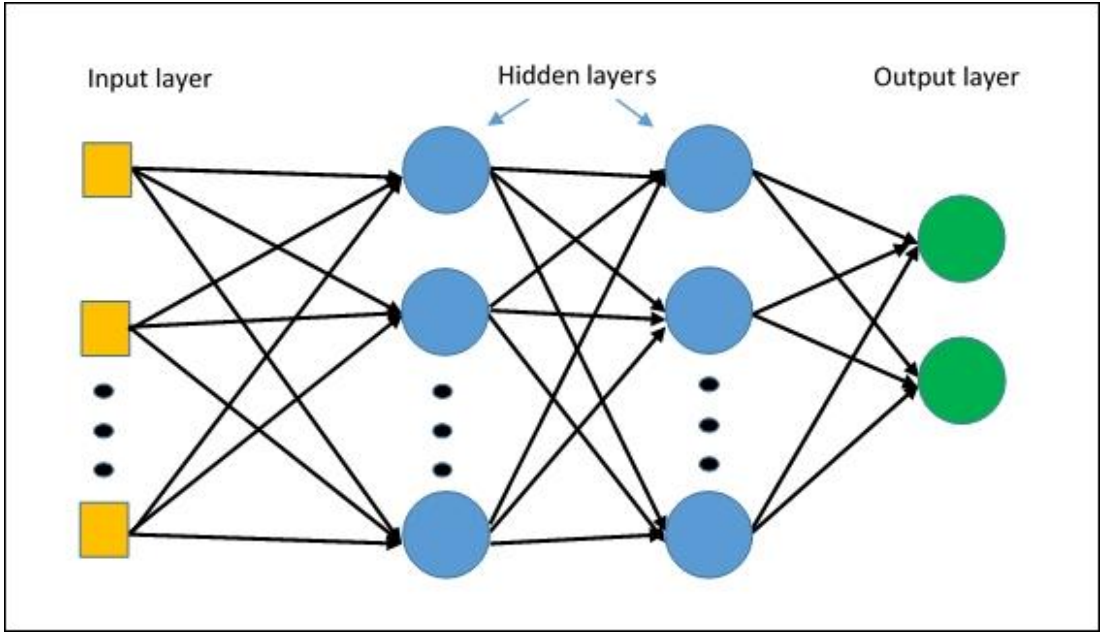


Single-layer VS Multi-layer Neural Networks

- Perceptron comes in 2 different Models for processing
    - **Multi Layer Perceptron (is a advanced Convolutional Neural Network)**
        - Called Feed Forward Networking
        - Form of Artificial Neural Network
        - Has more than 2 layers
        - Input Layer (**Visual Layer**)
        - Hidden Layer (1 or multi)
        - Output Layer
        - Used for Advanced Analysis

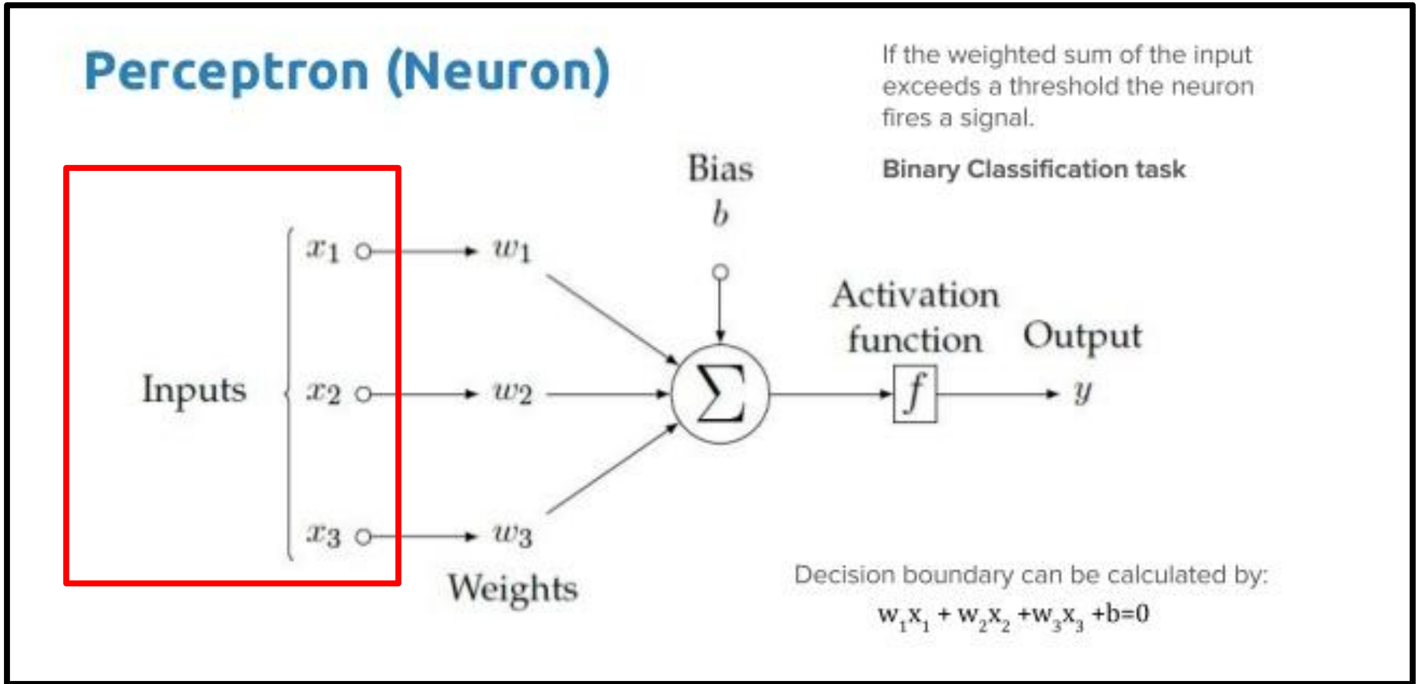


- **Multi Layer Perceptron**
    - The Input Layer or Visual Layer is the initial layer (the raw data)
    - The Hidden Layers start grouping things by weights (parameters) to figure out correlations
    - Output Layer is the final output ready for analysis
    - These hidden layers are made up of Neurons



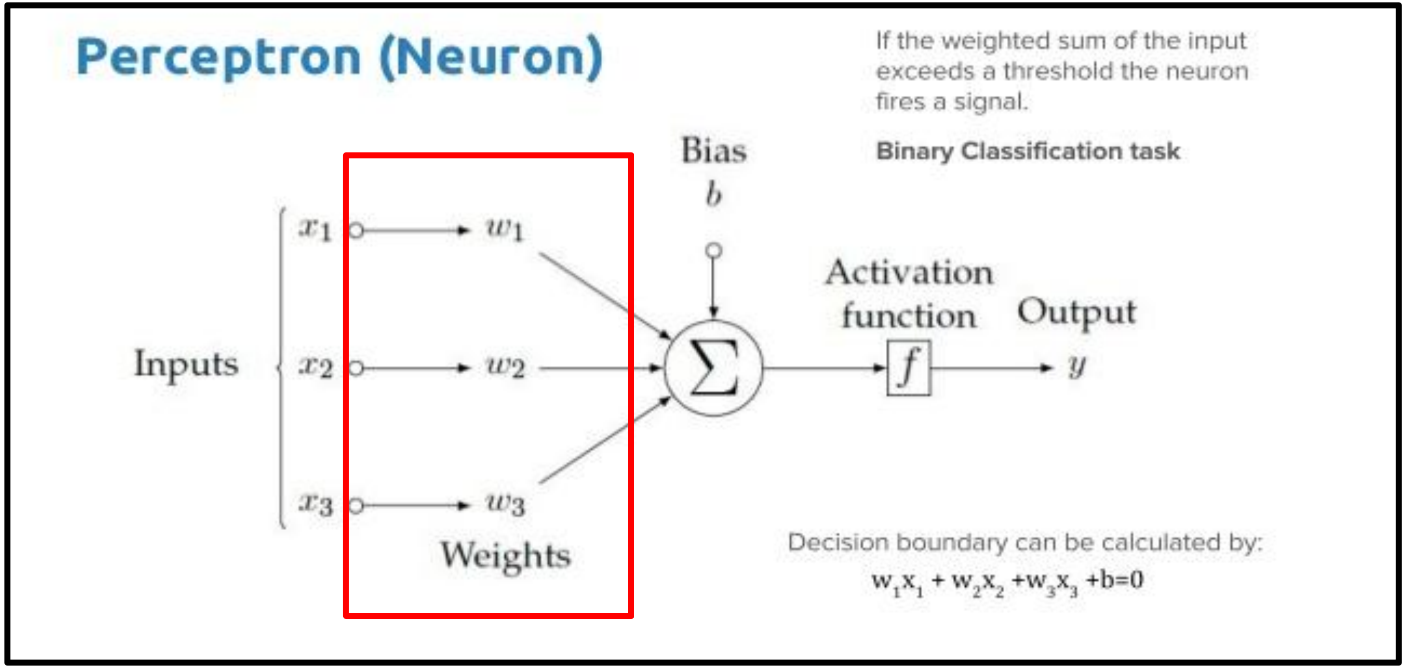
DATA SCIENCE INSTITUTE

- **Deep Learning is possible because of Neural Networks and Neurons**
  - Neurons are also referred to as Artificial Neurons
  - A Perceptron is a type of model used in Deep Learning
  - The first part is what's called the Inputs or the Data Layer



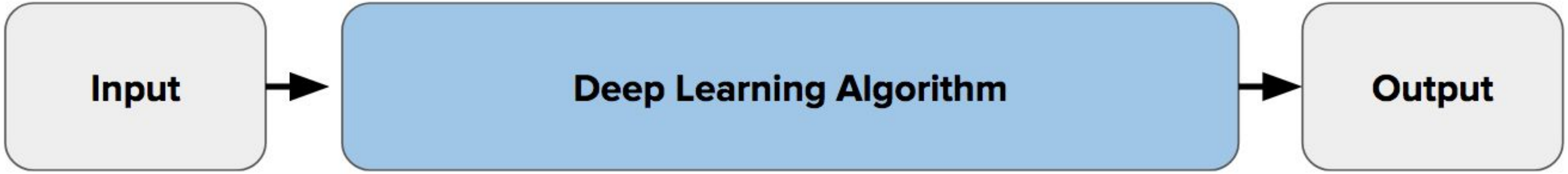
DATA SCIENCE INSTITUTE

- **Deep Learning is possible because of Neural Networks and Neurons**
    - The second part is called the **Weights**
    - Weights near zero mean changing this input will not change the output. Many networks will automatically set those weights to zero in order to simplify the network.
    - The Weights are calculated numerically to automatically detect correlations between features (**called Feature Learning**)



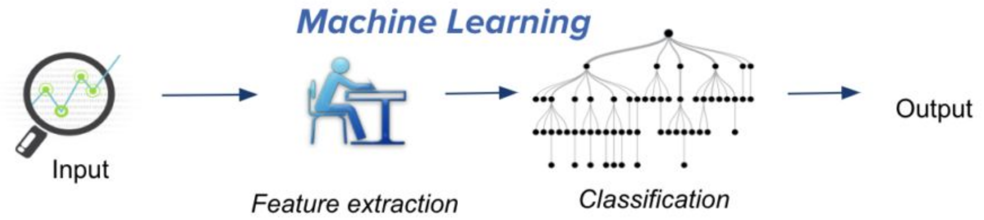
- **Deep Learning is possible because of Neural Networks**
  - Previously we would have to do test several machine learning techniques/algorithms
    - Now Deep Learning does this for us

Traditional Machine Learning Flow

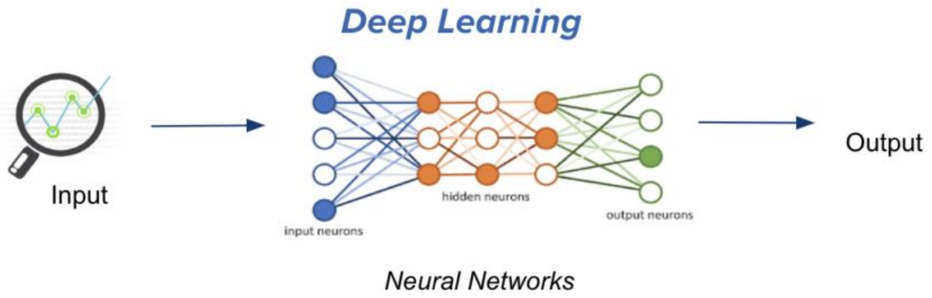


Deep Learning Flow

- **Deep Learning is possible because of Neural Networks and Neurons**
  - Previously we would have to do test several machine learning techniques/algorithms (now Deep Learning does this for us)
  - This is referred to as **Feature Learning** also called **Representation Learning**
    - This is the alternative to the manual way called **Feature Engineering**



Traditional machine learning uses hand-crafted features, which is tedious and costly to develop.



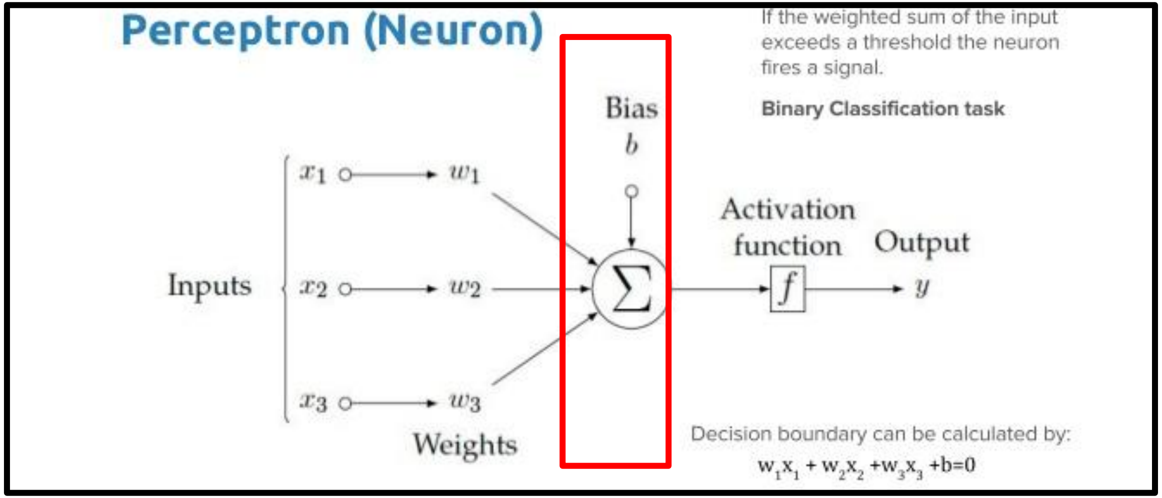
Deep learning learns hierarchical representation from the data itself, and scales with more data.

- Deep learning is more than a algorithm, its a network of different architectures
- Deep Learning can be used for Supervised, Unsupervised, and Reinforcement Learning
- In Deep Learning Features (Relevant columns) are automatically chosen for best output
- This is called automatic Feature Extraction **(Feature Learning)**
- The more data the better
- These networks vary by the use case
- Powerful computation is needed to properly implement a Neural Network

As Howard Rheingold said, "The neural network is this kind of technology that is not an algorithm, it is a network that has weights on it, and you can adjust the weights so that it learns. You teach it through trials."

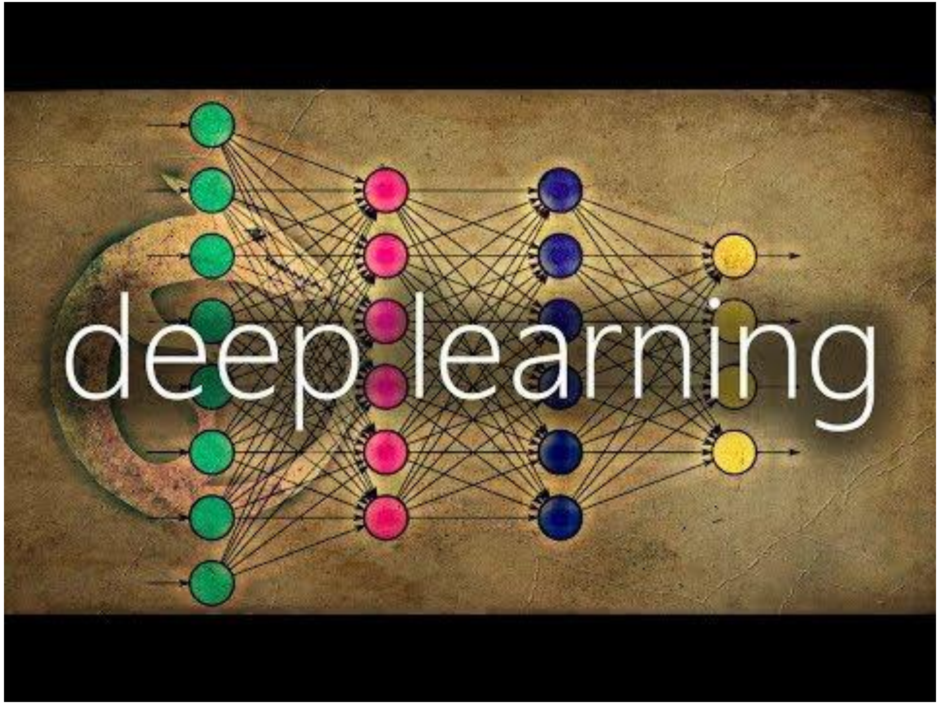
Deep Learning is possible because of Neural Networks and Neurons

- The third part is called the **Sum** (helps identify the pattern and correlations of data)
- **Bias** - the measurement our predictions are from real values
  - A high bias (underfitting) makes the model faster and easier to perform (but less accurate)
  - A low bias (overfitting) makes model slower and harder to perform (but possibly more accurate)
- Bias It takes the weighted number and sends to Activation Function
- All can be edited based on Optimizers (example: SGD, Backpropogation) or Neural Network choice



Deep Learning is possible because of Neural Networks and Neurons

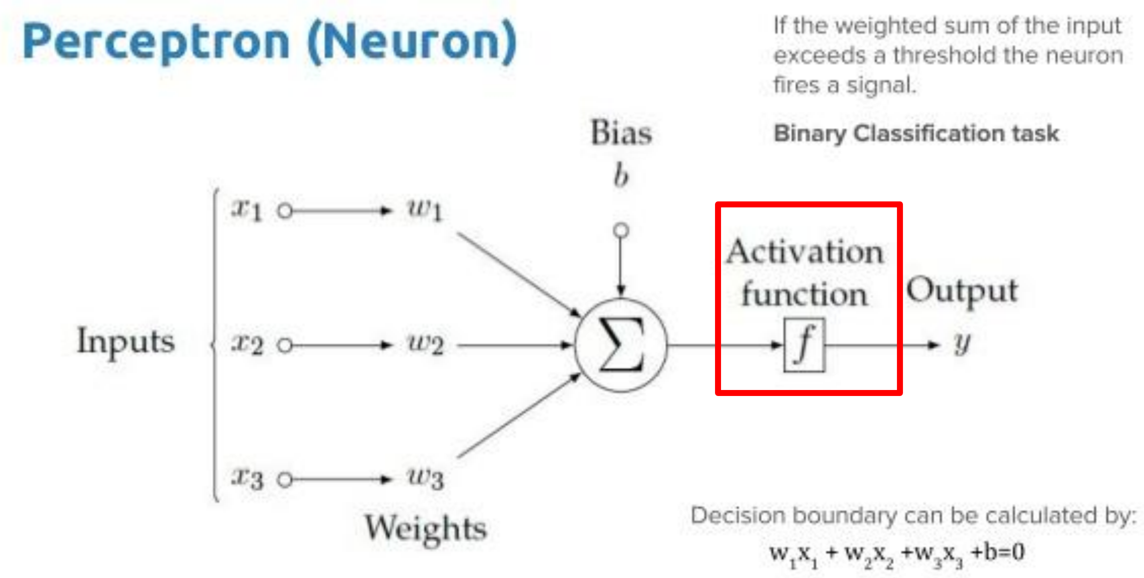
- **Bias** - the measurement our predictions are from real values
  - High bias (underfitting) makes model faster and easier to perform (less accurate)
  - Low bias (overfitting) makes model slower and harder to perform (but possibly more accurate)



DATA SCIENCE INSTITUTE

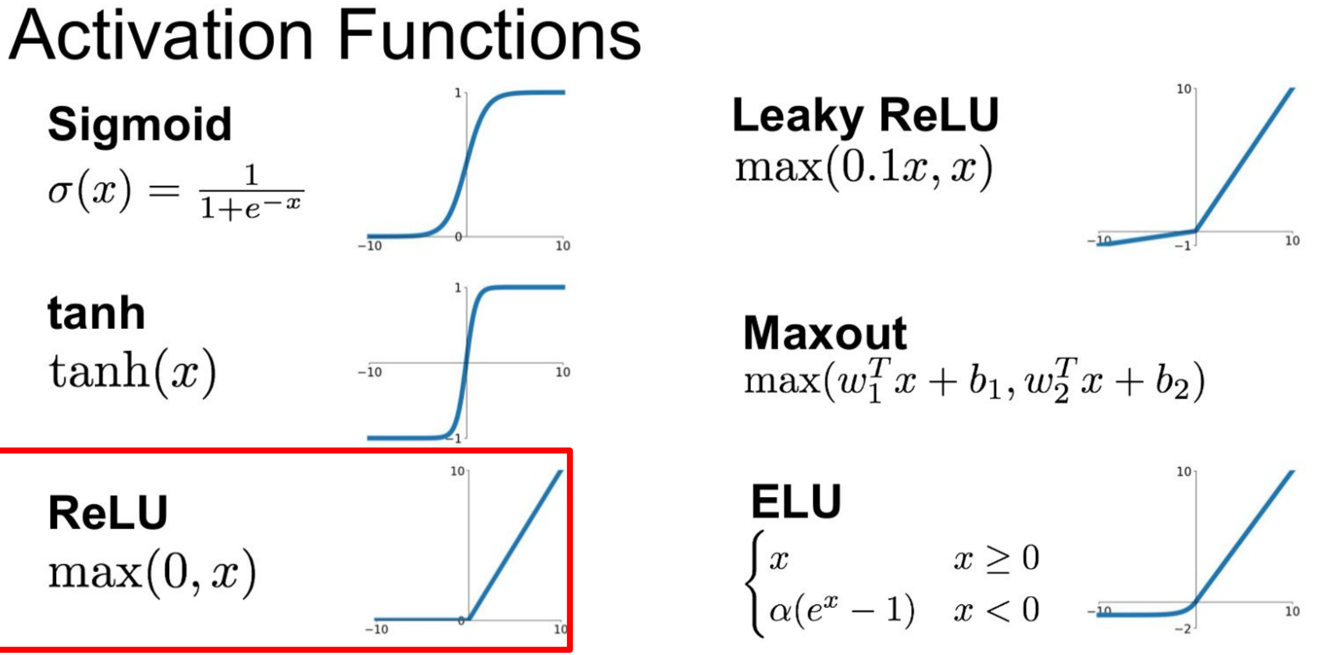
Deep Learning is possible because of Neural Networks and Neurons

- **Neurons** are also referred to as **Artificial Neurons**
- The Activation Function reads non linear (non numerical) also
  - ReLU function only linear (numerical) data can be used
  - ReLU is the default Activation Function for a CNN network



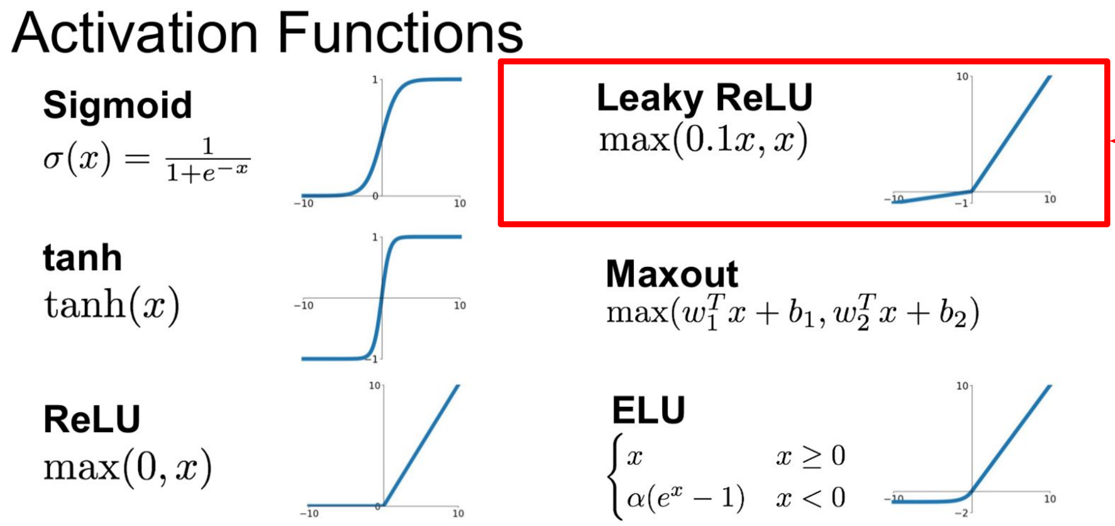
Deep Learning is possible because of Neural Networks and Neurons

- Types of Activation Functions
  - ReLU function only linear (numerical) data can be used
    - ReLU is the default Activation Function for a CNN network



Deep Learning is possible because of Neural Networks and Neurons

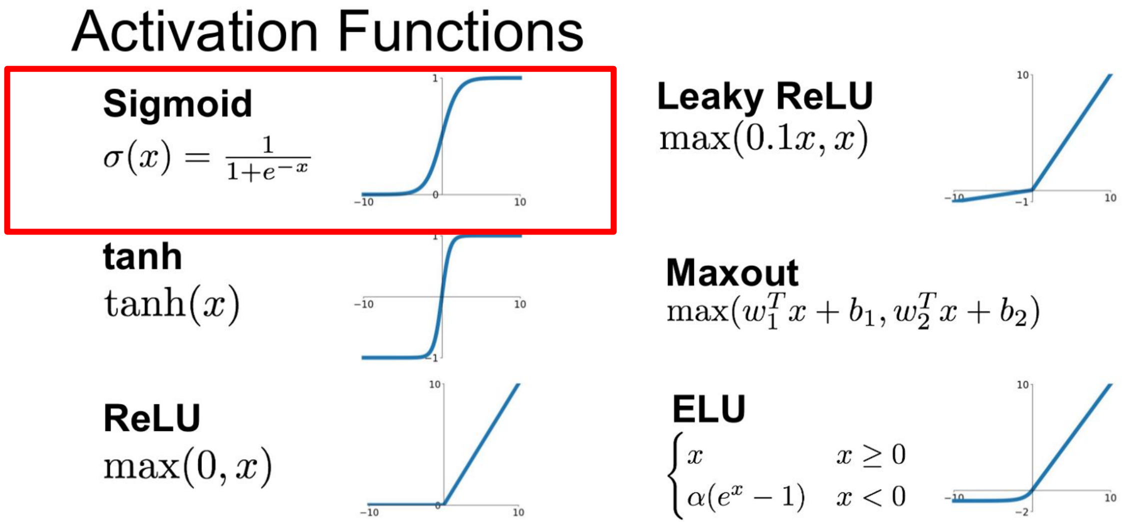
- Types of Activation Functions
  - **Leaky ReLU Function (used when Backpropagation is added)**
    - Used as an enhancement to ReLU
      - (Rectified Exponential Linear Unit)



Linear plots is for numerical models

Deep Learning is possible because of Neural Networks and Neurons

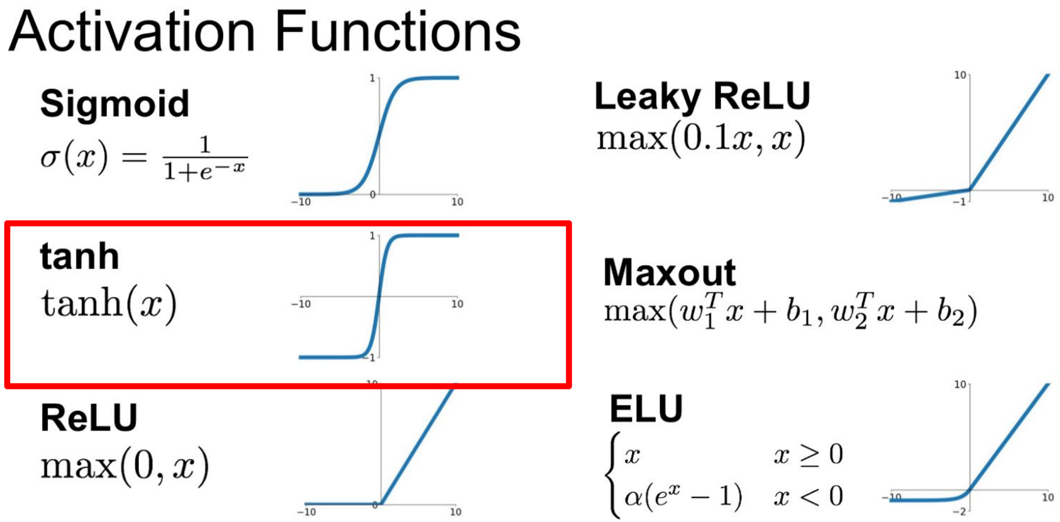
- Types of Activation Functions
  - **Sigmoid Function (also known as Logistic Sigmoid)**
    - Used for Classification or Binary classification (Categorical)
    - Bi means two (Example: Cat or Dog/ Win or Lose)



Deep Learning is possible because of Neural Networks and Neurons

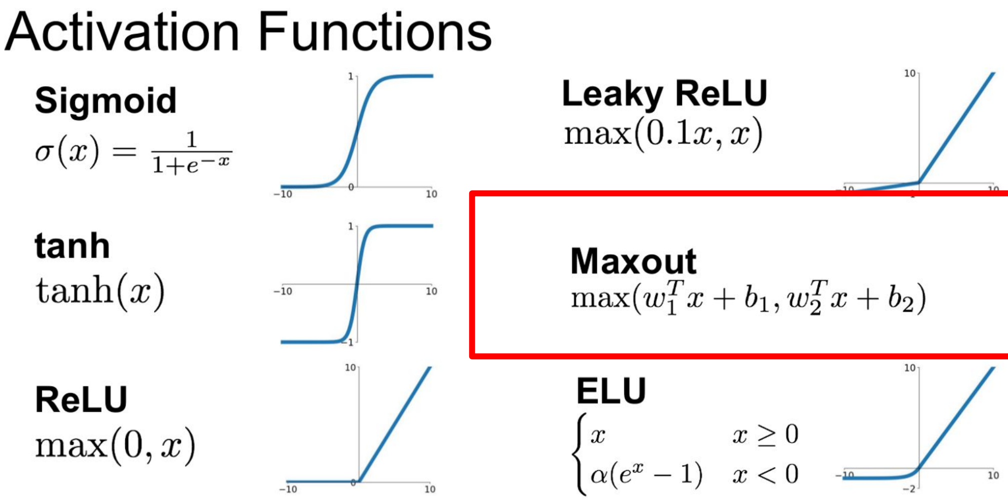
- Types of Activation Functions
  - **Tanh Function (Considered more accurate than Sigmoid)**
    - Used for Classification or Binary classification (Categorical)
    - Bi means two (Example: Cat or Dog/ Win or Lose)

S plots for categorical models



Deep Learning is possible because of Neural Networks and Neurons

- Types of Activation Functions
  - **Maxout (also known as Softmax function)**
    - Used as a function for our last layer
    - Used when we need numbers to predict a class or category
      - Ex. 5 stars equal a great movie

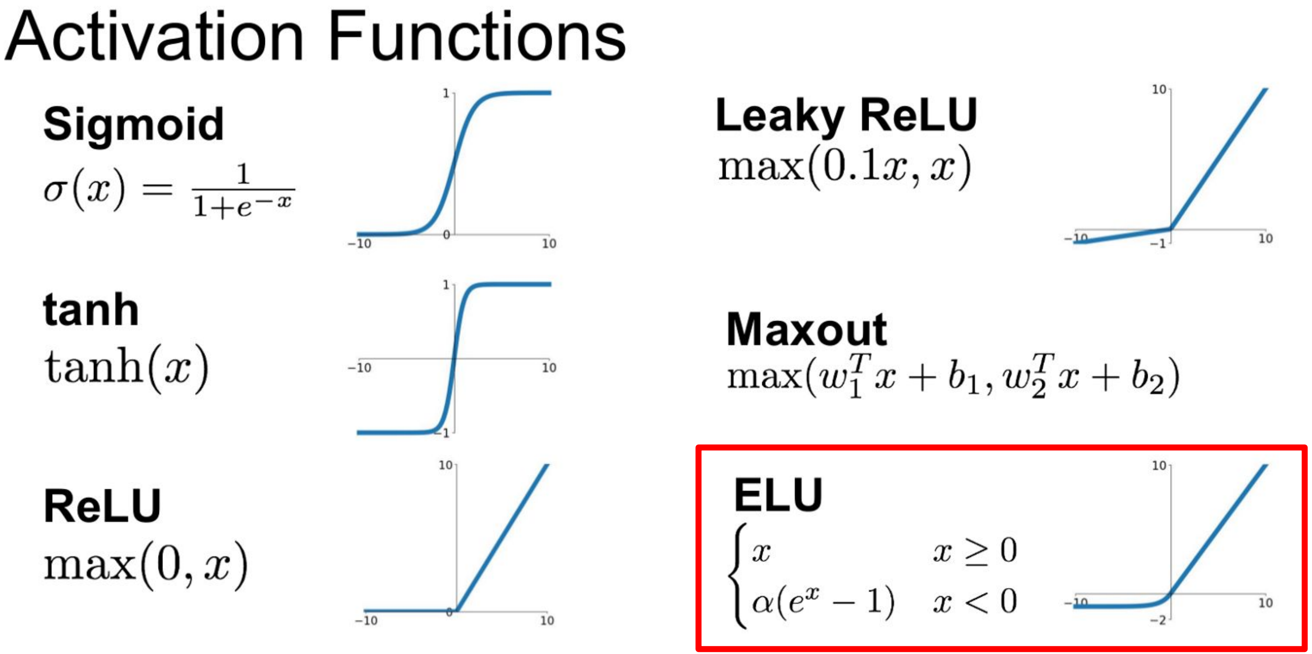


Also known as Logistic regression

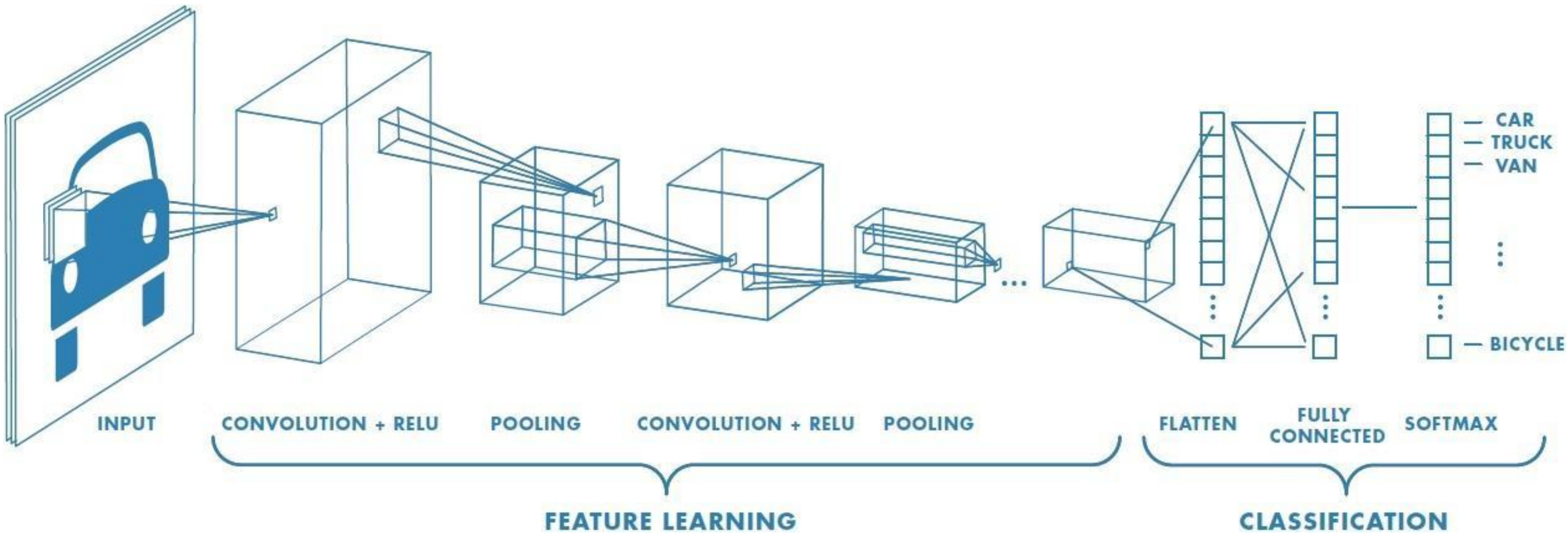
DATA SCIENCE INSTITUTE

Deep Learning is possible because of Neural Networks and Neurons

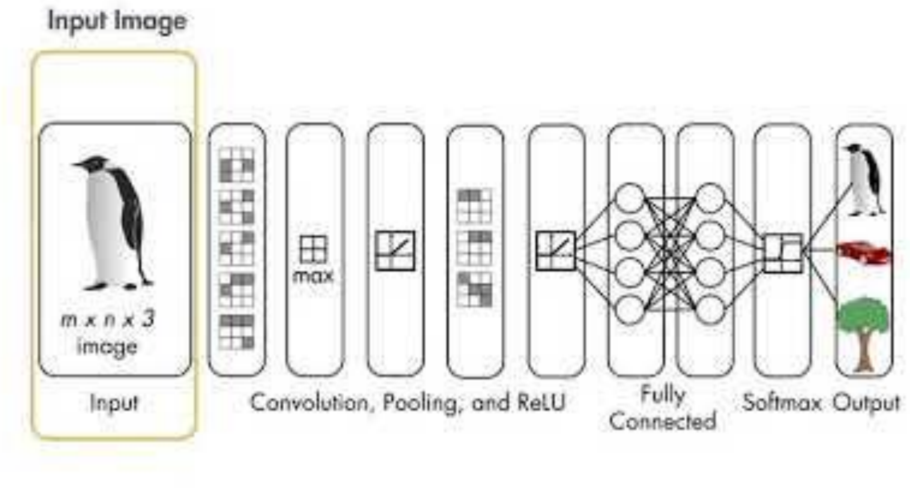
- Types of Activation Functions
  - **ELU (exponential Linear Unit)**
    - Considered to be faster and more accurate than ReLU



- **Deep Learning is possible because of Neural Networks**
  - Previously we would have to do test several machine learning techniques/algorithms
    - Now Deep Learning does this for us



CONVOLUTIONAL NEURAL NETWORK (IMAGE CLASSIFICATION)



Anade Davis - Data Science Manager - Conclusion

- The Best Neural Network for Advanced Image Recognition appears to be A Convolutional Neural Network with a Activation Network of ELU

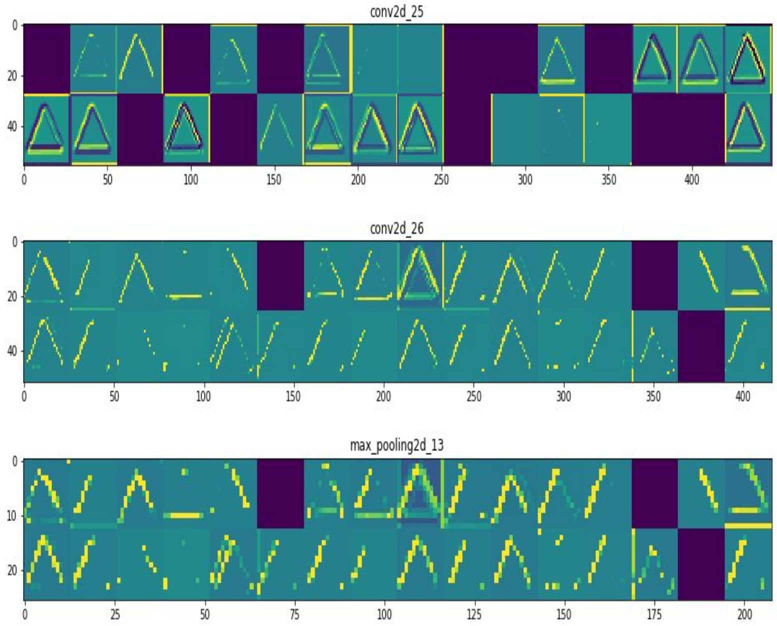
Muhammad Jawwad Javeed Iqbal - Financial Data Scientist- Points of Discussion

- The problem with Feedforward Neural Networks
- Convolutions and Pooling Layers
- Structure of Convolutional Neural Network
- Activation Functions
- InceptionV3/Loading Models
- Evaluation/Results
- Conclusion

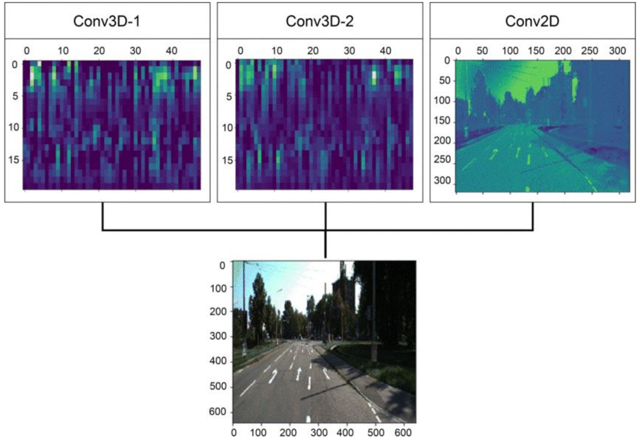
- **The problem with Feedforward Neural Networks**
    - The amount of neurons required for a Feedforward neural network directly scales with the matrix dimensions of the input:
        - A Feedfoward Neural Network will work fine with monochrome/cropped images for object classification/detection, but will quickly fall apart at higher resolution pictures
        - For example, a single 1000x1000 image with (R,G,B) dimensions would require over 3,000,000(1000x1000x3) neurons in the input layer alone!
        - This method would be too computationally intensive for long-term use, so developers use **Convolutions** and **Pooling** layers to simplify the input layer without losing the features of the image.

Convolutions and Pooling layers

- **Convolutions** are able to extract the general features of picture such as pixel density and the location of certain features in an image
- The problem with using Convolutions alone is that they are heavily reliant on the precision of the pictures;if the image changes in anyway,the Convolutional layer will regard it as a completely different picture and classify it as such
- **Pooling** solves this problem through **Down Sampling**; by purposefully decreasing the quality of the image using an filter that averages neighboring pixels,it can summarize important feat.

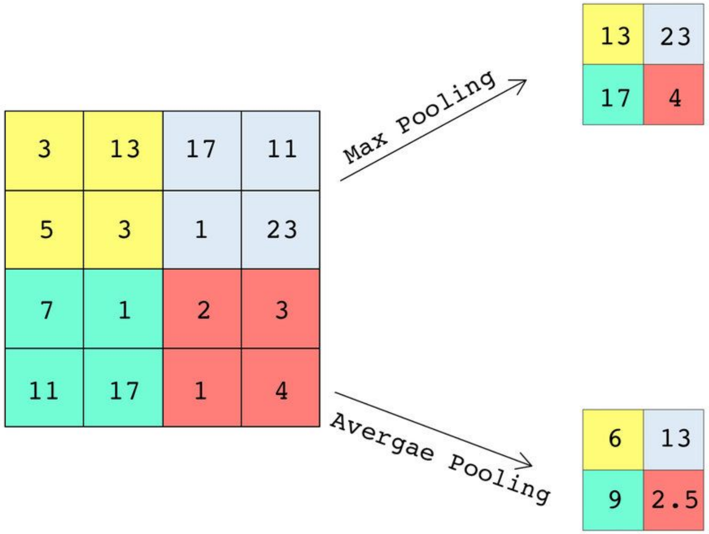


- **Convolutional and Pooling Layers**
  - Convolutions can be applied to 4 matrix dimensions, with each dimension representing:
    - 1 Dimension(**Conv1d**): Used for Time series and Recurrent neural networks(sequence/sentence sentiment and analysis)
    - 2 Dimensions(**Conv2d**):Used for Images
    - 3 Dimensions(**Conv3d**):Used for Videos



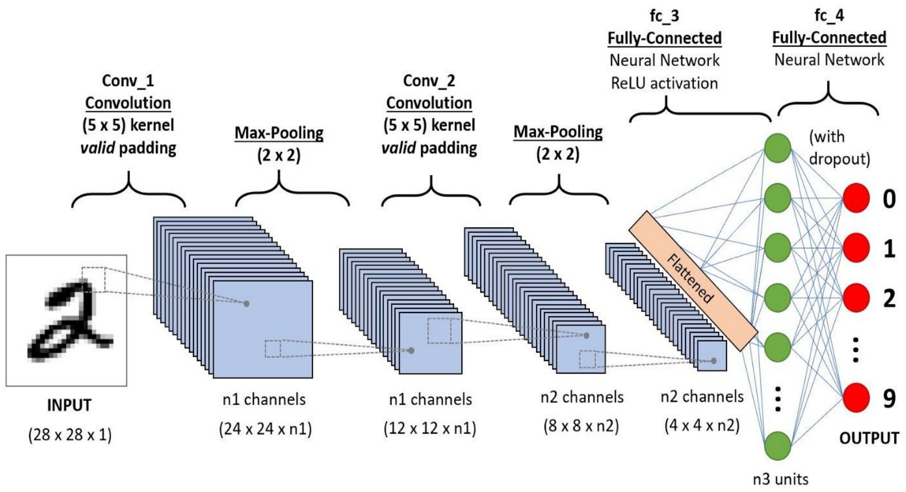
DATA SCIENCE INSTITUTE

- **Convolutional and Pooling Layers**



- Like Convolutions, Pooling functions can also be applied to three dimensions(**Pooling1D**,**Pooling2D**,**Pooling3D**)
- There are two unique Pooling functions that are used depending on the input data :
  - **Max Pooling** takes the maximum value of a stride(cross-section of pixels) in a feature map and is used when detail of the picture is not necessary (i.e. **Object detection**)
  - **Average Pooling** takes the average of a stride in a feature map and is used when detail of the input data is necessary (i.e. **Object recognition**)

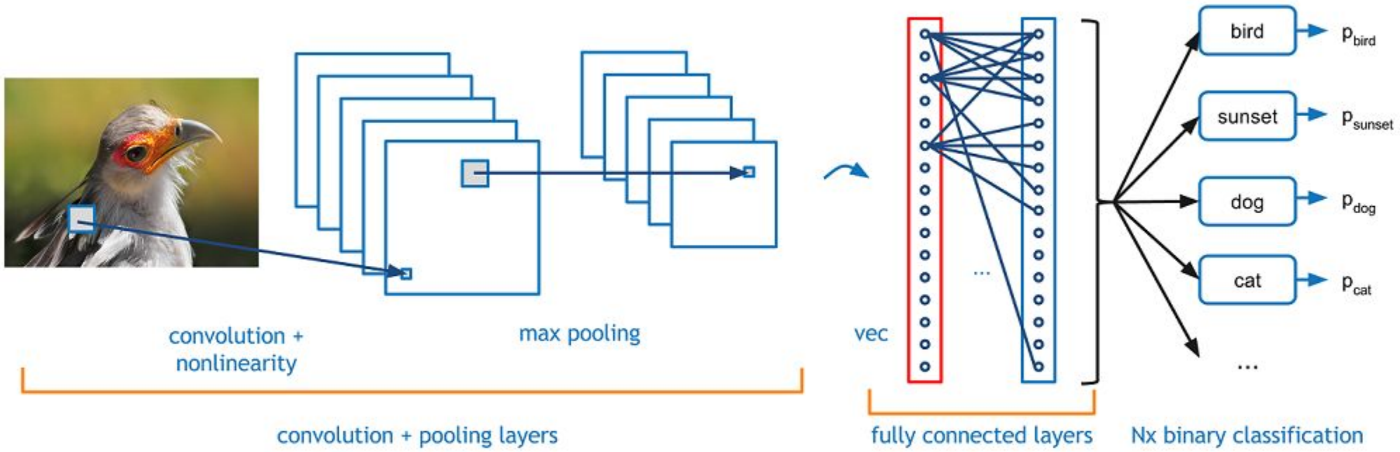
● **Convolutions** and **Pooling** layers



- **Padding** is used in a Convolutional Neural Network to process the input data into having the same amount of matrix dimensions
- These layers are then passed on to a **Flatten()** layer, which prepares the array for the neuron layer by making the array into a single matrix dimension

DATA SCIENCE INSTITUTE

- **Structure of a Convolutional Neural Network using Convolutions and Pooling**
    - Convolutions and Pooling layers are always the first step in creating a Convolutional Neural Network
    - After the data is flattened, the data is then sent through the Hidden **Dense** layer, which is a layer of Neurons that exists before the final classification perceptrons



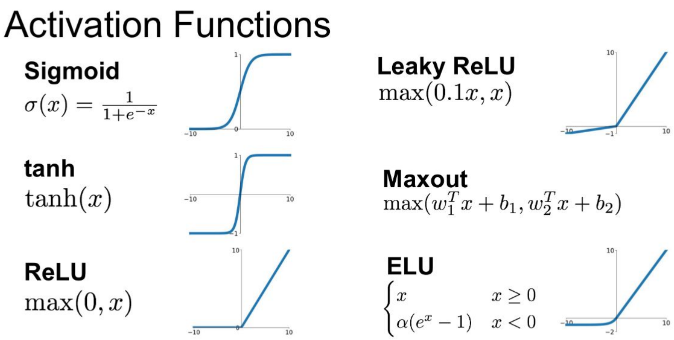
DATA SCIENCE INSTITUTE

- **Structure of a Convolutional Neural Network using Convolutions and Pooling**
    - Model layer structure can be identified using **model.summary()**
    - In this case, the **Max_Pooling_2d()** layers have halved the image resolution quality from (None,26,26,32) to (None,13,13,32)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
flatten (Flatten)	(None, 5408)	0
dense (Dense)	(None, 16)	86544
dense_1 (Dense)	(None, 10)	170

Activation Functions

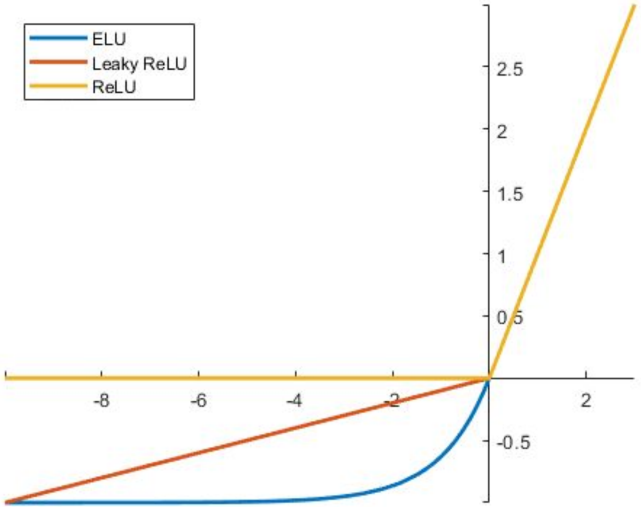
- An Activation Function is a function that receives a set of input data, and changes the output of the data using an assigned mathematical formula
- There are many types of activation functions, but the main ones used in Convolutional Neural Networks are Ridge Activation Functions and Folding Activation Functions:
  - Ridge Activation Functions are functions that have a binary output, and are used mostly in the hidden dense layers of a Neural network
  - Folding Activation Functions are functions that are able to have multiple outputs, and are commonly used in the final layer of object classification problems

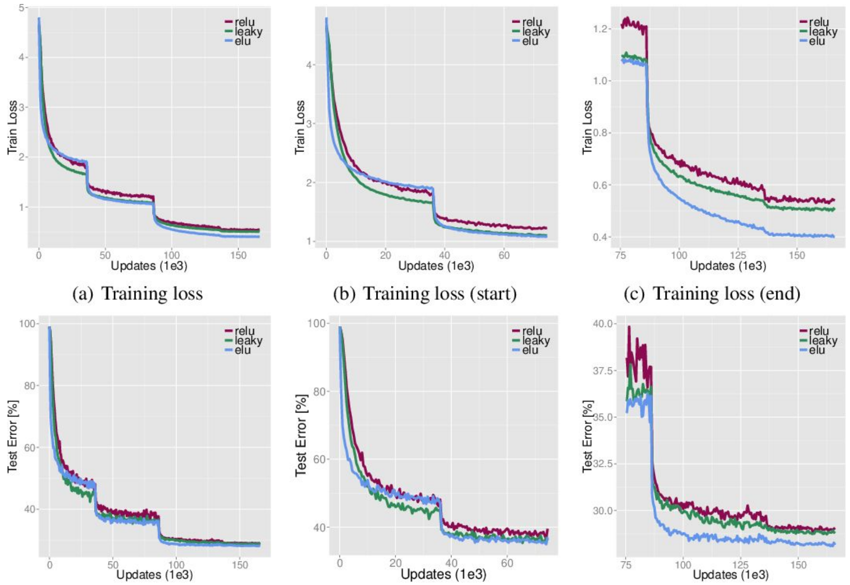


DATA SCIENCE INSTITUTE

- ## Activation Functions (Ridge Activation Functions)
  - There are numerous Linear Ridge Activation functions that can be used in TensorFlow and other machine learning frameworks, including but not limited to:
    - **ReLU**(**Rectified Linear Unit**)**:** Most commonly used Ridge Activation Function; excellent in most applications, but suffers from a problem where some of the neurons in the model shut down and only output zero
    - **LReLU** (**Leaky ReLU**)**:**Improvement over ReLU that has the same properties as ReLU, but is modified to mitigate the halting neuron problem
  - There are also Non-Linear Ridge Activation functions:
    - **ELU**(**Exponential Linear Unit**):Improvement over ReLU that uses logarithmic curves instead of linear lines for the activation function; has the greatest accuracy out of all activation functions, but has the drawback of higher computation time due to the exponential nature of the activation function
    - **Sigmoid**: Usually used as an output function for object classification problems; Like ELU, computationally intensive

Activation Functions

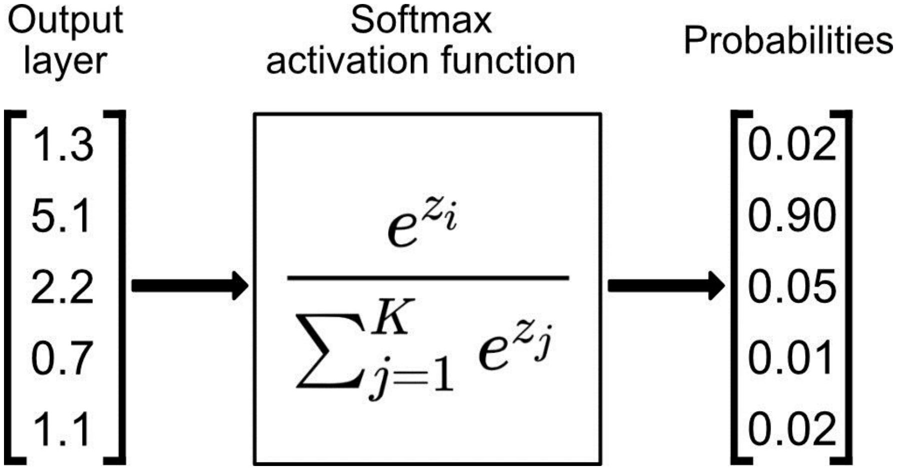




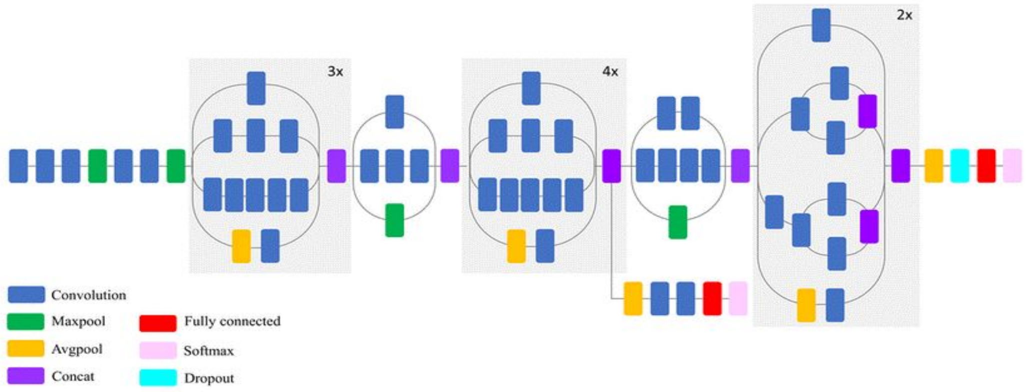
(a) Training loss (b) Training loss (start) (c) Training loss (end)

(d) Test error (e) Test error (start) (f) Test error (end)

- **Activation Functions (Folding Activation Functions)**
  - Folding Activation functions are generally used in multiclass classification output layers, including but not limited to :
    - **Softmax:** Most commonly used output function for multiclass classifications due to its ability to calculate probability distributions over any number of outputs
  - Folding Activation Functions are also used in pooling layers



- **InceptionV3/Loading Models**
  - Tensorflow and most machine learning frameworks provide the ability to save and store the weights of a model, and allow for those models to be loaded in different neural networks
    - **InceptionV3** is one of the most popular convolutional neural networks for object detection; developed by Google's Inception project as a part of the ImageNet recognition challenge,it is a highly advanced Neural network used for detecting a list of 1000 different objects.

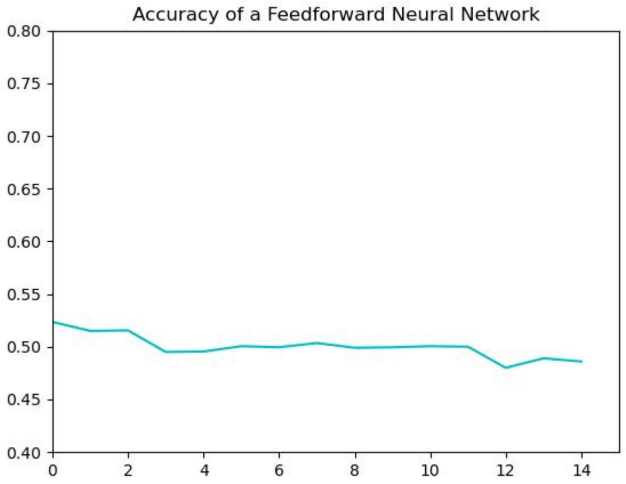


- ## Evaluations/Research
  - To objectively test what Convolutional Neural Network would be the most accurate in binary object detection, We will run a simple image dataset on these models, and record the results using Matplotlib.
    - The 4 methods that we are testing will be **Feedfoward Neural Networks**,**1 Convolution/Pooling layer**,**4 Convolutions/Pooling layers**, and finally (although not designed for this dataset specifically) **InceptionV3**.
    - The Activation Functions used for this test are ReLU and Sigmoid
    - This test was performed using TensorFlow-GPU for 15 epochs (code cycles) with the following hyperparameters:

```
model.compile(loss="binary_crossentropy",optimizer=RMSprop(lr=.001),metrics=["acc"])
```

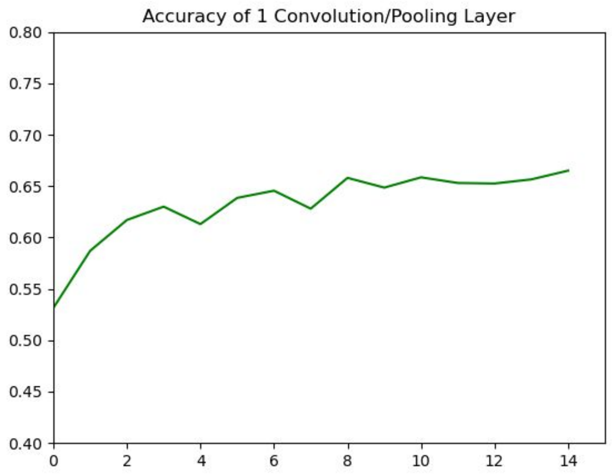
- **Evaluations/Research (FeedFoward Neural Network)**

    - The Feedforward Neural Network took the longest time to complete, and also provided the least accurate results, with approximately 49% accuracy.
        - Because the data was fed directly into a flatten layer, the model was not able to generate any patterns from the data, and as a result the accuracy of the model decreased over time.

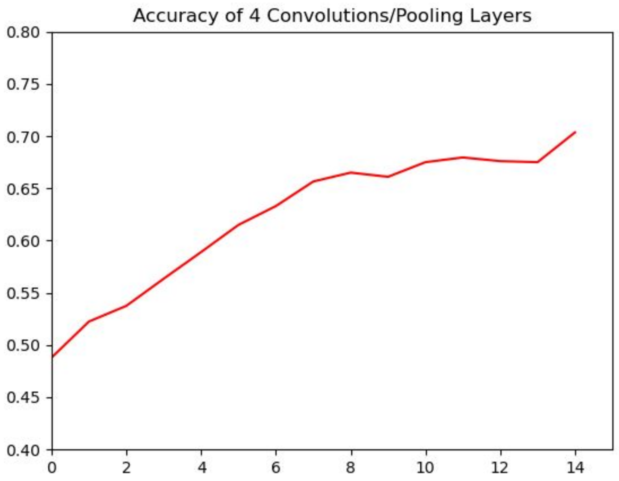


- **Evaluations/Research (1 Convolution+Pooling Layer)**

    - The introduction of a 1 Convolution/Pooling layer increased the accuracy of the model by 18% when compared to the Feedfoward Neural network (from 49% accuracy to 67% accuracy).
        - Because the Convolutions/Pooling layer reduced the amount of data sent to the flatten layer through down sampling, the neural network is able to discern patterns much more efficiently than with a Feedforward Neural network.

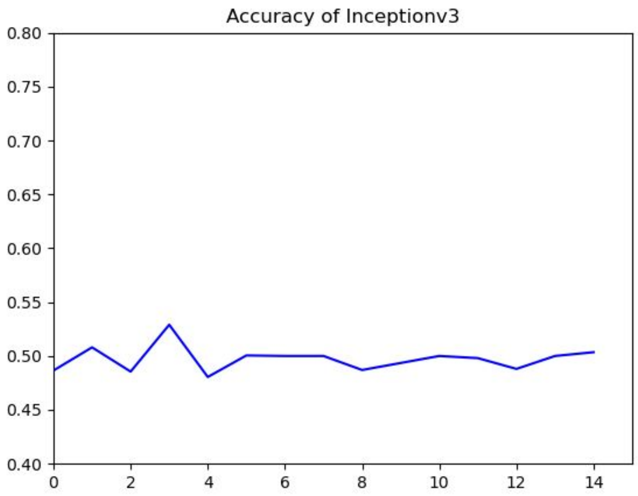


Evaluations/Research (4 Convolution+Pooling Layers)



- Out of all of the Neural Networks researched, the model with 4 Convolutions/Pooling Layers performed the best, at 72% accuracy .
  - It should be noted that there is a significant level of diminishing returns for each Convolution/Pooling layer introduced.
  - From 0 to 1 Convolutions/Pooling layers, there was a 18% increase in the accuracy of the model, but from 1 to 4 Convolutions/Pooling layers there was only a 5% increase.

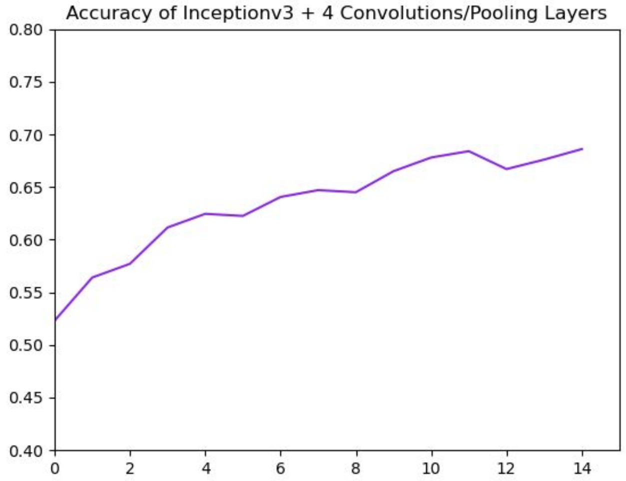
- # Evaluations/Research (InceptionV3)



- Inceptionv3 fared slightly better than the Feedforward Neural network (53% accuracy to 49% accuracy), but still was significantly worse than the Neural Networks that included Convolutional layers .
    - This result highlights the importance of understanding when to load certain weighted models versus when to construct one specifically for a given dataset; because Inceptionv3 was made with the ImageNet outputs in mind, It wasn't able to adjust its weights for a different dataset and therefore performed worse than a 1 Convolution/Pooling layer model.

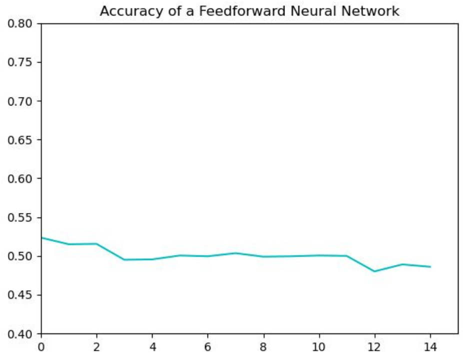
- **Evaluations/Research (InceptionV3+4 Convolutional/Pooling Layers)**

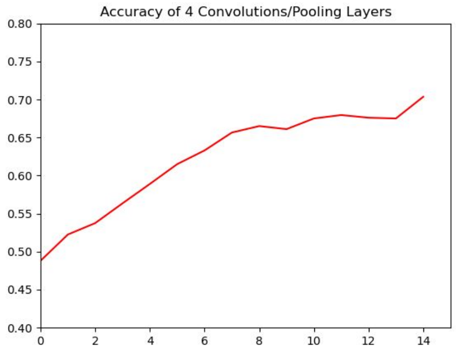
    - With the introduction of 4 Convolutions/Pooling Layers, The InceptionV3 model was able to significantly increase its accuracy from 53% to 68%.
        - Because InceptionV3 was able to adjust its weights to a new dataset, it was able to extract patterns from the dataset and classify the data more accurately.

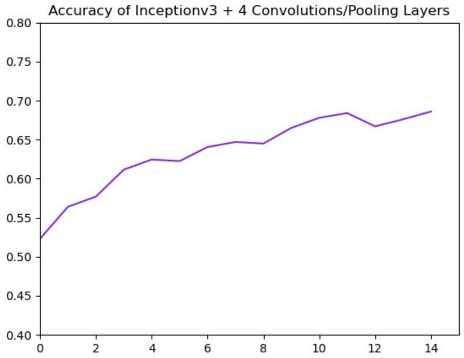


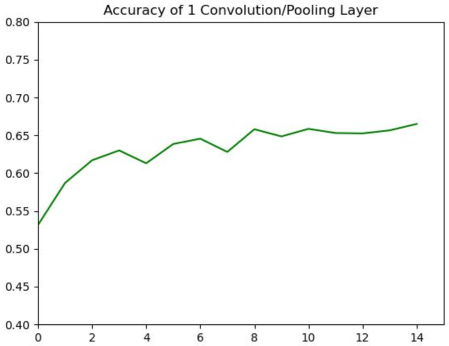
DATA SCIENCE INSTITUTE

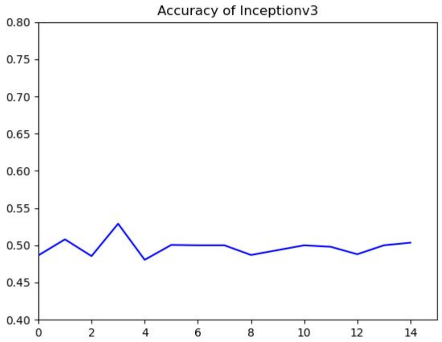
Evaluations/Research (Comparsion)











DATA SCIENCE INSTITUTE

- **Muhammad Jawwad Javeed Iqbal - Financial Data Scientist- Conclusion**
  - The order of accuracy from the different Neural Networks is as follows:

    1. 4 Convolutions/Pooling Layers: 72% Accuracy
    2. InceptionV3 + 4 Convolutions/Pooling Layers: 68% Accuracy
    3. 1 Convolution/Pooling Layer: 67% Accuracy
    4. InceptionV3: 53% Accuracy
    5. Feedfoward Neural Network: 49% Accuracy

  - From the results, a Neural Network with 4 Convolutions/Pooling layers had the highest accuracy and therefore is the best Convolutional Neural Network model to use for image recognition
  - Because the test included the Feedfoward Neural network, It was not able to test an ELU activation function due to the resulting high computational intensity from using both techniques at the same time

Alberto Navarrete - Blockchain Engineer/Project Lead - Points of Discussion

- Fast Image Restoration with Multi-bin Trainable Linear Units
- Face recognition with Bayesian convolutional networks for robust surveillance systems
- An ELU Network with Total Variation for Image Denoising
- ELU-vs-ReLU-for-Image-Recognition
- Face Recognition Based On Deep Learning.
- Conclusion

- **Fast Image Restoration with Multi-bin Trainable Linear Units**

  - Deep Neural Networks (DNNs) have shown delivering excellent performance on image restoration tasks. (Having good training and deep models)
    - The DNNS restoration method is based in:
      - Training the NN to capture mapping functions between degraded images and their corresponding images.
      - Staking standard convolution layers and ReLU functions as well as VDSR, and DnCNN approaches and it’s performance on image Super Resolution and denoising tasks
    - It is possible to increase the restoration process result by increasing the number of layers thanks to the Deep Network Structures Nonlinear modeling capacity.
  - The Fast Image Restoration Networks still have 2 challenges (with solutions)

- **Fast Image Restoration with Multi-bin Trainable Linear Units**

  - The Fast Image Restoration Networks still have 2 challenges (with solutions):
    - Sufficient receptive fields:
      - Solution: Utilize efficient sub-pixel convolutional NN, and denoise images, and conduct computations into lower spatial resolution.
    - Increasing the non-linear modeling capacity networks:
      - Solution: Use a Multi-Bin Trainable Linear Unit (MTLU) as the activation function in the current networks.
  - What MTLU does?
    - Multi-bin Trainable Linear Unit allows joint training group of activation function parameters with other network parameters by parameterizing each on-course activation function.

- **Fast Image Restoration with Multi-bin Trainable Linear Units**

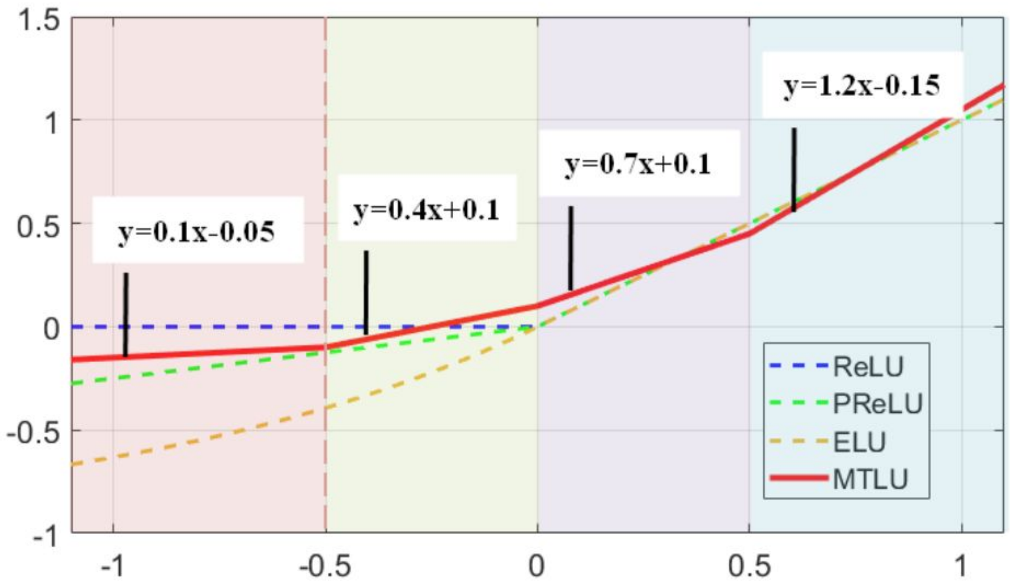
  - What **MTLU** does?

    - Multi-bin Trainable Linear Unit allows joint training group of activation function parameters with other network parameters by parameterizing each on-course activation function.
    - It divides the activation spaces into equidistant bins and with an independent linear function approximates each activation function bin.
    - This process allows to enhance the nonlinear capacity of each activation function without increasing the computational burden.

- **Fast Image Restoration with Multi-bin Trainable Linear Units**

    - The nonlinear capacity of Deep Neural Networks come from the non-linear activation functions.
    - **Some activation Functions of DNN:**
        - **ReLU** Function enables better training of deeper networks.
        - Leaky ReLU **(LReLU)**, Exponential Linear Units **(ELU)**, and Max-Out **(MU)** are designed for improving the performance of DNN.
    - The choice of the AF really affects the network capacity.
    - For improving model fitting ability of network, the Parametric ReLU (PReLU) by learning parameters is proposed for controlling the slopes in the negative part of each channel of features maps.

- **Fast Image Restoration with Multi-bin Trainable Linear Units**



- <u>4-bin MTLU, ReLU, PReLU, and ELU examples.</u>

The MTLU divides the activation space into equidistant bins and makes an approximation of each activation function bin with a linear function.

- **Fast Image Restoration with Multi-bin Trainable Linear Units**

    - In a study conducted using different AF and methods such as: ReLU, PReLu, SR, SRResNet, Max-out Unit, ESPCN, FRSRnet with different AF, VDSR, LapSRN, and several variants using MTLU just for mentioning some of the ones used for comparing and achieving the state-of-art. The study concludes, that using the multi-bin trainable linear unit (MTLU) for increasing the capacity of the neural networks improves at a great scale the gotten results compared to the current activation functions.

    - For the efficient networks proposal (FDnet) it was compared with CNN-based DnCNN and several un-supervised approaches using the same 80000 image crops with a learning rate of 1x10-3 and cropped by 2 every 100k iterations.
    - The FDNet performed 10 times faster than DnCNN used in Image Denoising, declaring the best two networks trading off between speed and performance which were: FSRnet and FDnet given the similar results but much faster and lower memory requirements.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

    - Facial recognition as it is the way of identifying and / or confirming individuals identity using biometrical data, in this case their face.
    - When making a facial recognition system, research there are several difficulties that can be present like the pose, expression, image resolution, and most important the illumination when the face image is captured.
    - The big challenge when developing facial recognition solutions is to deal with low-quality face images, so deep convolutional neural networks are used with softmax for quantifying the model confidence of each classes for each face inputs, but softmax might be an misleading representation of the model confidence.
    - One of the main goals when making facial recognition research is to deal with false positives with uncertainty models using Deep convolutional neural networks and machine learning techniques.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

  - Most of the face recognition research has been over surveillance systems, and crowd control applications (access, law enforcement), and the main Face Recognition process is divided by 4 principal phases known as:
    - Face region detection
    - Alignment
    - Feature extraction
    - Classification
  - Being extraction the fundamental phase.
  - Hand-crafted features have good results only in controlled environments.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

    - Improving the recognition accuracy in real life and not controlled environments has been a challenge, so different approaches have already been made using different classification techniques such as:
        - Support vector machine (SVM)
        - Stochastic modeling
        - Neural networks and ensemble classifiers

    - But no approach has given better results until **Deep-Learning based techniques were used,** using convolutional neural networks, and resulting:
    -
        - Giving several intricate features in large datasets using backpropagation.
        - Excellent results regarding accuracy.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

  - Face recognition techniques can be divided into:
    - Holistic
    - Feature-based
    - Hybrid face matching

  - Brief process description:
    - Holistic: The face is modeled by extracting a set of global features.
    - Feature-based: Robustness to variance in pose and illumination.
      - Note: Extracting facial landmarks with Scale-Invariant Feature Transform features in non-frontal images has a degrading effect in general recognition results.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

  - Brief process description:
    - Holistic: The face is modeled by extracting a set of global features.
    - Feature-based: Robustness to variance in pose and illumination.
      - Note: Extracting facial landmarks with Scale-Invariant Feature Transform features in non-frontal images has a degrading effect in general recognition results.
    - Hybrid face matching
      - Uses holistic features for essential facial points fused component-level and landmark-level approaches with Dual-Cross Pattern feature landmarks that belong to the same analyzed facial component

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

  - Deep Learning algorithms have been proving successful in learning high-dimensional face data dominant representations, but in DL-based classification, predictive probabilities are erroneously interpreted as model confidence.
  - Conventional DL tools for identification and regression are not able to detect uncertainty of the used model, so understanding what does your model cant do and don’t know becomes a fundamental part of machine learning systems.
  - DL methods are capable of achieving better generalization on highly complex tasks, and Deep Convolutional Neural Networks (DCNN) has become a fundamental method for image and face recognition tasks.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**
    - Convolutional Neural Networks architecture is typically composed of these layers:
        - 1.- Convolution layer.
        - 2.- Pooling layer.
        - 3.- Fully connected layer.
        - 4.- Softmax layer.

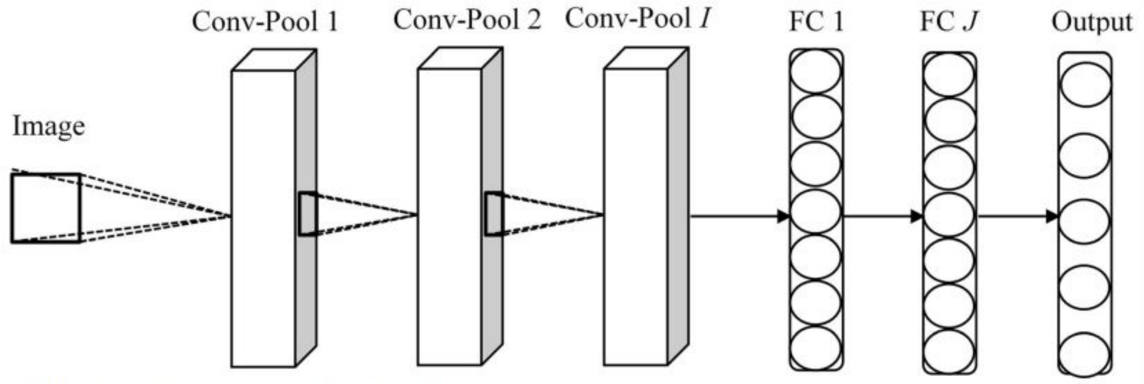


Fig. 1 Deep convolutional neural network (DCNN) architecture for face recognition

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

    - Bayesian CN are a type of DCNNs that have prior probability distributions over a set of model parameters.

      $$\omega = \{W_1, \ldots, W_L\}:$$

    - B-DCNN performs under employing stochastic regularization techniques, and is trained with dropout before every network layer, also testing and sampling from the approximate posterior for performing inference; being the equivalent of performing an approximate variational inference finding a tractable distribution using a training dataset. All by minimizing Kullback-Leibler divergence with the true model posterior.

    - Dropout regularization technique can be interpreted as a variational bayesian approximation type, where the approximated distribution is a blend of two Gaussians with small variances and one of the Gaussians is fixed at zero mean.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

    - In a study, focusing B-DCNN using Bernouli approximate variational inference, was shown a relationship between dropout and variational inference in B-DCNN. This can be used to represent model uncertainties when classifying facial images finding posterior distribution over the convlutional weights of a B-DCNN described in the next Equaton, representing face training data X and respective labels Y in:

$$p(W|X,Y)$$

    - However, the distribution is not tractable and distribution over the weights and variational inference for approximating there weights is needed.
    - This process facilitates optimization of the approximation of the distribution over weights minimizing the Kullback-Leibler divergence, without loosing the primary objective of training the B-DCNN model for facial recognition with dropout to get the posterior distribution of class probabilities.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

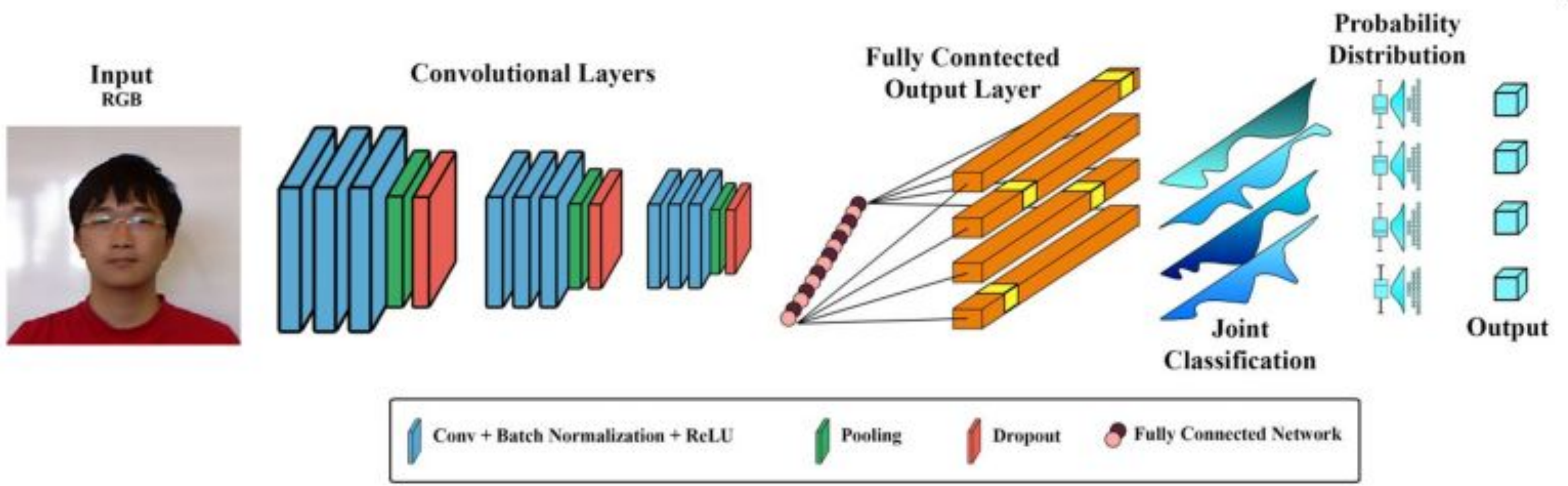


Fig. 2 Schematic illustrations of the proposed face recognition method

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

    - Two real life cases with databases were used for experimentation:
        - AT&T Face Database. With 320 images for training and 80 for testing
        - EURECOM Kinect Face Database (EKFD). With 936 images of 52 different individuals with different poses, expressions, eyes, and illumination.

    - **<u>Results:</u>**
    - Using Bayesian DCNN over DCNN achieved an additional improvement of around 3 - 4% using specifically Arc-2 B-DCNN methodology with 98.1% and 100% accuracy in EKFD and AT&T test cases respectively.

- **Face recognition with Bayesian convolutional networks for robust surveillance systems**

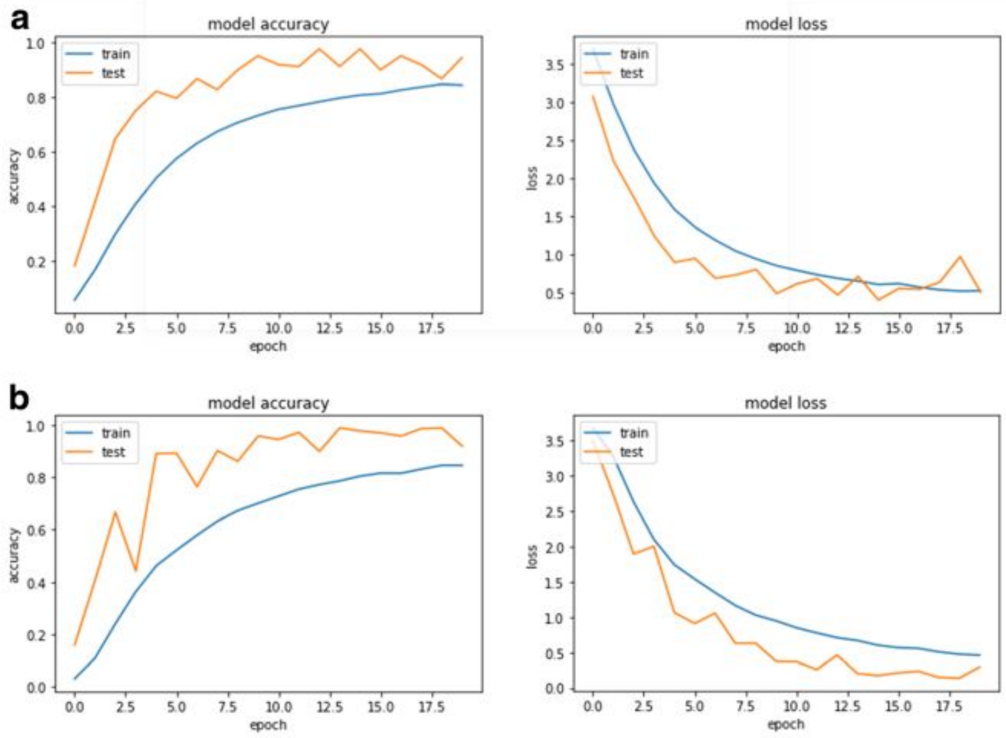


Fig. 4 Model accuracy and loss of proposed DCNN architecture, Arc-2. **a** EKFD [55]. **b** AT&T Database [56]

An ELU Network with Total Variation for Image Denoising

- Image denoising has always been a challenging research topic that aims to restore latent clean imge from a noisy observation.
- Recently it was discovered a method that provides excellent results using deep residual denoising for noisy mapping learning.
- There is a noticeable improvement when using a DCNN with exponential linear unit (ELU) instead of ReLU, cause ELU is more suitable for image denoising when using Trainable NonLinear Reaction Diffusion with residual denoising.
- Batch normalization (BN) is applied to the ELU model for training convergence, however the direct combination of layers directly affects the network performance in a bad way; hence, a new combination of layers with 1 x 1 convolutional layers are proposed for a better integration of BN and ELU layers.
- The proposed model is set as “Conv-ELU-Conv-BN’ as the fundamental block with a second ‘Conv’ in the 1x1 convolutional layer using TV as the regularizer for evaluating the training effect during iterations.

An ELU Network with Total Variation for Image Denoising

- The network architecture used in the proposal, is made up from the model that is derived from the vgg-verydeep19 pre-trained network including 15 convolutional layer blocks and 2 separate convolutional layers with no fully connected layer making the 'Conv-ELU-Conv-BN' pattern.
- This model doesn't have any pooling layer for avoiding up-sampling operation and preserving the same input size as the output.

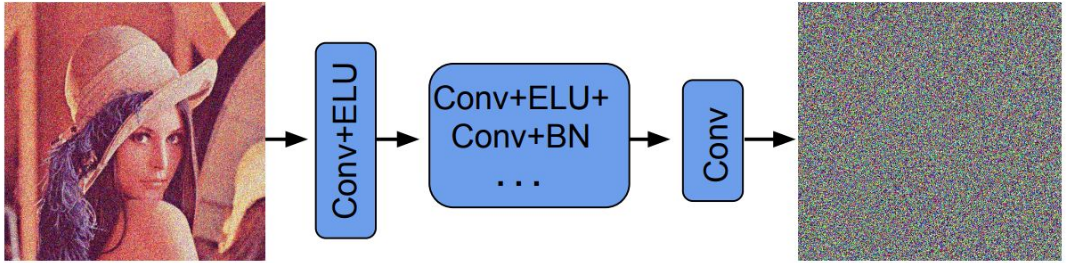


Fig. 1: The network architecture with pipe-lined components.

An ELU Network with Total Variation for Image Denoising

- Experiments were made conducted in Matlab with the MatConvNet framework, processed in a Nvidia Geforce TITAN X GPU.
- 3 experiments training 3 different networks:
  - First network for gray Gaussian denoising with specific noise levels
  - Second network with color image Gaussian denoising with specific noise levels.
  - Third network with color image Gaussian denoising with specific noise levels.
- Datasets:
  - For gray images, BSD68 with 68 images for testing, and 400 images 180 x 180 from Berkley segmentation dataset (BSD500)
  - For color images, color version of BSD68 is employed and testing, and 432 images from BSD500 for training.

- **An ELU Network with Total Variation for Image Denoising**

  - Results for gray images with different methods and the proposed one:

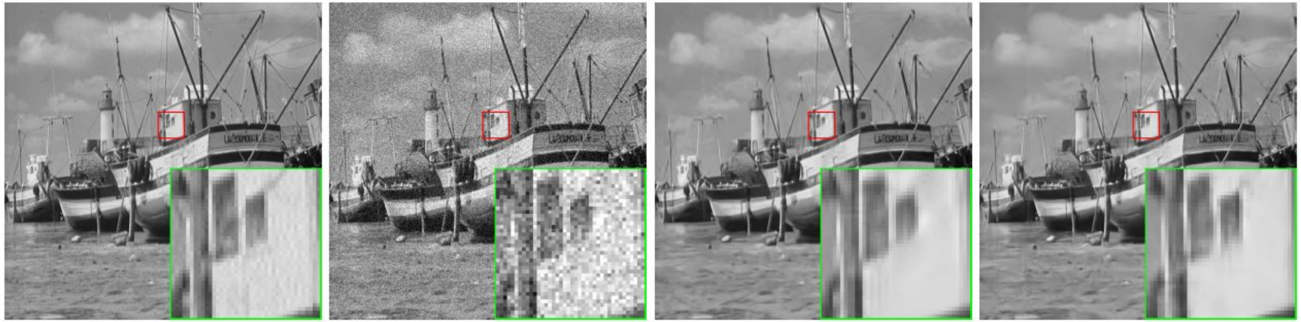
Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN	Ours
$\sigma = 15$	32.37	32.70	32.14	-	32.32	32.50	32.86	**32.96**
$\sigma = 25$	29.97	30.26	29.69	30.03	29.84	30.06	30.44	**30.55**
$\sigma = 50$	26.72	27.05	26.47	26.78	-	26.81	27.21	**27.29**

  - Results for colored images with different methods and the proposed one:

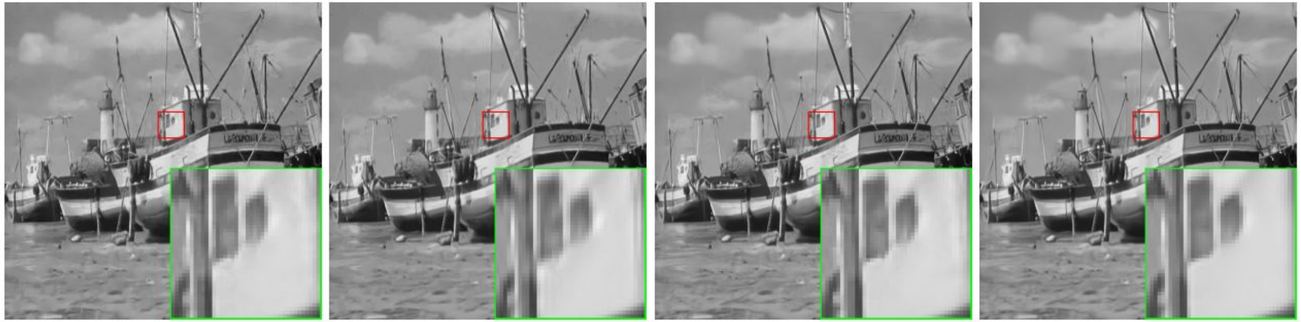
Methods	CBM3D	MLP	TNRD	NLNet	DnCNN	Ours
$\sigma = 15$	33.50	-	31.37	33.69	33.99	**34.10**
$\sigma = 25$	30.69	28.92	28.88	30.96	31.31	**31.41**
$\sigma = 50$	27.37	26.01	25.96	27.64	28.01	**28.11**

An ELU Network with Total Variation for Image Denoising

- Visual results for gray images with different methods and the proposed one:



(a) Clean (b) Noisy/20.18dB (c) BM3D/29.91dB (d) MLP/29.95dB



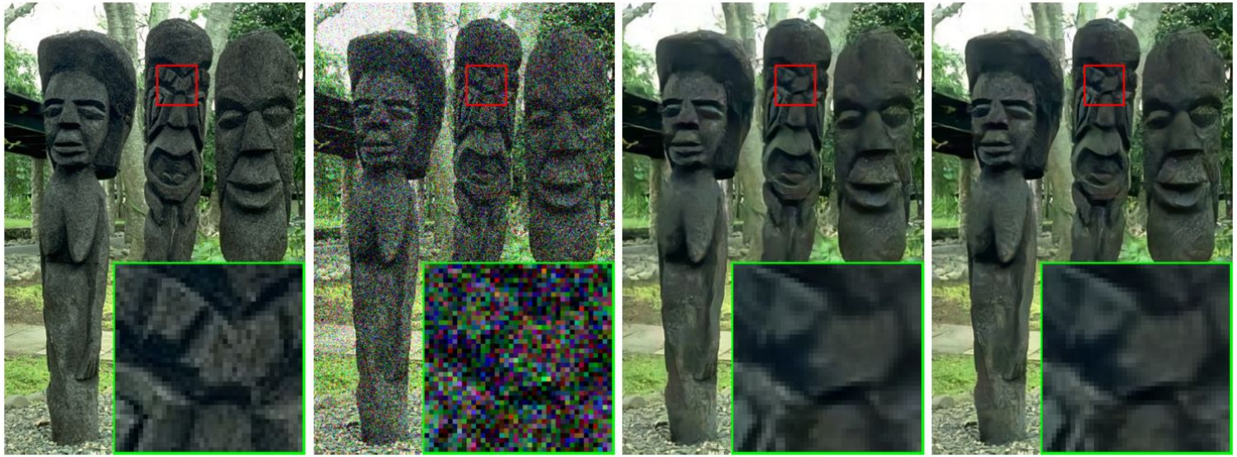
(e) TNRD/29.92dB (f) WNNM/30.03dB (g) DnCNN/30.22dB (h) Ours/30.32dB

- **An ELU Network with Total Variation for Image Denoising**
  - Visual results for colored images with different methods and the proposed one:

(a) Clean (b) Noisy/17.70dB (c) DnCNN/28.18dB (d) Ours/28.28dB

- # An ELU Network with Total Variation for Image Denoising

  - With the gotten results, it can be shown that, a Deep Convolutional Neural Network with exponential linear unit as the activation function and total variation as the regularizer of L2 loss for Gaussian image denoising, a better denoising task is made when using the ELU AF, and even better results when accommodating the ELU and the BN layer properly.



(a) Clean (b) Noisy/15.10dB (c) DnCNN/24.97dB (d) Ours/25.06dB

ELU vs ReLU for Image Recognition

- Chuk Yong GitHub user made a test experiment based in the “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)” from Cornell University utilizing ELU and ReLU activation functions in a Deep Neural Network oriented to Image recognition.
- The two DNN utilized Cifar-10 dataset, and Keras as the main neural networks library.
- Experiment written in Python and 50,000 samples for each network and 20 epochs each
  - ReLU:
    - Returned 75% accuracy
  - ELU:
    - Returned 95% accuracy.

ELU vs ReLU for Image Recognition

- Detailed results:
  - ReLU:
    - Returned 75% accuracy

```
Train on 50000 samples, validate on 10000 samples
```

Epoch 1/20 50000/50000 [==============================] - 59s 1ms/step - loss: 1.8266 - acc: 0.3307 - val_loss: 1.5874 - val_acc: 0.4371 Epoch 2/20 50000/50000 [==============================] - 42s 845us/step - loss: 1.5042 - acc: 0.4534 - val_loss: 1.4038 - val_acc: 0.5006 : : : Epoch 18/20 50000/50000 [==============================] - 41s 825us/step - loss: 0.7802 - acc: 0.7313 - val_loss: 0.7838 - val_acc: 0.7340 Epoch 19/20 50000/50000 [==============================] - 41s 813us/step - loss: 0.7692 - acc: 0.7366 - val_loss: 0.7380 - val_acc: 0.7462 Epoch 20/20 50000/50000 [==============================] - 41s 814us/step - loss: 0.7545 - acc: 0.7409 - val_loss: 0.7362 - val_acc: 0.7489

- # ELU vs ReLU for Image Recognition

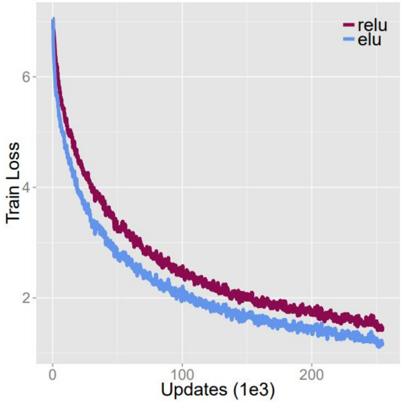
    - Detailed results:
        - ELU:
            - Returned 95% accuracy

```
Train on 50000 samples, validate on 10000 samples
```

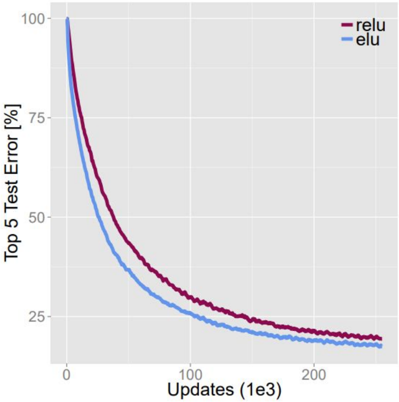
Epoch 1/20 50000/50000 [==============================] - 39s 782us/step - loss: 1.5519 - acc: 0.4505 - val_loss: 1.3890 - val_acc: 0.5069 Epoch 2/20 50000/50000 [==============================] - 39s 789us/step - loss: 1.2326 - acc: 0.5713 - val_loss: 1.1883 - val_acc: 0.5878 Epoch 3/20 50000/50000 [==============================] - 39s 776us/step - loss: 1.0905 - acc: 0.6217 - val_loss: 1.1572 - val_acc: 0.6045 : : : Epoch 18/20 50000/50000 [==============================] - 39s 774us/step - loss: 0.2033 - acc: 0.9349 - val_loss: 1.0350 - val_acc: 0.7375 Epoch 19/20 50000/50000 [==============================] - 39s 775us/step - loss: 0.1642 - acc: 0.9489 - val_loss: 1.1101 - val_acc: 0.7324 Epoch 20/20 50000/50000 [==============================] - 39s 779us/step - loss: 0.1296 - acc: 0.9598 - val_loss: 1.1817 - val_acc: 0.7277

• **ELU vs ReLU for Image Recognition**

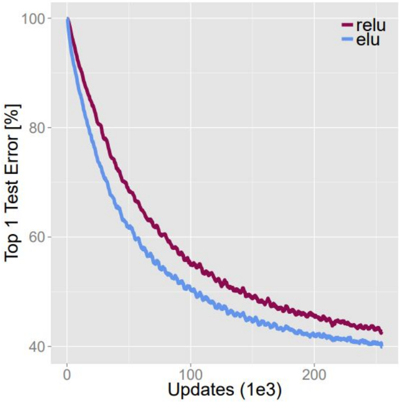
- Analyzing the given results, ELU not only speeds up learning in DNN, but it increases the classification accuracies and the general performance once it has more than 4 layers.
- However, the validation accuracy was better in ReLU than in ELU even though all the general results were favorable to using ELU model.



(a) Training loss



(b) Top-5 test error



(c) Top-1 test error

DATA SCIENCE INSTITUTE

- **Face Recognition Based on Deep Learning**

  - A SIAMESE Convolutional Neural Network made and proposed by Weihong Want, Jie Yang, Jianwei Xiao, Sheng Li, and Dixin Zhou a solution to the current demand of non-contact biometrics systems and the low recognition accuracy problems.

  - The SIAMESE CNN uses a layer-by-layer training method of a DCNN to help the CNN to converge and avoid over-fitting in the model, and solving the face landmark detection problem.

  - This method achieved an 91% accuracy on ORL database and 81% LFW face database.

  - More information in the original investigation paper:

    https://www.researchgate.net/publication/301980999_Face_Recognition_Based_on_Deep_Learning

- **Alberto Navarrete - Blockchain Engineer/Project Lead - Conclusion**
  - When fast restoring images, to use multi-bin trainable linear units is one of the best choices to the nonlinear capacity of DNN. Also using MTLU model, is considered for better results thanks to network parametrization, and enhancing nonlinear capacity of each activation function without increasing the current computational burden.
  - For face recognition using Bayesian DCNN, the improvement in accuracy was around 35 and 4% just by implementing DL methods, and dropout regularization technique being able to reduce the Kullback-Leibler divergence. All this to achieve 98% and 100% accuracy in the test cases.
  - ELU Activation Function is better in several areas than ReLU, like Image Denoising and Image Recognition in terms of learning velocity, accuracy and general results, but not in validation accuracy.
  - In facial recognition, the uncontrolled environments are still a big challenge cause the pose, gestures and primarily illumination can affect in a very big way the Network training and the future given results such as classification and general facial recognition, but these problems can be reduced by using Bayesian Deep Convolutional Neural Networks, and better training images in uncontrolled environments.

Mahmoud Hosny Elhadidy - Financial Data Scientist - Point of Discussions

- Type & Process of Image Recognitions
- Algorithms of DL of Face Recognitions
- Challenges in Image Detection
- Databases
- Object Tracking
- Conclusion

Types & Process of Image Recognition

1) Face Verification:

   One-to-one mapping of a given face (Is this the person?)

2) Face Identification:

   One-to-many mapping for a given face against Db. (Who is this person?)

Types & Process of Image Recognition

Process of Automatic Face Recognition:

1) Face Detection: Locate one or more faces in the image & work with a bounding box.
2) Face Alignment: Normalize the face to be consistent with the database, such as geometry & photometrices.
3) Feature Extraction: Extract features from the face that can be used for the recognition task.
4) Face Recognition: Perform matching of the face against one or more of known faces in a prepared DB.

Types & Process of Image Recognition

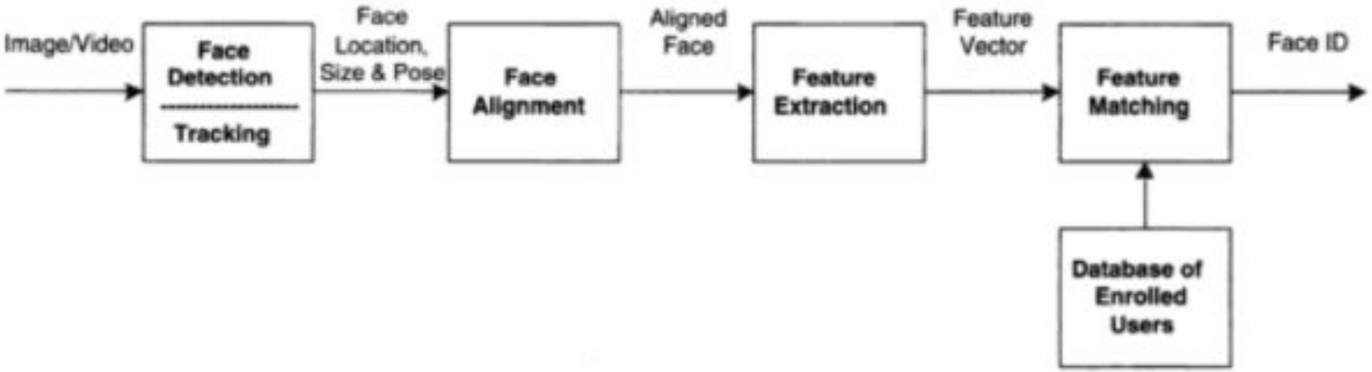


Fig. 1.2. Face recognition processing flow.

Overview of the Steps in a Face Recognition Process. Taken from "Handbook of Face Recognition," 2011.

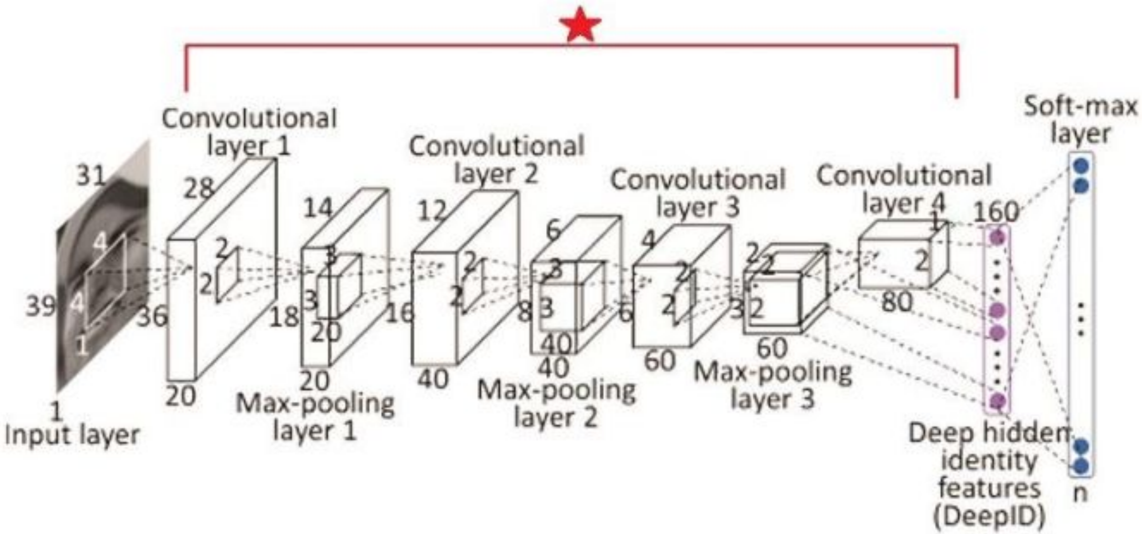
- [Algorithms of DL of Face Recognitions]()

DeepFace: Based on CNN (First One to approach to near human)

With accuracy = 97.25% while Human accuracy = 97.35%.

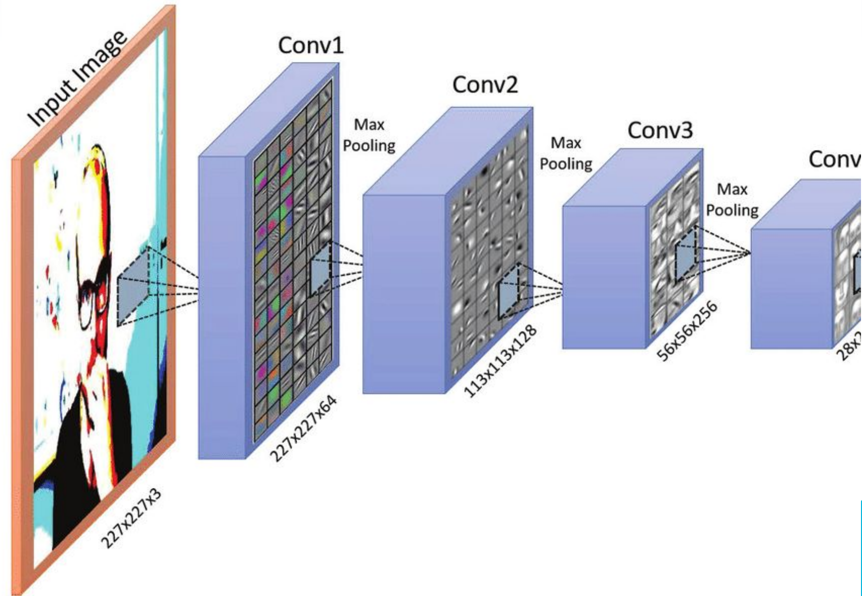
- [Algorithms of DL of Face Recognitions]()

DeepID: Is a series of systems (DeepID, DeepID2,,etc), it's similar to DeepFace but expanding in subset publications, it's accuracy (99.15%) which exceeds human level.



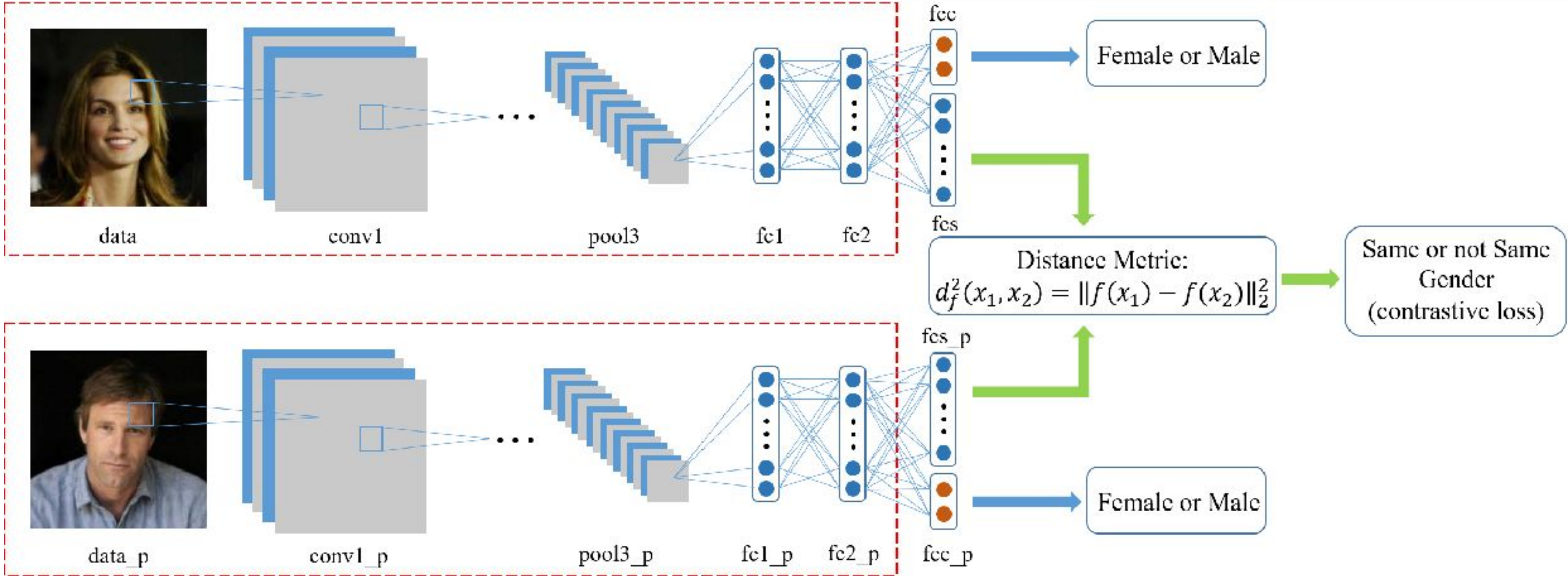
- [Algorithms of DL of Face Recognitions]()

VGGFace: a series of models developed for face recognition and demonstrated on benchmark computer vision datasets by members of the Visual Geometry Group (VGG) at the University of Oxford. And it's accuracy = 98.78%



- [Algorithms of DL of Face Recognitions]()

FaceNet : allow images to be encoded efficiently as feature vectors that allowed rapid similarity and it's accuracy can reach about 100% on some dataset like Yale.

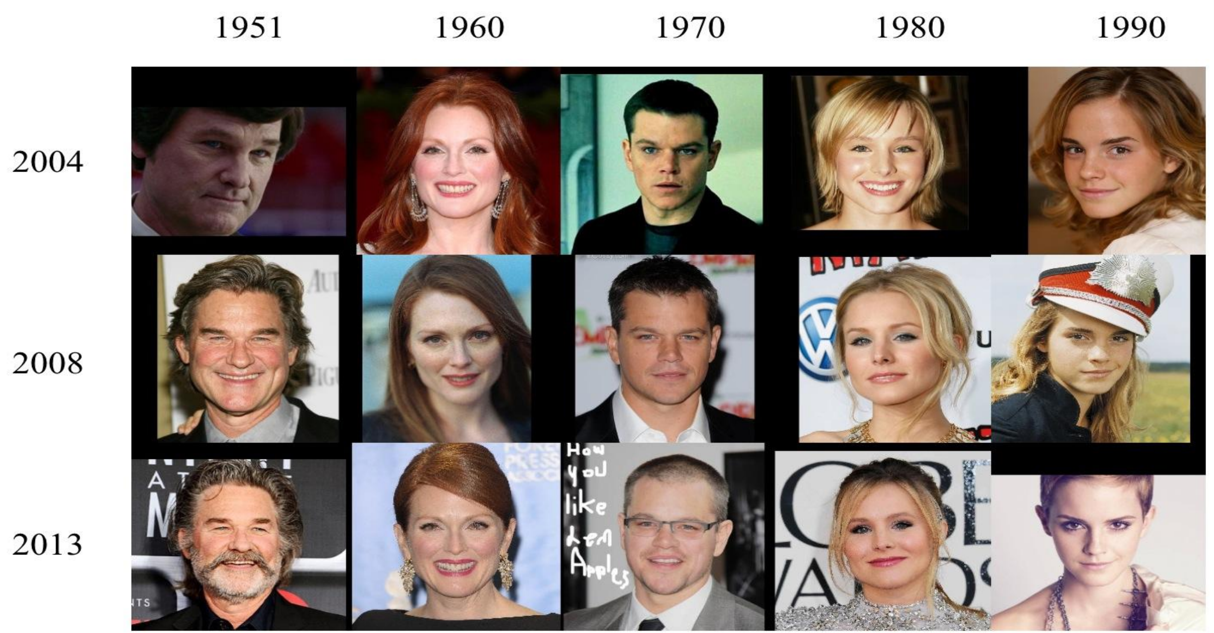


- [Challenges in Image Detection]()

1) Pose Variations: training data maybe shows person face with different orientation than test data.

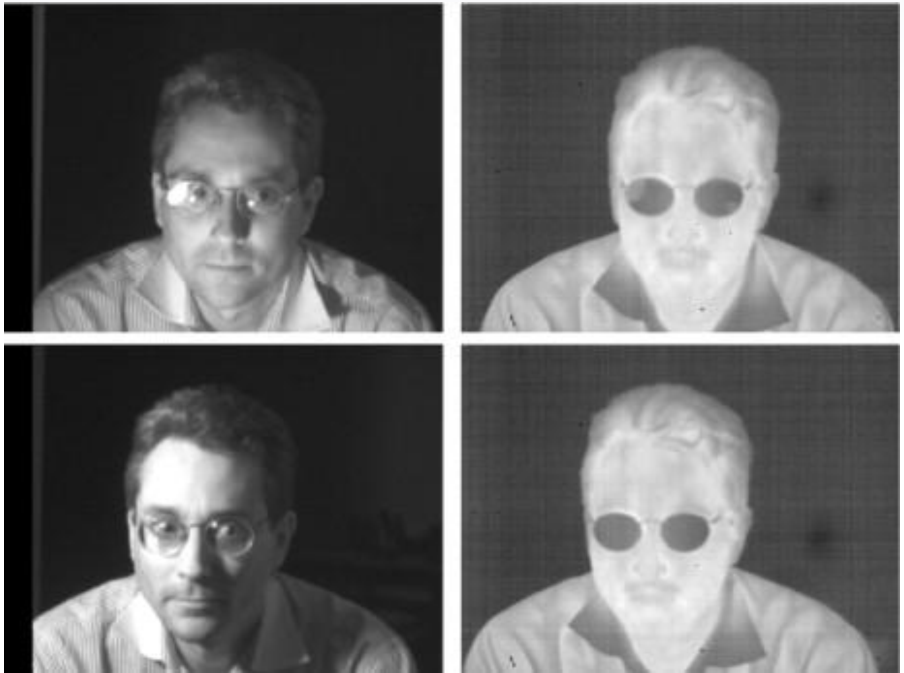
- [Challenges in Image Detection]()

2) Cross-Age: The test data maybe for the same person but in younger/elder age.



- [Challenges in Image Detection]

3) Illumination Changes: training data sets maybe captured in different illuminations.



- [Challenges in Image Detection]

4) Partial Face Images

- [Challenges in Image Detection]()

5) Facial Makeup: People make up some time changes their appearance totally which make it harder to recognize them.

- Challenges in Image Detection

6) Facial Image Expression Variations: People have many emotions (Laugh, Sad,,etc) and that consider a challenge for image recognition.

- Database Sources

1) Still Image Face Databases:

In some sense, the face recognition research is driven by face data. Early face datasets were often collected under pre-defined or controlled environments, such as the CMU PIE (Sim et al., 2002), FERET (Phillips et al., 2000). Along with the practical requirement, more attentions are paid to uncontrolled or un[1]constrained scenarios. i.e., face recognition in the wild. With the advent of Labeled Faces in the Wild (LFW) (Huang et al., 2007), research activity in unconstrained face recognition was accelerated rapidly

- Database Sources

2) Video Face Databases:

Video based face recognition has also gained much attention, and several video face datasets have been released. Table 22 lists serveral datasets with both still and video faces (e.g., COX Face, PaSC, IJB-A, IJB-B, IJB-C). Table 24 shows a list of video face datasets. Most of them are public available. YTF and PaSC are often used to test the recognition performance of various deep models.

- Database Sources

3) Heterogeneous Face Databases:

For heterogeneous face recognition, multi-modal data are needed, e.g., visible, thermal, sketch, RGB-D. Table 26 shows a list of heterogeneous face datasets. These sets are divided into six groups: (1) Still-to-Video faces, such as COX-S2V (Huang et al., 2012c), (2) NIR-VIS faces, (3) Sketch-Photo.

- Conclusion

- There are many DL algorithms for image detection but I would recommend FaceNet due to its accuracy.
- To get database we can use multiple techniques and the most suitable way would be depend on the availability of sources and cooperations with other platforms such as Youtube,,,etc.

- **Amirhossein Abaskohi - Data Scientist - Points of discussion**

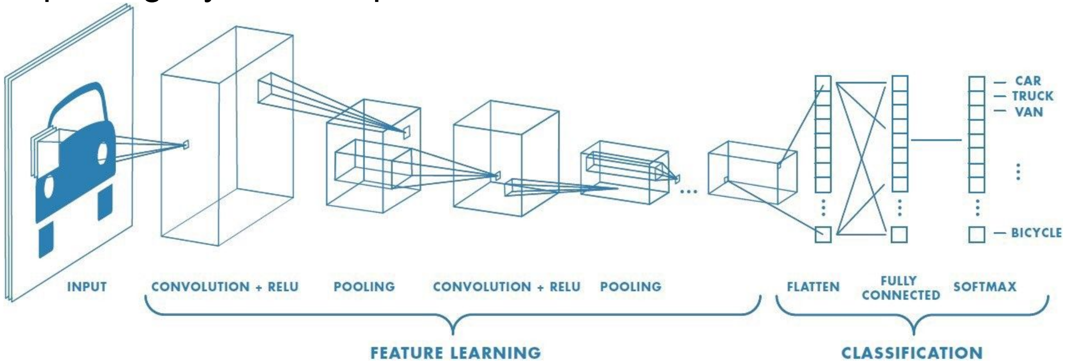
- Image Processing Using Convolutional Neural Networks
- Problem with multiple channels
- Why ReLU for Image Processing?
- Role Of Pooling Layer
- Image Flattening
- Object Detection
- Object Tracking
- Conclusion

Image Processing Using Convolutional Neural Networks

- Image classification is the process of segmenting images into different categories based on their features. A feature could be the edges in an image, the pixel intensity, the change in pixel values, and many more.
- The biggest challenge when working with images is the uncertainty of these features.
- An image consists of the smallest indivisible segments called pixels and every pixel has a strength often known as the pixel intensity.
- Whenever we study a digital image, it usually comes with three color channels, i.e. the Red-Green-Blue (RGB) channels.

● Image Processing Using Convolutional Neural Networks

- CNN works by extracting features from the images. Any CNN consists of the following:
  a. The input layer which is a grayscale image
  b. The Output layer which is a binary or multi-class labels
  c. Hidden layers consisting of convolution layers, ReLU layers, the pooling layers, and a fully connected Neural Network
- ANN made up of multiple neurons is not capable of extracting features from the image.
- The convolution and pooling layers can't perform classification.



- **Problem with multiple channels**

- There are several color spaces like the grayscale, CMYK, HSV in which an image can exist.
- The challenge with images having multiple color channels is that we have huge volumes of data to work with which makes the process computationally intensive.
- The role of CNN is to reduce the images into a form that is easier to process, without losing features critical towards a good prediction.
- This is important when we need to make the algorithm scalable to massive datasets.

Why ReLU in Image Processing?

- ReLU or rectified linear unit is a process of applying an activation function to increase the non-linearity of the network without affecting the receptive fields of convolution layers.
- ReLU allows faster training of the data, whereas Leaky ReLU can be used to handle the problem of vanishing gradient.
- Some of the other activation functions include Leaky ReLU, Randomized Leaky ReLU, Parameterized ReLU Exponential Linear Units (ELU), Scaled Exponential Linear Units Tanh, hardtanh, softtanh, softsign, softmax, and softplus.

Illustrates the output of the Convolution layer followed by a ReLU activation

Role Of Pooling Layer

- The pooling layer applies a non-linear down-sampling on the **convolved feature** often referred to as the **activation maps.**
- This is mainly to reduce the computational complexity required to process the huge volume of data linked to an image.
- Usually, there are two types of pooling: **Max Pooling**, **Average Pooling**
- **Max Pooling** returns the maximum value from the portion of the image covered by the Pooling Kernel
- **Average Pooling** averages the values covered by a Pooling Kernel.

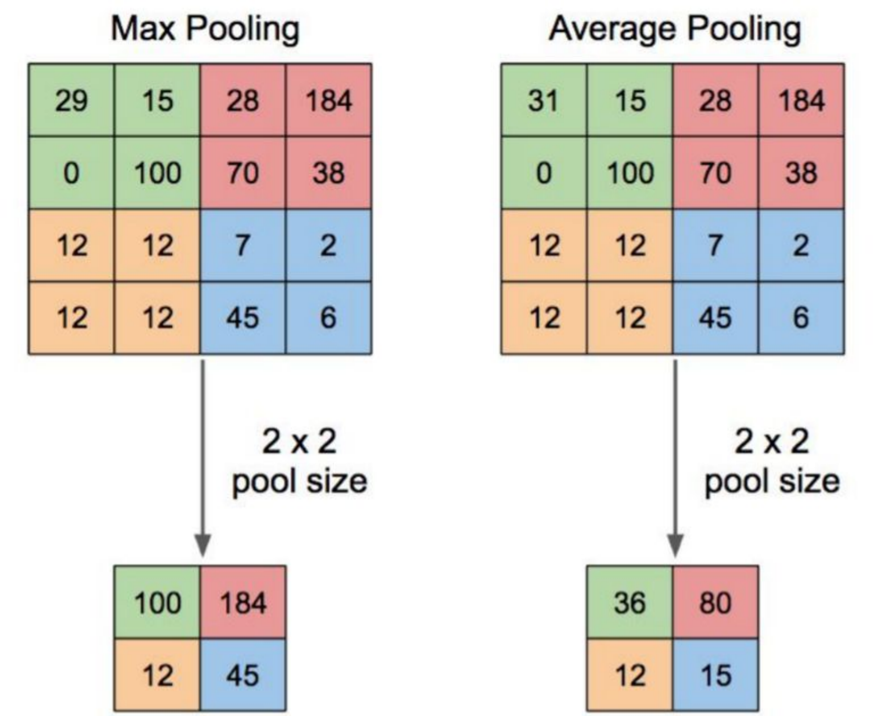
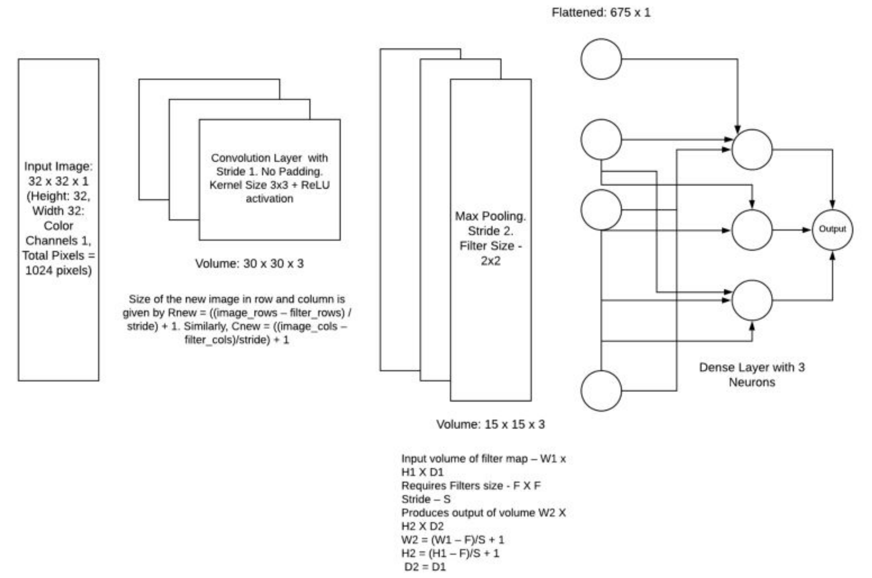


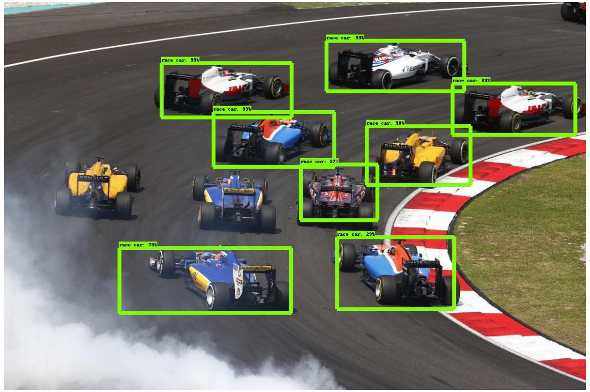
Image Flattening

- Once the pooling is done the output needs to be converted to a tabular structure that can be used by an artificial neural network to perform the classification.
- Note the number of the dense layer as well as the number of neurons can vary depending on the problem statement.
- Also often a drop out layer is added to prevent overfitting of the algorithm.
- Dropouts ignore few of the activation maps while training the data however use all activation maps during the testing phase.
- It prevents overfitting by reducing the correlation between neurons.



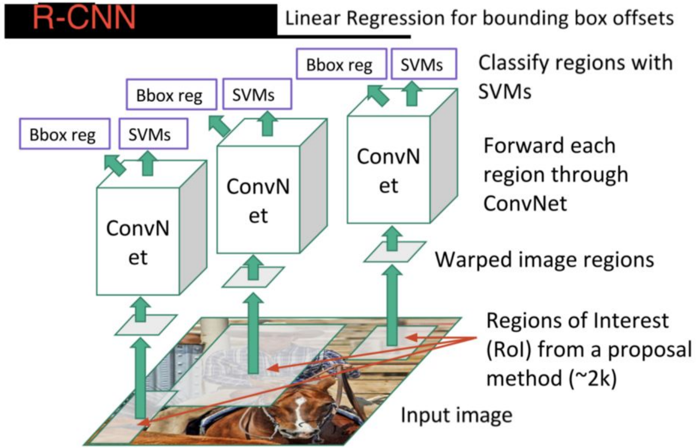
Object Detection

- The task to define objects within images usually involves outputting bounding boxes and labels for individual objects.
- This differs from the classification / localization task by applying classification and localization to many objects instead of just a single dominant object.
- You only have 2 classes of object classification, which means object bounding boxes and non-object bounding boxes.
- RCNN, Fast RCNN and Faster RCNN are different models for this task



Object Detection - RCNN

- In a R-CNN, we first scan the input image for possible objects using an algorithm called Selective Search, generating ~2,000 region proposals.
- Then we run a CNN on top of each of these region proposals.
- Finally, we take the output of each CNN and feed it into an SVM to classify the region and a linear regression to tighten the bounding box of the object.

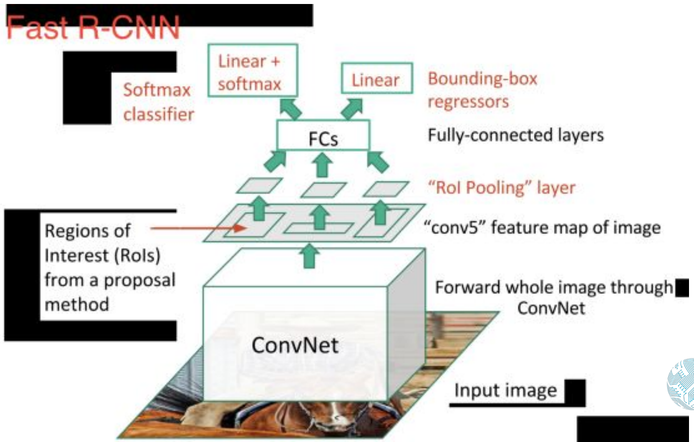


Object Detection - Fast RCNN

- An immediate descendant to R-CNN is **Fast R-CNN**, which improves the detection speed through 2 augmentations:
    a. Performing feature extraction before proposing regions, thus only running one CNN over the entire image
    b. Replacing SVM with a softmax layer, thus extending the neural network for predictions instead of creating a new model.

Object Detection - Fast RCNN

- Faster RCNN performs much better as it uses one CNN layer on the whole image.
- However, the selective search algorithm is still taking a lot of time to generate region proposals.



Object Detection - Faster RCNN

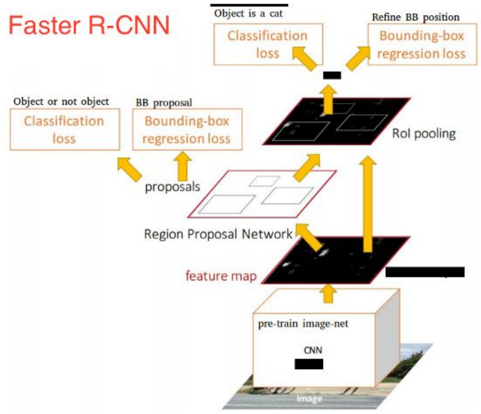
- It replaces the slow selective search algorithm with a fast neural network by inserting a **Region Proposal Network** (RPN) to predict proposals from features.

- The RPN is used to decide “where” to look in order to reduce the computational requirements of the overall inference process.

- The RPN quickly and efficiently scans every location in order to assess whether further processing needs to be carried out in a given region.

Object Detection - Faster RCNN

- Once we have our region proposals, we feed them straight into what is essentially a Fast R-CNN. We add a pooling layer, some fully-connected layers, and finally a softmax classification layer and bounding box regressor.



Object Tracking

- Object Tracking refers to the process of following a specific object of interest, or multiple objects, in a given scene.

- It traditionally has applications in video and real-world interactions where observations are made following an initial object detection.

- Object Tracking methods can be divided into 2 categories according to the observation model: generative method and discriminative method.

Object Tracking

- The discriminative method can be used to distinguish between the object and the background, its performance is more robust, and it gradually becomes the main method in tracking. Examples: **CNN, SAE**
- The most popular deep network for tracking tasks using SAE is **Deep Learning Tracker,** which proposes offline pre-training and online fine-tuning the net.

- 2 representative CNN-based tracking algorithms are **fully-convolutional network tracker (FCNT)** and **multi-domain CNN (MD Net).**

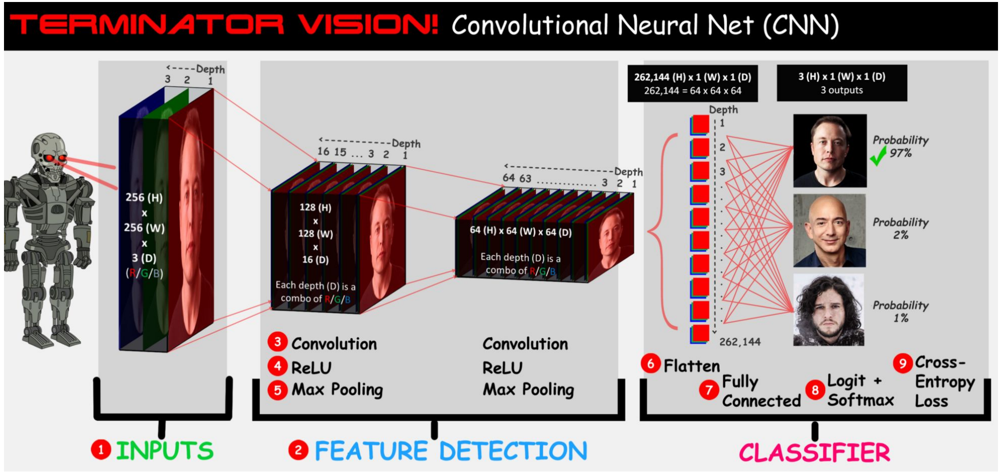
Conclusion

- To sum up, image processing and its variations such as face recognition, object detection, and object tracking are in between the most important usages of deep learning in the real world.
- CNN is the most important type of ANN which can help us to extract features of the image and also helps us to decrease the memory usage.
- RCNN, Fast RCNN and Faster RCNN are the networks which are used in the object detection.

- **Amirhossein Abaskohi - Data Scientist - Points of discussion**
- [Deep learning for speech recognition]
- [Top 8 algorithms in object detection]

Deep learning for face recognition

- The challenge of recognizing and validating persons in an image by their faces is known as face recognition.
- It's a job that humans can easily do, even under different lighting conditions and when their faces have altered with age or are obscured by accessories and facial hair. Nonetheless, until recently, it has remained a difficult computer vision challenge.
- Deep learning algorithms may use very huge datasets of faces to build rich and compact representations of faces, allowing contemporary models to perform as well as, if not better than, human face recognition abilities.



Deep learning for face recognition

- In reality, face recognition has been in development since the 1960s. Artificial Intelligence facial recognition is currently a rapidly evolving technology, thanks to modern digital technological advances and approaches such as artificial intelligence and the Internet of Things.
- Face recognition has also grown in popularity due to a number of causes, including the growth in public security concerns and the demand for more accurate identification verification techniques.
- Deep learning is a subset of machine learning that combines ML techniques and a large amount of data to train deep neural networks for increased accuracy. Deep learning uses an ANN (Artificial Neural Network) to learn and is hence more human-like.

Deep learning for face recognition

- There is often a need to automatically recognize the people in a photograph
- There are many reasons why we might want to automatically recognize a person in a photograph.
- This is sometimes referred to as the challenge of automatic "facial recognition," and it may apply to both still images and video feeds.
- This is a task that humans can readily complete.
- If the individuals in a photograph are known, we may discover their faces and remark on who they are. When individuals have aged, are wearing sunglasses, have different colored hair, are gazing in different directions, and so on, we can accomplish this extremely effectively. We've gotten so good at it that we can detect faces in places where there aren't any, like clouds.

Automatic speech recognition

- Face recognition is sometimes defined as a four-step process that begins with face detection, then moves on to face alignment, feature extraction, and ultimately face identification.
    a. Face detection
    b. Face alignment
    c. Face recognition
    d. Feature extraction

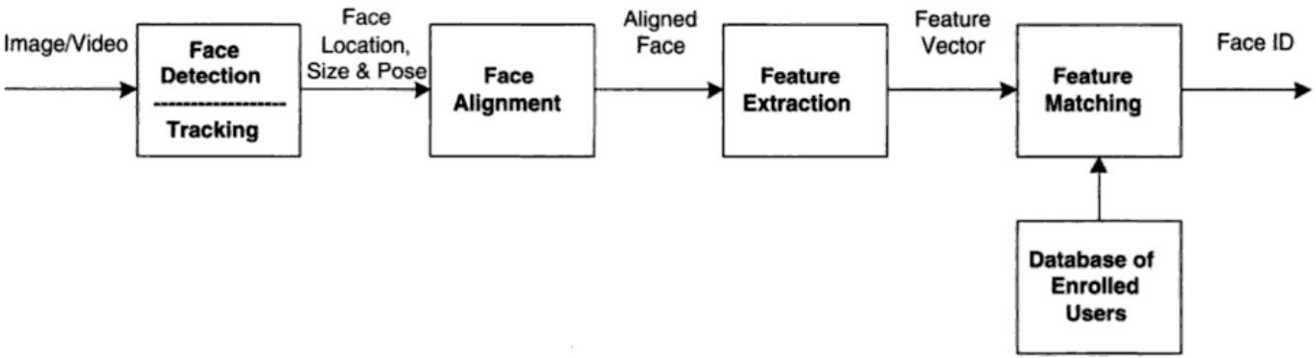


Fig. 1.2. Face recognition processing flow.

Deep learning for face recognition

- Face recognition has remained an active area of research in computer vision.
- The 1991 publication "Face Identification Using Eigenfaces" presented one of the more commonly recognized and utilized "machine learning" approaches for face recognition. Their concept, dubbed "Eigenfaces," was a watershed moment because it produced remarkable results and highlighted the power of basic holistic methods.
- The 2018 paper titled "Deep Face Recognition: A Survey," provides a helpful summary of the state of face recognition research over the last nearly 30 years, highlighting the broad trend from holistic learning methods (such as Eigenfaces), to local handcrafted feature detection, to shallow learning methods, to finally deep learning methods that are currently state of the art.
- In 2014 and 2015, there was a flurry of research and publications on deep learning algorithms for face recognition, after AlexNet's success in 2012 for the easier problem of picture categorization. Within a three-year period, capabilities attained near-human-level performance, then exceeded human-level performance on a typical face recognition dataset, an incredible pace of development considering decades of work.
- Yaniv Taigman, et al. from Facebook AI Research and Tel Aviv developed DeepFace, a system based on deep convolutional neural networks. DeepFace: Closing the Gap to Human-Level Performance in Face Verification was detailed in a 2014 study. On a conventional benchmark dataset, it was probably the first big step forward utilizing deep learning for face recognition, attaining near-human-level performance.
-

Deep learning for face recognition

- The DeepID, or "Deep hidden IDentity features," is a collection of systems (such as DeepID, DeepID2, and others) initially introduced by Yi Sun et al. in their work "Deep Learning Face Representation from Predicting 10,000 Classes" in 2014. Their method was initially introduced in a similar way to DeepFace, but it was later modified to accommodate both identification and verification tasks using contrastive loss training in subsequent papers.
- DeepID2 earned 99.15 percent on the Labeled Faces in the Wild (LFW) dataset, which is better-than-human performance of 97.53 percent, making it one of the first deep learning models to achieve better-than-human performance on the challenge. These findings were improved upon by subsequent systems such as FaceNet and VGGFace.
- In their 2015 article, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” Florian Schroff et al. from Google described FaceNet. Their method produced state-of-the-art results at the time, and they introduced an invention known as "triplet loss," which allowed pictures to be stored effectively as feature vectors, allowing for quick similarity computation and matching through distance calculations.
- Omkar Parkhi, et al. from the Visual Geometry Group (VGG) at Oxford created the VGGFace (for want of a better term), which was reported in their 2015 publication “Deep Face Recognition.” Their study focused on how to acquire a huge training dataset and utilize it to train an extremely deep CNN model for face recognition, which allowed them to obtain state-of-the-art performance on standard datasets.

Deep learning for face recognition

- Face recognition, also known as Biometric Artificial Intelligence, recognizes and validates an individual in a database. With the growth of the database, deep learning gets more and more accurate.
- As a result, after the deep learning algorithm has gained expertise utilizing both old and new huge datasets, any system that uses deep learning may make accurate predictions and offer factual replies to real-time data.
- Main Deep Learning Systems Used for Face Recognition:
  - DeepFace
  - DeepID series of systems
  - VGGFace
  - FaceNet

Deep face, DeepID, and VGGFace

- DeepFace is a deep learning facial recognition system based on deep convolutional neural networks. It was developed by Facebook and is capable of detecting and determining the identity of an individual's face from digital photos with a stated accuracy of 97.35 percent.
- Deep hidden identity for generic object identification was one of the earliest deep learning models for face recognition, initially coined by Yi Sun in his work Deep Learning Face Representation from forecasting 10,000 classes. On one project, DeepID outperformed humans in terms of accuracy.
- The study contributes to a better understanding of the creation of a massive data collection required to train contemporary CNN-based facial recognition systems. The obtained data set is then utilized to train deep CNNs for facial recognition tasks.
- FaceNet employs a triplet loss function to learn score vectors for improved outcomes in feature extraction and, consequently, identity verification, achieving state-of-the-art performance on standard data sets.

Top 8 algorithms in object detection

1. Fast R-CNN
    - Fast Region-Based Convolutional Network Method, or Fast R-CNN, is a training technique for object detection written in Python and C++ (Caffe).
    - This method primarily addresses the shortcomings of R-CNN and SPPnet while also increasing their speed and accuracy.
    - Advantages of Fast R-CNN:
        - No disk storage is required for feature caching
        - Training can update all network layers
        - Training is single-stage, using a multi-task loss
        - Higher detection quality (mAP) than R-CNN, SPPnet

Top 8 algorithms in object detection

- Fast R-CNN
  - Fast Region-Based Convolutional Network Method, or Fast R-CNN, is a training technique for object detection written in Python and C++ (Caffe).
  - This method primarily addresses the shortcomings of R-CNN and SPPnet while also increasing their speed and accuracy.
  - Advantages of Fast R-CNN:
    - No disk storage is required for feature caching
    - Training can update all network layers
    - Training is single-stage, using a multi-task loss
    - Higher detection quality (mAP) than R-CNN, SPPnet

Top 8 algorithms in object detection

- Faster R-CNN
  - Faster R-CNN is a comparable object detection technique to R-CNN.
  - This technique uses the Region Proposal Network (RPN), which is more cost-effective than R-CNN and Fast R-CNN at sharing full-image convolutional features with the detection network.
  - A Region Proposal Network is a fully convolutional network that predicts object boundaries and objectness scores at each point of the object and is trained end-to-end to create high-quality region proposals, which are then employed by Fast R-CNN for object identification.

Top 8 algorithms in object detection

- Histogram of Oriented Gradients (HOG)
    - The histogram of oriented gradients (HOG) is a feature descriptor used in image processing and other computer vision techniques to recognize objects.
    - The Histogram of Oriented Gradients descriptor approach captures gradient orientation in certain areas of an image, such as the detection window and the region of interest (ROI).
    - One advantage of HOG-like characteristics is their simplicity, which makes the information they convey easier to comprehend.

Top 8 algorithms in object detection

- Region-based Convolutional Neural Networks (R-CNN)
    - The Region-based Convolutional Network technique (RCNN) combines region suggestions with Convolution Neural Networks to form the Region-based Convolutional Network method (CNNs).
    - With only a little amount of annotated detection data, R-CNN can help localize objects using a deep network and train a high-capacity model.
    - It achieves high object detection accuracy by classifying object suggestions with a deep ConvNet.
    - R-CNN can grow to thousands of object classes without having to rely on approximation approaches like hashing.

Top 8 algorithms in object detection

- Region-based Fully Convolutional Network (R-FCN)
  - R-FCN (Region-based Fully Convolutional Networks) is a region-based object detection detector.
  - Unlike previous region-based detectors like Fast R-CNN or Faster R-CNN, which use an expensive per-region subnetwork, this region-based detector is completely convolutional, with nearly all processing shared across the whole picture.
  - R-FCN is made up of shared, fully convolutional architectures, similar to FCN, which is known to produce superior results than Faster R-CNN. All learnable weight layers in this method are convolutional and meant to categorize ROIs into object categories and backgrounds.

Top 8 algorithms in object detection

- Single Short Detector
  - The Single Shot Detector (SSD) is a deep neural network-based technique for identifying objects in pictures.
  - Over a range of aspect ratios, the SSD method discretizes the output space of bounding boxes into a collection of default boxes.
  - The technique scales per feature map position after discretization.
  - To naturally handle objects of varying sizes, the Single Shot Detector network integrates predictions from several feature maps with different resolutions.

Top 8 algorithms in object detection

- Single Short Detector (SSD)
  - The Single Shot Detector (SSD) is a deep neural network-based technique for identifying objects in pictures.
  - Over a range of aspect ratios, the SSD method discretizes the output space of bounding boxes into a collection of default boxes.
  - The technique scales per feature map position after discretization.
  - To naturally handle objects of varying sizes, the Single Shot Detector network integrates predictions from several feature maps with different resolutions.
  - Advantages of SSD:
    - SSD has competitive accuracy to methods that utilise an additional object proposal step, and it is much faster while providing a unified framework for both training and inference.
    - Easy to train and straightforward to integrate into systems that require a detection component.
    - SSD completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network.

Top 8 algorithms in object detection

- Spatial Pyramid Pooling (SPP-net)
    - SPP-net (Spatial Pyramid Pooling) is a network topology that generates a fixed-length representation independent of picture size or scale.
    - Pyramid pooling is believed to be resistant to deformations of objects, and SPP-net outperforms all CNN-based image classification techniques.
    - Researchers may use SPP-net to build fixed-length representations for training detectors by computing feature maps from the full image once and then pooling features in random areas (sub-images).
    - This technique eliminates having to compute the convolutional features several times.

Top 8 algorithms in object detection

- YOLO (You Only Look Once)
    - You Only Look Once, or YOLO, is a prominent object detection method utilized by academics all around the world.
    - The unified architecture of YOLO, according to experts at Facebook AI Research, is incredibly fast.
    - The standard YOLO model processes images at 45 frames per second in real time, whereas Fast YOLO processes images at an incredible 155 frames per second while still attaining double the mAP of existing real-time detectors.
    - When generalizing from natural pictures to other domains such as artwork, this technique outperforms existing detection approaches such as DPM and R-CNN.

Conclusion

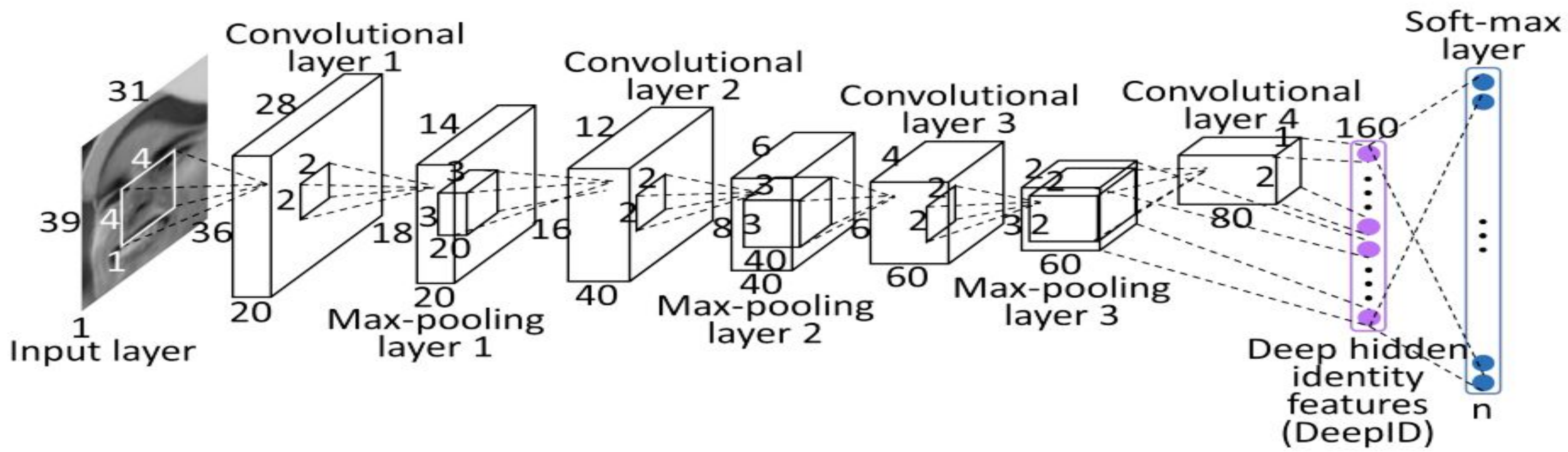
- Face recognition is one of the most important applications of Deep Learning and Computer Vision and there are different models for that.
- The Association of Data Scientists (ADaSci), a prominent global professional association of data science and machine learning experts announces the commencement of its entirely virtual, online conference Computer Vision DEVCON or CVDC 2020 for computer vision enthusiasts throughout the globe.
- It's a two-day conference aimed at bringing together computer vision practitioners and innovators on a single platform to exchange and discuss current advances in the area.
- On the 13th and 14th of August 2020, the conference will be held virtually.

- **Moxú - Financial Data Scientist - Point of Discussions**
    - [Animal Face Classification using Dual Deep Convolutional Neural Network.]
    - [davidsandberg/facenet: Face recognition using Tensorflow]
    - [deepinsight/insightface: State-of-the-art 2D and 3D Face Analysis Project]
    - [Deep Learning Face Representation from Predicting 10,000 Classes]
    - [Conclusion]

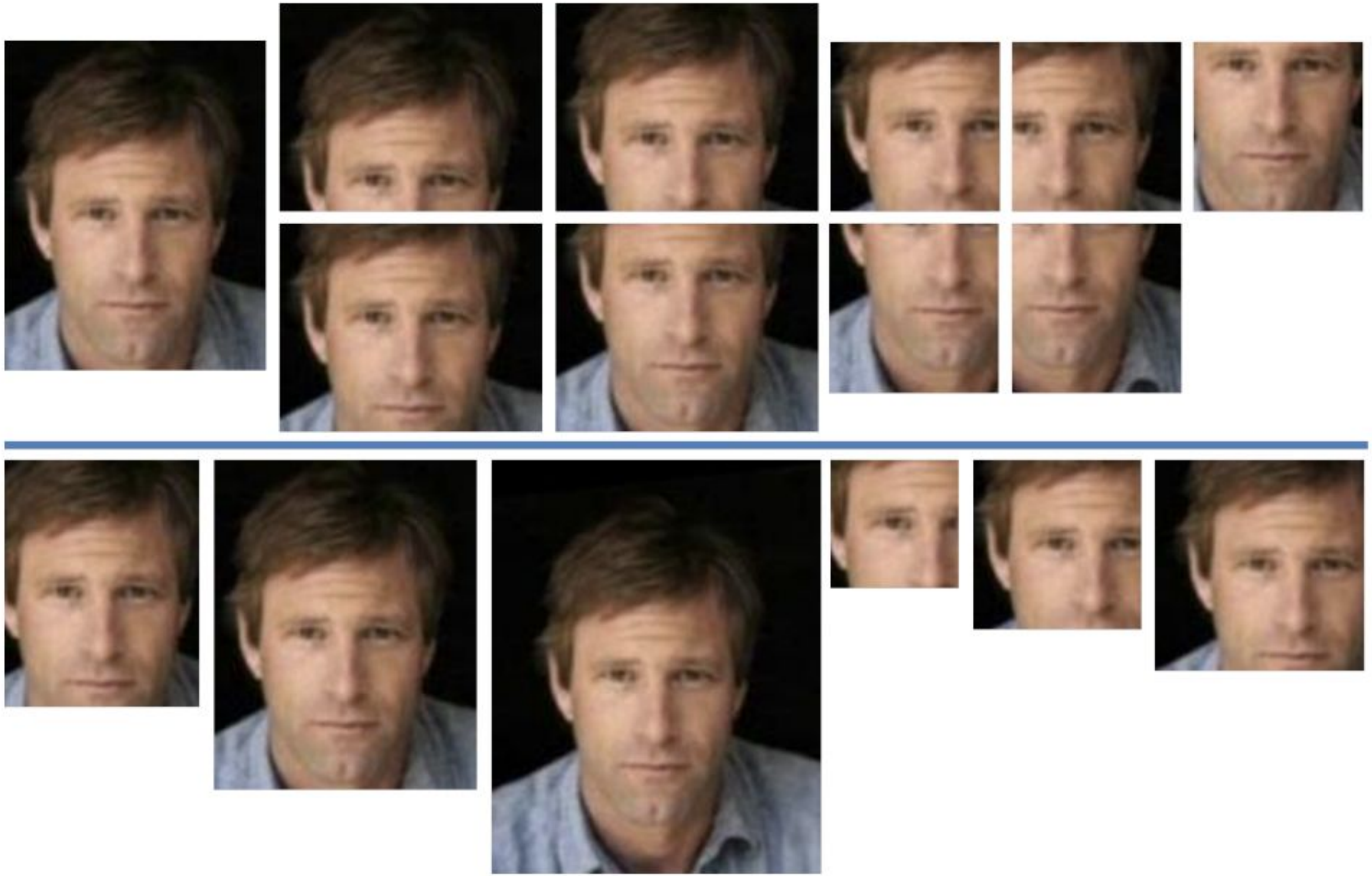
Deep Learning Face Representation

Deep Hidden IDentity features (DeepID) can be useful in many challenges task for identification, but can also be used as face verification tool. In the article was concluded that to increase the generalization of DeepID models, as more face classes are to be predicted at training.

- ConvNets contain four convulational layers to extract features hierarchically, followed by the fully-connected DeepID layer and the softmax output layer indicating identity classes. This structure is flexible, according to size and color of the input image.



- The dimension of the DeepID layer is fixed to 160.
- Was detected five facil landmarks: two eye centers, the nose tip, and the two mouth corners.
- Features are extracted from 60 face patches with ten regions, three scales, and RGB or gray channels.



- In the image above, at the top position, we can see the ten face regions extracted.
- At the bottom, two particular face regions with three different scales.

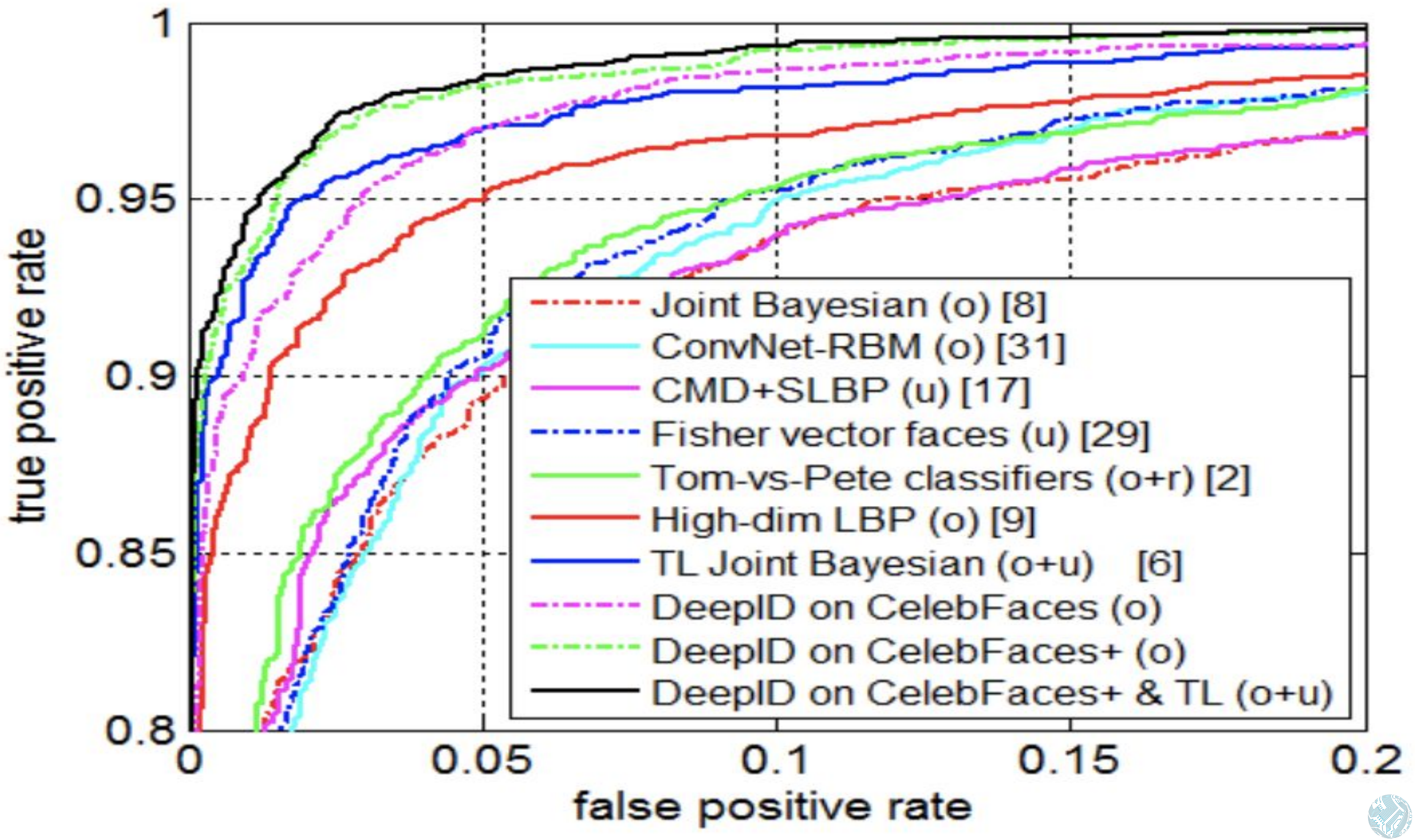
- It was used the Joint Bayesian technique for face verification.
- It has been highly successful for this task, that's why it was chosen.
- It represents the extracted facial features x by the sum of two independent Gaussian Variables.

$$x = \mu + \epsilon \, ,$$

$$x = faceIdentity + intraPersonalVariations$$

Experiments

- It was used CelebFaces, beacuse LFW contain just few images per person.
- The data was split randomly into training and test
- The test was comparing DeepID performance with Joint Bayesian.



true positive rate
false positive rate
1
0.95
0.9
0.85
0.8
0
0.05
0.1
0.15
0.2
Joint Bayesian (o) [8]
ConvNet-RBM (o) [31]
CMD+SLBP (u) [17]
Fisher vector faces (u) [29]
Tom-vs-Pete classifiers (o+r) [2]
High-dim LBP (o) [9]
TL Joint Bayesian (o+u) [6]
DeepID on CelebFaces (o)
DeepID on CelebFaces+ (o)
DeepID on CelebFaces+ & TL (o+u)

Animal Face Classification using Dual Deep Convolutional Neural Network

A practical animal face classification system that classifies animals in image and video data is considered as a pivotal topic in machine learning.

The results demonstrate that, with an accuracy rate of 92.0%, the proposed DCNN outruns its counterparts while causing less computing costs.

Method

The proposed dual DCNN can be divided into two parts:

- Feature extraction layers.
- Fully connected layers.

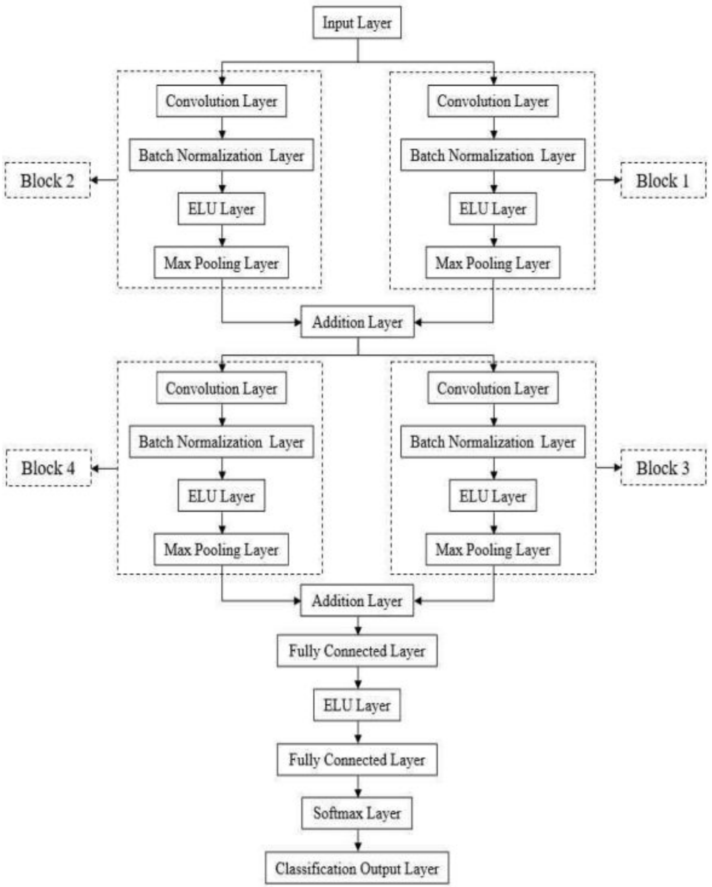


Fig. 4. Our proposed DCNN structure.

Block 1 to Block 4 represents features extraction layers and the rest of it represents fully connected layer.

Tests

They have collected images from different noncommercial sources, i.e. Google search and many other websites. Their dataset includes ten classes of animals i.e. bear, cat, deer, dog, elephant, horse, lion, rabbit, raccoon, and rat. Each of the animal classes contains 1000 images.

Tests

Since the goal was to work with the faces, so all the images were taken in the frontal position with the easiness for some side movements. Some animal images had a different dimension.

So, all those images were resized to $227 \times 227 \times 3$ since their network only takes images with the same dimension.

The accuracy of the animal recognition system depends on the quality of the image dataset.

Tests

- The more different features it has, the more accuracy the animal classification system can achieve.

- The variance in color, edges, and corners will differentiate between classes

DATA SCIENCE INSTITUTE

Results

- To reach the proposed method, they went through several simulations.
- These simulations have helped to decide which parameters or structure to choose.
- Among all of those simulations, they discussed two structures below which we have labeled as “Linear DCNN + Batch Normalization + ELU” and “Proposed dual DCNN + Cross Channel Normalization + ReLU”.

Results

- To reach the proposed method, they went through several simulations.
- These simulations have helped to decide which parameters or structure to choose.
- Among all of those simulations, they discussed two structures below which we have labeled as “Linear DCNN + Batch Normalization + ELU” and “Proposed dual DCNN + Cross Channel Normalization + ReLU”.

Results

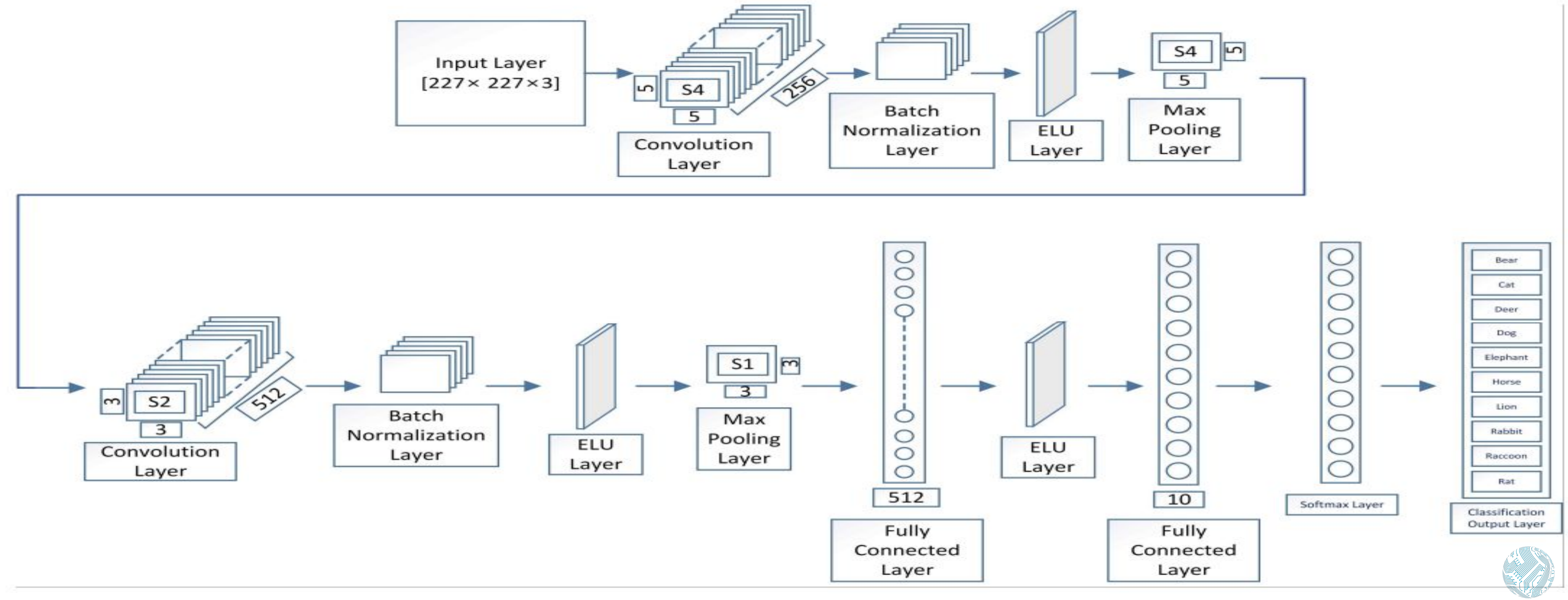
"Linear DCNN + Batch Normalization + ELU"

- A linear form of DCNN with the Batch Normalization layers and the ELU layer.

- In this structure, they have used two Convolution layers, two Batch Normalization layers, three ELU layers, two Max-Pooling layers, two Fully Connected layers and one of each Input layer, Softmax layer, Classification Output layer

Results

“Linear DCNN + Batch Normalization + ELU”



Results

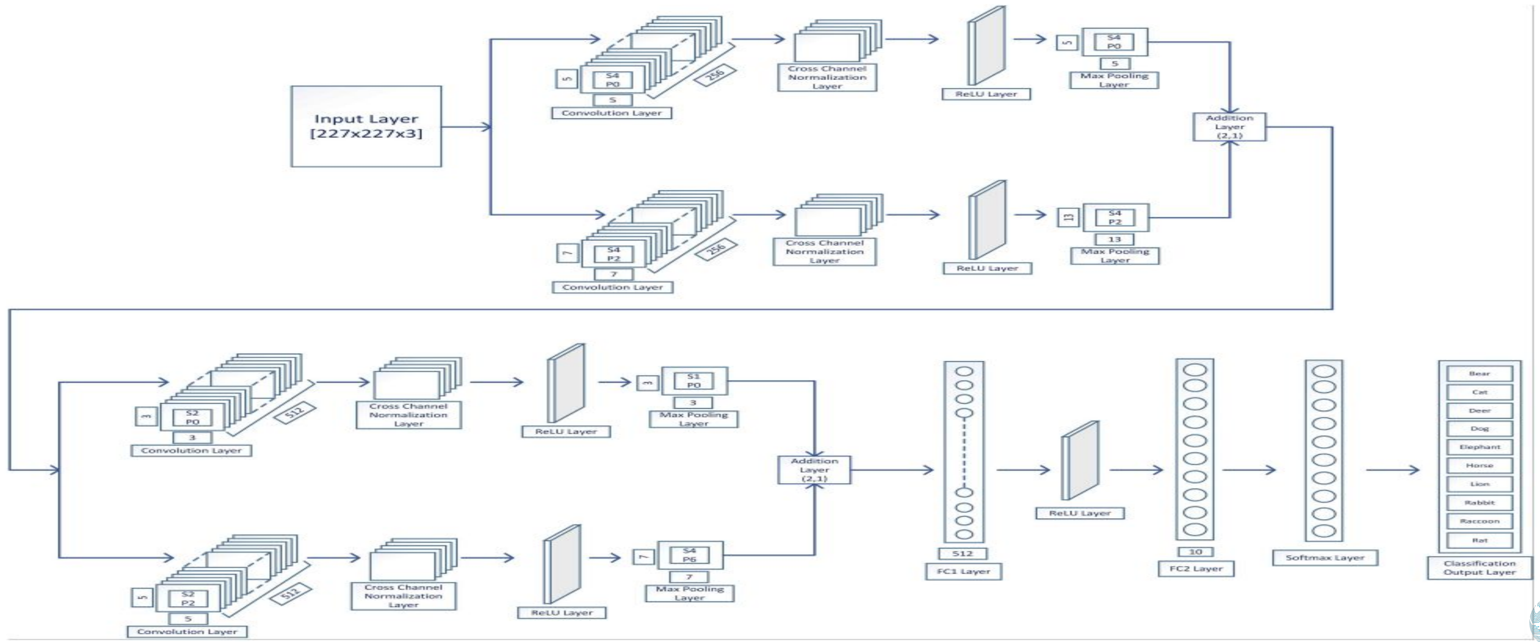
"Proposed dual DCNN + Cross Channel Normalization + ReLU".

- The proposed dual DCNN with the conventional CrossChannel Normalization layers and ReLU layers.

- The construction of this network follows the proposed dual DCNN except this one contains CrossChannel Normalization layers and ReLU layer instead of Batch Normalization layers and ELU layers respectively.

Results

"Proposed dual DCNN + Cross Channel Normalization + ReLU".



Confusion Matrix												
		Target Class										
		Bear	Cat	Deer	Dog	Elephant	Horse	Lion	Rabbit	Raccoon	Rat	Total
Output Class	Bear	179 8.9%	0 0.0%	0 0.0%	2 0.1%	3 0.1%	2 0.1%	1 0.1%	0 0.0%	2 0.1%	0 0.0%	
	Cat	1 0.1%	170 8.5%	2 0.1%	5 0.3%	4 0.2%	0 0.0%	2 0.1%	1 0.1%	0 0.0%	2 0.1%	
	Deer	0 0.0%	5 0.3%	179 8.9%	2 0.1%	1 0.1%	0 0.0%	0 0.0%	2 0.1%	0 0.0%	5 0.3%	
	Dog	2 0.1%	13 0.7%	2 0.1%	182 9.1%	4 0.2%	5 0.3%	2 0.1%	4 0.2%	2 0.1%	2 0.1%	
	Elephant	1 0.1%	2 0.1%	5 0.3%	2 0.1%	185 9.3%	4 0.2%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	
	Horse	8 0.4%	4 0.2%	6 0.3%	2 0.1%	1 0.1%	189 9.4%	1 0.1%	3 0.1%	1 0.1%	1 0.1%	
	Lion	2 0.1%	0 0.0%	1 0.1%	1 0.1%	1 0.1%	0 0.0%	194 9.7%	0 0.0%	0 0.0%	0 0.0%	
	Rabbit	1 0.1%	4 0.2%	2 0.1%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	183 9.2%	0 0.0%	5 0.3%	
	Raccoon	0 0.0%	0 0.0%	2 0.1%	2 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	195 9.8%	2 0.1%	
	Rat	6 0.3%	2 0.1%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	5 0.3%	0 0.0%	183 9.2%	
Total		89.5% 10.5%	85.0% 15.0%	89.5% 10.5%	91.0% 9.0%	92.5% 7.5%	94.5% 5.5%	97.0% 3.0%	91.5% 8.5%	97.5% 2.5%	91.5% 8.5%	92.0% 8.1%

Results

Methods	Accuracy%
DCNN of Guobin Chen et al [6].	68.8%
DCNN of Tibor Trnovszky et al [5].	74.9%
Proposed dual DCNN + Cross Channel Normalization + ReLU	88.9%
Linear DCNN + Batch Normalization + ELU	90.3%
Our proposed dual DCNN Model	92.0%

SOURCES:

- Financial Data Scientist- Muhammad Jawwad Javeed Iqbal
  - https://arxiv.org/pdf/1511.07289v5.pdf
  - https://www.sciencedirect.com/science/article/pii/S0925231220304240
  - https://www.docdroid.net/WoFdi1E/applsci-10-01897-pdf
  - https://www.researchgate.net/publication/319135247_An_ELU_Network_with_Total_Variation_for_Image_Denoising
  - https://towardsdatascience.com/visualizing-intermediate-activation-in-convolutional-neural-networks-with-keras-260b36d60d0
  - https://www.researchgate.net/figure/Visualization-of-the-Conv2D-and-Conv3D-feature-maps_fig2_343494739
  - https://www.researchgate.net/figure/Example-of-max-pooling-and-average-pooling-operations-In-this-example-a-4x4-image-is_fig4_332092821
  - https://towardsdatascience.com/a-simple-guide-to-convolutional-neural-networks-751789e7bd88
  - https://opendatascience.com/a-beginners-guide-to-understanding-convolutional-neural-networks/

SOURCES:

- Financial Data Scientist- Muhammad Jawwad Javeed Iqbal(Cont.)
  - https://www.researchgate.net/figure/Illustration-of-output-of-ELU-vs-ReLU-vs-Leaky-ReLU-function-with-varying-input-values_fig8_334389306
  - https://www.semanticscholar.org/paper/Fast-and-Accurate-Deep-Network-Learning-by-Linear-Clevert-Unterthiner/f63e917638553414526a0cc8550de4ad2d83fe7a
  - https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60
  - https://www.researchgate.net/figure/Schematic-diagram-of-InceptionV3-model-compressed-view_fig6_326421398
  - https://commons.wikimedia.org/wiki/File:Created_with_Matplotlib-logo.svg
  - https://paperswithcode.com/method/elu
  - https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

SOURCES:

- Blockchain Engineer / Project Lead - Alberto Navarrete
    - https://openaccess.thecvf.com/content_ICCV_2019/papers/Gu_Fast_Image_Restoration_With_Multi-Bin_Trainable_Linear_Units_ICCV_2019_paper.pdf
    - https://d-nb.info/1178101711/34
    - https://arxiv.org/pdf/1708.04317.pdf
    - https://github.com/chuk-yong/ELU-vs-ReLU-for-Image-Recognition
    - https://www.researchgate.net/publication/301980999_Face_Recognition_Based_on_Deep_Learning

- Financial Data Scientist - Moxú/Alberto Navarrete
    - Animal Face Classification using Dual Deep Convolutional Neural Network
    - FaceNet,CosFace,SphereFace, ArcFace.
    - davidsandberg/facenet: Face recognition using Tensorflow
    - deepinsight/insightface: State-of-the-art 2D and 3D Face Analysis Project
    - Deep Learning Face Representation from Predicting 10,000 Classes

SOURCES:

- Financial Data Scientist - Amirhossein Abaskohi
    - https://arxiv.org/abs/1503.03832
    - https://www.researchgate.net/profile/Shafagat_Mahmudova/publications
    - https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306
    - https://heartbeat.fritz.ai/the-5-computer-vision-techniques-that-will-change-how-you-see-the-world-1ee19334354b