PAPER REVIEW : ImageNet Classification with Deep Convolutional Neural Networks

~ Kevin Shah
20110096

MOTIVATION

This paper has been selected for review due to its significant impact on the field of deep learning and computer vision. This work unveiled the ground-breaking AlexNet architecture, which outperformed earlier state-of-the-art results and demonstrated impressive performance on the ImageNet dataset.

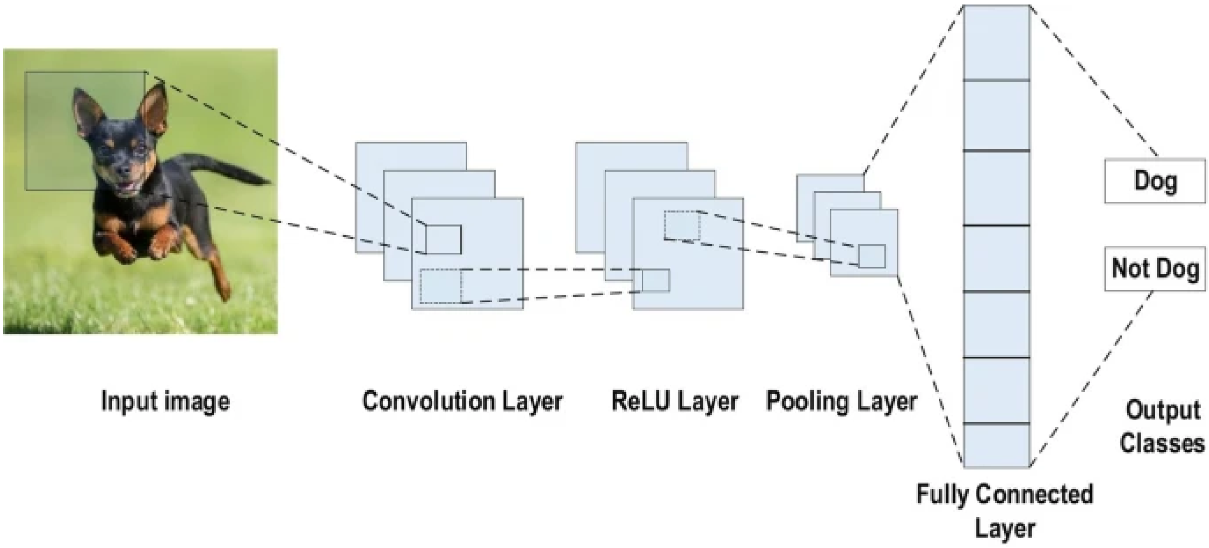
This paper's contributions to the advancement of deep convolutional neural networks serve as the inspiration for this review. The use of rectified linear units (ReLU), local response normalisation (LRN), overlapping pooling, and dropout regularisation are a few of the innovative design decisions that have gained traction in the deep learning community.

The paper's impact goes beyond its immediate effects as well, rekindling interest in deep learning and spurring new developments in computer vision. I seek to critically evaluate the methods, importance, and influence of this publication by doing a complete review, offering insightful analysis that advances on-going deep learning research.

In conclusion, the "ImageNet Classification with Deep Convolutional Neural Networks" paper's revolutionary advances in deep learning and computer vision, as well as their influence on following research, serve as the basis for this review. Through this review, I want to offer a thorough evaluation of the paper's techniques and determine how important it is for furthering the subject.

INTRODUCTION

One of the core problems in computer vision and a foundation of many imaging disciplines is image classification, a traditional study topic in recent years. The degree of its applications, such as object identification, segmentation, pose estimation, video classification, object tracking, and super-resolution technology, tends to considerably increase when classification network performance increases. Promoting the advancement of computer vision includes improving picture classification technology. Preprocessing of image data, feature extraction and representation, and classifier creation constitute the main components.



Basic CNN Architecture for Image Classification

Image feature extraction, a key component of image classification, has always been the subject of research. Traditional algorithms for extracting image features place more emphasis on manually specifying individual image features. Both the generalizability and portability of this approach are lacking. Therefore, researchers envision giving computers the ability to interpret images similarly to biological vision. A vast number of interconnected neurons make up an abstract biological neural network (ANN), a mathematical operation model. It simulates the processing of brain impulses by neural networks approximately. In the beginning, McCulloch and Pitts examined biological brain networks and produced the MP neuron model, a mathematical explanation of how neurons function internally. In order to bring the study on neural networks into practise, Rosenblatt created a single-layer perceptron model and added learning functions to the MP model.

They came to the conclusion that visual perception is activated layer by layer through multi-level receptive fields after Huber and Wiese et al. researched the visual cortex of the cat's brain and discovered that biological visual neurons interpret information based on local regional stimulation. Later, scientists attempted to learn features using the multilayer perceptron and trained the model using the backpropagation (BP) algorithm. The idea of building a computer neural network resembling a biological visual system was sparked by this discovery, and CNN was subsequently created. The CNN model's initial batch, known as LeNet-5, was presented by Lecun et al. However, the recognition results of LeNet-5 on complicated images were not perfect due to the lack of large-scale training data, it is also constrained by the theoretical underpinning, and computer computational capacity. This model performed exceptionally well at the time on handwriting recognition tests exclusively.

Hinton et al. opened a new chapter in deep learning by proposing an efficient learning algorithm for learning difficulties in multi-hidden-layer neural networks.

Later, researchers discovered the convolution operation on the GPU, considerably enhancing the network's computational effectiveness. It rose by 2–24 times faster than the CPU's operating speed. Deep learning has since garnered a growing amount of interest. The AlexNet model was created by Krizhevsky et al. based on the LeNet-5. It outperformed the second-best entry in the ILSVRC2012 ImageNet competition by a significant margin. After AlexNet excelled in the ImageNet image classification competition, academics started studying CNN more thoroughly. Zeiler and Fergus introduced ZFNet as a visualisation method to comprehend CNNs. In order to manage the parameter amount and channel count, Min Lin et al. proposed the NIN network. From 2017 to the present, successively more models with better performance have emerged. CNNs have consistently proven to be unbeatable at classifying images.

Around 2015, the use of CNNs for large-scale visual classification applications saw success, and the area of remote sensing image analysis has now fully embraced the technology. Several CNN-based scene categorization techniques have been developed by utilising various CNN-exploitation techniques. Generally speaking, there are three different types of CNN-based remote sensing picture scene categorization methods: The pre-trained CNNs are employed as a feature extractor; they are then fine-tuned on the dataset; and the weights of the CNNs are globally initialised for training. The CNN-based image classification technique was initially developed for computer vision, as is well known. However, numerous researchers have effectively used them in the remote sensing sector. To help researchers find inspiration for their future work, it is vital to thoroughly summarise the CNN-based image categorization techniques. Despite the fact that there have been surveys on CNNs, they haven't fully introduced practically all of the traditional CNN architectures. The goal of this review is to provide additional assistance for the inspiration of constructing CNN models in the field of remote sensing image scene classification by describing the evolution of practically all typical CNNs in image classification jobs.

IMPACT TO THE FIELD

To enhance the field of computer vision, the ImageNet Large Scale Visual Recognition Competition (ILSVRC), which was held annually from 2010 to 2017, was established. One of the declared purposes of the competition was "to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labelling effort." One of the largest labelled collections of high-resolution photos was used as the foundation for the competition, ImageNet.

The competition was essential for developing computer vision research, with the 2012 competition serving as a turning point when Alex Krizhevsky and colleagues introduced AlexNet, a deep convolutional neural network that achieved astounding results. According to Semantic Scholar, their study, ImageNet Classification using

Deep Convolutional Neural Networks, had a significant impact on the development of deep learning and received 60,000 citations. The authors' success can be attributed to reviving interest in deep learning and the development of deep learning techniques utilised in business applications today, even though convolutional neural networks (CNN) were not a new technology at the time but were not widely employed either.

PAPER SUMMARY

The first thing Krizhevsky, et al. mention is how machine learning techniques have the potential to accomplish straightforward recognition tasks that are comparable to human performance. However, object recognition in realistic environments is a far more difficult task that calls for a model with a lot higher learning capacity in addition to much larger datasets.

The authors outline a CNN called "AlexNet" to deal with these issues. Eight layers make up the CNN, five of which are convolutional and three of which are fully connected. Jason Brownlee provides a wonderful summary of CNN design and the functions of its layers.

Before delving into the specifics of the CNN design, it's vital to note that the authors highlight five things that are crucial to understanding how AlexNet is created:

1. ReLU Nonlinearity
2. Local Response Normalization
3. Overlapping Pooling
4. Data Augmentation
5. Dropout

1. ReLU Nonlinearity

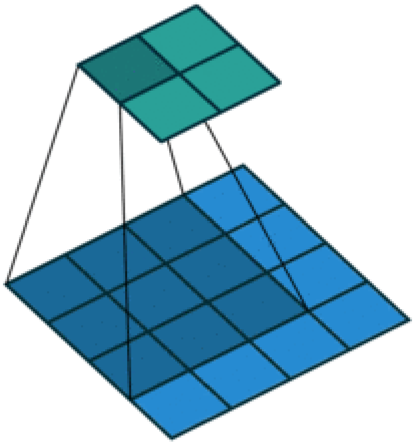
tanh and sigmoid were popular activation functions at the time of publication, but the authors remark that slow training times using gradient descent expose their flaws. According to them, utilising conventional techniques would not have allowed them to train such a vast network. The output of every layer in AlexNet is subjected to this ReLU nonlinearity. One of the long-lasting contributions to deep learning is their use of ReLU nonlinearity.

2. Local Response Normalization

Although the authors point out that input normalisation is not necessary for ReLUs, they discovered that some local normalisation enhanced generalisation. The first and second convolutional layers of AlexNet were followed by the use of ReLU nonlinearity, and then local normalisation was applied.

3. Overlapping Pooling

The authors provide a very simple explanation of their overlapping pooling strategy: "Pooling layers in CNNs summarize the outputs of neighboring groups of neurons in the same kernel map. Traditionally, the neighborhoods summarized by adjacent pooling units do not overlap. To be more precise, a pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a neighborhood of size z x z centered at the location of the pooling unit. If we set s = z, we obtain traditional local pooling as commonly employed in CNNs. If we set s < z, we obtain overlapping pooling. This is what we use throughout our network, with s=2 and z=3."



4. Data Augmentation

The authors employ two augmentation techniques to prevent overfitting, including "generating image translations and horizontal reflections" and "altering the intensities of RGB channels in training images."

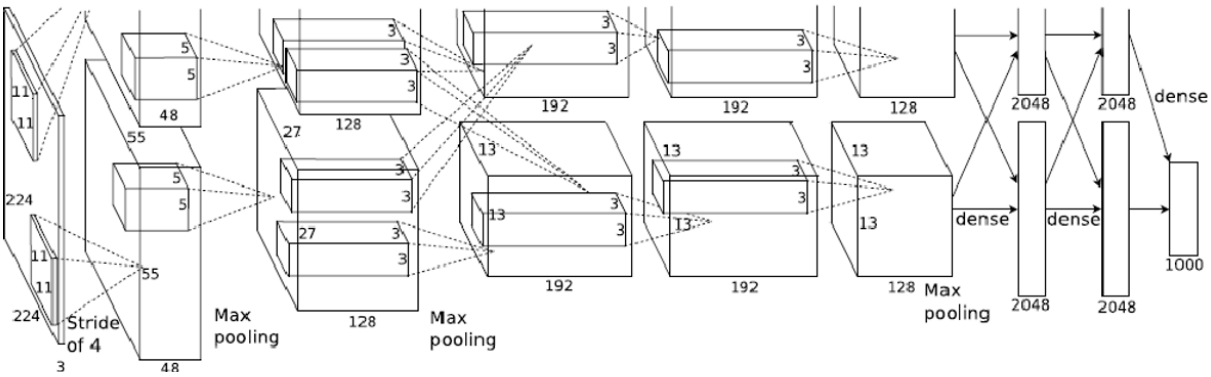
Since the dimensions of the images in the ImageNet dataset vary, they were down-sampled to a fixed resolution of 256 × 256 in order to give the CNN constant input dimensionality. Random 224 × 224 patches of those 256 × 256 images are extracted for data augmentation, and horizontal reflections of such images are also produced.

The RGB pixel values are subjected to PCA by the authors in order to change the intensity of the RGB channels. For each training image, they "add multiples of the found principal components, with magnitudes proportional to the corresponding eigenvalues times a random variable drawn from a Gaussian with mean zero and standard deviation 0.1."

5. Dropout

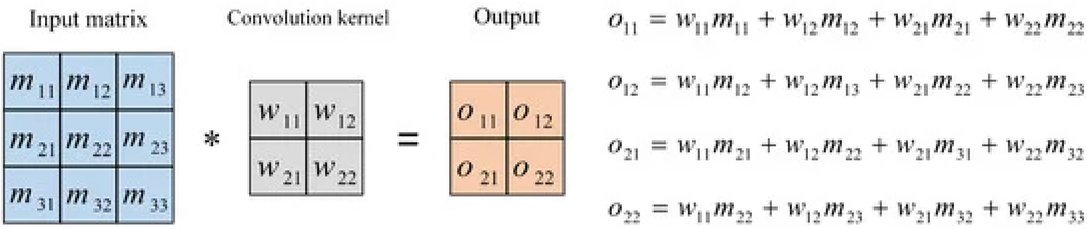
The output of each hidden neuron with probability 0.5 is set to 0. Therefore, neither the forward pass nor the back-propagation are affected by these neurons. The first two fully linked layers employ dropout, and the authors emphasise that their model would have overfitted in the absence of dropout.

ARCHITECTURE

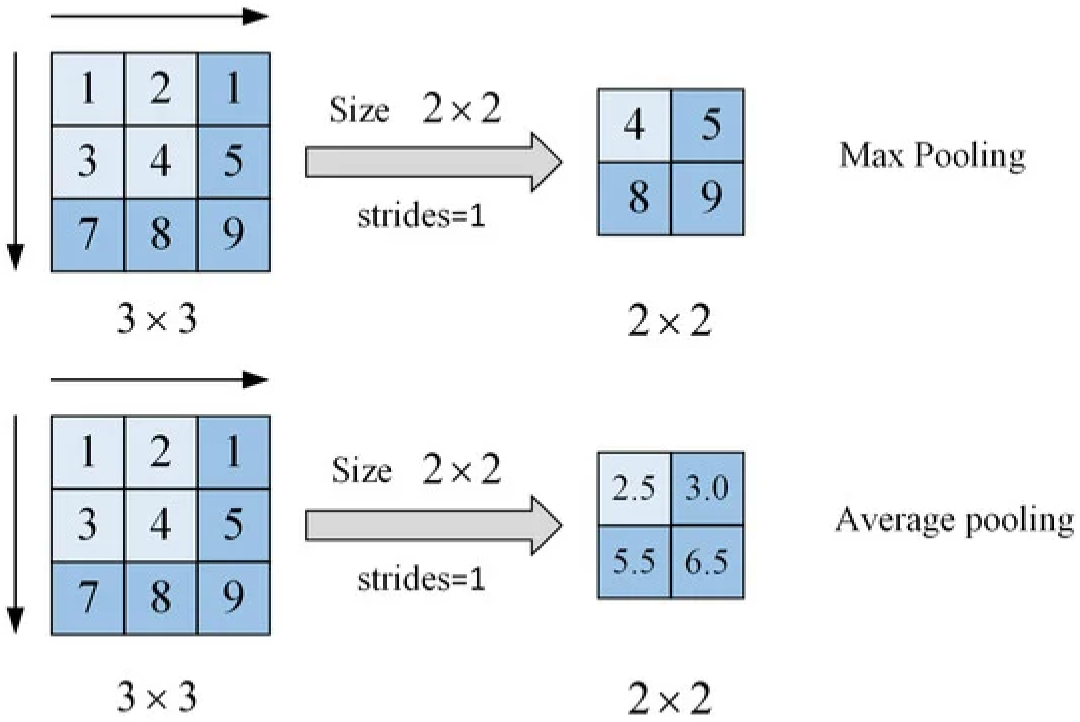


AlexNET Architecture

The model was trained on two GPUs with an initial input of 224x224x3 (for three colour channels). Below is a description of each layer. The architecture of AlexNet is shown in the paper's graphic above, along with the dual GPUs' concurrent workloads.


Input matrix
Convolution kernel
Output
m11 m12 m13
m21 m22 m23
m31 m32 m33
*
w11 w12
w21 w22
=
o11 o12
o21 o22
o11 = w11m11 + w12m12 + w21m21 + w22m22
o12 = w11m12 + w12m13 + w21m22 + w22m23
o21 = w11m21 + w12m22 + w21m31 + w22m32
o22 = w11m22 + w12m23 + w21m32 + w22m33

Convolution Operation (2-D), kernel size = 2, strides = 1, padding = 0



Max Pooling and Average pooling, it does not involve zero padding.

1. **Convolutional Layer** (ReLU, Max Pooling, Local Response Normalisation): The first layer is a convolutional layer with an input picture of $224 \times 224 \times 3$ dimensions and filters with 96 $11 \times 11 \times 3$ kernels (with a stride of 4 pixels), resulting in an output of $55 \times 55 \times 96$. In order to produce a $27 \times 27 \times 96$ tensor, this output is first subjected to a ReLU nonlinearity activation function, then max-pooling, and finally local response-normalization.

2. **Convolutional Layer** (ReLU, max pooling, local response normalization): The second convolutional layer filters the $27 \times 27 \times 96$ tensor that the first convolutional layer produced with 256 kernels of size $5 \times 5 \times 48$, resulting in a $27 \times 27 \times 256$ output. A ReLU nonlinearity activation function, max-pooling, local response normalisation, and output as a $13 \times 13 \times 256$ tensor are applied to this output.

3. **Convolutional Layer** (ReLU, no pooling, no response normalization): The third convolutional layer uses the output of the second layer, which is $13 \times 13 \times 256$, as its input. It filters the tensor with 384 kernels of size $3 \times 3 \times 256$, producing an output of size $13 \times 13 \times 384$, which is then passed through a ReLU nonlinearity activation function and on to the following layer (note that there is no pooling or local response normalisation as in the previous layers).

4. **Convolutional Layer** (ReLU, no pooling, no response normalization): The $13 \times 13 \times 384$ tensor produced by the third convolutional layer is used as the

input for the fourth convolutional layer, which subsequently filters the data using 384 kernels with a size of $13 \times 13 \times 192$ before activating ReLU nonlinearity. Size of the output is $13 \times 13 \times 384$.

5. **Convolutional Layer** (ReLU, max pooling, no response normalization): TThe fourth convolutional layer's $13 \times 13 \times 384$ tensor is fed into the fifth and final convolutional layer, which has 256 kernels each measuring $13 \times 13 \times 192$. An output tensor with a size of $6 \times 6 \times 256$ is created by passing the output through a ReLU nonlinearity activation function, max-pooling, and no local response normalisation.

6. **Fully-connected Layer** (ReLU): The output of the fifth layer, a $6 \times 6 \times 256$ tensor, is fed into the sixth layer, the first fully linked layer, which produces a 4096×1 tensor after passing it through a ReLU nonlinearity activation function. Additionally, the aforementioned dropout technique is used in this layer.

7. **Fully-connected Layer** (ReLU): Another fully-connected layer, the seventh layer receives as an input the 4096×1 tensor produced by the sixth layer and produces a 4096×1 tensor that is then processed through a ReLU nonlinearity activation function. In this layer, dropout is also utilised.

8. **Fully-connected Layer** (softmax): The final layer is a fully connected layer that receives as an input the 4096×1 tensor produced by the seventh layer and produces a 1000×1 tensor that is fed via a softmax activation function and contains the predictions.

EVALUATION METRICS

Achieving the optimised classifier depends heavily on the evaluation criteria used in DL tasks. They are used during two key stages of a typical data categorization process: training and testing. During the training phase, it is used to improve the classification algorithm. This indicates that the evaluation measure is used to discriminate and choose the optimised answer, such as as a discriminator that can produce an exceptionally accurate forecast of impending evaluations pertaining to a certain classifier. Currently, the constructed classifier's effectiveness is assessed using the evaluation metric, for example, as an evaluator during the model testing step utilising hidden data. According to $Eq.\ 1$, TN and TP are the number of correctly identified negative and positive examples, respectively. The quantity of incorrectly identified positive and negative instances is also defined as FN and FP, respectively. The following list includes some of the most popular evaluation metrics.

1. **Accuracy**: Estimates the proportion of correctly predicted classes to all the samples that have been assessed.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

2. **Recall**: Calculates the proportion of correctly recognised positive patterns

$$Recall = \frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

3. **Precision**: Determines the positive patterns that are appropriately predicted by all patterns in a class of positive patterns.

$$Precision = \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

4. **Specificity**: Determines the percentage of negative patterns that are classified properly.

$$Specificity = \frac{TN}{FP+TN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

5. **F1-Score**: Computes the harmonic mean of the recall and precision rates.

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision+Recal} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

6. **J Score**: This metric is also known as the Youdens J statistic. It represents the metric:

$$J_{score} = Sensitivity + Specificity - 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

7. **False Positive Rate (FPR)**: This metric relates to the potential for a false alarm ratio as determined by $Eq.\ 7$

$$FPR = 1 - Specificity \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

8. **Area Under the ROC Curve**: AUC is a widely used ranking statistic. It is used to create the best possible learning model and to compare different learning algorithms. The AUC value exposes the whole classifier ranking performance, unlike probability and threshold measurements. The AUC value for a two-class problem is calculated using the following formula:

$$AUC = \frac{S_p - \frac{n_p(n_n+1)}{2}}{n_p n_n} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Here, S_p stands for the total number of samples with a positive rank. The letters n_n and n_p, respectively, stand for the number of negative and positive samples. When compared to accuracy metrics, the AUC value was theoretically and experimentally supported, which made it particularly useful for locating an optimal solution and assessing classifier performance via training.

The AUC performance was outstanding when taking into account the evaluation and discriminating processes. However, when discriminating a large number of developed solutions for multiclass problems, the AUC computation is primarily cost-effective. Additionally, according to the Hand and Till AUC model and the Provost and Domingo AUC model, the time complexity for computing the AUC is $O(|C|2n\ log\ n$ and $O(|C|n\ log\ n)$, respectively.

WHEN TO APPLY DEEP LEARNING

Machine learning is helpful in many situations and is comparable to or superior to human specialists in various instances, therefore DL could be a solution to the following issues:

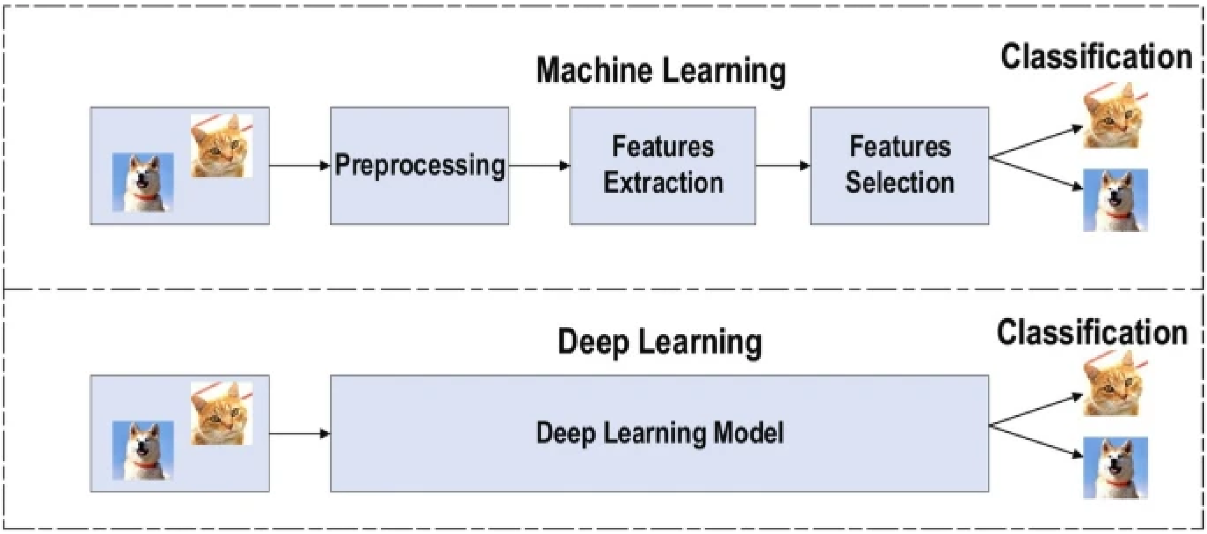
- Where no human experts are available.
- When individuals are unable to explain decisions made utilising their knowledge (such as in speech recognition, language understanding, and medical decisions).
- When the solution to the problem changes over time (price prediction, stock preference, weather prediction, and monitoring).
- When solutions need to be modified based on particular circumstances (personalization, biometrics).
- When the complexity of the problem is so high that it defies our finite capacity for thinking (sentiment analysis, matching Facebook advertisements to ads, ranking webpages).

WHY DEEP LEARNING?

The reasons for levargin deep learning are:

- **Approach to Universal Learning:** Deep Learning is frequently referred to as universal learning because it can function in nearly all application fields.
- **Robustness**: In general, Deep Learning approaches do not need precisely constructed features. Instead, the optimised traits are automatically taught in relation to the work at hand. Thus, robustness against the input data's typical variations is accomplished.
- **Generalization**: Different applications or data types can employ the same deep learning technology, a method known as transfer learning. Additionally, it is a helpful method for issues where there is a lack of data.
- **Scalability**: Deep Learning can scale up very well. Microsoft created ResNet, which consists of 1202 layers and is often used at a supercomputing scale. A similar strategy was followed by Lawrence Livermore National Laboratory

(LLNL), a big organisation focused on developing frameworks for networks that can support thousands of nodes.



Difference between Traditional Machine Learning and Deep Learning

SIGNIFICANCE AND ORIGINALITY

The paper's substantial influence on the fields of deep learning and computer vision is what provides it relevance. Traditional machine learning techniques had trouble achieving high accuracy on challenging image recognition tasks like the ImageNet dataset prior to the release of this study. Using an innovative strategy, the "ImageNet Classification with Deep Convolutional Neural Networks" study effectively overcame this difficulty.

The paper's introduction of the AlexNet architecture is one of its major accomplishments. The eight layers of the AlexNet model, comprising several convolutional and fully linked layers, were ground-breaking at the time of its publication for their depth and complexity. The authors showed that considerable gains in picture classification accuracy might be achieved by utilising deeper architectures and training on large-scale datasets.

The paper has various original elements. First, to solve the vanishing gradient problem and hasten training, the authors used rectified linear units (ReLU) as activation functions in the network. In contrast to more conventional activation functions like sigmoid or tanh, this choice of activation function proved to be very successful in training deeper networks.

In the convolutional layers, the authors also proposed the notion of local response normalisation (LRN). LRN boosted the network's generalisation skills and improved the contrast between various feature maps. This method improved the performance of the architecture by introducing a new component.

In addition, overlapping pooling layers were used by the authors rather than non-overlapping pooling layers. Important spatial information was maintained by allowing overlap between pooled regions, producing more informative downsampled feature maps.

The use of dropout regularisation was another innovative addition. This method, which involves setting a portion of neuron activations to zero at random during training, assisted in avoiding overfitting and enhanced the network's generalisation capabilities. Since then, dropout has gained popularity as a regularisation method in deep learning.

Beyond attaining state-of-the-art performance on the ImageNet dataset, the work had a wider influence. It reignited interest in deep learning and showed how deep CNNs may be used for a variety of computer vision tasks. The success of the AlexNet architecture and the methods presented in the study sparked additional research and deep learning improvements, which influenced the creation of more complex CNN architectures and aided in the quick development of computer vision.

In conclusion, the "ImageNet Classification with Deep Convolutional Neural Networks" paper's introduction of the AlexNet architecture, usage of ReLU activations, LRN, overlapping pooling, and dropout regularisation make it both highly significant and novel. The subject of deep learning and computer vision research saw a radical change as a result of these ground-breaking components and their effective application to the classification of massive amounts of images.

References

1. https://www.datascienceherald.com/post/classics-imagenet-classification-with-deep-convolutional-neural-networks
2. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8
3. https://www.mdpi.com/2072-4292/13/22/4712