Image Classification Based On CNN: A Survey

Ahmed A. Elngar[1], Mohamed Arafa[2], Amar Fathy[3], Basma Moustafa[4], Omar Mahmoud[5] and Mohamed Shaban[6]*, Nehal Fawzy[7]

[1] Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef City, 62511, Egypt, E-mail: elngar_7@yahoo.co.uk

[2] Teaching assistant, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: jenuiman4rt@gmail.com

[3] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: fathyamar78@gmail.com

[4] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: basmamoustafa31@gmail.com

[5] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: mora50017@gmail.com

[6] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: muahmmedshaban153@gmail.com

[7]Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt, Email: Neehal.fawzy91@yahoo.com

(Corresponding Author: Mohamed Shaban[6])

Abstract: Computer vision is one of the fields of computer science that is one of the most powerful and persuasive types of artificial intelligence. It is similar to the human vision system, as it enables computers to recognize and process objects in pictures and videos in the same way as humans do. Computer vision technology has rapidly evolved in many fields and contributed to solving many problems, as computer vision contributed to self-driving cars, and cars were able to understand their surroundings. The cameras record video from different angles around the car, then a computer vision system gets images from the video, and then processes the images in real-time to find roadside ends, detect other cars, and read traffic lights, pedestrians, and objects. Computer vision also contributed to facial recognition; this technology enables computers to match images of people's faces to their identities. which these algorithms detect facial features in images and then compare them with databases. Computer vision also play important role in Healthcare, in which algorithms can help automate tasks such as detecting Breast cancer, finding symptoms in x-ray, cancerous moles in skin images, and MRI scans. Computer vision also contributed to many fields such as image classification, object discovery, motion recognition, subject tracking, and medicine. The rapid development of artificial intelligence is making machine learning more important in his field of research. Use algorithms to find out every bit of data and predict the outcome. This has become an important key to unlocking the door to AI. If we had looked to deep learning concept, we find deep learning is a subset of machine learning, algorithms inspired by structure and function of the human brain called artificial neural networks, learn from large amounts of data. Deep learning algorithm perform a task repeatedly, each time tweak it a little to improve the outcome. So, the development of computer vision was due to deep learning.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020

DOI: 10.5281/zenodo.4897990

Now we'll take a tour around the convolution neural networks, let us say that convolutional neural networks are one of the most powerful supervised deep learning models (abbreviated as CNN or ConvNet). This name "convolutional" is a token from a mathematical linear operation between matrixes called convolution.

CNN structure can be used in a variety of real-world problems including, computer vision, image recognition, natural language processing (NLP), anomaly detection, video analysis, drug discovery, recommender systems, health risk assessment, and time-series forecasting. If we look at convolutional neural networks, we see that CNN are similar to normal neural networks, the only difference between CNN and ANN is that CNNs are used in the field of pattern recognition within images mainly. This allows us to encode the features of an image into the structure, making the network more suitable for image-focused tasks, with reducing the parameters required to set-up the model. One of the advantages of CNN that it has an excellent performance in machine learning problems. So, we will use CNN as a classifier for image classification. So, the objective of this paper is that we will talk in detail about image classification in the following sections.

Keywords: *Computer vision, Convolutional Neural Networks, Artificial neural network, KNN, Support Vector Machine, Image classification.*

1. Overview:

1.1 Introduction:

According to the Internet Center (IDC), the total global data volume has reached 42ZB in 2020. More than 70% of the information is transmitted as photos or video. To extract useful information from these images and video data, computer vision appeared as needed. At present, computer vision technology has developed rapidly in the field of image classification, object detection, face recognition, motion recognition, and tracking the goal, medicine. As an important research component of machine learning and computer vision analysis, image classification is an important theoretical basis and technical support to further the development of AI. Image classification began in the late 1950s and has been widely used in various fields of engineering, fingerprints, human vehicle tracking, geology, climate detection, resources, disaster monitoring, medical testing, communications, agricultural automation, the military, and other fields.

In fact, Image classification is a classical problem of image processing, computer vision, and machine learning fields. And there are several methods that can be used to obtain a higher discrimination capability in various classification problems such as (Convolutional Neural Networks, Artificial neural network, KNN, Support Vector Machine) and we will discuss about them through next. But in this paper, we use deep learning to image classification, During convolutional neural network.

In this paper, we will talk about many sections. in section (1), we discuss the definition of image classification, general approaches of image classification through Supervised and Unsupervised Classification, Supervised and Unsupervised processes, and how these algorithms work. Then in section (2), we will explain the steps of image classification, including digital data, preprocessing (Normalized image contrast enhancement, Gray scale image, Binary image, resize image, complemented binary image), feature extraction, selection of training data, classification of the image, and Accuracy assessment.

In section (3), we will turn to the algorithms used in the classification (Artificial neural network, KNN, Support Vector Machine), We will talk about the basic idea and characteristics of each algorithm, how it works by steps, the Mathematics of each algorithm, the advantages, and disadvantages of each. And finally, we will discuss all information related to CNN, CNN architecture, the four layers that Convolution used, including pooling layer, non-linearity layer, and fully-connected layer and we will explain the mathematics of each layer, and the advantages and disadvantages of Convolutional Network Architecture. In section (4), after an accurate study in the field of image classification, our research team has worked on an example in the field of

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020

DOI: 10.5281/zenodo.4897990

computer vision, which is the prediction of the object in the image and extract information from an image through using a library specialized in the field of deep learning such as TensorFlow, from which some libraries such as keras, Dense, Flatten, Conv2D, MaxPooling2D were used Dropout, and it will be detailed on the next pages.

1.2 Image Classification:

Image classification is one of the most well-known tasks in computer vision and it's a complex process that may be affected by many factors. It allows for the classification of a given image as belonging to one of a set of labeled and pre-defined categories. It relies mainly on deep learning in the classification or prediction process, and its function is to extract information from the image [1]. Let's take a simple example: we want to categorize images according to whether they contain a tourist attraction or not. we suppose that the classifier is built for this purpose and that the image below is provided and existed.

1.3. Image Classification approaches.

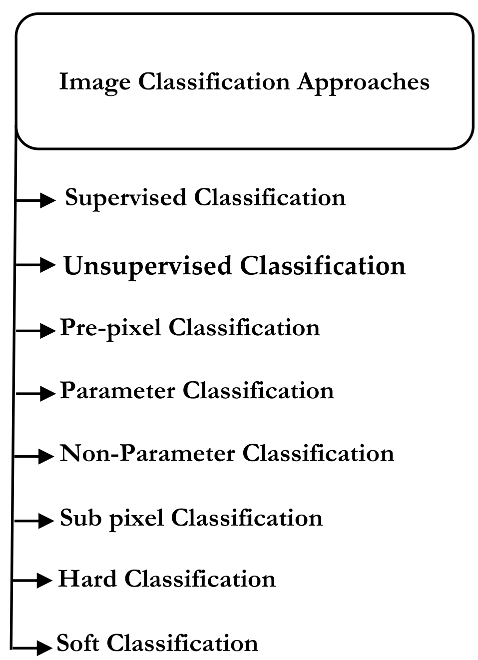
Image classification refers to the task of extracting information classes from an image, or as a some of image classification technique process of sorting pixels into a finite number of individual classes, or categories of data, based on their spectral response (the measured brightness of a pixel across the image bands, as reflected by the pixel's spectral signature)

The system of image Classification consists of a public database that contains predefined patterns that compare with an object to classify to the appropriate category.

We should know that Image classification is one of the important and complex processes in image processing. It's a crucial and challenging task in various application domains. Depending on the interaction between the analyst and the computer during classification. There are several image classification methods we will mention and explain extensively both type of them. The two main image classification methods are supervised classification and unsupervised classification.

1.4. Approaches of image classification:

obtaining data from various sensors that have distinctive features that compose the image. Fusion or integration of data is performed on multi-sensor or multi-resolution data which results in a marked improvement of visual interpretation and quantitative analysis. Basically, Data fusion has three levels in which are given, there are pixel, feature, and decision. Data fusion comprises of two main steps: First geometrical co-registration of two datasets and second a combination of spectral and spatial information contents used to generate a new dataset that contains information from both the datasets [3]. The best extraction and data extraction accuracy from both datasets, especially the line features, such as roads, rivers, and other known objects.



- **In our research paper we will focus on Supervised and Unsupervised Classification as they are widely used and are known in real.**

1.4.1 Supervised Classification:

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

Supervised classification uses image pixels representing regions of known, homogenous surface composition (training areas) or uses the spectral signatures obtained from training samples to classify an image.
In supervised classification image is divided into user classes defined by the user like (grass, dog, water), some pixels are known grouped and gives the label to these predefined classes. In this process, we call it training. After that classifier or method uses trained pixels to classify other images. In this method, prior information is needed and required before the testing process, evaluation process and it must be collected by the analyst. In this analyst identifies representative training sites for each informational class and here algorithm generates decision boundaries. Commonly used supervised classification approaches minimum distance to mean, and maximum likelihood. Process steps of supervised classification approach are [2]:

Supervised Classification Process

- Initially, we determine a classification scheme.
- Just create training area for each info class by analyst.
- Generate signatures identifies like (mean, variance, covariance).
- All pixels are classified.
- Map Informational Class.

Popular classifiers used: -

As known that there are varies classifiers in supervised classification techniques and they are Maximum likelihood, minimum distance, artificial neural network, decision tree classifier

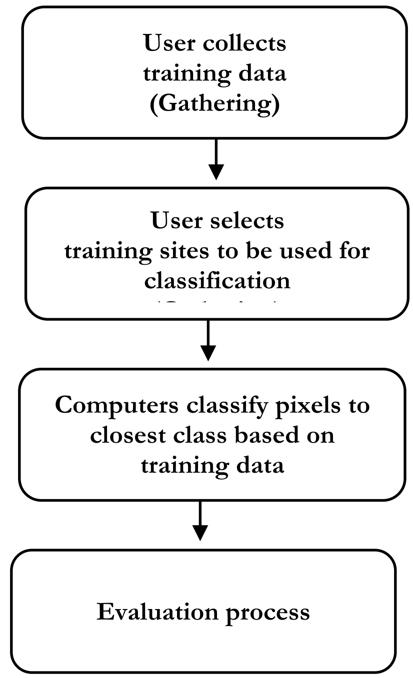


Fig 1. Supervised Classification

- **Initially, we determine a classification scheme.**

- The purpose of the classification is important to declare.
- Just making the scheme as precise as the resources and the reference data available allow.

  - **To make it less precise, you can always generalize the Classification scheme to make it more specific.**

- **Create Training Areas for each info class by analyst.**

- Digitizing drawing polygons around areas in the image.
- Seeding- areas based on spectral similarity to seed pixel.
- Using existing data existing maps, field data high-resolution imagery.
- Feature space image training areas.

Supervised classification algorithms mechanism.

When we talk about ML specially supervised classification, we can say that we have a set of instances which can be represented by all these black points in graph 1, so we have here an example of feature which I call X and the feature Y in combining these two features we get (feature space) when we apply a supervised classification algorithm on that we aim to get these points here and discover to each area all these points belong. [4]
When we talk about supervised classification we talk about a prior information or definition, this mean that we have a certain acknowledge about this graph, here we know the instances we know the classes to each some of the instances belong all what I say in previous we can see in graph two. After training stage all

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

points in the red region belongs to class A, all green region belongs to class B and all points in blue region belongs to class C as we see in fig.

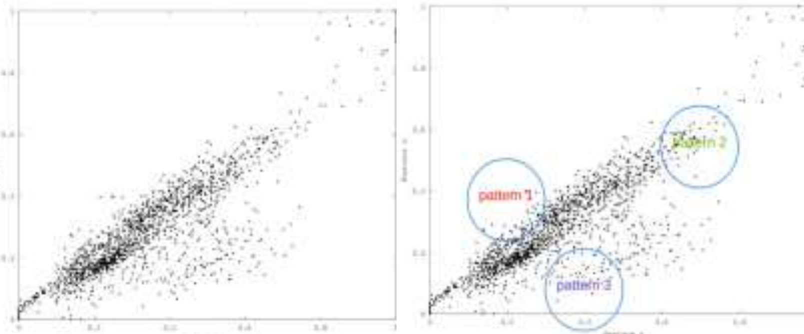
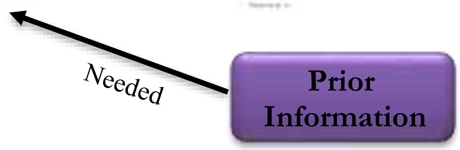


Fig 2. Supervised Classification



1.4.2 Unsupervised classification:

Unsupervised classification is q clustering-based algorithms that are used to partition the spectral image into a number of (Clusters) spectral classes based on the statistical information inherent in the image. No prior information and definitions of the classes are used, and this type of classification is a method that examines a large number of unknown pixels and divides it into a number of classes based on natural groupings present in the image values. The responsible person (analyst) role come on labeling and merging the spectral classes into meaningful classes. It's used when no trained pixels are available.

The machine determines the spectrally separable class and then defines its information value. No extensive prior information is required. Example: K-means algorithm

In unsupervised classification, pixels are grouped with the help of their properties. This process known as clustering and groups is known as a cluster. In this user decide how many clusters he wants. In unsupervised classification, prior information is not needed. It does not require human annotation; it is fully automated. This algorithm identifies clusters in data and also analyst labels clusters. The steps in unsupervised classification a

Unsupervised Classification Process:

- Initially, we determine a classification scheme.
- Clustering data.
- All pixels are classified based on clusters.
- Spectral class mapping.
- Analyst are labeling and merging clusters.
- Mapping informational class process.

Popular classifiers used:

From definition we realized that it is mainly based on K-means clustering algorithm, ISODATA.

The basic idea of unsupervised classification algorithms.

The mean idea of unsupervised classification algorithms is that a prior information is unavailable, so we do not have a prior information (we have defined prior info in supervised classify-cation) so the user only

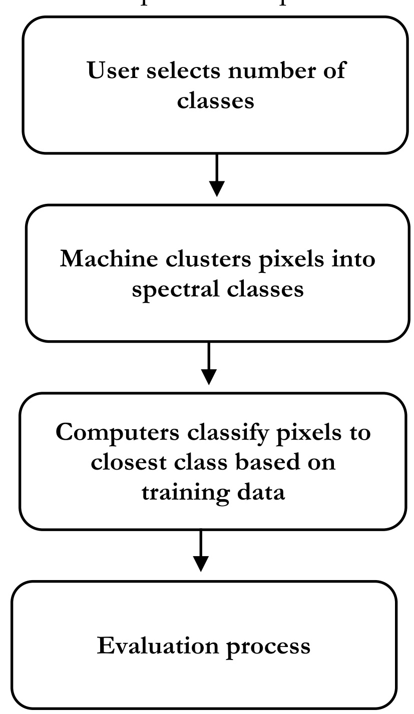


Fig 3. Unsupervised Classification

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

know the set of data in this features that belongs to the instances that we have but the user does not know anything else about data and want to apply some strategy that we will try to find groups of points (clusters) in feature space and the process of finding groups of these points is a test set we use to call (clustering)

- Clustering is that we try to find some groups of points that are similar in this feature space for example we will show in the following graph.

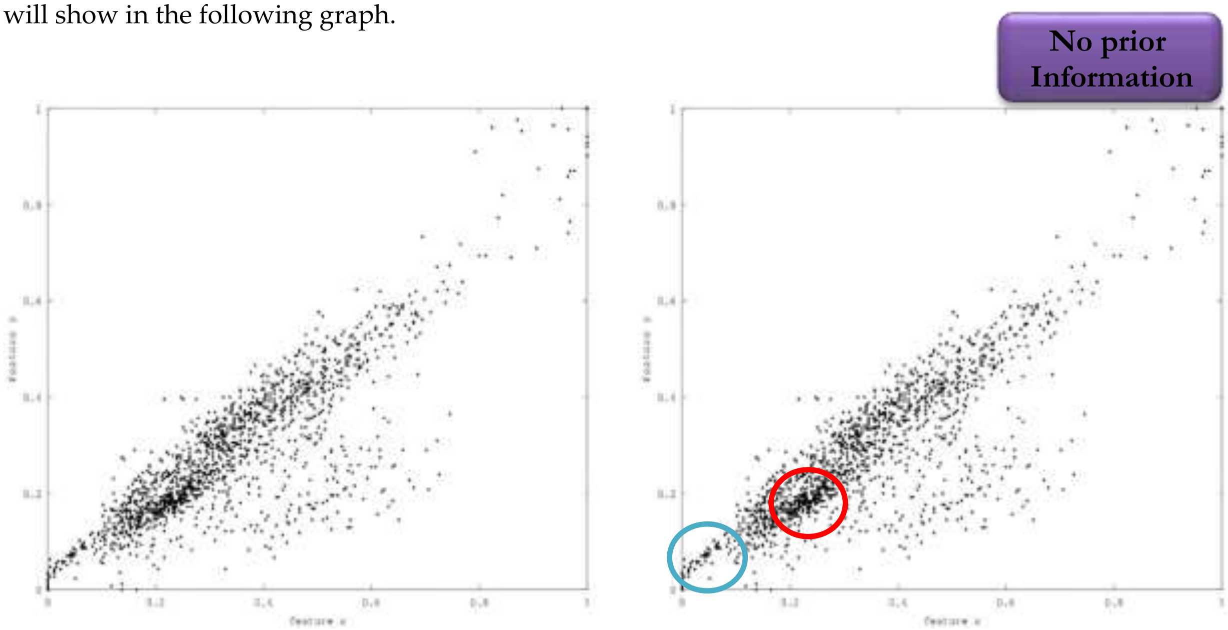


Fig 4. Unsupervised Classification

For example:

We can say the points in blue circle are closed to each other, so they belong to a cluster and also other group of points belongs to.

Another cluster in red circle that is not like the red circle cluster but is another cluster and also this other groups in space, but for ex we could also make some different inferences in this data and say for example that all these points in the midline are similar because they are continuous line that they seem to have the similar parameters.

2. Steps of Image Classification:

2.1. digital data:

In this step we capture images by using digital camera or any camera of mobile phone and in this stage the images must be captured in a high quality because this affects the accuracy of the output, So we must give it attention.

2.2. preprocessing:

In this step we improvement of the image data. Normalized image contrast enhancement, gray scale image, binary image, resize image, complemented binary image and noise removal boundary image.

I. Normalized image contrast enhancement:

Normalization is a mechanism that changes the range of pixel intensity values in image processing. Applications include images, for example, with poor contrast due to glare. Normalization is also called stretching of contrast or stretching of histograms. It is referred to as dynamic range expansion in more general fields of data processing, such as optical signal processing.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

In different applications, the object of dynamic range expansion is typically to put the image or other form of signal into a range that is more familiar or natural to the senses, hence the normalization of the term. The motivation is also to maintain dynamic range consistency for a collection of data, signals, or images to prevent mental interruption or fatigue. For example, a newspaper would aim to share a similar selection of grayscale with all the images in an issue.

II. Gray scale image:

A grayscale (or gray-level) image is simply one in which shades of gray are the only colors. The explanation for separating such images from any other kind of color image is that for each pixel, less information needs to be given. A 'gray' color is one in which all of the red, green and blue components have the same intensity in RGB space, Thus, it is only appropriate to specify a single intensity value for each pixel, as opposed to the three intensities required for a full color image to be defined for each pixel. The strength of the grayscale is also stored as an 8-bit integer that gives 256 possible different gray shades from black to white. The disparity between successive gray levels is slightly greater than the Gray level resolving power of the human eye if the levels are equally spaced. it happens by converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components:

$$0.2989 * R + 0.5870 * G + 0.1140 * B$$

Fig 5. Gray Scale Image

III. Binary image:

Fig 6. Binary Image

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

Binary image is the image that whose pixels only have two possible values for intensity. It represented as white and black. Numerically, for black, the value is always 0, and for white, either 1 or 255.

IV. resize image:

When you resize or distort the picture from one-pixel grid to another, image interpolation occurs. When you need to increase or decrease the total number of pixels, image resizing is important, while remapping can occur when you are correcting for lens distortion or rotating an image. Zooming refers to increasing the number of pixels, so you can see more detail when you zoom in on an image. To make resize, need to calculate the width after the rescale to relate to the height in the same way:

$$Wn = (Hn * Wo) / Ho$$

- **Wn:** new width
- **Wo:** old width
- **Hn:** new height
- **Ho:** old height

Fig 7. Resize Image

Note: the aspect ratio of the new size must be equal to the old size, for example:
assume an incoming image of 640x480. This has an aspect ratio of 1.33333. It will then take a new width of 60 * 640 / 480, or 80, to rescale this to 60 pixels high, which seems right because 80/60 is indeed 1.3333.

V. complemented binary image:

Fig 8. Complemented binary Image

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

Zeros become ones in the complement of a binary image, and ones become zeros. White and black are inverted. Each pixel value is subtracted from the maximum pixel value supported by the class to complement the grayscale or color image (or 1.0 for double-precision images). The difference will be used in the output image as the pixel value. Dark areas will be lighter in the output image, and light areas will be darker. Reds will be cyan, greens will be magenta, blues will be yellow.

2.3. feature extraction:

Feature extraction is part of the process of reducing dimensionality, in which an initial collection of raw data is separated and reduced to groups that are more manageable. So, it will be smoother when you want to process it. In these massive data sets, the most important feature is that they have a large number of variables. To process them, these variables need a lot of computational resources. So, the extraction of features allows to pick and merge variables into features to get the best feature from those big data sets, thus essentially reducing the amount of data. These features are simple to process, but the actual data set can still be represented with accuracy and originality. When you have a broad data set, the technique of extracting the characteristics is

useful and you need to minimize the number of resources without missing any significant or relevant information. Extraction of functionality helps decrease the amount of redundant data from the data collection.

2.4. selection of training data:

In general, the richer your dataset is, the better your model performs when it comes to machine learning. Moreover, in order to ensure the balancing of the dataset, the number of data points should be identical across groups. However, in terms of dataset size, how you interpret the labels will affect the minimum requirements. On our example, we have number of 6000 images per class that you want to detect. In certain cases, however, to achieve high-performing systems, more data per class is required. If you are trying to identify a higher number of marks, you must change the dataset of your image accordingly. If within a class you're looking for greater granularity, then you need a greater number of images. For each added sub-label, you need to ensure that you reach the threshold of at least 100 images.

2.5. classification of image:

Image classification is where an image can be processed by a computer and the 'class' that the image falls under is defined. (Or the possibility that the image is part of a 'class'.) A class is simply a name, such as 'car',' animal',' house' and so on, for example. You input an image of a dog, for instance. Image classification is the computer's method of processing the image and telling you that it's a dog. The early classification of images relied on raw data for pixels. This meant that pictures would be broken down into individual pixels by machines. The problem is that it can look really different in two images of the same object. They may have numerous backgrounds, angles, poses, etcetera. This made it quite the challenge for machines to 'see' and categorize images correctly.

2.6. Accuracy assessment:

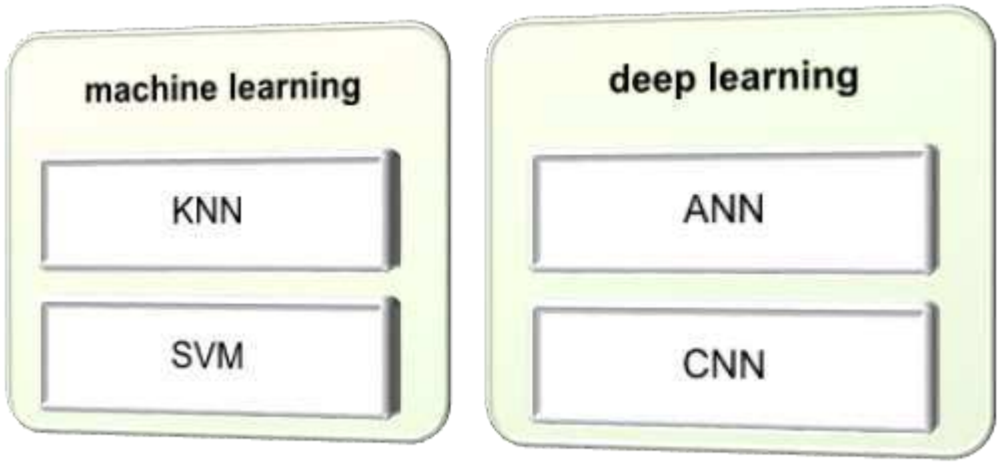
An important aspect of any classification project is accuracy assessment. It compares the classified image with another source of data that is known as true or ground-truth data. In the sector, ground truth can be obtained, but this is time consuming and costly. Ground truth data may also be extracted from high-resolution imagery interpretation or current classified imagery, the most common way to assess the accuracy of a classified map is to create a set of random points from the ground truth data and compare that to the classified data in a confusion matrix. Although this is a two-step process, you may need to compare the results of different classification methods or training sites, or you may not have ground truth data and are relying on the same imagery that you used to create the classification.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020

DOI: 10.5281/zenodo.4897990

the task of image classification is to take an image represented by array of pixels and give label to it the task of image classification is to take an image represented by array of pixels and give label to it. There are a lot of algorithms which can be used to classify the image.

3. Image Classification Algorithms:



3.1. k-nearest neighbors:

3.1.1.The basic idea of KNN algorithm.

It is one of the machine learning algorithms used to classify the image. It used to search the entire training set for the k number of the same cases, or neighbors, that clarifies the same patterns as the row with missing data. The average missing data variables were derived from kNN and used for each missing value. In the present study, k = 5, where the five closest individuals were used to refer to missing data; this has previously been shown to be appropriate. A disadvantage of kNN is that an appropriate number of values from the current data is necessary for each variable, which in the current study is five. It is a supervised learning algorithm that can be used for regression and classification problems. But it is always used for classification problems in machine learning. It has different names such as memory-based reasoning, Example-based reasoning, instance-Based Learning, and Lazy learning[7].

3.1.2.The working of the KNN algorithm.

KNN can be used for both classification problems and predictive regression. However, it is widely used in classification problems. It is very important because It easy for interpreting the output, Computation time of it, and it is the power of prediction. It uses a feature similarity to get the label of the new point. The new point will be assigned a value based on the closing of the match point in the training set. The algorithm goes through stages to predict the image classification.

Step 1: load the training set as well as the test set.

Step 2: choose the value of k, which is the closest data point. K can be any integer.

Step 3: For each new point, do the following:

- Calculate the distance between test data and each class of training data with the help of any of the methods: Euclidean, Manhattan, or chessboard distance. The most commonly used method for calculating distance is Euclidean.
- Sort the distance values in ascending order.
- Next, the upper K rows of the sorted array will be selected.
- Now, it will assign a category to the test point based on the most frequent category of those rows.

Step 4: End[8].

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

3.1.3.the main method for selecting the factor K.

First, let's try to understand exactly what influences K in the algorithm. It can be clarified by this example: The simple case to understand this algorithm. Here is a set of red circles (RC) and green squares (GS).

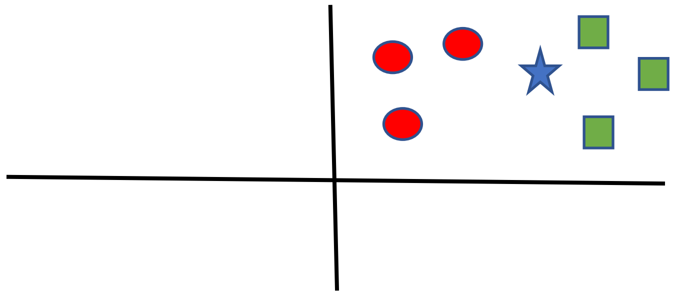


Fig 9. Example for selection the factor k

It will find the class of the blue star (BS). BS can be RC or GS and otherwise. "K" is the KNN algorithm that is the closest neighbor. Suppose K = 3. Hence, The label of a blue star is BS as the center of the large size to enclose only three data points on the plane. Refer to the following diagram for more details:

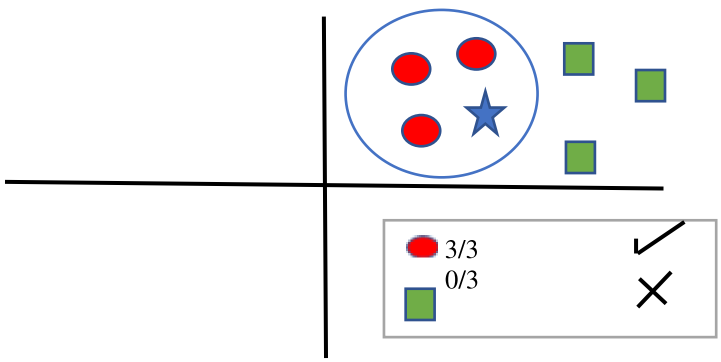


Fig 10. The class of blue star

The three closest points to the BS station are all RC. Hence, with a good level of confidence, that BS has to belong to the RC class. Here, the choice became so obvious as the three pings went from closest neighbors to RC. The selection of the K parameter is critical in this algorithm [9].

3.1.4.The Mathematics Behind KNN(equations):

$$Euclidean\ distance = \sqrt[2]{(x2-x1)^2+(y2-y1)^2} \quad (1)$$

$$Manhattan\ distance = |x2-x1|+|y2-y1| \quad (2)$$

$$chessboard\ distance = \max\left(|x2-x1|+|y2-y1|\right) \quad (3)$$

3.1.5.Advantages of KNN:

1-No period of training.

2-Since the KNN algorithm does not require any training before making predictions, new data can be added smoothly which will not affect the accuracy of the algorithm.

3-It is easy to implement, and simple.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

4-Its a powerful classification algorithm, effectiveness, intuitiveness, and competitive classification performance in many domains.
5-It is very effective when the training is large[10].

3.1.6.Disadvantages of KNN:

Despite the advantages given above, KNN has disadvantages such as :
1-it does not work with the large dataset
2-it does not work with the high dimensions.
3-It needs feature scaling.
5-Classifying unknown records is relatively expensive.
6-Computationally intense, especially as the training set size grows.

3.2. Support Vector Machine:

3.2.1.A brief overview of the SVM algorithm:

A Support Vector Machine (SVM) is a supervised machine learning model that uses classification algorithms for classification problems of two groups. After giving the SVM model sets of categorized training data for each category, they will be able to classify new image.it is powerful and well-used classification algorithms, which received a great deal of research attention before the deep learning we are currently in. Despite the fact that, to the viewer, they certainly have had great success in certain areas, and have remained among the most popular classification algorithms in the toolkit of machine learning practitioners and data scientists[11].

3.2.2.The working of the SVM algorithm:

The foundations of Support Vector Machines and how it works are best understood with a simple example. Suppose we have two tags: red and blue, and our data has two features: x and y. The main goal is to classify them, given a pair of coordinates (x, y), and the output if it's either red or _blue[12].

3.2.2.1- Linear data:

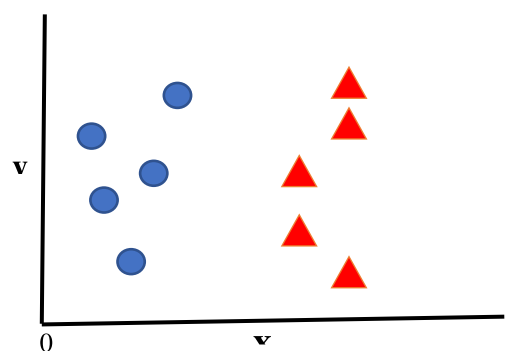


Fig 11.Our labeled data

The support vector machine takes these data points and outputs the hyperplane (which is just a line in two dimensions) that better separates the signs. This line is the decision boundary: Anything that falls on one side of it will classify it as blue, and anything that falls on the other side as red.

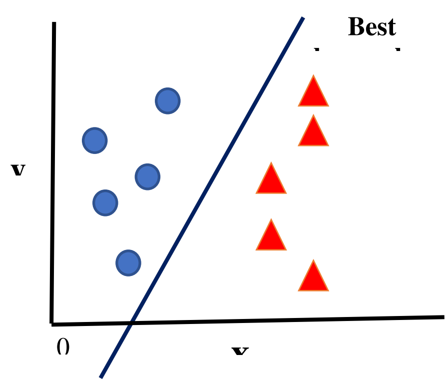


Fig 12.In 2D, the best hyperplane is

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

But, what exactly is the best hyperplane? For SVM, it's the one that increases the margins from both marks. In other words: the hyperplane (remember it's a line in this case) whose distance from the nearest element in each sign is greater.

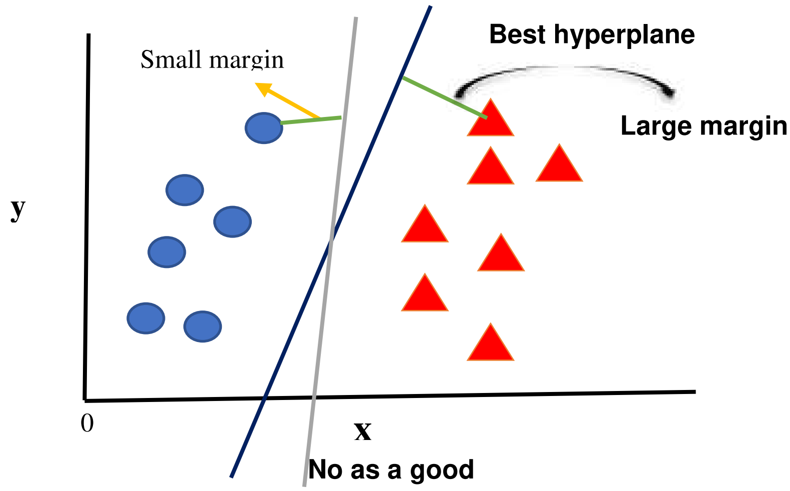


Fig 13.Not all hyperplanes are created equal

The formula of hyperplane:

$$\boldsymbol{ax + c = 0} \qquad \textbf{(4)}$$

But the maximum hyperplane:

$$x = c + \sum_{j=1}^{k} \alpha j * yj * a(j) - a \qquad \textbf{(5)}$$

where c and αj are learned parameters, k is the number of support vectors, j is a support vector instance, yj is the class value of a particular training instance of vector t, and a(j) is the vector of support vectors.

3.2.2.2-Non-linear data

The use of kernel trick in the non-linear data which it doesn’t separate by line. Note that the kernel trick is not actually part of SVM. It can be used with other linear classifiers such as logistic regression. Support vector machine is only concerned with finding resolution boundaries. It can be classified in this figure.

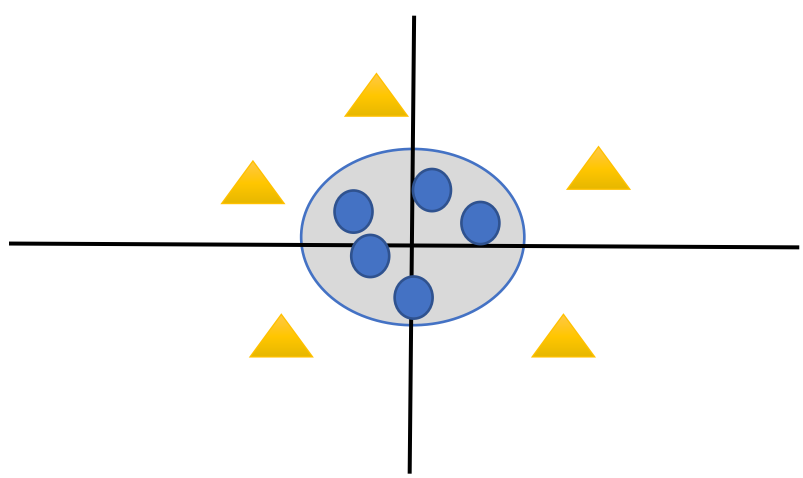


Fig 14.classification of non-linear data using kernel

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

-Examples of SVM Kernels:

- **Polynomial kernel:**

$$k(xi, xj) = (xi.xj + 1)^d \qquad (6)$$

where d is the degree of the polynomial.

- **Gaussian kernel:**

$$k(x, y) = exp(-\frac{||x - y||^2}{2\sigma^2}) \qquad (7)$$

- **Gaussian radial basis function (RBF):**

$$k(x, y) = exp\left(-\gamma||xi - xj||^2\right), \gamma > 0 \qquad (8)$$

Sometimes parametrized using:

$$\gamma = \frac{1}{2\sigma^2}$$

- **Linear splines kernel in one-dimension:**

$$k(x, y) = 1 + xy + xy * min(x, y) - \frac{x + y}{2} min(x, y)^2 + \frac{1}{3} min(x, y)^3 \qquad (9)$$

- **ANOVA radial basis kernel:**

$$k(x, y) = \sum_{k=1}^{n} exp(-\sigma(x^k - y^k)^2)^d \qquad (10)$$

3.2.3.Advantages of SVM:

1- Support vector machine is very efficient even with high dimensional data.
2-It is relatively memory efficient.
3-When the number of features is greater than the number of rows of data, SVM can perform in this case as well.
4-When the classes in the data are separating points well, SVM works well.
5-It can be used for both regression and classification problems. And last but not least, can do well with image data as well[13].

3.2.4.Disadvantages of SVM:

1-When classes in the data are points are not separated well, which means there are overlapping classes, SVM does not perform well.
2-it needs to detect an optimal kernel for SVM and this process is difficult.
3-SVM on huge training set comparatively takes more time to train.
4-The support vector machine or SVM is not a probabilistic model so it cannot be explained the classification in terms of probability.
5-It is difficult to understand and interpret the SVM compared to the decision tree as SVM is more complex.

3.3. Artificial neural network:

3.3.1.A brief overview of the SVM algorithm.

An artificial neural network (ANN) is the part of a computing system designed to simulate the human brain analyzes and processes information. It is the basis of artificial intelligence (AI) and solves problems that may be impossible or difficult for human or statistical standards. ANNs have self-learning capabilities that enable them to achieve better results as more data becomes available. It has a hundred or thousands of neurons called

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

processing unit, which are connected by nodes. These processing units consist of input and output units. Input units receive multiple forms and structures of information based on an internal weighing system, and the neural network tries to identify the information provided to produce a single output report. Just as humans need rules and guidelines to reach a result or outcome, humans also use a set of learning rules, called backpropagation, In order to achieve the desired results[14].

3.3.2. ANN architecture.

An artificial neural network model that can be identified by three sides[15]:

3.3.2.1-Interconnections.

Interconnection can be defined as the way in which processing elements (neurons) in artificial neural networks connect to each other. Hence, the arrangements of these processing elements and the geometry of interconnections are very important in ANN. These arrangements have two common layers for all network architectures, the Input layer and the output layer where the input layer stores the input signal and the output layer produces the output of the network. The third layer is the Hidden layer, in which neurons are neither saved in the input layer nor in the output layer. These neurons are hidden from people interacting with the system and act as a black box to them. When increasing the hidden layers of neurons, the computing and processing capacity of the system can be increased, but the training phenomenon of the system becomes more complex at the same time. There exist five fundamental types of neuron connection architecture :

1) Single-layer feed forward network.

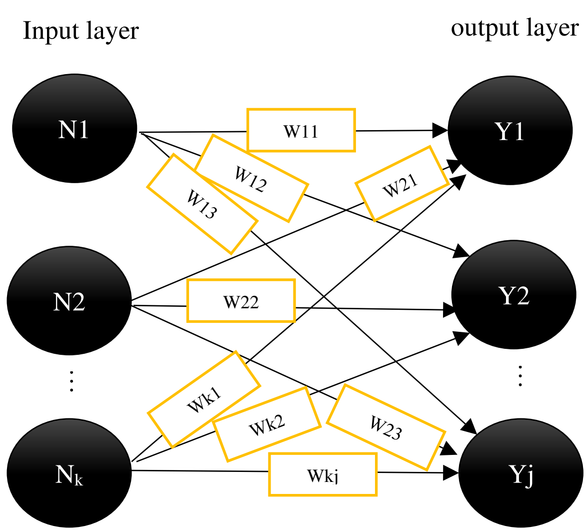


Fig 15.Single-layer feed forward network

In this type of network, it has two input layers and an output layer but the input layer is not counted because no calculation is done in this layer. The output layer is formed when different weights are applied to the input nodes and the cumulative effect is taken for each node. Then, the neurons collectively give the output layer to count the output signals.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

2) Multilayer feed forward network.

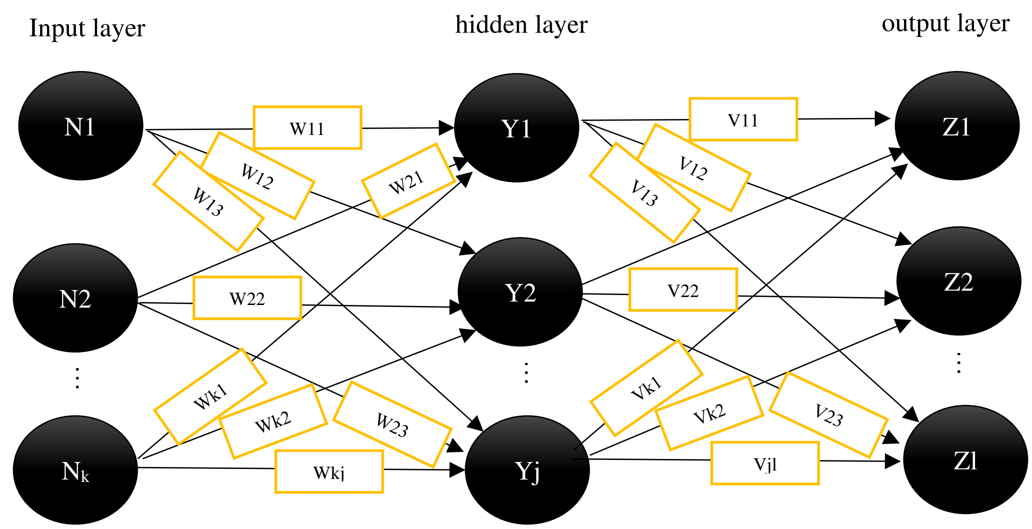


Fig 16. multilayer feed forward network

This layer also contains a hidden inner layer of the network and has no direct connection to the outer layer. Having one or more hidden layers allows the network to be mathematically stronger, and the intermediate calculations used to detect the output Z. There are no feedback connections in which model outputs are fed back into itself.

3) Single node with its own feedback.

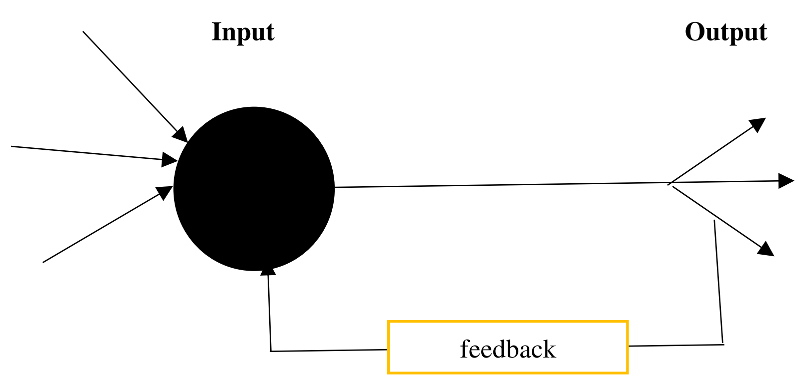


***Fig 17.* Single node with its own feedback.**

When outputs can be redirected as inputs to the corresponding layer or nodes prior to the layer, then it results in feedback networks. Recurrent networks are closed-loop feedback networks. The figure above shows a single recurrent network having a single neuron with feedback to itself.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

4) Single-layer recurrent network.

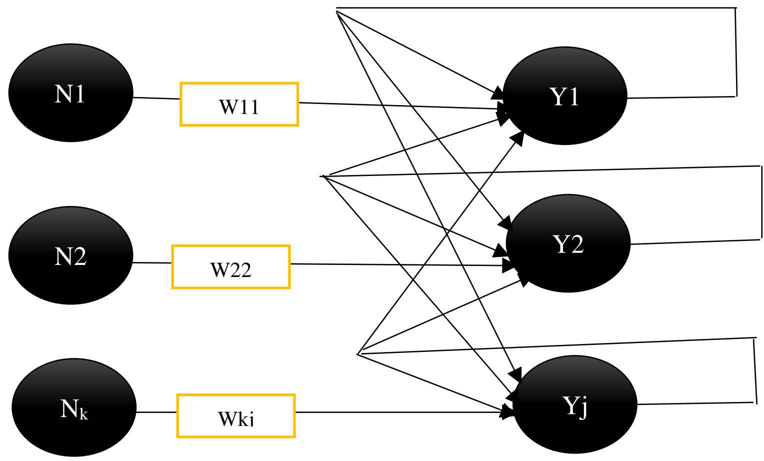


Fig 18. **Single-layer recurrent network.**

The above network is a single-layer network with a feedback connection in which the outputs of processing items can be directed back to itself or to other processing items or both. it is a class of ANN in which the connections between nodes form a directed graph along a chain. This allows it to exhibit dynamic temporal behavior for a time series. Unlike feed-forward neural networks, RNNs can use their internal state (memory) to process the input series.

5) Multilayer recurrent network.

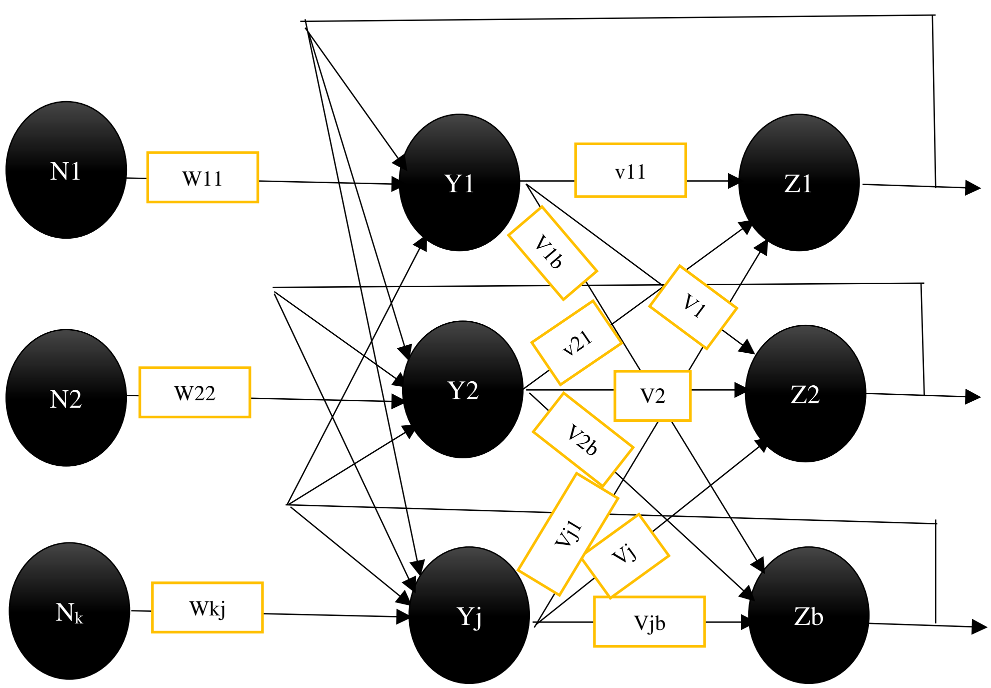


*Fig 19.***Multilayer recurrent network.**

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

In this type of network, the output of the processing element can be directed to the processing element in the same layer and in the previous layer which forms a recurrent multi-layer network. They perform the similar task for each element of a series, with the output being based on the previous computations. Inputs are not required at every time step. The main feature of the recurrent neural network is its hidden state, which captures some information about the sequence.

3.3.2.2. Activation functions.

The activation function is a mathematical "gateway" between the inputs feeding the current neuron and its outputs going to the next layer. It can be as simple as a step function that turns the neuron output on and off, depending on a base or a threshold. Or it could be a switch that maps the input signals into the output signals needed for the neural network to function.

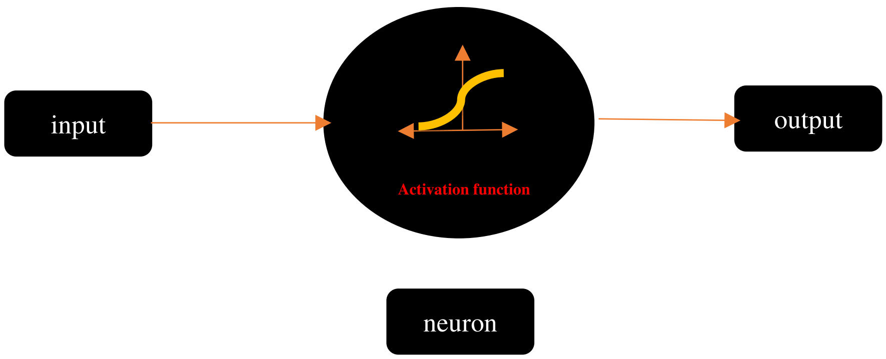


Fig 20. **Activation functions.**

-three types of Activation Functions:

1- Binary Step Function

The binary step function is a threshold-based activation function. If the input value is higher or lower than a certain threshold, the neuron is activated and the exact same signal is sent to the next layer and we can show it in blow figure.

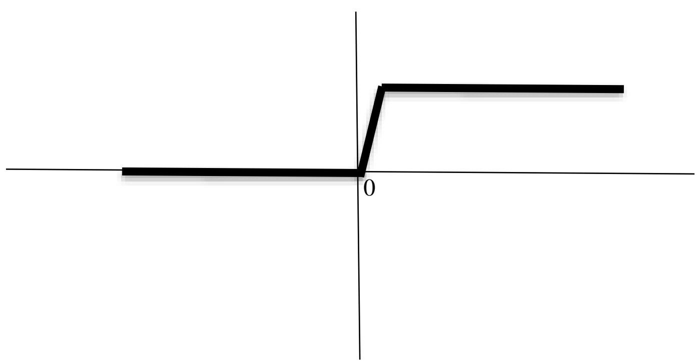


Fig 21. **Binary Step Function.**

$$f(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 1 \end{cases} \quad \textbf{(11)}$$

Properties:

1-Range - 0 to 1.

2-Monotonicity - provides a convex error surface so optimization can be produced faster.

3-Derivative - is 0 when $x \neq 0$, and undefined when $x = 0$.

4-Discontinuous.

Limits:

1-it does not responsible for inverse signals (i.e. negatives).

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

2- it does not allow multi-valued outputs - for example, it cannot support the classification of entries into one of several classes.

2-Linear Activation Function

It takes the input, multiplied by the weights of each neuron, and creates an output signal proportional to the input. In a sense, a linear function is better than a step function because it allows multiple outputs, not just a yes and no.
A linear activation function takes the form:

$$V=cX \quad \textbf{(12)}$$

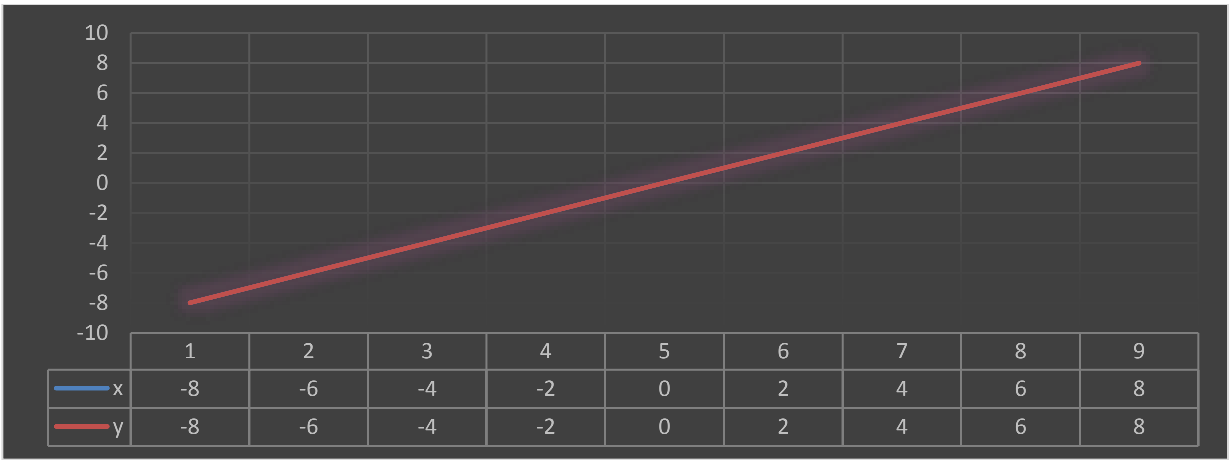


Fig 22. **Linear Activation Function.**

However, the linear activation function has two main problems:
1-Can't use backpropagation.
2-All layers of the neural network collapse into one layer. A neural network with a linear activation function is simply a linear regression model. It has limited power and the ability to handle complex variables of input data.

3- Non-Linear Activation Functions

Modern neural network models use non-linear activation functions. It allows the model to create complex mappings between network inputs and outputs, which are essential for learning and modeling complex data, such as images, video, audio, and non-linear or high-dimensional data sets. Almost any process imaginable can be represented as a functional computation in a neural network, provided the activation function is non-linear.
Non-linear functions address problems of linear activation function:
1-They allow backpropagation because they have a derivative function which is linked to the inputs.
2-It allows multiple layers of neurons to be 'stacked' to create a deep neural network. Multiple layers of hidden neurons are needed to learn complex data sets with high levels of accuracy.

seven Common Nonlinear Activation Functions and How to Choose an Activation Function:

- Sigmoid / Logistic.

$$\boldsymbol{\sigma(x) = \frac{1}{1+e^{-x}}} \qquad \textbf{(13)}$$

- TanH / Hyperbolic Tangent

$$\boldsymbol{\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^{x}\, e^{-x}}{e^{x}+e^{-x}}} \qquad \textbf{(14)}$$

- ReLU (Rectified Linear Unit)

$$\mathbf{y\ =\ max(x, 0)} \qquad \textbf{(15)}$$

the function equlal x if x>=0 and 0 other wise.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

- Leaky ReLU
  f(x)=1(x<0)(αx)+1(x>=0)(x) where α is a small constant. **(16)**

- Parametric ReLU

$$\boldsymbol{f(y) = y \ if \ y \geq 0} \quad \textbf{(17)}$$
$$\boldsymbol{f(y) = ay \ if \ y \leq 0}$$

- SoftMax

$$\boldsymbol{s(xi) = \frac{e^{xi}}{\sum_{j=1}^{n} e^{xj}}} \quad \textbf{(18)}$$

- Swish

$$\boldsymbol{\sigma(x) = \frac{x}{1 + e^{-x}}} \quad \textbf{(19)}$$

3.3.3-Backpropagation Networks

As in Perceptron, this training algorithm involves 2 passes:

The forward pass – outputs of different layers are calculated.

The backward pass – weight corrections are calcuated.

Consider a simple 3-layer network with a single neuron in every layer.

the steps of Backpropagation:

Step – 1: Forward Propagation.

$$\text{net } h_i = \sum wi * x + b \quad \textbf{(20)}$$

Step – 2: Backward Propagation .

$$error = \sum \frac{1}{2}(target - output)^2 \quad \textbf{(21)}$$

$$\theta = (1 - \alpha\lambda)\theta - \alpha \frac{1}{b} \sum_{k=i}^{i+b-1} \frac{\partial E}{\partial \theta}(x(k), y(k), \theta) \quad \textbf{(22)}$$

Step – 3: set all the values together and computing the updated weight value.

3.3.4.Advantages of ANN

1-it Stores information on the entire network.

2-It can work with incomplete knowledge.

3-It can have a distributed memory.

4-It can make machine learning because it learns events and can make decisions by commenting on similar events.

6-It can work parallel: it has a numerical strength that can perform more than one task at the same time[16].

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

3.3.5.Disadvantages of ANN

1-it needs processors with parallel processing power.

2-Unjustified behavior of the network.

3-There is no specific rule to determine the structure.

4-It is very difficult to show the problem of the network.

5-It will take much time if the network is unknown.

3.4. Convoution neural network :

3.4.1. A brief overview of the CNN algorithm.

A traditional neural network (ConvNet/CNN) is a deep learning algorithm that can take the image of inputs, identify the importance (weights and bias) of different aspects/objects in the image, and be able to distinguish between them and the other. The pre-processing required in a convnet is much lower than other classification algorithms. While in primitive methods the filters are manually designed, with adequate training, convection grids have the ability to learn these filters/properties. It designed for image classification.it also has an excellent capacity In sequent data analysis such as NLP.it contains two important operations, namely convolution, and pooling. the convolution operation used to extract features from images(dataset). the pooling operation used to reduce the dimensionality of features extracted from the convolution operation. maximum and average pooling are common Operations used in CNNs. it used the activation function is called relu to transfer the gradient in training using backpropagation[17].

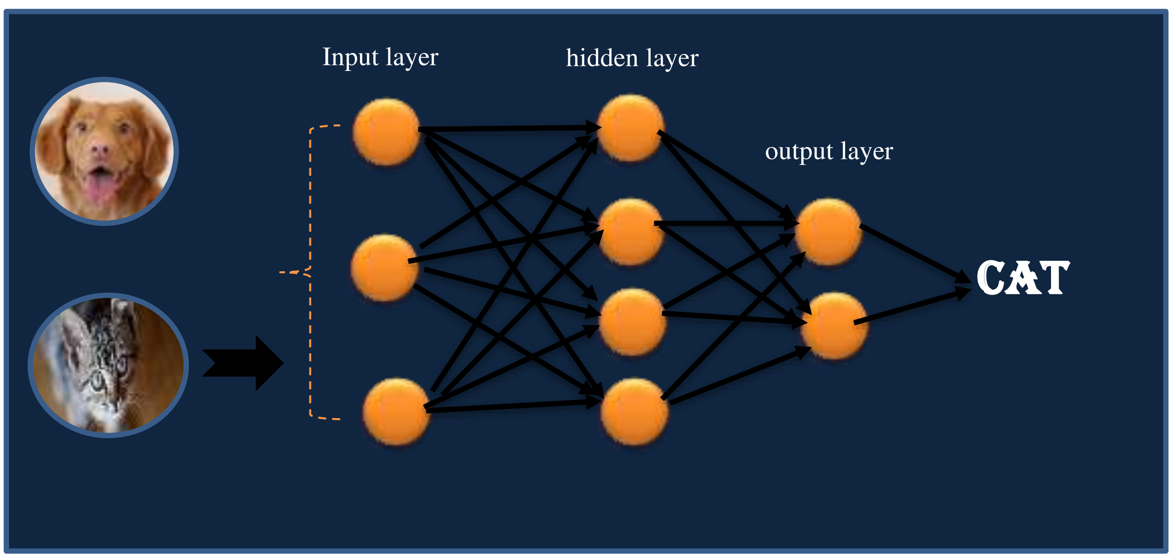


Fig 23. **A classic CNN classifying between a dog and a cat.**

3.4.2. Convolutional neural network–based methods.

it is a class of DNNs in deep learning that is used in computer vision and Natural language processing studies.it is similar to neurons connectivity patterns in human brains, and it is the regular version of multilayer perceptron found in fully connected networks. Precisely, CNNs made up of one input layer, multi hidden layer, and output layer. the hidden layer contains the Convolution layer, relu (activation function),

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

pooling layers, fully connected layers, and normalization layers.it is the best algorithm compared with other algorithms which used to image classification because it requires much less preprocessing and can do better results with as the number of training increase. In NLP, CNN is used to identify predictive features from large structures and produce vectors that represent this structure. CNN based method applies a 1-D convolution operation to reach the purpose of searching the necessary information for local word order. As shown in fig.24, Vocabulary as extracted features after grouping is the vocabulary present in the original sentence with correct word order.

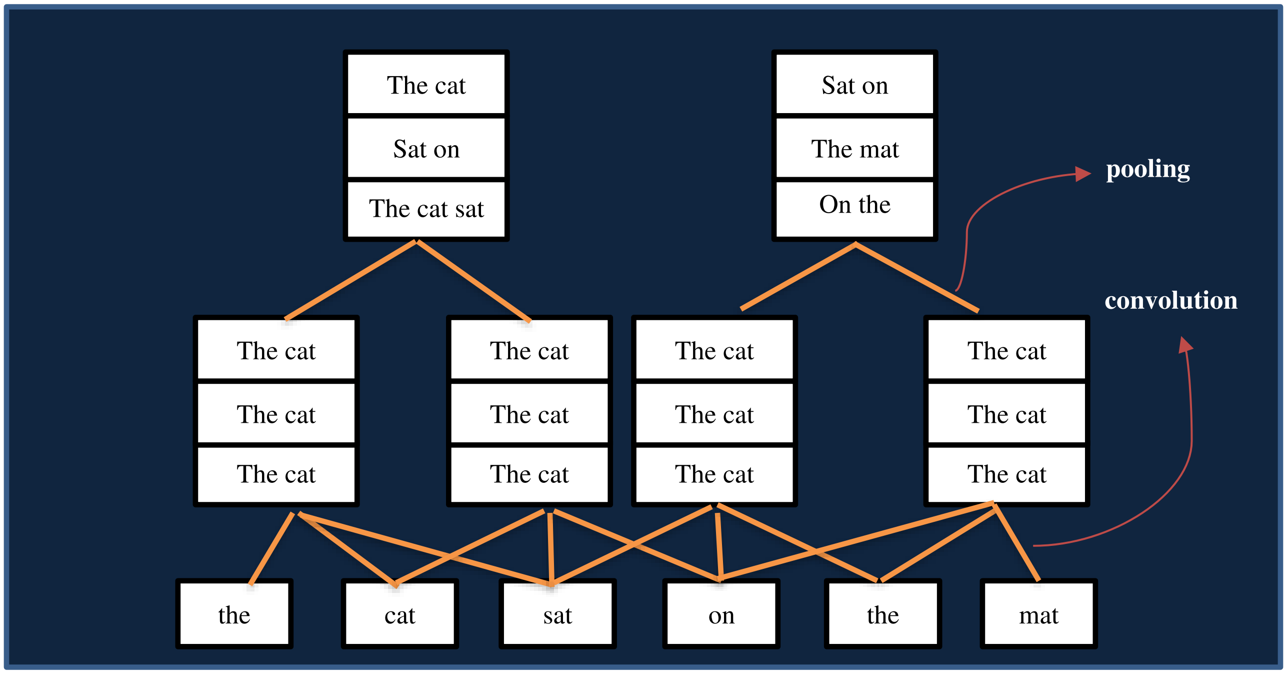


Figure 24. Example of CNN-based representation learning methods.

3.4.3. CNN architecture:

As mentioned earlier, CNN is mainly focused on the premise that the input will include images. This focuses on the architecture that will be created the way it fits the need to deal with a specific type of data. One of the differences between ANNs and CNNs is that the layers within the CNN are comprised of neurons organized into three dimensions (height, width, and depth). the third dimension refers to activation volume[18].

3.4.3.1. Overall architecture:

CNNs are consisted of three types of layers. these are convolution layer, pooling layer, and fully connected layer (relu , and output).these are stacked in CNNs which shown in fig.25

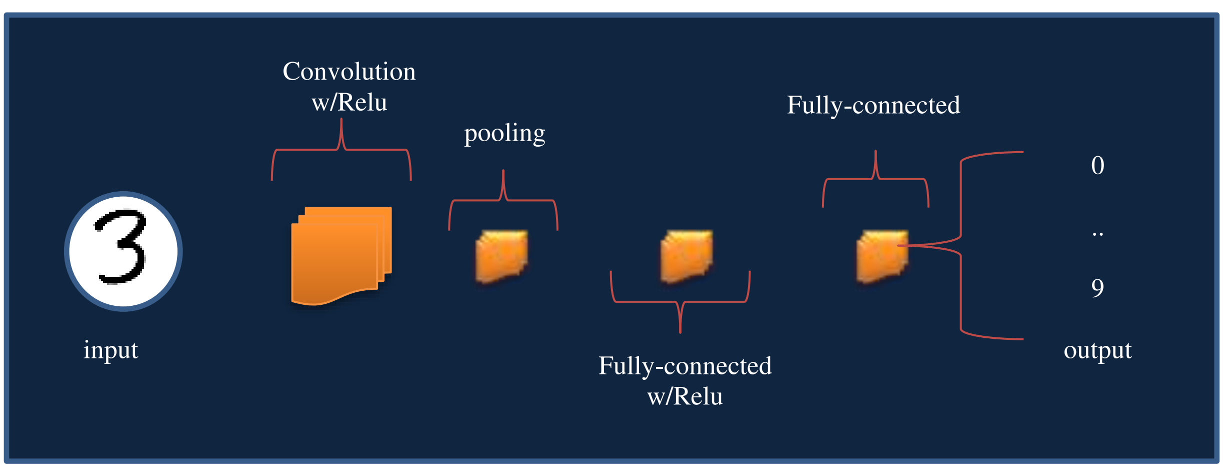


Fig 25. An simple CNN architecture, comprised of just five layers

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

The basic functionality of the CNN example above can be divided into Four main areas.

1. **The input layer** holds a set of pixels which represent the image.
2. **The convolutional layer** will determine the output of which neurons are Related to local areas of input through numerical calculation the product is between their weights and the area related to the input size. Linear Corrected Module (usually abbreviated to ReLu) is intended for introduction the "elementwise" activation function is like the sigmoid for an output activation caused by the previous layer.
3. **The pooling layer** will then simply perform the down sampling along the spatial space After the specified input, which reduces the number of parameters
   Within this activation.

The fully connected layers produce the class score from the activation function, to be used in classification. It suggests that relu can be used between these layers and improve the performance.

3.4.3.2. Convolution layer:

it plays a vital role in how CNNs operate. the Layer parameters focus on using the learnable kernel. These kernels are small in most special dimensions, but prevalent along the entirety of the depth of the input. When the data reaches the convolutional layer, the layer wraps each filter across the spatial dimensions of the input Produce a 2D activation map. These activation maps can be visualized.
the standard product is calculated for each value in That kernel. (Fig. 26) From this the network will learn which kernels "fire" when They see a specific feature of a particular input spatial position. here they are Known as activations [19].

Input vector

0	0	0	0	0	0
0	1	2	1	1	2
0	1	1	1	1	1
1	0	0	0	0	0
0	0	1	1	1	0
0	1	1	1	1	1

pooled vector

0	0	0
0	1	2
0	1	1

kernel

4	0	0
0	0	0
0	0	-4

Destination pixel

	-8	

Figure 26.A visual representation of a convolutional layer. The center component in The kernel is placed over the input vector, which is then calculated and replaced By a weighted sum of itself and any nearby pixels.

Each kernel will have a corresponding activation map, which will be stacked.Along the depth dimension to form all output volume of the convolutions layer.
For example, if the input to the network is an RGB colored image with a dimensionality of 64 x 64, and we put the receptive field size as 6 x 6, we would have a total of 108 weights on every neuron in the convolutional layer. (6 x 6 x 3 where 3 is the volume of connectivity across the depth of the volume) To set this into perspective, a basis neuron seen in other forms of ANN would contain 12, 288 weights each.
The Convolutional layers are used to reduce the complexity of the model through the optimization of its output. These are improved through three hyperparameters, the depth, the stride, and the setting zero-padding.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

The depth of output volume produced by the convolutional layer can be set manually over the number of neurons to the same region of the input. This can be seen in ANN form, where all neurons are connected directly in a hidden layer to every single neuron. It can reduce the total number of neurons of the network, but it can also be greatly reduced the Pattern recognition capabilities of the model.

the stride in which we put the depth over the spatial dimensionality of the to place the receptive field. For example, if we were to put a stride as 1, then we would have a heavily overlapped receptive field producing great activations. Instead of that, putting the stride to a greater number will reduce the amount of overlapping and produce an output of lower spatial dimensions.

Zero-padding is the simple operation of padding the border of the input and is an effective way to give further control as to the dimensionality of the output volumes. It is important to understand that through using these methods, we will change the spatial dimensionality of the output of the convolutional layer. you can use of the following formula to calculate this:

$$\frac{(I - F) + 2P}{S + 1}$$

I → refers to the input volume size (height x weight x depth).
F→ refers to the respective field size.
P→ refers to the amount of zero padding set
S→ refers to the stride.

If the result from this equation is not equal to an all integer then the stride has been the wrong set, as the neurons will be unable to map neatly across the given input.
Parameter sharing works on the proposition, that if one region feature is important to compute at a set spatial region, then it can be useful in another region. If we coerce every individual activation map within the output to the same weights and bias, then we will see a huge reduction in the number of parameters being produced by the convolutional layer.
All neurons within a feature map have weights that are fixed to be equal; however, different feature maps within the same convolutional layer have different weights so that several features can be extracted at every location. More formally, the kth output feature map Tv can be computed as

$$\mathbf{Tv}=f\,(Lv * a)$$

where the input image is denoted by a; the convolutional filter related to the Vth feature map is denoted by Lv.

3.4.3.3. Pooling / subsampling layer

Pooling layers used to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model. It makes the features strong against noise and distortion. There are two methods to do pooling: max pooling and average pooling. In two cases, the input is divided into two-dimensional spaces. For example, in Figure 1, layer 2 is the pooling layer. every input feature is 28x28 and is divided into 14x14 regions of size 2x2. For average pooling, calculate the average value of four values in the region. For max pooling, detect the maximum value of four values.
If the input is of size 4x4, it can be divided into four non-overlapping matrices of size 2 x 2. In the max pooling, the output is the max value of four in the 2 x 2 matrix. In the case of average pooling, the output is the average value of four in the 2 x 2 matrix. Please note that for the output with index (2,2), the result of averaging is a part that has been rounded to the nearest integer.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

21	8	8	12
12	19	9	7
8	10	4	3
18	12	9	10

15	9
12	7

21	12
18	10

Figure 27.Pictorial representation of max pooling and average pooling

Formally, max pooling detects the largest element within every receptive field such that:

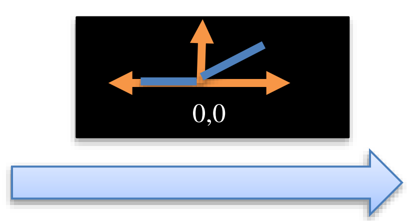
$$\mathbf{Y_{vug}=max_{\,(p,q)\in Rug}\,(a_{vpq})}$$

where the output of the pooling operation, associated with the Vth feature map, is denoted by Yvij, avpq denotes the element at location (p, q) contained by the pooling region Rug, which embodies a receptive field around the position (u, g). Fig.27 clarifies the difference between max pooling and average pooling.

3.4.3.3.1. Relu: A Relu performs the function y = max(x,0), so the size of input and output in this layer are similar. It increases the nonlinear features of the decision function and of every network without assuming the receptive fields of the convolution layer. In different to other non-linear function used in CNNs such as hyperbolic tangent, absolute of hyperbolic tangent, and sigmoid, the advantage of a ReLU is that the network trains many times faster. Relu functionality is evident in Figure 28, with its transfer function plotted above the arrow[20].

Transfer Function

15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23



15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Figure 28. Pictorial representation of ReLU functionality

3.4.3.3.2. Continuous trigger (non-linear) function:
The non-linear layer implements element by element in every feature. A continuous non-linear function can be a hyperbolic tangent, absolute of hyperbolic tangent, or sigmoid.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

A continuous non-linear function:

1-Sigmoid:

$$\boldsymbol{f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}}$$

2- hyperbolic tangent

$$\boldsymbol{\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^{x}\ e^{-x}}{e^{x} + e^{-x}} = \frac{e^{2x}\ 1}{e^{2x} + 1}}$$

or

$$\boldsymbol{f(x) = tanh(x) = 2\sigma(2x) - 1}$$

3.4.3.4. Fully connected layer:

Fully connected layers final layers of a CNN. These layers calculate the total weight of the past layer of features, referencing the precise mix of "ingredients" to detect a specific target result. In a fully connected layer, all the elements of all the features of the previous layer can be used in the calculation of every element of each output feature. Fig.13 clarify the fully connected layer L. Layer L-1 has two features, each of which is 2x2, i.e., which has four elements. Layer L has two features, every feature having a single element.

3.5 Advantages of Convolutional Network Architecture:

I. Reducing the calculation compared to a normal neural network.

II. In terms of performance, CNNs outperform NNs for traditional image recognition tasks and many other tasks.

III. Significantly simplify calculation in convolution process without losing the essence of the data.

IV. It is great in image classification

V. Another basic feature of CNNs is weight sharing. Let's take an example to explain this. you have a one layered CNN with ten filters of size 6x6. Now you can simply calculate parameters of such a CNN, it would be 6*6*10 weights and 10 biases i.e. 6* 6*10 + 10 = 370 parameters. Now let's take a simple one layered NN with 360 neurons, here the number of weight parameters based on the size of images is '360 x T' where the size of the image is Q X S and T = (Q *S). Additionally, you need 'Q' biases. For the MNIST data as input to such a NN, we will have (360*784+1 = 28225) parameters. Clearly, CNN is efficient in memory and complexity. Suppose that NNs and CNNs with millions of neurons, then CNNs could be reduced complex and saves memory compared to the NN [21].

3.6. Disadvantages of Convolutional Network Architecture

I. CNN's' weakness is the amount of data it provides them. If you supply them for less, expect CNNs to do poorly. CNN's contain millions of parameters and with a small dataset, they will run into a suitable problem as they need a massive amount of data to quench the thirst. So, you give out a lot of data, CNNs are stronger and more willing to give you better performance, you are giving fewer data.

II. Overfitting in CNN's is the biggest problem. when we change the camera and the lighting, most CNNs cannot form well.

III. The dependence of the CNNs on initial parameter tuning (to get a good point) was to avoid local Optima. Thus, the weakness of CNNs is the high volume of work they need to configure according to the problem at hand. This may require some specialist knowledge in the field

IV. It is Non-expressive logic and learning.

V. The computationally is expensive

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020

DOI: 10.5281/zenodo.4897990

Practical side

We have trained our classifier "model" on benchmark dataset cifar10 python, it's around 163 GB and it's consists of 60000 32x32 color images in 10 classes, with 6000 images per class (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). There are 50000 training images and 10000 test images as mentioned. You can see and download CIFAR dataset from this link https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz

```
#download the library
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense , Flatten , Conv2D ,MaxPooling2D,Dropout
from tensorflow.keras import layers
from keras.utils import to_categorical
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

```
#load the training data
 ===>There are 50000 training images and 10000 test images.
from keras.datasets import cifar10
(xTrain,yTrain),(xTest,yTest)  =  cifar10.load_data()
#get the shape of the arrays
print ('xTrain shape : ',xTrain.shape)
print ('yTrain shape : ',yTrain.shape)
print ('xTest shape : ',xTest.shape)
print ('yTest shape : ',yTest.shape)
```

```
# take a look at the first image as an array
Index = 1
xTrain [index]
#show the image as a picture
image = plt.imshow( xTrain[index] )
# get the image label
print ('the target of image is : ',yTrain[index])
#get the image classification
classification=['airplane','autobile','bird','cat','deer','dog','frog','hourse','ship','truck']
```


Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

```
#print the classification of image
print ('the image class is : ' , classification [ yTrain [index] [0] ])
#convert the labels into a set of 10 numbers to input into the neural network(NN)
y_train_epoch=to_categorical(yTrain)
y_test_epoch=to_categorical(yTest)
#print the new labels
print(y_train_epoch)
#print the new label of the image/picture above
print ('the right postion of this label is : ', y_train_epoch[index])
```

```
#Normalize the pixel to the values between 0 and 1
x_train = xTrain / 255
x_test = xTest / 255
#display x_train Now
print ( 'reducing value of training' , x_train)
```

```
#Create the classifier architecture
classifier = Sequential()
#Add the first layer
classifier.add( Conv2D( 32,(5,5) ,activation='relu' , input_shape =(32,32,3) ))
#add a pooling layer
classifier.add( MaxPooling2D(pool_size=(2,2)) )
#Add another convolution layer
classifier.add( Conv2D(32,(5,5),activation='relu') )
#add another pooling layer
classifier.add( MaxPooling2D(pool_size=(2,2)) )
#Add a flattening layer
classifier.add( Flatten() )
```

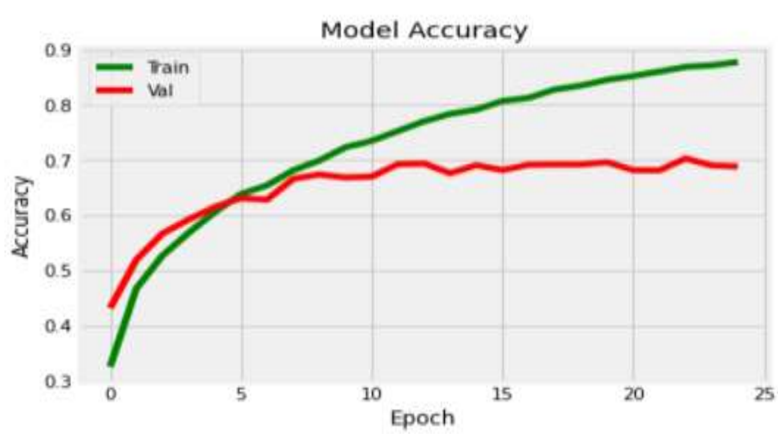
```
#Add a layer with 1000 neurons
classifier.add( Dense(1000,activation='relu') )
#Add drop out layer
classifier.add( Dropout(0.5) )
#Add a layer with 500 neurons
classifier.add( Dense(500,activation='relu') )
#Add drop out layer
classifier.add( Dropout(0.5) )
#Add a layer with 250 neurons
classifier.add( Dense(250,activation='relu') )
#Add a layer with 10 neurons
classifier.add( Dense(10,activation='softmax') )
```


Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

```
#Evalution stage
#Evalute the classifier using the test_data set
print ( 'Score :' , np.round( classifier.evaluate( x_test,y_test_epoch ) [1] * 100 , 2) ,'%')
```

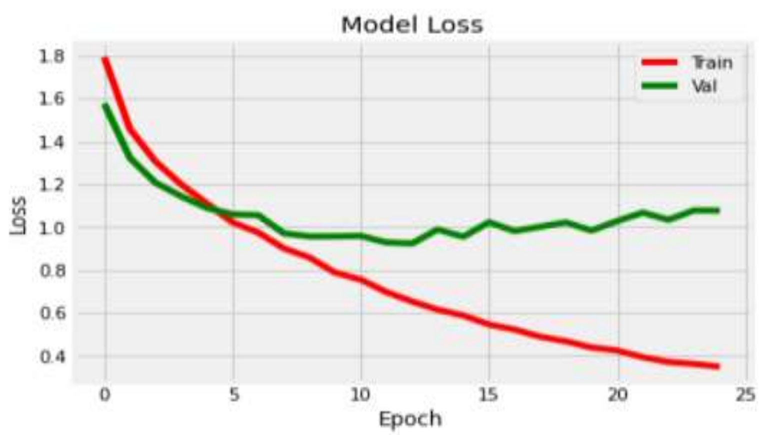
```
#compile the model
classifier.compile( loss = 'categorical_crossentropy' ,
                    optimizer = 'adam' ,
                    metrics = [ 'accuracy' ]
                  )
#training the classifier
# We have splited data into 80% for training and another 20% for testing
hist=classifier.fit( x_train,y_train_epoch ,
                     batch_size=256 ,
                     epochs=25 ,
                     validation_split=0.2
                   )
```

```
#Visualize the models accuracy
plt.plot( hist.history[ 'accuracy' ] , color= 'g' )
plt.plot( hist.history[ 'val_accuracy' ]  , color= 'r' )
plt.title( 'Model Accuracy' )
plt.ylabel( 'Accuracy' )
plt.xlabel( 'Epoch' )
plt.legend( [ 'Train' , 'Val' ] , loc= 'upper left' )
plt.show()
```

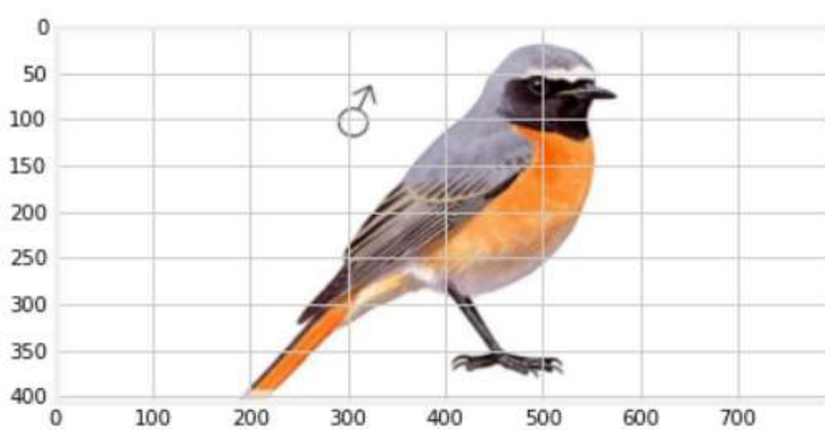


Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

```
#show the image
new_image = plt.imread( 'sample_data/as.jpg' )
img = plt.imshow( new_image )
#Visualize the models loss
plt.plot( hist.history['loss'] , color= 'red' )
plt.plot( hist.history['val_loss'] , color= 'g')
plt.title( 'Model Loss' )
plt.ylabel( 'Loss' )
plt.xlabel( 'Epoch' )
plt.legend( ['Train' , 'Val'] , loc= 'upper right' )
plt.show()
```

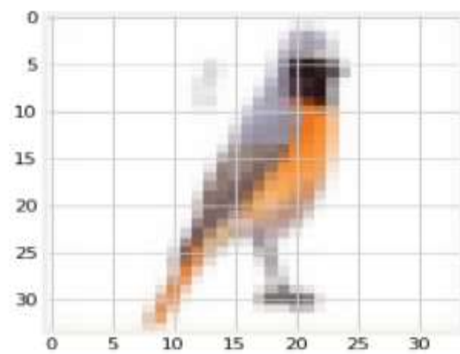



```
#show the image
new_image = plt.imread( 'sample_data/as.jpg' )
img = plt.imshow( new_image )
```



Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

```
#Resize the image
from skimage.transform import resize
resized_image = resize( new_image ,(34,34,3) )
im = plt.imshow( resized_image )
```




```
#get the classifier predictions
predictions = classifier.predict( np.array([resized_image]) )
#Sort the Predictions from least to greatest
list_index = [0,1,2,3,4,5,6,7,8,9]
x = predictions
for i in range (10):
    for j in range(10):
        if x[0] [ list_index[ i ] ]:
            temp = list_index[ i ]
            list_index[ i ] = list_index[ i ]
            list_index[ j ] = temp

#show sorted labels in order
print (list_index)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 0, 1]
```

```
#printing the image prediction
for i in range(10):
  if (round(predictions[0][list_index][i]*100,2)>50):
     print ( classification[ list_index[ i ] ] , ':' , round( predictions[0] [ list_index ] [ i ]*100,2) , '%' )
```

```
bird : 99.82 %
```


Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990

References

[1] D. Lu & Q. Weng (2007) A survey of image classification methods and techniques for improving classification performance, International Journal of Remote Sensing, 28:5,823-870, DOI: 10.1080/01431160600746456

[2] R, Ponnusamy & Sathiamoorthy, S. & Kaliyamoorthi, Manikandan. (2020). A Review of Image Classification Approaches and Techniques. 10.23883/IJRTER.2017.3033.XTS7Z.

[3] International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 1, January 2013

[4] UKEssays. (November 2018). Supervised Image Classification Techniques. Retrieved from https://www.ukessays.com/essays/engineering/supervised-image-classification-9746.php?vref=1

[5] Pal, Kuntal & Sudeep, K.. (2016). Preprocessing for image classification by convolutional neural networks. 1778-1781. 10.1109/RTEICT.2016.7808140.

[7] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.

[8] K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint

[9] A Survey of kNN Algorithm Jingwen Sun,Weixing Du,Niancai ShiInformation Engineering College, Panzhihua University of Technology, Sichuan, China.

[10] Handayani, Irma. "Application of K-Nearest Neighbor Algorithm on Classification of Disk Hernia and Spondylolisthesis in Vertebral Column." Indonesian Journal of Information Systems, vol. 2, no. 1, Aug. 2019, p. 57. DOI.org (Crossref), doi:10.24002/ijis.v2i1.2352.

[11] The research of the fast SVM classifier method - IEEE Conference Publication, Yujun Yang; Jianping Li; Yimei Yang, 18-20 Dec. 2015.

[12] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[13] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan and M. j. Rajabi, "Advantage and drawback of support vector machine functionality," 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, 2014, pp. 63-65, doi: 10.1109/I4CT.2014.6914146.

[14] A Review paper on Artificial Neural Network: A Prediction Technique Mitali S Mhatre1, Dr.Fauzia Siddiqui2, Mugdha Dongre3, Paramjit Thakur4

[15] Khalil, Kasem & Eddash, Omar & Kumar, Ashok & Bayoumi, Magdy. (2018). An Efficient Approach for Neural Network Architecture. 745-748. 10.1109/ICECS.2018.8617887.

[16] Suka, Machi & Oeda, Shinichi & Ichimura, Takumi & Yoshida, Katsumi & Takezawa, Jun. (2007). Advantages and Disadvantages of Neural Networks for Predicting Clinical Outcomes.. IMECS 2007: International Multiconference of engineers and computer scientists. 839-844.

[17] Sakib, Shadman & Ahmed, & Jawad, Ahmed & Kabir, Jawad & Ahmed, Hridon. (2018). An Overview of Convolutional Neural Network: Its Architecture and Applications. 10.20944/preprints201811.0546.v1.

[18] Sun, Yanan & Xue, Bing & Zhang, Mengjie & Yen, Gary. (2019). Completely Automated CNN Architecture Design Based on Blocks. IEEE Transactions on Neural Networks and Learning Systems. PP. 1-13. 10.1109/TNNLS.2019.2919608.

[19] Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach Author links open overlay panelSakshiIndoliaaAnil KumarGoswamibS.P.MishrabPoojaAsopaa

[20] An Introduction to Convolutional Neural Networks,Keiron O'Shea1 and Ryan Nash2

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020

DOI: 10.5281/zenodo.4897990

[21] Kamilaris, Andreas & Prenafeta Boldú, Francesc. (2018). A review of the use of convolutional neural networks in agriculture. The Journal of Agricultural Science. 156. 1-11. 10.1017/S0021859618000436.

Received: December 26, 2020 Revised: March 10, 2021 Accepted: April 1, 2020
DOI: 10.5281/zenodo.4897990