Methodology

One kind of Deep Learning neural network architecture that is frequently utilized in computer vision is the convolutional neural network (CNN). The branch of artificial intelligence known as "Computer Vision" gives computers the ability to comprehend and analyze images and other visual input.

1. Convolutional Layer

The convolutional layer, which produces the majority of the network's computations, is the fundamental layer used to build convolutional neural networks. Keep in mind that the number of parameters does not equal the amount of calculation. Compared to a fully connected network of the same size, the convolution operation can effectively reduce the training complexity of the network model as well as the network connection and parameter weights. Standard convolution, transposed convolution, hole convolution, and depth separable convolution are examples of common convolution operations.

1. Activation Layer

Artificial neural networks can be filled with an Activation Function to aid in the network's ability to recognize complex patterns in data. Rectified Linear Units (ReLU), Randomized LeakyReLUs (RReLU), Exponential Linear Units (ELU), and others are examples of common activation functions. Among the most important unsaturated activation functions is the linear rectification function (ReLU). As shown in figure below its mathematical expression is as follows:

f(x)=max (0, x)

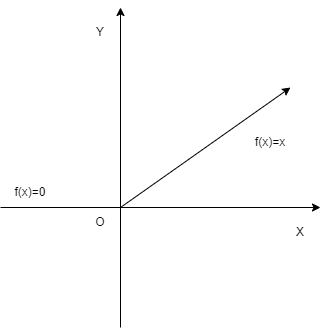


Fig: ReLU function image.

1. Pooling Layer

These days, convolutional neural networks frequently use it as one of their constituent parts. In order to reduce overfitting, the amount of data and parameters are compressed by placing the pooling layer between successive convolutional layers. The pooling layer's primary job is to compress images if that's the input type.The pooling layer's primary job when the input is an image is to compress it.

By performing collective statistical operations on the special diagnosis at various positions in the local area of the image, the pooling layer can effectively reduce the size of the matrix. This reduces the parameters in the final fully connected layer, speeds up calculation speed, and lessens the excessive sensitivity of the convolutional layer to the image position.  The common operations of the pooling layer include the following: max-pooling, average pooling, Spatial Pyramid Pooling etc.

Architecture of CNN:

**Working procedure of CNN**

Step:

import the necessary libraries

set the parameter

define the kernel

Load the image and plot it.

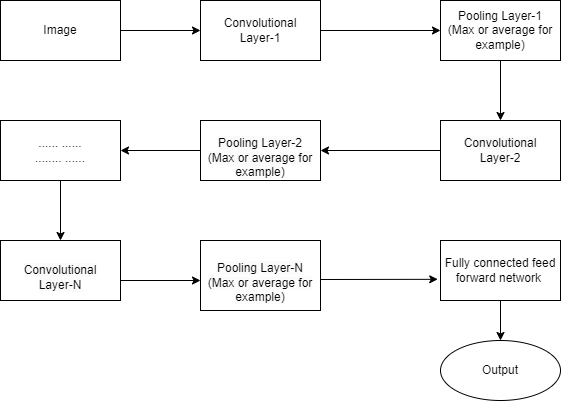
Reformat the image

Apply convolution layer operation and plot the output image.

Apply activation layer operation and plot the output image.

Apply pooling layer operation and plot the output image.

**Flowchart:**

****

**Here are some applications of CNNs:**

**Image Classification:**

CNNs are frequently used in image classification applications, where the network is trained to classify images into various groups. This program is widely used in many different domains, from diagnosing medical images to recognizing objects in photos.

**Object Detection:**

CNNs are useful for locating and identifying objects in pictures. CNNs are used in popular object detection frameworks such as Faster R-CNN and YOLO (You Only Look Once). Robotics, autonomous cars, and surveillance systems are a few examples of applications.

**Facial Recognition:**

Facial recognition systems use CNNs to recognize and verify people. They can be used for social media tagging, access control, and security systems.

**D2DET:**

A deformable two-stage detector is a type of object detection model used in deep learning for computer vision tasks. It combines two key concepts: the two-stage architecture and deformable convolutional networks.

1.Two-stage architecture: Object detection models typically follow a two-stage approach. In the first stage, they generate region proposals, which are potential bounding boxes that may contain objects. In the second stage, these proposed regions are classified and refined to produce the final object detections.

2.Deformable Convolutional Networks (DCN): Deformable Convolutional Networks are a type of convolutional layer that can adaptively adjust their receptive fields based on the features within an image. This allows them to capture more accurateand flexible information about object shapes, especially when objects are occluded, deformed, or occur at various scales.

**Working procedure of D2Det algorithm:**

Certainly, here's a step-by-step procedure for the working of a deformable two-stage detector algorithm:

Step 1:Input Image

Begin with an input image that you want to perform object detection on.

Step 2:Feature Extraction

Pass the input image through a convolutional neural network (CNN) to extract feature maps.These feature maps capture visual information from the image.

Step 3:Region Proposal Network (RPN)

The RPN operates on the feature maps and generates a set of anchor boxes (potential object bounding boxes) at various scales and aspect ratios. Each anchor box is associated with a score indicating the likelihood of containing an object.Apply deformable convolutions in the RPN to adaptively adjust the receptive fields and capture more relevant information.

Step 4:Anchor Scoring

Assign objectness scores to each anchor box based on how likely they are to contain an object. This is done using a classification layer.

Step 5:Anchor Refinement

Predict adjustments (bounding box regressions) for the anchor boxes to better align them with the true object locations. This is done using a regression layer.

Step 6:Anchor Selection

Filter the anchor boxes based on their objectness scores, selecting the top-ranked anchors as proposals for further processing. These proposals represent regions likely to contain objects.

Step 7:Region-of-Interest (RoI) Pooling

Extract fixed-size feature vectors for each selected proposal by applying RoI pooling or a similar technique on the feature maps. This step ensures that all proposals have consistent input sizes for further processing, regardless of their original sizes.

Step 8:Deformable Convolutional Networks (DCN)

Pass the RoI features through deformable convolutional layers, which adaptively adjust their receptive fields. Deformable convolutions help capture information from different position within the RoI, which is particularly beneficial for deformable or occluded objects.

Step 9:Object Classification

Use a classifier network to assign a class label to each RoI, determining the type of object contained in the proposal.

Step 10:Bounding Box Regression

Apply a regressor network to refine the coordinates of the bounding boxes associated with the RoIs. This step helps adjust the bounding boxes to better fit the precise object locations.

Step 11:Post-processing

Perform non-maximum suppression (NMS) to remove redundant or highly overlapping bounding boxes.This ensures that only the most confident detections are retained and eliminates duplicates.

Step 12:Output

Provide the final set of detected objects, along with their class labels and refined bounding box coordinates, as the output of the deformable two-stage detector.

The deformable two-stage detector algorithm combines the strengths of deformable convolutional layers with the two-stage object detection approach to achieve improved accuracy in detecting and localizing objects, especially in cases where objects exhibit deformations, occlusions, or variations in scale.

**SSD:**

SSD operates as a one-shot detector. It predicts the boundary boxes and the classes directly from feature maps in a single pass and does not have a gave region proposal network. SSD adds offsets to default boundary boxes and small convolutional filters to predict object classes in order to increase accuracy.

The SSD object detection composes of 2 parts:

Extract feature maps, and Apply convolution filters to detect objects.

SSD extracts feature maps using VGG16. Next, it makes use of the Conv4\_3 layer to detect objects. As an example, Four object predictions are made for each cell, also known as location. To improve accuracy, SSD can be trained from beginning to end. SSD has better coverage on location, scale, and aspect ratios and makes more predictions.

**Working procedure of ssd:**

A well-liked object detection model for real-time object detection in photos and videos via deep learning is the Single Shot MultiBox Detector (SSD). SSDs are made to anticipate a set of bounding boxes and the class labels that correspond with them in order to identify objects in an image.

Here's an overview of how an SSD works in deep learning:

1. Input Image:

A neural network is used by the SSD model to extract features from an input image of any size. The backbone network is usually a convolutional neural network (CNN) that has been pre-trained, like VGG16 or ResNet. CNN uses several scales to extract features from the picture.

2. Feature Maps:

Various-sized feature maps are produced as the image is processed by CNN. Information is captured at different spatial resolutions by these feature maps. In general, higher-level features are captured by the deeper layers of the CNN, whereas lower-level features are captured by the shallower layers.

3. Multi-scale Feature Fusion:

SSD uses convolutional layers with different kernel sizes and moves to obtain feature maps at different scales. These layers are in charge of combining features from the backbone network's various layers. To identify objects of various sizes, feature maps at various scales are utilized.

4. Default Anchor Boxes:

A set of default anchor boxes, also called default bounding boxes, with various aspect ratios and scales are defined for every location in the feature maps. To begin object detection, these anchor boxes are used. In order to better fit the real objects in the picture, the SSD model predicts offsets, or deltas, for these anchor boxes.

5. Object Detection Head:

After feature extraction and anchor box definition, the SSD network splits into two parallel subnetworks:

- Localization Head: This part predicts the offsets (deltas) for each anchor box to adjust their positions and sizes to match the ground-truth objects in the image.

- Classification Head: This part predicts the class probabilities for each anchor box, indicating the likelihood that an object of a particular class is present in that box.

6. Non-Maximum Suppression (NMS):

Following the receipt of predictions from the localization and classification heads, low-confidence and redundant detections are filtered out using a post-processing technique known as non-maximum suppression. NMS makes sure that as final detections, only the most certain and non-overlapping bounding boxes are kept.

7. Output:

An array of bounding boxes with the corresponding class labels and confidence scores is the SSD model's final result. The objects that were found in the input image are represented by these bounding boxes.

Because SSD is effective at predicting class labels and bounding boxes at multiple scales in a single forward pass, it can detect objects in real time. This makes it a well-liked option for many applications, such as image recognition, autonomous driving, and surveillance. By modifying the quantity and dimensions of the default anchor boxes and fine-tuning the model on particular datasets, the SSD architecture's flexibility allows it to be tailored for a variety of object detection applications.

**YOLOv:**

YOLO is an algorithm that provides real-time object detection using neural networks. The accuracy and speed of this algorithm makes it popular.

The abbreviation YOLO stands for "You Only Look Once." This algorithm (in real-time) finds and recognizes different objects in an image. YOLO employs object detection as a regression problem, resulting in the class probabilities of the identified images. Convolutional neural networks (CNN) are used by the YOLO algorithm to detect objects in real-time. As the name implies, the algorithm can detect objects with just one forward propagation via a neural network. This indicates that a single algorithm run is used to predict the entire image. Multiple class probabilities and bounding boxes are simultaneously predicted by the CNN.

There are several variations of the YOLO algorithm. Tiny YOLO and YOLOv3 are a couple of the popular ones.

**Working procedure of yolo algorithms:**

YOLO algorithm works using the following three techniques:

1. Residual blocks
2. Bounding box regression
3. Intersection Over Union (IOU)

**Residual blocks**

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

Bounding box regression

A bounding box is an outline that highlights an object in an image.

Every bounding box in the image consists of the following attributes:

* Width (bw)
* Height (bh)
* Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
* Bounding box center (bx,by)

**Intersection over union (IOU)**

In object detection, the phenomenon known as intersection over union (IOU) characterizes how boxes overlap. YOLO creates an output box that exactly covers the objects by using IOU. The task of predicting the bounding boxes and their confidence scores falls on each grid cell. If the expected and actual bounding boxes match, the IOU is equal to 1. Bounding boxes that are not equal to the actual box are removed by this mechanism.

**Combination of the three techniques**

The image is first split up into grid cells. B bounding boxes are predicted by each grid cell, along with their confidence scores. To determine each object's class, the cells make predictions about the class probabilities.

A car, a dog, and a bicycle are just a few examples of the at least three classes of objects that are visible. A single convolutional neural network is used to make all of the predictions at the same time.

The predicted bounding boxes and the actual boxes of the objects are guaranteed to be equal by intersection over union. The effect removes irrelevant bounding boxes that don't match the object's dimensions (height and width). The final detection will be made up of distinct bounding boxes that precisely match the objects.

For instance, the yellow bounding box surrounds the bicycle and the pink bounding box covers the car. The blue bounding box has been used to highlight the dog.

YOLO algorithm can be applied in the following fields:

Autonomous vehicles can utilize the YOLO algorithm to identify nearby objects, including people, cars, and parking signals. In forests, this algorithm is used to identify different kinds of animals. Journalists and wildlife rangers use this kind of detection to recognize animals in photos.

**Proposed System Model**

In our proposed topic we used four algorithms . Here is a flowchart of our proposed model

