OBJECT DETECTION USING DEEP LEARNING

With MASK RCNN and YOLOv8

Prepared by :

- Anamika Singh 2023CSB030
- Shahrin Fatima 2023CSB061
- Diptanil Guha 2023CSB077
- Jyotirmoy Dutta 2023CSB110

Under the mentorship of
Prof. Biplab K Sikdar

OBJECTIVE

To make a model that can detect and classify animals present in an input image.

Content

- Deep Learning
- Neural Network
- Neurons
- Learning Process of a Neural Network
  - Forward Propagation
  - Learning (Training, Backpropagation)
- Convolutional Neural Networks (CNN)
- Deep Learning Models
  - Yolov 8
  - Mask RCNN
  - Comparison Between YOLOv8 and Mask R-CNN
- The model to detect and classify the animals
  - Specifications
  - Experimentation and Performance Evaluation
- Conclusion

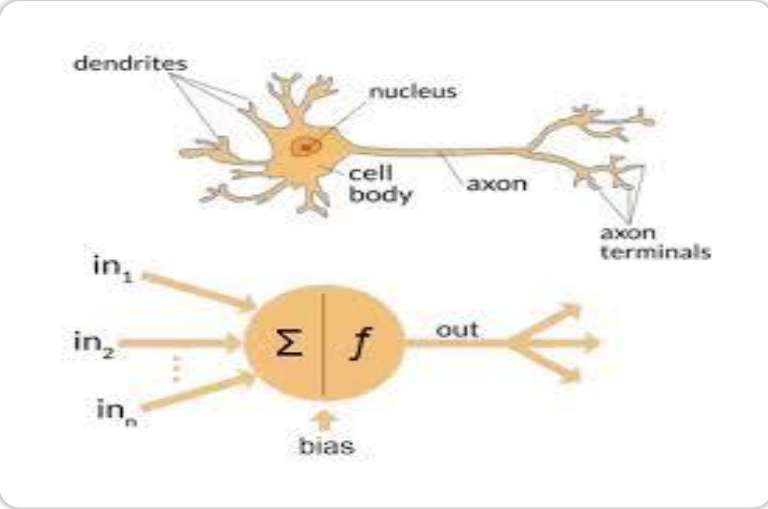
Deep Learning

- **Deep learning** is a subset of machine learning
- Uses special models called **neural networks** with many layers
- These help computers to learn complex things from data

- like recognizing images or understanding speech

- It **doesn't need humans** to tell it what features to look for

Neural networks forms the foundation of Deep learning, They mimic how the human brain works by using **layers of interconnected nodes** (like neurons) to process information. Deep learning uses **many layers** of neural networks (called deep neural networks) to learn complex patterns from data. Hence it is important for us to study about Neural Networks.



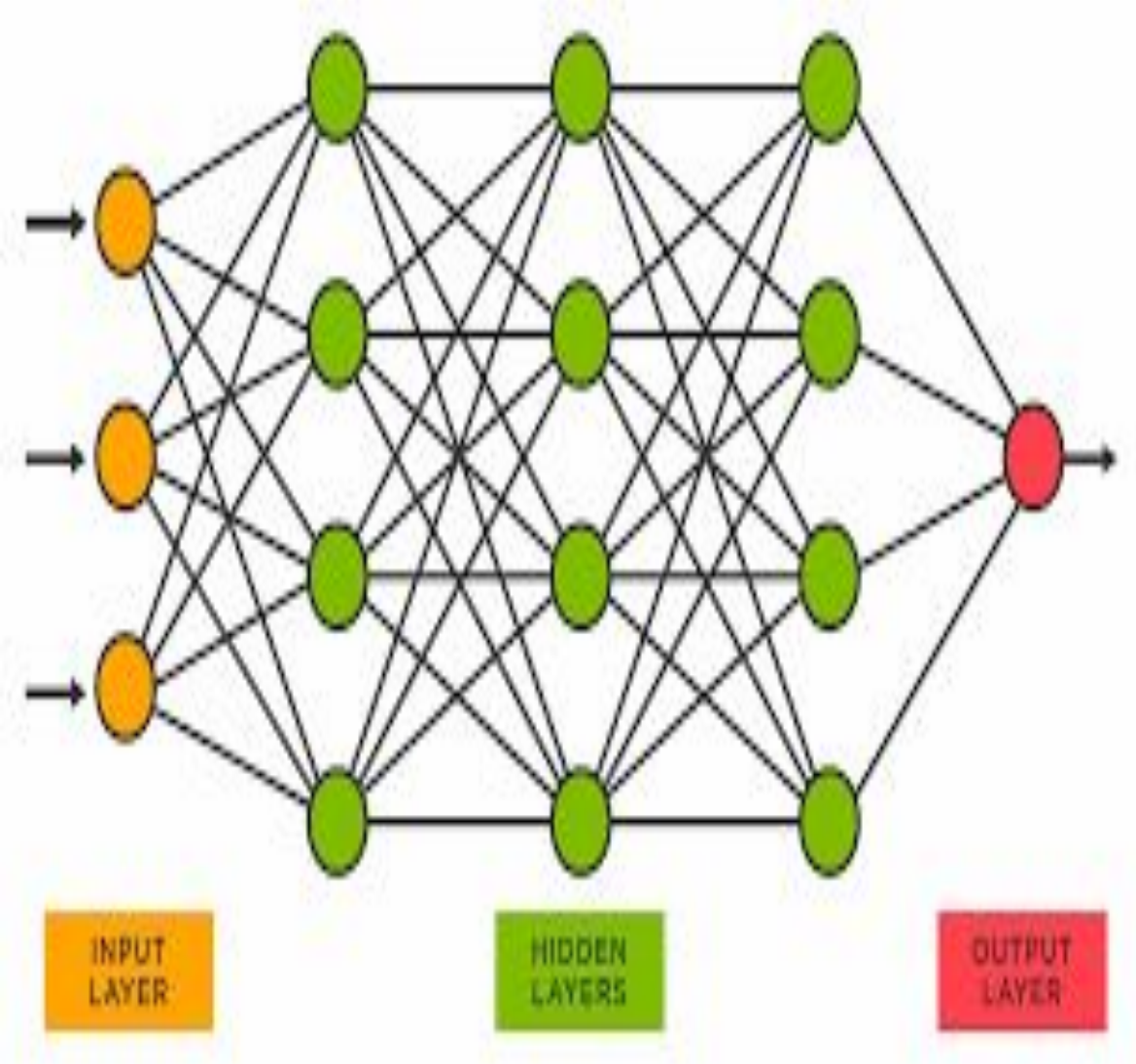
Neural Network

- A Neural network consists of several layers.
- A layer can be defined as a group of neurons which takes same or similar inputs and gives similar output.
- Layers can be classified into 3 types :

  - **Input Layer (Orange)**: This is where the data enters. Each neuron represents a feature (like a pixel in an image).

  - **Hidden Layers (Green)**: These layers process the data. They help the network learn complex patterns. The more layers, the more complex the learning.

  - **Output Layer (Red)**: This is where the result comes out. The number of neurons depends on the task (e.g., 10 neurons for classifying into 10 categories)

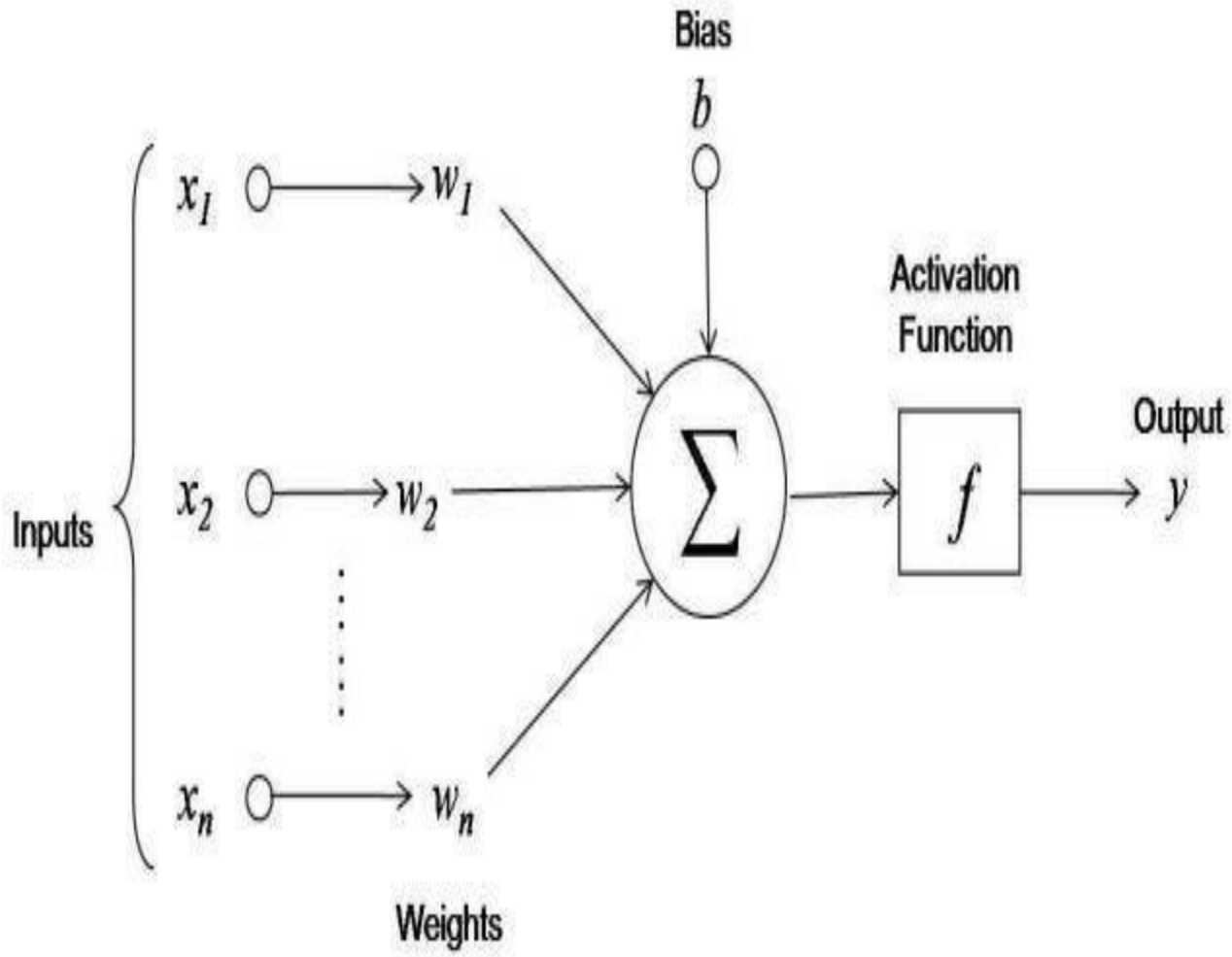


Neurons

- A neuron receives inputs, processes them, and produces an output

- **Inputs (x1, x2, ..., xn):** These are the values that the neuron receives as input. They can be individual features or the outputs of other neurons.

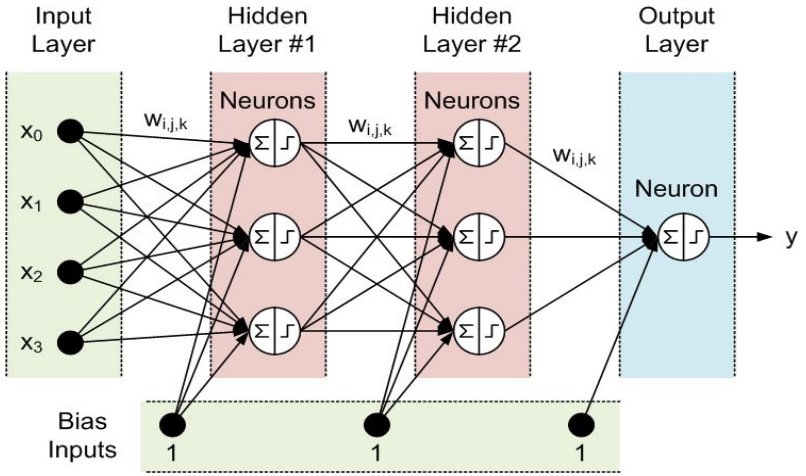
- **Weights (w1, w2, ..., wn):** Each input is multiplied by a corresponding weight. These weights determine the importance of each input in the neuron's output.



Neurons contd....

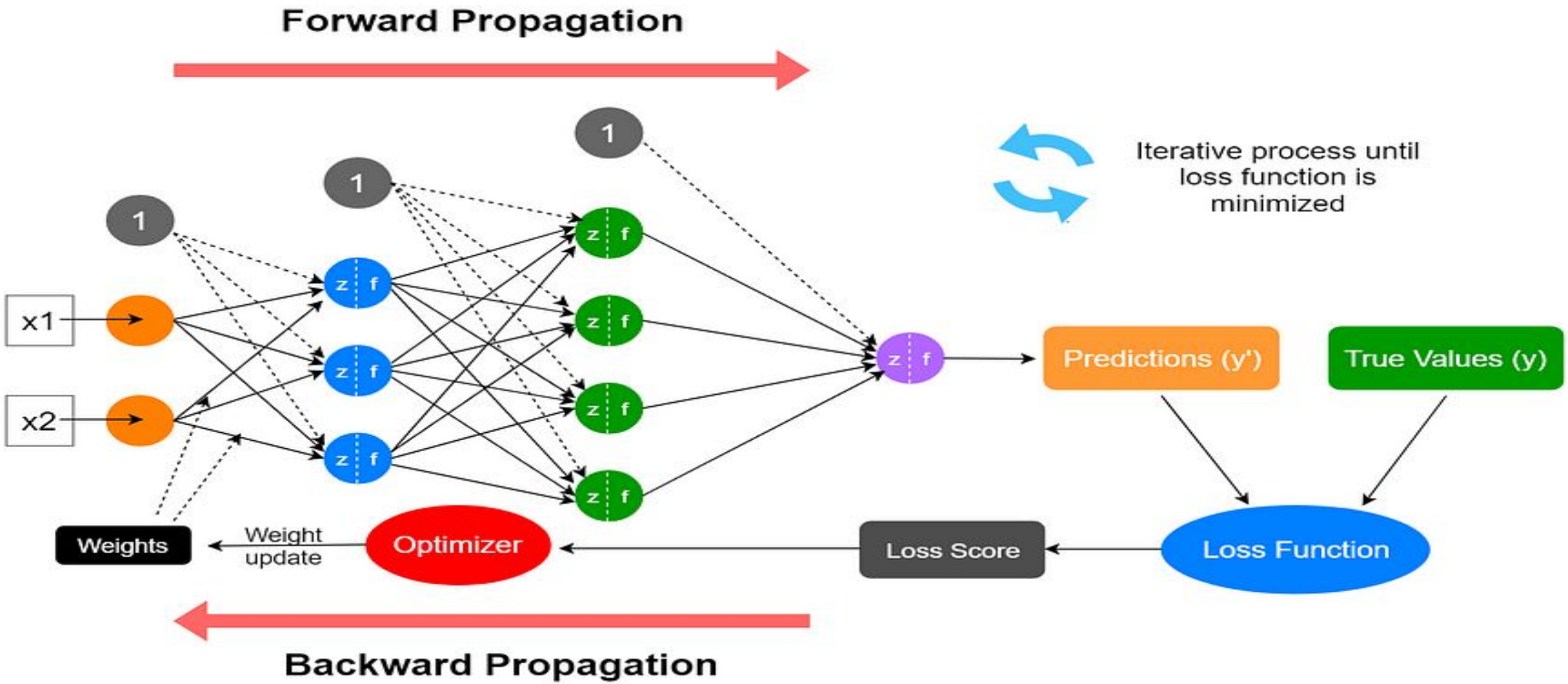
- **Bias (b):** The bias is an additional input to each neuron.
- **Summation:** The weighted inputs and bias are summed together to form a linear combination.
- **Activation Function (f()):** The sum is passed through an activation function, which introduces non-linearity and helps the neuron learn complex patterns. Common activation functions include ReLU, sigmoid, and tanh.
- **Output (y):** The output of the activation function becomes the neuron's output.

Structure of a neural network with working of each neuron mentioned.



Now, Since we know the basic structure of a Neural Network it is important for us to know how neural networks learn and train themselves.

Learning Process of a Neural Network



learning contd....

1. Forward Propagation

- **Input:** Data is fed into the input layer.
- **Layer-wise Computation:** Each neuron in a layer receives inputs from neurons in the previous layer, multiplies them by corresponding weights, adds the bias, and applies the activation function to produce an output.
- **Output:** The output from the final layer represents the network's prediction

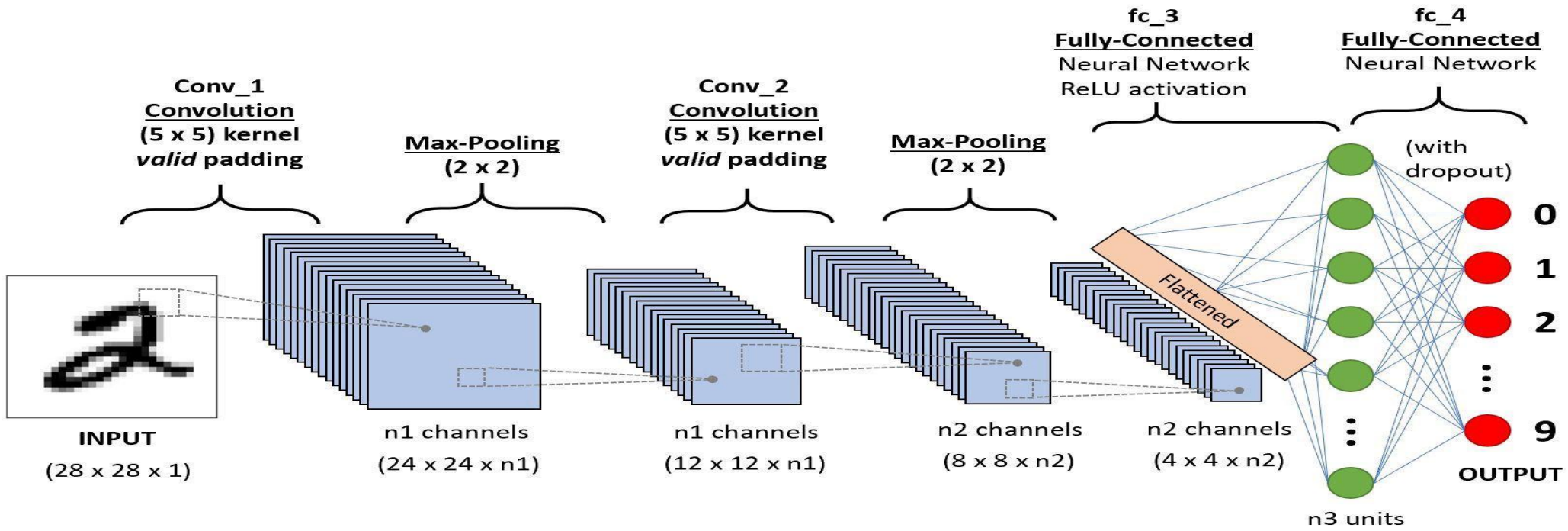
2. Learning (Training, Backpropagation)

- **Loss/Cost Function:**
  - A function that quantifies the error between the network's predictions and the true labels in the training data.
  - The goal of training is to minimize this loss function.
- **Gradient Descent(Optimizers)**
  - Algorithm that adjusts the weights and biases of the network to minimize the loss function.
  - It calculates the gradient of the loss function with respect to each parameter and updates the parameters in the opposite direction of the gradient.

learning contd....

- The training process is repeated multiple times until the loss function is minimized.
- The input dataset is randomly divided into batches and they are used recursively for training.

- Now, Since we know how a basic neural network works we now need to know about **Convolutional Neural Networks**.
- Utilizes specialized layers and operations to extract meaningful features and patterns.
- Specifically designed for processing grid-like data, such as images.



Convolutional Neural Networks

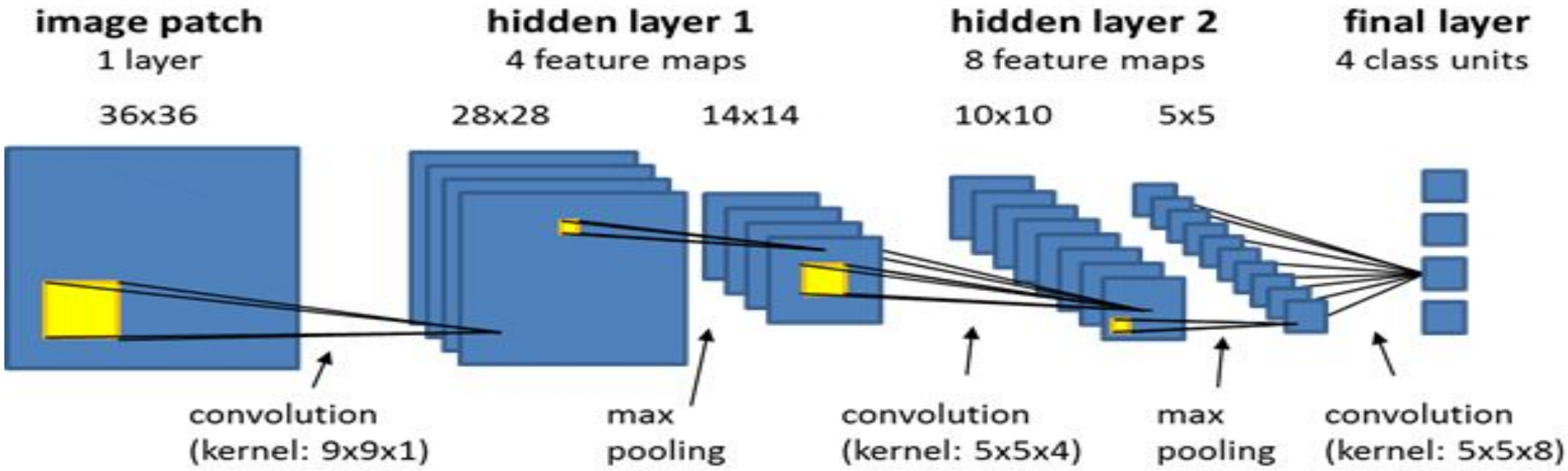
Convolutional Neural Networks(CNN)

Key Components and Processes in CNNs:

- **Kernel (or Filter):** A small matrix that slides across the input image, performing element-wise multiplication and summation to extract features like edges and textures.
- **Feature Map:** The output produced by applying a convolutional layer to an input. Each feature map represents the presence and intensity of a particular feature across the image.
- **Max Pooling:** A down sampling technique applied after convolution layers. It selects the maximum value from a small region of the feature map, reducing dimensionality and helping to prevent overfitting.
- **Flattening:** The process of converting multidimensional feature maps into a single-dimensional vector, preparing it for input to fully connected layers..

CNN contd....

- **Fully Connected Layers:** Standard neural network layers where all neurons are connected to every neuron in the previous layer.



- Since we know the working of Neural Networks and Convolutional Neural Networks we are now in a position to understand the pre-build models like Yolo v8 and Mask RCNN.
- These Models mainly make use of a certain number of NNs and CNNs tuned in such a way to detect images efficiently.

Deep Learning Models:-

1. Yolo V8 :-

1. **About :-** It's a state-of-the-art deep learning model for real time object detection in computer vision application. YOLOv8 was developed by ultralytics. It's more efficient and less likely to overfit.

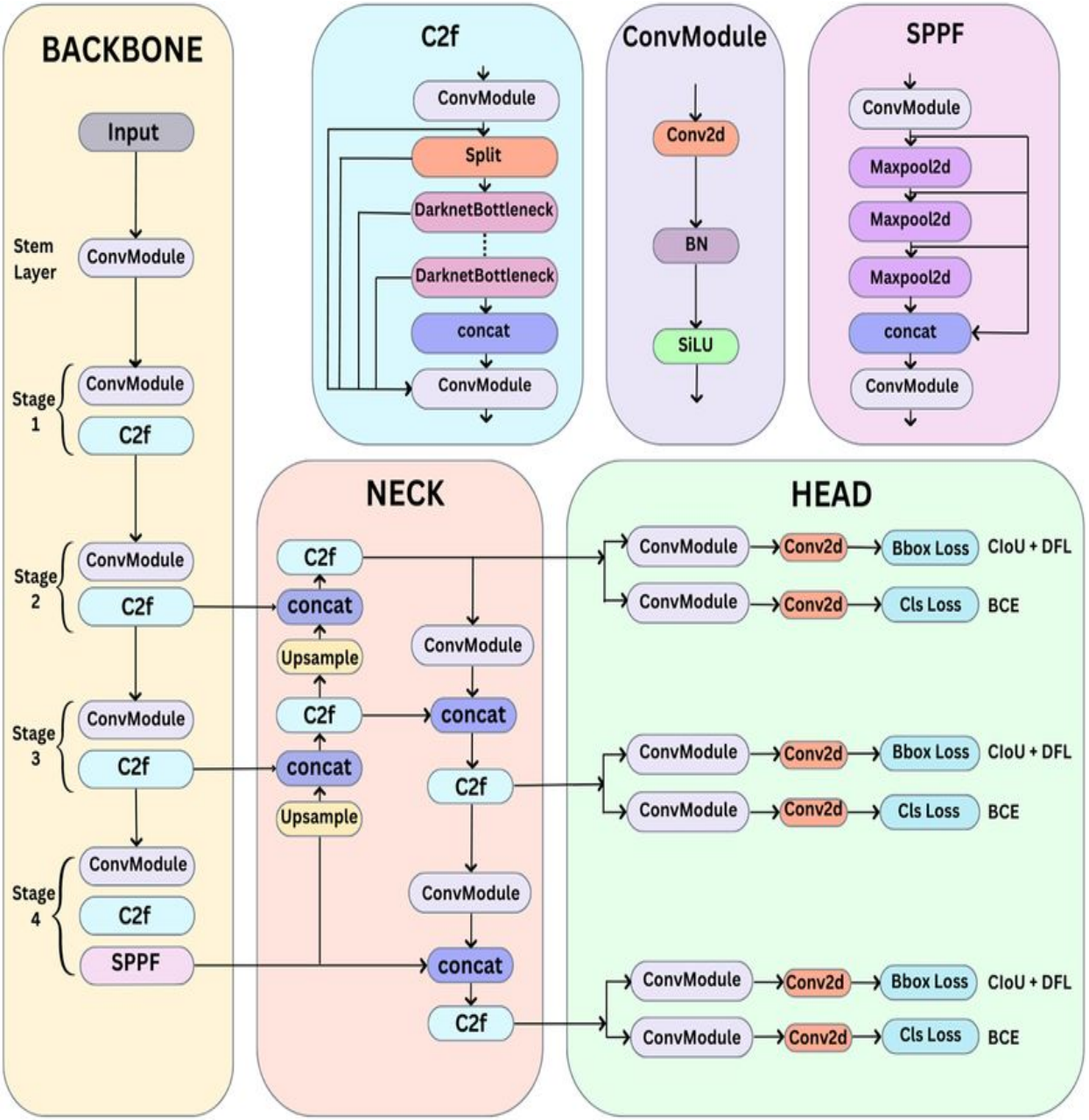
2. **Architecture :-** This model is divided onto 3 main components:-

   - **Backbone :-** The backbone is responsible for extracting features from the input image, and YOLOv8 employs a variety of backbones, including CSPDarknet 53 and efficient Det.

   - **Neck :-** The neck connects the backbone to the head and is crucial for feature fusion.

   - **Head :-** The head is responsible for predicting bounding boxes, object classes, and confidence scores.

Yolo contd....

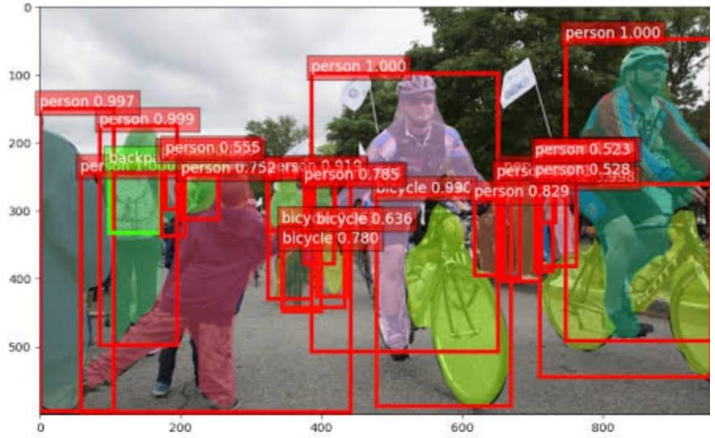


- **CONVOLUTIONAL Block:-**
  - The most commonly used block.
  - In YOLOv8 the conv block consists of a 2D conv layer, a 2D based normalization and silU activation function.
- **C2f Block:-**
  - This block contains a conv block which then the resulting feature maps will be split.
  - One goes to the bottleneck block whereas the other goes directly into the concat block.
  - In C2f block we can have many bottleneck blocks. At the end there is another conv block.
  - Bottleneck itself is a sequence of conv block with a shortcut.
- **SPPF Block(Special pyramid pooling fusion):-**
  - It stands for Special Pyramid Pulling Fast. It's a modification of SPP with a highest speed.
  - Inside the SPPF there is a conv block at the beginning and followed by three 2D max pulling layer.
  - Every resulting F map is concatenated right before the end of SPPF. It's ending with a conv block.
- **DETECT Block:-**
  - This is where detection happens.
  - The prediction happens in the grid cell. The detect block contains two tracks.
  - The first track is for bonding box prediction whereas the other is class prediction. Both tracks has the same block sequence which is two conv block and a single 2D conv layer.

2. Mask R-CNN:-

1. **About :- Mask R-CNN** is an extension of **Faster R-CNN** and is used for **instance segmentation**. It not only detects objects but also creates a pixel-wise **mask** for each object, giving a more detailed output compared to YOLO.

2. **Key features of Mask RCNN :-** Two-Stage Detection:



Region Proposal Network (RPN): In the first stage, the model generates candidate regions where objects might be located (region proposals).

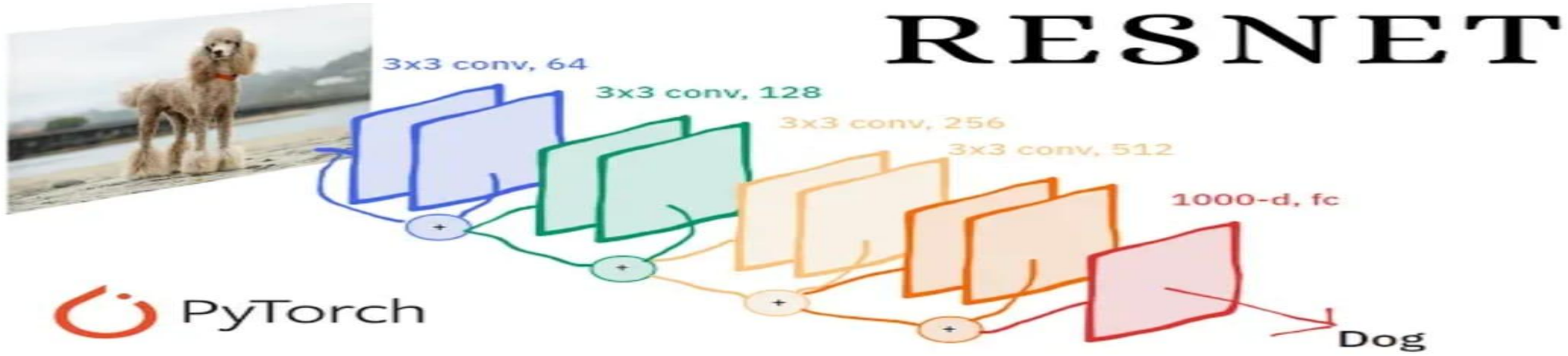
Classification and Mask Prediction: In the second stage, it refines these regions by classifying the object, refining the bounding box, and generating a pixel-level **mask** for each detected object.

Mask RCNN contd....

3. **Object Masking :-** Mask R-CNN does not just detect and classify objects, but it also predicts a **segmentation mask**. This means it draws an outline around the object, identifying every pixel that belongs to the object (useful for very precise object localization).

4. **Improved Accuracy :-** The two-stage process (first proposing regions and then refining them) generally results in higher accuracy for both **object detection** and **instance segmentation**.

5. **Backbone Networks :-** Mask R-CNN typically uses **ResNet** or similar deep architectures as the backbone network for feature extraction.



Comparison Between YOLOv8 and Mask R-CNN

Feature	YOLOv8	Mask R-CNN
Type	Single-stage detector	Two-stage detector
Speed	Very fast (real-time performance)	Slower due to two-stage process
Accuracy	High, but typically less accurate	Higher accuracy, especially for detailed
Output	Bounding boxes around objects	Bounding boxes + precise pixel-level masks
Use Case	Real-time applications, such as video or wildlife monitoring	Applications requiring high precision, like medical imaging or instance segmentation
Complexity	Simpler and faster	More complex due to mask prediction
Strength	Fast object detection	Accurate object segmentation

The model to detect and classify the animals

Our model can detect and classify 4 animals namely elephant, zebra, buffalo and rhino and give 3 outputs i.e. bounding box, animal name and the confidence score.

1. Model Specifications :-

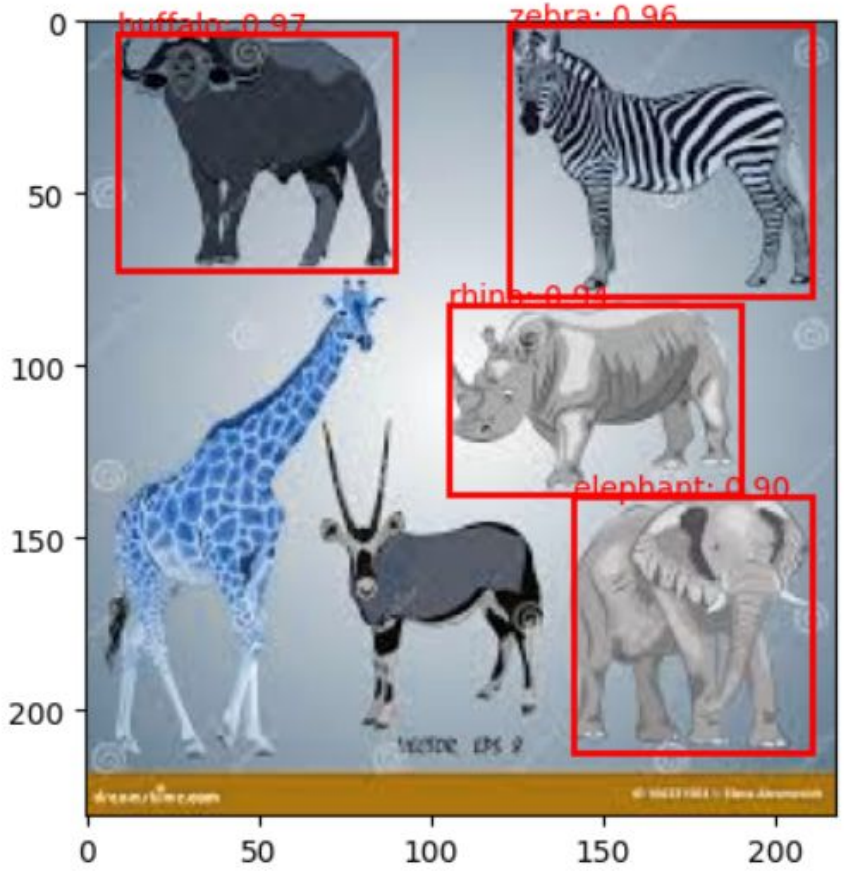
- **Yolov8n** has been used.
- 225 layers and 3,011,628 parameters present
- Data Augmentation used :- Blur, Median blur, ToGrey, CLAHE.
- Optimiser :- Adam.
- Batch Size :- 16
- Epoch :- 40
- Dataset used :- Custom Dataset
  - Training :- 1052
  - Validation :- 225

2. Model Performance :-

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	225	379	0.967	0.9	0.955	0.798
buffalo	62	89	0.976	0.896	0.952	0.808
elephant	53	91	0.931	0.886	0.95	0.764
rhino	55	85	0.983	0.941	0.961	0.835
zebra	59	114	0.98	0.877	0.958	0.786

3. Model Output :-

INPUT IMAGE



OUTPUT IMAGE

Conclusion

- **Studied Deep Learning Fundamentals:** We gained a decent understanding of the core concepts of deep learning, including neural networks, convolutional neural networks (CNNs), and object detection techniques.

- **Explored Image Detection and Classification:** We studied the application of deep learning for image detection and classification, learning about techniques like region-based convolutional neural networks (Mask R-CNN) and You Only Look Once (YOLO).

- **Compared YOLOv8 and Mask R-CNN:** We analyzed the strengths and weaknesses of YOLOv8 and Mask R-CNN, considering factors like speed, accuracy, and complexity and selected Yolo v8 for our project.

- **Developed the YOLOv8 Model:** We trained and fine-tuned the YOLOv8 model on a dataset of 1052 images, configuring it to classify the four target animal species: Buffalo, Zebra, Elephant, and Rhino.

- **Evaluated Model Performance:** We assessed the performance of our trained YOLOv8 model using the mAP50 metric, achieving a score of 0.955, indicating a decent level of accuracy.

- **Areas for Improvement:** We recognized the potential to further enhance the model's performance by exploring techniques like using transformer-based architectures (ViT, Swin Transformer) for feature extraction and modifying the YOLOv8 architecture.