Report on TASK2:AI-PROCTORING SYSTEM

Problem Statement:

Research study of various Deep Learning models:

- CNN
- ResNet
- Yolo (different versions)

Compare and contrast the pros and cons of each of the models and make an analysis report about the complete working and in-depth architecture. List down the features of each model which according to you are useful.

Neural Networks

Neural Networks is a subset of machine learning and forms the heart of deep learning.
It is based on the human brain consisting of neurons connected to each other.

Neural networks are comprised of node layers:

1. input layer
2. hidden layer
3. output layer

Each node acts like an artificial neuron - it connects to another - has an associated weight and threshold.
If output of any node is above the threshold value, the node is activated and sends data along to the next node. Otherwise no data is passed along.
They rely on training data to learn and improve their accuracy(Fine tuning is also done to improve the accuracy)
Powerful tools allowing us to classify and cluster data at very high speeds

Types of Neural Network

- FeedForward neural network
  - Uses sigmoid neurons
  - Form the foundation of computer vision, natural language processing or other neural networks
  - CV - field of AI that enables a computer to understand and interpret the image or visual data
- Convolutional Neural Networks

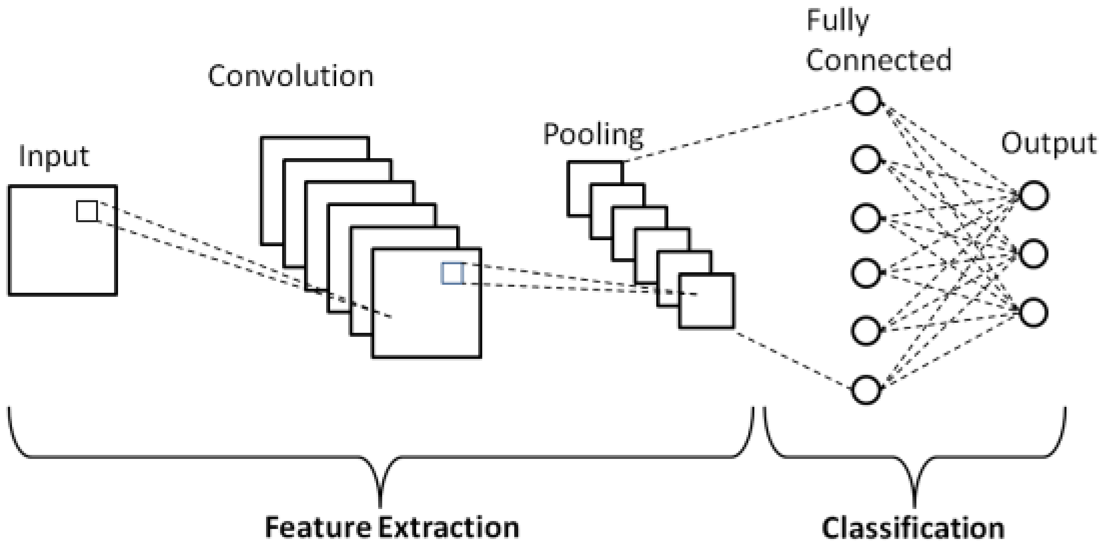
    - Similar to FeedForward but are utilized for image recognition, pattern recognition and/or computer vision

- Recurrent Neural Networks
    - These are identified by their feedback loops
    - These are leveraged when using time-series data to make predictions
        - stock market predictions
        - sales forecasting

Convolutional Neural Networks

Convolutional neural network (CNN) is an extended version of ANN which is predominantly used to extract features from a grid like matrix dataset.
Their superiority in performance Convolutional Neural Networks are distinguished from normal neural networks by their superior performance with image.

<u>CNN ARCHITECTURE</u>



Three main layers:
1. Convolutional Layer-applies filters to extract features
2. Pooling layer-downsamples the image to reduce computation
3. Fully-connected Layer makes the final prediction

The network learns the optimal filters through back-propagation and gradient descent.

<u>**Convolutional Layer**</u>

Components that make up the convolutional layer include:
Input data
Filter
Feature Map

Process of convolution:
Filter or a kernel is a feature detector which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution. The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image(usually it is 3x3). The size of the filter determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output is the feature map, activation map or a convolved feature.
After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

<u>**Pooling Layer:**</u>
This layer conducts dimensionality reduction, reducing the number of parameters in the input. The kernel applies an aggregation function to the values within the receptive field, populating the output array.

Types of pooling:
- Max Pooling - selects the (i,j) with the maximum value. More often.
- Average Pooling - Calculates the average value within the receptive field

Advantages of Pooling layer:
- Reduce Complexity
- Improve efficiency
- Limit risk of overfitting

<u>**FC layer**</u>
Each node in the output layer is fully connected to its previous nodes
This layer performs the task of classification based on the features extracted through the previous layers and their different filters.

Pros and cons of CNNs

PROS	CONS
Efficient image processing: Faster and more efficient than other algorithms	High computational requirements: CNNs typically have a large number of layers and parameters, which require a lot of processing power and memory to train and run.
High accuracy rates: can learn to recognize complex patterns in images by analyzing large datasets.	Difficulty with small datasets: CNNs also require large datasets to achieve high accuracy rates. This is because they learn to recognize patterns in images by analyzing many examples of those patterns
Robust to noise: they can still recognise patterns in images even if they are distorted or corrupted. multiple layers make them more resilient to noise	Lack of Interpretability- Difficult to understand how the CNN makes its decisions. This can be problematic in applications where it is important to know why a certain decision was made.
Transfer learning: they can be trained on one task and then used to perform another task with little or no additional training	Vulnerability to adversarial attacks: Intentionally manipulating the input data to fool the CNN into making incorrect decisions.
Automated feature extraction: they can learn to recognize patterns in images without the need for manual feature engineering.	Limited ability to generalize: They may perform poorly on images that are very different from those in the training dataset.

ResNet

ResNet or Residual Network is a deep learning model based on the convolutional neural network architecture designed to support a large number(hundreds or thousands) of convolutional layers. Adding more layers to the convolutional neural network caused the problem of "vanishing gradient".

Since neural networks are trained by back propagation which relies on gradient descent,if there are too many layers, repeated functions will eventually reduce the gradient causing the performance to deteriorate with each function added.
Resnet solves this problem of vanishing gradient with a solution called "skip connections"

ResNet stacks and skips multiple identity mappings and reuses the activations of the previous layers. This speeds up the initial training.
Then, when the network is retrained, all layers are expanded and the remaining parts of the network—known as the residual parts—are allowed to explore more of the feature space of the input image.
Most ResNet models skip two or three layers at a time with nonlinearity and batch normalization in between. More advanced ResNet architectures, known as HighwayNets, can learn “skip weights”, which dynamically determine the number of layers to skip.

Residual Block
The most important part of a ResNet architecture.
The ResNet architecture introduces the simple concept of adding an intermediate input to the output of a series of convolution blocks.
In previous approaches, each layer feed to its next layer and the same goes on subsequently. In a residual network, each layer feeds to its next layer and directly to the 2-3 layers below it.

- It provides an alternative path for the better flow of gradients during the backpropagation. It helps the earlier layers to learn better, which helps to improve the performance of the network.
- The identity mapping learns an identity function that ensures that the residual block performs as good as the lower layer.

PROS	CONS
Improved training of deep networks	Increased memory consumption:
High accuracy rates	Computational complexity:Training very deep ResNet models requires significant computational power, which can be a bottleneck in resource-constrained scenarios.

Reduced Overfitting: ResNet incorporates batch normalization and dropout techniques, which help in reducing overfitting. Batch normalization normalizes the activations, making the network more robust to variations in input data, and dropout randomly drops out units during training, which regularizes the model.	Over-reliance on depth: Very deep models can suffer from diminishing returns, as the network may struggle to optimize the gradients and require more data and computational resources for training.
Transfer learning	Interpretability: Understanding the specific representations and decision-making processes of individual layers can be challenging, making it harder to gain insights into the inner workings of the network.
VersatilityResNet can be used for various computer vision tasks such as image classification, object detection, semantic segmentation, and more. Its flexibility and scalability make it suitable for different applications and enable researchers to adapt it to their specific needs.	Limited performance on small datasets

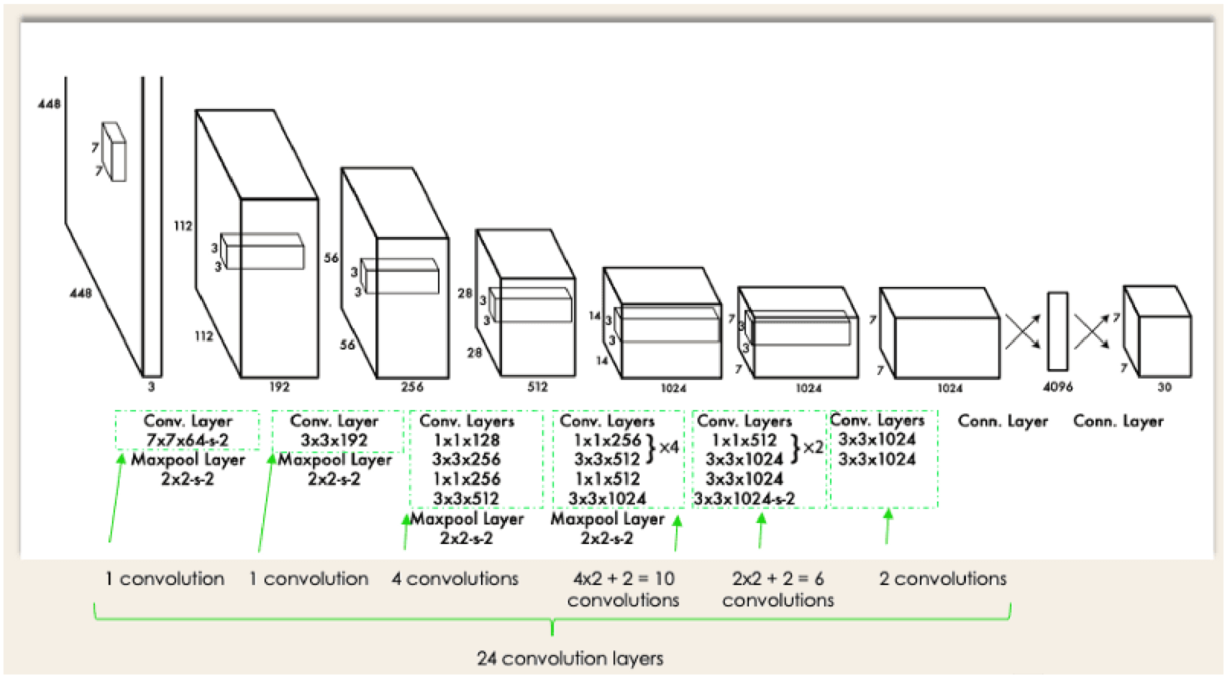
YOLO (You Only Look Once)

You Only Look Once (YOLO) is a state-of-the-art, real-time object detection algorithm.
YOLO authors frame the object detection problem as a regression problem instead of a classification task by spatially separating bounding boxes and associating probabilities to each of the detected images using a single convolutional neural network (CNN).

Architecture

YOLO architecture is similar to GoogleNet. As illustrated below, it has overall 24 convolutional layers, four max-pooling layers, and two fully connected layers.

(diagram in next page)



- Resizes the input image into 448x448 before going through the convolutional network.
- A 1x1 convolution is first applied to reduce the number of channels, which is then followed by a 3x3 convolution to generate a cuboidal output.
- The activation function under the hood is ReLU, except for the final layer, which uses a linear activation function.
- Some additional techniques, such as batch normalization and dropout, respectively regularize the model and prevent it from overfitting.

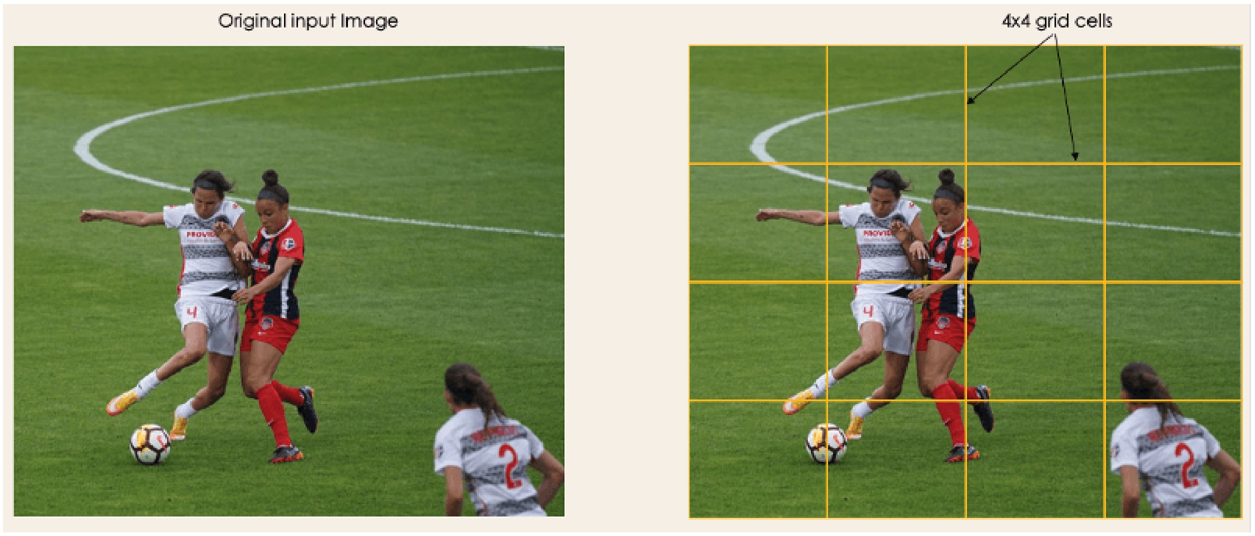
YOLO algorithm

The YOLO algorithm(which in simple words detects Image B from Image A) is based on the following approaches:

- Residual blocks
- Bounding box regression
- Intersection Over Unions or IOU for short
- Non-Maximum Suppression.

Residual Blocks:

This first step starts by dividing the original image (A) into NxN grid cells of equal shape. Each cell in the grid is responsible for localizing and predicting the class of the object that it covers, along with the probability/confidence value.

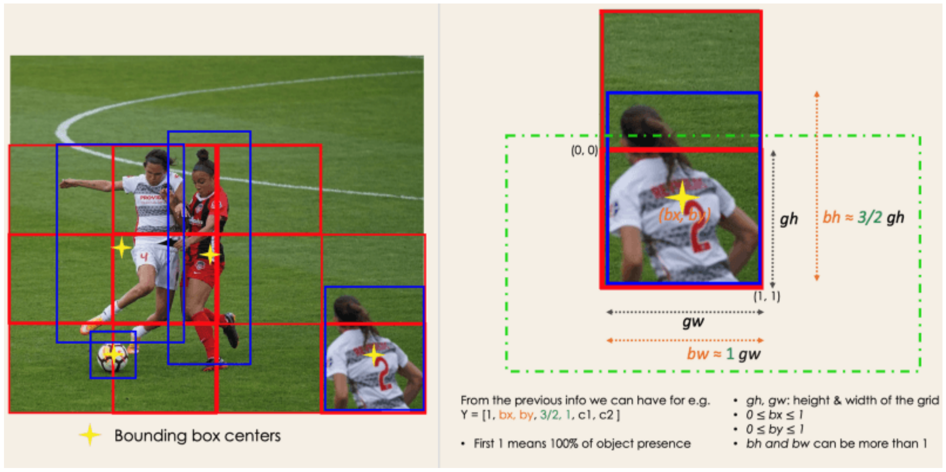


Bounding Box Regression:

The next step is to determine the bounding boxes which correspond to rectangles highlighting all the objects in the image.

YOLO determines the attributes of these bounding boxes using a single regression module in the following format, where Y is the final vector representation for each bounding box.

Y = [pc, bx, by, bh, bw, c1, c2]



From the previous info we can have for e.g.
Y = [1, bx, by, 3/2, 1, c1, c2]

- First 1 means 100% of object presence

- *gh, gw*: height & width of the grid
- *0 ≤ bx ≤ 1*
- *0 ≤ by ≤ 1*
- *bh and bw* can be more than 1

where,

pc: probability score of grid containing an object

(grid with score>0 and grid with score=0 are important)

bx, by: coordinates of the center of the bounding box with respect to enveloping grid center

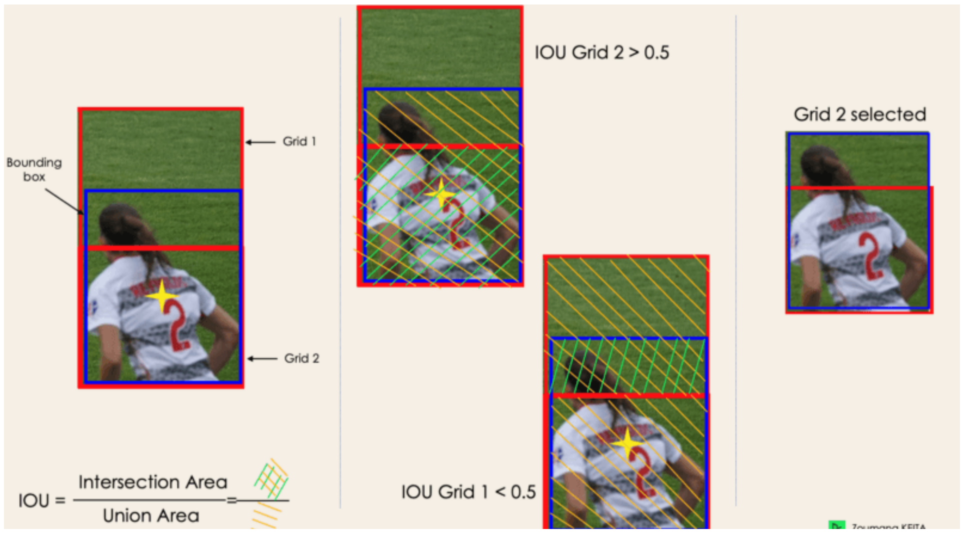
bh,bw: height and width of the bounding box with respect to the enveloping grid cell

c1,c2,c3,....cn: classes required

Intersection Over Unions or IOU:

There may be many grid box candidates for selection. Goal is to discard irrelevant grid boxes.

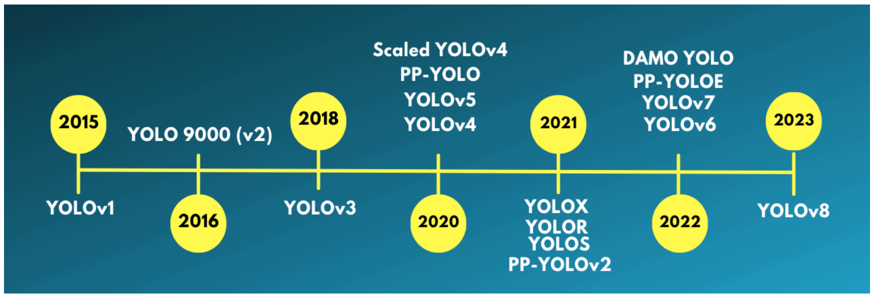
- The user defines its IOU selection threshold, which can be, for instance, 0.5.
- Then YOLO computes the IOU of each grid cell which is the Intersection area divided by the Union Area.
- Finally, it ignores the prediction of the grid cells having an IOU ≤ threshold and considers those with an IOU > threshold.



Non-Max Suppression or NMS

Setting a threshold for the IOU is not always enough because an object can have multiple boxes with IOU beyond the threshold, and leaving all those boxes might include noise. Here is where we can use NMS to keep only the boxes with the highest probability score of detection.

Different versions of YOLO



YOLOv1

Game changer for object detection because of its ability to quickly detect objects

Limitations:

- *Struggles to detect small objects*
- *Unable to detect new or unusual shapes*
- *Incorrect localisation due to bad design of loss function*

YOLOv2

Made the YOLO model better, faster and stronger due to its inclusion of Darknet-19 as new architecture, batch normalisation, convolutional layers using anchor boxes, dimensionality clustering and fine-grain features.

YOLOv3

Uses darknet-53 as the architecture which is bigger, faster and more accurate compared to darknet-19 which increased the prediction accuracy of the entire model.

YOLOv4

- Spatial Pyramid Pooling (SPP) block significantly increases the receptive field, segregates the most relevant context features, and does not affect the network speed.
- Instead of the Feature Pyramid Network (FPN) used in YOLOv3, YOLOv4 uses PANet for parameter aggregation from different detection levels.
- Data augmentation uses the mosaic technique that combines four training images in addition to a self-adversarial training approach.
- Perform optimal hyper-parameter selection using genetic algorithms.

YOLOv5

YOLOR is based on the unified network which is a combination of explicit and implicit knowledge approaches. Explicit knowledge is normal or conscious learning. Implicit learning on the other hand is one performed subconsciously (from experience).

Combining these two technics, YOLOR is able to create a more robust architecture based on three processes: *(1) feature alignment, (2) prediction alignment for object detection, and (3) canonical representation for multi-task learning*

YOLOv6

YOLOv6 is a single stage object detection framework for Industrial application. YOLOv6 introduced three significant improvements to the previous YOLOv's: *a hardware-friendly backbone and neck design, an efficient decoupled head, and a more effective training strategy.*

YOLOv7

This version has made a significant move in the field of object detection, and it surpassed all the previous models in terms of accuracy and speed. YOLOv7 has made a major change in its (1) architecture and (2) at the Trainable bag-of-freebies level:

Architectural level: YOLOv7 reformed its architecture by integrating the Extended Efficient Layer Aggregation Network (E-ELAN) which allows the model to learn more diverse features for better learning.

Trained Bag of freebies: The term bag-of-freebies refers to improving the model's accuracy without increasing the training cost, and this is the reason why YOLOv7 increased not only the inference speed but also the detection accuracy.

YOLOv8

YOLOv8 is the latest family of YOLO based Object Detection models from Ultralytics providing state-of-the-art performance.

Leveraging the previous YOLO versions, the YOLOv8 model is faster and more accurate while providing a unified framework for training models for performing

- Object Detection,
- Instance Segmentation, and
- Image Classification.

YOLOv8 is also highly efficient and flexible supporting numerous export formats and the model can run on CPUs & GPUs.