**Approach**:

For the given problem statement, of prediction of boundaries of a bound box around a detected object in the image, we have tried an approach similar to the established algorithm: YOLO (You Only Look Once).

During the initial pre-processing steps, the given training dataset was read as a csv file, separating the data column-wise; taking the first column as the image name, and the corresponding values as the coordinates of the bounding box (desired output). The corresponding images were opened using opencv and os libraries and the images were modified as per need.

Then, we defined a yolo based model that functions in an identical manner, but being user defined, it corresponds to the given image dataset. Older object detection algorithms utilise classifiers to perform detection. They apply the model to an image at multiple locations to find the location of the object. High scoring regions of the image are considered to be successfully detected regions. In our approach, we have applied a single neural network to the entire image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. The predictions are informed by the global context in the image.

We reframe the idea of object detection in an image as a regression problem; going straight from image pixels to the bounding box coordinates. Our model trains on 64x64 resolution images with 3 channels (RGB) and directly. Our model implicitly encodes contextual information about the bounding box and learns generic representations of objects.

Our model not only helps in better and more efficient detection of objects in an image, but it also learns to predict the object bounding coordinates at a much faster rate.

**Libraries used:**

**Pandas**

We have used this library to read each row of the training csv file.

**Pickle**

This library helps us to save the loaded images in a retrievable format for training.

**Os**

Once we take the name of the image from the training csv file, we make use of the os module to refer to the corresponding file in the images folder.

**Opencv**

OpenCV is the leading open source library for computer vision, image processing and machine learning. In our model, we have used this library to open the images and perform various operations on the images like resizing.

**Keras**

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function.

**Numpy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.