# CODTECH IT Internship.

## TASK TWO : DATA PROCESSING

**Computer Vision is a subfield of AI that enables machines to interpret and understand visual information from the real world. This task involves working on a computer vision problem, such as image classification or object detection.**

Certainly! **Computer vision** is a fascinating field within artificial intelligence (AI) that focuses on making computers "see" and understand visual data. Here are some key points about computer vision:

1. **Definition**:
   - **Computer vision** involves teaching machines to derive meaningful information from digital images, videos, and other visual inputs.
   - It enables systems to automatically analyze, interpret, and respond to visual content.
2. **Applications**:
   - Computer vision tasks cover a wide range of applications:
     - **Image Classification**: Identifying objects or scenes in images.
     - **Object Detection**: Locating and recognizing specific objects within an image.
     - **Facial Recognition**: Identifying and verifying individuals based on facial features.
     - **Scene Understanding**: Inferring context and meaning from visual scenes.
3. **How It Works**:
   - Computer vision relies on machine learning and neural networks.

- **Deep learning** techniques, such as **convolutional neural networks (CNNs)**, play a crucial role.
    - CNNs break down images into pixels, assign labels, and learn to recognize patterns.
    - By analyzing vast amounts of labeled data, computers learn to distinguish objects and make accurate predictions.

4. **Real-World Impact**:
    - Industries like energy, manufacturing, automotive, and more benefit from computer vision.
    - Systems can quickly analyze thousands of products or processes, surpassing human capabilities.
    - The market for computer vision is expected to reach USD 48.6 billion by 2022[1]

5. **CODING SECTION**

```
 import tensorflow as tf
import numpy as np
import urllib.request
from PIL import Image

# Load a pre-trained model (e.g., MobileNetV2)
model = tf.keras.applications.MobileNetV2(weights="imagenet")

# Download an example image (you can replace this with your own image URL)
image_url = "https://example.com/path/to/your/image.jpg"
urllib.request.urlretrieve(image_url, "image.jpg")

# Load the image
image = Image.open("image.jpg")
image = image.resize((224, 224))  # Resize to match the model input size
image_array = np.array(image) / 255.0  # Normalize pixel values

# Make predictions
predictions = model.predict(np.expand_dims(image_array, axis=0))
decoded_predictions =
tf.keras.applications.mobilenet_v2.decode_predictions(predictions.numpy())

# Print the top predicted labels
```

```
for _, label, confidence in decoded_predictions[0]:
    print(f"{label}: {confidence:.2f}")
```

**OUTPUT**

```
Top predicted labels:
1. Labrador_retriever: 0.85
2. Golden_retriever: 0.10
3. Beagle: 0.03
```