

**EX.NO.:** 09

**DATE:** 31.07.2025

## DEEP LEARNING ALGORITHM FOR FACE DETECTION

### AIM

To detect and align animals from a burst-mode sequence of images using a deep learning model (YOLOv8) and generate a flip-book style animation to visualize the motion or action sequence of the animal.

### ALGORITHM

1. Load a pre-trained YOLOv8 model for object detection.
2. For each image in the burst-mode sequence:
  - a. Detect animals using the YOLOv8 model.
  - b. Extract the bounding box of the detected animal.
  - c. Crop and resize the detected animal region to a standard size.
3. Store the aligned animal images.
4. Generate a GIF animation (**flip-book style**) from the aligned images.

### CODE AND OUTPUT

```
import os
import cv2
import imageio
import numpy as np
from PIL import Image
from ultralytics import YOLO

# Load YOLOv8 model (you can use yolov8n.pt for a lightweight model)
model = YOLO('yolov8n.pt') # Make sure this file is downloaded, or let ultralytics
auto-download

# Folder paths
input_folder = 'burst_images' # Folder with burst-mode images of animals
output_folder = 'aligned_animals'
os.makedirs(output_folder, exist_ok=True)

aligned_images = []

# Animal classes in COCO dataset
animal_classes = {
    15, 16, 17, 18, 19, 20, 21, 22 # dog, horse, sheep, cow, elephant, bear, zebra,
giraffe
}

# Loop through each image
for filename in sorted(os.listdir(input_folder)):
    if not filename.lower().endswith(('.jpg', '.png', '.jpeg')):
        continue

    img_path = os.path.join(input_folder, filename)
    results = model(img_path)[0]
```

```

# Find the first animal detection
found = False
for box in results.bboxes:
    cls_id = int(box.cls)
    if cls_id in animal_classes:
        found = True
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        img = cv2.imread(img_path)
        cropped = img[y1:y2, x1:x2]

        # Resize to standard size
        resized = cv2.resize(cropped, (224, 224))
        aligned_images.append(resized)
        out_path = os.path.join(output_folder, filename)
        cv2.imwrite(out_path, resized)
        break

if not found:
    print(f"No animal detected in: {filename}")

# Create GIF
if aligned_images:
    gif_path = 'animal_animation.gif'
    imageio.mimsave(gif_path, [Image.fromarray(cv2.cvtColor(im, cv2.COLOR_BGR2RGB)) for
im in aligned_images], duration=0.3)
    print(f"GIF created: {gif_path}")
else:
    print("No animals were detected in any images.")

```

```

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_000.jpg: 384x640 2 birds, 1 dog, 185.3ms
Speed: 5.2ms preprocess, 185.3ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_001.jpg: 384x640 2 birds, 1 dog, 200.5ms
Speed: 2.7ms preprocess, 200.5ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_002.jpg: 384x640 2 birds, 1 dog, 204.1ms
Speed: 2.5ms preprocess, 204.1ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_003.jpg: 384x640 2 birds, 1 dog, 215.2ms
Speed: 2.6ms preprocess, 215.2ms inference, 2.0ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_004.jpg: 384x640 2 birds, 1 dog, 248.8ms
Speed: 2.9ms preprocess, 248.8ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_005.jpg: 384x640 1 bird, 1 dog, 200.1ms
Speed: 2.9ms preprocess, 200.1ms inference, 3.0ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_006.jpg: 384x640 1 bird, 1 dog, 189.9ms
Speed: 2.7ms preprocess, 189.9ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_007.jpg: 384x640 1 bird, 1 dog, 1 surfboard, 206.8ms
Speed: 3.7ms preprocess, 206.8ms inference, 2.3ms postprocess per image at shape (1, 3, 384, 640)
...

image 1/1 d:\TARUUV th year\Computer Vision Lab\Ex9\burst_images\frame_030.jpg: 384x640 2 birds, 1 dog, 197.2ms
Speed: 3.6ms preprocess, 197.2ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)
GIF created: animal_animation.gif

```

## INFERENCE

The system successfully detects and aligns animals across sequential images captured in burst mode. The generated flip-book animation demonstrates smooth movement of the animal, effectively visualizing motion through deep learning-based object detection.