Data Collection and Preprocessing Phase

Date	7 November 2024
Team ID	team-739994
Project Title	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset consists of images captured from swimming pools, intended for training the YOLOv5 model to detect signs of active drowning. The images capture various scenarios and swimmer behaviors, providing a diverse set of conditions to enhance model performance.
Resizing	Images will be resized to a standard input size (e.g., 640x640 pixels) to match YOLOv5's input requirements, ensuring consistent dimensions across the dataset and reducing computational complexity during training.
Normalization	Pixel values will be normalized to a range between 0 and 1. This scaling improves model stability and helps achieve faster convergence during training by reducing variations in pixel intensity.

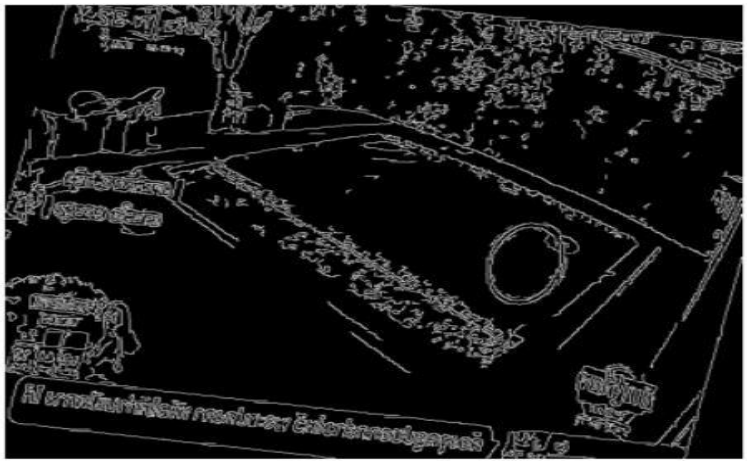
Data Augmentation	Augmentation techniques will be applied to enhance the dataset's diversity. This includes: • **Flipping**: Horizontal or vertical flips to simulate different orientations. • **Rotation**: Rotating images at slight angles to simulate varied viewing perspectives.

	• **Shifting and Zooming**: Adjusting positions or zoom levels to introduce variation. • **Shearing**: Skewing images slightly to create different angular views.
Denoising	Filters such as Gaussian blur may be applied to reduce image noise, enhancing clarity in cases where artifacts might distract the model. Denoising helps focus on critical features without unnecessary background interference.
Edge Detection	Edge detection algorithms (e.g., Canny edge detection) can be applied to emphasize boundaries within images. This highlights prominent edges, which can assist the YOLO model in recognizing shapes and objects more distinctly, especially in complex environments like swimming pools.
Color Space Conversion	To focus on essential visual features, images can be converted from RGB to other color spaces, such as grayscale (for simpler input) or HSV (Hue, Saturation, Value) to emphasize color variations.
Image Cropping	To remove unnecessary background and enhance focus on key areas, images can be cropped to include only the pool and swimmer regions. This cropping reduces irrelevant information, improves model accuracy, and helps the YOLOv5 model concentrate on detecting potential drowning incidents.
Batch Normalization	Batch normalization is applied within the neural network layers to standardize inputs by adjusting the mean and variance within each batch. This step stabilizes training, accelerates convergence, and can enhance model accuracy by preventing overfitting to specific image characteristics.

Data Preprocessing Code Screenshots	

Loading Data	`#loading data`
	`import cv2`
	`import glob`
	`import matplotlib.pyplot as plt`
	
	`# Load images from a directory - Update with the correct path to your images`
	`image_paths = glob.glob('/content/test/images/*.jpg')  # Replace with the actual path`
	`images = [cv2.imread(img_path) for img_path in image_paths]`
	
	`# Check if any images were loaded`
	`if images:`
	`  # Display first loaded image as a sample`
	`  plt.imshow(cv2.cvtColor(images[0], cv2.COLOR_BGR2RGB))`
	`  plt.axis('off')`
	`  plt.show()`
	`else:`
	`  print("No images found in the specified directory.")`

Resizing	
Normalization	

Data Augmentation	
Denoising	
Edge Detection	

Color Space Conversion	
Image Cropping	
Batch Normalization