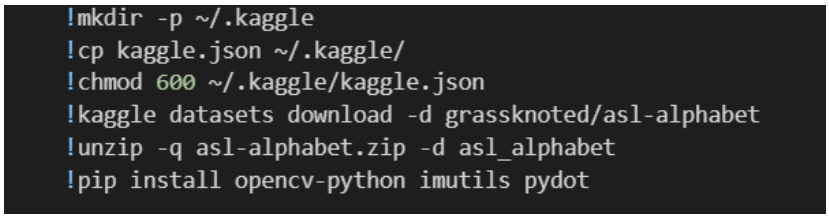
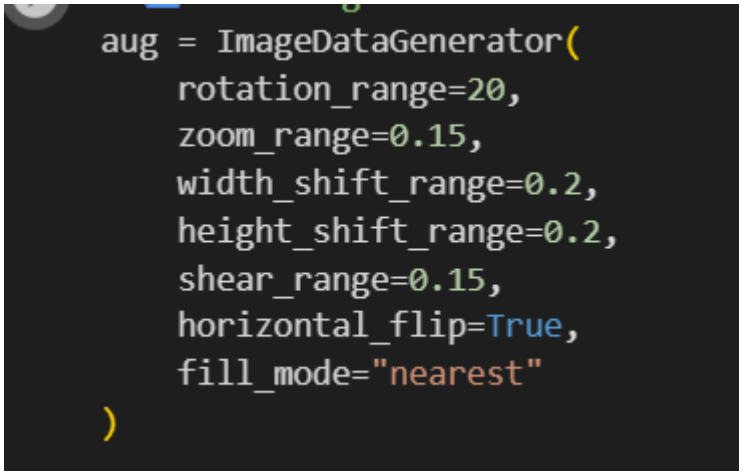
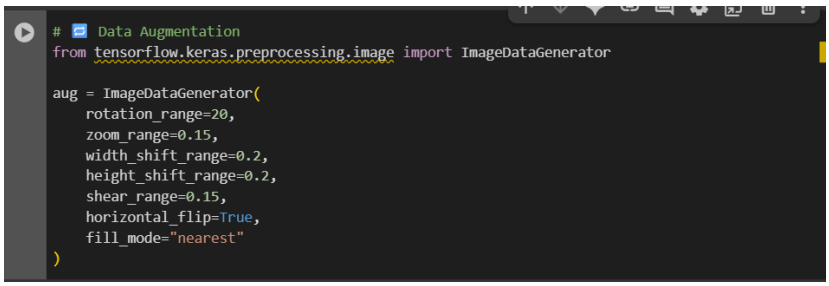
Data Collection and Preprocessing Phase

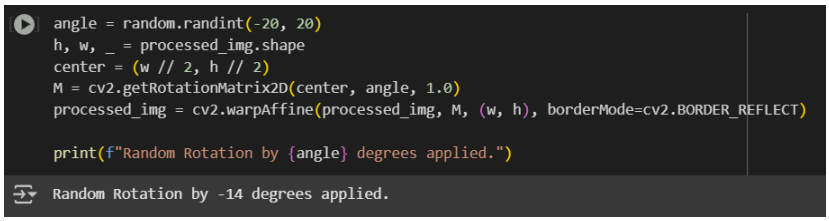
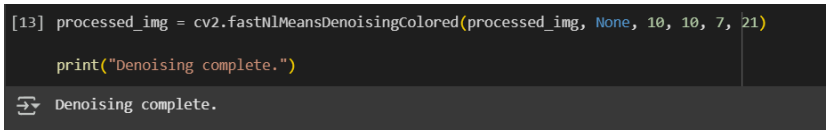
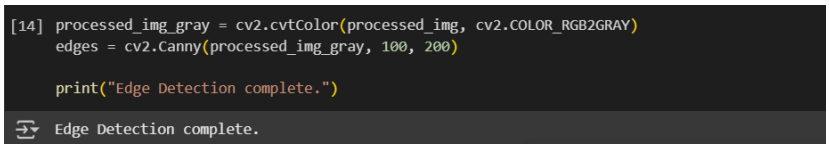
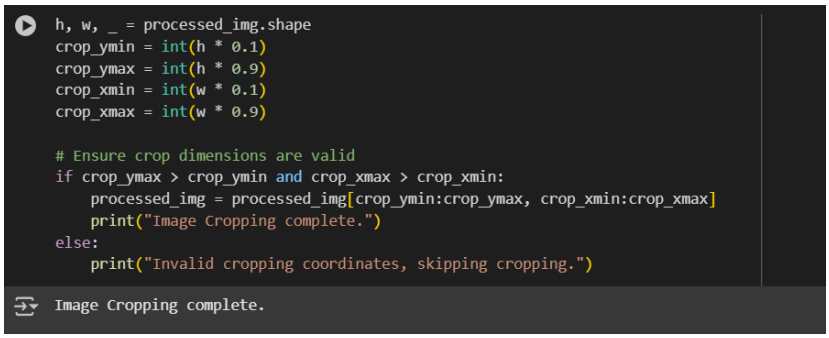
Date	20 June 2025
Team ID	SWTID1750316859
Project Title	ASL - Alphabet Image Recognition
Maximum Marks	6 Marks

Preprocessing:

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	Total number of images in the dataset: 87000.
Resizing	All images are resized to a target size of 128x128 pixels.
Normalization	Pixel values are rescaled from the range [0, 255] to [0, 1] by dividing by 255.
Data Augmentation	Randomly rotated images by up to 20 degrees and zoomed in on images randomly by upto 15%.
Denoising	Used cv2.fastNlMeansDenoisingColored to perform non- local means denoising.
Edge Detection	Used cv2.Canny to find edges. The parameters 100 and 200 are the lower and upper hysteresis thresholds. Edges with intensity gradient more than the upper threshold are sure to be edges, and those below the lower threshold are sure not to be edges. Those between these two thresholds are classified based on their connectivity.

Color Space Conversion	Converted both the original and cropped images from BGR to RGB for correct display with Matplotlib.
Image Cropping	Crop images to focus on the regions containing objects of interest.
Batch Normalization	In TensorFlow/Keras, batch normalization is implemented as a dedicated layer (tf.keras.layers.BatchNormalization). Adding this layer directly to the model architecture handles the normalization process automatically during training and inference, making manual implementation for each batch unnecessary.
Data Preprocessing Code Screenshots	
Loading Data	
Resizing	
Normalization	

Data Augmentation	
Denoising	
Edge Detection	
Color Space Conversion	
Image Cropping	
Batch Normalization	In TensorFlow/Keras, batch normalization is implemented as a dedicated layer (tf.keras.layers.BatchNormalization). Adding this layer directly to the model architecture handles the normalization process automatically during training and inference, making manual implementation for each batch unnecessary.