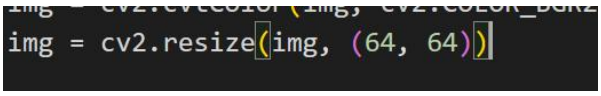
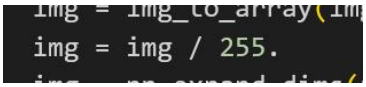
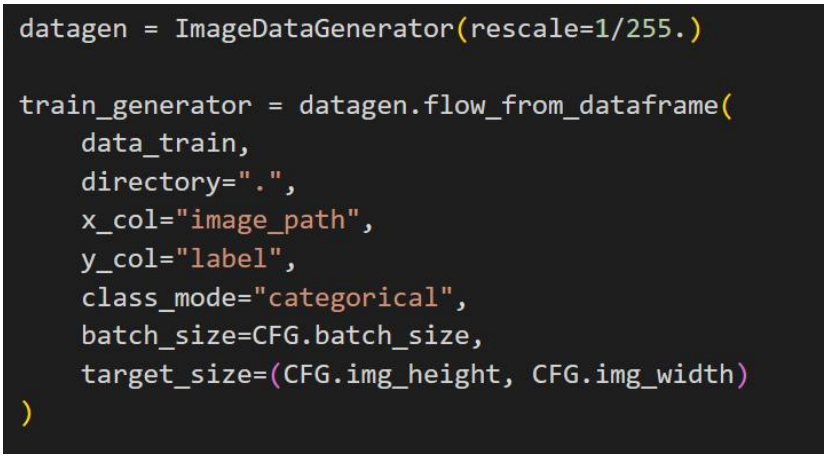
Data Collection and Preprocessing Phase

Date	15 June 2025
Team ID	SWTID1750006853
Project Title	ASL- Alphabet Image Recognition
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 used is the **ASL Alphabet Dataset** from Kaggle. It contains approximately 87,000 images categorized into 29 classes: 26 alphabets (A–Z) and 3 special symbols (del, nothing, and space). Each image represents a static hand gesture and is stored in JPEG format.
Resizing	All images are resized to a uniform size of 64x64 pixels to ensure consistent input dimensions for the neural network.
Normalization	Pixel values are scaled to the range [0, 1] by dividing by 255. This helps stabilize and speed up training.
Data Augmentation	Augmentation includes rescaling, flipping, and slight transformations to artificially increase the dataset size and improve model generalization.
Denoising	Denoising can be applied using filters like Gaussian blur or Non-local Means Denoising.
Edge Detection	Edge detection is useful for highlighting gesture outlines.

Color Space Conversion	Images are converted from BGR to RGB color space for compatibility with deep learning models trained on RGB inputs.
Image Cropping	Crop images to focus on the regions containing objects of interest.
Batch Normalization	Apply batch normalization to the input of each layer in the neural network.
Data Preprocessing Code Screenshots	
Loading Data	
Resizing	
Normalization	
Data Augmentation	


```
img = cv2.resize(img, (64, 64))
```

```
img = img / 255.
```

```
datagen = ImageDataGenerator(rescale=1/255.)

train_generator = datagen.flow_from_dataframe(
    data_train,
    directory=".",
    x_col="image_path",
    y_col="label",
    class_mode="categorical",
    batch_size=CFG.batch_size,
    target_size=(CFG.img_height, CFG.img_width)
)
```

Denoising	Give the code snippet as an image (copy and paste the picture in this block).
Edge Detection	Give the code snippet as an image (copy and paste the picture in this block).
Color Space Conversion	`img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)`
Image Cropping	Give the code snippet as an image (copy and paste the picture in this block).
Batch Normalization	Give the code snippet as an image (copy and paste the picture in this block).