Q3. Short Video Classification

The algorithm is divided into two parts. The first part uses OpenCV to convert videos into images, and the second part uses YOLOv8 to classify images.

Frame extraction

1. Import train_tag.txt as DataFrame.

2. Create a series of folders saving images.

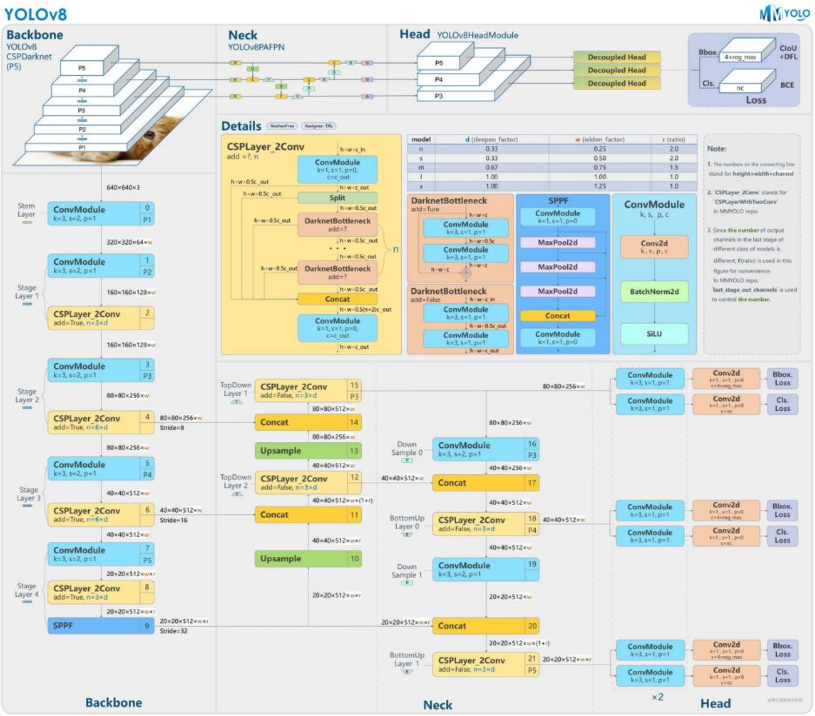
3. Sample each video every twenty frames and save it to the corresponding folder.
   We can easily find that many videos have identical icons and animations in the second half, so we only operate on the first half of the video.

4. Use the OS library to construct a validation set suitable for YOLO format.
   We randomly select 10% of the images as the validation set to verify whether our model is running properly.

Image classification

1. Import pre trained YOLOv8 classification model

   ### Overall architecture:

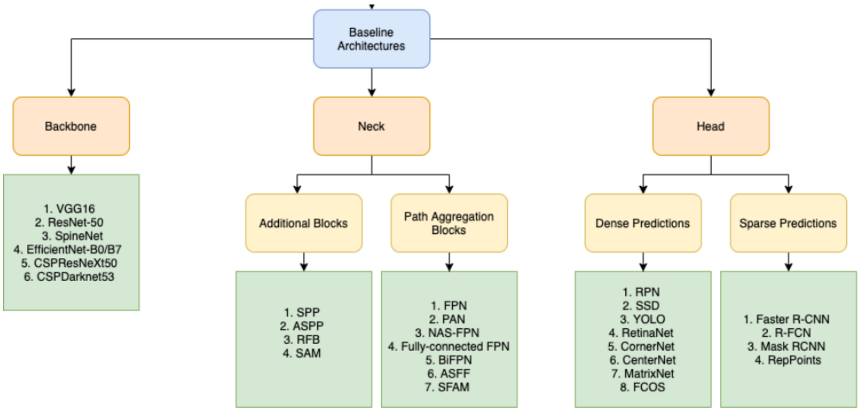


Backbone: The feature extractor is a network that extracts features from images, and its function is to extract information from the images for use by subsequent networks

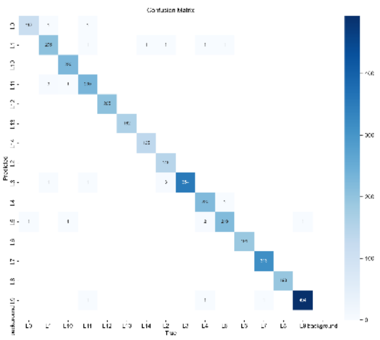
Neck: It is placed between the backbone and head to better utilize the features extracted by the backbone, playing a role in "feature fusion".

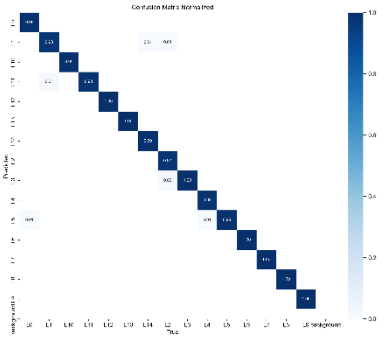
Head: Use the previously extracted features to make recognition

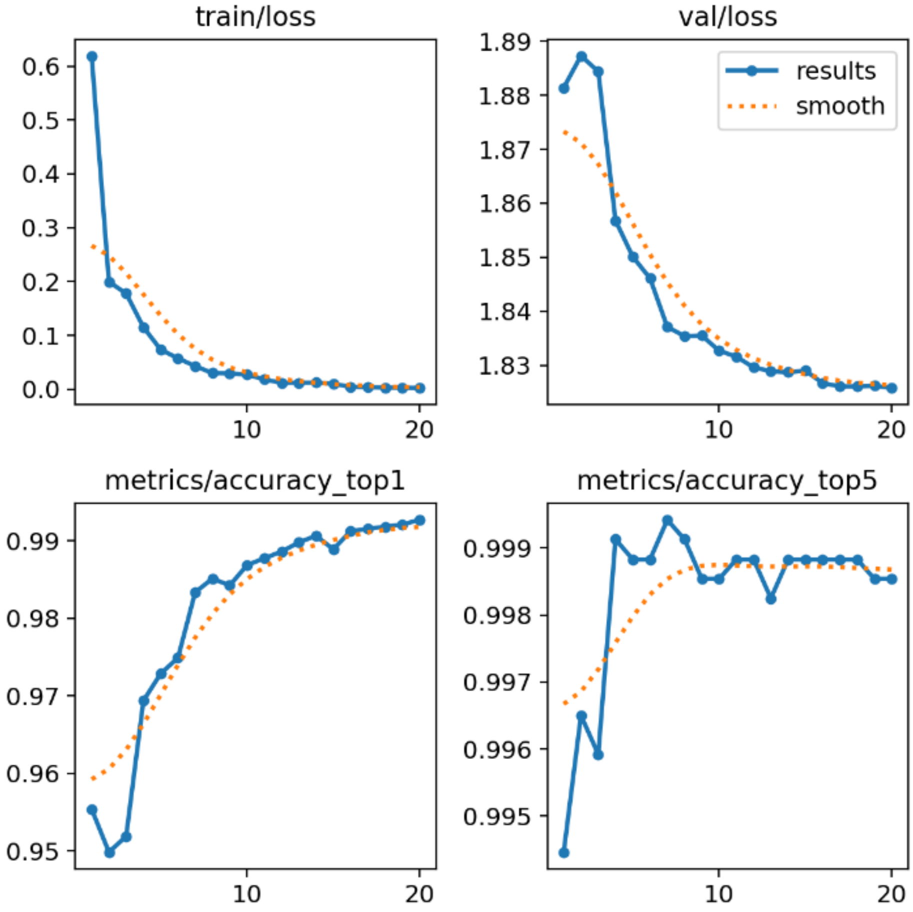
Common Backbone, Neck, and Head networks:



2. Fine-tuning 20 epochs







3. Predicting test videos

Using the same method as above to obtain a set of images, the most predicted labels will become the labels of the video.

4. Output results