# Load video data

Here’s a short and simple guide to the **UCF101 Video Classification Project** for certification/interview purposes:

**Key Points to Know:**

1. **What is UCF101?**
   * **UCF101** is a dataset containing 101 categories of human actions, widely used for action recognition tasks.
   * The dataset includes video clips with different human activities like playing soccer, applying makeup, etc.
2. **Data Preprocessing:**
   * **Extract Frames**: Each video is converted into individual frames to feed into the model.
   * **Frame Selection**: A specific number of frames (n\_frames) is selected from each video to maintain uniformity for training.
   * **Resizing & Padding**: Each frame is resized to a fixed output size (e.g., 224x224) and padded to ensure consistent dimensions.
3. **Frame Generator for Video Data:**
   * A custom **FrameGenerator** class is used to convert video files into frames.
   * The generator yields frames and labels in batches, which is necessary for training deep learning models with video data.
4. **TensorFlow Data Pipeline:**
   * **tf.data.Dataset** is used to create a data pipeline for efficient input feeding into the model.
   * **Prefetching** and **caching** are used to optimize data loading and prevent I/O bottlenecks during training.
5. **Model Setup:**
   * A pre-trained **EfficientNetB0** model is used for feature extraction from video frames, followed by a dense layer and pooling for action classification.
   * **TimeDistributed Layer**: Used to apply a model to each frame of a sequence, which is important for video data (since each video is a sequence of frames).
6. **Training & Evaluation:**
   * The dataset is split into **train**, **validation**, and **test** sets.
   * **EarlyStopping** is used to prevent overfitting by stopping training if the validation loss doesn't improve after a set number of epochs.
7. **Project Workflow:**
   * **Data Loading**: Videos are downloaded and split into frames.
   * **Model Training**: A deep learning model is trained using these frames to recognize actions.
   * **Evaluation**: The model is evaluated on the test data to check its performance.

**Must-Know Concepts:**

* **Video Classification**: Converting video data into frames, handling sequence data, and using CNN-based models.
* **EfficientNet**: A pre-trained model used for feature extraction in video classification tasks.
* **TimeDistributed Layer**: Applies a model to every frame in a video, useful for video classification tasks.

These points cover the essential aspects of video data preprocessing and training for action recognition tasks using the UCF101 dataset in TensorFlow.