Q2

A data processing pipeline can be thought of as a collection of connected stages that work together to process data sequentially or in parallel, depending on their dependencies and resource needs. Each node in the pipeline may be thought of as a processing step, and each edge can be thought of as a data flow between the stages. The pipeline's overarching purpose is to convert input data into the desired output while maximizing performance and scalability.

At a lower level, a distributed system architecture can be used to build a data processing pipeline in which numerous computing nodes collaborate to process data simultaneously. Data is exchanged between nodes via a message system or standard data storage as each node completes a distinct processing phase. The nodes can be dynamically scaled up or down depending on the workload and available resources.

Consider this straightforward data pipeline for picture categorization as an illustration:

Step 1: Input Data: a group of photographs that need to be categorized.
Step 2: Preprocessing: Resize and normalize the photos to a standard format.
Step 3: Feature extraction: Using a convolutional neural network that has already been trained, extract features from the images.
Step 4: Classification: Sort the photos into groups using a linear SVM classifier.
Step 5: Output: A list of category names for each input image

Steps 2 and 3 in this pipeline can be completed concurrently because they are independent of one another. It is necessary to complete steps 4 and 5 in order since step 4 needs the output from step 3 as an input and step 5 needs the output from step 4.

At a lower level, we can build this pipeline using a distributed system architecture, which processes the input photos concurrently across numerous processing nodes. Each batch of the input photos is handled by a different node after being separated into smaller halves. It is possible to combine Nodes 2 and 3 into a single processing node that receives a batch of photos as input and outputs a batch of feature vectors. Nodes 4 and 5 identify the photos and generate the final output after receiving this node's output.