APPROACH

- **FashionSnap: Real-Time Dress Classification Using FashionMNIST**

TEAM MEMBERS

1.Rishi Bose .S

2.Sai Charan

3.Mohamed Yaseer.M

Steps taken to achieve the solution for the given problem statement*:*

- ***Dataset Preparation***
- ***Model Selection***
- ***Model Training***
- ***Model Evaluation***
- ***Camera Integration***
- ***Classification***

***Dataset Preparation*:**

Obtain the Fashion MNIST dataset, which consists of grayscale images of fashion items divided into 10 classes

Split the dataset into training and testing sets,typically using an 80:20 or 70:30 ratio

Model Selection:

Choose an appropriate deep learning architecture for dress classification. In this case, we select the FashionSnap model, which is based on Convolutional Neural Networks (CNNs).

Consider the number of convolutional layers, filter sizes, pooling layers, and fully connected layers based on the complexity of the dataset.

Model Training:

Preprocess the images by resizing them to a suitable input size, converting to grayscale, and normalizing the pixel values to the range [0, 1].

Split the training set further into training and validation sets for model evaluation during training.

Configure the FashionSnap model with the desired architecture and compile it with an appropriate loss function (e.g., categorical cross-entropy) and optimizer (e.g., Adam).

Train the model using the training set, iterating over the data for a specified number of epochs. Monitor the validation loss and accuracy to assess the model's performance and prevent overfitting.

Fine-tune the model *Evaluate the trained FashionSnap model using the testing set to assess its generalization performance.*

Model Evaluation:

Evaluate the trained FashionSnap model using the testing set to assess its generalization performance.

Calculate evaluation metrics, such as accuracy, precision, recall, and F1-score, to measure the model's effectiveness in classifying dress images.

Generate a confusion matrix to visualize the model's performance across different dress categories

Camera Integration:

Utilize computer vision libraries like OpenCV to integrate the camera functionality into the dress classification

Capture frames from the camera in real-time and preprocess them using the same preprocessing steps as during training.Feed the preprocessed frames into the trained FashionSnap model to obtain predictions for the dress category.

Overlay the predicted dress category on the captured frame and display it to the user.

Classification:

The model predicts the probability of the input image belonging to each class. The class with the highest probability is selected as the predicted class for the input image