Task 2

To develop an age and gender detection model for the horror roller coaster task, we can break down the project into several key steps:

**Step 1: Dataset Collection and Preprocessing**

1. **Data Collection**:
   * We'll need a dataset containing images labeled with the age and gender of the individuals. Common datasets for this purpose include the Adience, UTKFace, or IMDB-WIKI datasets.
2. **Data Preprocessing**:
   * Resize images for consistent input dimensions (e.g., 224x224 pixels).
   * Normalize pixel values (e.g., scaling pixel values between 0 and 1).
   * Augmentation techniques (e.g., rotations, flips) can be applied to increase dataset diversity.

**Step 2: Model Selection and Training**

1. **Model Architecture**:
   * A convolutional neural network (CNN) is ideal for image-based tasks like this. Transfer learning using pre-trained models (e.g., VGG16, ResNet50) can be used to enhance performance and reduce training time.
   * For the output layer:
     + Age can be a regression task (predicting continuous age).
     + Gender can be a classification task (male/female).
2. **Training the Model**:
   * Split the dataset into training, validation, and test sets.
   * Train the model using appropriate loss functions (Mean Squared Error for age and Binary Cross-Entropy for gender).
   * Regularization techniques (e.g., dropout) to prevent overfitting.

**Step 3: Post-Processing and Business Logic**

1. **Age and Gender Detection**:
   * Implement the model's prediction logic, which detects age and gender in real-time.
   * If age is below 13 or above 60, display the "Not allowed" message and mark with a red rectangle.
2. **Data Storage**:
   * Store the detected age, gender, and entry time in an Excel or CSV file using libraries like pandas.

**Step 4: Implementation and Integration**

1. **Real-Time Video Processing**:
   * Capture video input using OpenCV.
   * Apply the model to each frame to detect faces and predict age/gender.
   * Overlay results on the video stream with bounding boxes and the "Not allowed" message where appropriate.
2. **Data Export**:
   * Implement the logic to save each detection’s information to a CSV/Excel file.

**Step 5: (Optional) GUI Development**

* If you choose to develop a GUI, use Tkinter or a similar Python framework to create an interface that displays the video stream, provides controls for starting/stopping detection, and displays stored data.

**Model Weights and Saved Models**

After training, the models will be saved as age\_model.h5 and gender\_model.h5. These files might be large, so you can upload them to Google Drive and share the links.

**Google Drive Links (Example):**

* Age Model Weights
* Gender Model Weights

**Note:** You will need to replace the example links with the actual links after uploading your models.

**Model Performance Evaluation**

In your Jupyter Notebook (train\_model.ipynb), you should evaluate your models' performance using the following:

* **Accuracy**: For the gender detection model.
* **Mean Absolute Error (MAE)**: For the age detection model.
* **Confusion Matrix**: To visualize how well the gender detection model is performing.