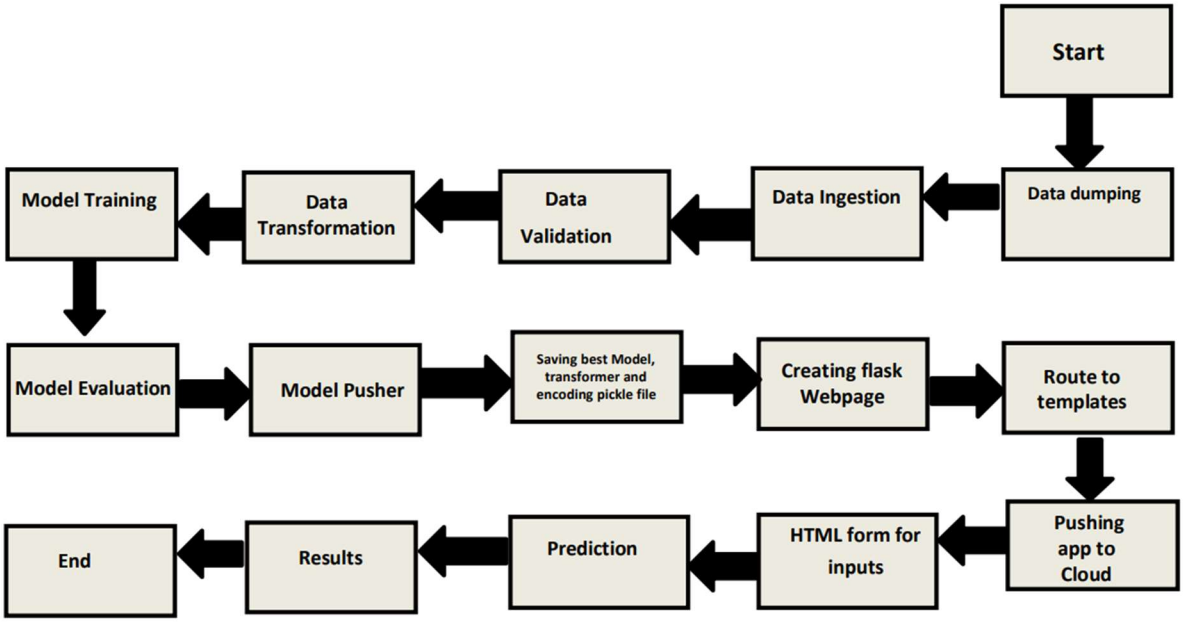
Architecture

Face image BMI prediction

Date Issued	Version	Description	Author
23-03-2024	1	Architecture-V1.0	Ranjith V

Contents

3. Architecture



4. Architecture Description

1. Client-Side (Frontend)

- **Technologies**: HTML, CSS, JavaScript, Bootstrap
- **Components**:
  - **HTML Form**: Allows users to upload a face image for BMI prediction.
  - **Bootstrap**: Provides responsive and styled UI components for a better user experience.
  - **JavaScript**: Handles form submission, makes an HTTP POST request to the Flask API, and displays the prediction result.

2. Server-Side (Backend)

- **Technologies**: Flask, Python
- **Components**:
  - **Flask API**:
    - **Endpoint**: /predict
    - **Function**: Receives image files from the client, processes them, and returns the BMI prediction.
    - **Processing**:
      - **Image Handling**: Decodes and preprocesses the uploaded image.
      - **Model Inference**: Uses a pre-trained Convolutional Neural Network (CNN) model to predict BMI from the processed image.
      - **Response**: Sends the BMI prediction or an error message back to the client.

3. Model

- **Technologies**: Keras/TensorFlow
- **Components**:
  - **Convolutional Neural Network (CNN)**:
    - **Architecture**: Consists of convolutional layers, pooling layers, and fully connected layers to extract features and predict BMI.
    - **Training**: Trained on a dataset of face images with corresponding BMI values.
    - **Deployment**: Saved as a .h5 file and loaded by the Flask API for inference.

4. Data Flow

1. **User Interaction**: A user uploads a face image via the HTML form.

2. **Form Submission**: JavaScript captures the image file and sends it to the Flask API using an HTTP POST request.
3. **Image Processing**: Flask API receives the image, preprocesses it, and feeds it into the CNN model.
4. **Model Inference**: The CNN model processes the image and generates a BMI prediction.
5. **Response Handling**: Flask API sends the BMI prediction or an error message back to the client.
6. **Display Result**: JavaScript updates the HTML page with the prediction result or an error message.

5. Deployment

- **Technologies**: AWS/Azure/GCP (for cloud deployment), Docker (optional)
- **Components**:
    - **Hosting**: Deploy the Flask API on a cloud platform for accessibility.
    - **Scalability**: Ensure the solution can handle multiple requests and scale as needed.

6. Security and Error Handling

- **Security**: Ensure secure handling of image uploads and predictions.
- **Error Handling**: Provide meaningful error messages for issues during image processing or model inference.

This architecture ensures a modular and scalable approach for predicting BMI from face images, leveraging modern web technologies and cloud-based deployment.