Video-Based Human Activity Recognition

By:
K. Sai Nithin - 1005-21-729081
S. Vara Prasad - 1005-21-729083

TABLE OF CONTENTS

Problem Statement

Human activity recognition is a critical task in computer vision with applications in surveillance, healthcare, and sports analytics. Traditional methods often face challenges such as handling large-scale datasets with diverse actions and dynamic environments, high computational costs for real-time applications, and limited accuracy in recognizing overlapping or complex activities. Efficient feature extraction is essential to identify relevant patterns from image and video data without compromising performance, while ensuring generalization to unseen data and diverse scenarios. Moreover, achieving real-time predictions for dynamic video inputs requires lightweight yet robust models. The goal of this project is to develop an efficient and scalable Human Action Recognition (HAR) system capable of accurately classifying actions from images, recognizing action sequences in videos, and leveraging lightweight deep learning models for real-time applications.

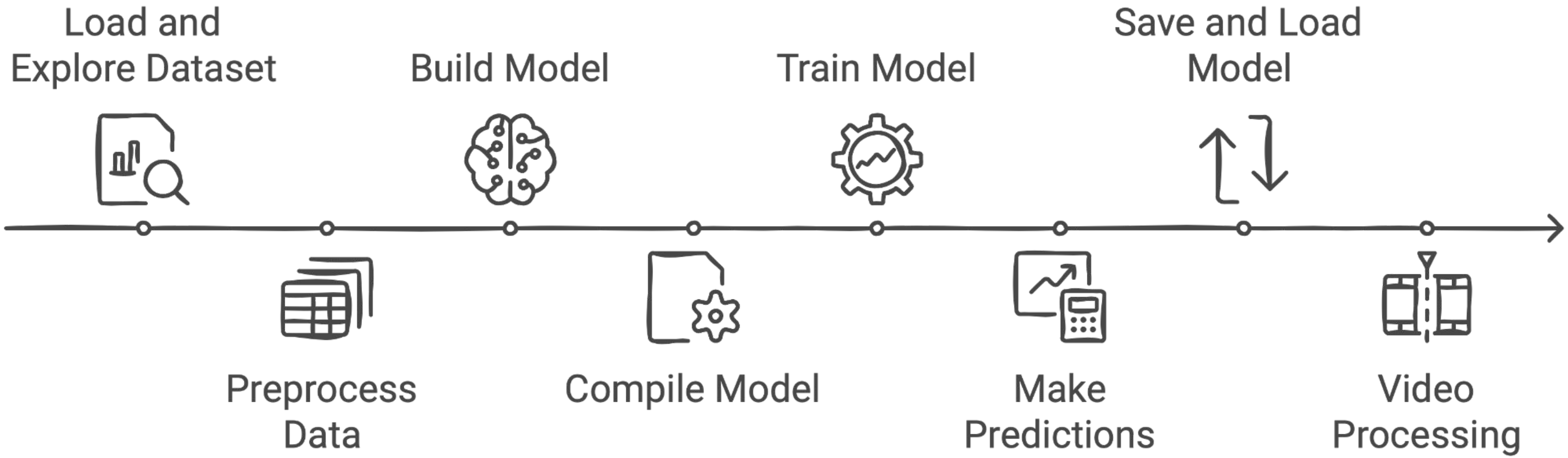
Dataset

The dataset used for this project focuses on Human Activity Recognition (HAR) and includes images and videos depicting a variety of human actions. Key features of the dataset include:

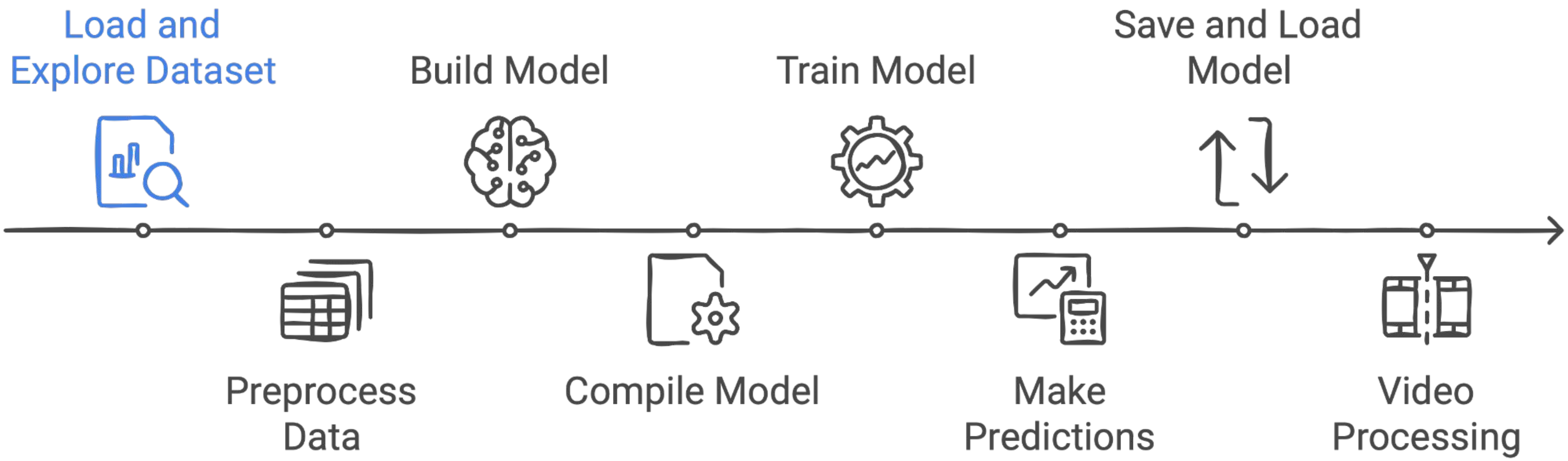
- The dataset features **15 different classes** of human activities, such as walking, sitting, standing, running, etc.
- It contains over **12,000 labeled** images, each categorized under one of the 15 activity classes.
- **Each image** belongs to a **single activity** class and is stored in a folder according to its **label** in the annotations (**csv**), ensuring organized and accessible data for model training after preprocessing.
- A distinct set of images and videos is used to evaluate the model's performance, ensuring it can generalize to unseen data.

This well-organized dataset is crucial for training the HAR model, enabling accurate recognition of human activities in both still images and video frames.

WorkFlow

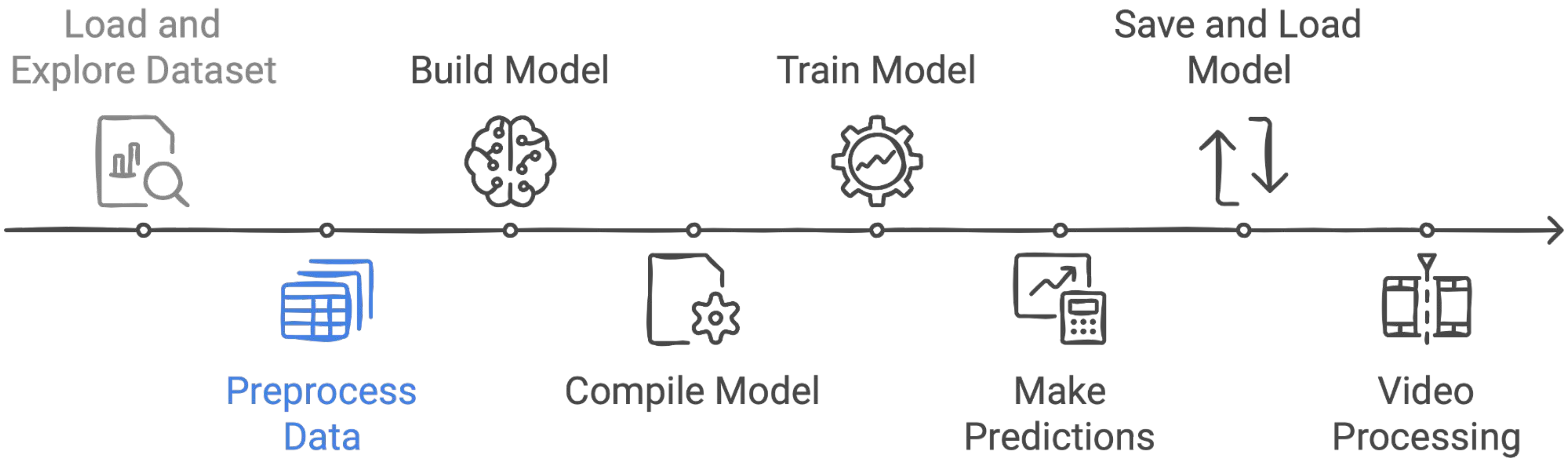

Load and
Explore Dataset
Build Model
Train Model
Save and Load
Model
Preprocess
Data
Compile Model
Make
Predictions
Video
Processing

WorkFlow



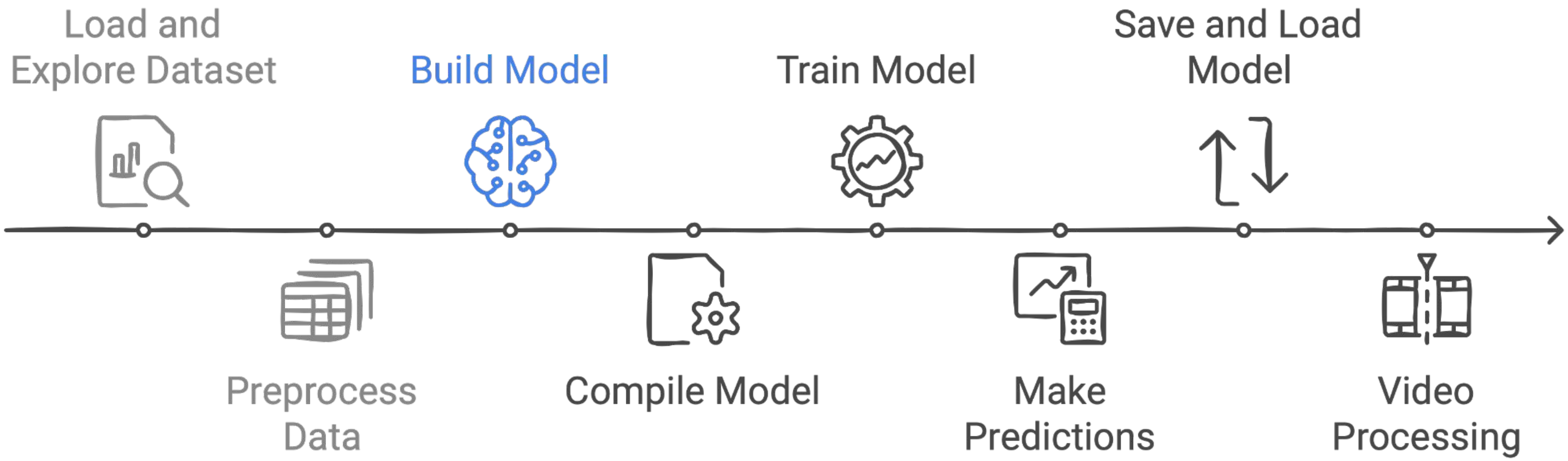
The dataset was loaded using pandas, with training and testing data accessed from structured folders. The class distribution was explored using value counts and visualized through a pie chart.

WorkFlow



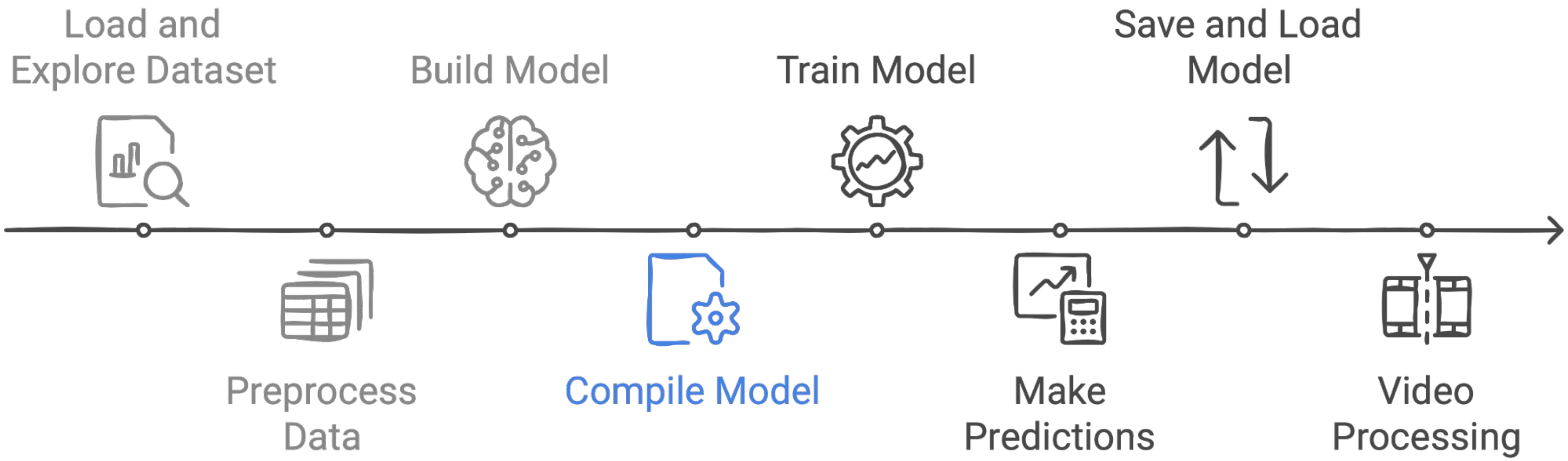
The images were resized to 160x160 pixels, and pixel values were normalized. Labels were encoded using one-hot encoding to prepare the data for model training.

WorkFlow



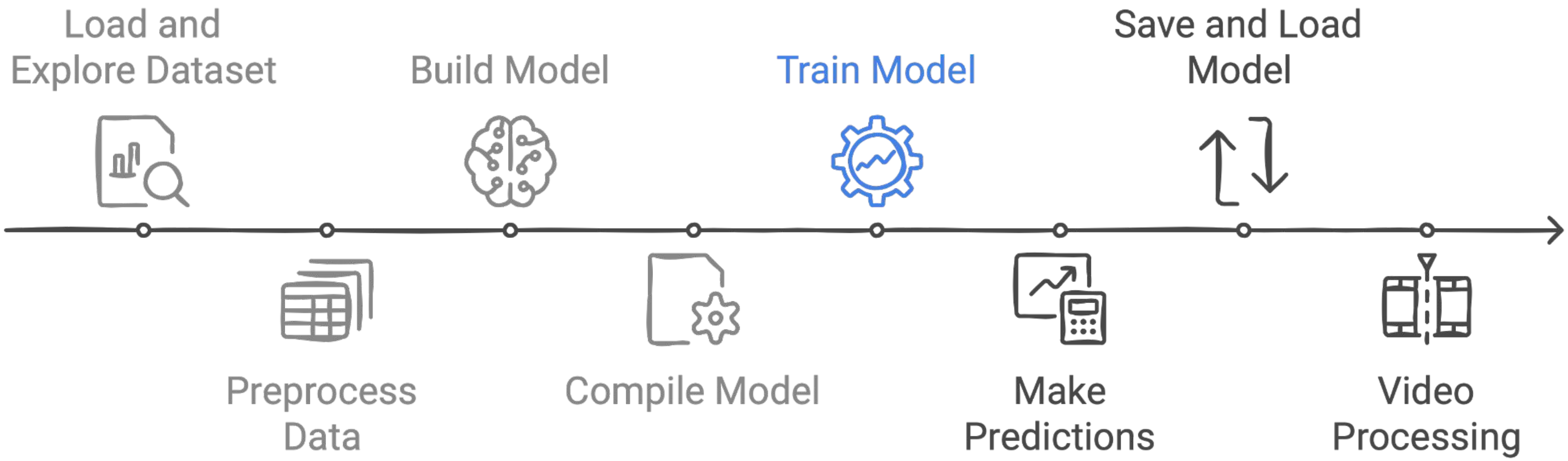
The model was built using MobileNetV3Large, a pre-trained deep learning model, with the top layers removed. It was fine-tuned by adding a flatten layer followed by dense layers to classify the 15 human activity classes.

WorkFlow



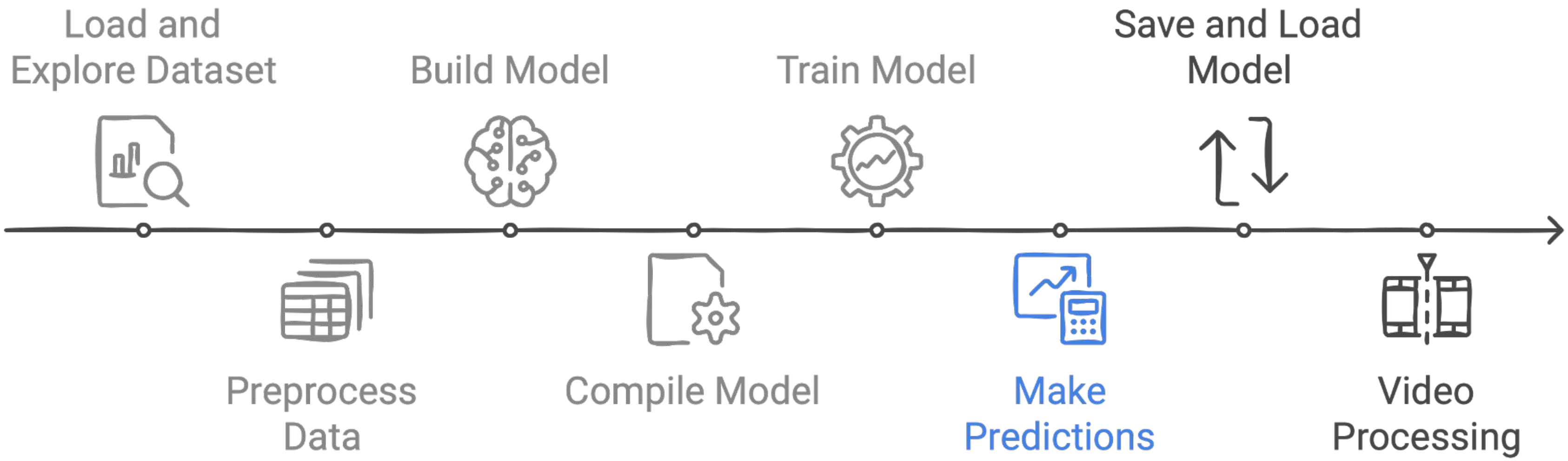
The model was compiled using the Adam optimizer, which adapts the learning rate during training for better convergence. The loss function used was categorical cross-entropy, suitable for multi-class classification tasks, and the accuracy metric was tracked to evaluate model performance during training.

WorkFlow



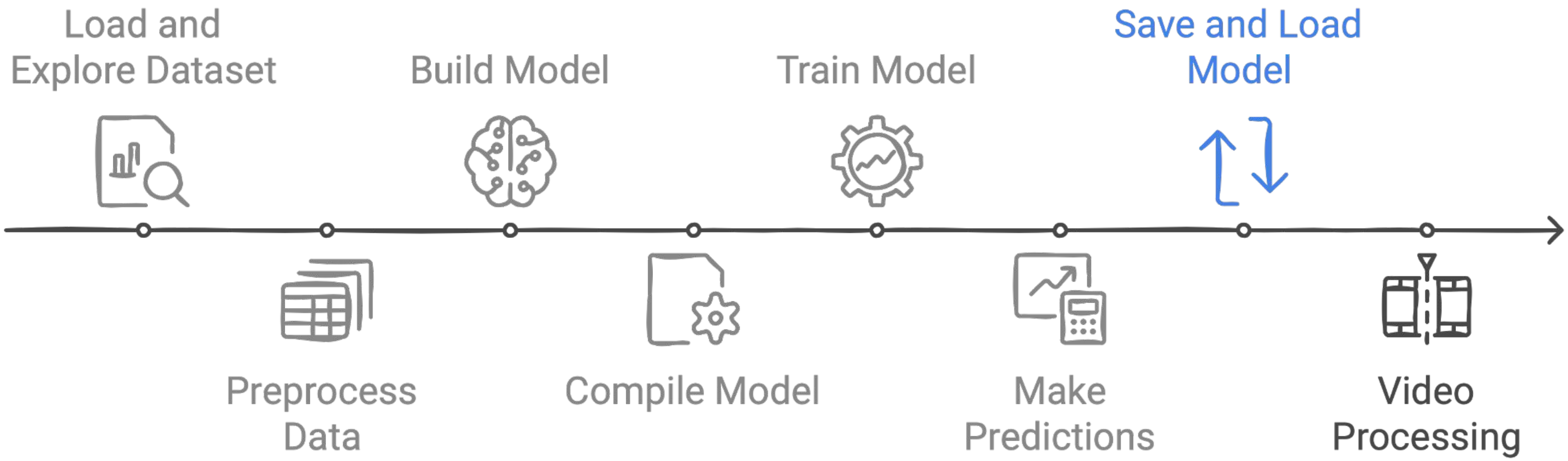
The model was trained using the preprocessed data, employing the Adam optimizer and categorical cross-entropy loss function. The training process involved fitting the model for 10 epochs, optimizing the weights to minimize loss and improve accuracy in recognizing human activities.

WorkFlow



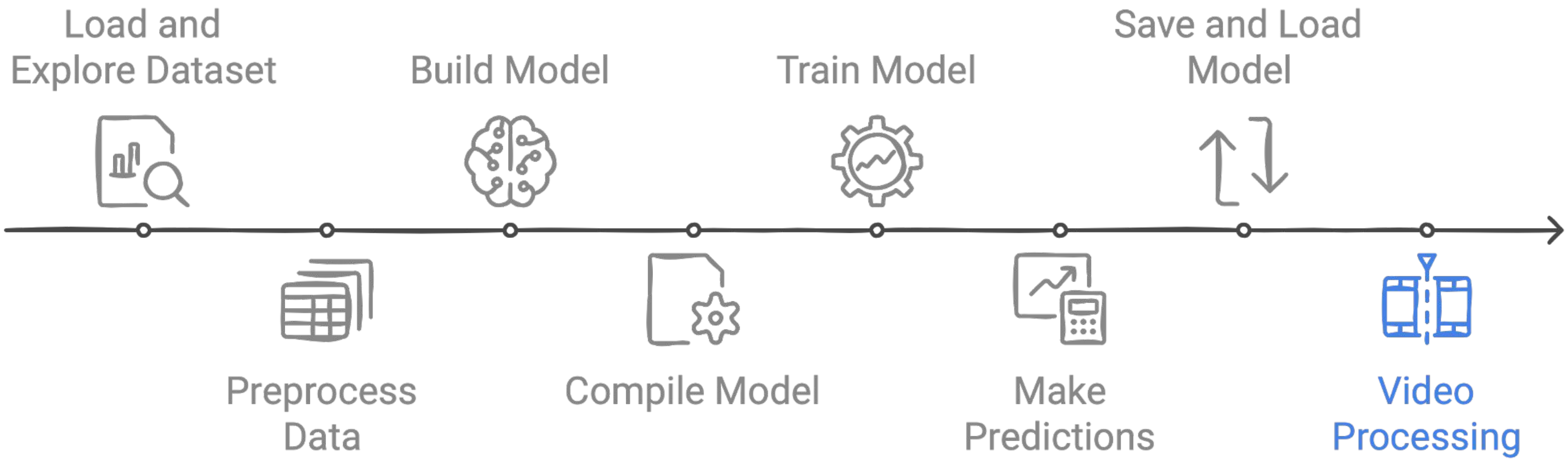
Predictions were made using the trained model on test images. For each image, the model's output probabilities were analyzed, and the class with the highest probability was selected as the predicted human activity. The confidence level of the prediction was also calculated to indicate the certainty of the model's output.

WorkFlow



The trained model was saved using the model.save() method to persist the model for future use. In this project, the model was saved as mobilenetv3large.keras. When needed for inference or further evaluation, the model can be loaded using the load_model() function from Keras, which loads the saved model back into memory for predictions.

WorkFlow

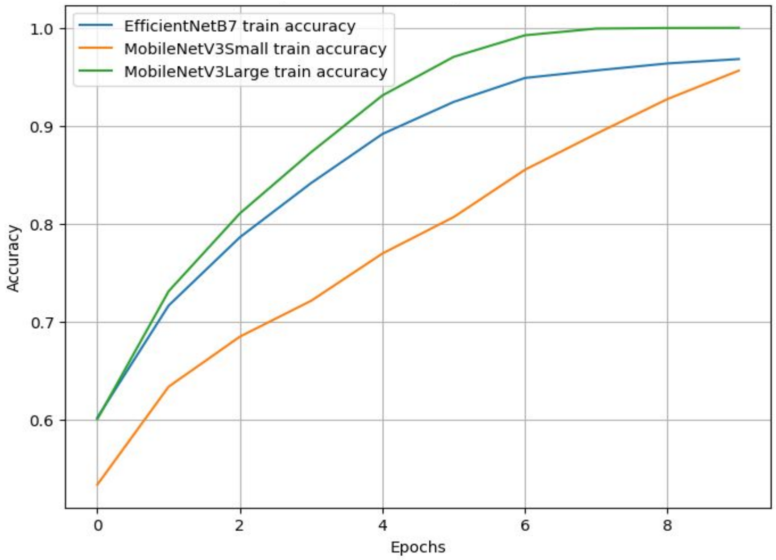


Video processing involves extracting and resizing frames from the input video, then passing them through the trained model to predict human actions. The action sequence is then generated based on the predictions from each frame.

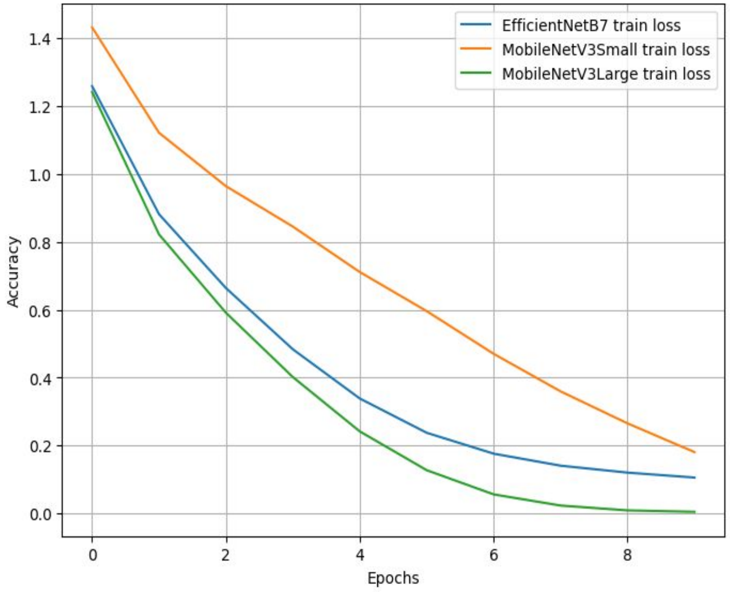
Experimentation

- The models tested include EfficientNetB7, MobileNetV3Small, and MobileNetV3Large.
- All models were trained on the same dataset of labeled images.
- During training, training accuracy and loss were tracked across 10 epochs for evaluation.
- The results showed that EfficientNetB7 had moderate accuracy with slower training time, MobileNetV3Small provided faster performance but with slightly lower accuracy compared to MobileNetV3Large, and MobileNetV3Large achieved the highest accuracy and best overall performance.
- Based on the results, MobileNetV3Large was selected for deployment due to its superior accuracy.

Experimentation



Comparison of Accuracy across Models



Comparison of Loss across Models

Results

Human Action Recognition

Upload Video: Choose File relaxed_wo...review.mp4 Submit

Results

Action Sequence: listening_to_music -> sleeping -> listening_to_music -> sleeping

Human Action Recognition

Upload Video: Choose File business_wo...review.mp4 Submit

Results

Action Sequence: drinking -> listening_to_music -> laughing -> drinking -> laughing -> drinking

Conclusion

Key Findings:

- MobileNetV3Large outperformed other models in terms of accuracy and training loss.
- Efficient performance on human action recognition tasks with high accuracy.

Model Integration:

- Integrated the trained MobileNetV3Large model into a Flask app for real-time video action prediction.

Future Work:

- Explore further improvements in model accuracy by tuning hyperparameters or experimenting with different architectures.
- Potential application in real-time action recognition for security and healthcare systems.

References

- Nagadia, M. (2024). *Human Action Recognition (HAR) Dataset*. Kaggle. Available at: https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset
- Keras Documentation. (2024). *Keras: The Python Deep Learning Library*. Available at: https://keras.io/
- TensorFlow Documentation. (2024). *TensorFlow: An Open Source Platform for Machine Learning*. Available at: https://www.tensorflow.org/
- OpenCV Documentation. (2024). *OpenCV: Open Source Computer Vision Library*. Available at: https://opencv.org/
- Zhang, Y., et al. (2020). *Human Activity Recognition: A Review*. Journal of Machine Learning Research, 21(1), 1-31.