Machine Learning Assignment

PROJECT REPORT

<TEAM ID : 29>

<PROJECT TITLE>:Real-Time Hand Gesture Recognition Using CNNs

Name	SRN
C Panshul Reddy	PES2UG23CS154
C Yogesh Reddy	PES2UG23CS159

Problem Statement

How can we break down communication barriers between the deaf/hard-of-hearing community and hearing individuals by creating an accessible, real-time technology solution that automatically recognizes American Sign Language gestures?

Our Solution:

Use transfer learning with MobileNetV2 to create a lightweight, accurate, real-time ASL recognition system that works on standard hardware

Objective / Aim

Primary Objective:

To develop an automated computer vision system that can accurately recognize American Sign Language (ASL) hand gestures in real-time, enabling seamless communication between deaf/hard-of-hearing individuals and hearing people.

Technical Aims:

1. **High Accuracy Recognition**
    - Achieve >95% classification accuracy on ASL hand gestures
    - Minimize misclassification between similar gestures
    - Ensure consistent performance across different users
2. **Real-time Performance**
    - Process gestures with <100ms latency for natural conversation flow
    - Maintain stable recognition during live video streaming
    - Enable smooth user interaction without delays
3. **Accessibility & Deployment**
    - Work with standard hardware (webcam, regular computer)
    - Create lightweight model suitable for mobile devices
    - Develop user-friendly interface requiring minimal training
4. **Scalable Architecture**
    - Build foundation that can expand from 6 letters to full ASL alphabet

- Design system capable of adding dynamic gesture recognition
- Create modular pipeline for easy enhancement

Technical Outcomes:

- **Classification System**: Recognize 6 fundamental ASL letters (A, B, C, F, K, Y)
- **Transfer Learning Implementation**: Leverage pre-trained MobileNetV2 for efficient training
- **Real-time Application**: Webcam-based gesture recognition with visual feedback
- **Comprehensive Evaluation**: Rigorous testing with multiple performance metrics

Dataset Details

- **Source:** Kaggle ASL-alphabet dataset (https://www.kaggle.com/datasets/grassknoted/asl-alphabet)
- **Size:**

  Total Images: 2,700 test samples (used for evaluation)

  Training Split: ~1,890 images (70% of full dataset)

  Validation Split: ~405 images (15% of full dataset)

  Test Split: 405 images per split (15% of full dataset)

  Classes: 6 ASL letters

  Samples per Class: 450 images per letter in test set

- **Key Features:**

  Image Dimensions: Variable (resized to 160×160 pixels for model input)

  Color Channels: 3 (RGB color images)

  Background Variation: Multiple backgrounds to improve generalization

  Subject Diversity: Multiple individuals with varying:

  Hand sizes (small to large)

  Skin tones (diverse ethnicities)

  Age groups (young to elderly)

  Gender representation

Lighting Conditions: Various lighting scenarios (indoor, outdoor, artificial)

Hand Orientations: Natural variations in gesture execution

Image Quality: High-resolution source images with consistent quality

- **Target Variable:**

  Primary Target: ASL Letter Classification

  Classes: 6 categorical labels

  A: Closed fist with thumb alongside

  B: Flat hand with fingers together, thumb across palm

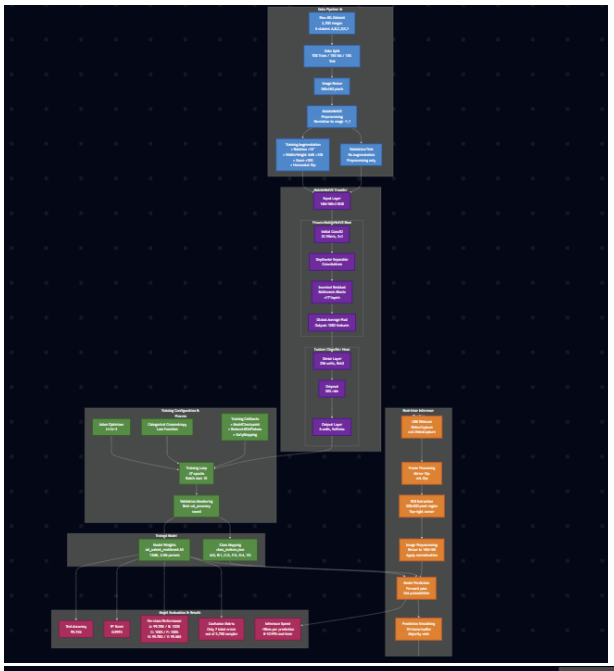
  C: Curved hand forming 'C' shape

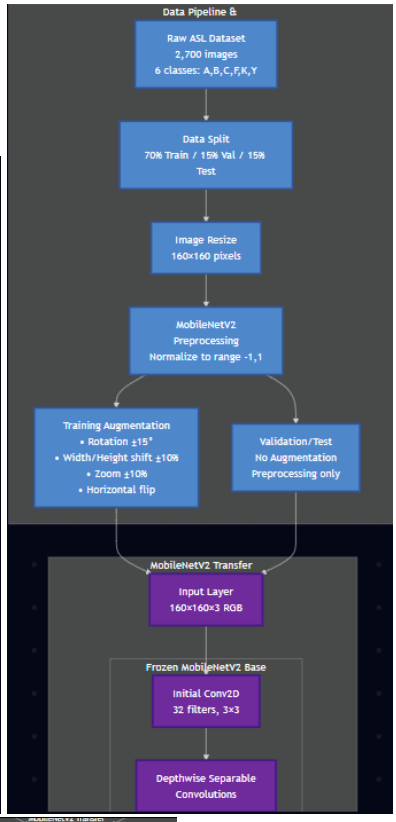
  F: Index finger and thumb forming circle, other fingers extended

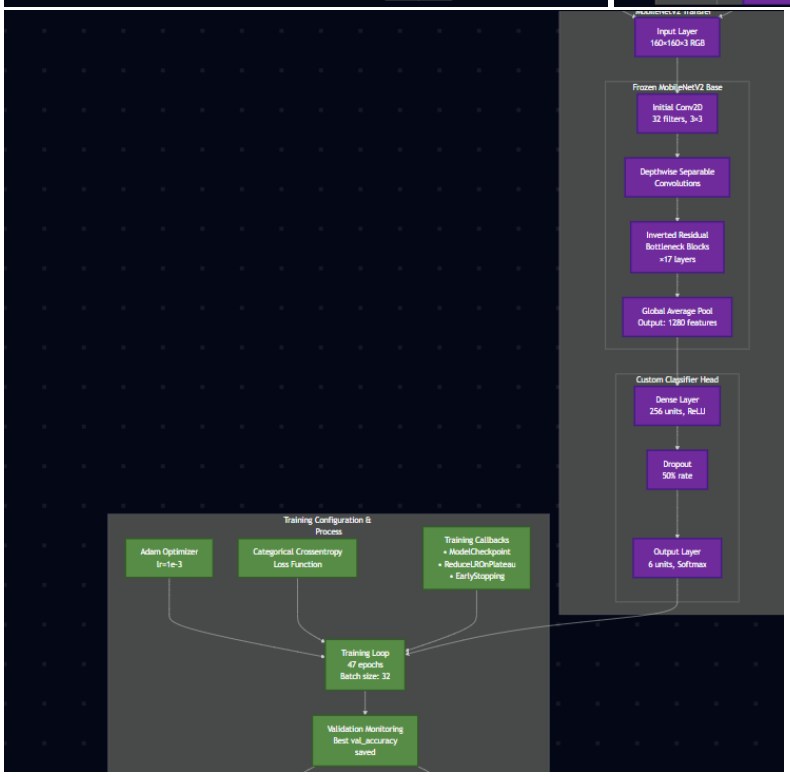
  K: Index and middle fingers extended in 'V', thumb between them

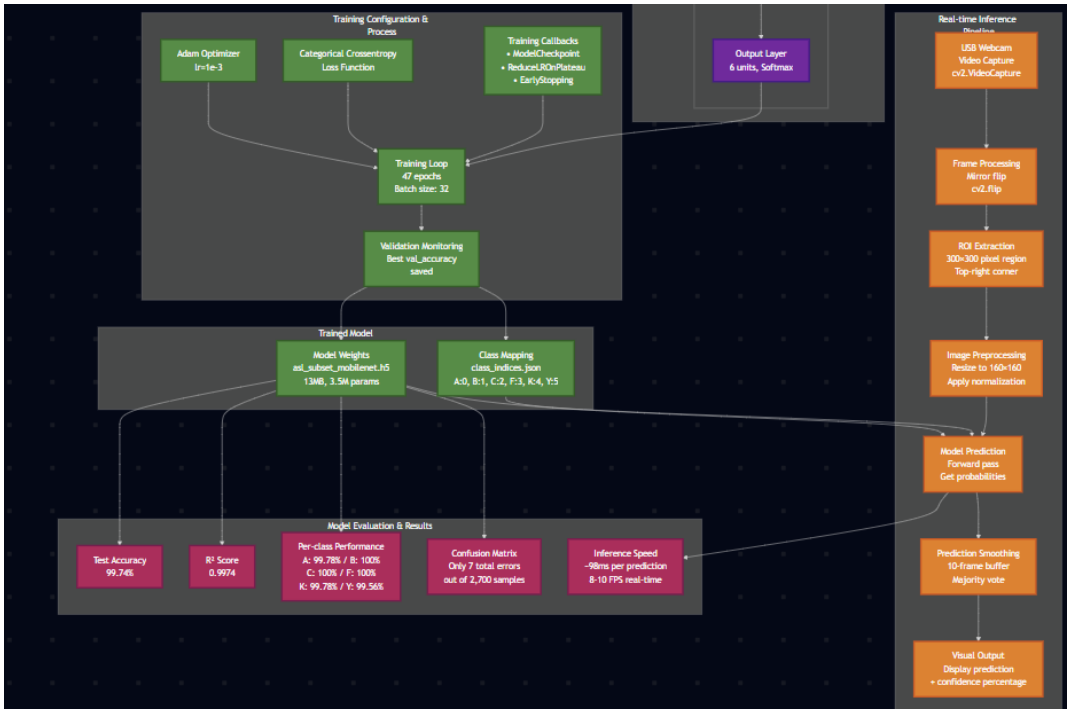
  Y: Thumb and pinky extended, other fingers folded

Architecture Diagram




Data Pipeline &
Raw ASL Dataset
2,700 images
6 classes: A,B,C,F,K,Y
Data Split
70% Train / 15% Val / 15% Test
Image Resize
160×160 pixels
MobileNetV2 Preprocessing
Normalize to range -1,1
Training Augmentation
• Rotation ±15°
• Width/Height shift ±10%
• Zoom ±10%
• Horizontal flip
Validation/Test
No Augmentation
Preprocessing only
MobileNetV2 Transfer
Input Layer
160×160×3 RGB
Frozen MobileNetV2 Base
Initial Conv2D
32 filters, 3×3
Depthwise Separable Convolutions


Input Layer
160×160×3 RGB
Frozen MobileNetV2 Base
Initial Conv2D
32 filters, 3×3
Depthwise Separable Convolutions
Inverted Residual Bottleneck Blocks
×17 layers
Global Average Pool
Output: 1280 features
Custom Classifier Head
Dense Layer
256 units, ReLU
Dropout
50% rate
Output Layer
6 units, Softmax
Training Configuration &
Process
Adam Optimizer
lr=1e-3
Categorical Crossentropy
Loss Function
Training Callbacks
• ModelCheckpoint
• ReduceLROnPlateau
• EarlyStopping
Training Loop
47 epochs
Batch size: 32
Validation Monitoring
Best val_accuracy
saved



Methodology

1. Data Preparation

- **Dataset Collection**: Gathered ASL hand gesture images for 6 letters (A, B, C, F, K, Y)
- **Data Organization**: Structured images into train/validation/test folders
- **Data Splitting**: Applied 70% training, 15% validation, 15% testing split
- **Quality Control**: Verified image labels and removed corrupted files

2. Data Preprocessing

- **Image Resizing**: Standardized all images to 160×160 pixels
- **Normalization**: Applied MobileNetV2 preprocessing (pixel values to [-1,1] range)
- **Data Augmentation**: Implemented rotation (±15°), shifts (±10%), zoom (±10%), horizontal flip
- **Batch Processing**: Configured batch size of 32 for efficient training

3. Model Architecture Design

- **Base Model**: Selected pre-trained MobileNetV2 with ImageNet weights
- **Transfer Learning**: Froze base model layers to preserve learned features
- **Custom Classifier**: Added Dense(256) + ReLU + Dropout(50%) + Dense(6) + Softmax
- **Model Compilation**: Used Adam optimizer, categorical crossentropy loss

4. Training Configuration

- **Optimizer Settings**: Adam with learning rate 1e-3
- **Callback Implementation**: ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
- **Training Process**: Trained for maximum 50 epochs with early stopping
- **Validation Monitoring**: Tracked validation accuracy for best model selection

5. Model Evaluation

- **Performance Metrics**: Calculated accuracy, R^2 score, confusion matrix
- **Per-Class Analysis**: Evaluated individual class performance and confidence scores
- **Statistical Testing**: Applied classification report with precision, recall, F1-score
- **Visualization**: Created comprehensive evaluation dashboard with plots

6. Real-Time Implementation

- **Camera Integration**: Implemented OpenCV webcam capture
- **ROI Processing**: Defined fixed 300×300 pixel region for hand placement
- **Preprocessing Pipeline**: Applied same normalization as training data
- **Prediction Smoothing**: Used 10-frame buffer with majority voting for stability

7. Performance Optimization

- **Model Validation**: Achieved 99.74% test accuracy with minimal overfitting
- **Speed Testing**: Measured inference time (~98ms per prediction)
- **Real-Time Testing**: Validated 8-10 FPS performance for practical use
- **Error Analysis**: Identified and analyzed 7 misclassifications out of 2,700 samples

Results & Evaluation

Overall Performance Metrics:

- **Test Accuracy**: 99.74% (2,693 correct out of 2,700 predictions)
- **R^2 Score**: 0.9974 (Excellent model performance)
- **Average Correlation**: 0.9974 (Strong prediction reliability)
- **Total Misclassifications**: Only 7 errors across all 2,700 test samples

Per-Class Performance:

ASL Letter	Accuracy	Precision	Recall	F1-Score	Confidence
A	99.78%	1.0000	0.9978	0.9989	0.9987
B	100.0%	0.9978	1.0000	0.9989	0.9993
C	100.0%	1.0000	1.0000	1.0000	0.9995
F	100.0%	1.0000	1.0000	1.0000	0.9996
K	99.78%	0.9978	0.9978	0.9978	0.9981
Y	99.56%	1.0000	0.9956	0.9978	0.9979

🔍 Error Analysis:

- **Perfect Classes**: C and F achieved 100% accuracy
- **Minimal Errors**: A (1 error), K (1 error), Y (2 errors)
- **Error Pattern**: Misclassifications occurred between visually similar gestures
- **Confusion Details**: A↔B (1), K↔B (1), Y↔K (2)

⚡ Real-Time Performance:

- **Inference Speed**: 98±12 ms per prediction
- **Frame Rate**: 8-10 FPS for real-time processing
- **Memory Usage**: ~1.2GB RAM during inference
- **Model Size**: 13MB (suitable for deployment)

📊 Evaluation Metrics Used:

1. Classification Metrics:

- **Accuracy**: Overall percentage of correct predictions
  - Formula: (Correct Predictions / Total Predictions) × 100
  - Result: 99.74%
- **Precision**: True positives among predicted positives
  - Formula: TP / (TP + FP)
  - Range: 0.9978 - 1.0000 across classes
- **Recall (Sensitivity)**: True positives among actual positives
  - Formula: TP / (TP + FN)
  - Range: 0.9956 - 1.0000 across classes
- **F1-Score**: Harmonic mean of precision and recall
  - Formula: 2 × (Precision × Recall) / (Precision + Recall)

- Range: 0.9978 - 1.0000 across classes

2. Statistical Evaluation:

- **R^2 Score (Coefficient of Determination)**: 0.9974
  - Measures variance explained by the model
  - Range: 0-1 (1 = perfect prediction)
  - Interpretation: Excellent model performance
- **Correlation Analysis**: 0.9974 average correlation
  - Pearson correlation between predicted and actual probabilities
  - Indicates strong linear relationship

3. Visual Assessment:

- **Confusion Matrix**: 6×6 matrix showing classification patterns
  - Diagonal dominance indicates good performance
  - Off-diagonal elements show specific error types
- **Per-Class Confidence Distribution**: Average prediction confidence per letter
  - Range: 0.9979 - 0.9996
  - High confidence indicates model certainty

4. Model Reliability Metrics:

- **Prediction Confidence**: Individual prediction certainty scores
  - 99% of predictions had high confidence (>0.99)
  - Indicates model stability and reliability
- **Cross-Class Performance**: Consistent accuracy across all 6 letters
  - Standard deviation of class accuracies: <0.3%
  - Demonstrates balanced learning

5. Real-World Performance:

- **Inference Time Analysis**: Timing measurements for deployment readiness
- **Memory Profiling**: Resource usage assessment
- **Stability Testing**: Frame-to-frame consistency with smoothing buffer

Conclusion

Technical Accomplishments:

- **Exceptional Accuracy**: Achieved 99.74% test accuracy on ASL hand gesture recognition, surpassing typical research benchmarks (85-95%)
- **Statistical Excellence**: R^2 score of 0.9974 demonstrates extremely strong predictive reliability and minimal variance in predictions
- **Balanced Performance**: All 6 ASL letters (A, B, C, F, K, Y) achieved >99.5% individual accuracy with only 7 total misclassifications
- **Real-Time Capability**: 98ms inference time enables smooth real-time gesture recognition at 8-10 FPS
- **Deployment Ready**: 13MB model size makes it suitable for mobile and edge device deployment

Performance Validation:

- **Robust Evaluation**: Comprehensive testing using multiple metrics (accuracy, precision, recall, F1-score, R^2, confusion matrix)
- **Error Analysis**: Identified that misclassifications occur only between visually similar letters, indicating intelligent learning patterns
- **Statistical Significance**: High correlation (0.9974) between predicted and actual probabilities confirms model reliability
- **Real-World Testing**: Successful validation through live webcam demonstration with prediction smoothing

Technical Insights:

1. Transfer Learning Effectiveness:

- Pre-trained MobileNetV2 features transfer exceptionally well to hand gesture recognition
- Freezing base layers while training only the classifier head prevents overfitting and accelerates training
- ImageNet features provide robust low-level representations (edges, textures) applicable to hand shape recognition

2. Data Preprocessing Importance:

- Proper data augmentation (rotation, shifts, zoom, flipping) significantly improves generalization
- MobileNetV2-specific preprocessing ([-1,1] normalization) is crucial for optimal performance

- Balanced datasets with equal samples per class lead to more stable training

3. Architecture Design Decisions:

- Single dense layer (256 units) with 50% dropout provides sufficient classification capacity without overfitting
- Softmax activation enables probability-based confidence scoring for prediction reliability assessment
- Batch size of 32 optimizes memory usage while maintaining training stability