Hybrid Neural Network Model for Face Recognition

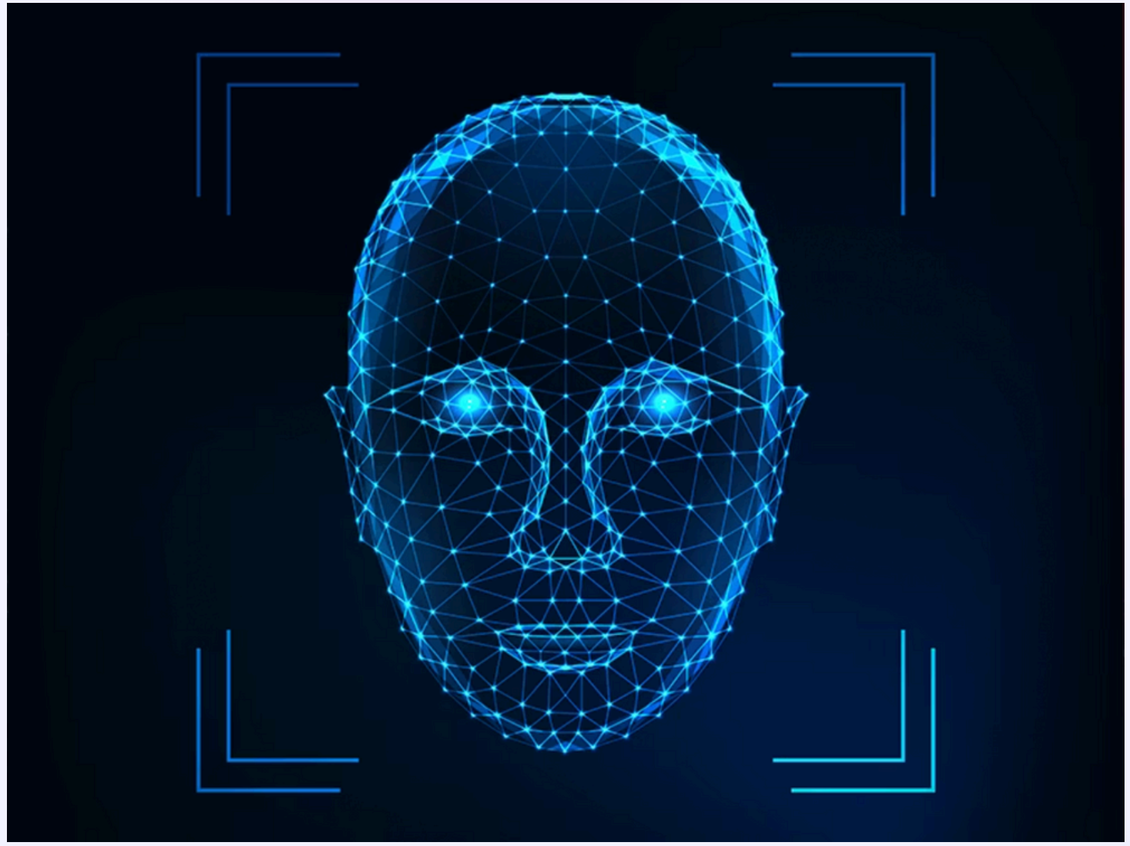
Deep Learning-Based Face Identification using CNN and ANN

Presented By: **Nandhakumar Chitra Ragupathy**

Understanding Face Recognition Technology

Facial recognition is a powerful biometric technique that identifies individuals based on unique facial features and geometric patterns.

This project leverages a **hybrid deep learning approach** that combines the strengths of two neural architectures: CNNs extract complex visual features while ANNs perform sophisticated classification.



The system performs both **static image analysis** and **video processing**, demonstrating practical applications in security, authentication, and human-computer interaction.

Project Objectives

Multi-Modal Detection

Detect and recognize faces from both static images and live video streams with consistent accuracy

Deep Feature Extraction

Extract hierarchical facial features using convolutional layers that capture spatial patterns

Identity Classification

Classify and match faces to known identities using artificial neural network layers

Performance Evaluation

Assess model effectiveness using accuracy, precision, recall, and F1-score metrics

Dataset & Preprocessing Pipeline

Dataset Organization

Images organized in person-specific directories for efficient training

Face Detection

OpenCV identifies and isolates facial regions from raw images

Image Processing

Cropping, resizing to uniform dimensions, and pixel normalization

Data Splitting

Separation into training and testing sets for robust evaluation

System Workflow Architecture

1

Input Acquisition

Images or video frames captured from various sources

2

Preprocessing

Face detection, normalization, and quality enhancement

3

CNN Feature Extraction

Convolutional layers identify spatial patterns and facial features

4

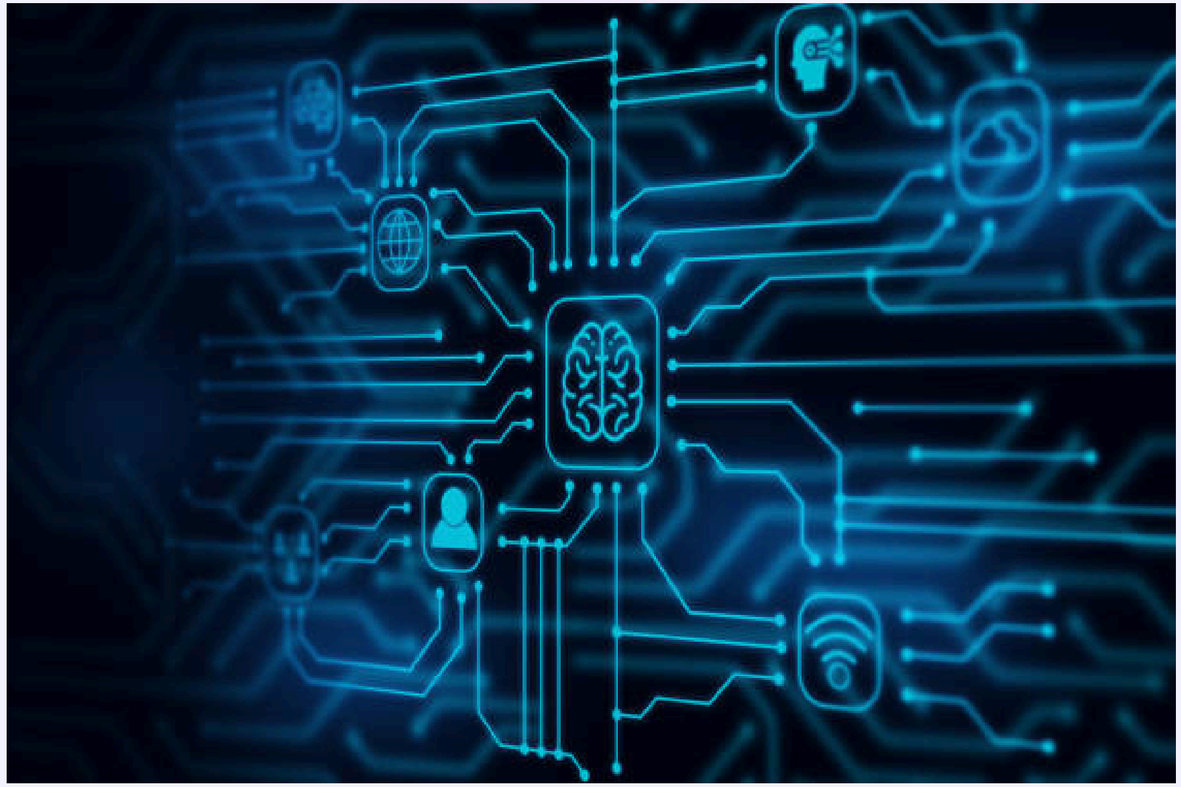
ANN Classification

Dense layers map features to identity classes

5

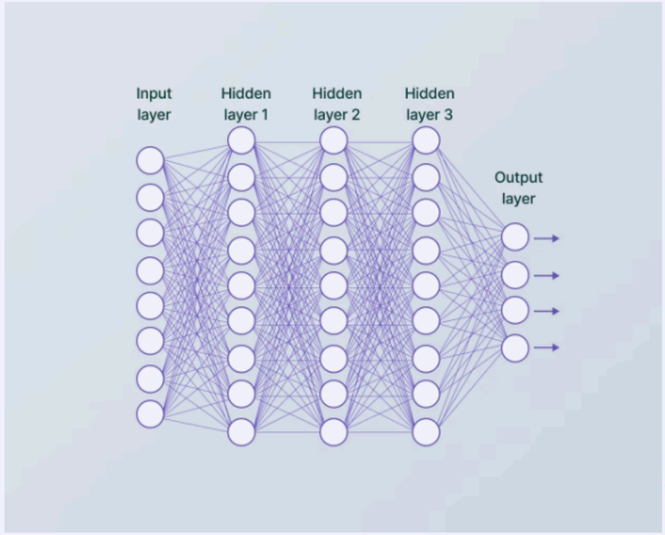
Recognition Output

Identified person with confidence score



CNN Architecture Design

Convolutional Neural Network Layers



01

Convolutional Layers

Multiple filters extract spatial facial features like edges, textures, and complex patterns at different scales

02

Pooling Layers

Reduce spatial dimensions while preserving important features, improving computational efficiency

03

Flatten Layer

Convert 2D feature maps into 1D feature vector for dense layer processing

04

Feature Output

High-dimensional representation passed to ANN for classification

ANN Classification Network

Dense Layers

Fully connected layers learn complex relationships between extracted features and identity classes

Activation Functions

ReLU introduces non-linearity in hidden layers; **Softmax** produces probability distribution over classes

Loss Function

Categorical Crossentropy measures prediction error for multi-class classification tasks

Optimizer

Adam optimizer adapts learning rates for efficient training with momentum and adaptive gradients

Implementation Stack

Core Technologies

Python

Primary programming language for deep learning implementation

TensorFlow & Keras

Deep learning frameworks for building and training neural networks

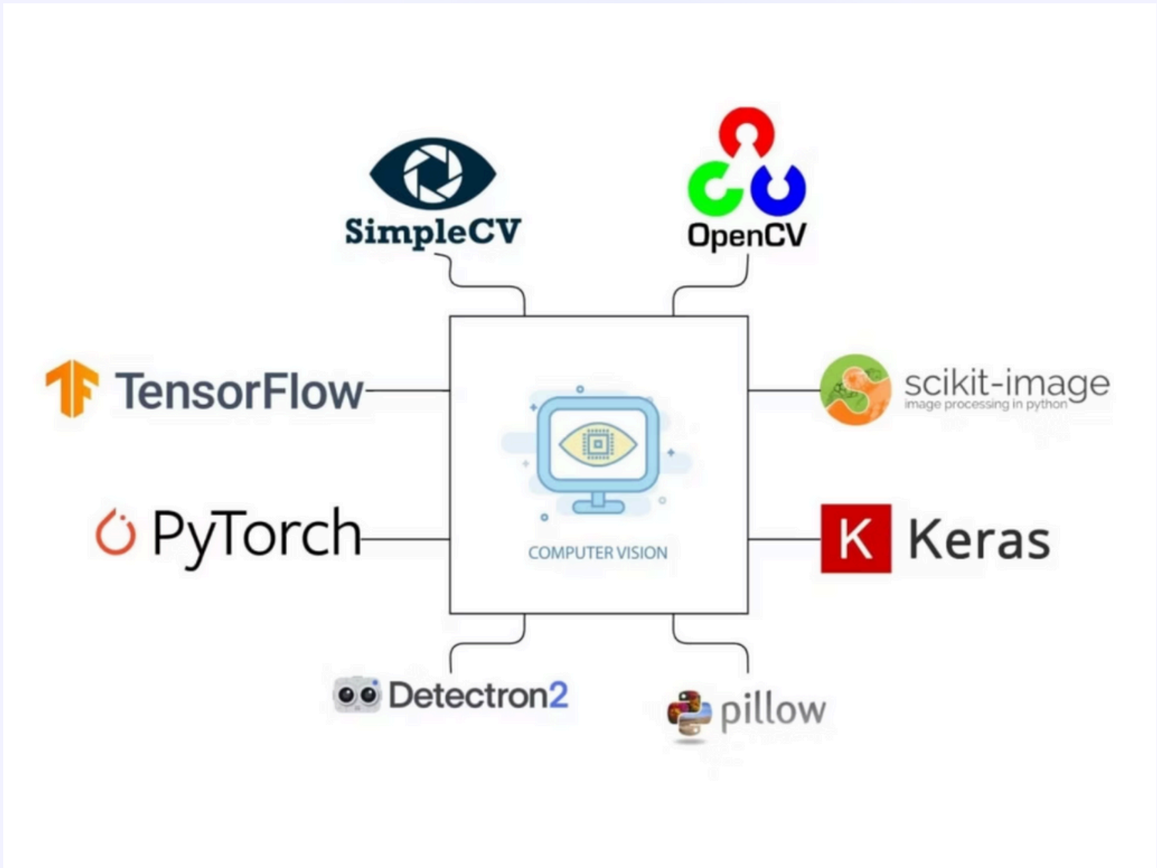
OpenCV

Computer vision library for image processing and face detection

Supporting Libraries

- **NumPy:** Numerical computing and array operations
- **Matplotlib:** Data visualization and plotting
- **scikit-learn:** Machine learning utilities and metrics

Development Environment: Jupyter Notebook for interactive development and experimentation



Model Performance Metrics

92%

Overall Accuracy

Correctly classified faces across test dataset

90.1%

Precision

Positive predictions that were actually correct

91.4%

Recall

Actual positives correctly identified

90.7%

F1-Score

Harmonic mean of precision and recall

The model demonstrates **strong performance** across all evaluation metrics, indicating balanced precision and recall.

System performs efficiently in both **static image recognition** and **real-time video detection** using OpenCV's video processing capabilities.

Future Directions & Conclusion

Enhancement Opportunities

- **Transfer Learning**

  Implement pre-trained models like VGGFace, ResNet, or MobileNet for improved feature extraction

- **Face Embeddings**

  Integrate advanced embedding techniques using FaceNet or ArcFace for better representation

- **Multi-Task Learning**

  Add emotion recognition and facial expression analysis capabilities

- **Deployment**

  Create web interfaces using Flask or Streamlit for broader accessibility

Key Takeaways

This **hybrid CNN-ANN architecture** successfully demonstrates the complete deep learning pipeline from preprocessing to real-time recognition.

Achieving ~92% accuracy, the model performs effectively in both controlled and dynamic scenarios, showcasing practical viability for security and authentication applications.