THESIS DEFENSE

POCKET LENS

GRADUATION PROJECT

Prepared by:
Ahmed Mohamed Ismail
Moaz Mohamed Elsherbini
Mostafa Ashraf Kamal
Nader Youhanna Khalil

Supervised by:
Assoc. Prof. Dr. Mona Farouk

CAIRO UNIVERSITY

ACKNOWLEDGEMENT

THESIS DEFENSE

OUTLINE

INTRODUCTION

According to the WHO, around 2 billion people are visually impaired or blind. This is not a minority. Nevertheless, very little has been done to help them throughout their day. The proposed system offers a mobile application that uses AI to help VIB people complete their daily tasks. It captures images from the user's camera as input and gives the user feedback through a text-to-speech module.

POCKET LENS

LITTERATURE REVIEW

VIB users are often put at a disadvantage regarding their visually able peers. Technological advancements have always been concerned with providing better and easier-to-use solutions.

These efforts have been largely directed toward the use of sensors, which in many are not available to every user.

Moreover, many of the applications that can be found in the market are not particularly easy to use. They often require some degree of tactile interaction, which VIB users will most probably not be able to provide.

Some of these applications are designed to be used by sighted people alongside VIB users, which can come as impractical.

POCKET LENS

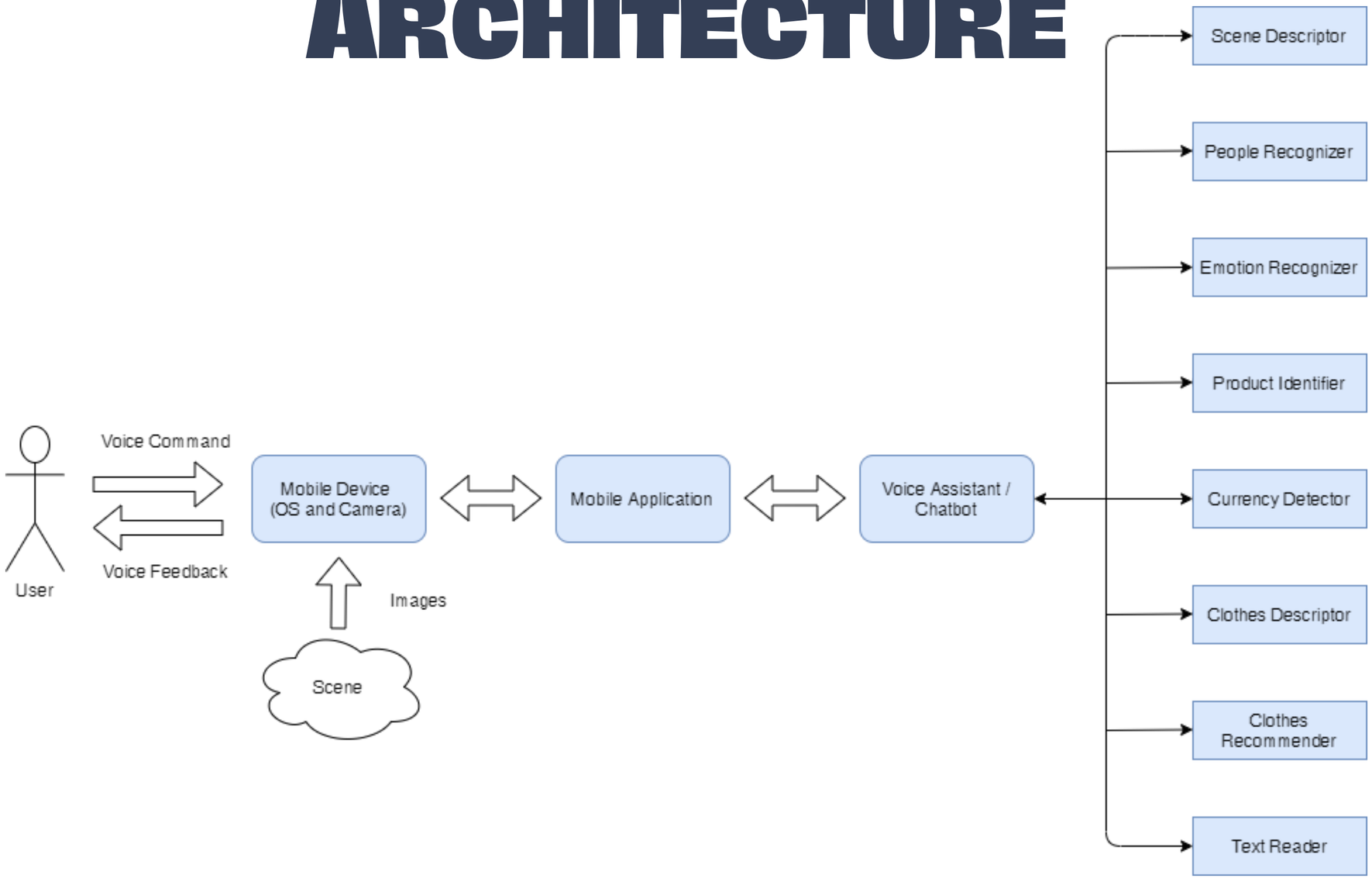
PROBLEM STATEMENT

FILLING THE GAP

- The essential question is how to use AI and Machine Learning techniques to create a mobile application that can serve as an assistant to visually impaired and blind (VIB) people.

- The proposed system aims to address previous problems by rendering the contact between the application and the VIB user purely vocal, allowing for easier communication and interaction.

SYSTEM DESIGN AND ARCHITECTURE



MODULAR DECOMPOSITION

MOBILE APPLICATION

- Framework: Flutter
- Programming Language: Dart
- High performance, cross platform applications

MODULAR
DECOMPOSITION

MOBILE APPLICATION

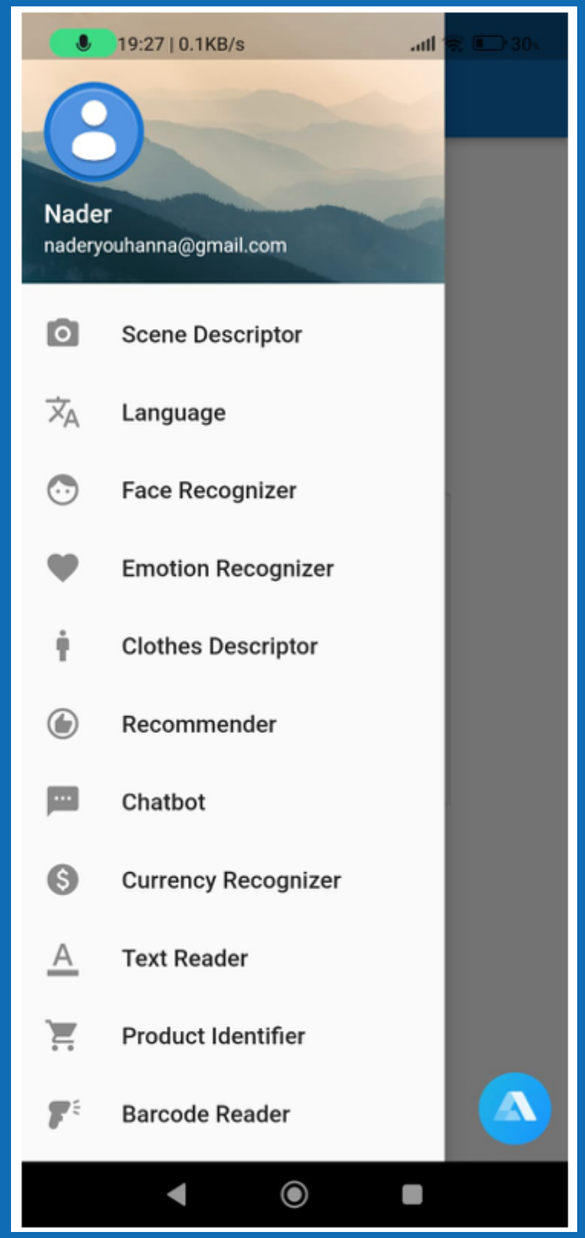
HOME
PAGE

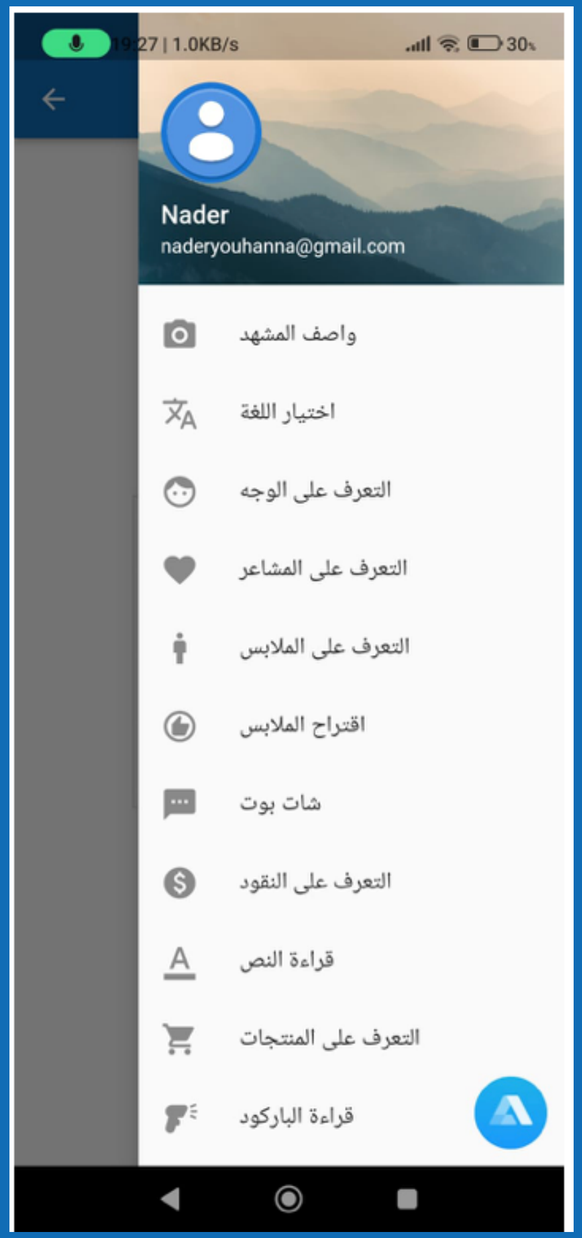
MODULAR DECOMPOSITION

MOBILE APPLICATION

NAVIGATION

Using Side Bar Menu
OR
Using voice commands

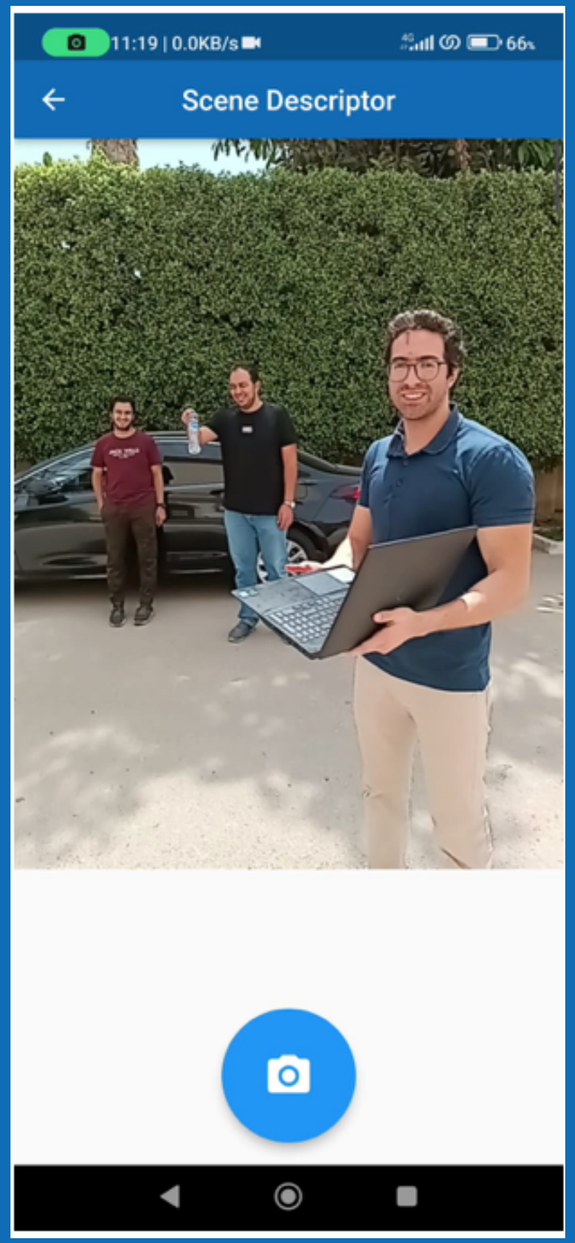




MODULAR
DECOMPOSITION

MOBILE APPLICATION

APPLICATION MODULES

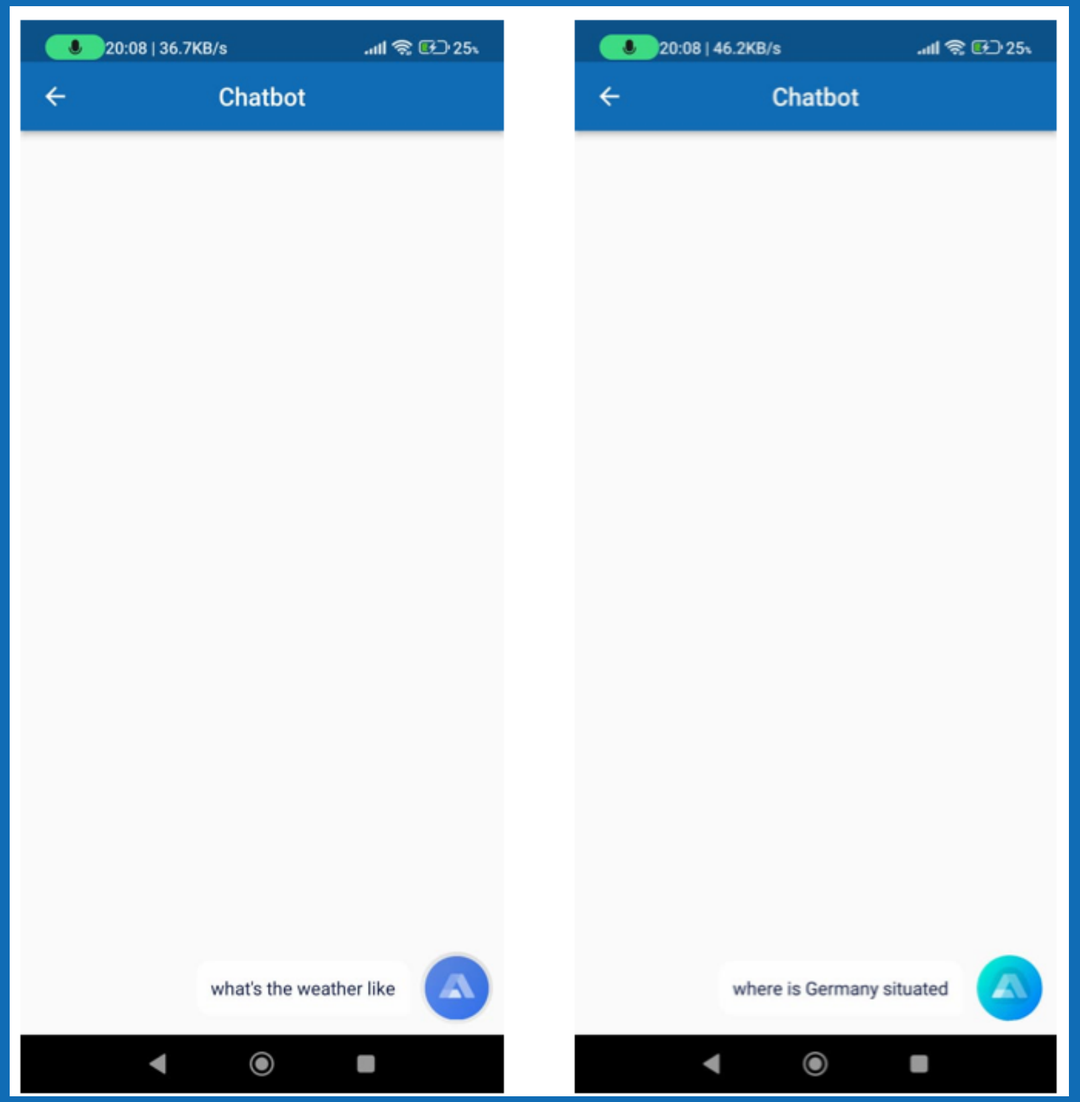


MODULAR
DECOMPOSITION

MOBILE APPLICATION

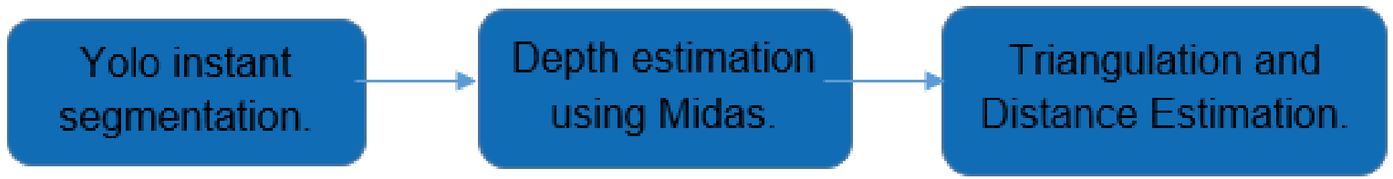
CHATBOT

Q: What’s the weather like ?
A: (sound) It’s currently sunny and warm
Q: Where is Germany located ?
A: (sound) Germany is situated in central Europe



SCENE DESCRIPTOR

- COCO Dataset
- Object Detection and Instant Segmentation using Yolov8
- Depth Estimation using Midas
- Triangularization and Distance Estimation



SCENE DESCRIPTOR

MODULAR DECOMPOSITION

COCO DATASET

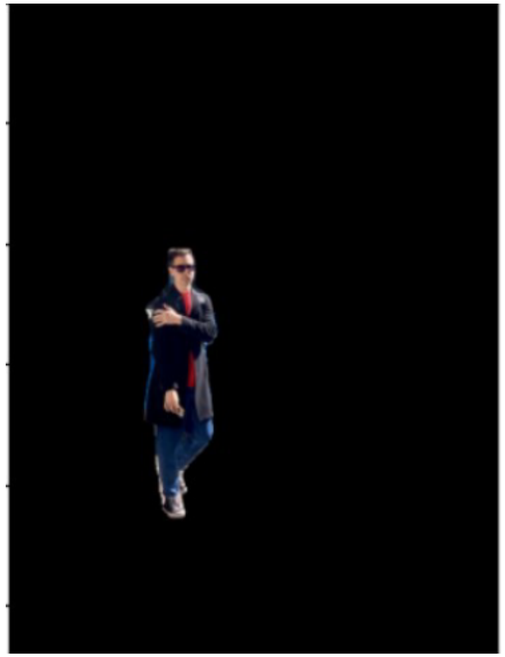
- Dataset Description
- Training With COCO dataset
- Advantages and limitations

MODULAR DECOMPOSITION

SCENE DESCRIPTOR

- Architecture and Worflow
- Bounding boxes prediction
- Class Probability Prediction

- Instant Segmentation
- Non-Maxima Suppression
- Training YOLOv8

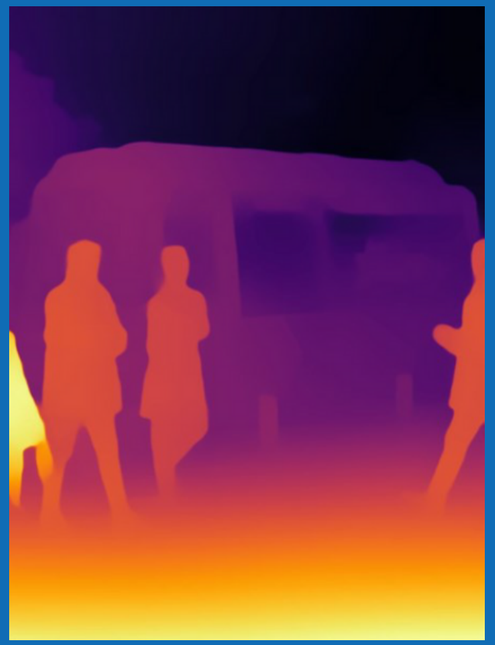


MODULAR
DECOMPOSITION

SCENE DESCRIPTOR

- Architecture and Worflow
- Bounding boxes prediction
- Class Probability Prediction

- Instant Segmentation
- Non-Maxima Suppression
- Training YOLOv8

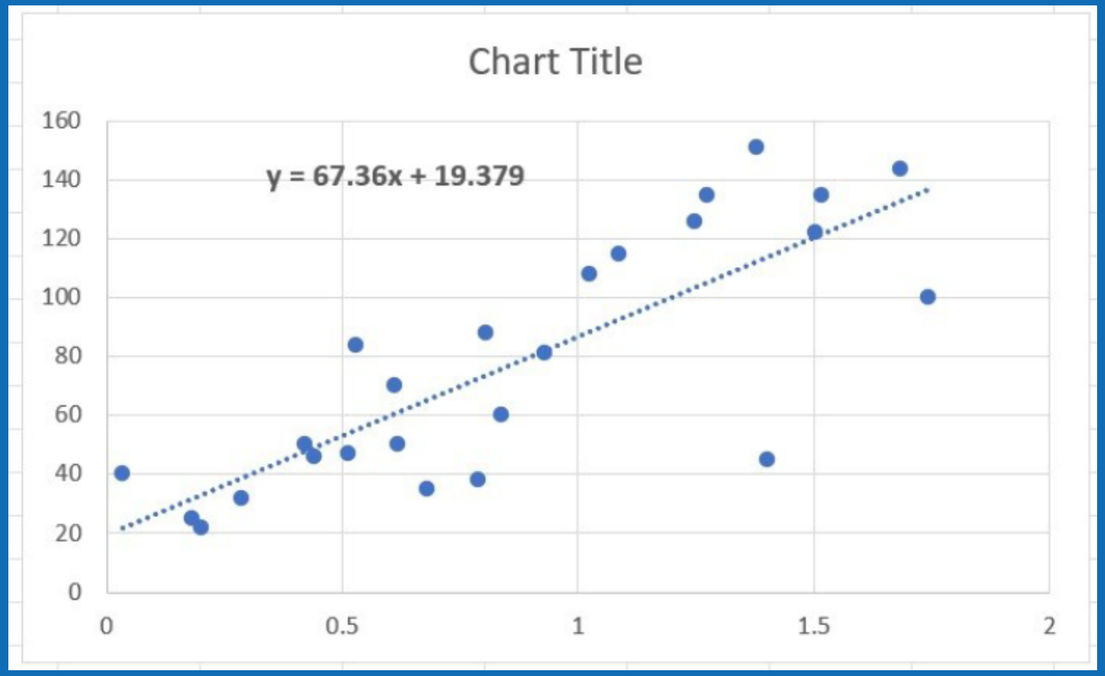


MODULAR DECOMPOSITION

SCENE DESCRIPTOR

- Heat map conversion to grayscale
- Triangularization parameters
- distance to object = (object width in real life * focal length) / (object pixel width in image * depth value)
- Y = a * X + b, where Y represents the distance to the object (in centimeters), X represents the calculated value (object width in real life) / (object pixel width in image * depth value), and "a" and "b" are constants.
- Obtaining values of X
- Plotting distance estimation functions

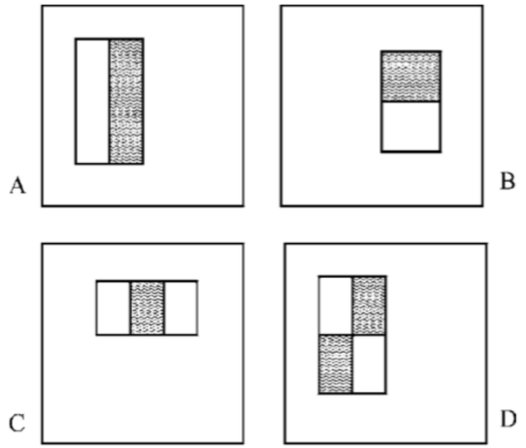
Name	Real Distance	Width	Height	Object Real (Avg) Width	Depth Value	X	Prediction
Apple	38	166	202	10	0.062745098	0.788985149	74.2
Chair	100	851	1126	100	0.101960784	1.742041262	123.8
Chair	135	466	731	100	0.090196078	1.516683519	114.2
Chair	151	598	841	100	0.08627451	1.378229381	107.6
Chair	170	406	696	100	0.050980392	2.818302387	151.6
Chair	225	378	506	100	0.058823529	3.359683794	154.3
Chair	50	999	1526	100	0.105882353	0.618901995	62.9
Chair	84	1188	1156	100	0.121568627	0.5286508	56.6
Chair	126	554	888	100	0.090196078	1.24853114	101
Chair	144	461	757	100	0.078431373	1.684280053	121.5
Cup	100	109	103	11	0.047058824	2.144495413	137.7
Cup	150	85	77	11	0.031372549	4.125	145.2
Cup	25	425	395	11	0.141176471	0.183333333	30.5
Cup	50	215	199	11	0.121568627	0.420855214	48.8
Fork	45	77	273	30	0.078431373	1.401098901	108.7
Laptop	40	1198	1064	40	1	0.033388982	18.3
Laptop	46	823	646	40	0.109803922	0.442631488	50.4
Laptop	88	576	526	40	0.08627451	0.804924242	75.2
Laptop	122	323	306	40	0.082352941	1.503759398	113.6
Mouse	22	373	283	10	0.133333333	0.201072386	32
Mouse	32	216	297	10	0.117647059	0.286195286	38.6
Mouse	47	284	114	10	0.078431373	0.514112903	55.6
Refrigerator	81	978	1568	200	0.137254902	0.929300292	83
Refrigerator	108	816	1466	200	0.133333333	1.02319236	88.6
Refrigerator	135	718	1381	200	0.11372549	1.273440036	102.3
Spoon	35	127	511	30	0.08627451	0.6804839	67.1
Suitcase	60	445	563	50	0.105882353	0.838760608	77.4
Suitcase	70	469	719	50	0.11372549	0.611481464	62.4
Suitcase	115	469	373	50	0.098039216	1.087420043	92.3
Suitcase	130	277	309	50	0.066666667	2.427184466	144.9
Suitcase	146	338	336	50	0.066666667	2.218934911	139.8
Vase	72	191	298	80	0.094117647	2.852348993	152
Vase	105	136	226	80	0.070588235	5.014749263	115.6



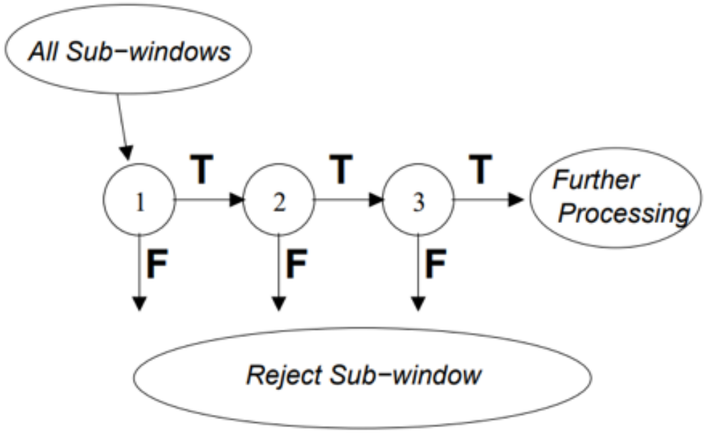
MODULAR DECOMPOSITION

FACE DETECTION

- It uses Haar cascades to extract features from an image and train a machine learning model to recognize a face.
- Haar features are extracted on 24x24 rectangles.



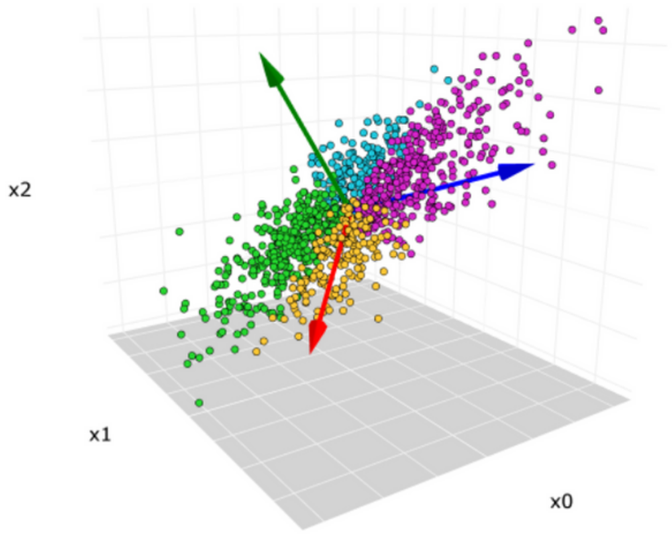
- Adaboost was used to select a number of Haar features.
- The selected Haar features forms a strong classifier
- It uses a Cascade classifier for rapid detection.

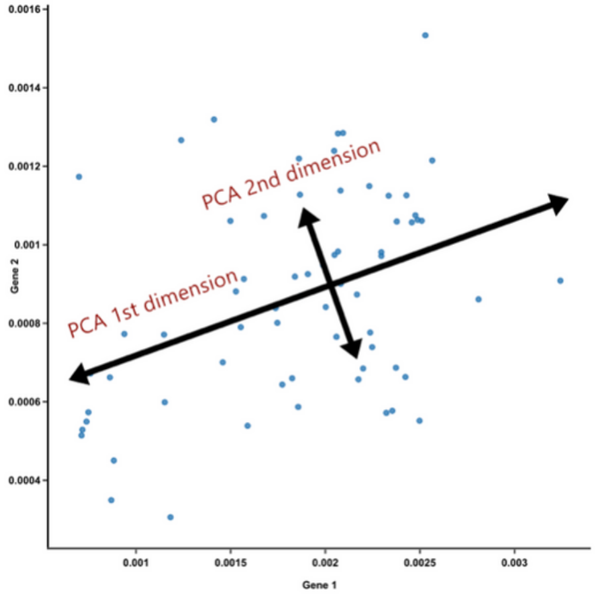


MODULAR DECOMPOSITION

EIGENFACES

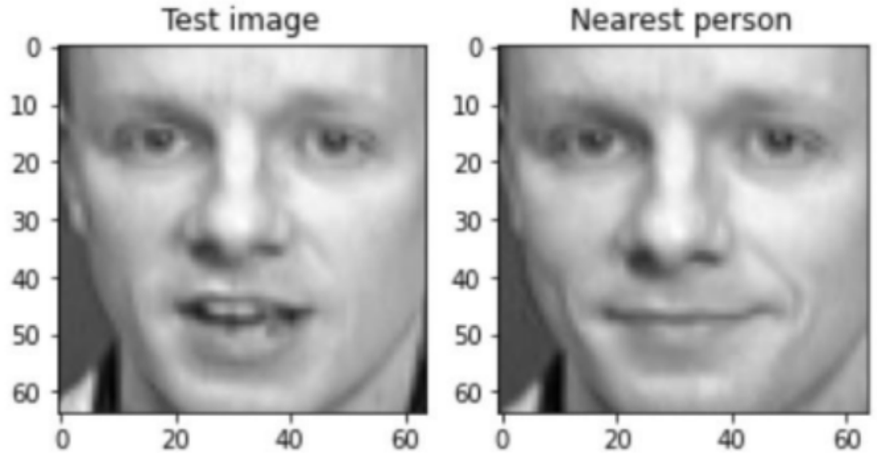
- Uses Principal Component Analysis
- Training examples are points in the feature space
- Select features with greatest variability largest variance
- They help distinguish classes the most





MODULAR
DECOMPOSITION

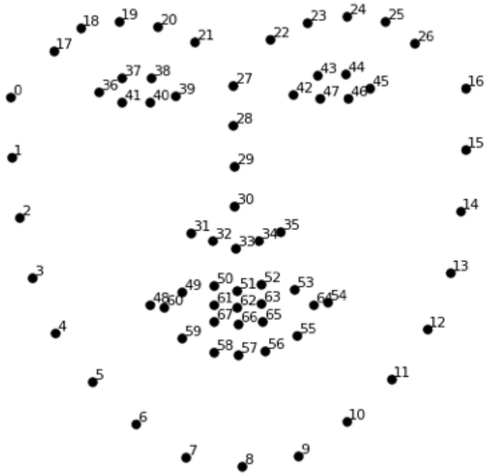
FACE RECOGNITION: EIGENFACES



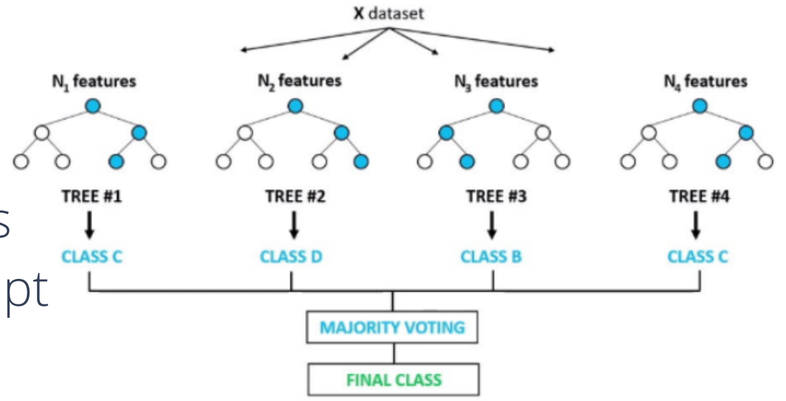
MODULAR DECOMPOSITION

EMOTION DETECTION

- Extraction of facial landmarks
  -The process extracts 68 facial landmarks on the face
- The facial landmarks can be analysis to detect shape of the face, the position and orientation of the eyes, the position of the nose, and the shape of the mouth.



- The extracted facial landmark are used to predict the emotion
- The prediction is done using Random Forest Classifier
- Emotions that can be detected are
  - Fear
  - Happy
  - Neutral
  - Anger
  - Sadness
  - Contempt
  - Disgust
  - Surprise

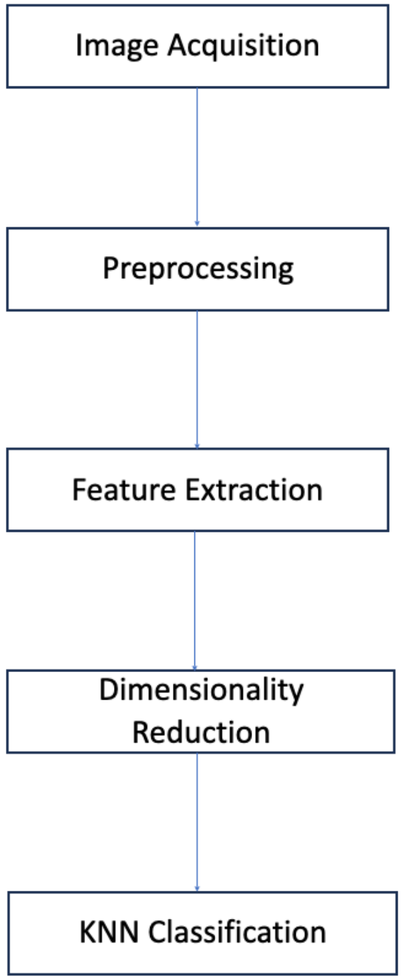


CURRENCY RECOGNIZER

- Currency recognition model using KNN, histogram, texture, and ORB features
- Improves quality of life for visually impaired individuals
- Key benefits:
  - Enhanced autonomy: Enables independent financial management
  - Increased confidence: Boosts self-assurance in daily tasks and social interactions
  - Reduced risk of fraud: Accurate identification of currency denominations
- Overall impact: Contributes to better quality of life and inclusive society

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER



STEP BY STEP PROCESS

- Image Acquisition: Capture and collect currency images
- Preprocessing: Enhance input images (resizing, denoising, normalization)
- Feature Extraction: Extract histogram, texture, and ORB features
- Dimensionality Reduction: Transform original feature space into lower-dimensional space using PCA
- KNN Classification: Apply k-Nearest Neighbors algorithm for recognition

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

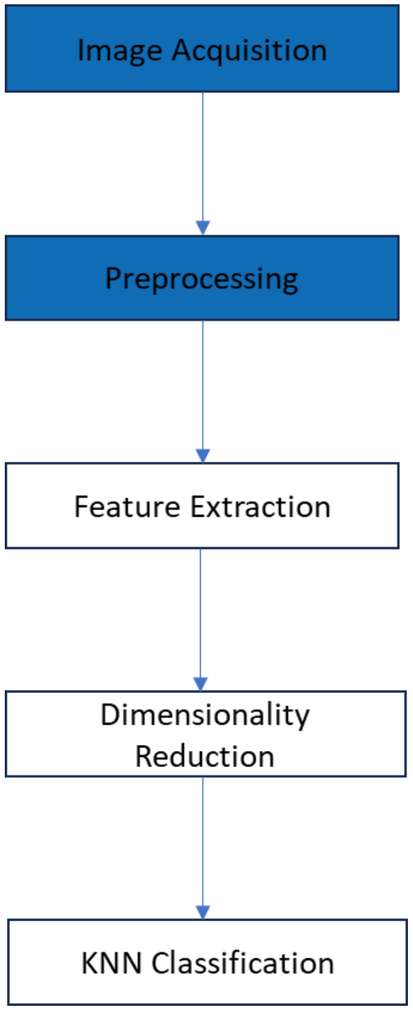


IMAGE ACQUISITION AND DATA PREPROCESSING

- Currency images are captured using our application and transmitted for preprocessing and detection in this model
- Preparing the input dataset and ensuring images are suitable for feature extraction
- Objective: Augment dataset and enhance model robustness for better recognition performance

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

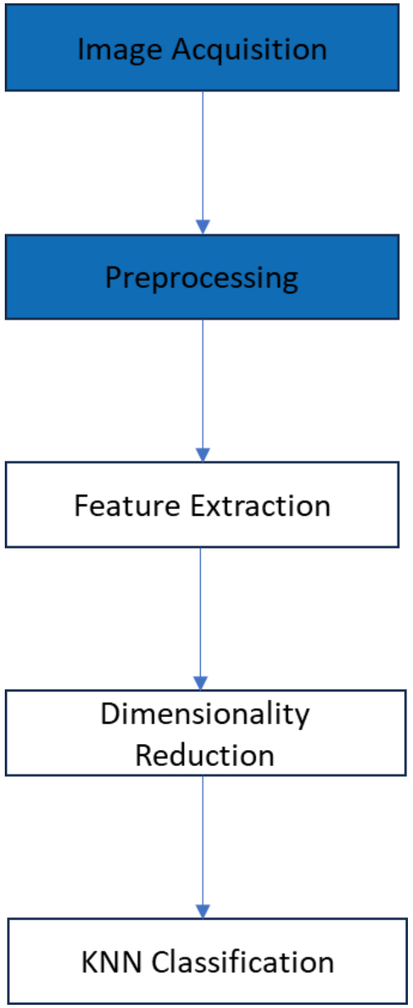
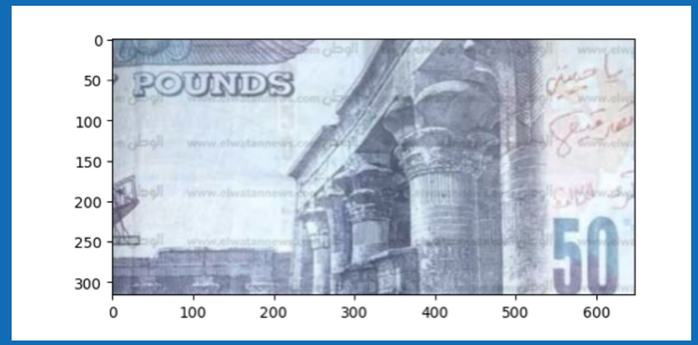


IMAGE ACQUISITION AND DATA PREPROCESSING

- Rotation and Scaling
  - Functions used: rotate(img, angle) and scale(img, scale)
  - Purpose: Introduce variability in image orientation and size
  - Rotation: Apply different angles to input images to account for real-world variations
  - Scaling: Adjust image dimensions to normalize image sizes and improve computational efficiency



MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

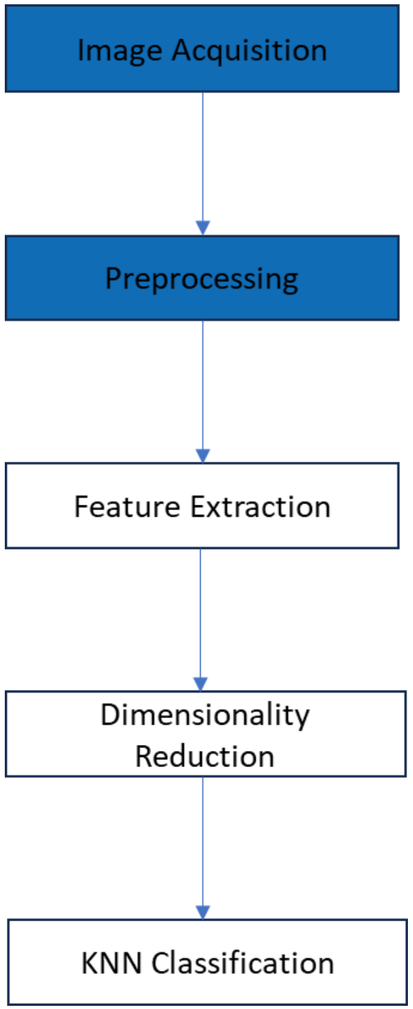
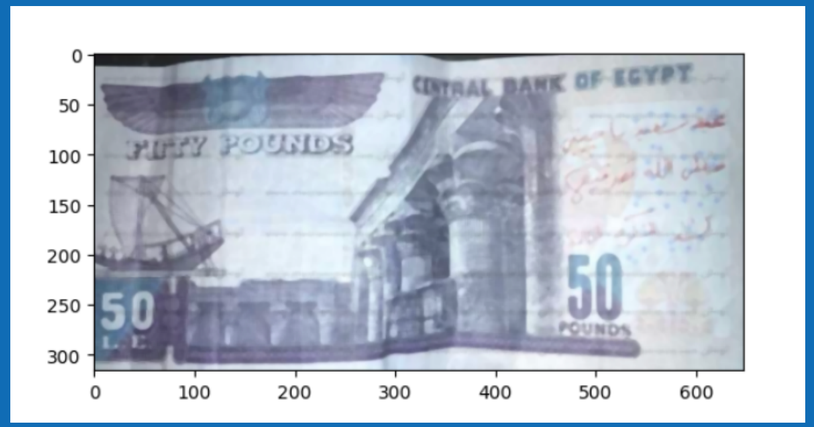


IMAGE ACQUISITION AND DATA PREPROCESSING

- Blurring
  - Function used: blur(img, blur_type)
  - Purpose: Simulate the effect of out-of-focus images and account for different imaging conditions
  - Blurring techniques: Average, Gaussian, Median, and Bilateral
  - Helps improve classifier performance by handling varying degrees of image sharpness



MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

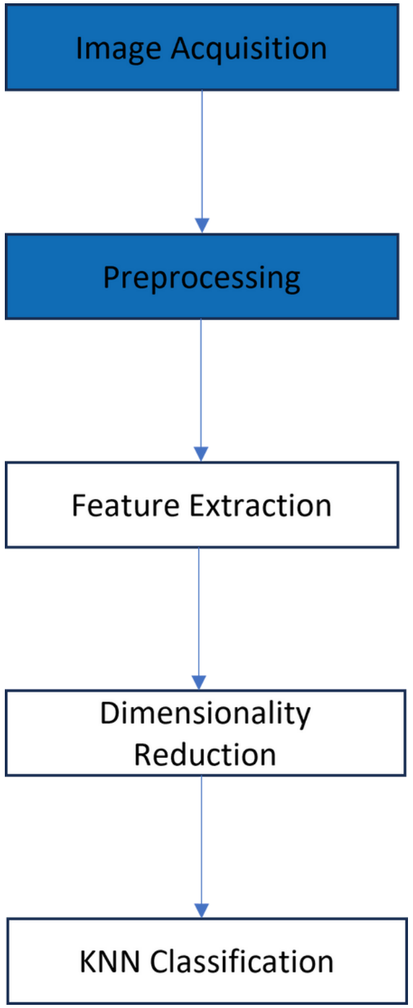
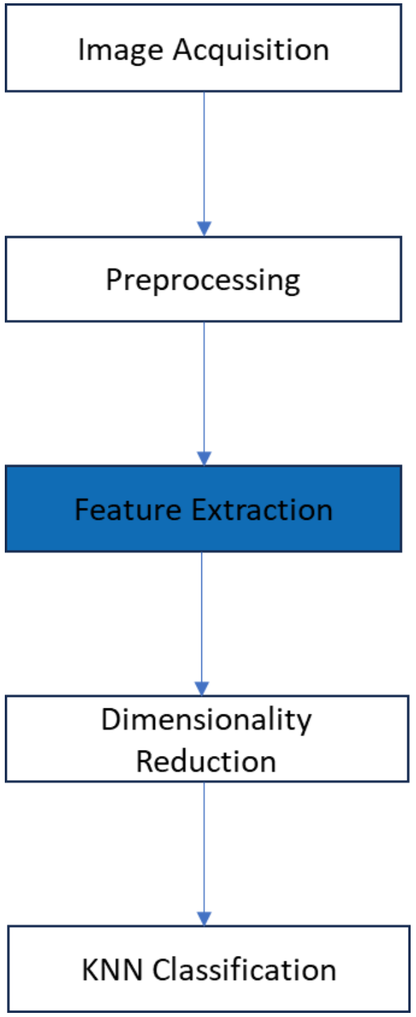


IMAGE ACQUISITION AND DATA PREPROCESSING

- Lighting Adjustment
  - Function used: lighting(img, lighting_type)
  - Purpose: Adapt to different lighting conditions and address illumination variations in input images
  - Lighting adjustments: Brightness, Contrast, Gamma Correction, Histogram Equalization
  - Increases model robustness by handling diverse lighting scenarios encountered in real-world applications

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

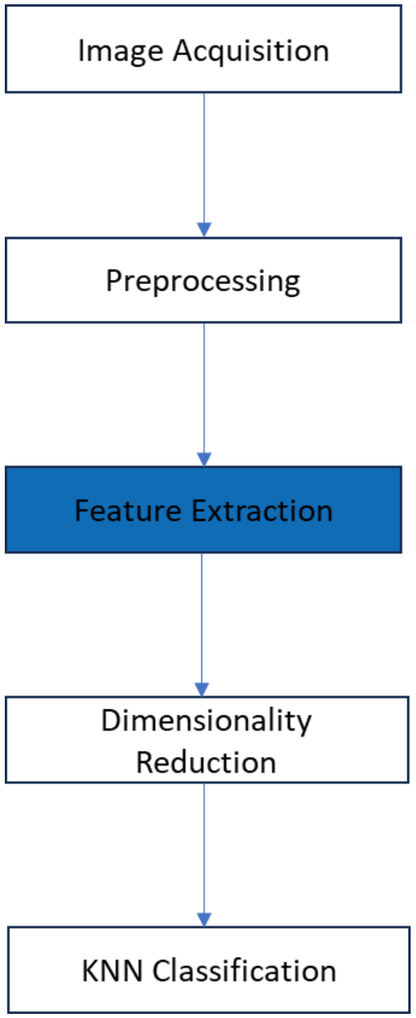


FEATURE EXTRACTION

- Third stage of the currency recognition model for visually impaired people
- Importance of generating a comprehensive feature vector to represent currency images
- Combine histogram, texture, and ORB features to create a comprehensive feature vector
- Utilize feature vector as input for the k-Nearest Neighbors (kNN) algorithm during classification
- Objective: Accurately classify different currency denominations using extracted features

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

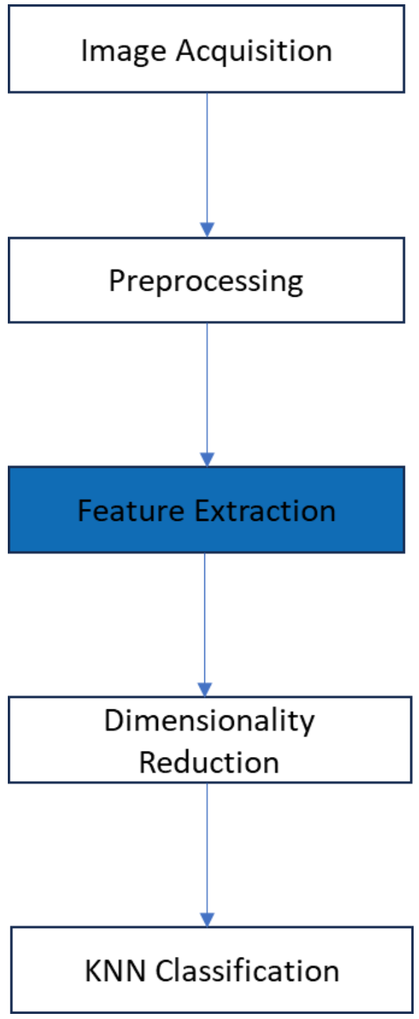


FEATURE EXTRACTION

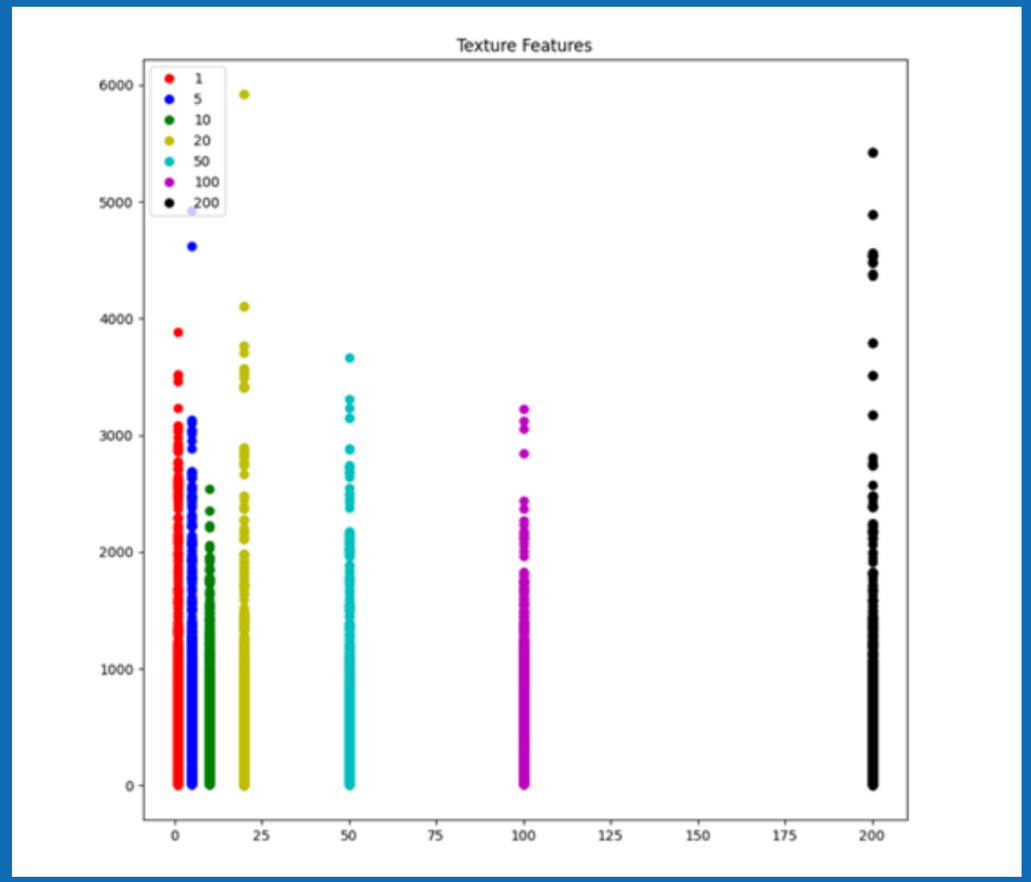
- Histogram Features
  - Extract color information using histograms
  - Graphical representation of color value distribution in an image
  - Computed for each color channel (e.g., Red, Green, and Blue)
  - Concatenate histograms to form a single feature vector
- ORB Features
  - Oriented FAST and Rotated BRIEF (ORB) algorithm for keypoint and descriptor extraction
  - Invariant to rotation, scale, and illumination changes
  - Identify distinctive points (keypoints) in images and compute binary descriptors
  - Capture local patterns around keypoints for classification

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER



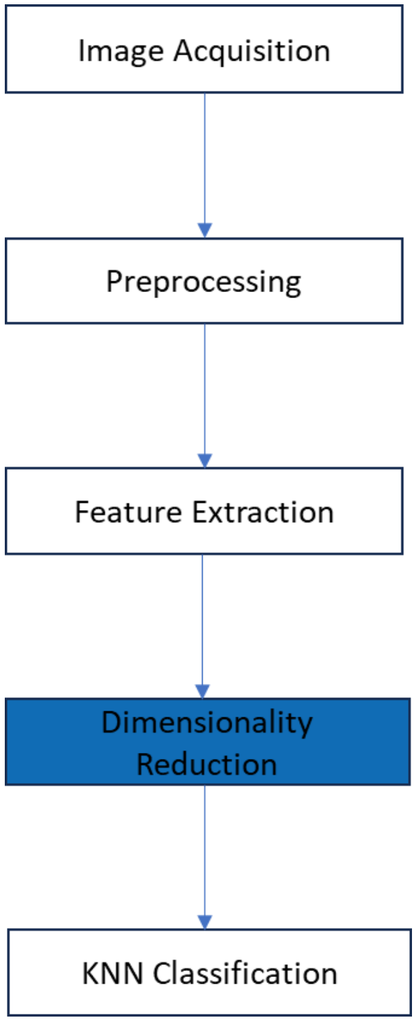
FEATURE EXTRACTION



GLCM of each currency which shows difference in texture for each cuurency

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER

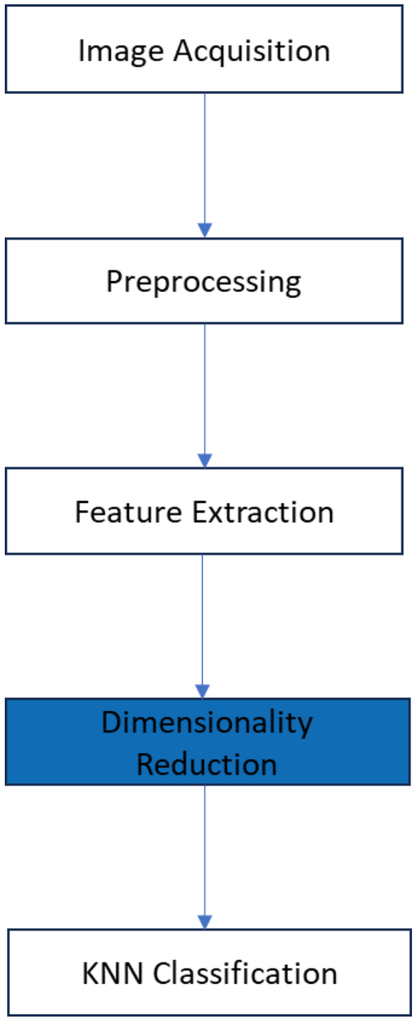


DIMENSIONALITY REDUCTION

- Dimensionality reduction using Principal Component Analysis (PCA)
- Address issues of redundant information and computational complexity
- Key steps:
  1. Standardize feature data
  2. Determine optimal number of principal components
  3. Apply PCA with optimal number of components
  4. Update feature vector
- Lower-dimensional representation used as input for kNN algorithm during classification stage

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER



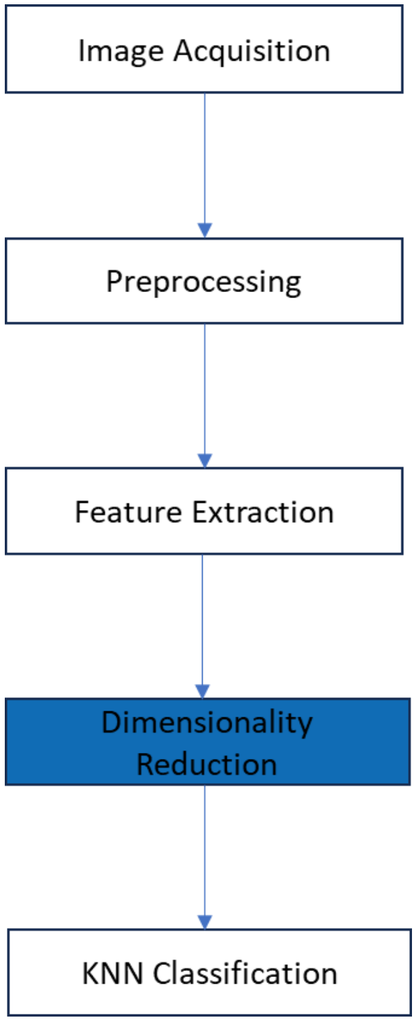
KNN CLASSIFICATION

- Final stage: Classification using k-Nearest Neighbors (kNN) algorithm
- Key steps:
  1. Training kNN classifier
  2. Choosing value of 'k'
  3. Distance metric

- Classification: Calculate distance between input image's feature vector and training dataset, assign majority class among 'k' nearest neighbors
- Outcome: Accurate recognition of different currency denominations for visually impaired users

MODULAR DECOMPOSITION

CURRENCY RECOGNIZER



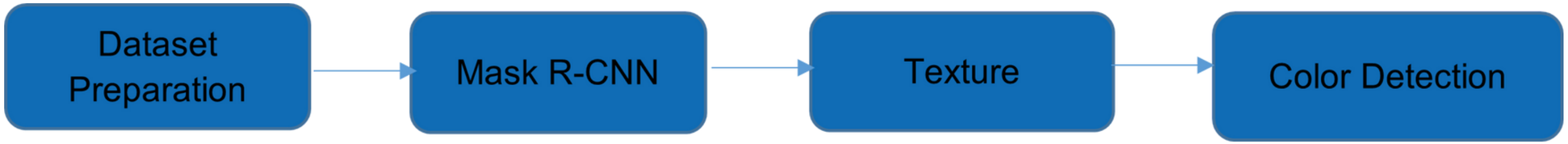
KNN CLASSIFICATION

- Final stage: Classification using k-Nearest Neighbors (kNN) algorithm
- Key steps:
  1. Training kNN classifier
  2. Choosing value of 'k'
  3. Distance metric

- Classification: Calculate distance between input image's feature vector and training dataset, assign majority class among 'k' nearest neighbors
- Outcome: Accurate recognition of different currency denominations for visually impaired users

CLOTHES DESCRIPTOR

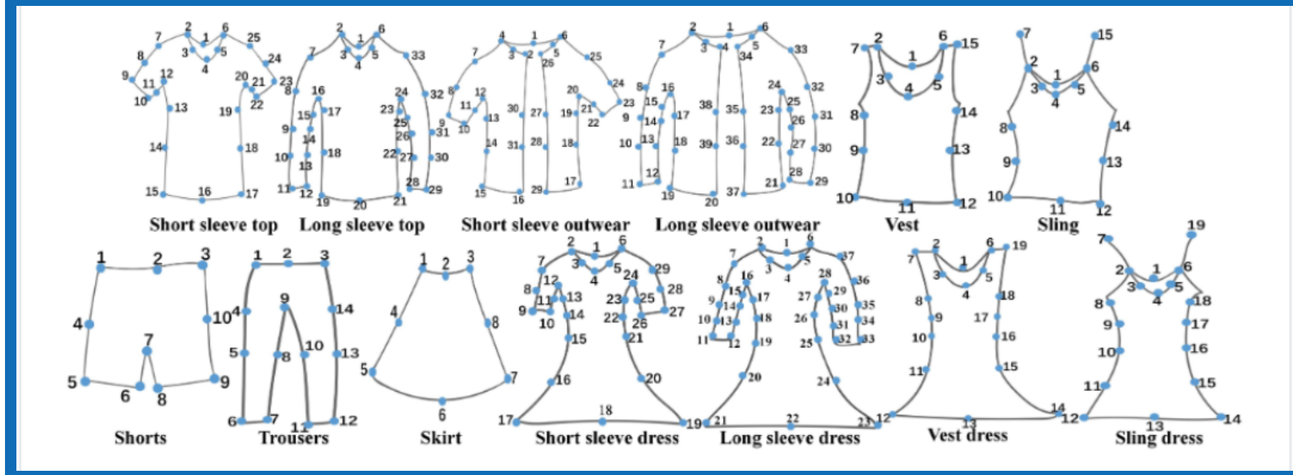
- Dataset Preparation
- Mask R-CNN
- Texture Detection
- Color Detection



MODULAR
DECOMPOSITION

CLOTHES DESCRIPTOR

- DeepFasion2 Dataset
- Clothing Detection E-commerce Dataset
- Custom Texture Dataset
- Dataset Annotation



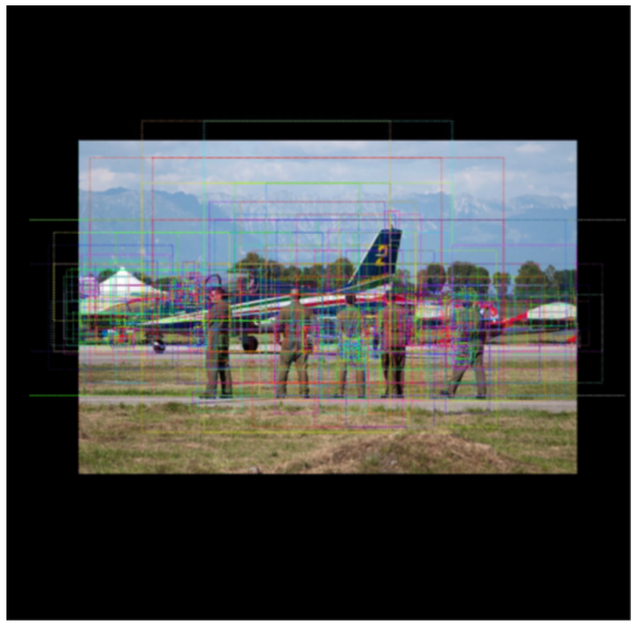
trousers	coat	jacket	belt	necklace
vest	ring	shirt	hat	bracelet
tie	bag	watch	scarf	boot
sportshoes	sunglasses	earrings	skirt	dress
socks and tight	shoes	underwear	backpack	short
night morning	gloves and mitten	wallet and purse	cufflinks	swimwear
makeup	suitcase	jumpsuit	suspenders	pouchbag

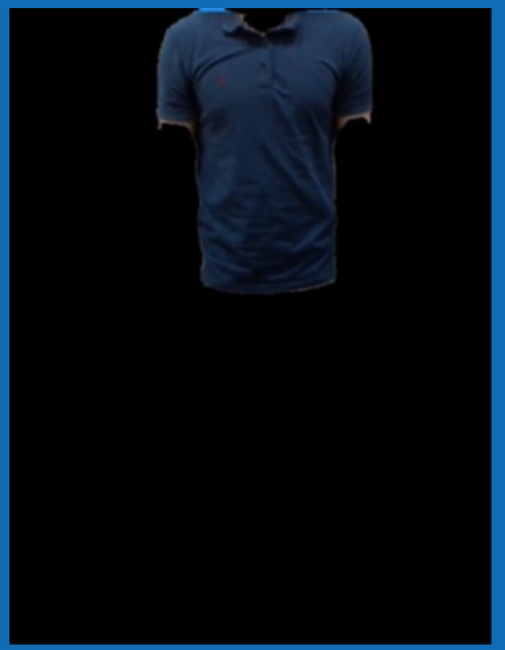
MODULAR DECOMPOSITION

CLOTHES DESCRIPTOR

MASK R-CNN

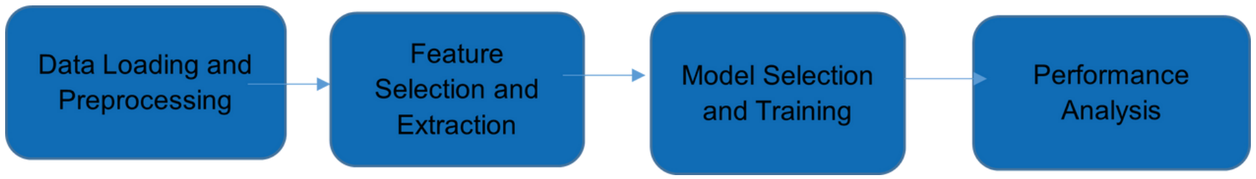
- Architecture and workflow (RPN and Head)
- Instant Segmentation





MODULAR DECOMPOSITION

CLOTHES DESCRIPTOR



TEXTURE DETECTION

- Data Loading and Preprocessing
  - Loading
  - Resizing
  - Split Shuffling
- Feature Selection and Extraction
  - GLCM
  - Daisy
  - PCA
  - Standardization
- Model Selection and Training
  - KNN
  - ANN
  - SVM
  - Random Forest
- •Performance Analysis

MODULAR
DECOMPOSITION

CLOTHES DESCRIPTOR

COLOR DETECTION

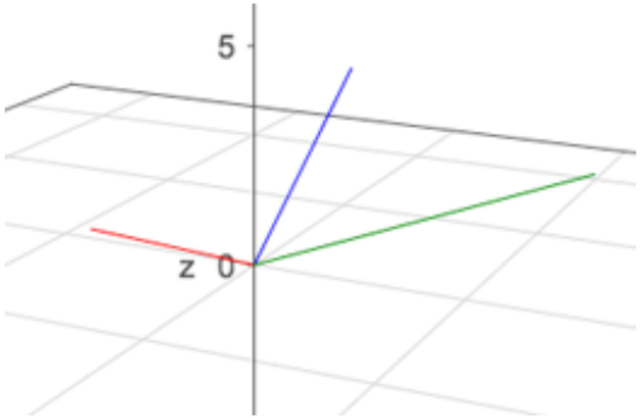
- Process segmented Image
- Extract most 3 dominant color
- Calculate distance to the ten most principal colors
- Choose colorswith minimum distance

MODULAR DECOMPOSITION

APPAREL RECOMMENDER

- Recommend to the user an outfit based on their preference
- Also follows some generic fashion rules
- TF-IDF is used to vectorize the user's wardrobe

  -Results are 3D vectors since three features were used

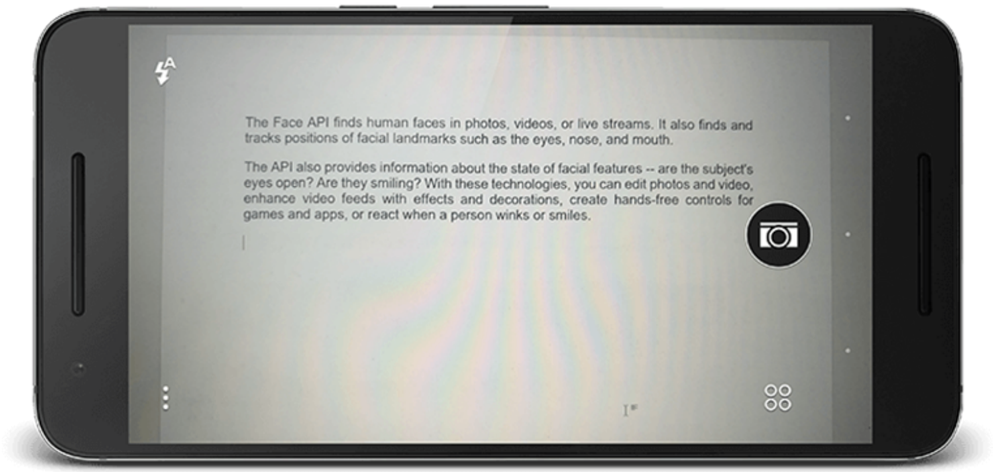


- Cosine similarity matrix is used to get the closest outfit.
- The outfit must follow some rules
- E.g. Wearing a short and a blazer together is not recommended.

	0	1	2	3	4	5
0	1.00	0.57	0.51	0.26	0.31	0.33
1	0.57	1.00	0.54	0.25	0.31	0.43
2	0.51	0.54	1.00	0.19	0.25	0.36
3	0.26	0.25	0.19	1.00	0.50	0.38
4	0.31	0.31	0.25	0.50	1.00	0.56
5	0.33	0.43	0.36	0.38	0.56	1.00

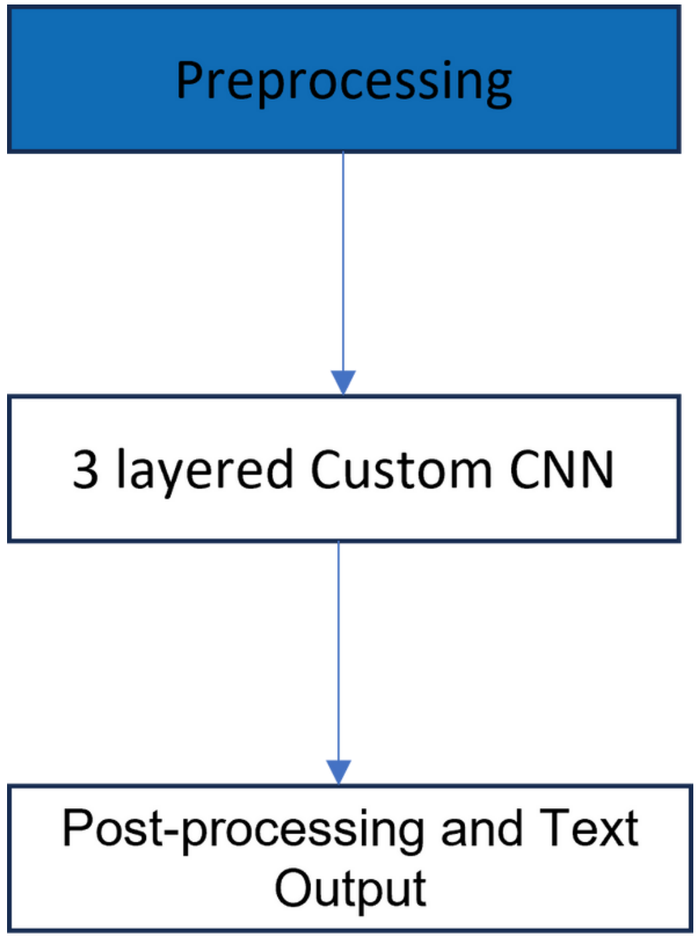
TEXT READER

- Text recognition module for visually impaired individuals
- Improves quality of life for visually impaired users
- Adaptable module recognizes various languages, fonts, sizes, and orientations
- Versatile tool for enhancing accessibility in diverse contexts
- Powerful solution for breaking down barriers and fostering a more inclusive society



MODULAR DECOMPOSITION

TEXT READER

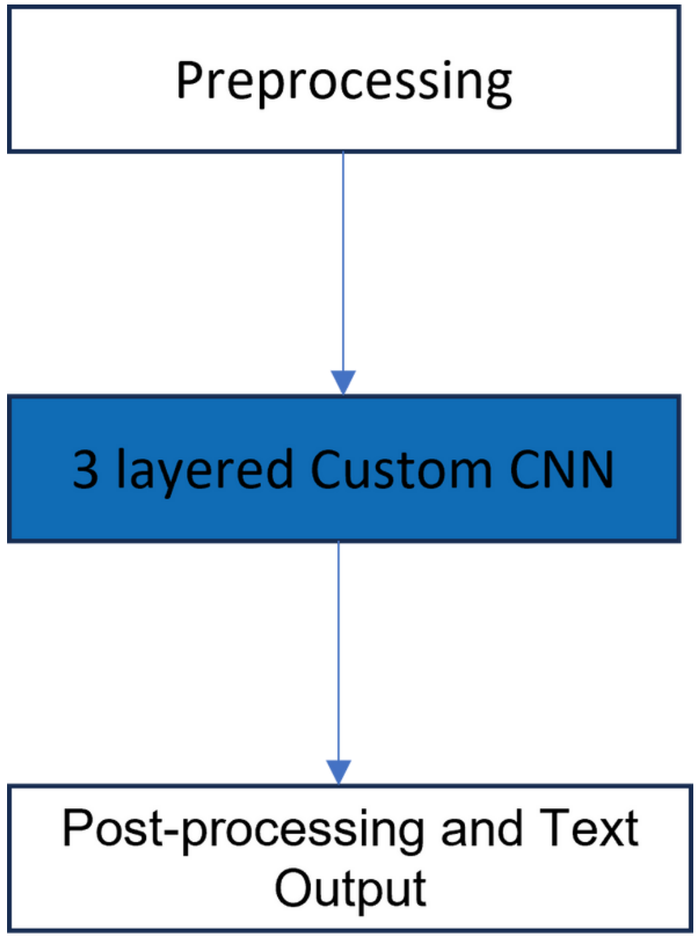


PREPROCESSING

- Stage 1: Preprocessing
- Objective: Prepare EMNIST dataset for training 3-layered CNN architecture
- Steps:
  - Removing N/A Labels: Eliminate instances with missing or not applicable character labels
  - Normalization: Normalize pixel values to a range of 0 to 1 for improved numerical stability
  - Data Augmentation: Apply rotation and lighting adjustments to enhance model robustness
  - Reshaping & Grayscale Conversion: Convert images to grayscale and reshape to match CNN input dimensions
  - Dataset Splitting: Divide preprocessed dataset into training, validation, and testing subsets using stratified sampling
- Outcomes:
  - Clean, accurate data for model training
  - Improved model resilience to various transformations and real-world scenarios

MODULAR DECOMPOSITION

TEXT READER

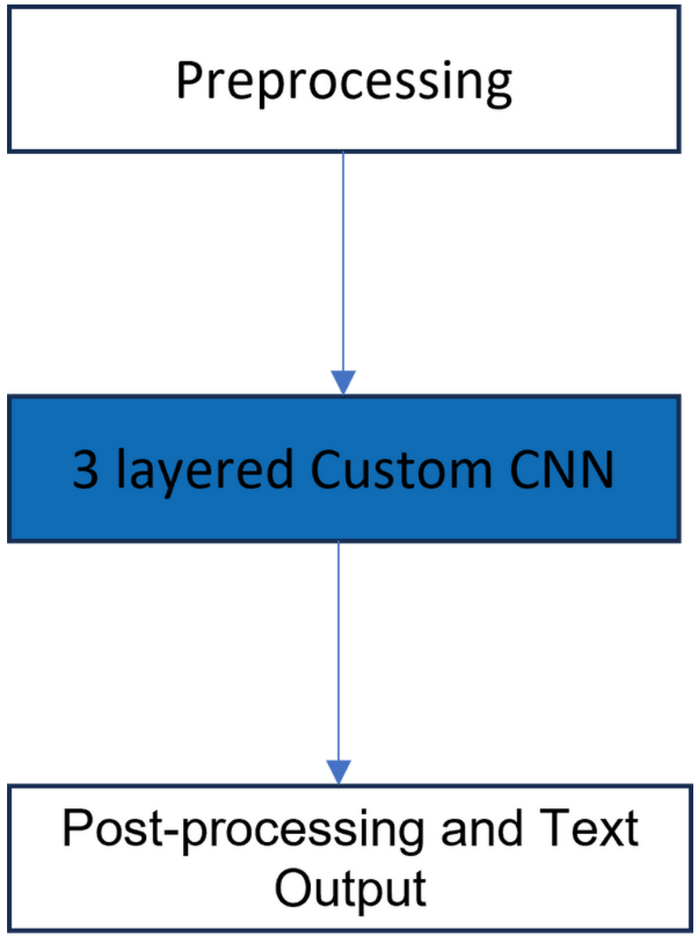


3 LAYERED CUSTOM CNN

- Stage 2: 3 layered Custom CNN
- EMNISTNet CNN Architecture
  - Custom CNN architecture for character recognition using PyTorch framework
  - Components:
    - Convolutional Layers: Detect and combine low-level to complex features
      - conv1, conv2, conv3 with increasing output channels
    - Batch Normalization Layers: Improve convergence and generalization
      - bnorm1, bnorm2, bnorm3 applied after each convolutional layer
    - Fully Connected Layers: Form abstract representation and classify images
      - fc1, fc2, fc3 with decreasing output sizes
    - Dropout Layers: Prevent overfitting (dropout rate: 0.5)
      - dropout1 and dropout2 after fc1 and fc2
  -

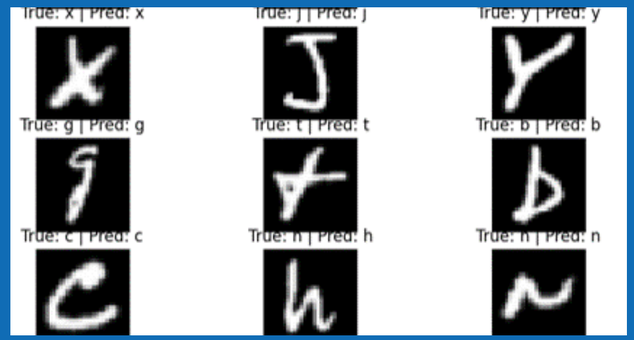
MODULAR DECOMPOSITION

TEXT READER



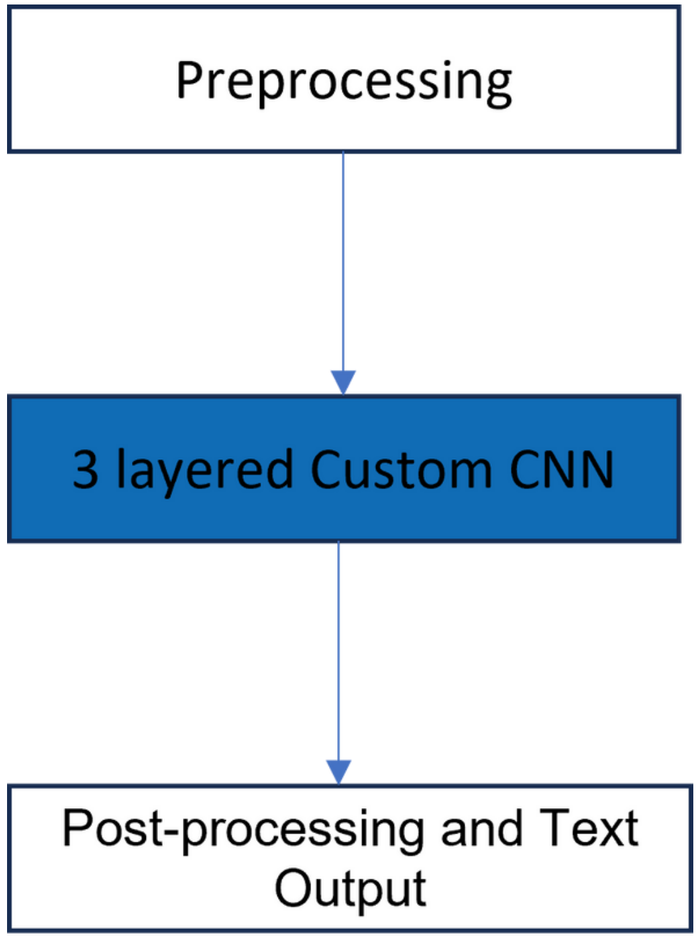
3 LAYERED CUSTOM CNN

- Stage 2: 3 layered Custom CNN
- Activation Functions and Training Process
  - Activation function: Replace ReLU with Leaky ReLU to avoid "dying ReLU" problem
  - Training process:
    - Train for 30 epochs using Adam optimizer (learning rate: 0.0005) and ReduceLROnPlateau scheduler
    - Minimize categorical cross-entropy loss with backpropagation
    - Record training and test errors and losses for performance evaluation and model selection
  - Outcome: Robust character recognition model with improved convergence and reduced overfitting
-



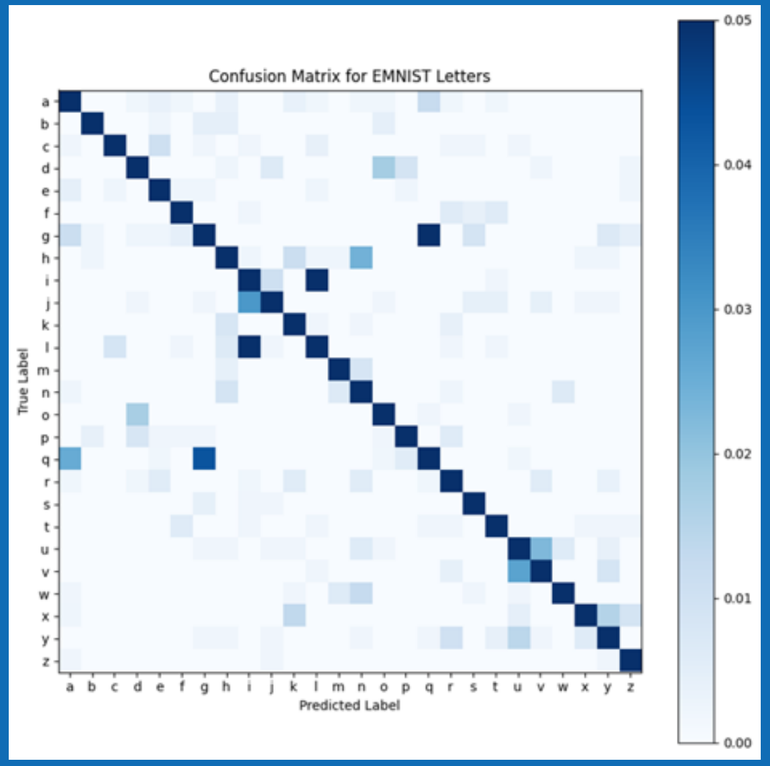
MODULAR DECOMPOSITION

TEXT READER



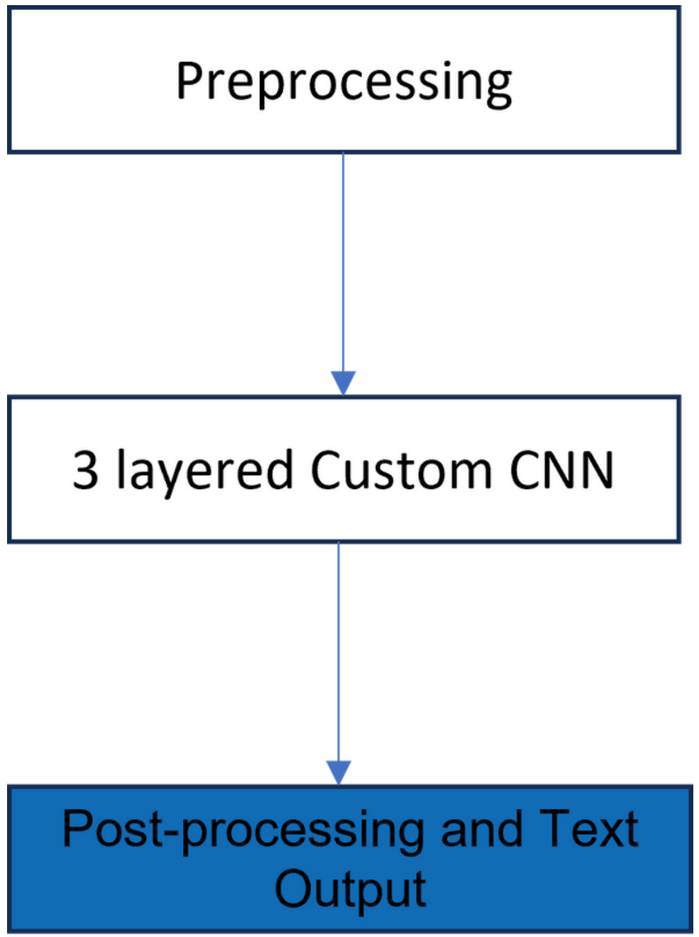
3 LAYERED CUSTOM CNN

- Stage 2: 3 layered Custom CNN
- 26x26 grid plot: Rows are true labels, columns are predicted labels, color indicates prediction proportion
- Ideal classifier: Diagonal dark blue cells (high correct predictions), light blue cells elsewhere (low incorrect predictions)



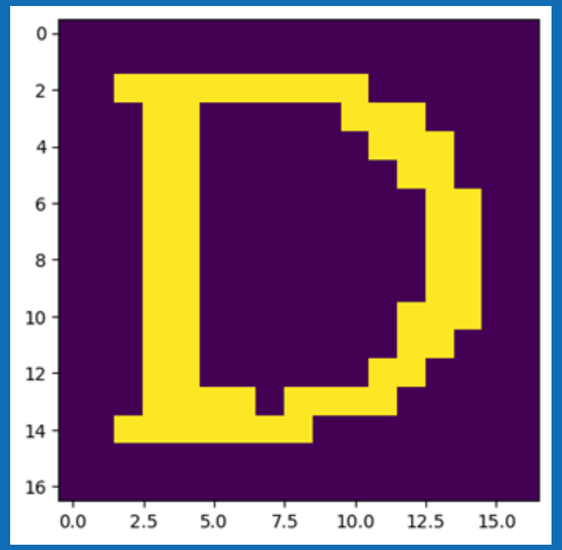
MODULAR DECOMPOSITION

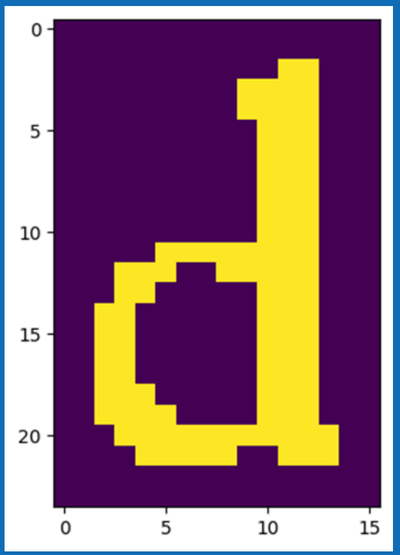
TEXT READER



3 LAYERED CUSTOM CNN

- •Stage 3: Post-processing and Text Output
- Objective: Predict characters from document images using trained CNN model
- Key steps:
  - Preprocess images
  - Add padding
  - Segment images into lines and characters
  - Skeletonize and resize characters
  - Recognize characters and extract text

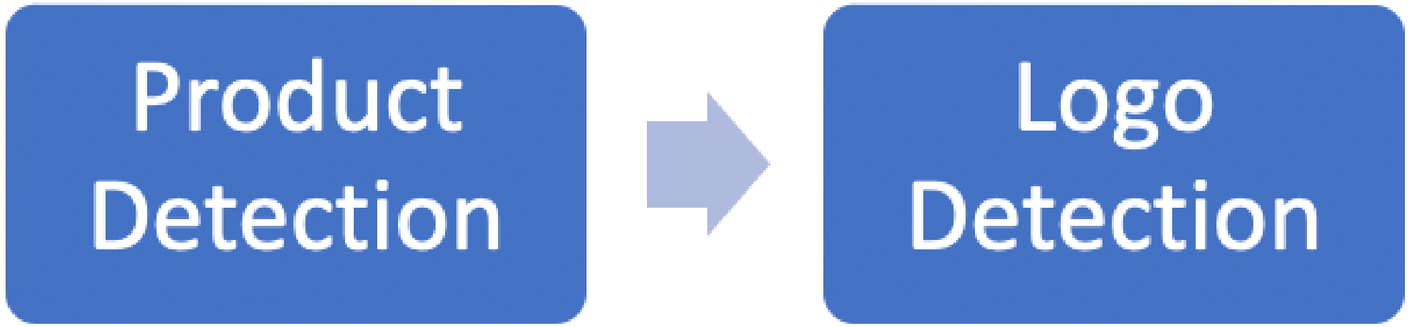




MODULAR DECOMPOSITION

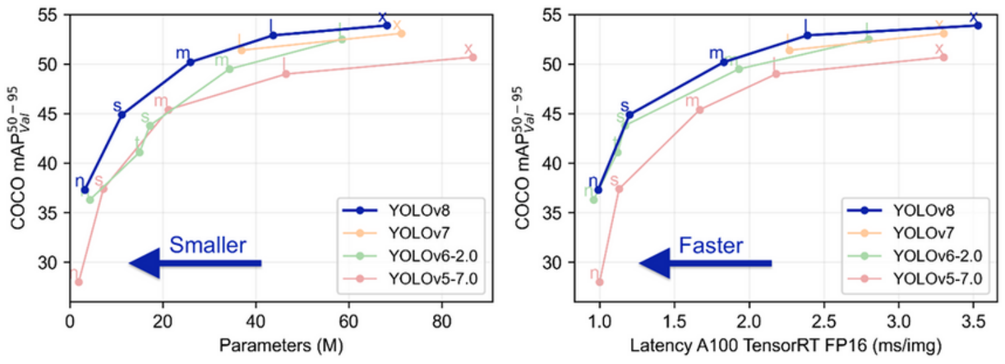
PRODUCT IDENTIFIER

- Product identification and detection algorithm
- Two cascaded CNN was to detect and identify the product.
- First CNN detects retail products based
- Second CNN detects the logo retail products

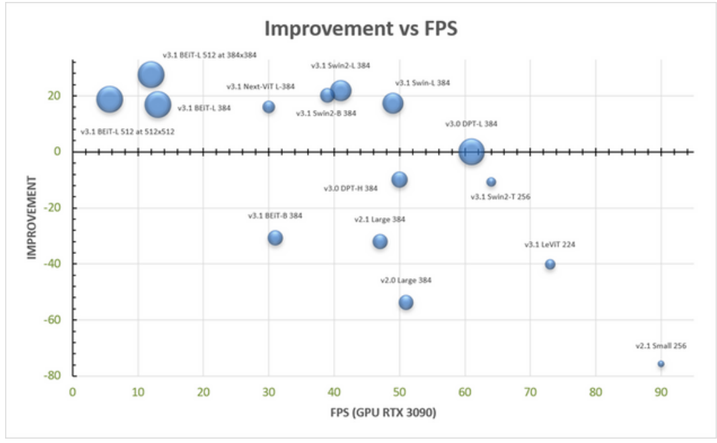


SYSTEM TESTING AND VERIFICATION

SCENE DESCRIPTOR

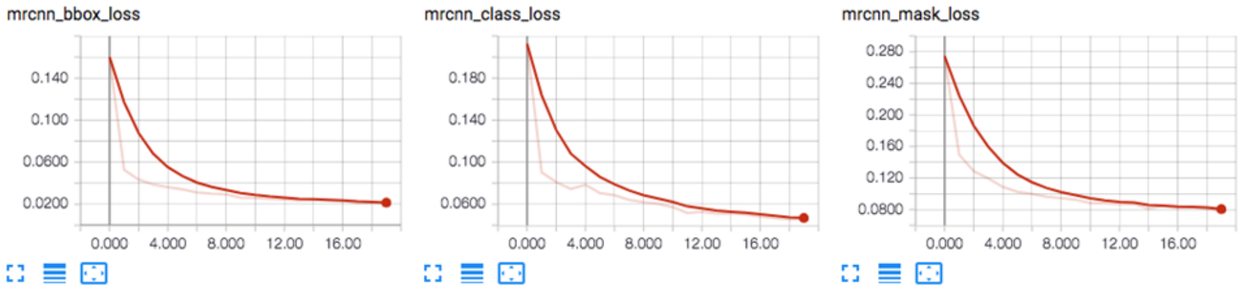


Model	size (pixels)	mAP^{box} 50-95	mAP^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1



SYSTEM TESTING AND VERIFICATION

CLOTHES DESCRIPTOR



Model: SVM - Dataset: Train
Accuracy: 58.14%
Model: SVM - Dataset: Validation
Accuracy: 58.68%
==========
Model: KNN - Dataset: Train
Accuracy: 90.49%
Model: KNN - Dataset: Validation
Accuracy: 66.12%
==========
Model: Ensemble - Dataset: Train
Accuracy: 100.0%
Model: Ensemble - Dataset: Validation
Accuracy: 94.21%
==========
Model: AdaBoost - Dataset: Train
Accuracy: 59.05%
Model: AdaBoost - Dataset: Validation
Accuracy: 55.37%
==========
Model: ANN - Dataset: Train
Accuracy: 97.81%
Model: ANN - Dataset: Validation
Accuracy: 90.91%

SYSTEM TESTING AND VERIFICATION

CURRENCY RECOGNITION

- Evaluate k-NN classifier performance for various 'k' values and distance metrics

K = 3

- 91% accuracy with Euclidean distance
- 89% accuracy with Hamming distance

K = 5

- 92% accuracy with Euclidean distance
- 89% accuracy with Hamming distance

K =7

- 93.5% accuracy with Euclidean distance
- 90% accuracy with Hamming distance

Best performing combination:
k = 7 with Euclidean distance metric (93.5% accuracy)

SYSTEM TESTING AND VERIFICATION

FACE DETECTION

Dataset	Accuracy
Olivetti	97%
LFW	93%

SYSTEM TESTING AND VERIFICATION

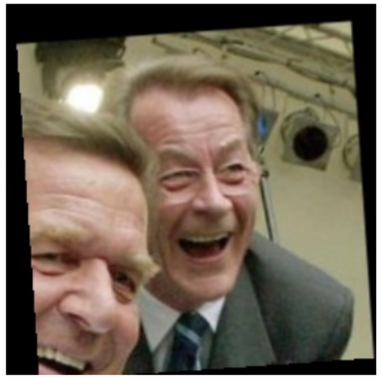
FACE RECOGNITION: EIGENFACES

Datasets:

1. Olivetti (97% accuracy) (40 classes)
2. LFW (Bad accuracy ~15%) (1473 classes)
3. Our own images

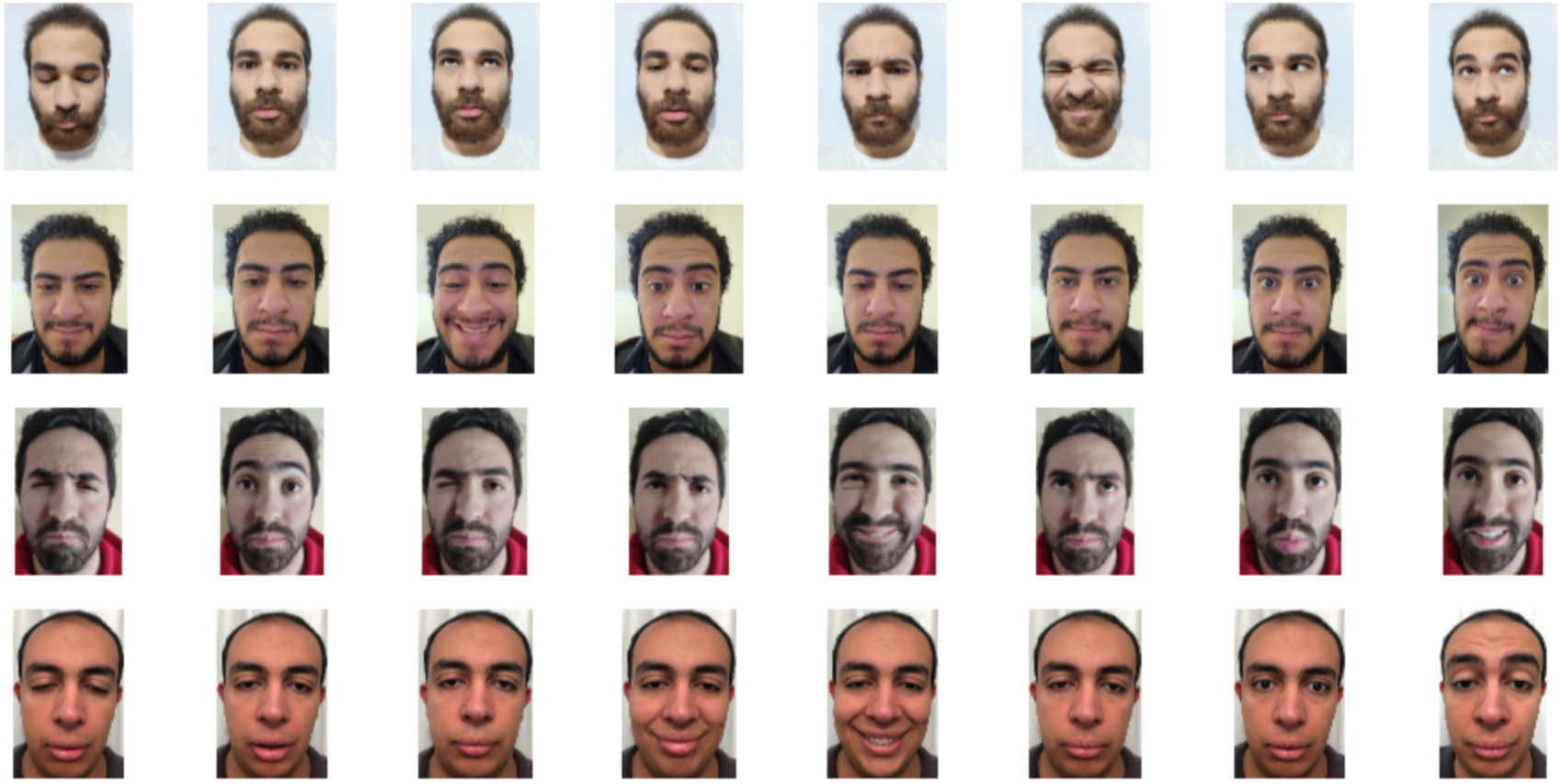







SYSTEM TESTING AND VERIFICATION

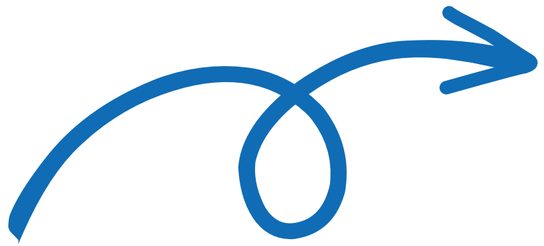
FACE RECOGNITION: EIGENFACES



SYSTEM TESTING AND VERIFICATION

EMOTION DETECTION

The model was train and tested using the Cohn-Kanade (CK and CK+) Dataset.

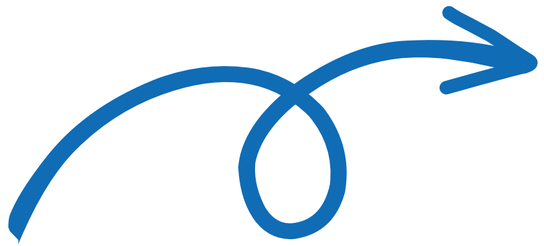


The system was able to achieve a real-time accuracy rate of 77% in detecting eight different emotions.

SYSTEM TESTING AND VERIFICATION

PRODUCT IDENTIFIER

The original dataset was divided 85% training and 15% testing.



The model reach an accuracy of 75%

SYSTEM TESTING AND VERIFICATION

APPAREL RECOMMENDER

- Testing the apparel recommender system proved to be a challenging task.
- As a result, the system was tested using user feedback as the primary evaluation metric.
- Specifically, users were asked to provide feedback on whether they liked the recommended outfits or not.
- Overall, the user feedback was positive, indicating that the system provided satisfactory recommendations.

POCKET LENS

LIMITATIONS

Overcoming Computational Challenges in System Development: Dealing with Limited Power and Large Datasets	Addressing Dataset Limitations and Real-World Performance: Optimizing System Development for Valuable Insights	Overcoming Challenges in Obtaining Feedback from Visually Impaired and Blind Individuals: Navigating Limited Accessibility and Engagement

FUTURE WORK

IMPROVING PRODUCT RECOGNITION FOR VISUALLY IMPAIRED USERS

There is an area for improvement in the product identifier module, as accurately recognizing retail products.

ENHANCING APPLICATION FUNCTIONALITY

Additionally, expanding the application's functionality to include a complete virtual assistant would be a significant improvement.

CONLUSION

Creating a machine learning system to assist visually impaired individuals presented both challenges and rewards. Overcoming obstacles such as limited computing power and dataset diversity, we successfully developed a reliable tool that has had a profound impact on the lives of people worldwide.

THANK YOU

Any questions?