**CNN Image Classification – Glasses, Earrings, Watch**

## 1. Introduction

Image classification is one of the most common and important applications in computer vision. It involves categorizing images into predefined classes based on their visual content. Traditional machine learning methods often relied on handcrafted features, but deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized this field by automatically learning hierarchical features directly from raw image data.

In this project, a CNN model was designed and trained to classify images into three categories: Glasses, Earrings, and Watches. The dataset is relatively small, with ~35 images per class, which poses a challenge for generalization. To address this, appropriate data preprocessing and augmentation techniques were applied before training the CNN.

## 2. Objective

The main objectives of this project are:

* To design a Convolutional Neural Network (CNN) capable of classifying images into three categories.
* To apply data preprocessing techniques such as cleaning, train-validation splitting, augmentation, and normalization.
* To train the CNN on the processed dataset and evaluate its performance.
* To analyze results using metrics such as accuracy, confusion matrix, precision, recall, and F1-score.
* To organize the entire workflow into a structured GitHub repository containing the raw dataset, augmented dataset, codes, trained model, evaluation results, and a detailed report.

## 

## 

## 3. Methodology

### 3.1 Dataset Preparation

* Dataset contained 105 images (35 per class: Glasses, Earrings, Watches).
* Images were cleaned:  
  + Non-image files were removed.
  + Filenames were normalized to avoid errors.
* The dataset was split into 80% training and 20% validation to ensure fair evaluation.

### 3.2 Data Preprocessing

To improve generalization, preprocessing steps included:

1. Normalization: Pixel values scaled from [0,255] → [0,1].
2. Data Augmentation (training set only):  
   * Rotation: ±20°
   * Width/Height shift: ±10%
   * Shear: ±10%
   * Zoom: ±20%
   * Horizontal Flip  
      This increased dataset diversity and reduced overfitting risk.

### 3.3 Model Training

* Implemented in TensorFlow/Keras.
* Training Parameters:  
  + Optimizer: Adam
  + Loss: Categorical Crossentropy
  + Batch size: 16
  + Epochs: 25
* Dropout was used to reduce overfitting.

### 3.4 Evaluation Metrics

The model was evaluated using:

* Accuracy (training & validation)
* Confusion Matrix
* Classification Report (precision, recall, F1-score)
* Training Curves (accuracy and loss vs epochs)

## 4. Model Architecture

The CNN model consists of:

1. Conv2D (32 filters, 3×3 kernel, ReLU activation)
2. MaxPooling2D (2×2)
3. Conv2D (64 filters, 3×3 kernel, ELU activation)
4. MaxPooling2D (2×2)
5. Conv2D (128 filters, 3×3 kernel, ReLU activation)
6. MaxPooling2D (2×2)
7. Flatten
8. Dense (128 units, ELU activation)
9. Dropout (0.5)
10. Dense (3 units, Softmax activation)

This architecture balances simplicity (suitable for small datasets) with enough depth to capture image patterns.

## 5. Results

### 5.1 Training and Validation Accuracy

* Training Accuracy: ~100%
* Validation Accuracy: ~86%

Training accuracy reached 100%, while validation accuracy plateaued at ~86%, suggesting mild overfitting but still strong performance given the limited dataset size.

### 5.2 Confusion Matrix

The confusion matrix showed that the model correctly classified most images. Misclassifications occurred mainly between visually similar categories.

### 5.3 Classification Report

* Precision: High across all classes, indicating low false positives.
* Recall: Slightly lower for some classes, indicating a few missed samples.
* F1-Score: Balanced performance across categories.

### 5.4 Training Curves

* Accuracy curve showed steady growth.
* Loss curve decreased consistently.
* A small gap between training and validation curves suggested overfitting but within acceptable limits.

## 

## 6. Conclusion

This project successfully demonstrated the use of a Convolutional Neural Network (CNN) for classifying images into Glasses, Earrings, and Watches. Despite the small dataset size, preprocessing and augmentation significantly improved generalization.

Key Achievements:

* Designed and implemented a CNN architecture suitable for small datasets.
* Applied effective preprocessing: cleaning, augmentation, normalization.
* Achieved ~86% validation accuracy.
* Organized results into a structured GitHub repository.

Limitations & Future Improvements:

* The dataset size was limited, which led to mild overfitting.
* Performance could be further improved using:  
  + Early stopping & learning rate scheduling
  + Transfer learning with pre-trained models like VGG16 or MobileNet
  + Collecting a larger dataset